RTI FPGA Programming Blockset

FPGA Handcode 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

Contents

About This Reference 11

General Information on the I/O Functions Available with FPGA Frameworks 13

Overview of the Frameworks Available for MicroLabBox ... 14
Overview of the DS2655 FPGA Base Board Frameworks ... 17
Overview of the DS6601 FPGA Base Board Frameworks ... 19
Overview of the DS6602 FPGA Base Board Frameworks ... 21
Overview of the DS2655M1 I/O Module Framework ... 23
Overview of the DS2655M2 I/O Module Framework ... 25
Overview of the DS6651 Multi-I/O Module Framework ... 27
Overview of the DS660X_MGT Framework ... 29
Overview of the Inter-FPGA Interface Framework ... 30
Overview of the DS5203 Onboard I/O Frameworks ... 32
Overview of the DS5203M1 Multi-I/O Module Frameworks of DS5203 ... 34
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) ... 37
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) ... 40

I/O Functions of the DS1202 FPGA I/O Type 1 Framework 45

ADC (Class 1) ... 46
ADC (Class 2) ... 48
Buffer In ... 49
Buffer64 In ... 51
Buffer Out ... 53
Buffer64 Out ... 55
Buzzer ... 57
DAC (Class 1) ... 58
Digital InOut (Class 1) ... 60
Digital InOut (Class 2) ... 62
Interrupt ... 65
LED Out ... 65
Proc App Status ... 66

Register In..... 67
Register64 In..... 69
Register Out..... 71
Register64 Out..... 72
Resolver..... 74
Status In..... 77
UART (RS232)..... 78
UART (RS422/485)..... 82

I/O Functions of the DS2655 FPGA Base Board Framework 87

APU Master..... 89
APU Slave..... 91
Buffer In..... 93
Buffer64 In..... 95
Buffer Out..... 98
Buffer64 Out..... 100
CN App Status..... 103
I-FPGA In (IOCNET)..... 104
I-FPGA64 In (IOCNET)..... 105
I-FPGA Out (IOCNET)..... 107
I-FPGA64 Out (IOCNET)..... 109
Interrupt..... 111
IOCNET Global Time..... 112
LED Out..... 113
Register In..... 114
Register64 In..... 115
Register Out..... 117
Register64 Out..... 118
Status In..... 120

I/O Functions of the DS6601 FPGA Base Board Framework 123

APU Master..... 125
APU Slave..... 127
Buffer In..... 129
Buffer64 In..... 131
Buffer Out..... 133
Buffer64 Out..... 136
CN App Status..... 138

I-FPGA In (IOCNET)..... 139
I-FPGA64 In (IOCNET)..... 141
I-FPGA Out (IOCNET)..... 143
I-FPGA64 Out (IOCNET)..... 145
Interrupt..... 147
IOCNET Global Time..... 148
LED Out..... 149
Register In..... 150
Register64 In..... 151
Register Out..... 153
Register64 Out..... 154
Status In..... 156

I/O Functions of the DS6602 FPGA Base Board Framework 159

APU Master..... 161
APU Slave..... 163
Buffer In..... 165
Buffer64 In..... 167
Buffer Out..... 170
Buffer64 Out..... 172
CN App Status..... 175
DDR4 32 Mode 1..... 176
DDR4 32 Mode 2..... 178
DDR4 64 Mode 1..... 181
DDR4 64 Mode 2..... 183
I-FPGA In (IOCNET)..... 186
I-FPGA64 In (IOCNET)..... 188
I-FPGA Out (IOCNET)..... 189
I-FPGA64 Out (IOCNET)..... 191
Interrupt..... 193
IOCNET Global Time..... 194
LED Out..... 195
Register In..... 196
Register64 In..... 197
Register Out..... 199
Register64 Out..... 201
Status In..... 202

I/O Functions of the DS2655M1 I/O Module Framework 205

Analog In 206
Analog Out 208
Digital In 210
Digital InOut 211
Digital Out 214

I/O Functions of the DS2655M2 I/O Module Framework 217

Digital In 218
Digital Out 221
Digital Out-Z 224
RS232 Rx 227
RS232 Tx 228
RS485 Rx 230
RS485 RxTx 232
RS485 Tx 235

I/O Functions of the DS6651 Multi-I/O Module Framework 239

Analog In 240
Analog In-L 244
Analog Out 247
Analog Out-T 248
Digital In 250
Digital In/Out-Z 251
Digital Out 255
Digital Out-Z 257
RS485 Rx 260
RS485 Rx/Tx 262
RS485 Tx 265
Trigger 267

I/O Functions of the DS660X_MGT Framework 269

Aurora 64b66b In 270
Aurora 64b66b Out 271
Aurora 64b66b 128 Bit In 273
Aurora 64b66b 128 Bit Out 275

MGT In 277
MGT Out 278

I/O Functions of the Inter-FPGA Interface Framework 281

I-FPGA In 282
I-FPGA Out 287

I/O Functions of the DS5203 with Onboard I/O Frameworks 291

ADC 293
APU Master 294
APU Slave 295
Buffer In 297
Buffer64 In 298
Buffer Out 300
Buffer64 Out 302
DAC 304
Digital In 305
Digital Out 307
I-FPGA Master 309
I-FPGA Slave 312
Interrupt 315
LED Out 315
Register In 316
Register64 In 318
Register Out 320
Register64 Out 321
Status In 323

I/O Functions of the DS5203M1 Multi-I/O Module Frameworks 325

ADC (M1) 326
DAC (M1) 327
Digital In (M1) 329
Digital Out (M1) 331
Sensor Supply 333

I/O Functions of the FPGA1401Tp1 with Multi-I/O Module Frameworks 335

ADC (Type A) 337
ADC (Type B) 338
Buffer In 340
Buffer64 In 341
Buffer Out 343
Buffer64 Out 345
DAC 347
Digital Crank/Cam Sensor 348
Digital In (Type A) 350
Digital In (Type B) 352
Digital Out (Type A) 354
Digital Out (Type B) 356
Inductive Zero Voltage Detector 358
Interrupt 359
LED Out 360
Register In 361
Register64 In 362
Register Out 364
Register64 Out 366
Sensor Supply 368
Status In 368
UART (RS232) 370
UART (RS422/485) 375

I/O Functions of the FPGA1401Tp1 with Engine Control I/O Module Framework 381

ADC (Type A) 382
Buffer In 385
Buffer64 In 386
Buffer Out 388
Buffer64 Out 390
Digital Crank/Cam Sensor 392
Digital In (Type B) 394
Digital Out (Type A) 396
Digital Out (Type B) 399
Inductive Zero Voltage Detector 401
Interrupt 402

Knock Sensor..... 403
LED Out..... 404
Register In..... 405
Register64 In..... 407
Register Out..... 409
Register64 Out..... 410
Status In..... 412
Temperature..... 413

I/O Functions of the FPGA1403Tp1 with Multi-I/O Module Frameworks 415

ADC (Type A)..... 417
ADC (Type B)..... 418
Buffer In..... 420
Buffer64 In..... 421
Buffer Out..... 423
Buffer64 Out..... 425
DAC..... 427
Digital Crank/Cam Sensor..... 428
Digital In (Type A)..... 430
Digital In (Type B)..... 432
Digital Out (Type A)..... 434
Digital Out (Type B)..... 436
Inductive Zero Voltage Detector..... 438
Interrupt..... 439
LED Out..... 440
Register In..... 441
Register64 In..... 442
Register Out..... 444
Register64 Out..... 446
Sensor Supply..... 447
Status In..... 448
UART (RS232)..... 450
UART (RS422/485)..... 455

I/O Functions of the FPGA1403Tp1 with Engine Control I/O Module Framework 461

ADC (Type A)..... 462
Buffer In..... 465
Buffer64 In..... 466

Buffer Out ... 468
Buffer64 Out ... 470
Digital Crank/Cam Sensor ... 472
Digital In (Type B) ... 473
Digital Out (Type A) ... 475
Digital Out (Type B) ... 478
Inductive Zero Voltage Detector ... 480
Interrupt ... 481
Knock Sensor ... 482
LED Out ... 483
Register In ... 484
Register64 In ... 486
Register Out ... 487
Register64 Out ... 489
Status In ... 490
Temperature ... 491

Index 493

About This Reference

Content

This reference provides detailed information about the I/O functions provided by the FPGA handcode frameworks of your dSPACE installation. For example, it contains descriptions of the parameters and ports of the available functions.

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.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.

Symbol	Description
	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.

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 I/O Functions Available with FPGA Frameworks

Introduction

Overview of the FPGA handcode frameworks provided by the RTI FPGA Programming Blockset.

Where to go from here

Information in this section

Overview of the Frameworks Available for MicroLabBox..... 14
The DS1202 FPGA I/O Type 1 frameworks are the standard frameworks supporting MicroLabBox. They provide access to analog and digital signals, and to the internal bus buffers and registers.

Overview of the DS2655 FPGA Base Board Frameworks..... 17
The frameworks comes with the DS2655 FPGA Base Board, providing access to APU signals, and to the IOCNET buffers and registers.

Overview of the DS6601 FPGA Base Board Frameworks..... 19
The framework comes with the DS6601 FPGA Base Board, providing access to APU signals, and to the IOCNET buffers and registers.

Overview of the DS6602 FPGA Base Board Frameworks..... 21
The framework comes with the DS6602 FPGA Base Board, providing access to APU signals, and to the IOCNET buffers and registers.

Overview of the DS2655M1 I/O Module Framework..... 23
The DS2655M1 I/O Module framework comes with the DS2655M1 Multi-I/O Module, providing access to analog and digital signals.

Overview of the DS2655M2 I/O Module Framework..... 25
The DS2655M2 I/O Module framework comes with the DS2655M2 Digital I/O Module, providing access to digital signals.

Overview of the DS6651 Multi-I/O Module Framework..... 27
The DS6651 Multi-I/O Module framework comes with the DS6651 Multi-I/O Module, providing access to digital signals.

Overview of the DS660X_MGT Framework....29
The DS660X_MGT framework provides MGT communication for DS6601/DS6602 FPGA Base Boards.

Overview of the Inter-FPGA Interface Framework....30
The Inter-FPGA Interface framework provides inter-FPGA communication between SCALEXIO FPGA Base Boards.

Overview of the DS5203 Onboard I/O Frameworks....32
The DS5203 onboard I/O frameworks are the standard frameworks available for DS5203 FPGA boards (7K325 and 7K410). The frameworks provide access to analog and digital signals, and to the PHS-bus buffers and registers.

Overview of the DS5203M1 Multi-I/O Module Frameworks of DS5203....34
The DS5203M1 Multi-I/O module frameworks are the standard frameworks available for the DS5203 FPGA Boards (7K325 and 7K410) with a DS5203M1 Multi-I/O module. The frameworks provide the module-specific I/O functions and the functions provided by the DS5203 FPGA Board. They include access to the PHS-bus buffers and registers.

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)....37
The FPGA1401Tp1 frameworks are the standard frameworks for MicroAutoBox II with DS1552 or DS1552B1 Multi-I/O Module. The frameworks provide access to analog and digital signals, and to the intermodule-bus buffers and registers.

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)....40
The FPGA1403Tp1 frameworks are the standard frameworks for MicroAutoBox III with DS1552 or DS1552B1 Multi-I/O Module. The frameworks provide access to analog and digital signals, and to the intermodule-bus buffers and registers.

Overview of the Frameworks Available for MicroLabBox

Introduction

The *DS1202 FPGA I/O Type 1* frameworks are the standard frameworks supporting MicroLabBox. They provide access to analog and digital signals, and to the internal bus buffers and registers.

Framework location

Depending on the use case, there are two frameworks:

- Framework to handcode a custom FPGA application without using the standard I/O features:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS1302_XC7K325T`

- Framework to handcode a custom FPGA application that additionally supports the standard I/O features to use remaining I/O channels with the RTI blocksets/Real-Time Libraries (RTLib) for MicroLabBox:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS1302_XC7K325T_FLEXIBLEIO`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_DS1302.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

For each I/O function category a specific range of channels is reserved. With the I/O function number, you can specify a specific I/O function and its corresponding channel.

Functions for exchanging data with the processor application

The following I/O functions can be used to exchange data with the processor application.

I/O Function	Available Channels	I/O Function Numbering
Register In	256	1 … 256
Buffer In	32	257 … 288
Register64 In[1)]	256	289 … 544
Buffer64 In[1)]	32	545 … 576
Register Out	256	1 … 256
Buffer Out	32	257 … 288
Register64 Out[1)]	256	289 … 544
Buffer64 Out[1)]	32	545 … 576

[1)] All 64-bit fixed-point data types are converted to double in the processor model. Therefore, the fixed-point resolution of double is restricted to 53 bits.

Functions for exchanging data with the I/O module

The following I/O functions can be used to exchange data with the I/O of MicroLabBox's DS1302 board.

I/O Function	Available Channels	I/O Function Numbering
ADC (Class 1)	24	2 … 25
ADC (Class 2)	8	26 … 33
Digital InOut (Class 1)	48	5 … 52
DAC (Class 1)	16	53 … 68
Digital InOut (Class 2)	12	73 … 84

Additional I/O functions for special purposes

The following I/O functions can be used for special purposes.

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	32	1 … 32
Status In	To get information on the state of the FPGA programming sequence.	1	1
Proc App Status	To get information on the state of the processor application.	1	34
LED Out	To set the FPGA status LEDs near the I/O connectors.	4	1 … 4
Buzzer	To generate an acoustic signal.	1	85
UART (RS232)	To implement communication via the serial interface.	2	69 … 70
UART (RS422/485)		2	71 … 72
Resolver	To get the rotor's position via a resolver sensor.	2	35 ... 36

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS1202 FPGA I/O Type 1 Framework on page 45. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *Fct(<IOFunction_Number>).Parameter(1).Init / Binary point position* describes the **Binary point position** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL or Verilog code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL or Verilog code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

I/O Functions of the DS1202 FPGA I/O Type 1 Framework.....45

Overview of the DS2655 FPGA Base Board Frameworks

Introduction

The frameworks comes with the DS2655 FPGA Base Board, providing access to APU signals, and to the IOCNET buffers and registers.

Each variant of the DS2655 FPGA Base Board is supported by its own framework:

- *DS2655 (7K160) FPGA Base Board* framework
- *DS2655 (7K410) FPGA Base Board* framework

Framework location

The location of the frameworks depends on the used variant of the DS2655 FPGA Base Boards:

- DS2655 (7K160) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K160T`

- DS2655 (7K410) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K410T`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_DS2655.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for exchanging data with the processor application

The following I/O functions can be used to exchange data with the processor application, either with 32 bit or with 64 bit data width.

I/O Function	Available Channels	I/O Function Numbering
Register In	256	1 … 256
Register64 In[1)]	256	289 … 544
Register Out	256	1 … 256
Register64 Out[1)]	256	289 … 544
Buffer In	32	257 … 288
Buffer64 In[1)]	32	545 … 576
Buffer Out	32	257 … 288
Buffer64 Out[1)]	32	545 … 576

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

For detailed information, refer to I/O Functions of the DS2655 FPGA Base Board Framework on page 87.

Additional I/O functions for special purposes

The following I/O functions can be used for special purposes.

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	8	1 … 8
Status In	To get information on the state of the FPGA programming sequence.	1	1
CN App Status	To get information on the state of the controller application.	1	2
LED Out	To set the LED on the board's bracket.	1	1
IOCNET Global Timer	To implement angle-based applications.	1	3
APU Master		6	2 … 7
APU Slave		6	4 … 9
I-FPGA In (IOCNET)	To implement inter-FPGA communication between FPGA base boards via IOCNET.	32	10 … 41
I-FPGA64 In (IOCNET)		32	42 … 73
I-FPGA Out (IOCNET)		32	8 … 39
I-FPGA64 Out (IOCNET)		32	40 … 71

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS2655 FPGA Base Board Framework on page 87. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *Fct(<IOFunction_Number>).Parameter(1).Init / Binary point position* describes the **Binary point position** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the DS6601 FPGA Base Board Frameworks

Introduction

The *DS6601 (KU035) FPGA Base Board* framework comes with the DS6601 FPGA Base Board, providing access to APU signals, and to the IOCNET buffers and registers.

Framework location

The framework is located in the follwing folder:

- `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6601_XCKU035`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_DS6601_XCKU035.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for exchanging data with the processor application

The following I/O functions can be used to exchange data with the processor application, either with 32 bit or with 64 bit data width.

I/O Function	Available Channels	I/O Function Numbering
Register In	256	1 … 256
Register64 In[1)]	256	289 … 544
Register Out	256	1 … 256
Register64 Out[1)]	256	289 … 544
Buffer In	32	257 … 288
Buffer64 In[1)]	32	545 … 576
Buffer Out	32	257 … 288
Buffer64 Out[1)]	32	545 … 576

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

For detailed information, refer to I/O Functions of the DS6601 FPGA Base Board Framework on page 123.

Additional I/O functions for special purposes

The following I/O functions can be used for special purposes.

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	16	1 … 16
Status In	To get information on the state of the FPGA programming sequence.	1	1
CN App Status	To get information on the state of the controller application.	1	2
LED Out	To set the LED on the board's bracket.	1	1
IOCNET Global Timer	To implement angle-based applications.	1	3
APU Master		6	2 … 7
APU Slave		6	4 … 9
I-FPGA In (IOCNET)	To implement inter-FPGA communication between FPGA base boards via IOCNET.	32	11 … 42
I-FPGA64 In (IOCNET)		32	43 … 74
I-FPGA Out (IOCNET)		32	8 … 39
I-FPGA64 Out (IOCNET)		32	40 … 71

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS6601 FPGA Base Board Framework on page 123. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *Fct(<IOFunction_Number>).Parameter(1).Init / Binary point position* describes the **Binary point position** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the DS6602 FPGA Base Board Frameworks

Introduction

The *DS6602 (KU15P) FPGA Base Board* framework comes with the DS6602 FPGA Base Board, providing access to APU signals, and to the IOCNET buffers and registers.

Framework location

The framework is located in the follwing folder:

- `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6602_XCKU15P`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_DS6602_XCKU15P.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for exchanging data with the processor application

The following I/O functions can be used to exchange data with the processor application, either with 32 bit or with 64 bit data width.

I/O Function	Available Channels	I/O Function Numbering
Register In	256	1 … 256
Register64 In[1)]	256	289 … 544
Register Out	256	1 … 256
Register64 Out[1)]	256	289 … 544
Buffer In	32	257 … 288
Buffer64 In[1)]	32	545 … 576
Buffer Out	32	257 … 288
Buffer64 Out[1)]	32	545 … 576

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

For detailed information, refer to I/O Functions of the DS6602 FPGA Base Board Framework on page 159.

Additional I/O functions for special purposes

The following I/O functions can be used for special purposes.

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	16	1 … 16
Status In	To get information on the state of the FPGA programming sequence.	1	1
CN App Status	To get information on the state of the controller application.	1	2
LED Out	To set the LED on the board's bracket.	1	1
IOCNET Global Timer	To implement angle-based applications.	1	3
APU Master		6	2 … 7
APU Slave		6	4 … 9
DDR4 32 Mode 1	To provide read/write access to the DDR4 RAM.	1	8
DDR4 32 Mode 2		1	9
DDR4 64 Mode 1		1	10
DDR4 64 Mode 2		1	11
I-FPGA In (IOCNET)	To implement inter-FPGA communication between FPGA base boards via IOCNET.	32	11 … 42
I-FPGA64 In (IOCNET)		32	43 … 74
I-FPGA Out (IOCNET)		32	12 … 43
I-FPGA64 Out (IOCNET)		32	44 … 75

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS6602 FPGA Base Board Framework on page 159. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *Fct(<IOFunction_Number>).Parameter(1).Init / Binary point position* describes the **Binary point position** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL code. In your Vivado project, you must adapt the related `cm` file to the required FPGA

functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the DS2655M1 I/O Module Framework

Introduction

The *DS2655M1 I/O Module* framework comes with the DS2655M1 Multi-I/O Module, providing access to analog and digital signals.

Framework location

The location of the frameworks depends on the used variant of the SCALEXIO FPGA base board:

- DS2655 (7K160) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K160T`
- DS2655 (7K410) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K410T`
- DS6601 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6601_XCKU035`
- DS6602 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6602_XCKU15P`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_<FPGA base board>.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for exchanging data with the I/O module

The DS2655M1 Multi-I/O Module is an I/O module for the SCALEXIO FPGA base boards. A SCALEXIO FPGA Base Board and one or more I/O modules mounted together and connected via ribbon cables form an FPGA board in a SCALEXIO system.

The following I/O functions can be used to exchange data with the I/O of the DS2655M1 Multi-I/O Module. Because you can use up to five I/O modules, you have to specify not only the I/O function number and the channel number to configure a specific I/O function, but also the module number. The module number is the number of the slot the I/O module is connected to.

I/O Function	Available Channels	I/O Function Numbering	Channel Numbering	Module Numbering
Analog In	5	11 … 15	1 ... 5	1 ... 5
Analog Out	5	21 … 25	6 ... 10	
Digital In	10	1 … 10	1 ... 10	
Digital InOut	10	11 … 20	1 ... 10	
Digital Out	10	1 … 10	1 ... 10	

For detailed information, refer to I/O Functions of the DS2655M1 I/O Module Framework on page 205.

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS2655M1 I/O Module Framework on page 205. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *IOProperties.In.Fct(<IOFunction_Number> + ioInOffset<Module_number>).Parameter(3).Init / Input range* describes the **Input Range** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the DS2655M2 I/O Module Framework

Introduction

The *DS2655M2 I/O Module* framework comes with the DS2655M2 Digital I/O Module, providing access to digital signals.

Framework location

The location of the frameworks depends on the used variant of the SCALEXIO FPGA base board:

- DS2655 (7K160) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K160T`
- DS2655 (7K410) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K410T`
- DS6601 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6601_XCKU035`
- DS6602 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6602_XCKU15P`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_<FPGA base board>.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Note

To use DS2655M2 Digital I/O Modules, you must additionally copy files from the RTL folder of the DS2655M2 I/O Module framework folder into your project once and customize them. Refer to Configuring the FPGA Code With the Specified I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Functions for exchanging data with the I/O module

The DS2655M2 Digital I/O Module is an I/O module for the SCALEXIO FPGA base boards. A SCALEXIO FPGA base board and one or more I/O modules mounted together and connected via ribbon cables form an FPGA board in a SCALEXIO system.

The following I/O functions can be used to exchange data with the I/O of the DS2655M2 Digital I/O Module. Because you can use up to five I/O modules, you have to specify not only the I/O function number and the channel number to configure a specific I/O function, but also the module number. The module number is the number of the slot the I/O module is connected to.

I/O Function	Available Number of I/O Functions	I/O Function Numbering	Channel Numbering	Module Numbering
Digital In	32	1 … 32	1 ... 32	1 ... 5
Digital Out	32	1 … 32	1 … 32	
Digital Out-Z	16	33 … 48	1-2, 3-4, ... , 31-32	
RS232 Rx	8	33 … 40	2, 6, ... , 30	
RS232 Tx	8	49 … 56	1, 5, ... , 29	
RS485 Rx	8	41 … 48	1-2, 5-6, ... , 29-30	
RS485 RxTx	8	65 … 72	1-3, 5-7, ... , 29-31	
RS485 Tx	8	57 … 64	1-2, 5-6, ... , 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 🕮).

For detailed information, refer to I/O Functions of the DS2655M2 I/O Module Framework on page 217.

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS2655M2 I/O Module Framework on page 217. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(8).Init / Threshold init voltage* describes the **Threshold init voltage** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the DS6651 Multi-I/O Module Framework

Introduction

The *DS6651 Multi-I/O Module* framework comes with the DS6651 Multi-I/O Module, providing access to digital signals.

Framework location

The location of the frameworks depends on the used variant of the SCALEXIO FPGA base board:

- DS2655 (7K160) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K160T`
- DS2655 (7K410) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K410T`
- DS6601 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6601_XCKU035`
- DS6602 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6602_XCKU15P`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_<FPGA base board>.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Note

To use DS6651 Multi-I/O Modules, you must additionally copy files from the RTL folder of the DS6651 Multi-I/O Module framework folder into your project once and customize them. Refer to Configuring the FPGA Code With the Specified I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Functions for exchanging data with the I/O module

The DS6651 Multi-I/O Module is an I/O module for the SCALEXIO FPGA base boards. A SCALEXIO FPGA base board and one or more I/O modules mounted together and connected via ribbon cables form an FPGA board in a SCALEXIO system.

The following I/O functions can be used to exchange data with the I/O of the DS6651 Multi-I/O Module. Because you can use up to five I/O modules, you have to specify not only the I/O function number and the channel number to configure a specific I/O function, but also the module number. The module number is the number of the slot the I/O module is connected to.

I/O Function	Available Number of I/O Functions	I/O Function Numbering	Channel Numbering	Module Numbering
Analog In	4	25 … 28	23 … 26	1 ... 5
Analog In-L	2	29, 30	27, 28	
Analog Out	4	41 … 44	17 … 20	
Analog Out-T	2	45, 46	21, 22	
Digital In	Up to 16	1 … 16	1 … 16	
Digital In/Out-Z	Up to 4	25 … 28	1, 5, 9, 13	
Digital Out	Up to 16	1 … 16	1 … 16	
Digital Out-Z	Up to 8	17 … 24	1, 3, 5, … , 15	
RS485 Rx	Up to 8	17 … 24	1, 3, 5, … , 15	
RS485 Rx/Tx	Up to 4	37 … 40	1, 5, 9, 13	
RS485 Tx	Up to 8	29 … 36	1, 3, 5, … , 15	
Trigger	2	47, 48	17, 18	

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 🕮).

For detailed information, refer to I/O Functions of the DS6651 Multi-I/O Module Framework on page 239.

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS6651 Multi-I/O Module Framework on page 239. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(8).Init / Threshold init voltage* describes the **Threshold init voltage** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function

parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the DS660X_MGT Framework

Framework location

The location of the frameworks depends on the used variant of the SCALEXIO FPGA base board:

- DS6601 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6601_XCKU035`

- DS6602 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6602_XCKU15P`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_<FPGA base board>.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for MGT communication

The following I/O functions can be used to exchange data via the MGT communication bus.

I/O Function	Available Number of Lanes	I/O Function Numbering
Aurora 64b66b In	4	1 … 4
Aurora 64b66b Out	4	1 … 4
Aurora 64b66b 128 Bit In	4	5 … 8
Aurora 64b66b 128 Bit Out	4	5 … 8
MGT In	1	9
MGT Out	1	9

For more information on inter-FPGA communication, refer to Handcoding Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS660X_MGT Framework on page 269. Additionally, the parameter and port descriptions contain more descriptive names for a simple identification of the parameters and ports. For example, *hcfw.IOProperties.In.Fct(<ChannelNumber+8> + ioInOffset(ioModuleNr)).HcCustomName / Channel name* describes the **Channel name** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For more information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For more information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Handcoding Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the Inter-FPGA Interface Framework

Framework location

The location of the frameworks depends on the used variant of the SCALEXIO FPGA base board:

- DS2655 (7K160) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K160T`
- DS2655 (7K410) FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS2655_XC7K410T`
- DS6601 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6601_XCKU035`
- DS6602 FPGA Base Board:

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS6602_XCKU15P`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file

`hc_fpga_framework_ini_<FPGA base board>.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for inter-FPGA communication

Inter-FPGA communication is a point-to-point connection between the I/O module slots of the SCALEXIO FPGA base boards.

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.

The following I/O functions can be used. Because you can use up to five I/O module slots, you have to specify the channel number to configure a specific I/O function and the used I/O module slot. The module number represents the number of the slot the I/O module is connected to.

I/O Function	Available Number of I/O Functions	Channel Numbering	Module Numbering
I-FPGA In	8	1 ... 8	1 ... 5
I-FPGA Out	8	1 ... 8	

For more information on inter-FPGA communication, refer to Handcoding Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the Inter-FPGA Interface Framework on page 281. Additionally, the parameter and port descriptions contain more descriptive names for a simple identification of the parameters and ports. For example, *hcfw.IOProperties.In.Fct(<ChannelNumber+8> + ioInOffset(ioModuleNr)).HcCustomName / Channel name* describes the **Channel name** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For more information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For more information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Handcoding Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the DS5203 Onboard I/O Frameworks

Introduction

The DS5203 onboard I/O frameworks are the standard frameworks available for DS5203 FPGA boards. The frameworks provide access to analog and digital signals, and to the PHS-bus buffers and registers.

Framework location

Depending on the FPGA type of the DS5203 FPGA board there are two frameworks.

The frameworks are stored in the following folders:

- Using DS5203 FPGA Board (7K325):

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS5203_XC7K325T`
- Using DS5203 FPGA Board (7K410):

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS5203_XC7K410T`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_DS5203.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for exchanging data with the processor application

The following I/O functions can be used to exchange data with the processor application.

I/O Function	Available Channels	I/O Function Numbering
Register In	128	1 … 128
Register64 In[1)]	128	289 ... 416
Register Out	128	1 … 128
Register64 Out[1)]	128	289 ... 416

I/O Function	Available Channels	I/O Function Numbering
Buffer In	32	129 … 160
Buffer64 In[1)]	32	545 ... 576
Buffer Out	32	129 … 160
Buffer64 Out[1)]	32	545 ... 576

[1)] All 64-bit fixed-point data types are converted to double in the processor model. Therefore, the fixed-point resolution of double is restricted to 53 bits.

For detailed information, refer to I/O Functions of the DS5203 with Onboard I/O Frameworks on page 291.

Functions for exchanging data with the I/O of the FPGA board

The following I/O functions can be used to exchange data with the I/O of the DS5203 FPGA board.

I/O Function	Available Channels	I/O Function Numbering
Digital In	16	1 … 16
Digital Out	16	1 … 16
ADC	6	17 … 22
DAC	6	17 … 22

For detailed information, refer to I/O Functions of the DS5203 with Onboard I/O Frameworks on page 291.

Additional I/O functions for special purposes

The following I/O functions can be used for special purposes.

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	8	1 … 8
Status In	To get information on the state of the FPGA programming sequence.	1	23
LED Out	To set the LED on the board's bracket.	1	23
APU Master	To implement angle-based applications by using the time-base connector.	1	24
APU Slave		1	24
I-FPGA Master	To communicate with a second FPGA board.	8	25 … 32
I-FPGA Slave		8	25 … 32

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS5203 with Onboard I/O Frameworks on page 291. The parameter and port descriptions additionally

contain more descriptive names for a simple identification of the parameters and ports. For example, *Fct(<IOFunction_Number>).Parameter(1).Init / Binary point position* describes the **Binary point position** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL or Verilog code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL or Verilog code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the DS5203M1 Multi-I/O Module Frameworks of DS5203

Introduction

The DS5203M1 Multi-I/O module frameworks are the standard frameworks available for the DS5203 FPGA Boards with a DS5203M1 Multi-I/O module. The frameworks provide the module-specific I/O functions and the functions provided by the DS5203 FPGA Boards. They include access to the PHS-bus buffers and registers.

Framework location

Based on the FPGA types of the DS5203 FPGA Boards there are two frameworks.

The frameworks are stored in the following folders:

- Using DS5203M1 Multi-I/O Module with DS5203 FPGA Board (7K325):
  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS5203_DS5203M1_XC7K325T`
- Using DS5203M1 Multi-I/O Module with DS5203 FPGA Board (7K410):
  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS5203_DS5203M1_XC7K410T`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_DS5203_DS5203M1.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for exchanging data with the processor application

The following I/O functions of the DS5203 FPGA Board can be used to exchange data with the processor application.

I/O Function	Available Channels	I/O Function Numbering
Register In	128	1 … 128
Register64 In[1)]	128	289 ... 416
Register Out	128	1 … 128
Register64 Out[1)]	128	289 ... 416
Buffer In	32	129 … 160
Buffer64 In[1)]	32	545 ... 576
Buffer Out	32	129 … 160
Buffer64 Out[1)]	32	545 ... 576

[1)] All 64-bit fixed-point data types are converted to double in the processor model. Therefore, the fixed-point resolution of double is restricted to 53 bits.

For detailed information, refer to I/O Functions of the DS5203 with Onboard I/O Frameworks on page 291.

Functions for exchanging data with the I/O of the FPGA board and its module

The following I/O functions can be used to exchange data with the I/O of the DS5203 FPGA Board and the DS5203M1 Multi-I/O Module.

I/O Function	Available Channels	I/O Function Numbering
DS5203 FPGA Board		
Digital In	16	1 … 16
Digital Out	16	1 … 16
ADC	6	17 … 22
DAC	6	17 … 22
DS5203M1 Multi-I/O Module		
Digital In (M1)	16	24 … 39
Digital Out (M1)	16	24 … 39
ADC (M1)	6	40 … 45
DAC (M1)	6	40 … 45

For detailed information, refer to I/O Functions of the DS5203 with Onboard I/O Frameworks on page 291 and I/O Functions of the DS5203M1 Multi-I/O Module Frameworks on page 325.

Additional I/O functions for special purposes

The following I/O functions can be used for special purposes.

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	8	1 … 8
Status In	To get information on the state of the FPGA programming sequence.	1	23
LED Out	To set the LED on the board's bracket.	1	23
Sensor Supply (provided by the DS5203M1 Multi-I/O Module)	To provide a supply voltage at a connected sensor.	1	46

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the DS5203 with Onboard I/O Frameworks on page 291 for the I/O functions of the base board and I/O Functions of the DS5203M1 Multi-I/O Module Frameworks on page 325 for the I/O functions of the DS5203M1 Multi-I/O Module. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *Fct(<IOFunction_Number>).Parameter(1).Init / Binary point position* describes the **Binary point position** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL or Verilog code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL or Verilog code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)

Introduction

The *FPGA1401Tp1* frameworks are the standard frameworks for MicroAutoBox II 1401/1511/1514 and MicroAutoBox II 1401/1513/1514 with one of the following I/O modules:

- DS1552 Multi-I/O Module
- DS1552B1 Multi-I/O Module
- DS1554 Engine Control I/O Module

The frameworks provide access to analog and digital signals, and to the intermodule-bus buffers and registers.

Framework location

Depending on the I/O module, there are three frameworks.

The frameworks are stored in the following folders:

- MicroAutoBox II with DS1514 and DS1552 Multi-I/O Module

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA1401Tp1_DS1552_XC7K325T`
- MicroAutoBox II with DS1514 and DS1552B1 Multi-I/O Module

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA1401Tp1_DS1552B1_XC7K325T`
- MicroAutoBox II with DS1514 and DS1554 Engine Control I/O Module

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA1401Tp1_DS1554_XC7K325T`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_FPGA1401Tp1_DS1552.m`, `hc_fpga_framework_ini_FPGA1401Tp1_DS1552B1.m`, or `hc_fpga_framework_ini_FPGA1401Tp1_DS1554_XC7K325T.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for exchanging data with the processor application

The following I/O functions can be used to exchange data with the processor application.

I/O Function	Available Channels	I/O Function Numbering
Register In	128	1 … 128
Register Out	128	1 … 128
Register64 In[1)]	128	161 … 288
Register64 Out[1)]	128	161 … 288
Buffer In	32	129 … 160
Buffer Out	32	129 … 160

I/O Function	Available Channels	I/O Function Numbering
Buffer64 In[1)]	32	289 … 320
Buffer64 Out[1)]	32	289 ... 320

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

Refer to I/O Functions of the FPGA1401Tp1 with Multi-I/O Module Frameworks on page 335 or I/O Functions of the FPGA1401Tp1 with Engine Control I/O Module Framework on page 381.

Functions for exchanging data with the I/O module

The following I/O functions can be used to exchange data with the I/O modules.

DS1552 and DS1552B1 Multi-I/O Modules

I/O Function	Available Channels	I/O Function Numbering
Digital In (Type A)	16	2 … 17
Digital In (Type B)	8	18 … 25
Digital Out (Type A)	16	2 … 17
Digital Out (Type B)	8	18 … 25
ADC (Type A)	8	26 … 33
ADC (Type B)	16	34 … 49
DAC	4	26 … 29

Refer to I/O Functions of the FPGA1401Tp1 with Multi-I/O Module Frameworks on page 335.

DS1554 Engine Control I/O Module

I/O Function	Available Channels	I/O Function Numbering
Digital In (Type B)	8	2 … 9
Digital Out (Type A)	40	2 … 41
Digital Out (Type B)	8	42 … 49
ADC (Type A)	14	10 … 23

Refer to I/O Functions of the FPGA1401Tp1 with Engine Control I/O Module Framework on page 381.

Additional I/O functions for special purposes

The following I/O functions can be used for special purposes.

DS1552 and DS1552B1 Multi-I/O Modules

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	8	1 … 8

I/O Function	Purpose	Available Channels	I/O Function Numbering
Status In	To get information on the state of the FPGA programming sequence.	1	1
LED Out	To set the FPGA status LED near the DS1514 ZIF I/O connector.	1	1
Sensor Supply	To provide a supply voltage.	1	30
UART (RS232)	To implement communication via the serial interface.	2[1)]	31 … 32
UART (RS422/485)		2[1)]	33 … 34
Digital Crank/Cam Sensor	To access digital camshaft and crankshaft sensors.	3	50 ... 52
Inductive Zero Voltage Detector	To access inductive zero voltage detectors.	1	53

[1)] To use UART 2, your DS1552 has to be modified by dSPACE.

Refer to I/O Functions of the FPGA1401Tp1 with Multi-I/O Module Frameworks on page 335.

DS1554 Engine Control I/O Module

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	8	1 … 8
Status In	To get information on the state of the FPGA programming sequence.	1	1
LED Out	To set the FPGA status LED near the DS1514 ZIF I/O connector.	1	1
Knock Sensor	To access knock sensors.	4	24 ... 27
Digital Crank/Cam Sensor	To access digital camshaft and crankshaft sensors.	5	28 ... 32
Inductive Zero Voltage Detector	To access inductive zero voltage detectors.	1	33
Temperature	To access the FPGA die temperature.	1	34

Refer to I/O Functions of the FPGA1401Tp1 with Engine Control I/O Module Framework on page 381.

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the FPGA1401Tp1 with Multi-I/O Module Frameworks on page 335 and I/O Functions of the FPGA1401Tp1 with Engine Control I/O Module Framework on page 381. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *Fct(<IOFunction_Number>).Parameter(1).Init / Binary point position* describes the **Binary point position** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL or Verilog code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL or Verilog code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)

Introduction

The *FPGA1403Tp1* frameworks are the standard frameworks for MicroAutoBox III with DS1514 and one of the following I/O modules:

- DS1552 Multi-I/O Module
- DS1552B1 Multi-I/O Module
- DS1554 Engine Control I/O Module

The frameworks provide access to analog and digital signals, and to the intermodule-bus buffers and registers.

Framework location

Depending on the I/O module, there are three frameworks.

The frameworks are stored in the following folders:

- MicroAutoBox III with DS1552 Multi-I/O Module

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA1403Tp1_DS1552_XC7K325T`

- MicroAutoBox III with DS1552B1 Multi-I/O Module

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA1403Tp1_DS1552B1_XC7K325T`
- MicroAutoBox III with DS1554 Engine Control I/O Module

  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA1403Tp1_DS1554_XC7K325T`

The entire framework folder must be copied to your working folder (see also Preparing Your Environment (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)). The included handcode FPGA framework INI file `hc_fpga_framework_ini_FPGA1403Tp1_DS1552.m`, `hc_fpga_framework_ini_FPGA1403Tp1_DS1552B1.m`, or `hc_fpga_framework_ini_FPGA1403Tp1_DS1554_XC7K325T.m` must be adapted to your specific framework configuration. You find the configuration options in this reference.

Functions for exchanging data with the processor application

The following I/O functions can be used to exchange data with the processor application.

I/O Function	Available Channels	I/O Function Numbering
Register In	128	1 … 128
Register Out	128	1 … 128
Register64 In[1)]	128	161 … 288
Register64 Out[1)]	128	161 … 288
Buffer In	32	129 … 160
Buffer Out	32	129 … 160
Buffer64 In[1)]	32	289 … 320
Buffer64 Out[1)]	32	289 ... 320

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

Refer to I/O Functions of the FPGA1403Tp1 with Multi-I/O Module Frameworks on page 415 or I/O Functions of the FPGA1403Tp1 with Engine Control I/O Module Framework on page 461.

Functions for exchanging data with the I/O module

The following I/O functions can be used to exchange data with the I/O modules.

DS1552 and DS1552B1 Multi-I/O Modules

I/O Function	Available Channels	I/O Function Numbering
Digital In (Type A)	16	2 … 17
Digital In (Type B)	8	18 … 25
Digital Out (Type A)	16	2 … 17
Digital Out (Type B)	8	18 … 25

I/O Function	Available Channels	I/O Function Numbering
ADC (Type A)	8	26 … 33
ADC (Type B)	16	34 … 49
DAC	4	26 … 29

Refer to I/O Functions of the FPGA1403Tp1 with Multi-I/O Module Frameworks on page 415.

DS1554 Engine Control I/O Module

I/O Function	Available Channels	I/O Function Numbering
Digital In (Type B)	8	2 … 9
Digital Out (Type A)	40	2 … 41
Digital Out (Type B)	8	42 … 49
ADC (Type A)	14	10 … 23

Refer to I/O Functions of the FPGA1403Tp1 with Engine Control I/O Module Framework on page 461.

Additional I/O functions for special purposes

The following I/O functions can be used for special purposes.

DS1552 and DS1552B1 Multi-I/O Modules

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	8	1 … 8
Status In	To get information on the state of the FPGA programming sequence.	1	1
LED Out	To set the FPGA status LED near the DS1514 ZIF I/O connector.	1	1
Sensor Supply	To provide a supply voltage.	1	30
UART (RS232)	To implement communication via the serial interface.	2[1)]	31 … 32
UART (RS422/485)		2[1)]	33 … 34
Digital Crank/Cam Sensor	To access digital camshaft and crankshaft sensors.	3	50 ... 52
Inductive Zero Voltage Detector	To access inductive zero voltage detectors.	1	53

[1)] To use UART 2, your DS1552 has to be modified by dSPACE.

Refer to I/O Functions of the FPGA1403Tp1 with Multi-I/O Module Frameworks on page 415.

DS1554 Engine Control I/O Module

I/O Function	Purpose	Available Channels	I/O Function Numbering
Interrupt	To implement interrupt handling.	8	1 … 8
Status In	To get information on the state of the FPGA programming sequence.	1	1
LED Out	To set the FPGA status LED near the DS1514 ZIF I/O connector.	1	1
Knock Sensor	To access knock sensors.	4	24 ... 27
Digital Crank/Cam Sensor	To access digital camshaft and crankshaft sensors.	5	28 ... 32
Inductive Zero Voltage Detector	To access inductive zero voltage detectors.	1	33
Temperature	To access the FPGA die temperature.	1	34

Refer to I/O Functions of the FPGA1403Tp1 with Engine Control I/O Module Framework on page 461.

Parameters and ports

For detailed information on the I/O functions and their parameters and ports, refer to the function descriptions in I/O Functions of the FPGA1403Tp1 with Multi-I/O Module Frameworks on page 415 and I/O Functions of the FPGA1403Tp1 with Engine Control I/O Module Framework on page 461. The parameter and port descriptions additionally contain more descriptive names for a simple identification of the parameters and ports. For example, *Fct(<IOFunction_Number>).Parameter(1).Init / Binary point position* describes the **Binary point position** parameter of the I/O function.

The port definitions are required to specify the FPGA functionality in VHDL or Verilog code. In your Vivado project, you must adapt the related `cm` file to the required FPGA functionality. For further information, refer to Specifying the FPGA Functionality (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

For each accessed port in the VHDL or Verilog code, you must configure the corresponding I/O function in the handcode FPGA framework INI file. Most of the I/O function parameters configure the function behavior. For further information, refer to Specifying the FPGA I/O Interface (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Related topics

Basics

Detailed Instructions on the Handcode Workflow (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

I/O Functions of the DS1202 FPGA I/O Type 1 Framework

Introduction

The *DS1202 FPGA I/O Type 1* frameworks `DS1302_XC7K325T` and `DS1302_XC7K325T_FLEXIBLEIO` provide the custom I/O functionality of MicroLabBox.

Where to go from here

Information in this section

ADC (Class 1)..... 46
To read data from an analog input signal in the FPGA application using the class 1 A/D conversion function.

ADC (Class 2)..... 48
To read data from an analog input signal in the FPGA application using the class 2 A/D conversion function.

Buffer In..... 49
To read data from an internal bus buffer with a data width of 32 bits.

Buffer64 In..... 51
To read data from an internal bus buffer with a data width of 64 bits.

Buffer Out..... 53
To write data to an internal bus buffer with a data width of 32 bits.

Buffer64 Out..... 55
To write data to an internal bus buffer with a data width of 64 bits.

Buzzer..... 57
To generate an acoustic signal.

DAC (Class 1)..... 58
To write data to an analog output signal in the FPGA application using the class 1 D/A conversion function.

Digital InOut (Class 1)..... 60
To read or write data to a digital I/O signal in the FPGA application using the class 1 digital I/O function.

Digital InOut (Class 2)..... 62
To read or write data to a digital I/O signal in the FPGA application using the class 2 digital I/O function.

Interrupt..... 65
To request a processor interrupt outside of the FPGA application.

LED Out..... 65
To write a digital signal that controls the color of one FPGA status LED on the board.

Proc App Status..... 66
To read the status of application that is running on the computation node.

Register In..... 67
To read data from an internal bus register with a data width of 32 bits.

Register64 In..... 69
To read data from an internal bus register with a data width of 64 bits.

Register Out..... 71
To write data to an internal bus register with a data width of 32 bits.

Register64 Out..... 72
To write data to an internal bus register with a data width of 64 bits.

Resolver..... 74
To get the rotor's position via a resolver sensor in the FPGA application.

Status In..... 77
To read a digital signal that outputs the state of the FPGA initialization sequence.

UART (RS232)..... 78
To implement communication via serial interface for RS232 UART type.

UART (RS422/485)..... 82
To implement communication via serial interface for RS422/485 UART type.

ADC (Class 1)

Purpose

To read data from an analog input signal in the FPGA application using the class 1 A/D conversion function.

Description

According to the number of physical connections available on MicroLabBox's DS1302 board, you can select the ADC (Class 1) I/O functions. There are 24 differential analog input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 25.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 23.

adctp1_<ChannelNumber>_value / Data Outputs the current results of an analog input channel.

Data type: Fix_16_0

Range: -32767 … +32767 (-10 V … +10 V)

Update rate: 1 Msps

adctp1_<ChannelNumber>_convert / 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

adctp1_<ChannelNumber>_busy / Busy 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

I/O mapping

The signals are available at the Analog In connector.

The channel numbers 00 … 23 corresponds to the channels 1 … 24.

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 🕮).

Related topics

References

ADC (Class 2) 48
Overview of the Frameworks Available for MicroLabBox 14

ADC (Class 2)

Purpose

To read data from an analog input signal in the FPGA application using the class 2 A/D conversion function.

Description

According to the number of physical connections available on MicroLabBox's DS1302 board, you can select the **ADC (Class 2)** I/O functions. There are 8 differential analog input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 26 … 33.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 7.

adctp2_<ChannelNumber>_value / Data	Outputs the current results of analog input channel. Data type: Fix_16_0 Range: -32767 … +32767 (-10 V … +10 V) Update rate: 10 Msps

I/O mapping

The signals are available at the Analog In connector.

The channel numbers 0 … 7 corresponds to the channels 1 … 8.

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 🕮).

Related topics

References

ADC (Class 1) 46
Overview of the Frameworks Available for MicroLabBox 14

Buffer In

Purpose

To read data from an internal bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is read from an internal bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 257 … 288.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf_<ChannelNumber>_count / 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).

xmemf_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an internal bus buffer. The data format depends on the related parameter settings.

xmemf_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer Out 53
Buffer64 In 51
Overview of the Frameworks Available for MicroLabBox 14

Buffer64 In

Purpose

To read data from an internal bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is read from an internal bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 545 … 576.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunction_Number>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmem64f_<ChannelNumber>_count / 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).

xmem64f_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an internal bus buffer. The data format depends on the related parameter settings.

xmem64f_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmem64f_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer In....49
Buffer64 Out....55
Overview of the Frameworks Available for MicroLabBox....14

Buffer Out

Purpose

To write data to an internal bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is written to an internal bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 257 … 288.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an internal bus buffer. The data format depends on the related parameter settings.

xmemp_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via the internal bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer In 49
Buffer64 Out 55
Overview of the Frameworks Available for MicroLabBox 14

Buffer64 Out

Purpose

To write data to an internal bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is written to an internal bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 545 … 576.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmem64p_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an internal bus buffer. The data format depends on the related parameter settings.

xmem64p_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmem64p_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via the internal bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmem64p_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer Out....53
Buffer64 In....51
Overview of the Frameworks Available for MicroLabBox....14

Buzzer

Purpose

To generate an acoustic signal.

Description

You can add the Buzzer I/O function to your application to access the board's buzzer.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is 85.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

buzzer_frequency / Frequency Specifies the period of the acoustic signal in steps of 40 µs. You calculate the frequency with the following equation:

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)

buzzer_beep_duration / 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: The beep is generated permanently

buzzer_pause_duration / 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)

buzzer_beep_count / 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

buzzer_start / 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 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

Related topics

References

Overview of the Frameworks Available for MicroLabBox..... 14

DAC (Class 1)

Purpose

To write data to an analog output signal in the FPGA application using the class 1 D/A conversion function.

Description

According to the number of physical connections available on MicroLabBox's DS1302 board, you can select the DAC (Class 1) I/O functions. There are 16 single-ended analog output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 53 … 68.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

dactp1_<ChannelNumber>_value / Data Outputs the current results of an analog output channel.

Data type: Fix_16_0

Value range: -32767 … +32767 (-10 V … +10 V)

Update rate: 2.78 Msps

dactp1_<ChannelNumber>_convert / 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

dactp1_<ChannelNumber>_busy / 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

I/O mapping

The signals are available at the Analog Out connector.

The channel numbers 00 … 15 corresponds to the channels 1 … 16.

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 🕮).

Related topics

References

Overview of the Frameworks Available for MicroLabBox..... 14

Digital InOut (Class 1)

Purpose

To read or write data to a digital I/O signal in the FPGA application using the class 1 digital I/O function.

Description

According to the number of physical connections available on MicroLabBox's DS1302 board, you can select the **Digital InOut (Class 1)** I/O functions. There are 48 single-ended digital I/O channels, which you can separately configure for input or output.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 5 … 52.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Invert values Lets you specify whether to invert the input and output values of the digital channel.

- 0: The values are not inverted.
- 1: The values are inverted.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Input filter Lets you specify the minimum pulse length for detecting a valid input in the range 0 … 10,000,000 ns.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(5).Init / High supply Lets you specify the high level voltage for the digital outputs.

- 0: 5 V
- 1: 3.3 V
- 2: 2.5 V

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(6).Init / Rising edge delay Lets you specify the delay for the rising edge detection in the range 0 … 65500 ns.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 47.

diotp1_<ChannelNumber>_ena / 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

diotp1_<ChannelNumber>_dir / 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.

diotp1_<ChannelNumber>_in / Data In Outputs the current results of the digital input channel.

Data type: UFix_1_0

- 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 🕮).

diotp1_<ChannelNumber>_out / 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).

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 🕮).

I/O mapping

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

The channel numbers 00 … 47 corresponds to the channels 1 … 48.

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 🕮)

Related topics

References

Digital InOut (Class 2) 62
Overview of the Frameworks Available for MicroLabBox 14

Digital InOut (Class 2)

Purpose

To read or write data to a digital I/O signal in the FPGA application using the class 2 digital I/O functions.

Description

According to the number of physical connections available on MicroLabBox's DS1302 board, you can select the **Digital InOut (Class 2)** I/O functions. There are 12 differential digital I/O channels, which you can separately configure for input or output.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 73 … 84.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Invert values Lets you specify whether to invert the input and output values of the digital channel.

- 0: The values are not inverted.
- 1: The values are inverted.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Input filter Lets you specify the minimum pulse length for detecting a valid input in the range 0 … 10,000,000 ns.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(6).Init / Rising edge delay Lets you specify the delay for the rising edge detection in the range 0 … 65,500 ns.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 11.

diotp1_<ChannelNumber>_ena / Enable Controls the hardware output. If set to 1, the hardware output reacts to the Data Out outport, otherwise it is set to High-Z.

Data type: UFix_1_0

diotp1_<ChannelNumber>_dir / 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.

diotp1_<ChannelNumber>_in / Data In Outputs the current results of the digital input channel.

Data type: UFix_1_0

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

diotp1_<ChannelNumber>_out / 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).

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 2 I/O (Bidirectional) (MicroLabBox Hardware Installation and Configuration 🕮).

I/O mapping

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

The channel numbers 00 … 11 corresponds to the channels 1 … 12.

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 🕮)

Related topics

References

Digital InOut (Class 1)..... 60
Overview of the Frameworks Available for MicroLabBox..... 14

Interrupt

Purpose

To request a processor interrupt outside of the FPGA application.

Description

MicroLabBox provides 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.

Parameters

The Interrupt I/O function can be used for up to 32 channels / interrupt lines. You will find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 32.

IRQProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

usr_interrupt_<ChannelNumber> / Int Provides the interrupt request line.

- 0 to 1: Interrupt is requested (edge-triggered).
- 0: No interrupt is requested. Last requested interrupt is saved.

Related topics

References

Overview of the Frameworks Available for MicroLabBox..... 14

LED Out

Purpose

To write a digital signal that controls the color of one FPGA status LED on the board.

Description

According to the number of physical connections available on MicroLabBox's DS1302 board, you can select the LED Out I/O functions. There are 4 digital output channels. For each FPGA status LED you can separately configure the RGB color value.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 4.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

led_<ChannelNumber>_red / Red Specifies the red portion of the LED's color value.

Data type: UFix8_0

Value range: 0 … 255

led_<ChannelNumber>_green / Green Specifies the green portion of the LED's color value.

Data type: UFix8_0

Value range: 0 … 255

led_<ChannelNumber>_blue / Blue Specifies the blue portion of the LED's color value.

Data type: UFix8_0

Value range: 0 … 255

Related topics

References

Overview of the Frameworks Available for MicroLabBox 14

Proc App Status

Purpose

To read the status of application that is running on the computation node.

Description

There is one digital input channel that is used for the Proc App Status I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 34.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

appl_run/ Processor Application Status Outputs the state of the application that is running on the computation node.

Data type: UFix_1_0

- 0: The application on the computation node is stopped.
- 1: The application on the computation node is running.

Related topics

References

Overview of the Frameworks Available for MicroLabBox 14

Register In

Purpose

To read data from an internal bus register with a data width of 32 bits.

Description

If you select Register as the access type, the data is read from an internal bus register. 256 registers are available with a data width of 32 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 256.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the internal bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an internal bus register. The data format depends on the related parameter settings.

xreg_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Overview of the Frameworks Available for MicroLabBox 14
Register64 In 69

Register64 In

Purpose

To read data from an internal bus register with a data width of 64 bits.

Description

If you select Register64 as the access type, the data is read from an internal bus register. 256 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 … 544.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunction_Number>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the internal bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an internal bus register. The data format depends on the related parameter settings.

xreg64_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Overview of the Frameworks Available for MicroLabBox 14
Register In 67

Register Out

Purpose

To write data to an internal bus register with a data width of 32 bits.

Description

If you select Register as the access type, the data is written to an internal bus register. 256 registers are available with a data width of 32 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 256.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 internal bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an internal bus register. The data format depends on the related parameter settings.

Related topics

References

Overview of the Frameworks Available for MicroLabBox 14
Register In 67
Register64 Out 72

Register64 Out

Purpose

To write data to an internal bus register with a data width of 64 bits.

Description

If you select **Register64** as the access type, the data is written to an internal bus register. 256 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 … 544.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Register64 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 internal bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an internal bus register. The data format depends on the related parameter settings.

Related topics

References

Overview of the Frameworks Available for MicroLabBox 14
Register Out 71
Register64 In 69

Resolver

Purpose

To get the rotor's position via a resolver sensor in the FPGA application.

Description

According to the number of physical connections available on MicroLabBox's DS1302 board, you can select the Resolver I/O functions. There are two resolver input channels.

This I/O function is not taken into account if you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 35 … 36.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Desired excitation frequency Lets you specify 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.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / Excitation RMS voltage Lets you specify the voltage level of the excitation output signal:

- 0: 3.0 V_{RMS}
- 1: 7.0 V_{RMS}
- 2: 10.0 V_{RMS}

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(4).Init / Input RMS voltage Lets you specify the voltage level of the sine and cosine input signals:

- 0: 1.5 V_{RMS}
- 1: 3.5 V_{RMS}
- 2: 5.0 V_{RMS}

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(5).Init / Maximum speed Lets you specify the maximum speed to be measured in revolutions per minute. By specifying the speed range, you set the related resolution.

- 0: Specifies a maximum speed of 150,000 rpm and a resolution of 10 bits.
- 1: Specifies a maximum speed of 60,000 rpm and a resolution of 12 bits.
- 2: Specifies a maximum speed of 30,000 rpm and a resolution of 14 bits.
- 3: Specifies a maximum speed of 7500 rpm and a resolution of 16 bits.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 1.

resolver_<ChannelNumber>_enable / Enable Enables the excitation voltage:

- 0: The resolver interface provides no excitation voltage.
- 1: The resolver interface provides the excitation voltage that you set with the Excitation RMS voltage parameter.

Data type: UFix_1_0

Data width: 1

resolver_<ChannelNumber>_mech_pos / 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

resolver_<ChannelNumber>_valid / 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

resolver_<ChannelNumber>_update / 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 only within one clock cycle.

Data type: UFix_1_0

Data width: 1

resolver_<ChannelNumber>_fault / 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 🕮).

resolver_<ChannelNumber>_err_rst / 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

I/O mapping

The signals are available at the Resolver connector.

The channel numbers 0 … 1 correspond to the channels 1 … 2.

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 🕮).

Related topics

Basics

Resolver Interface (MicroLabBox Features 🕮)

References

Overview of the Frameworks Available for MicroLabBox..... 14

Status In

Purpose

To read a digital signal that outputs the state of the FPGA initialization sequence.

Description

The DS1302 framework provides one digital input channel for the Status In I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is to be specified with 1.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

init_done/ 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.

Related topics

References

Overview of the Frameworks Available for MicroLabBox..... 14

UART (RS232)

Purpose

To implement communication via serial interface for RS232 UART type.

Description

According to the number of physical connections available on MicroLabBox's DS1302 board, you can select the **UART (RS232)** I/O function. There are two channels for this function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 69 … 70.

Most of the parameters are used for the UART (RS232) and UART (RS422/485) I/O functions.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Baud rate Lets you specify the baud rate of the UART in the range 50 … 1,000,000 baud (bits per second).

The baud rate depends on the parameters 2, 3 and 4, and can be calculated by the following formula:

```
BaudRate = (10^8 · uart_x_dcm_m) / (4 · uart_x_dcm_d
· (uart_dcm_clk_divider+1))
```

With:

Variable	Parameter	Description
uart_x_dcm_m[1)]	Parameter(2).Init	Multiplier for the digital clock manager (DCM) module in the range 2 … 255.
uart_x_dcm_d[1)]	Parameter(3).Init	Divisor for the digital clock manager (DCM) module in the range 1 … 255.
uart_x_dcm_clk_divider[1)]	Parameter(4).Init	UART clock divider in the range 0 … 262,143.

[1)] x=1 for UART 1; x=2 for UART 2

Note

Limitations:

- The maximum baud rate of 1,000,000 baud must not be exceeded.

Tip

You find the `DS1302_uart_parameters.mat` file in `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS1302_XC7K325T` that you can open in MATLAB. It contains some calculated baud rates and the percentage deviations to the supported baud rates according to the parameters m, d and the clock divider.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(5).Init / Word length Lets you specify the word length in the range 5 … 9 bit. The word length includes the number of data bits and the optional parity bit. Exceeding bits in a message are ignored at the transmitter or cleared at the receiver.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(6).Init / Stop bits Lets you specify the length of the stop bits in half of bits.

Stop Bits	Parameter Value
1	2
1.5	3
2	4

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(7).Init / UART type Lets you specify the UART type.

Value	UART Type
0	RS232
1	RS422/485

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(8).Init / Termination Lets you specify the termination state.

Note

For the RS232 UART type, the termination must be set to 0 (disconnected).

Value	Termination State	Description
0	Disconnected	▪ The RXD/CTS and TXD/RTS signals are not terminated.
1	Connected	▪ Not allowed

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 1.

There is only one section in the file that is valid for both UART types.

uart_<ChannelNumber>_rd / 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 FPGA clock cycle from the UART.

uart_<ChannelNumber>_rd_data_count / 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.

Value range: 0 … 2047

uart_<ChannelNumber>_rd_fifo_empty / 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.

Range:

- 0: The RX FIFO buffer is not empty.
- 1: The RX FIFO buffer is empty.

uart_<ChannelNumber>_rd_data / 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.

Range: 0 … 511

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).

uart_<ChannelNumber>_wr / 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.

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).

uart_<ChannelNumber>_wr_data_count / 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.

Range: 0 … 2047

uart_<ChannelNumber>_wr_fifo_full / 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.

Range:

- 0: The TX FIFO buffer is not full.
- 1: The TX FIFO buffer is full.

uart_<ChannelNumber>_wr_data / Write Data Specifies the value to be send.

The Write_Data signal is transferred at each clock cycle with Write_Enable set to 1.

Range: 0 … 511

uart_<ChannelNumber>_rts / 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.

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).

uart_<ChannelNumber>_cts / 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.

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).

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 🕮)

Related topics

References

Overview of the Frameworks Available for MicroLabBox 14
UART (RS422/485) 82

UART (RS422/485)

Purpose

To implement communication via serial interface for RS422/485 UART type.

Description

According to the number of physical connections available on MicroLabBox's DS1302 board, you can select the I/O function **UART (RS422/485)**. There are two channels for this function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 71 … 72.

Most of the parameters are used for the UART (RS232) and UART (RS422/485) I/O functions.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Baud rate Lets you specify the baud rate of the UART in the range 50 … 10,000,000 baud (bits per second).

The baud rate depends on the parameters 2, 3 and 4, and can be calculated by the following formula:

```
BaudRate = (10^8 · uart_x_dcm_m) / (4 · uart_x_dcm_d · (uart_dcm_clk_divider+1))
```

With:

Variable	Parameter	Description
uart_x_dcm_m[1)]	Parameter(2).Init	Multiplier for the digital clock manager (DCM) module in the range 2 … 255.
uart_x_dcm_d[1)]	Parameter(3).Init	Divisor for the digital clock manager (DCM) module in the range 1 … 255.
uart_x_dcm_clk_divider[1)]	Parameter(4).Init	UART clock divider in the range 0 … 262,143.

[1)] x=1 for UART 1; x=2 for UART 2

Tip

You find the `DS1302_uart_parameters.mat` file in `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\DS1302_XC7K325T` that you can open in MATLAB. It contains some calculated baud rates and the percentaged deviations to the supported baud rates according to the parameters m, d and the clock divider.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(5).Init / Word length Lets you specify the word length in the range 5 … 9 bit. The word length includes the number of data bits and the optional parity bit. Exceeding bits in a message are ignored at the transmitter or cleared at the receiver.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(6).Init / Stop bits Lets you specify the length of the stop bits in half of bits.

Stop Bits	Parameter Value
1	2
1.5	3
2	4

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(7).Init / UART mode Lets you specify the mode when using the RS422/485 UART type.

Value	UART Mode
0	Full-duplex mode
1	Half-duplex mode

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(8).Init / UART type Lets you specify the UART type.

Value	UART Type
0	RS232
1	RS422/485

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(9).Init / Termination Lets you specify the termination state.

Value	Termination State	Description
0	Disconnected	▪ Full-duplex mode: RX-/RX+ and TX-/TX+ signals are not terminated. ▪ Half-duplex mode: BM/BP signals are not terminated.
1	Connected	▪ Full-duplex mode: RX-/RX+ and TX-/TX+ signals are terminated via 120 Ω resistors. ▪ Half-duplex mode: BM/BP signal are terminated via a 120 Ω resistor.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 1.

There is only one section in the file that is valid for both UART types.

uart_<ChannelNumber>_rd / 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 FPGA clock cycle from the UART.

uart_<ChannelNumber>_rd_data_count / 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.

Value range: 0 … 2047

uart_<ChannelNumber>_rd_fifo_empty / 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.

Range:

- 0: The RX FIFO buffer is not empty.
- 1: The RX FIFO buffer is empty.

uart_<ChannelNumber>_rd_data / 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.

Range: 0 … 511

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).

uart_<ChannelNumber>_wr / 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.

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).

uart_<ChannelNumber>_wr_data_count / 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.

Range: 0 … 2047

uart_<ChannelNumber>_wr_fifo_full / 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.

Range:

- 0: The TX FIFO buffer is not full.
- 1: The TX FIFO buffer is full.

uart_<ChannelNumber>_wr_data / Write Data Specifies the value to be send.

The **Write_Data** signal is transferred at each clock cycle with **Write_Enable** set to 1.

Range: 0 … 511

uart_<Channel_Number>_driver_en / 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.

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 🕮)

Related topics

References

Overview of the Frameworks Available for MicroLabBox 14
UART (RS232) 78

I/O Functions of the DS2655 FPGA Base Board Framework

Introduction

The following frameworks of the DS2655 FPGA Base Boards provide the standard I/O functionality of the boards:

- *DS2655 (7K160) FPGA Base Board* framework
- *DS2655 (7K410) FPGA Base Board* framework

Where to go from here

Information in this section

APU Master....89
To distribute angle values over IOCNET for synchronizing angle-based applications.

APU Slave....91
To read angle values distributed by an APU Master over IOCNET for synchronizing angle-based applications.

Buffer In....93
To read data from an IOCNET buffer with a data width of 32 bits.

Buffer64 In....95
To read data from an IOCNET buffer with a data width of 64 bits.

Buffer Out....98
To write data to an IOCNET buffer with a data width of 32 bits.

Buffer64 Out....100
To write data to an IOCNET buffer with a data width of 64 bits.

CN App Status....103
To read the status of application that is running on the computation node.

I-FPGA In (IOCNET)....104
To read a 32-bit raw data value from an IOCNET buffer.

I-FPGA64 In (IOCNET)..... 105
To read a 64-bit raw data value from an IOCNET buffer.

I-FPGA Out (IOCNET)..... 107
To write a 32-bit raw data value to an IOCNET buffer.

I-FPGA64 Out (IOCNET)..... 109
To write a 64-bit raw data value to an IOCNET buffer.

Interrupt..... 111
To request a processor interrupt outside of the FPGA application.

IOCNET Global Time..... 112
To read the number of hardware ticks.

LED Out..... 113
To write a digital signal that controls the LED on the board.

Register In..... 114
To read data from an IOCNET register with a data width of 32 bits.

Register64 In..... 115
To read data from an IOCNET register with a data width of 64 bits.

Register Out..... 117
To write data to an IOCNET register with a data width of 32 bits.

Register64 Out..... 118
To write data to an IOCNET register with a data width of 64 bits.

Status In..... 120
To read a digital signal that outputs the state of the FPGA initialization sequence.

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

I/O Functions of the DS2655M1 I/O Module Framework..... 205
The DS2655M1 I/O Module framework provides analog and digital I/O functionality of SCALEXIO FPGA base board with at least one DS2655M1 Multi-I/O Module.

I/O Functions of the DS2655M2 I/O Module Framework..... 217
The DS2655M2 I/O Module framework provides digital I/O functionality of a SCALEXIO FPGA base board with at least one DS2655M2 Digital I/O Module.

I/O Functions of the DS6651 Multi-I/O Module Framework..... 239
The DS6651 Multi-I/O Module framework provides analog and digital I/O functionality of a SCALEXIO FPGA base board with at least one DS6651 Multi-I/O Module.

I/O Functions of the Inter-FPGA Interface Framework..... 281
The Inter-FPGA Interface framework provides access to the I/O module slots of a SCALEXIO FPGA base board to implement an inter-FPGA communication bus.

APU Master

Purpose

To distribute angle values over IOCNET for synchronizing angle-based applications.

Description

According to the number of physical connections available on the DS2655 FPGA Base Board, you can select the APU Master I/O functions. There are six digital output channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 7.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Angle range Lets you specify the angle value range of the Phi Read port.

- 0: The angle range is 720° and cannot be changed in ConfigurationDesk.
- 1: The angle range is 360° and cannot be changed in ConfigurationDesk.
- 2: 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°.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Initial position Lets you set the initial APU master position in degree.

- Value range: -1440° ... +1440°

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 01 … 06.

iocnet_glob_master_angle<ChannelNumber>_ctr / Phi Read HD For internal use only.

iocnet_glob_master_angle<ChannelNumber>_pos / 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}

The value range depends on the settings of the Angle Range:

- 360° angle range: 0 ... 32767 (2^{15}-1)
- 720° angle range: 0 ... 65535 (2^{16}-1)

Data type: UFix_16_0

Data width: 1

APU bus clock cycle: 8 ns

Range exceeding is not possible.

iocnet_glob_master_angle<ChannelNumber>_rev / Rev Read Specifies the 37 bit total revolution (rev) value for the APU bus.

The APU bus clock cycle is 8 ns. The 37 Bit range is -2^{36} … 2^{36} - 1.

Data type: Double

Data width: 1

Range exceeding is not possible.

iocnet_glob_master_angle<ChannelNumber>_vel / Delta Phi For internal use only.

iocnet_glob_master_angle<ChannelNumber>_en / Delta Phi Enable For internal use only.

iocnet_glob_master_angle<ChannelNumber>_busy / Busy Specifies whether APU master is busy to set the last velocity value. If Busy is 1 (high), new velocity values cannot be set.

Busy stays active for at least 10 µs depending on the IOCNET structure..

iocnet_glob_master_angle<ChannelNumber>_res / Angle Range Specifies the angle value range of Phi Read.

- 0: 720° angle range
- 1: 360° angle range

iocnet_glob_master_angle<ChannelNumber>_upd_vel_deg_sec / 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

Value range: -1,200,000 °/s ... +1,200,000 °/s

Range exceeding is not possible. The port is saturated at the higher or lower limit.

iocnet_glob_master_angle<ChannelNumber>_upd_trig / Set Velocity Specifies the actual value of Velocity as new velocity value. The new value is set only if Set Velocity is 1 (high) and Busy is 0 (low).

Notes on updating velocity values:

- Setting the velocity values at very short intervals (e.g. every 80 ns) leads to high data traffic on IOCNET.

  High data traffic might freeze your SCALEXIO system.
- Setting a new velocity value before the last setting is executed overwrites the last setting.

  To distribute and execute a new velocity value takes about 10 µs. If the APU master always sets new velocity values before the last value is executed, the velocity value will never change.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

APU Slave

Purpose

To read angle values distributed by an APU Master over IOCNET for synchronizing angle-based applications.

Description

According to the number of physical connections available on the DS2655 FPGA Base Board, you can select the APU Slave I/O functions. There are six digital input channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 4 … 9.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(11).Init Lets you inherit the angle range of the APU bus or specify a local angle range independent from the APU bus:

- 0: 720° angle range.
- 1: 360° angle range.
- 2: The angle range is inherited 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°	2 (Inherit)	360°
720°		720°
360°	1 (360°)	360°
	0 (720°)	720° [1]
720°	1 (360°)	360° [2]
	0 (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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 01 … 06.

iocnet_glob_angle<ChannelNumber>_pos / 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}

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)

Data type: UFix_16_0

Data width: 1

APU bus clock cycle: 8 ns

Range exceeding is not possible.

iocnet_glob_angle<ChannelNumber>_rev / Rev Read Outputs the 37 bit total revolution (rev) value for the APU bus.

The APU bus clock cycle is 8 ns. The 37 Bit range is -2^{36} … 2^{36} - 1.

Data type: Double

Data width: 1

Range exceeding is not possible.

iocnet_glob_angle<ChannelNumber>_ctr / Phi Read HD For internal use only.

iocnet_glob_angle<ChannelNumber>_vel / Delta Phi For internal use only.

iocnet_glob_angle<ChannelNumber>_en / Delta Phi Enable For internal use only.

iocnet_glob_angle<ChannelNumber>_res / Angle Range Outputs a flag whether the angle range of the APU bus is 360° or 720°. The angle range has been sent by an I/O board in the hardware system specified as APU master.

Data type: Double

Data width: 1

- 0: The angle range is 720°.
- 1: The angle range is 360°.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

Buffer In

Purpose

To read data from an IOCNET buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is read from an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 257 … 288.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf_<ChannelNumber>_count / 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).

xmemf_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an IOCNET buffer. The data format depends on the related parameter settings.

xmemf_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of

the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

xmemf_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf_<ChannelNumber>_read_req / Read Request Outputs a flag that indicates that a data transmission is requested via IOCNET. With Buffer In port Read Request and the Buffer Out port Send Acknowledge 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 the Buffer Out port Send Acknowledge, the Buffer In port Read Request 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.

Usable only if Enable Read_Req and Send_Ack ports for explicit data transmit value is enabled in the FPGA framework INI file.

- 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 the Buffer Out port Send Acknowledge 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.

For details on acknowledging a data transmission with Send Acknowledge, refer to Buffer Out on page 98.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

Buffer64 In

Purpose

To read data from an IOCNET buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is read from an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 545 … 576.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmem64f_<ChannelNumber>_count / 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).

xmem64f_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an IOCNET buffer. The data format depends on the related parameter settings.

xmem64f_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

xmem64f_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmem64f_<ChannelNumber>_read_req / Read Request Outputs a flag that indicates that a data transmission is requested via IOCNET. With Buffer In port **Read Request** and the Buffer Out port **Send Acknowledge** 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 the Buffer Out port **Send Acknowledge**, the Buffer In port **Read Request** 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.

Usable only if **Enable Read_Req and Send_Ack ports for explicit data transmit value** is enabled in the FPGA framework INI file.

- 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 the Buffer Out port **Send Acknowledge** 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.

For details on acknowledging a data transmission with **Send Acknowledge**, refer to Buffer64 Out on page 100.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

Buffer Out

Purpose

To write data to an IOCNET buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is written to an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.

- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.

- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 ... 32768.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(9).Init / Enable Read_Req and Send_Ack ports for explicit data transmit value Lets you enable the Buffer In port Read Request and the Buffer Out port Send Acknowledge. With Read Request and Send Acknowledge you can trigger a processor synchronous data exchange.

- 0: The ports are disabled. Each data request will instantly be acknowledged.
- 1: The ports are enabled. Each data request must be acknowledged by your handcode.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an IOCNET buffer. The data format depends on the related parameter settings.

xmemp_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via PHS bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp_<ChannelNumber>_overflow / 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.

xmemp_<ChannelNumber>_send_ack / Send Acknowledge Triggers a data transmission to IOCNET. With the Buffer Out port Send Acknowledge and the Buffer In port Read Request 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 the Buffer Out port **Send Acknowledge**, the Buffer In port **Read Request** 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.

Usable only if **Enable Read_Req and Send_Ack ports for explicit data transmit value** is enabled in the FPGA framework INI file.

- 0: No data transmission is acknowledged..
- 1: A data transmission is acknowledged and the current data values will be transmitted via IOCNET.

A data transmission request that is not acknowledged by the Buffer Out port **Send Acknowledge** 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.

For details on the port **Read Request**, refer to Buffer In on page 93.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks 17

Buffer64 Out

Purpose

To write data to an IOCNET buffer with a data width of 64 bits.

Description

If you select **Buffer64** as the access type, the data is written to an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 545 … 576.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the **Data** inport depending on the format selected in the **Format** setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(9).Init / Enable Read_Req and Send_Ack ports for explicit data transmit value Lets you enable the Buffer In port Read Request and the Buffer Out port Send Acknowledge. With Read Request and Send Acknowledge you can trigger a processor synchronous data exchange.

- 0: The ports are disabled. Each data request will instantly be acknowledged.
- 1: The ports are enabled. Each data request must be acknowledged by your handcode.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmem64p_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an IOCNET buffer. The data format depends on the related parameter settings.

xmem64p_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmem64p_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via IOCNET bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmem64p_<ChannelNumber>_overflow / 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.

xmem64p_<ChannelNumber>_send_ack / Send Acknowledge Triggers a data transmission to IOCNET. With the Buffer Out port Send Acknowledge and the Buffer In port Read Request 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 the Buffer Out port Send Acknowledge, the Buffer In port Read Request 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.

Usable only if Enable Read_Req and Send_Ack ports for explicit data transmit value is enabled in the FPGA framework INI file.

- 0: No data transmission is acknowledged..
- 1: A data transmission is acknowledged and the current data values will be transmitted via IOCNET.

A data transmission request that is not acknowledged by the Buffer Out port Send Acknowledge 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.

For details on the port Read Request, refer to Buffer64 In on page 95.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

CN App Status

Purpose

To read the status of application that is running on the computation node.

Description

There is one digital input channel that is used for the CN App Status I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 2.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

iocnet_appl_status/ CN Application Status Outputs the state of the application that is running on the computation node.

Data type: UFix_1_0

- 0: The application on the computation node is stopped.
- 1: The application on the computation node is running.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

I-FPGA In (IOCNET)

Purpose

To read a 32-bit raw data value from an IOCNET buffer.

Description

You can select I-FPGA In (IOCNET) I/O functions to implement an inter-FPGA communication between FPGA base boards. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 10 … 41.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmemf_inter_<ChannelNumber>_addr / 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** parameter. The address range with valid data values can be derived from the value of the **Data Count** port.

xmemf_inter_<ChannelNumber>_count / 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** parameter.

xmemf_inter_<ChannelNumber>_dout / Data Outputs a 32-bit raw data value from the specified address of the IOCNET buffer.

Data type: UFix_32_0

Data width: 1

xmemf_inter_<ChannelNumber>_new_data / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

I-FPGA64 In (IOCNET)

Purpose

To read a 64-bit raw data value from an IOCNET buffer.

Description

According to the number of physical connections available on the DS2655 FPGA Base Board, you can select the **I-FPGA64 In (IOCNET)** I/O functions. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 42 … 73.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmem64f_inter_<ChannelNumber>_addr / 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 parameter. The address range with valid data values can be derived from the value of the Data Count port.

xmem64f_inter_<ChannelNumber>_count / 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 parameter.

xmem64f_inter_<ChannelNumber>_dout / Data Outputs a 64-bit raw data value from the specified address of the IOCNET buffer.

Data type: UFix_64_0

Data width: 1

xmem64f_inter_<ChannelNumber>_new_data / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

I-FPGA Out (IOCNET)

Purpose

To write a 32-bit raw data value to an IOCNET buffer.

Description

You can select I-FPGA Out (IOCNET) I/O functions to implement an inter-FPGA communication between FPGA base boards. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 8 … 39.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmemp_inter_<ChannelNumber>_din / Data Specifies a 32-bit raw data value to be written to an IOCNET buffer.

Data type: UFix_32_0

Data width: 1

xmemp_inter_<ChannelNumber>_strobe / 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.

xmemp_inter_<ChannelNumber>_finished / 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.

xmemp_inter_<ChannelNumber>_write / 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.

xmemp_inter_<ChannelNumber>_overflow / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

I-FPGA64 Out (IOCNET)

Purpose

To write a 64-bit raw data value to an IOCNET buffer.

Description

You can select **I-FPGA64 Out (IOCNET)** I/O functions to implement an inter-FPGA communication between FPGA base boards. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 40 … 71.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmem64p_inter_<ChannelNumber>_din / Data Specifies a 64-bit raw data value to be written to an IOCNET buffer.

Data type: UFix_64_0

Data width: 1

xmem64p_inter_<ChannelNumber>_strobe / 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.

xmem64p_inter_<ChannelNumber>_finished / 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.

xmem64p_inter_<ChannelNumber>_write / 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.

xmem64p_inter_<ChannelNumber>_overflow / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

Interrupt

Purpose

To request a processor interrupt outside of the FPGA application.

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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 8.

IRQProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 7.

usr_<ChannelNumber>_interrupt / Int Provides the interrupt request line.

- 0 to 1: Interrupt is requested (edge-triggered).
- 0: No interrupt is requested. Last requested interrupt is saved.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

IOCNET Global Time

Purpose

To read the number of hardware ticks.

Description

There is one digital input channel that is used for the IOCNET Global Time I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 3.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

iocnet_glob_time/ IOCNET Global 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: UFix_56_0

Tick step-width: 8.5 ns

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

LED Out

Purpose

To write a digital signal that controls the LED on the board.

Description

There is one digital output channel that is used for the LED Out I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

led_out / LED Out Controls the LED on the board.

Data type: UFix_1_0

- 0: LED lights green.
- 1: LED lights orange.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

Register In

Purpose

To read data from an IOCNET register with a data width of 32 bits.

Description

If you select Register as the access type, the data is read from an IOCNET register. 256 registers are available with a data width of 32 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 256.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the IOCNET bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an IOCNET register. The data format depends on the related parameter settings.

xreg_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

Register64 In

Purpose

To read data from an IOCNET register with a data width of 64 bits.

Description

If you select Register as the access type, the data is read from an IOCNET register. 256 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 … 544.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the IOCNET bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an IOCNET register. The data format depends on the related parameter settings.

xreg64_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

Register Out

Purpose

To write data to an IOCNET register with a data width of 32 bits.

Description

If you select Register as the access type, the data is written to an IOCNET register. 256 registers are available with a data width of 32 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 256.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an IOCNET register. The data format depends on the related parameter settings.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

Register64 Out

Purpose

To write data to an IOCNET register with a data width of 64 bits.

Description

If you select Register64 as the access type, the data is written to an IOCNET register. 256 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 … 544.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Register64 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 bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an IOCNET register. The data format depends on the related parameter settings.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

Status In

Purpose

To read a digital signal that outputs the state of the FPGA initialization sequence.

Description

There is one digital input channel that is used for the Status In I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

init_done/ 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.

Related topics

References

Overview of the DS2655 FPGA Base Board Frameworks..... 17

I/O Functions of the DS6601 FPGA Base Board Framework

Introduction

The *DS6601 (KU035) FPGA Base Board* framework of the DS6601 FPGA Base Board provide the standard I/O functionality of the board.

Where to go from here

Information in this section

APU Master 125
To distribute angle values over IOCNET for synchronizing angle-based applications.

APU Slave 127
To read angle values distributed by an APU Master over IOCNET for synchronizing angle-based applications.

Buffer In 129
To read data from an IOCNET buffer with a data width of 32 bits.

Buffer64 In 131
To read data from an IOCNET buffer with a data width of 64 bits.

Buffer Out 133
To write data to an IOCNET buffer with a data width of 32 bits.

Buffer64 Out 136
To write data to an IOCNET buffer with a data width of 64 bits.

CN App Status 138
To read the status of application that is running on the computation node.

I-FPGA In (IOCNET) 139
To read a 32-bit raw data value from an IOCNET buffer.

I-FPGA64 In (IOCNET) 141
To read a 64-bit raw data value from an IOCNET buffer.

I-FPGA Out (IOCNET)..... 143
To write a 32-bit raw data value to an IOCNET buffer.

I-FPGA64 Out (IOCNET)..... 145
To write a 64-bit raw data value to an IOCNET buffer.

Interrupt..... 147
To request a processor interrupt outside of the FPGA application.

IOCNET Global Time..... 148
To read the number of hardware ticks.

LED Out..... 149
To write a digital signal that controls the LED on the board.

Register In..... 150
To read data from an IOCNET register with a data width of 32 bits.

Register64 In..... 151
To read data from an IOCNET register with a data width of 64 bits.

Register Out..... 153
To write data to an IOCNET register with a data width of 32 bits.

Register64 Out..... 154
To write data to an IOCNET register with a data width of 64 bits.

Status In..... 156
To read a digital signal that outputs the state of the FPGA initialization sequence.

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

I/O Functions of the DS2655M1 I/O Module Framework..... 205
The DS2655M1 I/O Module framework provides analog and digital I/O functionality of SCALEXIO FPGA base board with at least one DS2655M1 Multi-I/O Module.

I/O Functions of the DS2655M2 I/O Module Framework..... 217
The DS2655M2 I/O Module framework provides digital I/O functionality of a SCALEXIO FPGA base board with at least one DS2655M2 Digital I/O Module.

I/O Functions of the DS6651 Multi-I/O Module Framework..... 239
The DS6651 Multi-I/O Module framework provides analog and digital I/O functionality of a SCALEXIO FPGA base board with at least one DS6651 Multi-I/O Module.

I/O Functions of the Inter-FPGA Interface Framework..... 281
The Inter-FPGA Interface framework provides access to the I/O module slots of a SCALEXIO FPGA base board to implement an inter-FPGA communication bus.

APU Master

Purpose

To distribute angle values over IOCNET for synchronizing angle-based applications.

Description

According to the number of physical connections available on the DS6601 FPGA Base Board, you can select the APU Master I/O functions. There are six digital output channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 7.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Angle range Lets you specify the angle value range of the Phi Read port.

- 0: The angle range is 720° and cannot be changed in ConfigurationDesk.
- 1: The angle range is 360° and cannot be changed in ConfigurationDesk.
- 2: 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°.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Initial position Lets you set the initial APU master position in degree.

- Value range: -1440° ... +1440°

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 01 … 06.

iocnet_glob_master_angle<ChannelNumber>_ctr / Phi Read HD For internal use only.

iocnet_glob_master_angle<ChannelNumber>_pos / 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}

The value range depends on the settings of the Angle Range:

- 360° angle range: 0 ... 32767 (2^{15}-1)
- 720° angle range: 0 ... 65535 (2^{16}-1)

Data type: UFix_16_0

Data width: 1

APU bus clock cycle: 8 ns

Range exceeding is not possible.

iocnet_glob_master_angle<ChannelNumber>_rev / Rev Read Specifies the 37 bit total revolution (rev) value for the APU bus.

The APU bus clock cycle is 8 ns. The 37 Bit range is -2^{36} … $2^{36} - 1$.

Data type: Double

Data width: 1

Range exceeding is not possible.

iocnet_glob_master_angle<ChannelNumber>_vel / Delta Phi For internal use only.

iocnet_glob_master_angle<ChannelNumber>_en / Delta Phi Enable For internal use only.

iocnet_glob_master_angle<ChannelNumber>_busy / Busy Specifies whether APU master is busy to set the last velocity value. If Busy is 1 (high), new velocity values cannot be set.

Busy stays active for at least 10 µs depending on the IOCNET structure..

iocnet_glob_master_angle<ChannelNumber>_res / Angle Range Specifies the angle value range of Phi Read.

- 0: 720° angle range
- 1: 360° angle range

iocnet_glob_master_angle<ChannelNumber>_upd_vel_deg_sec / 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

Value range: -1,200,000 °/s ... +1,200,000 °/s

Range exceeding is not possible. The port is saturated at the higher or lower limit.

iocnet_glob_master_angle<ChannelNumber>_upd_trig / Set Velocity Specifies the actual value of Velocity as new velocity value. The new value is set only if Set Velocity is 1 (high) and Busy is 0 (low).

Notes on updating velocity values:

- Setting the velocity values at very short intervals (e.g. every 80 ns) leads to high data traffic on IOCNET.

  High data traffic might freeze your SCALEXIO system.
- Setting a new velocity value before the last setting is executed overwrites the last setting.

  To distribute and execute a new velocity value takes about 10 µs. If the APU master always sets new velocity values before the last value is executed, the velocity value will never change.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

APU Slave

Purpose

To read angle values distributed by an APU Master over IOCNET for synchronizing angle-based applications.

Description

According to the number of physical connections available on the DS6601 FPGA Base Board, you can select the APU Slave I/O functions. There are six digital input channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 4 … 9.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(11).Init Lets you inherit the angle range of the APU bus or specify a local angle range independent from the APU bus:

- 0: 720° angle range.
- 1: 360° angle range.
- 2: The angle range is inherited 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°	2 (Inherit)	360°
720°		720°
360°	1 (360°)	360°
	0 (720°)	720° [1)]

APU Bus Setting	Slave APU Setting	Resulting Angle Range of the Slave APU
720°	1 (360°)	360° [2]
	0 (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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 01 … 06.

iocnet_glob_angle<ChannelNumber>_pos / 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}

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)

Data type: UFix_16_0

Data width: 1

APU bus clock cycle: 8 ns

Range exceeding is not possible.

iocnet_glob_angle<ChannelNumber>_rev / Rev Read Outputs the 37 bit total revolution (rev) value for the APU bus.

The APU bus clock cycle is 8 ns. The 37 Bit range is -2^{36} … 2^{36} - 1.

Data type: Double

Data width: 1

Range exceeding is not possible.

iocnet_glob_angle<ChannelNumber>_ctr / Phi Read HD For internal use only.

iocnet_glob_angle<ChannelNumber>_vel / Delta Phi For internal use only.

iocnet_glob_angle<ChannelNumber>_en / Delta Phi Enable For internal use only.

iocnet_glob_angle<ChannelNumber>_res / Angle Range Outputs a flag whether the angle range of the APU bus is 360° or 720°. The angle range has been sent by an I/O board in the hardware system specified as APU master.

Data type: Double

Data width: 1

- 0: The angle range is 720°.
- 1: The angle range is 360°.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Buffer In

Purpose

To read data from an IOCNET buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is read from an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 257 … 288.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf_<ChannelNumber>_count / 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).

xmemf_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an IOCNET buffer. The data format depends on the related parameter settings.

xmemf_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

xmemf_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf_<ChannelNumber>_read_req / Read Request Outputs a flag that indicates that a data transmission is requested via IOCNET. With Buffer In port Read Request and the Buffer Out port Send Acknowledge 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 the Buffer Out port **Send Acknowledge**, the Buffer In port **Read Request** 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.

Usable only if **Enable Read_Req and Send_Ack ports for explicit data transmit value** is enabled in the FPGA framework INI file.

- 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 the Buffer Out port **Send Acknowledge** 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.

For details on acknowledging a data transmission with **Send Acknowledge**, refer to Buffer Out on page 133.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Buffer64 In

Purpose

To read data from an IOCNET buffer with a data width of 64 bits.

Description

If you select **Buffer64** as the access type, the data is read from an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 545 … 576.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the **Data** outport depending on the format selected in the **Format** setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmem64f_<ChannelNumber>_count / 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).

xmem64f_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an IOCNET buffer. The data format depends on the related parameter settings.

xmem64f_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

xmem64f_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmem64f_<ChannelNumber>_read_req / Read Request Outputs a flag that indicates that a data transmission is requested via IOCNET. With Buffer In port **Read Request** and the Buffer Out port **Send Acknowledge** 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 the Buffer Out port **Send Acknowledge**, the Buffer In port **Read Request** 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.

Usable only if **Enable Read_Req and Send_Ack ports for explicit data transmit value** is enabled in the FPGA framework INI file.

- 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 the Buffer Out port **Send Acknowledge** 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.

For details on acknowledging a data transmission with **Send Acknowledge**, refer to Buffer64 Out on page 136.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Buffer Out

Purpose

To write data to an IOCNET buffer with a data width of 32 bits.

Description

If you select **Buffer** as the access type, the data is written to an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 ... 32768.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(9).Init / Enable Read_Req and Send_Ack ports for explicit data transmit value Lets you enable the Buffer In port Read Request and the Buffer Out port Send Acknowledge. With Read Request and Send Acknowledge you can trigger a processor synchronous data exchange.

- 0: The ports are disabled. Each data request will instantly be acknowledged.
- 1: The ports are enabled. Each data request must be acknowledged by your handcode.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an IOCNET buffer. The data format depends on the related parameter settings.

xmemp_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via PHS bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp_<ChannelNumber>_overflow / 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.

xmemp_<ChannelNumber>_send_ack / Send Acknowledge Triggers a data transmission to IOCNET. With the Buffer Out port Send Acknowledge and the Buffer In port Read Request 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 the Buffer Out port Send Acknowledge, the Buffer In port Read Request 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.

Usable only if Enable Read_Req and Send_Ack ports for explicit data transmit value is enabled in the FPGA framework INI file.

- 0: No data transmission is acknowledged..
- 1: A data transmission is acknowledged and the current data values will be transmitted via IOCNET.

A data transmission request that is not acknowledged by the Buffer Out port Send Acknowledge 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.

For details on the port **Read Request**, refer to Buffer In on page 129.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Buffer64 Out

Purpose

To write data to an IOCNET buffer with a data width of 64 bits.

Description

If you select **Buffer64** as the access type, the data is written to an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 545 … 576.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the **Data** inport depending on the format selected in the **Format** setting (see below).

- signed/unsigned

  The values of the **Data** inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the **Data** inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(9).Init / Enable Read_Req and Send_Ack ports for explicit data transmit value Lets you enable the Buffer In port Read Request and the Buffer Out port Send Acknowledge. With Read Request and Send Acknowledge you can trigger a processor synchronous data exchange.

- 0: The ports are disabled. Each data request will instantly be acknowledged.
- 1: The ports are enabled. Each data request must be acknowledged by your handcode.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmem64p_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an IOCNET buffer. The data format depends on the related parameter settings.

xmem64p_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmem64p_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via IOCNET bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmem64p_<ChannelNumber>_overflow / 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.

xmem64p_<ChannelNumber>_send_ack / Send Acknowledge Triggers a data transmission to IOCNET. With the Buffer Out port **Send Acknowledge** and the Buffer In port **Read Request** 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 the Buffer Out port **Send Acknowledge**, the Buffer In port **Read Request** 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.

Usable only if **Enable Read_Req and Send_Ack ports for explicit data transmit value** is enabled in the FPGA framework INI file.

- 0: No data transmission is acknowledged..
- 1: A data transmission is acknowledged and the current data values will be transmitted via IOCNET.

A data transmission request that is not acknowledged by the Buffer Out port **Send Acknowledge** 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.

For details on the port **Read Request**, refer to Buffer64 In on page 131.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

CN App Status

Purpose

To read the status of application that is running on the computation node.

Description

There is one digital input channel that is used for the **CN App Status** I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 2.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

iocnet_appl_status/ CN Application Status Outputs the state of the application that is running on the computation node.

Data type: UFix_1_0

- 0: The application on the computation node is stopped.
- 1: The application on the computation node is running.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

I-FPGA In (IOCNET)

Purpose

To read a 32-bit raw data value from an IOCNET buffer.

Description

You can select I-FPGA In (IOCNET) I/O functions to implement an inter-FPGA communication between FPGA base boards. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 11 … 42.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmemf_inter_<ChannelNumber>_addr / 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 parameter. The address range with valid data values can be derived from the value of the Data Count port.

xmemf_inter_<ChannelNumber>_count / 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 parameter.

xmemf_inter_<ChannelNumber>_dout / Data Outputs a 32-bit raw data value from the specified address of the IOCNET buffer.

Data type: UFix_32_0

Data width: 1

xmemf_inter_<ChannelNumber>_new_data / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

I-FPGA64 In (IOCNET)

Purpose

To read a 64-bit raw data value from an IOCNET buffer.

Description

According to the number of physical connections available on the DS6601 FPGA Base Board, you can select the **I-FPGA64 In (IOCNET)** I/O functions. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 43 … 74.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmem64f_inter_<ChannelNumber>_addr / 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 parameter. The address range with valid data values can be derived from the value of the Data Count port.

xmem64f_inter_<ChannelNumber>_count / 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 parameter.

xmem64f_inter_<ChannelNumber>_dout / Data Outputs a 64-bit raw data value from the specified address of the IOCNET buffer.

Data type: UFix_64_0

Data width: 1

xmem64f_inter_<ChannelNumber>_new_data / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

I-FPGA Out (IOCNET)

Purpose

To write a 32-bit raw data value to an IOCNET buffer.

Description

You can select I-FPGA Out (IOCNET) I/O functions to implement an inter-FPGA communication between FPGA base boards. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 8 … 39.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmemp_inter_<ChannelNumber>_din / Data Specifies a 32-bit raw data value to be written to an IOCNET buffer.

Data type: UFix_32_0

Data width: 1

xmemp_inter_<ChannelNumber>_strobe / 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.

xmemp_inter_<ChannelNumber>_finished / 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.

xmemp_inter_<ChannelNumber>_write / 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.

xmemp_inter_<ChannelNumber>_overflow / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

I-FPGA64 Out (IOCNET)

Purpose

To write a 64-bit raw data value to an IOCNET buffer.

Description

You can select I-FPGA64 Out (IOCNET) I/O functions to implement an inter-FPGA communication between FPGA base boards. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 40 … 71.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmem64p_inter_<ChannelNumber>_din / Data Specifies a 64-bit raw data value to be written to an IOCNET buffer.

Data type: UFix_64_0

Data width: 1

xmem64p_inter_<ChannelNumber>_strobe / 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.

xmem64p_inter_<ChannelNumber>_finished / 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.

xmem64p_inter_<ChannelNumber>_write / 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.

xmem64p_inter_<ChannelNumber>_overflow / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Interrupt

Purpose

To request a processor interrupt outside of the FPGA application.

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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 16.

IRQProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 15.

usr_<ChannelNumber>_interrupt / Int Provides the interrupt request line.

- 0 to 1: Interrupt is requested (edge-triggered).
- 0: No interrupt is requested. Last requested interrupt is saved.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

IOCNET Global Time

Purpose

To read the number of hardware ticks.

Description

There is one digital input channel that is used for the IOCNET Global Time I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 3.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

iocnet_glob_time/ IOCNET Global 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: UFix_56_0

Tick step-width: 8.5 ns

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

LED Out

Purpose

To write a digital signal that controls the LED on the board.

Description

There is one digital output channel that is used for the LED Out I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

led_out / LED Out Controls the LED on the board.

Data type: UFix_1_0

- 0: LED lights green.
- 1: LED lights orange.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Register In

Purpose

To read data from an IOCNET register with a data width of 32 bits.

Description

If you select Register as the access type, the data is read from an IOCNET register. 256 registers are available with a data width of 32 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 256.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the .

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an IOCNET register. The data format depends on the related parameter settings.

xreg_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Register64 In

Purpose

To read data from an IOCNET register with a data width of 64 bits.

Description

If you select Register as the access type, the data is read from an IOCNET register. 256 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 … 544.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the .

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an IOCNET register. The data format depends on the related parameter settings.

xreg64_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Register Out

Purpose

To write data to an IOCNET register with a data width of 32 bits.

Description

If you select Register as the access type, the data is written to an IOCNET register. 256 registers are available with a data width of 32 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 256.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an IOCNET register. The data format depends on the related parameter settings.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Register64 Out

Purpose

To write data to an IOCNET register with a data width of 64 bits.

Description

If you select Register64 as the access type, the data is written to an IOCNET register. 256 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 … 544.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Register64 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 sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an IOCNET register. The data format depends on the related parameter settings.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

Status In

Purpose

To read a digital signal that outputs the state of the FPGA initialization sequence.

Description

There is one digital input channel that is used for the Status In I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

init_done/ 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.

Related topics

References

Overview of the DS6601 FPGA Base Board Frameworks..... 19

I/O Functions of the DS6602 FPGA Base Board Framework

Introduction

The *DS6602 (KU15P) FPGA Base Board* framework of the DS6602 FPGA Base Board provide the standard I/O functionality of the board.

Note

If you use a DS6602 FPGA Base Board, the build process issues a critical warning about a specific timing requirement.
For more information, refer to Problems and Their Solutions (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Where to go from here

Information in this section

APU Master 161
To distribute angle values over IOCNET for synchronizing angle-based applications.

APU Slave 163
To read angle values distributed by an APU Master over IOCNET for synchronizing angle-based applications.

Buffer In 165
To read data from an IOCNET buffer with a data width of 32 bits.

Buffer64 In 167
To read data from an IOCNET buffer with a data width of 64 bits.

Buffer Out 170
To write data to an IOCNET buffer with a data width of 32 bits.

Buffer64 Out 172
To write data to an IOCNET buffer with a data width of 64 bits.

CN App Status....175
To read the status of application that is running on the computation node.

DDR4 32 Mode 1....176
To provide 32-bit read/write access to the DDR4 RAM using the memory access mode 1.

DDR4 32 Mode 2....178
To provide 32-bit read/write access to the DDR4 RAM using the memory access mode 2.

DDR4 64 Mode 1....181
To provide 64-bit read/write access to the DDR4 RAM using the memory access mode 1.

DDR4 64 Mode 2....183
To provide 64-bit read/write access to the DDR4 RAM using the memory access mode 2.

I-FPGA In (IOCNET)....186
To read a 32-bit raw data value from an IOCNET buffer.

I-FPGA64 In (IOCNET)....188
To read a 64-bit raw data value from an IOCNET buffer.

I-FPGA Out (IOCNET)....189
To write a 32-bit raw data value to an IOCNET buffer.

I-FPGA64 Out (IOCNET)....191
To write a 64-bit raw data value to an IOCNET buffer.

Interrupt....193
To request a processor interrupt outside of the FPGA application.

IOCNET Global Time....194
To read the number of hardware ticks.

LED Out....195
To write a digital signal that controls the LED on the board.

Register In....196
To read data from an IOCNET register with a data width of 32 bits.

Register64 In....197
To read data from an IOCNET register with a data width of 64 bits.

Register Out....199
To write data to an IOCNET register with a data width of 32 bits.

Register64 Out....201
To write data to an IOCNET register with a data width of 64 bits.

Status In....202
To read a digital signal that outputs the state of the FPGA initialization sequence.

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

I/O Functions of the DS2655M1 I/O Module Framework.....205
The DS2655M1 I/O Module framework provides analog and digital I/O functionality of SCALEXIO FPGA base board with at least one DS2655M1 Multi-I/O Module.

I/O Functions of the DS2655M2 I/O Module Framework.....217
The DS2655M2 I/O Module framework provides digital I/O functionality of a SCALEXIO FPGA base board with at least one DS2655M2 Digital I/O Module.

I/O Functions of the DS6651 Multi-I/O Module Framework.....239
The DS6651 Multi-I/O Module framework provides analog and digital I/O functionality of a SCALEXIO FPGA base board with at least one DS6651 Multi-I/O Module.

I/O Functions of the Inter-FPGA Interface Framework.....281
The Inter-FPGA Interface framework provides access to the I/O module slots of a SCALEXIO FPGA base board to implement an inter-FPGA communication bus.

APU Master

Purpose

To distribute angle values over IOCNET for synchronizing angle-based applications.

Description

According to the number of physical connections available on the DS6602 FPGA Base Board, you can select the APU Master I/O functions. There are six digital output channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 7.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Angle range Lets you specify the angle value range of the Phi Read port.

- 0: The angle range is 720° and cannot be changed in ConfigurationDesk.
- 1: The angle range is 360° and cannot be changed in ConfigurationDesk.

- 2: 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°.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Initial position Lets you set the initial APU master position in degree.

- Value range: -1440° ... +1440°

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 01 … 06.

iocnet_glob_master_angle<ChannelNumber>_ctr / Phi Read HD For internal use only.

iocnet_glob_master_angle<ChannelNumber>_pos / 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}

The value range depends on the settings of the Angle Range:

- 360° angle range: 0 ... 32767 (2^{15}-1)
- 720° angle range: 0 ... 65535 (2^{16}-1)

Data type: UFix_16_0

Data width: 1

APU bus clock cycle: 8 ns

Range exceeding is not possible.

iocnet_glob_master_angle<ChannelNumber>_rev / Rev Read Specifies the 37 bit total revolution (rev) value for the APU bus.

The APU bus clock cycle is 8 ns. The 37 Bit range is -2^{36} … 2^{36} - 1.

Data type: Double

Data width: 1

Range exceeding is not possible.

iocnet_glob_master_angle<ChannelNumber>_vel / Delta Phi For internal use only.

iocnet_glob_master_angle<ChannelNumber>_en / Delta Phi Enable For internal use only.

iocnet_glob_master_angle<ChannelNumber>_busy / Busy Specifies whether APU master is busy to set the last velocity value. If Busy is 1 (high), new velocity values cannot be set.

Busy stays active for at least 10 µs depending on the IOCNET structure..

iocnet_glob_master_angle<ChannelNumber>_res / Angle Range Specifies the angle value range of Phi Read.

- 0: 720° angle range
- 1: 360° angle range

iocnet_glob_master_angle<ChannelNumber>_upd_vel_deg_sec / 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

Value range: -1,200,000 °/s ... +1,200,000 °/s

Range exceeding is not possible. The port is saturated at the higher or lower limit.

iocnet_glob_master_angle<ChannelNumber>_upd_trig / Set Velocity Specifies the actual value of Velocity as new velocity value. The new value is set only if Set Velocity is 1 (high) and Busy is 0 (low).

Notes on updating velocity values:

- Setting the velocity values at very short intervals (e.g. every 80 ns) leads to high data traffic on IOCNET.

  High data traffic might freeze your SCALEXIO system.
- Setting a new velocity value before the last setting is executed overwrites the last setting.

  To distribute and execute a new velocity value takes about 10 µs. If the APU master always sets new velocity values before the last value is executed, the velocity value will never change.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks.....21

APU Slave

Purpose

To read angle values distributed by an APU Master over IOCNET for synchronizing angle-based applications.

Description

According to the number of physical connections available on the DS6601 FPGA Base Board, you can select the APU Slave I/O functions. There are six digital input channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 4 … 9.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(11).Init Lets you inherit the angle range of the APU bus or specify a local angle range independent from the APU bus:

- 0: 720° angle range.
- 1: 360° angle range.
- 2: The angle range is inherited 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°	2 (Inherit)	360°
720°		720°
360°	1 (360°)	360°
	0 (720°)	720° [1]
720°	1 (360°)	360° [2]
	0 (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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 01 … 06.

iocnet_glob_angle<ChannelNumber>_pos / 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`

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)

Data type: UFix_16_0

Data width: 1

APU bus clock cycle: 8 ns

Range exceeding is not possible.

iocnet_glob_angle<ChannelNumber>_rev / Rev Read Outputs the 37 bit total revolution (rev) value for the APU bus.

The APU bus clock cycle is 8 ns. The 37 Bit range is -2^{36} … 2^{36} - 1.

Data type: Double

Data width: 1

Range exceeding is not possible.

iocnet_glob_angle<ChannelNumber>_ctr / Phi Read HD For internal use only.

iocnet_glob_angle<ChannelNumber>_vel / Delta Phi For internal use only.

iocnet_glob_angle<ChannelNumber>_en / Delta Phi Enable For internal use only.

iocnet_glob_angle<ChannelNumber>_res / Angle Range Outputs a flag whether the angle range of the APU bus is 360° or 720°. The angle range has been sent by an I/O board in the hardware system specified as APU master.

Data type: Double

Data width: 1

- 0: The angle range is 720°.
- 1: The angle range is 360°.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks..... 21

Buffer In

Purpose

To read data from an IOCNET buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is read from an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 257 … 288.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf_<ChannelNumber>_count / 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).

xmemf_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an IOCNET buffer. The data format depends on the related parameter settings.

xmemf_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of

the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

xmemf_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf_<ChannelNumber>_read_req / Read Request Outputs a flag that indicates that a data transmission is requested via IOCNET. With Buffer In port Read Request and the Buffer Out port Send Acknowledge 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 the Buffer Out port Send Acknowledge, the Buffer In port Read Request 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.

Usable only if Enable Read_Req and Send_Ack ports for explicit data transmit value is enabled in the FPGA framework INI file.

- 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 the Buffer Out port Send Acknowledge 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.

For details on acknowledging a data transmission with Send Acknowledge, refer to Buffer Out on page 170.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks.....21

Buffer64 In

Purpose

To read data from an IOCNET buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is read from an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 545 … 576.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmem64f_<ChannelNumber>_count / 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).

xmem64f_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an IOCNET buffer. The data format depends on the related parameter settings.

xmem64f_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

xmem64f_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmem64f_<ChannelNumber>_read_req / Read Request Outputs a flag that indicates that a data transmission is requested via IOCNET. With Buffer In port **Read Request** and the Buffer Out port **Send Acknowledge** 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 the Buffer Out port **Send Acknowledge**, the Buffer In port **Read Request** 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.

Usable only if **Enable Read_Req and Send_Ack ports for explicit data transmit value** is enabled in the FPGA framework INI file.

- 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 the Buffer Out port **Send Acknowledge** 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.

For details on acknowledging a data transmission with **Send Acknowledge**, refer to Buffer64 Out on page 172.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks.....21

Buffer Out

Purpose

To write data to an IOCNET buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is written to an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.

- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.

- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 ... 32768.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(9).Init / Enable Read_Req and Send_Ack ports for explicit data transmit value Lets you enable the Buffer In port Read Request and the Buffer Out port Send Acknowledge. With Read Request and Send Acknowledge you can trigger a processor synchronous data exchange.

- 0: The ports are disabled. Each data request will instantly be acknowledged.
- 1: The ports are enabled. Each data request must be acknowledged by your handcode.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an IOCNET buffer. The data format depends on the related parameter settings.

xmemp_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via PHS bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp_<ChannelNumber>_overflow / 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.

xmemp_<ChannelNumber>_send_ack / Send Acknowledge Triggers a data transmission to IOCNET. With the Buffer Out port Send Acknowledge and the Buffer In port Read Request 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 the Buffer Out port **Send Acknowledge**, the Buffer In port **Read Request** 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.

Usable only if **Enable Read_Req and Send_Ack ports for explicit data transmit value** is enabled in the FPGA framework INI file.

- 0: No data transmission is acknowledged..
- 1: A data transmission is acknowledged and the current data values will be transmitted via IOCNET.

A data transmission request that is not acknowledged by the Buffer Out port **Send Acknowledge** 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.

For details on the port **Read Request**, refer to Buffer In on page 165.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks 21

Buffer64 Out

Purpose

To write data to an IOCNET buffer with a data width of 64 bits.

Description

If you select **Buffer64** as the access type, the data is written to an IOCNET buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 545 … 576.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the **Data** inport depending on the format selected in the **Format** setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(9).Init / Enable Read_Req and Send_Ack ports for explicit data transmit value Lets you enable the Buffer In port Read Request and the Buffer Out port Send Acknowledge. With Read Request and Send Acknowledge you can trigger a processor synchronous data exchange.

- 0: The ports are disabled. Each data request will instantly be acknowledged.
- 1: The ports are enabled. Each data request must be acknowledged by your handcode.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmem64p_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an IOCNET buffer. The data format depends on the related parameter settings.

xmem64p_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmem64p_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via IOCNET bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmem64p_<ChannelNumber>_overflow / 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.

xmem64p_<ChannelNumber>_send_ack / Send Acknowledge Triggers a data transmission to IOCNET. With the Buffer Out port Send Acknowledge and the Buffer In port Read Request 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 the Buffer Out port Send Acknowledge, the Buffer In port Read Request 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.

Usable only if Enable Read_Req and Send_Ack ports for explicit data transmit value is enabled in the FPGA framework INI file.

- 0: No data transmission is acknowledged..
- 1: A data transmission is acknowledged and the current data values will be transmitted via IOCNET.

A data transmission request that is not acknowledged by the Buffer Out port Send Acknowledge 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.

For details on the port Read Request, refer to Buffer64 In on page 167.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks.....21

CN App Status

Purpose

To read the status of application that is running on the computation node.

Description

There is one digital input channel that is used for the CN App Status I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 2.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

iocnet_appl_status/ CN Application Status Outputs the state of the application that is running on the computation node.

Data type: UFix_1_0

- 0: The application on the computation node is stopped.
- 1: The application on the computation node is running.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks.....21

DDR4 32 Mode 1

Purpose

To provide 32-bit read/write access to the DDR4 RAM using the memory access mode 1.

Description

The DS6602 FPGA Base Board provides a 4 GB DDR4 RAM that can be used by the FPGA application.

The RAM interface always handles 512 bits at once. Therefore, the FPGA application can read/write 16 x 32 bits data or 8 x 64 bits data within one memory access.

You can select different I/O functions to access the DDR4 RAM:

- DDR4 32 Mode 1 and DDR4 64 Mode 1 to read/write 32/64-bit values with one memory address. These I/O types use the memory access mode 1.
- DDR4 32 Mode 2 and DDR4 64 Mode 2 to read/write 32/64-bit values with two memory addresses. These I/O types use the memory access mode 2.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is 8.

hcfw.IOProperties.Out.Fct(8).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

ddr4_init_done / Init Done Outputs a flag that indicates that the RAM is initialized with specified data values. For more information, refer to Initializing the DDR4 RAM of the DS6602 (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Data type: UFix_1_0

Data width: 1

Values:

- 0: The RAM is not initialized.
- 1: The RAM is initialized.

ddr4_init_fail / 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.

ddr4_busy_flag / Busy Inport to read a flag that indicates the state of the DDR4 RAM:

- 0: The DDR4 module is ready for new read/write operations.
- 1: The DDR4 module is busy.

Data type: UFix_1_0

ddr4_data_en / Enable Outport to enable the RAM access:

- 1: Data values are written to the RAM or read from the RAM.
- 0: No read/write access.

Data type: UFix_1_0

ddr4_direction / Direction Outport to control the direction of data access:

- 0: Write access
- 1: Read access

Data type: UFix_1_0

ddr4_address_block_1 / Address Outport to specify 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

Value range: 0 … 67,108,863 (2^{26}-1)

Data width: 1

ddr4_rd_data_valid_block_1 / Data valid Inport to read a flag that indicates that the data values of the Data Read ports are valid:

- 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 type: UFix_1_0

ddr4_data_<PortNumber>_wr_32_block_1 / Data Write Outport to specify 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

ddr4_data_<PortNumber>_rd_32_block_1 / Data Read Inport to read a 32-bit data value 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

I/O mapping

No external connection to the I/O connector of the board.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

DDR4 32 Mode 2

Purpose

To provide 32-bit read/write access to the DDR4 RAM using the memory access mode 2.

Description

The DS6602 FPGA Base Board provides a 4 GB DDR4 RAM that can be used by the FPGA application.

The RAM interface always handles 512 bits at once. Therefore, the FPGA application can read/write 16 x 32 bits data or 8 x 64 bits data within one memory access.

You can select different I/O functions to access the DDR4 RAM:

- **DDR4 32 Mode 1** and **DDR4 64 Mode 1** to read/write 32/64-bit values with one memory address. These I/O types use the memory access mode 1.
- **DDR4 32 Mode 2** and **DDR4 64 Mode 2** to read/write 32/64-bit values with two memory addresses. These I/O types use the memory access mode 2.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is 9.

hcfw.IOProperties.Out.Fct(8).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

ddr4_init_done / Init Done Outputs a flag that indicates that the RAM is initialized with specified data values. For more information, refer to Initializing the DDR4 RAM of the DS6602 (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Data type: UFix_1_0

Data width: 1

Values:

- 0: The RAM is not initialized.
- 1: The RAM is initialized.

ddr4_init_fail / 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.

ddr4_busy_flag / Busy Inport to read a flag that indicates the state of the DDR4 RAM:

- 0: The DDR4 module is ready for new read/write operations.
- 1: The DDR4 module is busy.

Data type: UFix_1_0

ddr4_data_en / Enable Outport to enable the RAM access:

- 1: Data values are written to the RAM or read from the RAM.
- 0: No read/write access.

Data type: UFix_1_0

ddr4_direction / Direction Outport to control the direction of data access:

- 0: Write access
- 1: Read access

Data type: UFix_1_0

ddr4_address_1_block_3 / Address A Outport to specify 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

Value range: 0 … 134,217,727 (2^{27}-1)

Data width: 1

ddr4_address_2_block_3 / Address B Outport to specify 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

Value range: 0 … 134,217,727 (2^{27}-1)

Data width: 1

ddr4_rd_data_valid_1_block_3 / Data valid A Inport to read a flag that indicates that the data values of the Data Read A ports are valid:

- 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 type: UFix_1_0

ddr4_rd_data_valid_2_block_3 / Data valid B Inport to read a flag that indicates that the data values of the Data Read B ports are valid:

- 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 type: UFix_1_0

ddr4_data_1_<PortNumber>_wr_32_block_3 / Data Write A Outport to specify 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

ddr4_data_2_<PortNumber>_wr_32_block_3 / Data Write B Outport to specify 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

ddr4_data_1_<PortNumber>_rd_32_block_3 / Data Read A Inport to read a 32-bit data value 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

ddr4_data_2_<PortNumber>_rd_32_block_3 / Data Read B Inport to read a 32-bit data value 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

I/O mapping

No external connection to the I/O connector of the board.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

DDR4 64 Mode 1

Purpose

To provide 64-bit read/write access to the DDR4 RAM using the memory access mode 1.

Description

The DS6602 FPGA Base Board provides a 4 GB DDR4 RAM that can be used by the FPGA application.

The RAM interface always handles 512 bits at once. Therefore, the FPGA application can read/write 16 x 32 bits data or 8 x 64 bits data within one memory access.

You can select different I/O functions to access the DDR4 RAM:

- DDR4 32 Mode 1 and DDR4 64 Mode 1 to read/write 32/64-bit values with one memory address. These I/O types use the memory access mode 1.
- DDR4 32 Mode 2 and DDR4 64 Mode 2 to read/write 32/64-bit values with two memory addresses. These I/O types use the memory access mode 2.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is 10.

hcfw.IOProperties.Out.Fct(8).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

ddr4_init_done / Init Done Outputs a flag that indicates that the RAM is initialized with specified data values. For more information, refer to Initializing the DDR4 RAM of the DS6602 (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Data type: UFix_1_0

Data width: 1

Values:

- 0: The RAM is not initialized.
- 1: The RAM is initialized.

ddr4_init_fail / 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.

ddr4_busy_flag / Busy Inport to read a flag that indicates the state of the DDR4 RAM:

- 0: The DDR4 module is ready for new read/write operations.
- 1: The DDR4 module is busy.

Data type: UFix_1_0

ddr4_data_en / Enable Outport to enable the RAM access:

- 1: Data values are written to the RAM or read from the RAM.
- 0: No read/write access.

Data type: UFix_1_0

ddr4_direction / Direction Outport to control the direction of data access:

- 0: Write access
- 1: Read access

Data type: UFix_1_0

ddr4_address_block_2 / Address Outport to specify 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

Value range: 0 … 67,108,863 (2^{26}-1)

Data width: 1

ddr4_rd_data_valid_block_2 / Data valid Inport to read a flag that indicates that the data values of the Data Read ports are valid:

- 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 type: UFix_1_0

ddr4_data_<PortNumber>_wr_64_block_2 / Data Write Outport to specify a 64-bit data value to be written to the RAM. 8 ports specify 8 data values. The Address port specifies the memory address to write the data values.

Data type: UFix_64_0

Data width: 1 per port

ddr4_data_<PortNumber>_rd_64_block_2 / Data Read Inport to read a 64-bit data value from the RAM. 8 ports output 8 data values. The Address port specifies the memory address to read the data values.

Data type: UFix_64_0

Data width: 1 per port

I/O mapping

No external connection to the I/O connector of the board.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

DDR4 64 Mode 2

Purpose

To provide 64-bit read/write access to the DDR4 RAM using the memory access mode 2.

Description

The DS6602 FPGA Base Board provides a 4 GB DDR4 RAM that can be used by the FPGA application.

The RAM interface always handles 512 bits at once. Therefore, the FPGA application can read/write 16 x 32 bits data or 8 x 64 bits data within one memory access.

You can select different I/O functions to access the DDR4 RAM:

- DDR4 32 Mode 1 and DDR4 64 Mode 1 to read/write 32/64-bit values with one memory address. These I/O types use the memory access mode 1.
- DDR4 32 Mode 2 and DDR4 64 Mode 2 to read/write 32/64-bit values with two memory addresses. These I/O types use the memory access mode 2.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is 11.

hcfw.IOProperties.Out.Fct(8).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

ddr4_init_done / Init Done Outputs a flag that indicates that the RAM is initialized with specified data values. For more information, refer to Initializing the DDR4 RAM of the DS6602 (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Data type: UFix_1_0

Data width: 1

Values:

- 0: The RAM is not initialized.
- 1: The RAM is initialized.

ddr4_init_fail / 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.

ddr4_busy_flag / Busy Inport to read a flag that indicates the state of the DDR4 RAM:

- 0: The DDR4 module is ready for new read/write operations.
- 1: The DDR4 module is busy.

Data type: UFix_1_0

ddr4_data_en / Enable Outport to enable the RAM access:

- 1: Data values are written to the RAM or read from the RAM.
- 0: No read/write access.

Data type: UFix_1_0

ddr4_direction / Direction Outport to control the direction of data access:

- 0: Write access
- 1: Read access

Data type: UFix_1_0

ddr4_address_1_block_4 / Address A Outport to specify 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

Value range: 0 … 134,217,727 (2^{27}-1)

Data width: 1

ddr4_address_2_block_4 / Address B Outport to specify 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

Value range: 0 … 134,217,727 (2^{27}-1)

Data width: 1

ddr4_rd_data_valid_1_block_4 / Data valid A Inport to read a flag that indicates that the data values of the Data Read A ports are valid:

- 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 type: UFix_1_0

ddr4_rd_data_valid_2_block_4 / Data valid B Inport to read a flag that indicates that the data values of the Data Read B ports are valid:

- 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 type: UFix_1_0

ddr4_data_1_<PortNumber>_wr_64_block_4 / Data Write A Outport to specify 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

ddr4_data_2_<PortNumber>_wr_64_block_4 / Data Write B Outport to specify 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

ddr4_data_1_<PortNumber>_rd_64_block_4 / Data Read A Inport to read a 64-bit data value 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

ddr4_data_2_<PortNumber>_rd_64_block_4 / Data Read B Inport to read a 64-bit data value 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

I/O mapping

No external connection to the I/O connector of the board.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

I-FPGA In (IOCNET)

Purpose

To read a 32-bit raw data value from an IOCNET buffer.

Description

You can select I-FPGA In (IOCNET) I/O functions to implement an inter-FPGA communication between FPGA base boards. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 11 … 42.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmemf_inter_<ChannelNumber>_addr / 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 parameter. The address range with valid data values can be derived from the value of the Data Count port.

xmemf_inter_<ChannelNumber>_count / 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 parameter.

xmemf_inter_<ChannelNumber>_dout / Data Outputs a 32-bit raw data value from the specified address of the IOCNET buffer.

Data type: UFix_32_0

Data width: 1

xmemf_inter_<ChannelNumber>_new_data / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks..... 21

I-FPGA64 In (IOCNET)

Purpose

To read a 64-bit raw data value from an IOCNET buffer.

Description

According to the number of physical connections available on the DS6602 FPGA Base Board, you can select the I-FPGA64 In (IOCNET) I/O functions. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 43 … 74.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmem64f_inter_<ChannelNumber>_addr / 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** parameter. The address range with valid data values can be derived from the value of the **Data Count** port.

xmem64f_inter_<ChannelNumber>_count / 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** parameter.

xmem64f_inter_<ChannelNumber>_dout / Data Outputs a 64-bit raw data value from the specified address of the IOCNET buffer.

Data type: UFix_64_0

Data width: 1

xmem64f_inter_<ChannelNumber>_new_data / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks....21

I-FPGA Out (IOCNET)

Purpose

To write a 32-bit raw data value to an IOCNET buffer.

Description

You can select **I-FPGA Out (IOCNET)** I/O functions to implement an inter-FPGA communication between FPGA base boards. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 12 … 43.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmemp_inter_<ChannelNumber>_din / Data Specifies a 32-bit raw data value to be written to an IOCNET buffer.

Data type: UFix_32_0

Data width: 1

xmemp_inter_<ChannelNumber>_strobe / 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.

xmemp_inter_<ChannelNumber>_finished / 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.

xmemp_inter_<ChannelNumber>_write / 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.

xmemp_inter_<ChannelNumber>_overflow / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks.....21

I-FPGA64 Out (IOCNET)

Purpose

To write a 64-bit raw data value to an IOCNET buffer.

Description

You can select I-FPGA64 Out (IOCNET) I/O functions to implement an inter-FPGA communication between FPGA base boards. There are 32 channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 44 … 75.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. To do this, you can reinterpret the 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.

xmem64p_inter_<ChannelNumber>_din / Data Specifies a 64-bit raw data value to be written to an IOCNET buffer.

Data type: UFix_64_0

Data width: 1

xmem64p_inter_<ChannelNumber>_strobe / 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.

xmem64p_inter_<ChannelNumber>_finished / 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.

xmem64p_inter_<ChannelNumber>_write / 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.

xmem64p_inter_<ChannelNumber>_overflow / 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.

I/O mapping

No external connection to the I/O connector of the board.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks....21

Interrupt

Purpose

To request a processor interrupt outside of the FPGA application.

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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 16.

IRQProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 15.

usr_<ChannelNumber>_interrupt / Int Provides the interrupt request line.

- 0 to 1: Interrupt is requested (edge-triggered).
- 0: No interrupt is requested. Last requested interrupt is saved.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks.....21

IOCNET Global Time

Purpose

To read the number of hardware ticks.

Description

There is one digital input channel that is used for the IOCNET Global Time I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 3.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

iocnet_glob_time/ IOCNET Global 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: UFix_56_0

Tick step-width: 8.5 ns

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks.....21

LED Out

Purpose

To write a digital signal that controls the LED on the board.

Description

There is one digital output channel that is used for the LED Out I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

led_out / LED Out Controls the LED on the board.

Data type: UFix_1_0

- 0: LED lights green.
- 1: LED lights orange.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks..... 21

Register In

Purpose

To read data from an IOCNET register with a data width of 32 bits.

Description

If you select Register as the access type, the data is read from an IOCNET register. 256 registers are available with a data width of 32 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 256.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.

- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the .

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an IOCNET register. The data format depends on the related parameter settings.

xreg_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks....21

Register64 In

Purpose

To read data from an IOCNET register with a data width of 64 bits.

Description

If you select Register as the access type, the data is read from an IOCNET register. 256 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 … 544.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the .

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1

- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an IOCNET register. The data format depends on the related parameter settings.

xreg64_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks....21

Register Out

Purpose

To write data to an IOCNET register with a data width of 32 bits.

Description

If you select Register as the access type, the data is written to an IOCNET register. 256 registers are available with a data width of 32 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 256.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an IOCNET register. The data format depends on the related parameter settings.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks 21

Register64 Out

Purpose

To write data to an IOCNET register with a data width of 64 bits.

Description

If you select Register64 as the access type, the data is written to an IOCNET register. 256 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 … 544.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Register64 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 sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 255.

xreg64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an IOCNET register. The data format depends on the related parameter settings.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks.....21

Status In

Purpose

To read a digital signal that outputs the state of the FPGA initialization sequence.

Description

There is one digital input channel that is used for the Status In I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

init_done/ 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.

Related topics

References

Overview of the DS6602 FPGA Base Board Frameworks..... 21

I/O Functions of the DS2655M1 I/O Module Framework

Introduction

The *DS2655M1 I/O Module* framework provides analog and digital I/O functionality of SCALEXIO FPGA base board with at least one DS2655M1 Multi-I/O Module.

Where to go from here

Information in this section

Analog In....206
To read data from an analog input signal in the FPGA application.

Analog Out....208
To write data to a analog output signal in the FPGA application.

Digital In....210
To read data from a digital input signal in the FPGA application.

Digital InOut....211
To read/write data to a digital output signal in the FPGA application.

Digital Out....214
To write data to a digital output signal in the FPGA application.

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

I/O Functions of the DS2655 FPGA Base Board Framework....87
The frameworks of the DS2655 FPGA Base Boards provide the standard I/O functionality of the boards.

I/O Functions of the DS6601 FPGA Base Board Framework....123
The DS6601 (KU035) FPGA Base Board framework of the DS6601 FPGA Base Board provide the standard I/O functionality of the board.

I/O Functions of the DS6602 FPGA Base Board Framework..... 159
The DS6602 (KU15P) FPGA Base Board framework of the DS6602 FPGA Base Board provide the standard I/O functionality of the board.

I/O Functions of the DS2655M2 I/O Module Framework..... 217
The DS2655M2 I/O Module framework provides digital I/O functionality of a SCALEXIO FPGA base board with at least one DS2655M2 Digital I/O Module.

I/O Functions of the DS6651 Multi-I/O Module Framework..... 239
The DS6651 Multi-I/O Module framework provides analog and digital I/O functionality of a SCALEXIO FPGA base board with at least one DS6651 Multi-I/O Module.

I/O Functions of the Inter-FPGA Interface Framework..... 281
The Inter-FPGA Interface framework provides access to the I/O module slots of a SCALEXIO FPGA base board to implement an inter-FPGA communication bus.

Analog In

Purpose

To read data from an analog input signal in the FPGA application.

Description

According to the number of physical connections available on the DS2655M1 Multi-I/O Module, you can select the Analog In I/O functions. There are five analog input channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 11 … 15.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(3).Init / Input range Lets you select the input range for all analog input channels.

- 0: -30 V … +30 V
- 1: -5 V … +5 V

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(5).Init / 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.

- 0: mV
- 1: Bit

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 6 … 10.

m<ModuleNumber>_rx<ChannelNumber>_data00 / Data Outputs the current results of analog input channel.

Data type: UFix_16_0

Update rate: 4 Msps

m<ModuleNumber>_rx<ChannelNumber>_ready_pulse / 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.

Data type: UFix_1_0

- 0: No new values are available at the Data port.
- 1: New values are available at the Data port.

m<Module_number>_tx<Channel_number>_data00 / Enable Controls the data port. If set to 1 the ADC is in freerun mode.

Data type: UFix_1_0

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 Multi-I/O Module (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	11	6	10	Analog In - Ch: 11 [Mod: x]
	12	7	27	Analog In - Ch: 12 [Mod: x]
	13	8	44	Analog In - Ch: 13 [Mod: x]
	14	9	12	Analog In - Ch: 14 [Mod: x]
	15	10	29	Analog In - Ch: 15 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M1 I/O Module Framework.....23

Analog Out

Purpose

To write data to a analog output signal in the FPGA application.

Description

According to the number of physical connections available on the DS2655M1 Multi-I/O Module, you can select the Analog Out I/O functions. There are five analog output channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 21 … 25.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(5).Init / 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).

- 0: mV
- 1: Bit

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1 … 5.

Note

The TX channels 1 … 5 are also used by the Digital InOut functions.

- If you want to use the analog out channels with an optimal timing behavior, do not use the related Digital InOut functions at a time.

m<ModuleNumber>_tx<ChannelNumber>_data00_14_0_14_0i / Data Outputs the current results of analog input channel.

Data type: UFix_15_0

Update rate: 7.8125 Msps

m<ModuleNumber>_tx<ChannelNumber>_data00_16_0_16_0i / Enable Controls the data port. If set to 1 the DAC is in freerun mode.

Data type: UFix_1_0

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 (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	21	1	14	Analog Out - Ch: 16 [Mod: x]
	22	2	31	Analog Out - Ch: 17 [Mod: x]
	23	3	48	Analog Out - Ch: 18 [Mod: x]
	24	4	16	Analog Out - Ch: 19 [Mod: x]
	25	5	33	Analog Out - Ch: 20 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M1 I/O Module Framework 23

Digital In

Purpose

To read data from a digital input signal in the FPGA application.

Description

According to the number of physical connections available on the DS2655M1 Multi-I/O Module, you can select the **Digital In** I/O functions. There are 10 digital input channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 10.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(4).Init / Threshold init voltage Specifies the initial voltage value that is used for the threshold in mV.

Data type: UFix_14_0

Range: 0 mV … 10500 mV in 100 mV steps

Update rate: 125 MHz

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1 … 10.

m<ModuleNumber>_rx<ChannelNumber>_trigger00 / Data Outputs the current results of digital input channel.

- 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.

Data type: UFix_1_0

Update rate: 125 MHz

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 Multi-I/O Module (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	1	1	2	Digital In - Ch: 1 [Mod: x]
	2	2	19	Digital In - Ch: 2 [Mod: x]
	3	3	36	Digital In - Ch: 3 [Mod: x]
	4	4	4	Digital In - Ch: 4 [Mod: x]
	5	5	21	Digital In - Ch: 5 [Mod: x]
	6	6	6	Digital In - Ch: 6 [Mod: x]
	7	7	23	Digital In - Ch: 7 [Mod: x]
	8	8	40	Digital In - Ch: 8 [Mod: x]
	9	9	8	Digital In - Ch: 9 [Mod: x]
	10	10	25	Digital In - Ch: 10 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M1 I/O Module Framework 23

Digital InOut

Purpose

To read/write data to a digital output signal in the FPGA application. The **Data direction** port lets you specify the data direction during run time.

Description

According to the number of physical connections available on the DS2655M1 Multi-I/O Module, you can select the **Digital InOut** I/O functions. There are 10 bidirectional digital channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 11 … 20.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(4).Init / Output mode Specifies the output mode.

- 1: LowSide switch

  To drive loads which are connected to VCC.
- 2: HighSide switch

  To drive loads which are connected to GND.
- 3: Push/Pull

  To switch the signal between two different potentials (for example, VCC and GND).

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(6).Init / Drive config Lets you enable/disable the termination of the signal line by a serial resistor.

- 0: 68 Ohm terminated

  The signal line is terminated with 68 Ω.
- 1: The termination is disabled.

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(8).Init / High supply Lets you select the voltage for the high side switch.

- 0: 5 V
- 1: 3.3 V

This electrical interface setting can be changed in ConfigurationDesk.

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.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(10).Init / Digital In threshold init voltage Lets you set the initial threshold voltage for a digital input signal in the range of 0 mV … 10500 mV in steps of 100 mV. This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1 … 10.

m<ModuleNumber>_tx<ChannelNumber>_data01_0_0_1_1i / Data direction Specifies the direction of the digital signal.

- 0: Digital in
- 1: Digital out

Data type: UFix_1_0

Update rate: 125 MHz

m<ModuleNumber>_rx<ChannelNumber>_trigger00 / Inport: Data Outputs the current results of digital input channel.

- 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.

Data type: UFix_1_0

Update rate: 125 MHz

m<ModuleNumber>_tx<ChannelNumber>_data01_0_0_0_0i / Outport: 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.

Data type: UFix_1_0

Update rate: 15.625 MHz

I/O mapping

The following I/O mapping is relevant if you use the *DS2655M1 I/O Module* framework for the bidirectional digital I/O channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M1 Multi-I/O Module (x = 1 … 5)..

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	11	1	2	Digital InOut - Ch: 1 [Mod: x]
	12	2	19	Digital InOut - Ch: 2 [Mod: x]
	13	3	36	Digital InOut - Ch: 3 [Mod: x]
	14	4	4	Digital InOut - Ch: 4 [Mod: x]
	15	5	21	Digital InOut - Ch: 5 [Mod: x]
	16	6	6	Digital InOut - Ch: 6 [Mod: x]
	17	7	23	Digital InOut - Ch: 7 [Mod: x]
	18	8	40	Digital InOut - Ch: 8 [Mod: x]
	19	9	8	Digital InOut - Ch: 9 [Mod: x]
	20	10	25	Digital InOut - Ch: 10 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M1 I/O Module Framework....23

Digital Out

Purpose

To write data to a digital output signal in the FPGA application.

Description

According to the number of physical connections available on the DS2655M1 Multi-I/O Module, you can select the **Digital Out** I/O functions. There are 10 digital output channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 10.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(3).Init / Output mode Specifies the output mode.

- 17: LowSide switch

  To drive loads that are connected to VCC.
- 18: HighSide switch

  To drive loads that are connected to GND.
- 19: Push/Pull

  To switch the signal between two different potentials (for example, VCC and GND).

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(5).Init / Drive config Lets you enable/disable the termination of the signal line by a serial resistor.

- 0: The signal line is terminated with 68 Ω.
- 1: The termination is disabled.

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(7).Init / High supply Lets you select the voltage for the high side switch.

- 0: 5 V
- 1: 3.3 V

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1 … 10.

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / 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.

Data type: UFix_1_0

Update rate: 125 MHz

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 (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	1	1	2	Digital Out - Ch: 1 [Mod: x]
	2	2	19	Digital Out - Ch: 2 [Mod: x]
	3	3	36	Digital Out - Ch: 3 [Mod: x]
	4	4	4	Digital Out - Ch: 4 [Mod: x]
	5	5	21	Digital Out - Ch: 5 [Mod: x]
	6	6	6	Digital Out - Ch: 6 [Mod: x]
	7	7	23	Digital Out - Ch: 7 [Mod: x]
	8	8	40	Digital Out - Ch: 8 [Mod: x]
	9	9	8	Digital Out - Ch: 9 [Mod: x]
	10	10	25	Digital Out - Ch: 10 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M1 I/O Module Framework..... 23

I/O Functions of the DS2655M2 I/O Module Framework

Introduction

The *DS2655M2 I/O Module* framework provides digital I/O functionality of a SCALEXIO FPGA base board with at least one DS2655M2 Digital I/O Module.

Where to go from here

Information in this section

Digital In....218
To read data from a digital input signal in the FPGA application.

Digital Out....221
To write data to a digital output signal in the FPGA application.

Digital Out-Z....224
To write data to a digital output signal in the FPGA application or to switch the output to a high-impedance state (tri-state).

RS232 Rx....227
To receive data values from a RS232 network.

RS232 Tx....228
To transmit data values to a RS232 network.

RS485 Rx....230
To receive data values from a RS485 network in simplex mode.

RS485 RxTx....232
To implement communication via a RS485 network in half-duplex mode.

RS485 Tx....235
To transmit data values to a RS485 network in simplex mode.

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

I/O Functions of the DS2655 FPGA Base Board Framework.....87
The frameworks of the DS2655 FPGA Base Boards provide the standard I/O functionality of the boards.

I/O Functions of the DS6601 FPGA Base Board Framework.....123
The DS6601 (KU035) FPGA Base Board framework of the DS6601 FPGA Base Board provide the standard I/O functionality of the board.

I/O Functions of the DS6602 FPGA Base Board Framework.....159
The DS6602 (KU15P) FPGA Base Board framework of the DS6602 FPGA Base Board provide the standard I/O functionality of the board.

I/O Functions of the DS2655M1 I/O Module Framework.....205
The DS2655M1 I/O Module framework provides analog and digital I/O functionality of SCALEXIO FPGA base board with at least one DS2655M1 Multi-I/O Module.

I/O Functions of the DS6651 Multi-I/O Module Framework.....239
The DS6651 Multi-I/O Module framework provides analog and digital I/O functionality of a SCALEXIO FPGA base board with at least one DS6651 Multi-I/O Module.

I/O Functions of the Inter-FPGA Interface Framework.....281
The Inter-FPGA Interface framework provides access to the I/O module slots of a SCALEXIO FPGA base board to implement an inter-FPGA communication bus.

Digital In

Purpose

To read data from a digital input signal in the FPGA application.

Description

According to the number of physical connections available on the DS2655M2 Digital I/O Module, you can select the **Digital In** I/O functions. There are up to 32 digital input channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 32.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(9).Init / Threshold init voltage Specifies the initial voltage value that is used for the threshold in mV.

Data type: UFix_14_0

Range: 0 mV … 10500 mV in 100 mV steps

Update rate: 125 MHz

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1 … 32.

m<ModuleNumber>_io2raw_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>i / 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: 125 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 🕮).

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)

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 🕮).

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M2 I/O Module Framework 25

Digital Out

Purpose

To write data to a digital output signal in the FPGA application.

Description

According to the number of physical connections available on the DS2655M2 Digital I/O Module, you can select the **Digital Out** I/O functions. There are up to 32 digital output channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 32.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(7).Init / Output mode Specifies the output mode.

- 5: LowSide switch

  To drive loads which are connected to VCC.
- 6: HighSide switch

  To drive loads which are connected to GND.
- 7: Push/Pull

  To switch the signal between two different potentials (for example, VCC and GND).

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(9).Init / Drive config Lets you enable/disable the termination of the signal line by a serial resistor.

- 0: The signal line is terminated with 68 Ω.
- 1: The termination is disabled.

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(11).Init / High supply Lets you select the voltage for the high side switch.

- 0: 5 V
- 1: 3.3 V

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1 … 32.

m<ModuleNumber>_raw2io_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>i / Data Outputs a signal in the specified range.

To set the voltage level, use the **High supply**.

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: 125 MHz

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 🕮).

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)

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 🕮).

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M2 I/O Module Framework 25

Digital Out-Z

Purpose

To write data to a digital output signal in the FPGA application or to switch the output to a high-impedance state (tri-state).

Description

According to the number of physical connections available on the DS2655M2 Digital I/O Module, you can select the Digital Out-Z I/O functions. There are up to 16 digital output channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 33 … 48.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(14).Init / Drive config Lets you enable/disable the termination of the signal line by a serial resistor.

- 0: The signal line is terminated with 68 Ω.
- 1: The termination is disabled.

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(16).Init / High supply Lets you select the voltage for the high side switch.

- 0: 5 V
- 1: 3.3 V

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1, 3, 5, … ,31.

m<ModuleNumber>_raw2io_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>i / 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 .

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: 125 MHz

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 🕮).

m<ModuleNumber>_raw2io_<ChannelNumber+1>_<ChannelNumber+1>_<ChannelNumber+1>_ <ChannelNumber+1>i / 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.

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 🕮).

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M2 I/O Module Framework 25

RS232 Rx

Purpose

To receive data values from a RS232 network.

Description

According to the number of physical connections available on the DS2655M2 Digital I/O Module, you can select the **RS232 Rx** I/O functions. There are up to 8 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 33 … 40.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 2, 6, 10, … ,30.

m<ModuleNumber>_io2raw_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>i / 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 🕮).

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 🕮).

Related topics

References

Overview of the DS2655M2 I/O Module Framework.....25

RS232 Tx

Purpose

To transmit data values to a RS232 network.

Description

According to the number of physical connections available on the DS2655M2 Digital I/O Module, you can select the **RS232 Tx** I/O functions. There are up to 8 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 49 … 56.

IOProperties.Out.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1 … 32.

m<ModuleNumber>_raw2io_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>i / Data Outputs the data to be send to the RS232 Tx channel.

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.

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 🕮).

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 🕮).

Related topics

References

Overview of the DS2655M2 I/O Module Framework 25

RS485 Rx

Purpose

To receive data values from a RS485 network in simplex mode.

Description

According to the number of physical connections available on the DS2655M2 Digital I/O Module, you can select the **RS485 Rx** I/O functions. There are up to 8 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 41 … 48.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(7).Init / RS485 Termination Lets you enable/disable the termination of the signal line by a serial resistor.

- 0: The termination is disabled.
- 1: The signal line is terminated by an internal 120 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1, 5, 9, ... , 29.

m<ModuleNumber>_io2raw_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>_<ChannelNumber>i / 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).

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 🕮).

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M2 I/O Module Framework 25

RS485 RxTx

Purpose

To implement communication via a RS485 network in half-duplex mode.

Description

According to the number of physical connections available on the DS2655M2 Digital I/O Module, you can select the **RS485 RxTx** I/O functions. There are up to 8 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 41 … 48.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(17).Init / RS485 Termination Lets you enable/disable the termination of the signal line by a serial resistor.

- 0: The termination is disabled.
- 1: The signal line is terminated by an internal 120 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1, 5, 9, … , 29.

m<ModuleNumber+1>_raw2io_<ChannelNumber+1>_ <ChannelNumber+1>_<ChannelNumber+1>_ <ChannelNumber+1>i / 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.

m<ModuleNumber>_raw2io_<ChannelNumber+2>_ <ChannelNumber+2>_<ChannelNumber+2>_ <ChannelNumber+2>i / 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.

m<ModuleNumber>_io2raw_<ChannelNumber>_ <ChannelNumber>_<ChannelNumber>_<ChannelNumber>i / 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).

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.

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 RxTx 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 🕮).

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M2 I/O Module Framework 25

RS485 Tx

Purpose

To transmit data values to a RS485 network in simplex mode.

Description

According to the number of physical connections available on the DS2655M2 Digital I/O Module, you can select the **RS485 Tx** I/O functions. There are up to 8 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 57 … 64.

IOProperties.Out.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(14).Init / RS485 Termination Lets you enable/disable the termination of the signal line by a serial resistor.

- 0: The termination is disabled.
- 1: The signal line is terminated by an internal 120 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1, 5, 9, … , 29.

m<ModuleNumber>_raw2io_<ChannelNumber>_ <ChannelNumber>_<ChannelNumber>_<ChannelNumber>i / 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.

m<ModuleNumber>_raw2io_<ChannelNumber+1>_ <ChannelNumber+1>_<ChannelNumber+1>_ <ChannelNumber+1>i / Data 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.

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	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 🕮).

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS2655M2 I/O Module Framework..... 25

I/O Functions of the DS6651 Multi-I/O Module Framework

Introduction

The *DS6651 Multi-I/O Module* framework provides analog and digital I/O functionality of a SCALEXIO FPGA base board with at least one DS6651 Multi-I/O Module.

Where to go from here

Information in this section

Analog In....240
To read data from an analog input signal in the FPGA application.

Analog In-L....244
To measure data of the 16-bit A/D converter.

Analog Out....247
To write data to an analog output signal in the FPGA application.

Analog Out-T....248
To write data to an analog output signal in the FPGA application.

Digital In....250
To read data from a digital input signal in the FPGA application.

Digital In/Out-Z....251
To read or write data to or from a digital signal in the FPGA application, or to switch the output to a high-impedance state (tristate).

Digital Out....255
To write data to a digital output signal in the FPGA application.

Digital Out-Z....257
To write data to a digital output signal in the FPGA application or to switch the output to a high-impedance state (tristate).

RS485 Rx....260
To implement communication via a RS485 network in simplex mode.

RS485 Rx/Tx.....262
To implement communication via a RS485 network in half-duplex mode.

RS485 Tx.....265
To implement communication via a RS485 network in simplex mode.

Trigger.....267
To trigger the analog measurement.

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

I/O Functions of the DS2655 FPGA Base Board Framework.....87
The frameworks of the DS2655 FPGA Base Boards provide the standard I/O functionality of the boards.

I/O Functions of the DS6601 FPGA Base Board Framework.....123
The DS6601 (KU035) FPGA Base Board framework of the DS6601 FPGA Base Board provide the standard I/O functionality of the board.

I/O Functions of the DS6602 FPGA Base Board Framework.....159
The DS6602 (KU15P) FPGA Base Board framework of the DS6602 FPGA Base Board provide the standard I/O functionality of the board.

I/O Functions of the DS2655M1 I/O Module Framework.....205
The DS2655M1 I/O Module framework provides analog and digital I/O functionality of SCALEXIO FPGA base board with at least one DS2655M1 Multi-I/O Module.

I/O Functions of the DS2655M2 I/O Module Framework.....217
The DS2655M2 I/O Module framework provides digital I/O functionality of a SCALEXIO FPGA base board with at least one DS2655M2 Digital I/O Module.

I/O Functions of the Inter-FPGA Interface Framework.....281
The Inter-FPGA Interface framework provides access to the I/O module slots of a SCALEXIO FPGA base board to implement an inter-FPGA communication bus.

Analog In

Purpose

To read data from an analog input signal in the FPGA application.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the **Analog In** I/O functions. There are four channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 25 … 28.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(2).Init / Input range Lets you select the input voltage range that can be converted from analog to digital for the chosen ADC channel.

- 0: -60 V ... +60 V
- 1: -10 V ... +10 V
- 2: -5 V ... +5 V
- 3: -1 V ... +1 V

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(4).Init / Scaling Lets you select whether the I/O function scales the measuring results of the A/D converter to mV.

- 0: 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.

- 1: Bit

  To output the raw measuring results as a signed Bit value.

  Value range: -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.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(6).Init / Trigger mode Lets you select the trigger mode and source for sampling the analog input voltage.

- 15: Free running

  The ADC samples the input voltage with a fixed sample period that is set by the Sample period parameter.
- 258: Trigger 1

  The ADC samples the input voltage with each trigger impulse provided by a Trigger I/O function. Refer to Trigger on page 267.
- 274: Trigger 2

  The ADC samples the input voltage with each trigger impulse provided by a Trigger I/O function. Refer to Trigger on page 267.
- Digital In:
  - `16 · (x-1) + 2`: Digital In, <x>, rising edge, 8 ns filter
  - `16 · (x-1) + 1026`: Digital In, <x>, rising edge, no filter
  - `16 · (x-1) + 1538`: Digital In, <x>, falling edge, 8 ns filter
  - `16 · (x-1) + 514`: Digital In, <x>, falling edge, no filter

  The selected digital input channel triggers the sampling of the analog input signal:
  - <x>: Indicates the channel number of the digital input channel.
  - Rising edge: A 0 to 1 transition of a digital input signal triggers the ADC.
  - Falling edge: A 1 to 0 transition of a digital input signal triggers the ADC.
  - 8 ns filter: The digital signal is filtered by a digital low-pass filter with a time constant of 8 ns.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(8).Init / 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 $25 \le n_{selected} \le 3{,}750{,}000{,}000$.

The resulting sample period is in the range 200 ns ... 30 s.

Ports

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 23 … 26.

m<ModuleNumber>_rx<ChannelNumber>_data00 / Data Outputs the measured values of the 16-bit A/D converter.

Data type: Fix_22_5

Data width: 1

Value range: Depends on the setting of the Scaling parameter.

m<ModuleNumber>_rx<ChannelNumber>_ready_pulse / 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.

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.

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 (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	25	23	29	Analog In - Ch: 23 [Mod: x]
	26	24	14	Analog In - Ch: 24 [Mod: x]
	27	25	31	Analog In - Ch: 25 [Mod: x]
	28	26	48	Analog In - Ch: 26 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework.....27

Analog In-L

Purpose

To measure data of the 16-bit A/D converter.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the Analog In-L I/O functions. There are two channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 29 … 30.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(2).Init / Input range Lets you select the input voltage range that can be converted from analog to digital for the chosen ADC channel.

- 0: -60 V ... +60 V
- 1: -10 V ... +10 V
- 2: -5 V ... +5 V
- 3: -1 V ... +1 V

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(4).Init / Scaling Lets you select whether the I/O function scales the measuring results of the A/D converter to mV.

- 0: 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.
- 1: Bit

  To output the raw measuring results as a signed Bit value.

  Value range: -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.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(6).Init / Load Config Lets you enable a 220 Ω resistor between the analog signal and the signal reference.

- 0: Disabled
- 1: 220 Ohm

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(8).Init / Trigger mode Lets you select the trigger mode and source for sampling the analog input voltage.

- 15: Free running

  The ADC samples the input voltage with a fixed sample period that is set by the **Sample period** parameter.
- 258: Trigger 1

  The ADC samples the input voltage with each trigger impulse provided by a **Trigger** I/O function. Refer to Trigger on page 267.
- 274: Trigger 2

  The ADC samples the input voltage with each trigger impulse provided by a **Trigger** I/O function. Refer to Trigger on page 267.
- Digital In:
  - `16 · (x-1) + 2`: Digital In, <x>, rising edge, 8 ns filter
  - `16 · (x-1) + 1026`: Digital In, <x>, rising edge, no filter
  - `16 · (x-1) + 1538`: Digital In, <x>, falling edge, 8 ns filter
  - `16 · (x-1) + 514`: Digital In, <x>, falling edge, no filter

  The selected digital input channel triggers the sampling of the analog input signal:
  - <x>: Indicates the channel number of the digital input channel.
  - Rising edge: A 0 to 1 transition of a digital input signal triggers the ADC.
  - Falling edge: A 1 to 0 transition of a digital input signal triggers the ADC.
  - 8 ns filter: The digital signal is filtered by a digital low-pass filter with a time constant of 8 ns.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(10).Init / 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.

Ports

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 27 … 28.

m<ModuleNumber>_rx<ChannelNumber>_data00 / Data Outputs the measured values of the 16-bit A/D converter.

Data type: Fix_22_5

Data width: 1

Value range: Depends on the setting of the Scaling parameter.

m<ModuleNumber>_rx<ChannelNumber>_ready_pulse / 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.

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.

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 (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	29	27	16	Analog In-L - Ch: 27 [Mod: x]
	30	28	33	Analog In-L - Ch: 28 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework.....27

Analog Out

Purpose

To write data to an analog output signal in the FPGA application.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the **Analog Out** I/O functions. There are four channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 41 … 44.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(5).Init / Scaling Lets you select the scaling of the input data.

- 0: mV

  The valid input port range is -10,000 … +10,000 mV.
- 1: Bit

  The valid input port range is -32,768 … +32,776 (16-bit converter).

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 17 … 20.

m<ModuleNumber>_tx<ChannelNumber>_data00_17_0_17_0i / Data Outputs a voltage signal in the specified range.

Data type: Fix_18_2

Data width: 1

Update rate: 10.417 MS/s

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 (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	41	17	8	Analog Out - Ch: 17 [Mod: x]
	42	18	25	Analog Out - Ch: 18 [Mod: x]
	43	19	10	Analog Out - Ch: 19 [Mod: x]
	44	20	27	Analog Out - Ch: 20 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework 27

Analog Out-T

Purpose

To write data to an analog output signal in the FPGA application.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the Analog Out-T I/O functions. There are two channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 45 … 46.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(5).Init / Scaling Lets you select the scaling of the input data.

- 0: mV

  The valid input port range corresponds to the settings of the Mode parameter.
- 1: Bit

  The valid input port range is -32,768 … +32,776 (16-bit converter).

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(7).Init / Mode Lets you select the converter mode of the analog output channel.

- 0: ±10 VDC

  The DA converter directly outputs the voltage signal without using a transformer. The output voltage range is -10 VDC … +10 VDC.
- 1: ±20 V transformer coupled AC

  The DA converter outputs the voltage signal via a transformer. The output voltage range is -20 VAC … +20 VAC.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 21 … 22.

m<ModuleNumber>_tx<ChannelNumber>_data00_17_0_17_0i / Data Outputs a voltage signal in the specified range.

Data type: Fix_18_2

Data width: 1

Update rate: 10.417 MS/s

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 (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	45	21	44	Analog Out-T - Ch: 21 [Mod: x]
	46	22	12	Analog Out-T - Ch: 22 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework 27

Digital In

Purpose

To read data from a digital input signal in the FPGA application.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the Digital In I/O functions. There are up to 16 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 16.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).Parameter(8).Init / 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.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1 … 16.

m<ModuleNumber>_rx<ChannelNumber>_trigger00 / 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

I/O mapping

The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital I/O channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module (x = 1 … 5).

Outport	I/O Function Number	Channel number	Connector Pin	Signal
Data	1	1	18	Digital In - Ch: 1 [Mod: x]
	2	2	2	Digital In - Ch: 2 [Mod: x]
	3	3	35	Digital In - Ch: 3 [Mod: x]
	4	4	19	Digital In - Ch: 4 [Mod: x]
	5	5	3	Digital In - Ch: 5 [Mod: x]
	6	6	36	Digital In - Ch: 6 [Mod: x]
	7	7	20	Digital In - Ch: 7 [Mod: x]
	8	8	4	Digital In - Ch: 8 [Mod: x]
	9	9	37	Digital In - Ch: 9 [Mod: x]
	10	10	21	Digital In - Ch: 10 [Mod: x]
	11	11	22	Digital In - Ch: 11 [Mod: x]
	12	12	6	Digital In - Ch: 12 [Mod: x]
	13	13	39	Digital In - Ch: 13 [Mod: x]
	14	14	23	Digital In - Ch: 14 [Mod: x]
	15	15	7	Digital In - Ch: 15 [Mod: x]
	16	16	40	Digital In - Ch: 16 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework 27

Digital In/Out-Z

Purpose

To read or write data to or from a digital signal in the FPGA application, or to switch the output to a high-impedance state (tristate).

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the **Digital In/Out-Z** I/O functions. There are up to four channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 4.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(6).Init / Output Mode Lets you select the output mode.

- 49: 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.
- 50: 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.
- 51: 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.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(8).Init / Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- 1: Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 0: 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.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(10).Init / High Supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

- 0: 5 V
- 1: 3.3 V

This electrical interface setting can be changed in ConfigurationDesk.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(14).Init / 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.

Ports

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1, 5, 9, 13.

m<ModuleNumber>_rx<ChannelNumber>_trigger00 / 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

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

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.

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / 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.

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:

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 🕮).

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / 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.

I/O mapping

The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital I/O channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module (x = 1 … 5).

Ports	I/O Function Number	Channel Number	Connector Pin	Signal
Data In and Data Out	1	1	18	Digital In/Out-Z - Ch: 1-3 [Mod: x]
			2	No signal
			35	No signal
			19	Usable by other I/O functions
	2	5	3	Digital In/Out-Z - Ch: 5-7 [Mod: x]
			36	No signal
			20	No signal
			4	Usable by other I/O functions
	3	9	37	Digital In/Out-Z - Ch: 9-11 [Mod: x]
			21	No signal
			22	No signal
			6	Usable by other I/O functions
	4	13	39	Digital In/Out-Z - Ch: 13-15 [Mod: x]
			23	No signal
			7	No signal
			40	Usable by other I/O functions

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework 27

Digital Out

Purpose

To write data to a digital output signal in the FPGA application.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the **Digital Out** I/O functions. There are up to 16 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 16.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(4).Init / Output Mode Lets you select the output mode.

- 17: 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.
- 18: 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.
- 19: 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.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(6).Init / Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- 1: Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 0: 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.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(8).Init / High Supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

- 0: 5 V
- 1: 3.3 V

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1 … 16.

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / Data Outputs a signal in the specified range.

To set the voltage level, use the High supply parameter.

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 🕮).

I/O mapping

The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital I/O channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	1	1	18	Digital Out - Ch: 1 [Mod: x]
	2	2	2	Digital Out - Ch: 2 [Mod: x]
	3	3	35	Digital Out - Ch: 3 [Mod: x]
	4	4	19	Digital Out - Ch: 4 [Mod: x]
	5	5	3	Digital Out - Ch: 5 [Mod: x]
	6	6	36	Digital Out - Ch: 6 [Mod: x]
	7	7	20	Digital Out - Ch: 7 [Mod: x]
	8	8	4	Digital Out - Ch: 8 [Mod: x]
	9	9	37	Digital Out - Ch: 9 [Mod: x]
	10	10	21	Digital Out - Ch: 10 [Mod: x]
	11	11	22	Digital Out - Ch: 11 [Mod: x]
	12	12	6	Digital Out - Ch: 12 [Mod: x]
	13	13	39	Digital Out - Ch: 13 [Mod: x]
	14	14	23	Digital Out - Ch: 14 [Mod: x]
	15	15	7	Digital Out - Ch: 15 [Mod: x]
	16	16	40	Digital Out - Ch: 16 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework 27

Digital Out-Z

Purpose

To write data to a digital output signal in the FPGA application or to switch the output to a high-impedance state (tristate).

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the **Digital Out-Z** I/O functions. There are up to 8 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 17 … 24.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(5).Init / Output Mode Lets you select the output mode.

- 33: 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.
- 34: 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.
- 35: 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.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(7).Init / Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- 1: Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 0: 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.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(9).Init / High Supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

- 0: 5 V
- 1: 3.3 V

This electrical interface setting can be changed in ConfigurationDesk.

Ports

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1, 3, 5, … 15.

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / Data Outputs a signal in the specified range.

To set the voltage level, use the **High supply** parameter.

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:

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 🕮).

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / 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.

I/O mapping

The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital I/O channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	17	1	18 2	Digital Out-Z - Ch: 1-2 [Mod: x] No signal
	18	3	35 19	Digital Out-Z - Ch: 3-4 [Mod: x] No signal
	19	5	3 36	Digital Out-Z - Ch: 5-6 [Mod: x] No signal
	20	7	20 4	Digital Out-Z - Ch: 7-8 [Mod: x] No signal
	21	9	37 21	Digital Out-Z - Ch: 9-10 [Mod: x] No signal
	22	11	22 6	Digital Out-Z - Ch: 11-12 [Mod: x] No signal
	23	13	39 23	Digital Out-Z - Ch: 13-14 [Mod: x] No signal
	24	15	7 40	Digital Out-Z - Ch: 15-16 [Mod: x] No signal

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework.....27

RS485 Rx

Purpose

To implement communication via a RS485 network in simplex mode.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the **RS485 Rx** I/O functions. There are up to 8 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 17 … 24.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(8).Init / RS485 Termination Lets you enable an internal termination between the signal lines. The setting can be overwritten by the RS485 termination ports.

- 0: Open

  The signal lines are not terminated.
- 1: Terminated

  An internal 120 Ω/5 nF RC termination terminates the signal lines.

This electrical interface setting can be changed in ConfigurationDesk.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1, 3, 5, … 15.

m<ModuleNumber>_rx<ChannelNumber>_trigger00 / RX 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).

I/O mapping

The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital I/O channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	17	1	18	RS485 Rx- - Ch: 1-2 [Mod: x]
			2	RS485 Rx+ - Ch: 1-2 [Mod: x]
	18	3	35	RS485 Rx- - Ch: 3-4 [Mod: x]
			19	RS485 Rx+ - Ch: 3-4 [Mod: x]
	19	5	3	RS485 Rx- - Ch: 5-6 [Mod: x]
			36	RS485 Rx+ - Ch: 5-6 [Mod: x]
	20	7	20	RS485 Rx- - Ch: 7-8 [Mod: x]
			4	RS485 Rx+ - Ch: 7-8 [Mod: x]
	21	9	37	RS485 Rx- - Ch: 9-10 [Mod: x]
			21	RS485 Rx+ - Ch: 9-10 [Mod: x]
	22	11	22	RS485 Rx- - Ch: 11-12 [Mod: x]
			6	RS485 Rx+ - Ch: 11-12 [Mod: x]
	23	13	39	RS485 Rx- - Ch: 13-14 [Mod: x]
			23	RS485 Rx+ - Ch: 13-14 [Mod: x]
	24	15	7	RS485 Rx- - Ch: 15-16 [Mod: x]
			40	RS485 Rx+ - Ch: 15-16 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework.....27

RS485 Rx/Tx

Purpose

To implement communication via a RS485 network in half-duplex mode.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the **RS485 Rx/Tx** I/O functions. There are up to four channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 37 … 40.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(10).Init / High supply Lets you select the differential output voltage.

- 0: 5 V
- 1: 3.3 V

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(16).Init / RS485 Termination Lets you enable an internal termination between the signal lines. The setting can be overwritten by the RS485 termination ports.

- 0: Open

  The signal lines are not terminated.
- 1: Terminated

  An internal 120 Ω/5 nF RC termination terminates the signal lines.

This electrical interface setting can be changed in ConfigurationDesk.

Ports

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1, 5, 9, 13.

m<ModuleNumber>_rx<ChannelNumber>_trigger00 / 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).

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / 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 setting of the High Supply parameter.

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.

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / 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 (tristate). 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.

I/O mapping

The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital I/O channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module (x = 1 … 5).

Ports	I/O Function Number	Channel Number	Connector Pin	Signal
Rx Data and Tx Data	37	1	18	RS485 RxTx- - Ch: 1-3 [Mod: x]
			2	RS485 RxTx+ - Ch: 1-3 [Mod: x]
			35	No signal
			19	Usable by other I/O functions
	38	5	3	RS485 RxTx- - Ch: 5-7 [Mod: x]
			36	RS485 RxTx+ - Ch: 5-7 [Mod: x]
			20	No signal
			4	Usable by other I/O functions
	39	9	37	RS485 RxTx- - Ch: 9-11 [Mod: x]
			21	RS485 RxTx+ - Ch: 9-11 [Mod: x]
			22	No signal
			6	Usable by other I/O functions
	40	13	39	RS485 RxTx- - Ch: 13-15 [Mod: x]
			23	RS485 RxTx+ - Ch: 13-15 [Mod: x]
			7	No signal
			40	Usable by other I/O functions

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework 27

RS485 Tx

Purpose

To implement communication via a RS485 network in simplex mode.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the **RS485 Tx** I/O functions. There are up to 8 channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 29 … 36.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(9).Init / High supply Lets you select the differential output voltage.

- 0: 5 V
- 1: 3.3 V

IOProperties.Out.Fct(<IOFunctionNumber> + ioOutOffset<ModuleNumber>).Parameter(15).Init / RS485 Termination Lets you enable an internal termination between the signal lines. The setting can be overwritten by the RS485 termination ports.

- 0: Open

  The signal lines are not terminated.
- 1: Terminated

  An internal 120 Ω/5 nF RC termination terminates the signal lines.

This electrical interface setting can be changed in ConfigurationDesk.

Ports

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 1, 3, 5, … 15.

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / Data Outputs the data to the RS485 network if the Enable port is set to 1.

The differential output voltage level depends on the setting of the High Supply parameter.

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.

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / 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.

I/O mapping

The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital I/O channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module (x = 1 … 5).

Outport	I/O Function Number	Channel Number	Connector Pin	Signal
Data	29	1	18	RS485 Tx- - Ch: 1-2 [Mod: x]
			2	RS485 Tx+ - Ch: 1-2 [Mod: x]
	30	3	35	RS485 Tx- - Ch: 3-4 [Mod: x]
			19	RS485 Tx+ - Ch: 3-4 [Mod: x]
	31	5	3	RS485 Tx- - Ch: 5-6 [Mod: x]
			36	RS485 Tx+ - Ch: 5-6 [Mod: x]
	32	7	20	RS485 Tx- - Ch: 7-8 [Mod: x]
			4	RS485 Tx+ - Ch: 7-8 [Mod: x]
	33	9	37	RS485 Tx- - Ch: 9-10 [Mod: x]
			21	RS485 Tx+ - Ch: 9-10 [Mod: x]
	34	11	22	RS485 Tx- - Ch: 11-12 [Mod: x]
			6	RS485 Tx+ - Ch: 11-12 [Mod: x]
	35	13	39	RS485 Tx- - Ch: 13-14 [Mod: x]
			23	RS485 Tx+ - Ch: 13-14 [Mod: x]
	36	15	7	RS485 Tx- - Ch: 15-16 [Mod: x]
			40	RS485 Tx+ - Ch: 15-16 [Mod: x]

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework 27

Trigger

Purpose

To trigger the analog measurement.

Description

According to the number of physical connections available on the DS6651 Multi-I/O Module, you can select the **Trigger** I/O functions. There are two channels.

The module number can be specified in the range 1 … 5.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 47 … 48.

IOProperties.In.Fct(<IOFunctionNumber> + ioInOffset<ModuleNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 17 … 18.

m<ModuleNumber>_tx<ChannelNumber>_trigger00 / 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 on page 240 and Analog In-L on page 244.

I/O mapping

No external connection.

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Overview of the DS6651 Multi-I/O Module Framework.....27

I/O Functions of the DS660X_MGT Framework

Introduction

The DS660X_MGT framework provides access to an MGT module. An MGT module can be installed to the following SCALEXIO FPGA base boards:

- DS6601 FPGA Base Board
- DS6602 FPGA Base Board

Where to go from here

Information in this section

Aurora 64b66b In....270
To provide read access to the installed MGT module.

Aurora 64b66b Out....271
To provide write access to the installed MGT module.

Aurora 64b66b 128 Bit In....273
To provide 128-bit-based read access to the installed MGT module.

Aurora 64b66b 128 Bit Out....275
To provide 128-bit-based write access to the installed MGT module.

MGT In....277
To provide the information about the connection between the GTH transceivers and the MGT module and to specify the reference clock frequency.

MGT Out....278
To provide the information about the connection between the GTH transceivers and the MGT module.

Aurora 64b66b In

Purpose

To provide read access to the installed MGT module.

Description

The DS6601 and DS6602 FPGA Base Boards provide an MGT module slot to install an MGT Module. The Aurora 64b66b In I/O function lets you configure read access to the MGT communication.

This I/O function is not considered when you generate the processor interface model.

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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file of the FPGA base board.

The I/O function number can be specified in the range 1 … 4.

The used module slot is 1 (mgtModuleNr = 1).

hcfw.IOProperties.In.Fct(<IOFunctionNumber> + mgtInOffset (1)).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The lane number can be specified in the range 1 … 4.

mgt_lane<LaneNumber>_rx_data / Data Inport to read 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 latency (latency between sender and receiver via an optical loopback):

- 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 Handcode Interface Guide 🕮).

Data type: U_Fix64_0

Data width: 1

mgt_lane<LaneNumber>_rx_valid / Data New Inport to indicate whether a new data value was received by the MGT module.

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.

Data type: U_Fix1_0

Data width: 1

I/O mapping

The MGT communication bus uses the MPO connector of the FPGA Base Board.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Aurora 64b66b Out

Purpose

To provide write access to the installed MGT module.

Description

The DS6601 and DS6602 FPGA Base Boards provide an MGT module slot to install an MGT Module. The Aurora 64b66b Out I/O function lets you configure write access to the MGT communication.

This I/O function is not considered when you generate the processor interface model.

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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file of the FPGA base board.

The I/O function number can be specified in the range 1 … 4.

The used module slot is 1 (mgtModuleNr = 1).

hcfw.IOProperties.Out.Fct(<IOFunctionNumber> + mgtOutOffset (1)).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The lane number can be specified in the range 1 … 4.

mgt_lane<LaneNumber>_tx_valid / Enable Outport to enable the write access to the MGT communication bus:

- 0: No write access.
- 1: The Data port value of the current clock cycle is written on the MGT communication bus.

Data type: UFix_1_0

Data width: 1

mgt_lane<LaneNumber>_tx_data / Data Outport to write 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 (latency between sender and receiver via an optical loopback):

- 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 Handcode Interface Guide 🕮).

Data type: U_Fix64_0

Data width: 1

mgt_lane<LaneNumber>_tx_ready / Ready Inport to read a flag that indicates that the MGT module is ready to write new data on the MGT communication bus:

- 0: The MGT module is busy.
- 1: New data values can be written to the MGT communication bus.

Data type: UFix_1_0

Data width: 1

I/O mapping

The MGT communication bus uses the MPO connector of the FPGA Base Board.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Aurora 64b66b 128 Bit In

Purpose

To provide 128-bit-based read access to the installed MGT module.

Description

The DS6601 and DS6602 FPGA Base Boards provide an MGT module slot to install an MGT Module. The Aurora 64b66b 128 Bit In I/O function lets you configure read access to the MGT communication.

This I/O function is not considered when you generate the processor interface model.

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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file of the FPGA base board.

The I/O function number can be specified in the range 5 … 8.

The used module slot is 1 (mgtModuleNr = 1).

hcfw.IOProperties.In.Fct(<IOFunctionNumber> + mgtInOffset (1)).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The lane number can be specified in the range 1 … 4.

mgt_lane<LaneNumber>_rx_data / Data Inport to read 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 (Latency between sender and receiver via an optical loopback):

- With maximum data rate: Max. 6.272 µs, typ. 6.192 µs

  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 Handcode Interface Guide 🕮).

Data type: UFix_128_0

Data width: 1

mgt_lane<LaneNumber>_rx_valid / Data New Inport to indicate whether a new data value was received by the MGT module.

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.

Data type: U_Fix1_0

Data width: 1

I/O mapping

The MGT communication bus uses the MPO connector of the FPGA Base Board.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

Aurora 64b66b 128 Bit Out

Purpose

To provide 128-bit-based write access to the installed MGT module.

Description

The DS6601 and DS6602 FPGA Base Boards provide an MGT module slot to install an MGT Module. The Aurora 64b66b 128 Bit Out I/O function lets you configure write access to the MGT communication.

This I/O function is not considered when you generate the processor interface model.

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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file of the FPGA base board.

The I/O function number can be specified in the range 5 … 8.

The used module slot is 1 (mgtModuleNr = 1).

hcfw.IOProperties.Out.Fct(<IOFunctionNumber> + mgtOutOffset (1)).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The lane number can be specified in the range 1 … 4.

mgt_lane<LaneNumber>_tx_valid / Enable Outport to enable the write access to the MGT communication bus:

- 0: No write access.
- 1: The Data port value of the current clock cycle is written on the MGT communication bus.

Data type: UFix_1_0

Data width: 1

mgt_lane<LaneNumber>_tx_data / Data Outport to write 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 (latency between sender and receiver via an optical loopback):

- With maximum data rate: Max. 6.272 µs, typ. 6.192 µs

  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 Handcode Interface Guide 🕮).

Data type: UFix_128_0

Data width: 1

mgt_lane<LaneNumber>_tx_ready / Ready Inport to read a flag that indicates that the MGT module is ready to write new data on the MGT communication bus:

- 0: The MGT module is busy.
- 1: New data values can be written to the MGT communication bus.

Data type: UFix_1_0

Data width: 1

I/O mapping

The MGT communication bus uses the MPO connector of the FPGA Base Board.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

MGT In

Purpose

To provide the information about the connection between the GTH transceivers and the MGT module and to specify the reference clock frequency.

Description

The DS6601 and DS6602 FPGA Base Boards provide an MGT module slot to install an MGT Module. The MGT In I/O function provides the information about the connection between the GTH transceivers and the MGT module and lets you 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file of the FPGA base board.

The I/O function number is 9.

The used module slot is 1 (mgtModuleNr = 1).

hcfw.IOProperties.In.Fct(<IOFunctionNumber> + mgtInOffset (1)).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

hcfw.IOProperties.In.Fct(<IOFunctionNumber> + mgtInOffset (1)).Parameter((<IOFunctionNumber> + 1)).Init / 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 DS6601, DS6602: Coding a Customized MGT Protocol (RTI FPGA Programming Blockset Handcode Interface Guide 🕮).

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

MGT_CLK_P / CLK_P Provides the MGT reference clock frequency.

The port represents the differential signal of an internal clock. The port must be connected to a function that provides the configuration for the GTH transceivers.

The reference frequency is specified by the MGT reference clock frequency parameter.

Data type: UFix_1_0

Data width: 1

MGT_CLK_N / CLK_N Provides the MGT reference clock frequency.

The port represents the differential signal of an internal clock. The port must be connected to a function that provides the configuration for the GTH transceivers.

The reference frequency is specified by the MGT reference clock frequency parameter.

Data type: UFix_1_0

Data width: 1

MGT_RX_P / RX_P Reads the raw data from the MGT module.

The port represents the differential output signals of the MGT module. The port must be connected to a function 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.

MGT_RX_N / RX_N Reads the raw data from the MGT module.

The port represents the differential output signals of the MGT module. The port must be connected to a function 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.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

MGT Out

Purpose

To provide the information about the connection between the GTH transceivers and the MGT module.

Description

The DS6601 and DS6602 FPGA Base Boards provide an MGT module slot to install an MGT Module. The MGT Out I/O function provides 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file of the FPGA base board.

The I/O function number is 9.

The used module slot is 1 (mgtModuleNr = 1).

hcfw.IOProperties.In.Fct(<IOFunctionNumber> + mgtInOffset (1)).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

MGT_TX_P / TX_P Writes data to the MGT module.

The port represents the differential signals of the GTH transceivers that are connected to the MGT module. The port must be connected to a function 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** I/O function lets you specify the reference frequency that is used to generate the MGT clock frequency. Refer to MGT In on page 277.

MGT_TX_N / TX_N Writes data to the MGT module.

The port represents the differential signals of the GTH transceivers that are connected to the MGT module. The port must be connected to a function 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** I/O function lets you specify the reference frequency that is used to generate the MGT clock frequency. Refer to MGT In on page 277.

I/O mapping

The MGT communication bus uses the MPO connector of the FPGA Base Board.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

I/O Functions of the Inter-FPGA Interface Framework

Introduction

The *Inter-FPGA Interface* framework provides access to the I/O module slots of a SCALEXIO FPGA base board to implement an inter-FPGA communication bus.

Where to go from here

Information in this section

I-FPGA In....282
To provide read access to the inter-FPGA communication bus with bus synchronization.

I-FPGA Out....287
To provide write access to the inter-FPGA communication bus with bus synchronization.

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 SCALEXIO system:

I/O Functions of the DS2655 FPGA Base Board Framework....87
The frameworks of the DS2655 FPGA Base Boards provide the standard I/O functionality of the boards.

I/O Functions of the DS6601 FPGA Base Board Framework..... 123
The DS6601 (KU035) FPGA Base Board framework of the DS6601 FPGA Base Board provide the standard I/O functionality of the board.

I/O Functions of the DS6602 FPGA Base Board Framework..... 159
The DS6602 (KU15P) FPGA Base Board framework of the DS6602 FPGA Base Board provide the standard I/O functionality of the board.

I/O Functions of the DS2655M1 I/O Module Framework..... 205
The DS2655M1 I/O Module framework provides analog and digital I/O functionality of SCALEXIO FPGA base board with at least one DS2655M1 Multi-I/O Module.

I/O Functions of the DS2655M2 I/O Module Framework..... 217
The DS2655M2 I/O Module framework provides digital I/O functionality of a SCALEXIO FPGA base board with at least one DS2655M2 Digital I/O Module.

I/O Functions of the DS6651 Multi-I/O Module Framework..... 239
The DS6651 Multi-I/O Module framework provides analog and digital I/O functionality of a SCALEXIO FPGA base board with at least one DS6651 Multi-I/O Module.

I-FPGA In

Purpose

To provide read access to the inter-FPGA communication bus with bus synchronization.

Description

With the *Inter-FPGA Interface* framework, you can use I/O module slots of a SCALEXIO FPGA base board as inter-FPGA interfaces. The module number represents the used I/O module slot of the FPGA board.

The **I-FPGA In** I/O functions let you configure up to eight subbuses with bus synchronization.

A previously configured communication channel is no more available for other I-FPGA I/O functions, such as **I-FPGA Out**. In total, you can use up to eight input *and* output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

ioModuleNr Lets you set the used I/O module slot to configure the following I/O functions. Value range: 1 ... 5

hcfw.IOProperties.In.Fct(<ChannelNumber> + ioInOffset(ioModuleNr)).HcCustomName / Channel name Lets you specify a custom name for the specified channel. The channel number reflects one of eight configurable subbuses.

hcfw.IOProperties.In.Fct(<ChannelNumber> + ioInOffset(ioModuleNr)).Parameter(startparam).Init / Mode You have to specify only the channel number. The channel number reflects one of eight configurable subbuses. To use the I-FPGA In function, the mode must be set to 2 or 4. Mode 4 is the recommended mode. Mode 2 is an expert mode that you schould use 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.

Possible modes for the inter-FPGA interface:

- 0: Unused
- 1: Write access to the inter-FPGA interface with a lower latency (2 clock cycles), but the bits are not synchronous (expert mode).
- 2: Read access to the inter-FPGA interface with a lower latency (2 clock cycles), but the bits are not synchronous (expert mode).
- 3: Write access to the inter-FPGA interface *with* bus synchronization.
- 4: Read access to the inter-FPGA interface *with* bus synchronization.

hcfw.IOProperties.In.Fct(<ChannelNumber> + ioInOffset(ioModuleNr)).Parameter(startparam+2).Init / Startbit Lets you specify the bit with which the transmission data starts in the range 0 … 27. The channel number reflects one of eight configurable subbuses.

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 Handcode Interface Guide 🕮).

hcfw.IOProperties.In.Fct(<ChannelNumber> + ioInOffset(ioModuleNr)).Parameter(startparam+4).Init / Endbit Lets you specify the bit with which the transmission data ends in the range 0 … 27. The channel number reflects one of eight configurable subbuses.

The range of the end bit is to be adapted to the specified start bit. It is not allowed to specify an end bit less than the corresponding start bit.

For each subbus with bus synchronization, one bit is to be reserved for synchronization. The maximum data width of a subbus is therefore `Endbit - Startbit`.

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 Handcode Interface Guide 🕮).

hcfw.IOProperties.In.Fct(<ChannelNumber> + ioInOffset(ioModuleNr)).Parameter(startparam+7).Init / Bit length Lets you specify the bit length used for the transmission in the range 3 … 128 cycles. The parameter effects only synchronized buses. The channel number reflects one of eight configurable subbuses.

The default value is 6 cycles.

Usable only for inter-FPGA communication buses with bus synchronization.

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.

A reference value for the bit length depends on the specified filter depth and can be calculated by $2 + 2 \cdot \mathrm{FilterDepth}_{\mathrm{In}}$.

hcfw.IOProperties.In.Fct(<ChannelNumber> + ioInOffset(ioModuleNr)).Parameter(startparam+9).Init / Clock Lets you specify the clock frequency used for the inter-FPGA communication. The parameter effects only synchronized buses. The channel number reflects one of eight configurable subbuses.

Usable only for inter-FPGA communication buses with bus synchronization.

Possible values:

- 1: 125 MHz
- 2: 250 MHz

The default value is 1 (125 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.

hcfw.IOProperties.In.Fct(<ChannelNumber> + ioInOffset(ioModuleNr)).Parameter(startparam+11).Init / Filter depth Lets you specify a spike filter with the specified length to reduce transmission errors in the range 0 … 32 cycles. The parameter effects only synchronized buses. The channel number reflects one of eight configurable subbuses.

Usable only for inter-FPGA communication buses with bus synchronization.

The default value is 2 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.

hcfw.IOProperties.In.Fct(<ChannelNumber> + ioInOffset(ioModuleNr)).Parameter(startparam+12).Init / Add internal pipeline register to relax timing Lets you enable an additional internal pipeline to relax timing especially for 250 MHz communication if a FPGA build process were otherwise not possible. The parameter effects only synchronized buses. The channel number reflects one of eight configurable subbuses.

Usable only for inter-FPGA communication buses with bus synchronization.

- 0: Off (default)
- 1: On

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

m<ModuleNumber>_intercom_<ChannelNumber>_data_in / Data Reads data from the inter-FPGA communication bus. The module number reflects the used I/O module slot. The channel number reflects one of eight configurable subbuses.. Bits which exceed the configured bus width are discarded. For each configured subbus one bit is automatically reserved for synchronization.

Data width: 1

Value range: 0 ... 2^{27}-1

m<ModuleNumber>_intercom_<ChannelNumber>_data_new / Data New Indicates whether new data was written to the Data register. The module number reflects the used I/O module slot. The channel number reflects one of eight configurable subbuses.

If the Data register contains new values, the flag changes from 0 to 1 for one clock cycle. If the transmission failed, the error counter increases.

Usable only for inter-FPGA communication buses with bus synchronization.

Data width: 1

- 0: No new data available in the Data register. Either the transmission is not yet finished, or the transmission failed (see Errors outport).
- 1: New data available in the Data register.

m<ModuleNumber>_intercom_<ChannelNumber>_errors / Errors Outputs the number of transmission errors. The counter is reset only at FPGA application start. If the range exceeds, the counter restarts with 0.

The module number reflects the used I/O module slot. The channel number reflects one of eight configurable subbuses.

Usable only for inter-FPGA communication buses with bus synchronization.

Data width: 1

Value range: 0 ... 2^{32}-1

m<ModuleNumber>_intercom_<ChannelNumber>_reset_error / Errors Reset Resets the Errors output. The module number reflects the used I/O module slot. The channel number reflects one of eight configurable subbuses.

Usable only for inter-FPGA communication buses with bus synchronization.

The I-FPGA In Errors counter potentially increases until the FPGA Base Board with the corresponding I-FPGA Out interface starts working.

- 0: Not used
- 1: Error output is reset

Data type: UFix1_0

Value range: 0 ... 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.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

I-FPGA Out

Purpose

To provide write access to the inter-FPGA communication bus with bus synchronization.

Description

With the *Inter-FPGA Interface* framework, you can use I/O module slots of a SCALEXIO FPGA base board as inter-FPGA interfaces. The module number represents the used I/O module slot of the FPGA board.

The **I-FPGA Out** I/O functions let you configure up to eight subbuses with bus synchronization.

A previously configured communication channel is no more available for other I-FPGA I/O functions, such as **I-FPGA In**. In total, you can use up to eight input *and* output channels.

This I/O function is not considered when you generate the processor interface model.

Avoiding hardware damage

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 Handcode Interface Guide 🕮).

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

ioModuleNr Lets you set the used I/O module slot to configure the following I/O functions. Value range: 1 ... 5

hcfw.IOProperties.Out.Fct(<ChannelNumber> + ioOutOffset(ioModuleNr)).HcCustomName / Channel name Lets you specify a custom name for the specified channel. The channel number reflects one of eight configurable subbuses.

hcfw.IOProperties.Out.Fct(<ChannelNumber> + ioOutOffset(ioModuleNr)).Parameter(startparam).Init / Mode You have to specify only the channel number. The channel number reflects one of eight configurable subbuses. To use the I-FPGA Out function, the mode must be set to

1 or 3. Mode 3 is the recommended mode. Mode 1 is an expert mode that you schould use 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.

Possible modes for the inter-FPGA interface:

- 0: Unused
- 1: Write access to the inter-FPGA interface with a lower latency (2 clock cycles), but the bits are not synchronous (expert mode).
- 2: Read access to the inter-FPGA interface with a lower latency (2 clock cycles), but the bits are not synchronous (expert mode).
- 3: Write access to the inter-FPGA interface *with* bus synchronization.
- 4: Read access to the inter-FPGA interface *with* bus synchronization.

hcfw.IOProperties.Out.Fct(<ChannelNumber> + ioOutOffset(ioModuleNr)).Parameter(startparam+2).Init / Startbit Lets you specify the bit with which the transmission data starts in the range 0 … 27. The channel number reflects one of eight configurable subbuses.

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 Handcode Interface Guide 🕮).

hcfw.IOProperties.Out.Fct(<ChannelNumber> + ioOutOffset(ioModuleNr)).Parameter(startparam+4).Init / Endbit Lets you specify the bit with which the transmission data ends in the range 0 … 27. The channel number reflects one of eight configurable subbuses.

The range of the end bit is to be adapted to the specified start bit. It is not allowed to specify an end bit less than the corresponding start bit.

For each subbus with bus synchronization, one bit is to be reserved for synchronization. The maximum data width of a subbus is therefore `Endbit - Startbit`.

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 Handcode Interface Guide 🕮).

hcfw.IOProperties.Out.Fct(<ChannelNumber> + ioOutOffset(ioModuleNr)).Parameter(startparam+7).Init / Bit length Lets you specify the bit length used for the transmission in the range 3 … 128 cycles. The parameter effects only synchronized buses. The channel number reflects one of eight configurable subbuses.

The default value is 6 cycles.

Usable only for inter-FPGA communication buses with bus synchronization.

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.

A reference value for the bit length depends on the specified filter depth of the related I-FPGA In, and can be calculated by $2 + 2 \cdot \mathtt{FilterDepth}_{\mathtt{In}}$.

hcfw.IOProperties.Out.Fct(<ChannelNumber> + ioOutOffset(ioModuleNr)).Parameter(startparam+9).Init / Clock Lets you specify the clock frequency used for the inter-FPGA communication. The parameter effects only synchronized buses. The channel number reflects one of eight configurable subbuses.

Usable only for inter-FPGA communication buses with bus synchronization.

Possible values:

- 1: 125 MHz
- 2: 250 MHz

The default value is 1 (125 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.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

m<ModuleNumber>_intercom_<ChannelNumber>_data_out / Data Out Outputs the data to be written to the inter-FPGA communication bus. The channel number reflects one of eight configurable bus segments. The module number reflects the used I/O module slot. The channel number reflects one of eight configurable subbuses. Bits which exceed the configured bus width are discarded. For each configured subbus one bit is automatically reserved for synchronization.

Data width: 1

Value range: 0 ... 2^{27}-1

m<ModuleNumber>_intercom_<ChannelNumber>_data_sent_v / Data Sent Outputs the data already transmitted to the inter-FPGA communication bus. The module number reflects the used I/O module slot. The channel number reflects one of eight configurable subbuses.

Usable only for inter-FPGA communication buses with bus synchronization.

Data width: 1

Value range: 0 ... 2^{27}-1

m<ModuleNumber>_intercom_<ChannelNumber>_ready / Ready Signals the clock cycle in which the data to be transmitted is sampled. The port is available only for synchronized buses.

The port is high for one clock cycle with the periodicity of the bit length.

Usable only for inter-FPGA communication buses with bus synchronization.

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.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 🕮)

I/O Functions of the DS5203 with Onboard I/O Frameworks

Introduction

The onboard I/O frameworks of DS5203 (7K325 and 7K410) provide the standard I/O functionality of the DS5203 FPGA board.

Where to go from here

Information in this section

ADC....293
To read data from an analog input signal in the FPGA application.

APU Master....294
To write a value to the time-base connector for synchronizing angle-based applications.

APU Slave....295
To read a value from the time-base connector for synchronizing angle-based applications.

Buffer In....297
To read data from a PHS-bus buffer.

Buffer64 In....298
To read data from an internal bus buffer with a data width of 64 bits.

Buffer Out....300
To write data to a PHS-bus buffer.

Buffer64 Out....302
To write data to PHS-bus buffer with a data width of 64 bits.

DAC....304
To write data to an analog output signal in the FPGA application.

Digital In....305
To read data from a digital input signal in the FPGA application.

Digital Out....307
To write data to a digital output signal in the FPGA application.

I-FPGA Master....309
To write data to the inter-FPGA communication bus.

I-FPGA Slave....312
To receive data from the inter-FPGA communication bus.

Interrupt....315
To request a processor interrupt outside of the FPGA application.

LED Out....315
To write a digital signal that controls the LED on the board's bracket.

Register In....316
To read data from a PHS-bus register.

Register64 In....318
To read data from a PHS-bus register with a data width of 64 bits.

Register Out....320
To write data to a PHS-bus register.

Register64 Out....321
To write data to a PHS-bus register with a data width of 64 bits.

Status In....323
To read a digital signal that outputs the state of the FPGA initialization sequence.

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:

I/O Functions of the DS5203M1 Multi-I/O Module Frameworks....325
The frameworks of DS5203 with Multi-I/O Module (DS5203M1) provide the standard I/O functionality of the DS5203M1 Multi-I/O Module.The frameworks include access functions to digital and analog input and output signals, and to PHS-bus buffers and registers of the DS5203 FPGA Board.

ADC

Purpose

To read data from an analog input signal in the FPGA application.

Description

According to the number of physical connections available on the DS5203 FPGA Board, you can select the ADC I/O functions. There are six analog input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 17 … 22.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Input range Lets you select the input voltage range for the analog input channel.

- 0: –5 V … +5 V
- 1: –30 V … +30 V

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / 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 of the specified input range.

- 0: Bit
- 1: mV

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 5.

adc<ChannelNumber>_data / Data Outputs the current results of the A/D conversions on the current channel.

Range:

- –5000 mV … +5000 mV
- –30000 mV … +30000 mV

  or

- –8192 … +8191

Update rate: 10 Msps

adc<ChannelNumber>_valid / Valid Represents the current status of the data output.

- 0: Converted value is out of range.
- 1: Converted value is within the specified input range.

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

Related topics

References

Overview of the DS5203 Onboard I/O Frameworks 32

APU Master

Purpose

To write a value to the time-base connector for synchronizing angle-based applications.

Description

The DS5203 frameworks provide one digital output channel for the APU Master I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is to be specified with 24.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

There is no channel number to be specified.

apu_phi_wr / Phi Write Specifies the value to be written to the time-base connector. Other I/O boards in the hardware system configured as APU slave receive the value and synchronize their timebases with the given angle value.

Value range: 0 … 65535 (0 … 720°)

Time-base clock cycle: 250 ns

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 🕮).

Related topics

References

APU Slave 295
Overview of the DS5203 Onboard I/O Frameworks 32

APU Slave

Purpose

To read a value from the time-base connector for synchronizing angle-based applications.

Description

The DS5203 frameworks provide one digital input channel for the APU Slave I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is to be specified with 24.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

There is no channel number to be specified.

apu_phi_rd / Phi Read Outputs the value that has been sent by an I/O board in the hardware system specified as APU master.

Value range: 0 … 65535 (0 … 720°)

Time-base clock cycle: 250 ns

apu_phi_nd / Phi New Indicates that new data was written to the Phi_Read register.

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 🕮).

Related topics

References

APU Master 294
Overview of the DS5203 Onboard I/O Frameworks 32

Buffer In

Purpose

To read data from a PHS-bus buffer.

Description

If you select Buffer as the access type, the data is read from a PHS-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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. 0 represents the lowest bit position, 32 the highest bit position.
- floating-point

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

  The fraction width is displayed.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you select the data format of the Data outport.

- Fixed-point format

  UFix_32_<Binary point position> or Fix_32_<Binary point position>
- Floating-point format

  XFloat_8_24

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf_<ChannelNumber>_count / 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).

xmemf_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from a PHS-bus buffer. The data format depends on the related parameter settings.

xmemf_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer Out 300
Buffer64 In 298
Overview of the DS5203 Onboard I/O Frameworks 32
Register In 316

Buffer64 In

Purpose

To read data from a PHS-bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is read from a PHS-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 ... 320.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunction_Number>).Parameter(3).Init / 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.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 63.

xmem64f_<ChannelNumber>_count / 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).

xmem64f_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from a PHS-bus buffer. The data format depends on the related parameter settings.

xmem64f_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmem64f_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer In 297
Buffer64 Out 302
Overview of the DS5203 Onboard I/O Frameworks 32
Register64 In 318

Buffer Out

Purpose

To write data to a PHS-bus buffer.

Description

If you select Buffer as the access type, the data is written to a PHS-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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. 0 represents the lowest bit position, 32 the highest bit position.
- floating-point

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

  The fraction width is displayed.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you select the data format of the Data inport.

- Fixed-point format

  UFix_32_<Binary point position> or Fix_32_<Binary point position>
- Floating-point format

  XFloat_8_24

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to a PHS-bus buffer. The data format depends on the related parameter settings.

xmemp_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp_<ChannelNumber>_finished / 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.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via PHS bus in the next clock cycle.

The ready flag must be set no later than the last data value. Otherwise the buffer switches twice.

xmemp_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer In 297
Buffer64 Out 302
Overview of the DS5203 Onboard I/O Frameworks 32
Register Out 320

Buffer64 Out

Purpose

To write data to a PHS-bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is written to a PHS-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 ... 320.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmem64p_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an internal bus buffer. The data format depends on the related parameter settings.

xmem64p_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmem64p_<ChannelNumber>_finished / 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.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via the internal bus in the next clock cycle.

The ready flag must be set no later than the last data value. Otherwise the buffer switches twice.

xmem64p_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer Out 300
Buffer64 In 298
Overview of the DS5203 Onboard I/O Frameworks 32
Register64 Out 321

DAC

Purpose

To write data to an analog output signal in the FPGA application.

Description

According to the number of physical connections available on the DS5203 FPGA Board, you can select the DAC I/O functions. There are six analog output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 17 … 22.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / 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).

- 0: Bit
- 1: mV

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 5.

dac<Channel_Number>_data / Data Outputs a signal in the specified range.

Output voltage range:

- -10000 mV … +10000 mV

  or

- -8192 … +8191

Range exceeding is possible and will be saturated to the 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).

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

Related topics

References

ADC 293
General Information on the I/O Functions Available with FPGA Frameworks 13

Digital In

Purpose

To read data from a digital input signal in the FPGA application.

Description

According to the number of physical connections available on the DS5203 FPGA Board, you can select the Digital In I/O functions. There are 16 digital input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 16.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

- 1000: 1000 mV threshold level
- …
- 7500: 7500 mV threshold level

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.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

digio_<ChannelNumber>_in / Data Outputs the current results of digital input channel.

- 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

Note

Asynchronous input data might lead to metastable register states. Further synchronization techniques might be necessary.

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.

Related topics

References

Digital Out 307
Overview of the DS5203 Onboard I/O Frameworks 32

Digital Out

Purpose

To write data to a digital output signal in the FPGA application.

Description

According to the number of physical connections available on the DS5203 FPGA Board, you can select the Digital Out I/O functions. There are 16 digital output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 16.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(1).Parameter(1).Init / High supply Lets you select the voltage for the high side switch for all digital output channels.

- 0: 5 V
- 1: 3.3 V

Note

You can specify the high supply voltage value only globally for all digital output channels.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

digio_<ChannelNumber>_out / 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).

Update rate: 100 MHz

digio_<Channel_Number>_oe / Enable Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z.

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.

Related topics

References

Digital In 305
Overview of the DS5203 Onboard I/O Frameworks 32

I-FPGA Master

Purpose

To write data to the inter-FPGA communication bus.

Description

If you have connected two DS5203 FPGA boards via their inter-FPGA communication connectors, you can use the **I-FPGA Master** I/O functions to configure up to eight subbuses.

A previously configured communication channel is no more available for the **I-FPGA Slave** function. In total, you can use up to eight channels for masters *and* slaves.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 25 … 32.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Mode Lets you select whether to use the function as I-FPGA Master to transmit data to the inter-FPGA communication bus or I-FPGA Slave to receive data from it.

- 0: Unused
- 3: I-FPGA Master
- 4: I-FPGA Slave

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Startbit Lets you specify the bit with which the transmission data starts in the range 0 … 31.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(5).Init / Endbit Lets you specify the bit with which the transmission data ends in the range 0 … 31.

The range of the end bit is to be adapted to the specified start bit. It is not allowed to specify an end bit less than the corresponding start bit.

For each configured subbus, one bit is to be reserved for synchronization. The maximum data width of a subbus is therefore `Endbit - Startbit`.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(8).Init / Bit length Lets you specify the bit length used for the transmission in the range 3 … 128 cycles.

The default value is 6 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.

A reference value for the bit length depends on the specified filter depth of the related I-FPGA Slave, and can be calculated by $2 + 2 \cdot \mathrm{FilterDepth}_{\mathrm{Slave}}$.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(10).Init / Clock Lets you specify the clock frequency used for the inter-FPGA communication.

- 1: 100 MHz
- 2: 200 MHz
- 3: 300 MHz

The default value is 1 (100 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.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 01 … 08.

intercom_<ChannelNumber>_data_out / Data Write Outputs the data to be written to the inter-FPGA communication bus. The channel number reflects one of eight configurable bus segments.
Data width: 0 … 31 bit

intercom_<ChannelNumber>_data_sent_v / Data Sent Outputs the data already transmitted to the inter-FPGA communication bus. The channel number reflects one of eight configurable bus segments.
Data width: 0 … 31 bit

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 🕮).

Related topics

References

I-FPGA Slave 312
Overview of the DS5203 Onboard I/O Frameworks 32

I-FPGA Slave

Purpose

To receive data from the inter-FPGA communication bus.

Description

If you have connected two DS5203 FPGA boards via their inter-FPGA communication connectors, you can use the I-FPGA Slave I/O functions to configure up to eight subbuses.

A previously configured communication channel is no more available for the I-FPGA Master function. In total, you can use up to eight channels for masters *and* slaves.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 25 … 32.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Mode Lets you select whether to use the function as I-FPGA Master to transmit data to the inter-FPGA communication bus or I-FPGA Slave to receive data from it.

- 0: Unused
- 3: I-FPGA Master
- 4: I-FPGA Slave

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / Startbit Lets you specify the bit with which the transmission data starts in the range 0 … 31.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(5).Init / Endbit Lets you specify the bit with which the transmission data ends in the range 0 … 31.

The range of the end bit is to be adapted to the specified start bit. It is not allowed to specify an end bit less than the corresponding start bit.

For each configured subbus, one bit is to be reserved for synchronization. The maximum data width of a subbus is therefore `Endbit - Startbit`.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(8).Init / Bit length Lets you specify the bit length used for the transmission in the range 3 … 128 cycles.

The default value is 6 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.

A reference value for the bit length depends on the specified filter depth of the related I-FPGA Slave, and can be calculated by $2 + 2 \cdot \mathtt{FilterDepth}_{\mathtt{Slave}}$.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(10).Init / Clock Lets you specify the clock frequency used for the inter-FPGA communication.

- 1: 100 MHz
- 2: 200 MHz
- 3: 300 MHz

The default value is 1 (100 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.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(12).Init / Filter depth Lets you specify a spike filter with the specified length to reduce transmission errors in the range 0 … 32 cycles.

The default value is 2 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.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(13).Init / Add internal pipeline register to relax timing Lets you enable an additional internal pipeline to relax timing especially for 300 MHz communication if a FPGA build process were otherwise not possible.

- 0: Off (default)
- 1: On

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 01 … 08.

intercom_<ChannelNumber>_data_in / Data Read Reads data from the inter-FPGA communication bus. The channel number reflects one of eight configurable bus segments. Bits which exceed the configured bus width are discarded. For each configured subbus one bit is automatically reserved for synchronization.

Data width: 0 … 31 bit

intercom_<ChannelNumber>_data_new / Data New Indicates whether new data was written to the Data Read register.

If the Data Read register contains new values, the flag changes from 0 to 1 for one clock cycle. If the transmission failed, the error counter increases.

Data width: 1 bit

- 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.

intercom_<ChannelNumber>_errors / Errors Outputs the number of transmission errors. The counter is reset only at FPGA application start. If the range is exceeded, the counter restarts with 0.

Data width: 0 … 31 bit

intercom_<ChannelNumber>_reset_error / Errors Reset Resets the Errors output.

The I-FPGA Slaves Errors counter potentially increases until the FPGA Board with the corresponding I-FPGA Master starts working.

- 0: Not used
- 1: Error output is reset

Data type: UFix1_0

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 🕮).

Related topics

References

I-FPGA Master 309
Overview of the DS5203 Onboard I/O Frameworks 32

Interrupt

Purpose

To request a processor interrupt outside of the FPGA application.

Description

The DS5203 with onboard I/O 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_INT_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 8.

IRQProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 7.

usr_<ChannelNumber>_interrupt / Int Provides the interrupt request line.

- 0 to 1: Interrupt is requested (edge-triggered).
- 0: No interrupt is requested. Last requested interrupt is saved.

Related topics

References

Overview of the DS5203 Onboard I/O Frameworks 32

LED Out

Purpose

To write a digital signal that controls the LED on the board's bracket.

Description

The DS5203 frameworks provide one digital output channel for the LED Out I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is to be specified with 23.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

There is no channel number to be specified.

led_out / Data Controls the LED on the board's bracket.

- 0: LED lights green.
- 1: LED lights orange.

Related topics

References

Overview of the DS5203 Onboard I/O Frameworks 32

Register In

Purpose

To read data from a PHS-bus register.

Description

If you select Register as the access type, the data is read from a PHS-bus register. 128 registers are available with a data width of 32 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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. 0 represents the lowest bit position, 32 the highest bit position.
- floating-point

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

  The fraction width is displayed.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you select the data format of the Data outport.

- Fixed-point format

  UFix_32_<Binary point position> or Fix_32_<Binary point position>
- Floating-point format

  XFloat_8_24

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the PHS bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from a PHS-bus register. The data format depends on the related parameter settings.

xreg_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer In 297
Overview of the DS5203 Onboard I/O Frameworks 32
Register Out 320
Register64 Out 321

Register64 In

Purpose

To read data from a PHS-bus register with a data width of 64 bits.

Description

If you select **Register64** as the access type, the data is read from a PHS-bus register. 128 registers are available 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 ... 288.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the **Data** outport depending on the format selected in the **Format** setting (see below).

- signed/unsigned

  The values of the **Data** outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the **Data** outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunction_Number>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the PHS bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from a PHS-bus register. The data format depends on the related parameter settings.

xreg64_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer Out 300
Overview of the DS5203 Onboard I/O Frameworks 32
Register In 316
Register64 Out 321

Register Out

Purpose

To write data to a PHS-bus register.

Description

If you select Register as the access type, the data is written to a PHS-bus register. 128 registers are available with a data width of 32 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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. 0 represents the lowest bit position, 32 the highest bit position.
- floating-point

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

  The fraction width is displayed.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you select the data format of the Data inport.

- Fixed-point format

  UFix_32_<Binary point position> or Fix_32_<Binary point position>
- Floating-point format

  XFloat_8_24

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 PHS bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to a PHS-bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer Out 300
Overview of the DS5203 Onboard I/O Frameworks 32
Register In 316
Register64 Out 321

Register64 Out

Purpose

To write data to a PHS-bus register with a data width of 64 bits.

Description

If you select **Register64** as the access type, the data is written to a PHS-bus register. 128 registers are available 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 ... 288.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Register64 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 PHS bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 128.

xreg64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an internal bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer64 Out 302
Overview of the DS5203 Onboard I/O Frameworks 32
Register Out 320
Register64 In 318

Status In

Purpose

To read a digital signal that outputs the state of the FPGA initialization sequence.

Description

The DS5203 frameworks provide one digital input channel for the Status In I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number is to be specified as 23.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

There is no channel number to be specified.

init_done/ Init Done Outputs the state of the initialization sequence that is started after programming the FPGA.

- 0: Initialization sequence is in progress.
- 1: Initialization sequence has finished.

Related topics

References

Overview of the DS5203 Onboard I/O Frameworks 32

I/O Functions of the DS5203M1 Multi-I/O Module Frameworks

Introduction

The frameworks of DS5203 (7K325 and 7K410) with Multi-I/O Module (DS5203M1) provide the standard I/O functionality of the DS5203M1 Multi-I/O Module.

Where to go from here

Information in this section

ADC (M1)....326
To read data from an analog input signal in the FPGA application.

DAC (M1)....327
To write data to an analog output signal in the FPGA application.

Digital In (M1)....329
To read data from a digital input signal in the FPGA application.

Digital Out (M1)....331
To write data to a digital output signal in the FPGA application.

Sensor Supply....333
To provide a supply voltage at a connected sensor.

Information in other sections

Other frameworks that provide access to the FPGA functionality of a PHS-bus-based system:

I/O Functions of the DS5203 with Onboard I/O Frameworks....291
The onboard I/O frameworks of DS5203 provide the standard I/O functionality of the DS5203 FPGA board. The frameworks include access functions to digital and analog input and output signals, and to PHS-bus buffers and registers.

ADC (M1)

Purpose

To read data from an analog input signal in the FPGA application.

Description

According to the number of physical connections available on the DS5203M1 Multi-I/O Module, you can select the ADC (M1) I/O functions. There are six analog input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 40 … 45.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Input range Lets you select the input voltage range for the analog input channel.

- 0: –5 V … +5 V
- 1: –30 V … +30 V

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / 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 of the specified input range.

- 0: Bit
- 1: mV

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 5.

m1_adc<ChannelNumber>_data / Data Outputs the current results of the A/D conversions on the current channel.

Range:

- –5000 mV … +5000 mV
- –30000 mV … +30000 mV

  or

- –8192 … +8191

Update rate: 10 Msps

m1_adc<ChannelNumber>_valid / Valid Represents the current status of the data output.

- 0: Converted value is out of range.
- 1: Converted value is within the specified input range.

I/O mapping

The following I/O mapping is relevant if you use one of the DS5203 with DS5203M1 Multi-I/O Module 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

Related topics

References

ADC 293
DAC (M1) 327
Overview of the DS5203M1 Multi-I/O Module Frameworks of DS5203 34

DAC (M1)

Purpose

To write data to an analog output signal in the FPGA application.

Description

According to the number of physical connections available on the DS5203M1 Multi-I/O Module, you can select the DAC (M1) I/O functions. There are six analog output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 40 … 45.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / 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).

- 0: Bit
- 1: mV

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 5.

m1_dac<ChannelNumber>_data / Data Outputs a signal in the specified range.

Output voltage range:

- -10000 mV … +10000 mV

  or

- -8192 … +8191

Range exceeding is possible and will be saturated to the 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).

I/O mapping

The following I/O mapping is relevant if you use one of the DS5203M1 Multi-I/O Module frameworks 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

Related topics

References

ADC (M1) 326
DAC 304
Overview of the DS5203M1 Multi-I/O Module Frameworks of DS5203 34

Digital In (M1)

Purpose

To read data from a digital input signal in the FPGA application.

Description

According to the number of physical connections available on the DS5203M1 Multi-I/O Module, you can select the **Digital In (M1)** I/O functions. There are 16 digital input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 24 … 39.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

- 1000: 1000 mV threshold level
- …
- 7500: 7500 mV threshold level

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.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

m1_digio_<ChannelNumber>_in / Data Outputs the current results of digital input channel.

- 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

Note

Asynchronous input data might lead to metastable register states. Further synchronization techniques might be necessary.

I/O mapping

The following I/O mapping is relevant if you use one of the DS5203M1 Multi-I/O Module 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.

Related topics

References

Digital In 305
Digital Out (M1) 331
Overview of the DS5203M1 Multi-I/O Module Frameworks of DS5203 34

Digital Out (M1)

Purpose

To write data to a digital output signal in the FPGA application.

Description

According to the number of physical connections available on the DS5203M1 Multi-I/O Module, you can select the **Digital Out (M1)** I/O functions. There are 16 digital output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 24 … 39.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(24).Parameter(1).Init / High supply Lets you select the voltage for the high side switch for all digital output channels.

- 0: 5 V
- 1: 3.3 V

Note

You can specify the high supply voltage value only globally for all digital output channels.
The I/O function number is 24.

Port

You must add the following signals of the I/O function to the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

m1_digio_<ChannelNumber>_out / 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).

Update rate: 100 MHz

m1_digio_<ChannelNumber>_oe / Enable Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z.

I/O mapping

The following I/O mapping is relevant if you use one of the DS5203M1 Multi-I/O Module frameworks 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.

Related topics

References

Digital In (M1)..... 329
Digital Out..... 307
Overview of the DS5203M1 Multi-I/O Module Frameworks of DS5203..... 34

Sensor Supply

Purpose

To provide a supply voltage, for example, for a connected sensor, in the range 2000 mV … 20000 mV in steps of 100 mV.

Description

The DS5203M1 frameworks provide one output channel for the Sensor Supply (M1) I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 46.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Supply voltage Lets you specify the supply voltage a connected sensor is to be driven with in the range 2000 mV … 20000 mV in steps of 100 mV.

Port

There is no port to be specified.

I/O mapping

The following I/O mapping is relevant if you use one of the DS5203M1 Multi-I/O Module frameworks for sensor supply channels.

Outport	Channel	Connector Pin	Signal
–	1	P2 5 P2 38	VSENS- VSENS+

Related topics

References

Overview of the DS5203M1 Multi-I/O Module Frameworks of DS5203 34

I/O Functions of the FPGA1401Tp1 with Multi-I/O Module Frameworks

Introduction

The FPGA1401Tp1 frameworks provide the standard I/O functionality of MicroAutoBox II with DS1552 (*FPGA1401Tp1_DS1552_XC7K325T*) or with DS1552B1 Multi-I/O Module (*FPGA1401Tp1_DS1552B1_XC7K325T*).

Where to go from here

Information in this section

ADC (Type A)..... 337
To read data from an analog input signal in the FPGA application using the Type A conversion function.

ADC (Type B)..... 338
To read data from an analog input signal in the FPGA application using the Type B conversion function.

Buffer In..... 340
To read data from an intermodule-bus buffer with a data width of 32 bits.

Buffer64 In..... 341
To read data from an intermodule-bus buffer with a data width of 64 bits.

Buffer Out..... 343
To write data to an intermodule-bus buffer with a data width of 32 bits.

Buffer64 Out..... 345
To write data to an intermodule-bus buffer with a data width of 64 bits.

DAC..... 347
To write data to an analog output signal in the FPGA application.

Digital Crank/Cam Sensor..... 348
To provide bit-wise read access to digital camshaft and crankshaft sensors.

Digital In (Type A)..... 350
To read data from a digital input signal in the FPGA application using a digital input channel.

Digital In (Type B)..... 352
To read data from a digital input signal in the FPGA application using a digital bidirectional channel.

Digital Out (Type A)..... 354
To write data to a digital output signal in the FPGA application using a digital output channel.

Digital Out (Type B)..... 356
To write data to a digital output signal in the FPGA application using a digital bidirectional channel.

Inductive Zero Voltage Detector..... 358
To provide read access to an inductive zero voltage detector.

Interrupt..... 359
To request a processor interrupt outside of the FPGA application.

LED Out..... 360
To write a digital signal that controls the FPGA status LED near the DS1514 ZIF I/O connector.

Register In..... 361
To read data from an intermodule-bus register with a data width of 32 bits.

Register64 In..... 362
To read data from an intermodule-bus register with a data width of 64 bits.

Register Out..... 364
To write data to an intermodule-bus register with a data width of 32 bits.

Register64 Out..... 366
To write data to an intermodule-bus register with a data width of 64 bits.

Sensor Supply..... 368
To provide a supply voltage at a connected sensor.

Status In..... 368
To read digital signals that outputs state information, e.g.: state of the FPGA initialization sequence or the FPGA die temperature.

UART (RS232)..... 370
To implement communication via serial interface for RS232 UART type.

UART (RS422/485)..... 375
To implement communication via serial interface for RS422/485 UART type.

ADC (Type A)

Purpose

To read data from an analog input signal in the FPGA application using the Type A conversion function.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the ADC (Type A) I/O functions. There are eight analog input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 26 … 33.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 7.

hq_adc_<ChannelNumber>_data / Data Outputs the current results of the A/D conversions on the current channel.

Range: 0 … +65535

Update rate: 1 Msps

hq_adc_<ChannelNumber>_soc / Start of conversion Lets you trigger the start of an A/D conversion on the specified channel. 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.

hq_adc_<ChannelNumber>_eoc / End of conversion Outputs an end of conversion signal if the conversion result is available on the specified channel. 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.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for analog input channels using the Type A conversion function. 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.

Related topics

References

ADC (Type B) 338
DAC 347
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37

ADC (Type B)

Purpose

To read data from an analog input signal in the FPGA application using the Type B conversion function.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **ADC (Type B)** I/O functions. There are 16 analog output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 34 … 49.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

Iq_adc_<ChannelNumber>_data / Data Outputs the current results of the A/D conversions on the current channel in the range -32768 … +32767.

Update rate: 0.2 Msps

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for analog input channels using the Type B conversion function. 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

Related topics

References

ADC (Type A)..... 337
DAC..... 347

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Buffer In

Purpose

To read data from an intermodule-bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is read from an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

The I/O function number can be specified in the range 129 … 160.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf_<ChannelNumber>_count / 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).

xmemf_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemf_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer Out 343
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register In 361

Buffer64 In

Purpose

To read data from an intermodule-bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is read from an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

The I/O function number can be specified in the range 289 … 320.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf64_<ChannelNumber>_count / 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).

xmemf64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemf64_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf64_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer64 Out 345
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register64 In 362

Buffer Out

Purpose

To write data to an intermodule-bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is written to an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemp_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer In 340
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register Out 364

Buffer64 Out

Purpose

To write data to an intermodule-bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is written to an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 ... 320.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemp64_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp64_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp64_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer64 In..... 341
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37
Register64 Out..... 366

DAC

Purpose

To write data to an analog output signal in the FPGA application.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the DAC I/O functions. There are four analog output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 26 … 29.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 3.

dac<ChannelNumber>_data / Data Outputs a signal in the range 0 … +65535.

Range exceeding is possible and will be saturated to the 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).

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for analog output channels. 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

Related topics

References

ADC (Type A)..... 337
ADC (Type B)..... 338
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Digital Crank/Cam Sensor

Purpose

To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the Digital Crank/Cam Sensor I/O functions. There are three input channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 50 … 52.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Low threshold voltage To 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.

- Range:
  -40000 mV … +40000 mV
- Resolution:
  20 mV
- Default:
  1000 mV

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / High threshold voltage To 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.

- Range:
  -40000 mV … +40000 mV
- Resolution:
  100 mV
- Default:
  1000 mV

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 2.

cam_<ChannelNumber> / Data 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: 80 MHz

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for analog input channels. Depending on the MicroAutoBox variant 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

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Digital In (Type A)

Purpose

To read data from a digital input signal in the FPGA application using a digital input channel.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **Digital In (Type A)** I/O functions. There are 16 digital input channels.

The threshold level is fix:

- 3.6 V for low-high transition
- 1.2 V for high-low transition

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.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 17.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

dig_<ChannelNumber>_in / Data Outputs the current results of digital input channel.

- 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: 80 MHz

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 🕮).
- Asynchronous input data might lead to metastable register states. Further synchronization techniques might be necessary.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for digital input channels. 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

Related topics

References

Digital In (Type B)..... 352
Digital Out (Type A)..... 354
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Digital In (Type B)

Purpose

To read data from a digital input signal in the FPGA application by using a digital bidirectional channel.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **Digital In (Type B)** I/O functions. There are eight digital input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 18 … 25.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

- 1000: 1000 mV threshold level
- …
- 7500: 7500 mV threshold level

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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 07.

bidir_<ChannelNumber>_in / Data Outputs the current results of digital input channel.

- 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: 80 MHz

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 🕮).
- Asynchronous input data might lead to metastable register states. Further synchronization techniques might be necessary.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for digital bidirectional channels. 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.

Related topics

References

Digital In (Type A) 350
Digital Out (Type B) 356
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37

Digital Out (Type A)

Purpose

To write data to a digital output signal in the FPGA application using a digital output channel.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **Digital Out (Type A)** I/O functions. There are 16 digital output channels.

The voltage range for the high side switch for all digital output channels is in the range 0 V … 45 V.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 17.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

dig_<ChannelNumber>_out / 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).

Update rate: 80 MHz

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 🕮).

dig_<ChannelNumber>_oe / Enable Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for digital output channels. 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

Related topics

References

Digital In (Type A)..... 350
Digital Out (Type B)..... 356
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Digital Out (Type B)

Purpose

To write data to a digital output signal in the FPGA application by using a digital bidirectional channel.

Description

According to the number of physical connections available on the DS1514 Multi-I/O Module, you can select the **Digital Out (Type B)** I/O functions. There are eight digital output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 18 … 25.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(18).Parameter(1).Init / High supply Lets you select the voltage for the high side switch for all digital output channels.

- 0: 3.3 V
- 1: 5 V

Note

You can specify the high supply voltage value only globally for all digital output channels.
The I/O function number must be specified as 18.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 07.

bidir_<ChannelNumber>_out / 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).

Update rate: 80 MHz

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 🕮).

bidir_<ChannelNumber>_oe / Enable Controls the hardware output. If set to 1, the hardware output reacts to the **Data** outport, otherwise it is set to High-Z.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for digital bidirectional channels. 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.

Related topics

References

Digital In (Type B) 352
Digital Out (Type A) 354
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37

Inductive Zero Voltage Detector

Purpose

To provide read access to an inductive zero voltage detector.

Description

The FPGA1401Tp1 frameworks provide one channel for the Inductive Zero Voltage Detector I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 53.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

crank / Data To detect 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: 80 MHz

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for analog input channels. The signals are available at the DS1514 ZIF I/O connector.

Outport	Connector Pin	Signal
Data	P3	ZeroDetection+
	P4	ZeroDetection–

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Interrupt

Purpose

To request a processor interrupt outside of the FPGA application.

Description

The FPGA1401Tp1 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_INT_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 8.

IRQProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 7.

usr_<ChannelNumber>_interrupt / Int Provides the interrupt request line.

- 0 to 1: Interrupt is requested (edge-triggered).
- 0: No interrupt is requested. Last requested interrupt is saved.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

LED Out

Purpose

To write a digital signal that controls the FPGA status LED.

You can find the FPGA status LED near the DS1514 ZIF I/O connector.

Description

The FPGA1401Tp1 frameworks provide one channel for the LED Out I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

led_out / Data Controls the Status LED on the board's bracket.

- 0: LED lights green.
- 1: LED lights orange.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Register In

Purpose

To read data from an intermodule-bus register with a data width of 32 bits.

Description

If you select Register 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. 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- 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 fraction width is provided by the **Binary point position (or fraction width)** setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the intermodule bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an intermodule-bus register. The data format depends on the related parameter settings.

xreg_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer In 340
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register Out 364

Register64 In

Purpose

To read data from an intermodule-bus register with a data width of 64 bits.

Description

If you select Register64 as the access type, the data is read from an intermodule-bus register. 128 registers are available 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 … 288.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the intermodule bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an intermodule-bus register. The data format depends on the related parameter settings.

xreg64_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer64 In 341
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register64 Out 366

Register Out

Purpose

To write data to an intermodule-bus register with a data width of 32 bits.

Description

If you select **Register** 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. 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an intermodule-bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer Out 343
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register In 361

Register64 Out

Purpose

To write data to an intermodule-bus register with a data width of 64 bits.

Description

If you select **Register64** as the access type, the data is written to an intermodule-bus register. 128 registers are available 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 … 288.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the **Data** inport depending on the format selected in the **Format** setting (see below).

- signed/unsigned

  The values of the **Data** inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an intermodule-bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer64 Out 345
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register64 In 362

Sensor Supply

Purpose

To provide a supply voltage, for example, for a connected sensor, in the range 2000 mV … 20000 mV in steps of 100 mV.

Description

The FPGA1401Tp1 frameworks provide one channel for the Sensor Supply I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 30.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Supply voltage Lets you specify the supply voltage a connected sensor is to be driven with in the range 2000 mV … 20000 mV in steps of 100 mV.

Port

There is no port to be specified.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for for sensor supply. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Sim_Data	1	b6 c6	VS- VS+

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Status In

Purpose

To read digital signals that outputs state information, e.g.: state of the FPGA initialization sequence or the FPGA die temperature.

Description

The FPGA1401Tp1 framework provides one channel for the Status In I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

init_done/ Init Done Outputs the state of the initialization sequence that is started after programming the FPGA.

- 0: Initialization sequence is in progress.
- 1: Initialization sequence has finished.

temperature / Temperature Outputs the raw value of the FPGA's die temperature measurement. Use the 12 MSB bits to calculate the die temperature.

Equation to calculate 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_temp / 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.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

UART (RS232)

Purpose

To implement communication via serial interface for RS232 UART type.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **UART (RS232)** I/O functions. There are two interfaces.

Note

UART 1 can be used without modification. To use UART 2, your DS1552 has to be modified by dSPACE.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 31 … 32.

Most of the parameters are used for the UART (RS232) and UART (RS422/485) I/O functions.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Baud rate Lets you specify the baud rate of the UART in the range 50 … 1,000,000 baud (bits per second).

The baud rate depends on the parameters 2, 3, 4 and the FPGA board, and can be calculated by the following formula:

```
BaudRate = (10^8 · uart_x_dcm_m) / (2 · uart_x_dcm_d
· (uart_dcm_clk_divider+1))
```

With:

Variable	Parameter	Description
uart_x_dcm_m[1)]	Parameter(2).Init	Multiplier for the digital clock manager (DCM) module in the range 2 … 255.
uart_x_dcm_d[1)]	Parameter(3).Init	Divisor for the digital clock manager (DCM) module in the range 1 … 255.
uart_x_dcm_clk_divider[1)]	Parameter(4).Init	UART clock divider in the range 0 … 262,143.

[1)] x=1 for UART 1; x=2 for UART 2

Note

Limitations:

- The maximum baud rate of 1,000,000 baud must not be exceeded.
- The output frequency of the digital clock manager (DCM) module should be between 40 MHz and 160 MHz:
  f_{DCM} = 200 MHz · uart_x_dcm_m / uart_x_dcm_d

Tip

In the framework folder you find a MATLAB file that provides some calculated baud rates and the percentage deviations to the supported baud rates according to the parameters m, d and the clock divider.

- Framework folder:
  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA1401Tp1_<Multi_I/O_Module_type>_<FPGA_type>`
- MATLAB file name: `FPGA1401Tp1_XC7K325T_uart_parameters.mat`

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(5).Init / Word length Lets you specify the word length in the range 5 … 9 bit. The word length includes the number of data bits and the optional parity bit. Exceeding bits in a message are ignored at the transmitter or cleared at the receiver.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(6).Init / Stop bits Lets you specify the length of the stop bits in half of bits.

Stop Bits	Parameter Value
1	2
1.5	3
2	4

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(7).Init / UART type Lets you specify the UART type.

Value	UART Type
0	RS232
1	RS422/485

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(8).Init / Termination Lets you specify the termination state.

Note

For the RS232 UART type, the termination must be set to 0 (disconnected).

Value	Termination State	Description
0	Disconnected	▪ The RX/CTS and TX/RTS signals are not terminated.
1	Connected	▪ Not allowed

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 1.

uart_<ChannelNumber>_rd / 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 FPGA clock cycle from the UART.

uart_<ChannelNumber>_rd_data_count / 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.

Value range: 0 … 2047

uart_<ChannelNumber>_rd_fifo_empty / 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.

Range:

- 0: The RX FIFO buffer is not empty.
- 1: The RX FIFO buffer is empty.

uart_<ChannelNumber>_rd_data / 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.

Range: 0 … 511

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).

uart_<ChannelNumber>_wr / 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.

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 baudrate. The hardware port has inverted voltage levels of -6 V (logical high) and +6 V (logical low).

uart_<ChannelNumber>_wr_data_count / 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.

Range: 0 … 2047

uart_<ChannelNumber>_wr_fifo_full / 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.

Range:

- 0: The TX FIFO buffer is not full.
- 1: The TX FIFO buffer is full.

uart_<ChannelNumber>_wr_data / Write Data Specifies the value to be send.

The **Write_Data** signal is transferred at each clock cycle with **Write_Enable** set to 1.

Range: 0 … 511

uart_<ChannelNumber>_rts / 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.

The hardware port is synchronously running to the UART clock defined by the UART baudrate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive).

uart_<ChannelNumber>_cts / 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.

Range:

- 0: CTS inactive
- 1: CTS active

The CTS hardware port is synchronously running to the UART clock defined by the UART baudrate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive).

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for serial communication using the UART (RS232) function. 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.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37
UART (RS422/485)..... 375

UART (RS422/485)

Purpose

To implement communication via serial interface for RS422/485 UART type.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **UART (RS422/485)** I/O functions. There are two interfaces.

Note

UART 1 can be used without modification. To use UART 2, your DS1552 has to be modified by dSPACE.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 33 … 34.

Most of the parameters are used for the UART (RS232) and UART (RS422/485) I/O functions.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Baud rate Lets you specify the baud rate of the UART in the range 50 … 10,000,000 baud (bits per second).

The baud rate depends on the parameters 2, 3 and 4, and can be calculated by the following formula:

```
BaudRate = (10^8 · uart_x_dcm_m) / (4 · uart_x_dcm_d · (uart_dcm_clk_divider+1))
```

With:

Variable	Parameter	Description
uart_x_dcm_m[1)]	Parameter(2).Init	Multiplier for the digital clock manager (DCM) module in the range 2 … 255.
uart_x_dcm_d[1)]	Parameter(3).Init	Divisor for the digital clock manager (DCM) module in the range 1 … 255.
uart_x_dcm_clk_divider[1)]	Parameter(4).Init	UART clock divider in the range 0 … 262,143.

[1)] x=1 for UART 1; x=2 for UART 2

Tip

In the framework folder you find a MATLAB file providing some calculated baud rates and the percentage deviations to the supported baud rates according to the parameters m, d and the clock divider.

- Framework folder:
  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA1401Tp1_<Multi_I/O_Module_type>_<FPGA_type>`
- MATLAB file name: `FPGA1401Tp1_XC7K325T_uart_parameters.mat`

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(5).Init / Word length Lets you specify the word length in the range 5 … 9 bit. The word length includes the number of data bits and the optional parity bit. Exceeding bits in a message are ignored at the transmitter or cleared at the receiver.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(6).Init / Stop bits Lets you specify the length of the stop bits in half of bits.

Stop Bits	Parameter Value
1	2
1.5	3
2	4

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(7).Init / UART mode Lets you specify the mode when using the RS422/485 UART type.

Value	UART Mode
0	Full-duplex mode
1	Half-duplex mode

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(8).Init / UART type Lets you specify the UART type.

Value	UART Type
0	RS232
1	RS422/485

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(9).Init / Termination Lets you specify the termination state.

Value	Termination State	Description
0	Disconnected	▪ Full-duplex mode: RX-/RX+ and TX-/TX+ signals are not terminated. ▪ Half-duplex mode: BM/BP signals are not terminated.
1	Connected	▪ Full-duplex mode: RX-/RX+ and TX-/TX+ signals are terminated via 120 Ω resistors.

Value	Termination State	Description
		▪ Half-duplex mode: BM/BP signal are terminated via a 120 Ω resistor.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 1.

uart_<ChannelNumber>_rd / 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 FPGA clock cycle from the UART.

uart_<ChannelNumber>_rd_data_count / 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.

Value range: 0 … 2047

The channel number can be specified in the range 0 … 1.

uart_<ChannelNumber>_rd_fifo_empty / 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.

Range:

- 0: The RX FIFO buffer is not empty.
- 1: The RX FIFO buffer is empty.

uart_<ChannelNumber>_rd_data / 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.

Range: 0 … 511

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).

uart_<ChannelNumber>_wr / 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.

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).

uart_<ChannelNumber>_wr_data_count / 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.

Range: 0 … 2047

uart_<ChannelNumber>_wr_fifo_full / 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.

Range:

- 0: The TX FIFO buffer is not full.
- 1: The TX FIFO buffer is full.

uart_<ChannelNumber>_wr_data / Write Data Specifies the value to be send.

The Write_Data signal is transferred at each clock cycle with Write_Enable set to 1.

Range: 0 … 511

uart_<ChannelNumber>_driver_en / 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.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1401Tp1 framework for serial communication using the UART (RS422/485) function. 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.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37
UART (RS232)..... 370

I/O Functions of the FPGA1401Tp1 with Engine Control I/O Module Framework

Introduction

The FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) provides the I/O functionality of MicroAutoBox II with a DS1554 Engine Control I/O Module.

Where to go from here

Information in this section

ADC (Type A)..... 382
To read data from an analog input signal in the FPGA application using the Type A conversion function.

Buffer In..... 385
To read data from an intermodule-bus buffer with a data width of 32 bits.

Buffer64 In..... 386
To read data from an intermodule-bus buffer with a data width of 64 bits.

Buffer Out..... 388
To write data to an intermodule-bus buffer with a data width of 32 bits.

Buffer64 Out..... 390
To write data to an intermodule-bus buffer with a data width of 64 bits.

Digital Crank/Cam Sensor..... 392
To provide bit-wise read access to digital camshaft and crankshaft sensors.

Digital In (Type B)..... 394
To read data from a digital input signal in the FPGA application using a digital bidirectional channel.

Digital Out (Type A)..... 396
To write data to a digital output signal in the FPGA application using a digital output channel.

Digital Out (Type B)..... 399
To write data to a digital output signal in the FPGA application using a digital bidirectional channel.

Inductive Zero Voltage Detector..... 401
To provide read access to an inductive zero voltage detector.

Interrupt..... 402
To request a processor interrupt outside of the FPGA application.

Knock Sensor..... 403
To read data from a knock sensor in the FPGA application.

LED Out..... 404
To write a digital signal that controls the FPGA status LED near the DS1514 ZIF I/O connector.

Register In..... 405
To read data from an intermodule-bus register with a data width of 32 bits.

Register64 In..... 407
To read data from an intermodule-bus register with a data width of 64 bits.

Register Out..... 409
To write data to an intermodule-bus register with a data width of 32 bits.

Register64 Out..... 410
To write data to an intermodule-bus register with a data width of 64 bits.

Status In..... 412
To read digital signals that outputs state information, e.g.: state of the FPGA initialization sequence.

Temperature..... 413
To read the FPGA die temperature.

ADC (Type A)

Purpose

To read data from an analog input signal in the FPGA application by using the Type A conversion function.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the ADC (Type A) I/O functions. There are 14 analog input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 10 … 23.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 13.

hq_adc_<ChannelNumber>_data / 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

hq_adc_<ChannelNumber>_soc / Start of conversion 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 `End of conversion` outport signals the end of the conversion process.

Setting this value permanently to 1 results in continuous sampling.

hq_adc_<ChannelNumber>_eoc / End of conversion Outputs an end of conversion signal if the conversion result is available on the specified channel. 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

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 V2	AnalogIn+ ch 1 AnalogIn- ch 1[1)]
	2	Y2 X2	AnalogIn+ ch 2 AnalogIn- ch 2[1)]
	3	S2 R2	AnalogIn+ ch 3 AnalogIn- ch 3[1)]
	4	T2 U2	AnalogIn+ ch 4 AnalogIn- ch 4[1)]
	5	V5 W6	AnalogIn+ ch 5 AnalogIn- ch 5[1)]
	6	W3 V3	AnalogIn+ ch 6 AnalogIn- ch 6[1)]
	7	T3 U3	AnalogIn+ ch 7 AnalogIn- ch 7[1)]
	8	U5 V6	AnalogIn+ ch 8 AnalogIn- ch 8[1)]
	9	S5 T6	AnalogIn+ ch 9 AnalogIn- ch 9[1)]
	10	T5 U6	AnalogIn+ ch 10 AnalogIn- ch 10[1)]
	11	R5 R6	AnalogIn+ ch 11 AnalogIn- ch 11[1)]
	12	S3 R3	AnalogIn+ ch 12 AnalogIn- ch 12[1)]
	13	P5 P6	AnalogIn+ ch 13 AnalogIn- ch 13[1)]
	14	P3 P2	AnalogIn+ ch 14 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 🕮).

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Buffer In

Purpose

To read data from an intermodule-bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is read from an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf_<ChannelNumber>_count / 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).

xmemf_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemf_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. The valid port range depends on the number of elements currently in the buffer. Refer to 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.

Related topics

References

Buffer Out 388
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register In 405

Buffer64 In

Purpose

To read data from an intermodule-bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is read from an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

The I/O function number can be specified in the range 289 … 320.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf64_<ChannelNumber>_count / 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).

xmemf64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemf64_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf64_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer64 Out 390
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register64 In 407

Buffer Out

Purpose

To write data to an intermodule-bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is written to an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemp_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp_<ChannelNumber>_finished / 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.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the next clock cycle.

The ready flag must be set no later than the last data value. Otherwise the buffer switches twice.

xmemp_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer In 385
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register Out 409

Buffer64 Out

Purpose

To write data to an intermodule-bus buffer with a data width of 64 bits.

Description

If you select **Buffer64** as the access type, the data is written to an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 ... 320.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemp64_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp64_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp64_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer64 In 386
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register64 Out 410

Digital Crank/Cam Sensor

Purpose

To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the Digital Crank/Cam Sensor I/O functions. There are five input channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 28 … 32.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Low threshold voltage 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.

- Range: -40000 mV … +40000 mV
- Resolution: 20 mV
- Default: 1000 mV

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / High threshold voltage 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.

- Range: -40000 mV … +40000 mV
- Resolution: 20 mV
- Default: 1000 mV

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 4.

cam_<ChannelNumber> / 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: 80 MHz

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

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Digital In (Type B)

Purpose

To read data from a digital input signal in the FPGA application by using a digital bidirectional channel.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the **Digital In (Type B)** I/O functions. There are eight digital input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 9.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Threshold voltage Lets you specify the threshold level for the current digital channel in steps of 100 mV. If the input signal is below this level, a logical 0 is detected. Otherwise, a logical 1 is detected.

- Range: 1000 mV … 7500 mV
- Resolution: 100 mV
- Default: 1500 mV

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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 07.

bidir_<ChannelNumber>_in / Data Outputs the current results of digital input channel.

- 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: 80 MHz

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 I/O (Bidirectional) (MicroAutoBox II Hardware Reference 🕮).
- Asynchronous input data might lead to metastable register states. Further synchronization techniques might be necessary.

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

Related topics

References

Digital Out (Type A) 396
Digital Out (Type B) 399
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37

Digital Out (Type A)

Purpose

To write data to a digital output signal in the FPGA application using a digital output channel.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the Digital Out (Type A) I/O functions. There are 40 digital output channels.

The voltage range for the high-side switch for all digital output channels is in the range 0 V … 45 V.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 41.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 39.

dig_<ChannelNumber>_out / 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: 80 MHz

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 🕮).

dig_<ChannelNumber>_oe / Enable Controls the hardware output. If set to 1, the hardware output reacts to the Data outport. Otherwise, it is set to High-Z.

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

Related topics

References

Digital In (Type B) 394
Digital Out (Type B) 399
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37

Digital Out (Type B)

Purpose

To write data to a digital output signal in the FPGA application by using a digital bidirectional channel.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the Digital Out (Type B) I/O functions. There are eight digital output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 42 … 49.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(42).Parameter(1).Init / High supply Lets you select the voltage for the high-side switch for all digital output channels.

- 0: 3.3 V
- 1: 5 V

Note

You can specify the high supply voltage value only globally for all digital output channels.
The I/O function number must be specified as 42.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 07.

bidir_<ChannelNumber>_out / 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: 80 MHz

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 🕮).

bidir_<ChannelNumber>_oe / 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

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

Related topics

References

Digital In (Type B) 394
Digital Out (Type A) 396

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Inductive Zero Voltage Detector

Purpose

To provide read access to an inductive zero voltage detector.

Description

The FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides one channel for the **Inductive Zero Voltage Detector** I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 33.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

crank_0 / 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 one clock cycle.

Data type: UFix_1_0

- 0: No zero crossing.
- 1: Zero crossing is detected.

Update rate: 80 MHz

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–

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Interrupt

Purpose

To request a processor interrupt outside of the FPGA application.

Description

The FPGA1401Tp1 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_INT_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 8.

IRQProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 7.

usr_<ChannelNumber>_interrupt / Int Provides the interrupt request line.

- 0 to 1: Interrupt is requested (edge-triggered).
- 0: No interrupt is requested. Last requested interrupt is saved.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Knock Sensor

Purpose

To read data from a knock sensor in the FPGA application.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the Knock Sensor I/O functions. There are 4 knock sensor input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 24 … 27.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 03.

knock_<ChannelNumber>_data / Data Outputs the current results of the A/D conversions on the current channel.

Data type: UFix_16_0

Range: 0 … +65535

Input voltage range: –5 V ... +5 V

Update rate: 1 Msps

knock_<ChannelNumber>_soc / Start of conversion Triggers the start of an A/D conversion on the specified channel. 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 `End of conversion` outport signals the end of the conversion process.

Setting this value permanently to 1 results in continuous sampling.

Data type: UFix_1_0

knock_<ChannelNumber>_eoc / End of conversion Outputs an end of conversion signal if the conversion result is available on the specified channel. 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

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 🕮).

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

LED Out

Purpose

To write a digital signal that controls the FPGA status LED.

You can find the FPGA status LED near the DS1514 ZIF I/O connector.

Description

The FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides one channel for the LED Out I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

led_out / Data Controls the FPGA status LED.

Data type: UFix_1_0

- 0: LED lights up green.
- 1: LED lights up orange.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

Register In

Purpose

To read data from an intermodule-bus register with a data width of 32 bits.

Description

If you select Register 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. 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

0 represents the lowest bit position, 32 the highest bit position.

- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the intermodule bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an intermodule-bus register. The data format depends on the related parameter settings.

xreg_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer In 385
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register Out 409

Register64 In

Purpose

To read data from an intermodule-bus register with a data width of 64 bits.

Description

If you select **Register64** as the access type, the data is read from an intermodule-bus register. 128 registers are available 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_WRITE_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 … 288.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the **Data** outport depending on the format selected in the **Format** setting (see below).

- signed/unsigned

  The values of the **Data** outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the **Data** outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the intermodule bus.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an intermodule-bus register. The data format depends on the related parameter settings.

xreg64_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer64 In ... 386
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) ... 37
Register64 Out ... 410

Register Out

Purpose

To write data to an intermodule-bus register with a data width of 32 bits.

Description

If you select Register 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. 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an intermodule-bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer Out 388
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register In 405

Register64 Out

Purpose

To write data to an intermodule-bus register with a data width of 64 bits.

Description

If you select **Register64** as the access type, the data is written to an intermodule-bus register. 128 registers are available 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_XDATA_READ_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 … 288.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an intermodule-bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer64 Out 390
Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37
Register64 In 407

Status In

Purpose

To read digital signals that output state information, e.g., the state of the FPGA initialization sequence.

Description

The FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides one channel for the **Status In** I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

init_done/ Init Done Outputs the state of the initialization sequence that is started after programming the FPGA.

- 0: Initialization sequence is in progress.
- 1: Initialization sequence has finished.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1) 37

Temperature

Purpose

To read the FPGA die temperature.

Description

The FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides one channel for the **Temperature** I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 34.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

temperature / Temperature 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_temp / 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.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox II (FPGA1401Tp1)..... 37

I/O Functions of the FPGA1403Tp1 with Multi-I/O Module Frameworks

Introduction

The FPGA1403Tp1 frameworks provide the standard I/O functionality of MicroAutoBox II with DS1552 (*FPGA1403Tp1_DS1552_XC7K325T*) or with DS1552B1 Multi-I/O Module (*FPGA1403Tp1_DS1552B1_XC7K325T*).

Where to go from here

Information in this section

ADC (Type A)..... 417
To read data from an Analog In 10/Analog In 11 channel.

ADC (Type B)..... 418
To read data from an Analog In 12 channel.

Buffer In..... 420
To read data from an intermodule-bus buffer with a data width of 32 bits.

Buffer64 In..... 421
To read data from an intermodule-bus buffer with a data width of 64 bits.

Buffer Out..... 423
To write data to an intermodule-bus buffer with a data width of 32 bits.

Buffer64 Out..... 425
To write data to an intermodule-bus buffer with a data width of 64 bits.

DAC..... 427
To write data to an Analog Out 13 channel in the FPGA application.

Digital Crank/Cam Sensor..... 428
To provide bit-wise read access to digital camshaft and crankshaft sensors.

Digital In (Type A)..... 430
To read data from a digital input signal in the FPGA application using a Digital In 5 channel.

Digital In (Type B)..... 432
To read data from a digital input signal in the FPGA application by using a Digital InOut 6 channel.

Digital Out (Type A)..... 434
To write data to a digital output signal in the FPGA application using a Digital Out 5 channel.

Digital Out (Type B)..... 436
To write data to a digital output signal in the FPGA application using a digital bidirectional channel.

Inductive Zero Voltage Detector..... 438
To provide read access to an inductive zero voltage detector.

Interrupt..... 439
To request a processor interrupt outside of the FPGA application.

LED Out..... 440
To write a digital signal that controls the FPGA status LED near the DS1514 ZIF I/O connector.

Register In..... 441
To read data from an intermodule-bus register with a data width of 32 bits.

Register64 In..... 442
To read data from an intermodule-bus register with a data width of 64 bits.

Register Out..... 444
To write data to an intermodule-bus register with a data width of 32 bits.

Register64 Out..... 446
To write data to an intermodule-bus register with a data width of 64 bits.

Sensor Supply..... 447
To provide a supply voltage at a connected sensor.

Status In..... 448
To read digital signals that outputs state information, e.g.: state of the FPGA initialization sequence or the FPGA die temperature.

UART (RS232)..... 450
To implement RS232 communication via a UART 3 channel.

UART (RS422/485)..... 455
To implement RS422/485 communication via a UART 3 channel.

ADC (Type A)

Purpose

To read data from an analog input signal in the FPGA application using the ADC (Type A) conversion function for the Analog In 10/Analog In 11 channel.

Description

According to the number of physical connections available on the DS1552/DS1552B1 I/O module, you can select the ADC (Type A) I/O functions. There are eight analog input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 26 … 33.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 7.

hq_adc_<ChannelNumber>_data / Data Outputs the current results of the A/D conversions on the current channel.

Range: 0 … +65535

Update rate: 1 Msps

hq_adc_<ChannelNumber>_soc / Start of conversion Lets you trigger the start of an A/D conversion on the specified channel. 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.

hq_adc_<ChannelNumber>_eoc / End of conversion Outputs an end of conversion signal if the conversion result is available on the specified channel. 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.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for analog input channels using the Type A conversion function. 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

Related topics

References

ADC (Type B)..... 418
DAC..... 427
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

ADC (Type B)

Purpose

To read data from an analog input signal in the FPGA application using the ADC (Type B) conversion function for the Analog In 12 channel.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the ADC (Type B) I/O functions. There are 16 analog output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 34 … 49.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

Iq_adc_<ChannelNumber>_data / Data Outputs the current results of the A/D conversions on the current channel in the range -32768 … +32767.

Update rate: 0.2 Msps

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for analog input channels using the ADC (Type B) conversion function. 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

Related topics

References

ADC (Type A)..... 417
DAC..... 427
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Buffer In

Purpose

To read data from an intermodule-bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is read from an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

The I/O function number can be specified in the range 129 … 160.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf_<ChannelNumber>_count / 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).

xmemf_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemf_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer Out 423
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register In 441

Buffer64 In

Purpose

To read data from an intermodule-bus buffer with a data width of 64 bits.

Description

If you select **Buffer64** as the access type, the data is read from an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

The I/O function number can be specified in the range 289 … 320.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf64_<ChannelNumber>_count / 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).

xmemf64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemf64_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf64_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer64 Out 425
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register64 In 442

Buffer Out

Purpose

To write data to an intermodule-bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is written to an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemp_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer In 420
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register Out 444

Buffer64 Out

Purpose

To write data to an intermodule-bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is written to an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 ... 320.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemp64_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp64_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp64_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer64 In 421
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register64 Out 446

DAC

Purpose

To write data to an Analog Out 13 channel in the FPGA application.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the DAC I/O functions. There are four analog output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 26 … 29.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 3.

dac<ChannelNumber>_data / Data Outputs a signal in the range 0 … +65535.

Range exceeding is possible and will be saturated to the 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).

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for analog output channels. 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

Related topics

References

ADC (Type A)..... 417
ADC (Type B)..... 418
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Digital Crank/Cam Sensor

Purpose

To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **Digital Crank/Cam Sensor** I/O functions. There are three input channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 50 … 52.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Low threshold voltage To 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.

- Range:
  -40000 mV … +40000 mV
- Resolution:
  20 mV
- Default:
  1000 mV

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / High threshold voltage To 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.

- Range:
  -40000 mV … +40000 mV
- Resolution:
  20 mV
- Default:
  1000 mV

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 2.

cam_<ChannelNumber> / Data 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: 80 MHz

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for analog input channels. 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 -

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Digital In (Type A)

Purpose

To read data from a digital input signal in the FPGA application using a Digital In 5 channel.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the Digital In (Type A) I/O functions. There are 16 digital input channels.

The threshold level is fix:

- 3.6 V for low-high transition
- 1.2 V for high-low transition

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.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 17.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

dig_<ChannelNumber>_in / Data Outputs the current results of digital input channel.

- 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: 80 MHz

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 🕮).
- Asynchronous input data might lead to metastable register states. Further synchronization techniques might be necessary.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for digital input channels. 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

Related topics

References

Digital In (Type A)..... 430
Digital In (Type B)..... 432
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Digital In (Type B)

Purpose

To read data from a digital input signal in the FPGA application by using a **Digital InOut 6** channel.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **Digital In (Type B)** I/O functions. There are eight digital input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 18 … 25.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / 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.

- 1000: 1000 mV threshold level
- …
- 7500: 7500 mV threshold level

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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 07.

bidir_<ChannelNumber>_in / Data Outputs the current results of digital input channel.

- 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: 80 MHz

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 🕮).
- Asynchronous input data might lead to metastable register states. Further synchronization techniques might be necessary.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for digital bidirectional channels. 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.

Related topics

References

Digital In (Type A) 430
Digital Out (Type B) 436
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40

Digital Out (Type A)

Purpose

To write data to a digital output signal in the FPGA application using a **Digital Out 5** channel.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **Digital Out (Type A)** I/O functions. There are 16 digital output channels.

The voltage range for the high side switch for all digital output channels is in the range 0 V … 45 V.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 17.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 15.

dig_<ChannelNumber>_out / 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).

Update rate: 80 MHz

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 Interface Characteristics (MicroAutoBox III Hardware Installation and Configuration 🕮).

dig_<ChannelNumber>_oe / Enable Controls the hardware output. If set to 1, the hardware output reacts to the **Data** outport, otherwise it is set to High-Z.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for digital output channels. 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

Related topics

References

Digital In (Type A)..... 430
Digital Out (Type B)..... 436
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Digital Out (Type B)

Purpose

To write data to a digital output signal in the FPGA application by using a **Digital InOut 6** channel.

Description

According to the number of physical connections available on the DS1514 Multi-I/O Module, you can select the **Digital Out (Type B)** I/O functions. There are eight digital output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 18 … 25.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(18).Parameter(1).Init / High supply Lets you select the voltage for the high side switch for all digital output channels.

- 0: 3.3 V
- 1: 5 V

Note

You can specify the high supply voltage value only globally for all digital output channels.
The I/O function number must be specified as 18.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 07.

bidir_<ChannelNumber>_out / 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).

Update rate: 80 MHz

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 🕮).

bidir_<ChannelNumber>_oe / Enable Controls the hardware output. If set to 1, the hardware output reacts to the **Data** outport, otherwise it is set to High-Z.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for digital bidirectional channels. 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.

Related topics

References

Digital In (Type B) 432
Digital Out (Type A) 434
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40

Inductive Zero Voltage Detector

Purpose

To provide read access to an inductive zero voltage detector.

Description

The FPGA1403Tp1 frameworks provide one channel for the Inductive Zero Voltage Detector I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 53.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

crank / Data To detect 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: 80 MHz

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for analog input channels. 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 -

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Interrupt

Purpose

To request a processor interrupt outside of the FPGA application.

Description

The FPGA1403Tp1 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_INT_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 8.

IRQProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 7.

usr_<ChannelNumber>_interrupt / Int Provides the interrupt request line.

- 0 to 1: Interrupt is requested (edge-triggered).
- 0: No interrupt is requested. Last requested interrupt is saved.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

LED Out

Purpose

To write a digital signal that controls the FPGA status LED.

You can find the FPGA status LED near the DS1514 ZIF I/O connector.

Description

The FPGA1403Tp1 frameworks provide one channel for the LED Out I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

led_out / Data Controls the Status LED on the board's bracket.

- 0: LED lights green.
- 1: LED lights orange.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Register In

Purpose

To read data from an intermodule-bus register with a data width of 32 bits.

Description

If you select Register 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. 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.

- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the .

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an intermodule-bus register. The data format depends on the related parameter settings.

xreg_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer In 420
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register Out 444

Register64 In

Purpose

To read data from an intermodule-bus register with a data width of 64 bits.

Description

If you select Register64 as the access type, the data is read from an intermodule-bus register. 128 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 … 288.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the .

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an intermodule-bus register. The data format depends on the related parameter settings.

xreg64_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer64 In....421
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)....40
Register64 Out....446

Register Out

Purpose

To write data to an intermodule-bus register with a data width of 32 bits.

Description

If you select **Register** 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. 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an intermodule-bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer Out....423
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)....40
Register In....441

Register64 Out

Purpose

To write data to an intermodule-bus register with a data width of 64 bits.

Description

If you select Register64 as the access type, the data is written to an intermodule-bus register. 128 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 … 288.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the **Binary point position (or fraction width)** setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an intermodule-bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer64 Out 425
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register64 In 442

Sensor Supply

Purpose

To provide a supply voltage, for example, for a connected sensor, in the range 2000 mV … 20000 mV in steps of 100 mV.

Description

The FPGA1403Tp1 frameworks provide one channel for the **Sensor Supply** I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 30.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Supply voltage Lets you specify the supply voltage a connected sensor is to be driven with in the range 2000 mV … 20000 mV in steps of 100 mV.

Port

There is no port to be specified.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for for sensor supply. Depending on the MicroAutoBox variant the signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Sim_Data	1	b6	VSENS-
		c6	VSENS+

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Status In

Purpose

To read digital signals that outputs state information, e.g.: state of the FPGA initialization sequence or the FPGA die temperature.

Description

The FPGA1403Tp1 framework provides one channel for the Status In I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

init_done/ Init Done Outputs the state of the initialization sequence that is started after programming the FPGA.

- 0: Initialization sequence is in progress.
- 1: Initialization sequence has finished.

temperature / Temperature Outputs the raw value of the FPGA's die temperature measurement. Use the 12 MSB bits to calculate the die temperature.

Equation to calculate 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_temp / 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.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

UART (RS232)

Purpose

To implement RS232 communication via a UART 3 channel.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the UART (RS232) I/O functions. There are two interfaces.

Note

UART 1 can be used without modification. To use UART 2, your DS1552 has to be modified by dSPACE.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 31 … 32.

Most of the parameters are used for the UART (RS232) and UART 3 (RS422/485) I/O functions.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Baud rate Lets you specify the baud rate of the UART in the range 50 … 1,000,000 baud (bits per second).

With:

Variable	Parameter	Description
uart_x_dcm_m[1)]	Parameter(2).Init	Multiplier for the digital clock manager (DCM) module in the range 2 … 255.
uart_x_dcm_d[1)]	Parameter(3).Init	Divisor for the digital clock manager (DCM) module in the range 1 … 255.
uart_x_dcm_clk_divider[1)]	Parameter(4).Init	UART clock divider in the range 0 … 262,143.

1) x=1 for UART 1; x=2 for UART 2

Note

Limitations:

- The maximum baud rate of 1,000,000 baud must not be exceeded.
- The output frequency of the digital clock manager (DCM) module should be between 40 MHz and 160 MHz:
  f_{DCM} = 200 MHz · uart_x_dcm_m / uart_x_dcm_d

Tip

In the framework folder you find a MATLAB file that provides some calculated baud rates and the percentage deviations to the supported baud rates according to the parameters m, d and the clock divider.

- Framework folder:
  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA 1403Tp1_<Multi_I/O_Module_type>_<FPGA_type>`
- MATLAB file name: `FPGA1403Tp1_XC7K325T_uart_parameters.mat`

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(5).Init / Word length Lets you specify the word length in the range 5 … 9 bit. The word length includes the number of data bits and the optional parity bit. Exceeding bits in a message are ignored at the transmitter or cleared at the receiver.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(6).Init / Stop bits Lets you specify the length of the stop bits in half of bits.

Stop Bits	Parameter Value
1	2
1.5	3
2	4

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(7).Init / UART type Lets you specify the UART type.

Value	UART Type
0	RS232
1	RS422/485

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(8).Init / Termination Lets you specify the termination state.

Note

For the RS232 UART type, the termination must be set to 0 (disconnected).

Value	Termination State	Description
0	Disconnected	▪ The RX/CTS and TX/RTS signals are not terminated.
1	Connected	▪ Not allowed

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 1.

uart_<ChannelNumber>_rd / 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 FPGA clock cycle from the UART.

uart_<ChannelNumber>_rd_data_count / 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.

Value range: 0 … 2047

uart_<ChannelNumber>_rd_fifo_empty / 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.

Range:

- 0: The RX FIFO buffer is not empty.
- 1: The RX FIFO buffer is empty.

uart_<ChannelNumber>_rd_data / 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.

Range: 0 … 511

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).

uart_<ChannelNumber>_wr / 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.

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 baudrate. The hardware port has inverted voltage levels of -6 V (logical high) and +6 V (logical low).

uart_<ChannelNumber>_wr_data_count / 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.

Range: 0 … 2047

uart_<ChannelNumber>_wr_fifo_full / 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.

Range:

- 0: The TX FIFO buffer is not full.
- 1: The TX FIFO buffer is full.

uart_<ChannelNumber>_wr_data / Write Data Specifies the value to be send.

The Write_Data signal is transferred at each clock cycle with Write_Enable set to 1.

Range: 0 … 511

uart_<ChannelNumber>_rts / 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.

The hardware port is synchronously running to the UART clock defined by the UART baudrate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive).

uart_<ChannelNumber>_cts / 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.

Range:

- 0: CTS inactive
- 1: CTS active

The CTS hardware port is synchronously running to the UART clock defined by the UART baudrate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive).

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for serial communication using the UART (RS232) function. 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.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40
UART (RS422/485)..... 455

UART (RS422/485)

Purpose

To implement RS422/485 communication via a UART 3 channel.

Description

According to the number of physical connections available on the DS1552 Multi-I/O Module, you can select the **UART 3 (RS422/485)** I/O functions. There are two interfaces.

Note

UART 1 can be used without modification. To use UART 2, your DS1552 has to be modified by dSPACE.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 33 … 34.

Most of the parameters are used for the UART 3 (RS232) and UART (RS422/485) I/O functions.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Baud rate Lets you specify the baud rate of the UART in the range 50 … 10,000,000 baud (bits per second).

The baud rate depends on the parameters 2, 3 and 4, and can be calculated by the following formula:

```
BaudRate = (10^8 · uart_x_dcm_m) / (4 · uart_x_dcm_d · (uart_dcm_clk_divider+1))
```

With:

Variable	Parameter	Description
uart_x_dcm_m[1)]	Parameter(2).Init	Multiplier for the digital clock manager (DCM) module in the range 2 … 255.
uart_x_dcm_d[1)]	Parameter(3).Init	Divisor for the digital clock manager (DCM) module in the range 1 … 255.
uart_x_dcm_clk_divider[1)]	Parameter(4).Init	UART clock divider in the range 0 … 262,143.

[1)] x=1 for UART 1; x=2 for UART 2

Tip

In the framework folder you find a MATLAB file providing some calculated baud rates and the percentage deviations to the supported baud rates according to the parameters m, d and the clock divider.

- Framework folder:
  `<RCP_HIL_InstallationPath>\MATLAB\RTIFPGA\Frameworks\FPGA1403Tp1_<Multi_I/O_Module_type>_<FPGA_type>`
- MATLAB file name: `FPGA1403Tp1_XC7K325T_uart_parameters.mat`

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(5).Init / Word length Lets you specify the word length in the range 5 … 9 bit. The word length includes the number of data bits and the optional parity bit. Exceeding bits in a message are ignored at the transmitter or cleared at the receiver.

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(6).Init / Stop bits Lets you specify the length of the stop bits in half of bits.

Stop Bits	Parameter Value
1	2
1.5	3
2	4

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(7).Init / UART mode Lets you specify the mode when using the RS422/485 UART type.

Value	UART Mode
0	Full-duplex mode
1	Half-duplex mode

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(8).Init / UART type Lets you specify the UART type.

Value	UART Type
0	RS232
1	RS422/485

IOProperties.Out.Fct(<IOFunctionNumber>).Parameter(9).Init / Termination Lets you specify the termination state.

Value	Termination State	Description
0	Disconnected	▪ Full-duplex mode: RX-/RX+ and TX-/TX+ signals are not terminated. ▪ Half-duplex mode: BM/BP signals are not terminated.
1	Connected	▪ Full-duplex mode: RX-/RX+ and TX-/TX+ signals are terminated via 120 Ω resistors.

Value	Termination State	Description
		▪ Half-duplex mode: BM/BP signal are terminated via a 120 Ω resistor.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 1.

uart_<ChannelNumber>_rd / 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 FPGA clock cycle from the UART.

uart_<ChannelNumber>_rd_data_count / 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.

Value range: 0 … 2047

The channel number can be specified in the range 0 … 1.

uart_<ChannelNumber>_rd_fifo_empty / 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.

Range:

- 0: The RX FIFO buffer is not empty.
- 1: The RX FIFO buffer is empty.

uart_<ChannelNumber>_rd_data / 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.

Range: 0 … 511

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).

uart_<ChannelNumber>_wr / 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.

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).

uart_<ChannelNumber>_wr_data_count / 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.

Range: 0 … 2047

uart_<ChannelNumber>_wr_fifo_full / 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.

Range:

- 0: The TX FIFO buffer is not full.
- 1: The TX FIFO buffer is full.

uart_<ChannelNumber>_wr_data / Write Data Specifies the value to be send.

The Write_Data signal is transferred at each clock cycle with Write_Enable set to 1.

Range: 0 … 511

uart_<ChannelNumber>_driver_en / 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.

I/O mapping

The following I/O mapping is relevant if you use a FPGA1403Tp1 framework for serial communication using the UART (RS422/485) function. 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.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40
UART (RS232)..... 450

I/O Functions of the FPGA1403Tp1 with Engine Control I/O Module Framework

Introduction

The FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) provides the I/O functionality of MicroAutoBox with a DS1554 Engine Control I/O Module.

Where to go from here

Information in this section

ADC (Type A)....462
To read data from an Analog In 14 channel in the FPGA application by using the ADC (Type A) conversion function.

Buffer In....465
To read data from an intermodule-bus buffer with a data width of 32 bits.

Buffer64 In....466
To read data from an intermodule-bus buffer with a data width of 64 bits.

Buffer Out....468
To write data to an intermodule-bus buffer with a data width of 32 bits.

Buffer64 Out....470
To write data to an intermodule-bus buffer with a data width of 64 bits.

Digital Crank/Cam Sensor....472
To provide bit-wise read access to digital camshaft and crankshaft sensors.

Digital In (Type B)....473
To read data from a digital input signal in the FPGA application by using a Digital InOut 8 channel.

Digital Out (Type A)....475
To write data to a digital output signal in the FPGA application using a Digital Out 7 channel.

Digital Out (Type B)....478
To write data to a digital output signal in the FPGA application by using a Digital In/Out 8 channel.

Inductive Zero Voltage Detector....480
To provide read access to an inductive zero voltage detector.

Interrupt....481
To request a processor interrupt outside of the FPGA application.

Knock Sensor....482
To read data from a knock sensor in the FPGA application.

LED Out....483
To write a digital signal that controls the FPGA status LED near the DS1514 ZIF I/O connector.

Register In....484
To read data from an intermodule-bus register with a data width of 32 bits.

Register64 In....486
To read data from an intermodule-bus register with a data width of 64 bits.

Register Out....487
To write data to an intermodule-bus register with a data width of 32 bits.

Register64 Out....489
To write data to an intermodule-bus register with a data width of 64 bits.

Status In....490
To read digital signals that outputs state information, e.g.: state of the FPGA initialization sequence.

Temperature....491
To read the FPGA die temperature.

ADC (Type A)

Purpose

To read data from an **Analog In 14** channel in the FPGA application by using the ADC (Type A) conversion function.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the **Analog In 14** I/O functions. There are 14 analog input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 10 … 23.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 13.

hq_adc_<ChannelNumber>_data / 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

hq_adc_<ChannelNumber>_soc / Start of conversion 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 `End of conversion` outport signals the end of the conversion process.

Setting this value permanently to 1 results in continuous sampling.

hq_adc_<ChannelNumber>_eoc / End of conversion Outputs an end of conversion signal if the conversion result is available on the specified channel. 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

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 -

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Buffer In

Purpose

To read data from an intermodule-bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is read from an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf_<ChannelNumber>_count / 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).

xmemf_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemf_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. The valid port range depends on the number of elements currently in the buffer. Refer to 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.

Related topics

References

Buffer Out 423
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register In 441

Buffer64 In

Purpose

To read data from an intermodule-bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is read from an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

The I/O function number can be specified in the range 289 … 320.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemf64_<ChannelNumber>_count / 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).

xmemf64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemf64_<ChannelNumber>_new_data / Data New Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then back to 0, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle.

xmemf64_<ChannelNumber>_addr / Address Specifies an element in the buffer you want to read. The block requires 1 clock cycle to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size. 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.

Related topics

References

Buffer64 Out ... 425
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) ... 40
Register64 In ... 442

Buffer Out

Purpose

To write data to an intermodule-bus buffer with a data width of 32 bits.

Description

If you select Buffer as the access type, the data is written to an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 32 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 129 … 160.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemp_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp_<ChannelNumber>_finished / 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.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the next clock cycle.

The ready flag must be set no later than the last data value. Otherwise the buffer switches twice.

xmemp_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer In ... 420
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) ... 40
Register Out ... 444

Buffer64 Out

Purpose

To write data to an intermodule-bus buffer with a data width of 64 bits.

Description

If you select Buffer64 as the access type, the data is written to an intermodule-bus buffer. 32 buffers are available. Each buffer has a variable buffer size of 1 up to 32768 elements. Each buffer element has a data width of 64 bits.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 289 ... 320.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 31.

xmemp64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an intermodule-bus buffer. The data format depends on the related parameter settings.

xmemp64_<ChannelNumber>_write / Enable Specifies the current valid Data port value.

- 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.

xmemp64_<ChannelNumber>_finished / Ready Explicitly specifies the buffer state as ready to send the buffer immediately, even if it is not completely filled. The data values will be written to a new buffer in the next clock cycle. While the port value is 1, transmission switches buffer in every clock cycle. The value should therefore be set for one clock cycle only. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer.

- 0: The buffer is not ready to send.
- 1: The buffer is marked as ready for sending, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the next clock cycle.

The ready flag must be set no later than the last data value, otherwise the buffer switches twice.

xmemp64_<ChannelNumber>_overflow / 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.

Related topics

References

Buffer64 In....421
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)....40
Register64 Out....446

Digital Crank/Cam Sensor

Purpose

To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the Digital Crank/Cam Sensor I/O functions. There are five input channels.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 28 … 32.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Low threshold voltage 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.

- Range: -40000 mV … +40000 mV
- Resolution: 20 mV
- Default: 1000 mV

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / High threshold voltage 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.

- Range: -40000 mV … +40000 mV
- Resolution: 20 mV
- Default: 1000 mV

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 4.

cam_<ChannelNumber> / 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: 80 MHz

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

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Digital In (Type B)

Purpose

To read data from a digital input signal in the FPGA application by using a **Digital InOut 8** channel.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the **Digital In (Type B)** I/O functions. There are eight digital input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 9.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Threshold voltage Lets you specify the threshold level for the current digital channel in steps of 100 mV. If the input signal is below this level, a logical 0 is detected. Otherwise, a logical 1 is detected.

- Range: 1000 mV … 7500 mV
- Resolution: 100 mV
- Default: 1500 mV

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.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 07.

bidir_<ChannelNumber>_in / Data Outputs the current results of digital input channel.

- 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: 80 MHz

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 🕮).
- Asynchronous input data might lead to metastable register states. Further synchronization techniques might be necessary.

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

Related topics

References

Digital Out (Type A)..... 475
Digital Out (Type B)..... 478
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Digital Out (Type A)

Purpose

To write data to a digital output signal in the FPGA application using a Digital Out 7 channel.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the Digital Out (Type A) I/O functions. There are 40 digital output channels.

The voltage range for the high-side switch for all digital output channels is in the range 0 V … 45 V.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 2 … 41.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 39.

dig_<ChannelNumber>_out / 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: 80 MHz

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 🕮).

dig_<ChannelNumber>_oe / Enable Controls the hardware output. If set to 1, the hardware output reacts to the Data outport. Otherwise, it is set to High-Z.

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

Related topics

References

Digital In (Type B) 473
Digital Out (Type B) 478
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40

Digital Out (Type B)

Purpose

To write data to a digital output signal in the FPGA application by using a Digital In/Out 8 channel.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the Digital Out (Type B) I/O functions. There are eight digital output channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 42 … 49.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

IOProperties.Out.Fct(42).Parameter(1).Init / High supply Lets you select the voltage for the high-side switch for all digital output channels.

- 0: 3.3 V
- 1: 5 V

Note

You can specify the high supply voltage value only globally for all digital output channels.
The I/O function number must be specified as 42.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 07.

bidir_<ChannelNumber>_out / 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: 80 MHz

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 🕮).

bidir_<ChannelNumber>_oe / 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

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

Related topics

References

Digital In (Type B)..... 473
Digital Out (Type A)..... 475

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Inductive Zero Voltage Detector

Purpose

To provide read access to an inductive zero voltage detector.

Description

The FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides one channel for the Inductive Zero Voltage Detector I/O function.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 33.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

crank_0 / 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 one clock cycle.

Data type: UFix_1_0

- 0: No zero crossing.
- 1: Zero crossing is detected.

Update rate: 80 MHz

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 -

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Interrupt

Purpose

To request a processor interrupt outside of the FPGA application.

Description

The FPGA1403Tp1 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.

If you generate the processor interface model for this FPGA I/O function, a PROC_INT_BL block is added to the processor model with the configured data formats.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 8.

IRQProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 0 … 7.

usr_<ChannelNumber>_interrupt / Int Provides the interrupt request line.

- 0 to 1: Interrupt is requested (edge-triggered).
- 0: No interrupt is requested. Last requested interrupt is saved.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Knock Sensor

Purpose

To read data from a knock sensor in the FPGA application.

Description

According to the number of physical connections available on the DS1554 Engine Control I/O Module, you can select the Knock Sensor I/O functions. There are 4 knock sensor input channels.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 24 … 27.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 00 … 03.

knock_<ChannelNumber>_data / Data Outputs the current results of the A/D conversions on the current channel.

Data type: UFix_16_0

Range: 0 … +65535

Input voltage range: –5 V ... +5 V

Update rate: 1 Msps

knock_<ChannelNumber>_soc / Start of conversion Triggers the start of an A/D conversion on the specified channel. 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 `End of conversion` outport signals the end of the conversion process.

Setting this value permanently to 1 results in continuous sampling.

Data type: UFix_1_0

knock_<ChannelNumber>_eoc / End of conversion Outputs an end of conversion signal if the conversion result is available on the specified channel. 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

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 -

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

LED Out

Purpose

To write a digital signal that controls the FPGA status LED.

You can find the FPGA status LED near the DS1514 ZIF I/O connector.

Description

The FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides one channel for the LED Out I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

There is no channel number to be specified.

led_out / Data Controls the FPGA status LED.

Data type: UFix_1_0

- 0: LED lights up green.
- 1: LED lights up orange.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Register In

Purpose

To read data from an intermodule-bus register with a data width of 32 bits.

Description

If you select Register 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. 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the .

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_dout / Data Outputs a 32-bit data value to be read from an intermodule-bus register. The data format depends on the related parameter settings.

xreg_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer In 465
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register Out 487

Register64 In

Purpose

To read data from an intermodule-bus register with a data width of 64 bits.

Description

If you select Register64 as the access type, the data is read from an intermodule-bus register. 128 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 … 288.

PHSProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data outport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data outport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data outport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.

- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.In.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 within the FPGA application. The values therefore form a consistent data group that is written to the .

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg64_<ChannelNumber>_dout / Data Outputs a 64-bit data value to be read from an intermodule-bus register. The data format depends on the related parameter settings.

xreg64_<ChannelNumber>_dout_wr / Data New Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then back to 0, the requested register contains a new value. The flag is set to 1 only within one clock cycle.

Related topics

References

Buffer64 In 466
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register64 Out 489

Register Out

Purpose

To write data to an intermodule-bus register with a data width of 32 bits.

Description

If you select Register 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. 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 1 … 128.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- 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.

  0 represents the lowest bit position, 32 the highest bit position.
- floating-point

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / 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.
- 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 fraction width is provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg_<ChannelNumber>_din / Data Specifies a 32-bit data value to be written to an intermodule-bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer Out 468
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1) 40
Register In 484

Register64 Out

Purpose

To write data to an intermodule-bus register with a data width of 64 bits.

Description

If you select Register64 as the access type, the data is written to an intermodule-bus register. 128 registers are available 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.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number can be specified in the range 161 … 288.

PHSProperties.Out.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(1).Init / Binary point position (or fraction width) Lets you specify the binary point position or returns the fraction width of the Data inport depending on the format selected in the Format setting (see below).

- signed/unsigned

  The values of the Data inport are in fixed-point format. You can specify the binary point position of the 64-bit value in the range 0 … 64.

  0 represents the lowest bit position, 64 the highest bit 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

  The values of the Data inport are in floating-point format. The parameter then provides the fraction width.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(2).Init / Format Lets you specify 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.
- floating-point

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

  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 provided by the Binary point position (or fraction width) setting.

PHSProperties.Out.Fct(<IOFunctionNumber>).Parameter(3).Init / 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 sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

- 0: Ungrouped access (default)
- 1: Register group 1
- …
- 63: Register group 63

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

The channel number can be specified in the range 000 … 127.

xreg64_<ChannelNumber>_din / Data Specifies a 64-bit data value to be written to an intermodule-bus register. The data format depends on the related parameter settings.

Related topics

References

Buffer64 Out....470
Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)....40
Register64 In....486

Status In

Purpose

To read digital signals that output state information, e.g., the state of the FPGA initialization sequence.

Description

The FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides one channel for the **Status In** I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 1.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

init_done/ Init Done Outputs the state of the initialization sequence that is started after programming the FPGA.

- 0: Initialization sequence is in progress.
- 1: Initialization sequence has finished.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

Temperature

Purpose

To read the FPGA die temperature.

Description

The FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides one channel for the **Temperature** I/O function.

This I/O function is not considered when you generate the processor interface model.

Parameters

You can find templates for the functions and the following parameters in the handcode FPGA framework INI file.

The I/O function number must be specified as 34.

IOProperties.In.Fct(<IOFunctionNumber>).HcCustomName / Channel name Lets you specify a custom name for the specified channel.

Port

The following signals of the I/O function can be found in the port definition of the custom module entity `cm`.

temperature / Temperature 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_temp / 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.

Related topics

References

Overview of the Frameworks Available for MicroAutoBox III (FPGA1403Tp1)..... 40

A

ADC (DS5203) 293
  I/O mapping 294
  parameters 293
  port 293
ADC (Type A) (FPGA1403TP1)
  DS1554 462
  I/O mapping
    DS1552 417
    DS1554 463
  parameters
    DS1552 417
    DS1554 463
  port
    DS1552 417
    DS1554 463
ADC (Type B) (FPGA1403TP1)
  I/O mapping 419
  parameters 418
  port 419
ADC (Type B) (FPGA1403TP3)
  DS1552 418
ADC Class 1 (DS1302) 46
  I/O mapping 47
  parameters 47
  port 47
ADC Class 2 (DS1302) 48
  I/O mapping 49
  parameters 48
  port 48
ADC M1 (DS5203M1) 326
  I/O mapping 327
  parameters 326
  port 326
ADC Type A (FPGA1401TP1)
  DS1552 337
  DS1554 382
  I/O mapping
    DS1552 337
    DS1554 383
  parameters
    DS1552 337
    DS1554 383
  port
    DS1552 337
    DS1554 383
ADC Type A (FPGA1403TP1)
  DS1552 417
ADC Type B (FPGA1401TP1) 338
  I/O mapping 339
  parameters 339
  port 339
adc<ChannelNumber>_data
  ADC (DS5203) 293
adc<ChannelNumber>_valid
  ADC (DS5203) 294
adctp1_<ChannelNumber>_busy
  ADC Class 1 (DS1302) 47
adctp1_<ChannelNumber>_convert
  ADC Class 1 (DS1302) 47
adctp1_<ChannelNumber>_value
  ADC Class 1 (DS1302) 47
adctp2_<ChannelNumber>_value
  ADC Class 2 (DS1302) 49
Add internal pipeline register to relax timing
  I-FPGA In (SCALEXIO) 285
  I-FPGA Slave (DS5203) 313
Address
  Buffer In (DS1302) 51
  Buffer In (DS2655) 94
  Buffer In (DS5203) 298
  Buffer In (DS6601) 130
  Buffer In (DS6602) 166
  Buffer In (FPGA1401TP1)
    DS1552 341
    DS1554 386
  Buffer In (FPGA1403TP1)
    DS1552 421
    DS1554 466
  Buffer64 In (DS1302) 52
  Buffer64 In (DS2655) 97
  Buffer64 In (DS5203) 300
  Buffer64 In (DS6601) 132
  Buffer64 In (DS6602) 169
  Buffer64 In (FPGA1401TP1)
    DS1552 343
    DS1554 388
  Buffer64 In (FPGA1403TP1)
    DS1552 423
    DS1554 468
  DDR4 32 Mode 1 177
  DDR4 64 Mode 1 182
  I-FPGA In (IOCNET) (DS2655) 104
  I-FPGA In (IOCNET) (DS6601) 140
  I-FPGA In (IOCNET) (DS6602) 187
  I-FPGA64 In (IOCNET) (DS2655) 106
  I-FPGA64 In (IOCNET) (DS6601) 142
  I-FPGA64 In (IOCNET) (DS6602) 188
Address A
  DDR4 32 Mode 2 179
  DDR4 64 Mode 2 184
Address B
  DDR4 32 Mode 2 179
  DDR4 64 Mode 2 184
Analog In (DS2655M1) 206
  I/O mapping 207
  parameters 206
  port 207
Analog In (DS6651) 240
  I/O mapping 243
  parameters 241
  ports 242
Analog In 10 (FPGA1403TP1)
  DS1552 417
Analog In 11 (FPGA1403TP1)
  DS1552B1 417
Analog In 12 (FPGA1403TP1)
  DS1552 418
Analog In 14 (FPGA1403TP1)
  DS1554 462
Analog In 15 (FPGA1403TP1)
  DS1554 482
Analog In-L (DS6651) 244
  I/O mapping 246
  parameters 244
  ports 246
Analog Out (DS2655M1) 208
  I/O mapping 209
  parameters 208
  port 209
Analog Out (DS6651) 247
  I/O mapping 247
  parameters 247
  port 247
Analog Out 13 (FPGA1403TP1)
  DS1552 427
Analog Out-T (DS6651) 248
  I/O mapping 249
  parameters 248
  port 249
Angle range
  APU Slave (DS2655) 92
  APU Slave (DS6601) 127, 164
Angle Range
  APU Slave (DS2655) 93
  APU Slave (DS6601) 128, 165
appl_run
  Proc App Status (DS1302) 67
APU Master (DS2655) 89
  I/O mapping 91
  parameters 89
  port 89
APU Master (DS5203) 294
  I/O mapping 295
  parameters 295
  port 295
APU Master (DS6601) 125
  I/O mapping 127
  parameters 125
  port 125
APU Master (DS6602) 161
  I/O mapping 163
  parameters 161
  port 162
APU Slave (DS2655) 91
  Angle range 92
  I/O mapping 93
  parameters 91
  port 92
APU Slave (DS5203) 295
  I/O mapping 296
  parameters 296
  port 296
APU Slave (DS6601) 127, 163
  Angle range 127, 164
  I/O mapping 129, 165
  parameters 127, 163
  port 128, 164
apu_phi_nd
  APU Slave (DS5203) 296
apu_phi_rd
  APU Slave (DS5203) 296

apu_phi_wr
APU Master (DS5203) 295
Aurora 64b66b 128 Bit In
DS6601 273
DS6602 273
I/O mapping 274
parameters 274
port 274
Aurora 64b66b 128 Bit Out
DS6601 275
DS6602 275
I/O mapping 276
parameters 275
port 276
Aurora 64b66b In
DS6601 270
DS6602 270
I/O mapping 271
parameters 270
port 270
Aurora 64b66b Out
DS6601 271
DS6602 271
I/O mapping 273
parameters 272
port 272

B

Baud rate
UART RS232 (DS1302) 78
UART RS232 (FPGA1401TP1) 371
UART RS422/485 (DS1302) 83
UART RS422/485 (FPGA1401TP1) 376
UART RS422/485 (FPGA1403TP1) 455
Beep Count
Buzzer (DS1302) 58
Beep Duration
Buzzer (DS1302) 57
bidir_<ChannelNumber>_in
Digital In (Type B) (FPGA1403TP1)
DS1552 433
DS1554 474
Digital In Type B (FPGA1401TP1)
DS1552 353
DS1554 395
bidir_<ChannelNumber>_oe
Digital Out (Type B) (FPGA1403TP1)
DS1552 437
DS1554 479
Digital Out Type B (FPGA1401TP1)
DS1552 357
DS1554 400
bidir_<ChannelNumber>_out
Digital Out (Type B) (FPGA1403TP1)
DS1552 437
DS1554 479
Digital Out Type B (FPGA1401TP1)
DS1552 357
DS1554 400
Binary point position
Buffer In (DS1302) 50
Buffer In (DS2655) 94
Buffer In (DS5203) 297
Buffer In (DS6601) 129
Buffer In (DS6602) 166
Buffer In (FPGA1401TP1)
DS1552 340
DS1554 385
Buffer In (FPGA1403TP1)
DS1552 420
DS1554 465
Buffer Out (DS1302) 53
Buffer Out (DS2655) 98
Buffer Out (DS5203) 300
Buffer Out (DS6601) 134
Buffer Out (DS6602) 170
Buffer Out (FPGA1401TP1)
DS1552 344
DS1554 389
Buffer Out (FPGA1403TP1)
DS1552 423
DS1554 468
Buffer64 In (DS1302) 51
Buffer64 In (DS2655) 96
Buffer64 In (DS5203) 299
Buffer64 In (DS6601) 131
Buffer64 In (DS6602) 168
Buffer64 In (FPGA1401TP1)
DS1552 342
DS1554 387
Buffer64 In (FPGA1403TP1)
DS1552 422
DS1554 467
Buffer64 Out (DS1302) 55
Buffer64 Out (DS2655) 100
Buffer64 Out (DS5203) 302
Buffer64 Out (DS6601) 136
Buffer64 Out (DS6602) 172
Buffer64 Out (FPGA1401TP1)
DS1552 346
DS1554 391
Buffer64 Out (FPGA1403TP1)
DS1552 425
DS1554 470
Register In (DS1302) 68
Register In (DS2655) 114
Register In (DS5203) 317
Register In (DS6601) 150
Register In (DS6602) 196
Register In (FPGA1401TP1)
DS1552 361
DS1554 405
Register In (FPGA1403TP1)
DS1552 441
DS1554 484
Register Out (DS1302) 71
Register Out (DS2655) 117
Register Out (DS5203) 320
Register Out (DS6601) 153
Register Out (DS6602) 199
Register Out (FPGA1401TP1)
DS1552 365
DS1554 409
Register Out (FPGA1403TP1)
DS1552 445
DS1554 488
Register64 In (DS1302) 69
Register64 In (DS2655) 116
Register64 In (DS5203) 318
Register64 In (DS6601) 152
Register64 In (DS6602) 198
Register64 In (FPGA1401TP1)
DS1552 363
DS1554 407
Register64 In (FPGA1403TP1)
DS1552 443
DS1554 486
Register64 Out (DS1302) 73
Register64 Out (DS2655) 119
Register64 Out (DS5203) 322
Register64 Out (DS6601) 155
Register64 Out (DS6602) 201
Register64 Out (FPGA1401TP1)
DS1552 366
DS1554 411
Register64 Out (FPGA1403TP1)
DS1552 446
DS1554 489
Bit length
I-FPGA In (SCALEXIO) 284
I-FPGA Master (DS5203) 310
I-FPGA Out (SCALEXIO) 288
I-FPGA Slave (DS5203) 312
Blue
LED Out (DS1302) 66
Buffer In (DS1302) 49
parameters 49
port 50
Buffer In (DS2655) 93
parameters 93
port 94
Buffer In (DS5203) 297
parameters 297
port 297
Buffer In (DS6601) 129
parameters 129
port 130
Buffer In (DS6602) 165
parameters 165
port 166
Buffer In (FPGA1401TP1)
DS1552 340
DS1554 385
parameters
DS1552 340
DS1554 385
port
DS1552 341, 343
DS1554 386, 387
Buffer In (FPGA1403TP1)
DS1552 420
DS1554 465
parameters

DS1552 420
DS1554 465
port
DS1552 420, 422
DS1554 466, 467
Buffer Out (DS1302) 53
parameters 53
port 54
Buffer Out (DS2655) 98
parameters 98
port 99
Buffer Out (DS5203) 300
parameters 300
port 301
Buffer Out (DS6601) 133
parameters 134
port 134
Buffer Out (DS6602) 170
parameters 170
port 171
Buffer Out (FPGA1401TP1)
DS1552 343
DS1554 388
parameters
DS1552 344
DS1554 388
port
DS1552 344
DS1554 389
Buffer Out (FPGA1403TP1)
DS1552 423
DS1554 468
parameters
DS1552 423
DS1554 468
port
DS1552 424
DS1554 469
Buffer size
Buffer In (DS1302) 50
Buffer In (DS2655) 94
Buffer In (DS5203) 297
Buffer In (DS6601) 130
Buffer In (DS6602) 166
Buffer In (FPGA1401TP1)
DS1552 341
DS1554 386
Buffer In (FPGA1403TP1)
DS1552 420
DS1554 465
Buffer Out (DS1302) 54
Buffer Out (DS2655) 99
Buffer Out (DS5203) 301
Buffer Out (DS6601) 134
Buffer Out (DS6602) 171
Buffer Out (FPGA1401TP1)
DS1552 344
DS1554 389
Buffer Out (FPGA1403TP1)
DS1552 424
DS1554 469
Buffer64 In (DS1302) 52
Buffer64 In (DS2655) 96
Buffer64 In (DS5203) 299
Buffer64 In (DS6601) 132
Buffer64 In (DS6602) 168
Buffer64 Out (DS1302) 55
Buffer64 Out (DS2655) 101
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 137
Buffer64 Out (DS6602) 173
Buffer64 Out (FPGA1401TP1)
DS1552 346
DS1554 391
Buffer64 Out (FPGA1403TP1)
DS1552 426
DS1554 471
I-FPGA In (IOCNET) (DS2655) 104
I-FPGA In (IOCNET) (DS6601) 140
I-FPGA In (IOCNET) (DS6602) 186
I-FPGA Out (IOCNET) (DS2655) 107
I-FPGA Out (IOCNET) (DS6601) 143
I-FPGA Out (IOCNET) (DS6602) 190
I-FPGA64 In (IOCNET) (DS2655) 106
I-FPGA64 In (IOCNET) (DS6601) 142
I-FPGA64 In (IOCNET) (DS6602) 188
I-FPGA64 Out (IOCNET) (DS2655) 110
I-FPGA64 Out (IOCNET) (DS6601) 146
I-FPGA64 Out (IOCNET) (DS6602) 192
Buffer64 In (DS1302) 51
parameters 51
port 52
Buffer64 In (DS2655) 95
parameters 96
port 96
Buffer64 In (DS5203) 298
parameters 298
port 299
Buffer64 In (DS6601) 131
parameters 131
port 132
Buffer64 In (DS6602) 167
parameters 168
port 168
Buffer64 In (FPGA1401TP1)
DS1552 341
DS1554 386
parameters
DS1552 342
DS1554 387
Buffer64 In (FPGA1403TP1)
DS1552 421
DS1554 466
parameters
DS1552 422
DS1554 467
Buffer64 Out (DS1302) 55
parameters 55
port 56
Buffer64 Out (DS2655) 100
parameters 100
port 101
Buffer64 Out (DS5203) 302
parameters 302
port 303
Buffer64 Out (DS6601) 136
parameters 136
port 137
Buffer64 Out (DS6602) 172
parameters 172
port 173
Buffer64 Out (FPGA1401TP1)
DS1552 345
DS1554 390
parameters
DS1552 346
DS1554 390
port
DS1552 346
DS1554 391
Buffer64 Out (FPGA1403TP1)
DS1552 425
DS1554 470
parameters
DS1552 425
DS1554 470
port
DS1552 426
DS1554 471
Buffer64 size
Buffer64 In (FPGA1401TP1)
DS1552 342
DS1554 387
Buffer64 In (FPGA1403TP1)
DS1552 422
DS1554 467
Busy
ADC Class 1 (DS1302) 47
DAC Class 1 (DS1302) 59
DDR4 32 Mode 1 177
DDR4 32 Mode 2 179
DDR4 64 Mode 1 182
DDR4 64 Mode 2 184
Buzzer (DS1302) 57
parameters 57
port 57
buzzer start
Buzzer (DS1302) 58
buzzer_beep_count
Buzzer (DS1302) 58
buzzer_beep_duration
Buzzer (DS1302) 57
buzzer_frequency
Buzzer (DS1302) 57
buzzer_pause_duration
Buzzer (DS1302) 57

C

cam_<ChannelNumber>
Digital Crank/Cam Sensor (FPGA1401TP1)
DS1552 349
DS1554 393
Digital Crank/Cam Sensor (FPGA1403TP1)

DS1552 429
DS1554 473
Channel name
ADC (DS5203) 293
ADC (Type A) (FPGA1403TP1)
DS1552 417
DS1554 463
ADC (Type B) (FPGA1403TP1) 418
ADC Class 1 (DS1302) 47
ADC Class 2 (DS1302) 48
ADC M1 (DS5203M1) 326
ADC Type A (FPGA1401TP1)
DS1552 337
DS1554 383
ADC Type B (FPGA1401TP1) 339
Analog In (DS2655M1) 206
Analog In (DS6651) 241
Analog In-L (DS6651) 244
Analog Out (DS2655M1) 208
Analog Out (DS6651) 247
Analog Out-T (DS6651) 248
APU Master (DS2655) 89
APU Master (DS5203) 295
APU Master (DS6601) 125
APU Master (DS6602) 161
APU Slave (DS2655) 91
APU Slave (DS5203) 296
APU Slave (DS6601) 127, 163
Buffer In (DS1302) 49
Buffer In (DS2655) 93
Buffer In (DS5203) 297
Buffer In (DS6601) 129
Buffer In (DS6602) 165
Buffer In (FPGA1401TP1)
DS1552 340
DS1554 385
Buffer In (FPGA1403TP1)
DS1552 420
DS1554 465
Buffer Out (DS1302) 53
Buffer Out (DS2655) 98
Buffer Out (DS5203) 300
Buffer Out (DS6601) 134
Buffer Out (DS6602) 170
Buffer Out (FPGA1401TP1)
DS1552 344
DS1554 388
Buffer Out (FPGA1403TP1)
DS1552 423
DS1554 468
Buffer64 In (DS1302) 51
Buffer64 In (DS2655) 96
Buffer64 In (DS5203) 298
Buffer64 In (DS6601) 131
Buffer64 In (DS6602) 168
Buffer64 In (FPGA1401TP1)
DS1552 342
DS1554 387
Buffer64 In (FPGA1403TP1)
DS1552 422
DS1554 467
Buffer64 Out (DS1302) 55
Buffer64 Out (DS2655) 100
Buffer64 Out (DS5203) 302
Buffer64 Out (DS6601) 136
Buffer64 Out (DS6602) 172
Buffer64 Out (FPGA1401TP1)
DS1552 346
DS1554 390
Buffer64 Out (FPGA1403TP1)
DS1552 425
DS1554 470
Buzzer (DS1302) 57
CN App Status (DS2655) 103
CN App Status (DS6601) 139
CN App Status (DS6602) 175
DAC (DS5203) 304
DAC (FPGA1401TP1) 348
DAC (FPGA1403TP1) 427
DAC Class 1 (DS1302) 58
DAC M1 (DS5203M1) 328
Digital Crank/Cam Sensor (FPGA1401TP1)
DS1552 349
DS1554 392
Digital Crank/Cam Sensor (FPGA1403TP1)
DS1552 428
DS1554 472
Digital In (DS2655M1) 210
Digital In (DS2655M2) 219
Digital In (DS5203) 306
Digital In (DS6651) 250
Digital In (Type A) (FPGA1403TP1) 430
Digital In (Type B) (FPGA1403TP1)
DS1552 432
DS1554 474
Digital In M1 (DS5203M1) 329
Digital In Type A (FPGA1401TP1) 351
Digital In Type B (FPGA1401TP1)
DS1552 352
DS1554 394
Digital In/Out-Z (DS6651) 252
Digital InOut (DS2655M1) 212
Digital InOut Class 1 (DS1302) 60
Digital InOut Class 2 (DS1302) 62
Digital Out (DS2655M1) 214
Digital Out (DS2655M2) 221
Digital Out (DS5203) 308
Digital Out (DS6651) 255
Digital Out (Type A) (FPGA1403TP1)
DS1552 434
DS1554 476
Digital Out (Type B) (FPGA1403TP1)
DS1552 436
DS1554 478
Digital Out M1 (DS5203M1) 332
Digital Out Type A (FPGA1401TP1)
DS1552 354
DS1554 396
Digital Out Type B (FPGA1401TP1)
DS1552 356
DS1554 399
Digital Out-Z (DS2655M2) 224
Digital Out-Z (DS6651) 258
I-FPGA In (IOCNET) (DS2655) 104
I-FPGA In (IOCNET) (DS6601) 139
I-FPGA In (IOCNET) (DS6602) 186
I-FPGA In (SCALEXIO) 283
I-FPGA Master (DS5203) 310
I-FPGA Out (IOCNET) (DS2655) 107
I-FPGA Out (IOCNET) (DS6601) 143
I-FPGA Out (IOCNET) (DS6602) 189
I-FPGA Out (SCALEXIO) 287
I-FPGA Slave (DS5203) 312
I-FPGA64 In (IOCNET) (DS2655) 105
I-FPGA64 In (IOCNET) (DS6601) 141
I-FPGA64 In (IOCNET) (DS6602) 188
I-FPGA64 Out (IOCNET) (DS2655) 109
I-FPGA64 Out (IOCNET) (DS6601) 145
I-FPGA64 Out (IOCNET) (DS6602) 192
Inductive Zero Voltage Detector (FPGA1401TP1)
DS1552 358
DS1554 401
Inductive Zero Voltage Detector (FPGA1403TP1)
DS1552 438
DS1554 480
Interrupt (DS1302) 65
Interrupt (DS2655) 112
Interrupt (DS5203) 315
Interrupt (DS6601) 148
Interrupt (DS6602) 194
Interrupt (FPGA1401TP1)
DS1552 359
DS1554 402
Interrupt (FPGA1403TP1)
DS1552 439
DS1554 481
IOCNET Global Time (DS2655) 112
IOCNET Global Time (DS6601) 148
IOCNET Global Time (DS6602) 194
Knock Sensor (FPGA1401TP1) 403
Knock Sensor (FPGA1403TP1) 482
LED Out (DS1302) 66
LED Out (DS2655) 113
LED Out (DS5203) 316
LED Out (DS6601) 149
LED Out (DS6602) 195
LED Out (FPGA1401TP1)
DS1552 360
DS1554 404
LED Out (FPGA1403TP1)
DS1552 440
DS1554 483
Proc App Status (DS1302) 67
Register In (DS1302) 67
Register In (DS2655) 114
Register In (DS5203) 316
Register In (DS6601) 150
Register In (DS6602) 196
Register In (FPGA1401TP1)
DS1552 361
DS1554 405

Register In (FPGA1403TP1)
DS1552 441
DS1554 484
Register Out (DS1302) 71
Register Out (DS2655) 117
Register Out (DS5203) 320
Register Out (DS6601) 153
Register Out (DS6602) 199
Register Out (FPGA1401TP1)
DS1552 365
DS1554 409
Register Out (FPGA1403TP1)
DS1552 444
DS1554 488
Register64 In (DS1302) 69
Register64 In (DS2655) 115
Register64 In (DS5203) 318
Register64 In (DS6601) 151
Register64 In (DS6602) 198
Register64 In (FPGA1401TP1)
DS1552 363
DS1554 407
Register64 In (FPGA1403TP1)
DS1552 443
DS1554 486
Register64 Out (DS1302) 72
Register64 Out (DS2655) 119
Register64 Out (DS5203) 321
Register64 Out (DS6601) 155
Register64 Out (DS6602) 201
Register64 Out (FPGA1401TP1)
DS1552 366
DS1554 410
Register64 Out (FPGA1403TP1)
DS1552 446
DS1554 489
Resolver (DS1302) 74
RS232 Rx (DS2655M2) 227
RS232 Tx (DS2655M2) 229
RS485 Rx (DS2655M2) 230
RS485 Rx (DS6651) 261
RS485 Rx/Tx (DS6651) 263
RS485 RxTx (DS2655M2) 232
RS485 Tx (DS2655M2) 235
RS485 Tx (DS6651) 265
Sensor Supply (FPGA1401TP1)
DS1552 368
Sensor Supply (FPGA1403TP1)
DS1552 448
Sensor Supply M1 (DS5203M1) 334
Status In (DS1302) 77
Status In (DS2655) 120
Status In (DS5203) 323
Status In (DS6601) 156
Status In (DS6602) 202
Status In (FPGA1401TP1)
DS1552 369
DS1554 412
Status In (FPGA1403TP1)
DS1552 449
DS1554 491
Temperature (FPGA1401TP1) 413
Temperature (FPGA1403TP1) 491
Trigger (DS6651) 268
UART 3 RS232 (FPGA1403TP1) 450
UART 3 RS422/485 (FPGA1403TP1) 455
UART RS232 (DS1302) 78
UART RS232 (FPGA1401TP1) 370
UART RS422/485 (DS1302) 82
UART RS422/485 (FPGA1401TP1) 376
CLK_N
MGT In 278
CLK_P
MGT In 277
Clock
I-FPGA In (SCALEXIO) 284
I-FPGA Master (DS5203) 310
I-FPGA Out (SCALEXIO) 289
I-FPGA Slave (DS5203) 313
CN App Status (DS2655) 103
parameters 103
port 103
CN App Status (DS6601) 138
parameters 139
port 139
CN App Status (DS6602) 175
parameters 175
port 175
CN Application Status
CN App Status (DS2655) 103, 113
CN App Status (DS6601) 139, 149
CN App Status (DS6602) 175, 195
Common Program Data folder 12
Convert
ADC Class 1 (DS1302) 47
DAC Class 1 (DS1302) 59
crank
Inductive Zero Voltage Detector (FPGA1401TP1)
DS1552 358
DS1554 401
Inductive Zero Voltage Detector (FPGA1403TP1)
DS1552 438
DS1554 480
CTS
UART 3 RS232 (FPGA1403TP1) 454
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374

D

DAC (DS5203) 304
I/O mapping 305
parameters 304
port 304
DAC (FPGA1401TP1) 347
I/O mapping 348
parameters 348
port 348
DAC (FPGA1403TP1)
DS1552 427
I/O mapping 428
parameters 427
port 427
DAC Class 1 (DS1302) 58
I/O mapping 59
parameters 58
port 59
DAC M1 (DS5203M1) 327
I/O mapping 329
parameters 328
port 328
dac<ChannelNumber>_data
DAC (FPGA1401TP1) 348
DAC (FPGA1403TP1) 427
dactp1_<ChannelNumber>_busy
DAC Class 1 (DS1302) 59
dactp1_<ChannelNumber>_convert
DAC Class 1 (DS1302) 59
dactp1_<ChannelNumber>_value
DAC Class 1 (DS1302) 59
dacxx_data
DAC (DS5203) 305
Data
ADC (DS5203) 293
ADC (Type A) (FPGA1403TP1)
DS1552 417
DS1554 463
ADC (Type B) (FPGA1403TP1) 419
ADC Class 1 (DS1302) 47
ADC Class 2 (DS1302) 49
ADC M1 (DS5203M1) 326
ADC Type A (FPGA1401TP1)
DS1552 337
DS1554 383
ADC Type B (FPGA1401TP1) 339
Analog In (DS2655M1) 207
Analog In (DS6651) 242
Analog In-L (DS6651) 246
Analog Out (DS2655M1) 209
Analog Out (DS6651) 247
Analog Out-T (DS6651) 249
Aurora 64b66b 128 Bit In 274
Aurora 64b66b 128 Bit Out 276
Aurora 64b66b In 270
Aurora 64b66b Out 272
Buffer In (DS1302) 50
Buffer In (DS2655) 94
Buffer In (DS5203) 298
Buffer In (DS6601) 130
Buffer In (DS6602) 166
Buffer In (FPGA1401TP1)
DS1552 341
DS1554 386
Buffer In (FPGA1403TP1)
DS1552 421
DS1554 466
Buffer Out (DS1302) 54
Buffer Out (DS2655) 99
Buffer Out (DS5203) 301
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171
Buffer Out (FPGA1401TP1)

DS1552 344
DS1554 389
Buffer Out (FPGA1403TP1)
DS1552 424
DS1554 469
Buffer64 In (DS1302) 52
Buffer64 In (DS2655) 97
Buffer64 In (DS5203) 299
Buffer64 In (DS6601) 132
Buffer64 In (DS6602) 169
Buffer64 In (FPGA1401TP1)
DS1552 343
DS1554 388
Buffer64 In (FPGA1403TP1)
DS1552 423
DS1554 468
Buffer64 Out (DS1302) 56
Buffer64 Out (DS2655) 101
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 137
Buffer64 Out (DS6602) 173
Buffer64 Out (FPGA1401TP1)
DS1552 346
DS1554 391
Buffer64 Out (FPGA1403TP1)
DS1552 426
DS1554 471
DAC (DS5203) 305
DAC (FPGA1401TP1) 348
DAC (FPGA1403TP1) 427
DAC Class 1 (DS1302) 59
DAC M1 (DS5203M1) 328
Digital Crank/Cam Sensor (FPGA1401TP1)
DS1552 349
DS1554 393
Digital Crank/Cam Sensor (FPGA1403TP1)
DS1552 429
DS1554 473
Digital In (DS2655M1) 210
Digital In (DS2655M2) 219
Digital In (DS5203) 306
Digital In (DS6651) 250
Digital In (Type A) (FPGA1403TP1) 431
Digital In (Type B) (FPGA1403TP1)
DS1552 433
DS1554 474
Digital In M1 (DS5203M1) 330
Digital In Type A (FPGA1401TP1) 351
Digital In Type B (FPGA1401TP1)
DS1552 353
DS1554 395
Digital Out (DS2655M1) 215
Digital Out (DS2655M2) 222
Digital Out (DS5203) 308
Digital Out (DS6651) 256
Digital Out (Type A) (FPGA1403TP1)
DS1552 435
DS1554 476
Digital Out (Type B) (FPGA1403TP1)
DS1552 437
DS1554 479
Digital Out M1 (DS5203M1) 332
Digital Out Type A (FPGA1401TP1)
DS1552 355
DS1554 396
Digital Out Type B (FPGA1401TP1)
DS1552 357
DS1554 400
Digital Out-Z (DS2655M2) 225
Digital Out-Z (DS6651) 259
I-FPGA In (IOCNET) (DS2655) 105
I-FPGA In (IOCNET) (DS6601) 140
I-FPGA In (IOCNET) (DS6602) 187
I-FPGA Out (IOCNET) (DS2655) 108
I-FPGA Out (IOCNET) (DS6601) 144
I-FPGA Out (IOCNET) (DS6602) 190
I-FPGA64 In (IOCNET) (DS2655) 106
I-FPGA64 In (IOCNET) (DS6601) 142
I-FPGA64 In (IOCNET) (DS6602) 189
I-FPGA64 Out (IOCNET) (DS2655) 110
I-FPGA64 Out (IOCNET) (DS6601) 146
I-FPGA64 Out (IOCNET) (DS6602) 192
Inductive Zero Voltage Detector (FPGA1401TP1)
DS1552 358
DS1554 401
Inductive Zero Voltage Detector (FPGA1403TP1)
DS1552 438
DS1554 480
Knock Sensor (FPGA1401TP1) 403
Knock Sensor (FPGA1403TP1) 482
LED Out (DS5203) 316
LED Out (FPGA1401TP1)
DS1552 360
DS1554 405
LED Out (FPGA1403TP1)
DS1552 440
DS1554 484
Register In (DS1302) 69
Register In (DS2655) 115
Register In (DS5203) 317
Register In (DS6601) 151
Register In (DS6602) 197
Register In (FPGA1401TP1)
DS1552 362
DS1554 406
Register In (FPGA1403TP1)
DS1552 442
DS1554 485
Register Out (DS1302) 72
Register Out (DS2655) 118
Register Out (DS5203) 321
Register Out (DS6601) 154
Register Out (DS6602) 200
Register Out (FPGA1401TP1)
DS1552 366
DS1554 410
Register Out (FPGA1403TP1)
DS1552 445
DS1554 489
Register64 In (DS1302) 70
Register64 In (DS2655) 117
Register64 In (DS5203) 319
Register64 In (DS6601) 153
Register64 In (DS6602) 199
Register64 In (FPGA1401TP1)
DS1552 364
DS1554 408
Register64 In (FPGA1403TP1)
DS1552 444
DS1554 487
Register64 Out (DS1302) 74
Register64 Out (DS2655) 120
Register64 Out (DS5203) 323
Register64 Out (DS6601) 156
Register64 Out (DS6602) 202
Register64 Out (FPGA1401TP1)
DS1552 367
DS1554 412
Register64 Out (FPGA1403TP1)
DS1552 447
DS1554 490
RS232 Rx (DS2655M2) 227
RS232 Tx (DS2655M2) 229
RS485 Rx (DS2655M2) 230
RS485 Rx (DS6651) 261
RS485 Tx (DS2655M2) 235
RS485 Tx (DS6651) 266
Data (In)
Digital InOut (DS2655M1) 213
Data (Out)
Digital InOut (DS2655M1) 213
Data Count
Buffer In (DS1302) 50
Buffer In (DS2655) 94
Buffer In (DS5203) 298
Buffer In (DS6601) 130
Buffer In (DS6602) 166
Buffer In (FPGA1401TP1)
DS1552 341
DS1554 386
Buffer In (FPGA1403TP1)
DS1552 421
DS1554 466
Buffer64 In (DS1302) 52
Buffer64 In (DS2655) 97
Buffer64 In (DS5203) 299
Buffer64 In (DS6601) 132
Buffer64 In (DS6602) 169
Buffer64 In (FPGA1401TP1)
DS1552 343
DS1554 388
Buffer64 In (FPGA1403TP1)
DS1552 422
DS1554 467
I-FPGA In (IOCNET) (DS2655) 105
I-FPGA In (IOCNET) (DS6601) 140
I-FPGA In (IOCNET) (DS6602) 187
I-FPGA64 In (IOCNET) (DS2655) 106
I-FPGA64 In (IOCNET) (DS6601) 142
I-FPGA64 In (IOCNET) (DS6602) 189
Data direction

Digital InOut (DS2655M1) 213
Data In
Digital In/Out-Z (DS6651) 253
Digital InOut Class 1 (DS1302) 61
Digital InOut Class 2 (DS1302) 63
Data New
Analog In (DS2655M1) 207
Analog In (DS6651) 242
Analog In-L (DS6651) 246
Aurora 64b66b 128 Bit In 274
Aurora 64b66b In 271
Buffer In (DS1302) 51
Buffer In (DS2655) 95
Buffer In (DS5203) 298
Buffer In (DS6601) 130
Buffer In (DS6602) 167
Buffer In (FPGA1401TP1)
DS1552 341
DS1554 386
Buffer In (FPGA1403TP1)
DS1552 421
DS1554 466
Buffer64 In (DS1302) 52
Buffer64 In (DS2655) 97
Buffer64 In (DS5203) 300
Buffer64 In (DS6601) 133
Buffer64 In (DS6602) 169
Buffer64 In (FPGA1401TP1)
DS1552 343
DS1554 388
Buffer64 In (FPGA1403TP1)
DS1552 423
DS1554 468
I-FPGA In (IOCNET) (DS2655) 105
I-FPGA In (IOCNET) (DS6601) 141
I-FPGA In (IOCNET) (DS6602) 187
I-FPGA In (SCALEXIO) 286
I-FPGA Slave (DS5203) 314
I-FPGA64 In (IOCNET) (DS2655) 107
I-FPGA64 In (IOCNET) (DS6601) 143
I-FPGA64 In (IOCNET) (DS6602) 189
Register In (DS1302) 69
Register In (DS2655) 115
Register In (DS5203) 317
Register In (DS6601) 151
Register In (DS6602) 197
Register In (FPGA1401TP1)
DS1552 362
DS1554 406
Register In (FPGA1403TP1)
DS1552 442
DS1554 485
Register64 In (DS1302) 70, 319
Register64 In (DS2655) 117
Register64 In (DS6601) 153
Register64 In (DS6602) 199
Register64 In (FPGA1401TP1)
DS1552 364
DS1554 408
Register64 In (FPGA1403TP1)
DS1552 444
DS1554 487
Data Out
Digital In/Out-Z (DS6651) 253
Digital InOut Class 1 (DS1302) 61
Digital InOut Class 2 (DS1302) 64
Data Read
DDR4 32 Mode 1 177
DDR4 64 Mode 1 182
I-FPGA In (SCALEXIO) 285
I-FPGA Slave (DS5203) 314
Data Read A
DDR4 32 Mode 2 180
DDR4 64 Mode 2 185
Data Read B
DDR4 32 Mode 2 180
DDR4 64 Mode 2 185
Data Sent
I-FPGA Master (DS5203) 311
I-FPGA Master (SCALEXIO) 290
Data valid
DDR4 32 Mode 1 177
DDR4 64 Mode 1 182
Data valid A
DDR4 32 Mode 2 180
DDR4 64 Mode 2 185
Data valid B
DDR4 32 Mode 2 180
DDR4 64 Mode 2 185
Data Write
DDR4 32 Mode 1 177
DDR4 64 Mode 1 182
I-FPGA Master (DS5203) 311
I-FPGA Master (SCALEXIO) 289
Data Write A
DDR4 32 Mode 2 180
DDR4 64 Mode 2 185
Data Write B
DDR4 32 Mode 2 180
DDR4 64 Mode 2 185
DDR4 32 Mode 1
DS6602 176
I/O mapping 177
parameters 176
port 176
DDR4 32 Mode 2
DS6602 178
I/O mapping 180
parameters 178
port 178
DDR4 64 Mode 1
DS6602 181
I/O mapping 183
parameters 181
port 181
DDR4 64 Mode 2
DS6602 183
I/O mapping 185
parameters 183
port 183
ddr4_address_1_block_3
DDR4 32 Mode 2 179
ddr4_address_1_block_4
DDR4 64 Mode 2 184
ddr4_address_2_block_3
DDR4 32 Mode 2 179
ddr4_address_2_block_4
DDR4 64 Mode 2 184
ddr4_address_block_1
DDR4 32 Mode 1 177
ddr4_address_block_2
DDR4 64 Mode 1 182
ddr4_busy_flag
DDR4 32 Mode 1 177
DDR4 32 Mode 2 179
DDR4 64 Mode 1 182
DDR4 64 Mode 2 184
ddr4_data_<PortNumber>_rd_32_block_1
DDR4 32 Mode 1 177
ddr4_data_<PortNumber>_rd_64_block_2
DDR4 64 Mode 1 182
ddr4_data_<PortNumber>_wr_32_block_1
Data Write 177
ddr4_data_<PortNumber>_wr_64_block_2
Data Write 182
ddr4_data_1_<PortNumber>_rd_32_block_3
DDR4 32 Mode 2 180
ddr4_data_1_<PortNumber>_rd_64_block_4
DDR4 64 Mode 2 185
ddr4_data_1_<PortNumber>_wr_32_block_3
DDR4 32 Mode 2 180
ddr4_data_1_<PortNumber>_wr_64_block_4
DDR4 64 Mode 2 185
ddr4_data_2_<PortNumber>_rd_32_block_3
DDR4 32 Mode 2 180
ddr4_data_2_<PortNumber>_rd_64_block_4
DDR4 64 Mode 2 185
ddr4_data_en
DDR4 32 Mode 1 177
DDR4 32 Mode 2 179
DDR4 64 Mode 1 182
DDR4 64 Mode 2 184
ddr4_direction
DDR4 32 Mode 1 177
DDR4 32 Mode 2 179
DDR4 64 Mode 1 182
DDR4 64 Mode 2 184
ddr4_init_done
DDR4 32 Mode 1 176
DDR4 32 Mode 2 178
DDR4 64 Mode 1 181
DDR4 64 Mode 2 183
ddr4_init_fail
DDR4 32 Mode 1 176
DDR4 32 Mode 2 179
DDR4 64 Mode 1 182
DDR4 64 Mode 2 184
ddr4_rd_data_valid_1_block_3
DDR4 32 Mode 2 180
ddr4_rd_data_valid_1_block_4
DDR4 64 Mode 2 185
ddr4_rd_data_valid_2_block_3
DDR4 32 Mode 2 180

ddr4_rd_data_valid_2_block_4
DDR4 64 Mode 2 185
ddr4_rd_data_valid_block_1
DDR4 32 Mode 1 177
ddr4_rd_data_valid_block_2
DDR4 64 Mode 1 182
Delta Phi
APU Master (DS2655) 90
APU Master (DS6601) 126
APU Master (DS6602) 162
APU Slave (DS2655) 93
APU Slave (DS6601) 128, 165
Delta Phi Enable
APU Master (DS2655) 90
APU Master (DS6601) 126
APU Master (DS6602) 162
APU Slave (DS2655) 93
APU Slave (DS6601) 128, 165
Desired excitation frequency
Resolver (DS1302) 74
dig_<ChannelNumber>_in
Digital In (Type A) (FPGA1403TP1) 431
Digital In Type A (FPGA1401TP1) 351
dig_<ChannelNumber>_oe
Digital Out (Type A) (FPGA1403TP1)
DS1552 435
DS1554 476
Digital Out Type A (FPGA1401TP1)
DS1552 355
DS1554 396
dig_<ChannelNumber>_out
Digital Out (Type A) (FPGA1403TP1)
DS1552 435
DS1554 476
Digital Out Type A (FPGA1401TP1)
DS1552 355
DS1554 396
digio_<ChannelNumber>_in
Digital In (DS5203) 306
digio_<ChannelNumber>_oe
Digital Out (DS5203) 308
digio_<ChannelNumber>_out
Digital Out (DS5203) 308
Digital Crank/Cam Sensor (FPGA1401TP1)
DS1552 348
DS1554 392
I/O mapping
DS1552 350
DS1554 393
parameters
DS1552 349
DS1554 392
port
DS1552 349
DS1554 393
Digital Crank/Cam Sensor (FPGA1403TP1)
DS1552 428
DS1554 472
I/O mapping
DS1552 429
DS1554 473
parameters
DS1552 428
DS1554 472
port
DS1552 429
DS1554 473
Digital In (DS2655M1) 210
I/O mapping 210
parameters 210
port 210
Digital In (DS2655M2) 218
I/O mapping 219
parameters 218
port 219
Digital In (DS5203) 305
I/O mapping 307
parameters 306
port 306
Digital In (DS6651) 250
I/O mapping 250
parameters 250
port 250
Digital In (Type A) (FPGA1403TP1)
I/O mapping 431
parameters 430
port 431
Digital In (Type B) (FPGA1403TP1)
DS1552 432
DS1554 473
I/O mapping
DS1552 434
DS1554 475
parameters
DS1552 432
DS1554 474
port
DS1552 433
DS1554 474
Digital In 10 (FPGA1403TP1)
DS1554 480
Digital In 5 (FPGA1403TP1)
DS1552 430
Digital In 6 (FPGA1403TP1)
DS1552 428
Digital In 7 (FPGA1403TP1)
DS1552 438
Digital In 9 (FPGA1403TP1)
DS1554 472
Digital In M1 (DS5203M1) 329
I/O mapping 331
parameters 329
port 330
Digital in threshold init voltage
Digital InOut (DS2655M1) 212
Digital In Type A (FPGA1401TP1) 350
I/O mapping 351
parameters 351
port 351
Digital In Type A (FPGA1403TP1)
DS1552 430
Digital In Type B (FPGA1401TP1)
DS1552 352
DS1554 394
I/O mapping
DS1552 354
DS1554 395
parameters
DS1552 352
DS1554 394
port
DS1552 353
DS1554 395
Digital In/Out 8 (FPGA1403TP1)
DS1554 478
Digital In/Out-Z (DS6651) 251
I/O mapping 254
parameters 252
ports 253
Digital InOut (DS2655M1) 211
I/O mapping 213
parameters 211
port 212
Digital InOut 6 (FPGA1403TP1)
DS1552 432
Digital InOut 6 (Out) (FPGA1403TP1)
DS1552 436
Digital InOut 8 (FPGA1403TP1)
DS1554 473
Digital InOut Class 1 (DS1302) 60
I/O mapping 62
parameters 60
port 60
Digital InOut Class 2 (DS1302) 62
I/O mapping 64
parameters 62
port 63
Digital Out (DS2655M1) 214
I/O mapping 215
parameters 214
port 215
Digital Out (DS2655M2) 221
I/O mapping 222
parameters 221
port 222
Digital Out (DS5203) 307
I/O mapping 309
parameters 308
port 308
Digital Out (DS6651) 255
I/O mapping 256
parameters 255
port 256
Digital Out (Type A) (FPGA1403TP1)
DS1552 434
DS1554 475
I/O mapping
DS1552 435
DS1554 476
parameters
DS1552 434
DS1554 476
port

DS1552 435
DS1554 476
Digital Out (Type B) (FPGA1403TP1)
DS1552 436
DS1554 478
I/O mapping
DS1552 437
DS1554 479
parameters
DS1552 436
DS1554 478
port
DS1552 437
DS1554 479
Digital Out 5 (FPGA1403TP1)
DS1552 434
Digital Out 7 (FPGA1403TP1)
DS1554 475
Digital Out M1 (DS5203M1) 331
I/O mapping 333
parameters 332
port 332
Digital Out Type A (FPGA1401TP1)
DS1552 354
DS1554 396
I/O mapping
DS1552 355
DS1554 397
parameters
DS1552 354
DS1554 396
port
DS1552 355
DS1554 396
Digital Out Type B (FPGA1401TP1)
DS1552 356
DS1554 399
I/O mapping
DS1552 357
DS1554 400
parameters
DS1552 356
DS1554 399
port
DS1552 357
DS1554 400
Digital Out-Z (DS2655M2) 224
I/O mapping 225
parameters 224
port 225
Digital Out-Z (DS6651) 257
I/O mapping 259
parameters 258
ports 258
diotp1_<ChannelNumber>_dir
Digital InOut Class 1 (DS1302) 61
Digital InOut Class 2 (DS1302) 63
diotp1_<ChannelNumber>_ena
Digital InOut Class 1 (DS1302) 61
Digital InOut Class 2 (DS1302) 63
diotp1_<ChannelNumber>_in
Digital InOut Class 1 (DS1302) 61
Digital InOut Class 2 (DS1302) 63
diotp1_<ChannelNumber>_out
Digital InOut Class 1 (DS1302) 61
Digital InOut Class 2 (DS1302) 64
Direction
DDR4 32 Mode 1 177
DDR4 32 Mode 2 179
DDR4 64 Mode 1 182
DDR4 64 Mode 2 184
Digital InOut Class 1 (DS1302) 61
Digital InOut Class 2 (DS1302) 63
Documents folder 12
Drive config
Digital InOut (DS2655M1) 212
Digital Out (DS2655M1) 215
Digital Out (DS2655M2) 221
Digital Out-Z (DS2655M2) 224
Drive Config
Digital In/Out-Z (DS6651) 252
Digital Out (DS6651) 255
Digital Out-Z (DS6651) 258
Driver Enable
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS422/485 (DS1302) 86
UART RS422/485 (FPGA1401TP1) 379

E

Enable
Analog In (DS2655M1) 207
Analog Out (DS2655M1) 209
Aurora 64b66b 128 Bit Out 276
Aurora 64b66b Out 272
Buffer Out (DS1302) 54
Buffer Out (DS2655) 99
Buffer Out (DS5203) 301
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171
Buffer Out (FPGA1401TP1)
DS1552 345
DS1554 389
Buffer Out (FPGA1403TP1)
DS1552 424
DS1554 469
Buffer64 Out (DS1302) 56
Buffer64 Out (DS2655) 102
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 137
Buffer64 Out (DS6602) 174
Buffer64 Out (FPGA1401TP1)
DS1552 347
DS1554 391
Buffer64 Out (FPGA1403TP1)
DS1552 426
DS1554 471
DDR4 32 Mode 1 177
DDR4 32 Mode 2 179
DDR4 64 Mode 1 182
DDR4 64 Mode 2 184
Digital In/Out-Z (DS6651) 253
Digital InOut Class 1 (DS1302) 61
Digital InOut Class 2 (DS1302) 63
Digital Out (DS5203) 308
Digital Out (Type A) (FPGA1403TP1)
DS1552 435
DS1554 476
Digital Out (Type B) (FPGA1403TP1)
DS1552 437
Digital Out (Type B)(FPGA1403TP1)
DS1554 479
Digital Out M1 (DS5203M1) 332
Digital Out Type A (FPGA1401TP1)
DS1552 355
DS1554 396
Digital Out Type B (FPGA1401TP1)
DS1552 357
DS1554 400
Digital Out-Z (DS2655M2) 225
Digital Out-Z (DS6651) 259
I-FPGA Out (IOCNET) (DS2655) 108
I-FPGA Out (IOCNET) (DS6601) 144
I-FPGA Out (IOCNET) (DS6602) 190
I-FPGA64 Out (IOCNET) (DS2655) 110
I-FPGA64 Out (IOCNET) (DS6601) 146
I-FPGA64 Out (IOCNET) (DS6602) 192
Resolver (DS1302) 75
RS485 Tx (DS2655M2) 235
RS485 Tx (DS6651) 266
Trigger (DS6651) 268
Enable explicit data transmission
Buffer Out (DS2655) 99
Buffer Out (DS6601) 134
Buffer Out (DS6602) 171
Buffer64 Out (DS2655) 101
Buffer64 Out (DS6601) 137
Buffer64 Out (DS6602) 173
End of conversion
ADC (Type A) (FPGA1403TP1)
DS1552 417
DS1554 463
ADC Type A (FPGA1401TP1)
DS1552 337
DS1554 383
Knock Sensor (FPGA1401TP1) 403
Knock Sensor (FPGA1403TP1) 482
Endbit
I-FPGA In (SCALEXIO) 283
I-FPGA Master (DS5203) 310
I-FPGA Out (SCALEXIO) 288
I-FPGA Slave (DS5203) 312
Errors
I-FPGA In (SCALEXIO) 286
I-FPGA Slave (DS5203) 314
Errors reset
I-FPGA In (SCALEXIO) 286
I-FPGA Slave (DS5203) 314
Excitation RMS voltage
Resolver (DS1302) 74

F

Fault
Resolver (DS1302) 76

Filter depth
I-FPGA In (SCALEXIO) 285
I-FPGA Slave (DS5203) 313
Format
Buffer In (DS1302) 50
Buffer In (DS2655) 94
Buffer In (DS5203) 297
Buffer In (DS6601) 130
Buffer In (DS6602) 166
Buffer In (FPGA1401TP1)
DS1552 340
DS1554 385
Buffer In (FPGA1403TP1)
DS1552 420
DS1554 465
Buffer Out (DS1302) 53
Buffer Out (DS2655) 98
Buffer Out (DS5203) 301
Buffer Out (DS6601) 134
Buffer Out (DS6602) 170
Buffer Out (FPGA1401TP1)
DS1552 344
DS1554 389
Buffer Out (FPGA1403TP1)
DS1552 424
DS1554 469
Buffer64 In (DS1302) 52
Buffer64 In (DS2655) 96
Buffer64 In (DS5203) 299
Buffer64 In (DS6601) 132
Buffer64 In (DS6602) 168
Buffer64 In (FPGA1401TP1)
DS1552 342
DS1554 387
Buffer64 In (FPGA1403TP1)
DS1552 422
DS1554 467
Buffer64 Out (DS1302) 55
Buffer64 Out (DS2655) 101
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 136
Buffer64 Out (DS6602) 173
Buffer64 Out (FPGA1401TP1)
DS1552 346
DS1554 391
Buffer64 Out (FPGA1403TP1)
DS1552 426
DS1554 471
Register In (DS1302) 68
Register In (DS2655) 114
Register In (DS5203) 317
Register In (DS6601) 150
Register In (DS6602) 196
Register In (FPGA1401TP1)
DS1552 361
DS1554 406
Register In (FPGA1403TP1)
DS1552 441
DS1554 485
Register Out (DS1302) 71
Register Out (DS2655) 118
Register Out (DS5203) 320
Register Out (DS6601) 154
Register Out (DS6602) 200
Register Out (FPGA1401TP1)
DS1552 365
DS1554 409
Register Out (FPGA1403TP1)
DS1552 445
DS1554 488
Register64 In (DS1302) 70
Register64 In (DS2655) 116
Register64 In (DS5203) 319
Register64 In (DS6601) 152
Register64 In (DS6602) 198
Register64 In (FPGA1401TP1)
DS1552 363
DS1554 408
Register64 In (FPGA1403TP1)
DS1552 443
DS1554 486
Register64 Out (DS1302) 73
Register64 Out (DS2655) 119
Register64 Out (DS5203) 322
Register64 Out (DS6601) 155
Register64 Out (DS6602) 201
Register64 Out (FPGA1401TP1)
DS1552 367
DS1554 411
Register64 Out (FPGA1403TP1)
DS1552 446
DS1554 490
Frequency
Buzzer (DS1302) 57

G

Green
LED Out (DS1302) 66

H

HcCustomName
ADC (DS5203) 293
ADC (Type A) (FPGA1403TP1)
DS1552 417
DS1554 463
ADC (Type B) (FPGA1403TP1) 418
ADC Class 1 (DS1302) 47
ADC Class 2 (DS1302) 48
ADC M1 (DS5203M1) 326
ADC Type A (FPGA1401TP1)
DS1552 337
DS1554 383
ADC Type B (FPGA1401TP1) 339
Analog In (DS2655M1) 206
Analog In (DS6651) 241
Analog In-L (DS6651) 244
Analog Out (DS2655M1) 208
Analog Out (DS6651) 247
Analog Out-T (DS6651) 248
APU Master (DS2655) 89
APU Master (DS5203) 295
APU Master (DS6601) 125
APU Master (DS6602) 161
APU Slave (DS2655) 91
APU Slave (DS5203) 296
APU Slave (DS6601) 127, 163
Aurora 64b66b 128 Bit In 274
Aurora 64b66b 128 Bit Out 275
Aurora 64b66b In 270
Aurora 64b66b Out 272
Buffer In (DS1302) 49
Buffer In (DS2655) 93
Buffer In (DS5203) 297
Buffer In (DS6601) 129
Buffer In (DS6602) 165
Buffer In (FPGA1401TP1)
DS1552 340
DS1554 385
Buffer In (FPGA1403TP1)
DS1552 420
DS1554 465
Buffer Out (DS1302) 53
Buffer Out (DS2655) 98
Buffer Out (DS5203) 300
Buffer Out (DS6601) 134
Buffer Out (DS6602) 170
Buffer Out (FPGA1401TP1)
DS1552 344
DS1554 388
Buffer Out (FPGA1403TP1)
DS1552 423
DS1554 468
Buffer64 In (DS1302) 51
Buffer64 In (DS2655) 96
Buffer64 In (DS5203) 298
Buffer64 In (DS6601) 131
Buffer64 In (DS6602) 168
Buffer64 In (FPGA1401TP1)
DS1552 342
DS1554 387
Buffer64 In (FPGA1403TP1)
DS1552 422
DS1554 467
Buffer64 Out (DS1302) 55
Buffer64 Out (DS2655) 100
Buffer64 Out (DS5203) 302
Buffer64 Out (DS6601) 136
Buffer64 Out (DS6602) 172
Buffer64 Out (FPGA1401TP1)
DS1552 346
DS1554 390
Buffer64 Out (FPGA1403TP1)
DS1552 425
DS1554 470
Buzzer (DS1302) 57
CN App Status (DS2655) 103
CN App Status (DS6601) 139
CN App Status (DS6602) 175
DAC (DS5203) 304
DAC (FPGA1401TP1) 348
DAC (FPGA1403TP1) 427
DAC Class 1 (DS1302) 58

DAC M1 (DS5203M1) 328
DDR4 32 Mode 1 176
DDR4 32 Mode 2 178
DDR4 64 Mode 1 181
DDR4 64 Mode 2 183
Digital Crank/Cam Sensor (FPGA1401TP1)
DS1552 349
DS1554 392
Digital Crank/Cam Sensor (FPGA1403TP1)
DS1552 428
DS1554 472
Digital In (DS2655M1) 210
Digital In (DS2655M2) 219
Digital In (DS6651) 250
Digital In (Type A) (FPGA1403TP1) 430
Digital In (Type B) (FPGA1403TP1)
DS1552 432
DS1554 474
Digital In M1 (DS5203M1) 329
Digital In Type A (FPGA1401TP1) 351
Digital In Type B (FPGA1401TP1)
DS1552 352
DS1554 394
Digital In/Out-Z (DS6651) 252
Digital InOut (DS2655M1) 212
Digital InOut Class 1 (DS1302) 60
Digital InOut Class 2 (DS1302) 62
Digital Out (DS2655M1) 214
Digital Out (DS2655M2) 221
Digital Out (DS5203) 308
Digital Out (DS6651) 255
Digital Out (Type A) (FPGA1403TP1)
DS1552 434
DS1554 476
Digital Out (Type B) (FPGA1403TP1)
DS1552 436
DS1554 478
Digital Out M1 (DS5203M1) 332
Digital Out Type A (FPGA1401TP1)
DS1552 354
DS1554 396
Digital Out Type B (FPGA1401TP1)
DS1552 356
DS1554 399
Digital Out-Z (DS2655M2) 224
Digital Out-Z (DS6651) 258
I-FPGA In (IOCNET) (DS2655) 104
I-FPGA In (IOCNET) (DS6601) 139
I-FPGA In (IOCNET) (DS6602) 186
I-FPGA In (SCALEXIO) 283
I-FPGA Master (DS5203) 310
I-FPGA Out (IOCNET) (DS2655) 107
I-FPGA Out (IOCNET) (DS6601) 143
I-FPGA Out (IOCNET) (DS6602) 189
I-FPGA Out (SCALEXIO) 287
I-FPGA Slave (DS5203) 312
I-FPGA64 In (IOCNET) (DS2655) 105
I-FPGA64 In (IOCNET) (DS6601) 141
I-FPGA64 In (IOCNET) (DS6602) 188
I-FPGA64 Out (IOCNET) (DS2655) 109
I-FPGA64 Out (IOCNET) (DS6601) 145
I-FPGA64 Out (IOCNET) (DS6602) 192
Inductive Zero Voltage Detector (FPGA1401TP1)
DS1552 358
DS1554 401
Inductive Zero Voltage Detector (FPGA1403TP1)
DS1552 438
DS1554 480
Interrupt (DS1302) 65
Interrupt (DS2655) 112
Interrupt (DS5203) 315
Interrupt (DS6601) 148
Interrupt (DS6602) 194
Interrupt (FPGA1401TP1)
DS1552 359
DS1554 402
Interrupt (FPGA1403TP1)
DS1552 439
DS1554 481
IOCNET Global Time (DS2655) 112
IOCNET Global Time (DS6601) 148
IOCNET Global Time (DS6602) 194
Knock Sensor (FPGA1401TP1) 403
Knock Sensor (FPGA1403TP1) 482
LED Out (DS1302) 66
LED Out (DS2655) 113
LED Out (DS5203) 316
LED Out (DS6601) 149
LED Out (DS6602) 195
LED Out (FPGA1401TP1)
DS1552 360
DS1554 404
LED Out (FPGA1403TP1)
DS1552 440
DS1554 483
MGT In 277
MGT Out 279
Proc App Status (DS1302) 67
Register In (DS1302) 67
Register In (DS2655) 114
Register In (DS5203) 316
Register In (DS6601) 150
Register In (DS6602) 196
Register In (FPGA1401TP1)
DS1552 361
DS1554 405
Register In (FPGA1403TP1)
DS1552 441
DS1554 484
Register Out (DS1302) 71
Register Out (DS2655) 117
Register Out (DS5203) 320
Register Out (DS6601) 153
Register Out (DS6602) 199
Register Out (FPGA1401TP1)
DS1552 365
DS1554 409
Register Out (FPGA1403TP1)
DS1552 444
DS1554 488
Register64 In (DS1302) 69
Register64 In (DS2655) 115
Register64 In (DS5203) 318
Register64 In (DS6601) 151
Register64 In (DS6602) 198
Register64 In (FPGA1401TP1)
DS1552 363
DS1554 407
Register64 In (FPGA1403TP1)
DS1552 443
DS1554 486
Register64 Out (DS1302) 72
Register64 Out (DS2655) 119
Register64 Out (DS5203) 321
Register64 Out (DS6601) 155
Register64 Out (DS6602) 201
Register64 Out (FPGA1401TP1)
DS1552 366
DS1554 410
Register64 Out (FPGA1403TP1)
DS1552 446
DS1554 489
Resolver (DS1302) 74
RS232 Rx (DS2655M2) 227
RS232 Tx (DS2655M2) 229
RS485 Rx (DS2655M2) 230, 232
RS485 Rx (DS6651) 261
RS485 Rx/Tx (DS6651) 263
RS485 Tx (DS2655M2) 235
RS485 Tx (DS6651) 265
Sensor Supply (FPGA1401TP1)
DS1552 368
Sensor Supply (FPGA1403TP1)
DS1552 448
Sensor Supply M1 (DS5203M1) 334
Status In (DS1302) 77
Status In (DS2655) 120
Status In (DS5203) 323
Status In (DS6601) 156
Status In (DS6602) 202
Status In (FPGA1401TP1)
DS1552 369
DS1554 412
Status In (FPGA1403TP1)
DS1552 449
DS1554 491
Temperature (FPGA1401TP1) 413
Temperature (FPGA1403TP1) 491
Trigger (DS6651) 268
UART 3 RS232 (FPGA1403TP1) 450
UART 3 RS422/485 (FPGA1403TP1) 455
UART RS232 (DS1302) 78
UART RS232 (FPGA1401TP1) 370
UART RS422/485 (DS1302) 82
UART RS422/485 (FPGA1401TP1) 376
HCCustomName
Digital In (DS5203) 306
High
FPGA1401TP1 port 413
DS1552 369
FPGA1403TP1 port 492

DS1552 449
High supply
Digital InOut (DS2655M1) 212
Digital InOut Class 1 (DS1302) 60
Digital Out (DS2655M1) 215
Digital Out (DS2655M2) 222
Digital Out (DS5203) 308
Digital Out (Type B) (FPGA1403TP1)
DS1552 437
DS1554 478
Digital Out M1 (DS5203M1) 332
Digital Out Type B (FPGA1401TP1)
DS1552 357
DS1554 399
Digital Out-Z (DS2655M2) 224
RS485 Rx/Tx (DS6651) 263
RS485 Tx (DS6651) 265
High Supply
Digital In/Out-Z (DS6651) 252
Digital Out (DS6651) 256
Digital Out-Z (DS6651) 258
High threshold voltage
Digital Crank/Cam Sensor (FPGA1401TP1)
DS1552 349
DS1554 393
Digital Crank/Cam Sensor (FPGA1403TP1)
DS1552 429
DS1554 472
high_temp
FPGA1401TP1 port 413
DS1552 369
FPGA1403TP1 port 492
DS1552 449
hq_adc_<ChannelNumber>_data
ADC (Type A) (FPGA1403TP1)
DS1552 417
DS1554 463
ADC Type A (FPGA1401TP1)
DS1552 337
DS1554 383
Knock Sensor (FPGA1401TP1) 403
Knock Sensor (FPGA1403TP1) 482
hq_adc_<ChannelNumber>_eoc
ADC (Type A) (FPGA1403TP1)
DS1552 417
DS1554 463
ADC Type A (FPGA1401TP1)
DS1552 337
DS1554 383
Knock Sensor (FPGA1401TP1) 403
Knock Sensor (FPGA1403TP1) 482
hq_adc_<ChannelNumber>_soc
ADC (Type A) (FPGA1403TP1)
DS1552 417
DS1554 463
ADC Type A (FPGA1401TP1)
DS1552 337
DS1554 383
Knock Sensor (FPGA1401TP1) 403
Knock Sensor (FPGA1403TP1) 482

I

I-FPGA In (IOCNET) (DS2655) 104
Buffer size 104
I/O mapping 105
parameters 104
port 104
I-FPGA In (IOCNET) (DS6601) 139
Buffer size 140
I/O mapping 141
parameters 139
port 140
I-FPGA In (IOCNET) (DS6602) 186
Buffer size 186
I/O mapping 187
parameters 186
port 186
I-FPGA In (SCALEXIO) 282
parameters 282
port 285
I-FPGA Master (DS5203) 309
I/O mapping 311
parameters 310
port 311
I-FPGA Out (IOCNET) (DS2655) 107
Buffer size 107
I/O mapping 109
parameters 107
port 108
I-FPGA Out (IOCNET) (DS6601) 143
Buffer size 143
I/O mapping 145
parameters 143
port 144
I-FPGA Out (IOCNET) (DS6602) 189
Buffer size 190
I/O mapping 191
parameters 189
port 190
I-FPGA Out (SCALEXIO) 287
I/O mapping 286, 290
parameters 287
port 289
I-FPGA Slave (DS5203) 312
I/O mapping 314
parameters 312
port 314
I-FPGA64 In (IOCNET) (DS2655) 105
Buffer size 106
I/O mapping 107
parameters 105
port 106
I-FPGA64 In (IOCNET) (DS6601) 141
Buffer size 142
I/O mapping 143
parameters 141
port 142
I-FPGA64 In (IOCNET) (DS6602) 188
Buffer size 188
I/O mapping 189
parameters 188
port 188
I-FPGA64 Out (IOCNET) (DS2655) 109
Buffer size 110
I/O mapping 111
parameters 109
port 110
I-FPGA64 Out (IOCNET) (DS6601) 145
Buffer size 146
I/O mapping 147
parameters 145
port 146
I-FPGA64 Out (IOCNET) (DS6602) 191
Buffer size 192
I/O mapping 193
parameters 192
port 192
Inductive Zero Voltage Detector (FPGA1401TP1)
DS1552 358
DS1554 401
I/O mapping
DS1552 359
DS1554 401
parameters
DS1552 358
DS1554 401
port
DS1552 358
DS1554 401
Inductive Zero Voltage Detector (FPGA1403TP1)
DS1552 438
DS1554 480
I/O mapping
DS1552 439
DS1554 480
parameters
DS1552 438
DS1554 480
port
DS1552 438
DS1554 480
Init Done
DDR4 32 Mode 1 176
DDR4 32 Mode 2 178
DDR4 64 Mode 1 181
DDR4 64 Mode 2 183
FPGA1401TP1 port
DS1552 369
DS1554 412
FPGA1403TP1 port
DS1552 449
DS1554 491
Status In (DS1302) 77
Status In (DS2655) 120
Status In (DS5203) 323
Status In (DS6601) 156
Status In (DS6602) 203
Init Failed
DDR4 32 Mode 1 176
DDR4 32 Mode 2 179
DDR4 64 Mode 1 182
DDR4 64 Mode 2 184
init_done

FPGA1401TP1 port
DS1552 369
DS1554 412
FPGA1403TP1 port
DS1552 449
DS1554 491
Status In (DS1302) 77
Status In (DS2655) 120
Status In (DS5203) 323
Status In (DS6601) 156
Status In (DS6602) 203
Input filter
Digital InOut Class 1 (DS1302) 60
Digital InOut Class 2 (DS1302) 63
Input range
ADC (DS5203) 293
ADC M1 (DS5203M1) 326
Analog In (DS2655M1) 206
Analog In (DS6651) 241
Analog In-L (DS6651) 244
Input RMS voltage
Resolver (DS1302) 75
Int
Interrupt (DS1302) 65
Interrupt (DS2655) 112
Interrupt (DS5203) 315
Interrupt (DS6601) 148
Interrupt (DS6602) 194
Interrupt (FPGA1401TP1)
DS1552 360
DS1554 402
Interrupt (FPGA1403TP1)
DS1552 440
DS1554 481
intercom_<ChannelNumber>_data_in
I-FPGA In (SCALEXIO) 285
I-FPGA Slave (DS5203) 314
intercom_<ChannelNumber>_data_new
I-FPGA In (SCALEXIO) 286
I-FPGA Slave (DS5203) 314
intercom_<ChannelNumber>_data_out
I-FPGA Master (DS5203) 311
I-FPGA Master (SCALEXIO) 289
intercom_<ChannelNumber>_data_sent_v
I-FPGA Master (DS5203) 311
I-FPGA Master (SCALEXIO) 290
intercom_<ChannelNumber>_errors
I-FPGA In (SCALEXIO) 286
I-FPGA Slave (DS5203) 314
intercom_<ChannelNumber>_ready
I-FPGA Master (SCALEXIO) 290
intercom_<ChannelNumber>_reset_error
I-FPGA In (SCALEXIO) 286
I-FPGA Slave (DS5203) 314
Intermodule bus access
buffer
DS1552 340, 343, 420, 423
DS1554 385, 388, 465, 468
buffer64
DS1552 341, 345, 421, 425
DS1554 386, 390, 466, 470
register
DS1552 361, 362, 364, 366, 441, 442, 444, 446
DS1554 405, 407, 409, 410, 484, 486, 487, 489
Internal bus access (DS1302)
buffer (32-bit) 49, 53
buffer (64-bit) 51, 55
register (32-bit) 67, 71
register (64-bit) 69, 72
Internal bus access (DS5203)
buffer (64-bit) 298, 302
Interrupt (DS1302) 65
parameters 65
port 65
Interrupt (DS2655) 111
parameters 112
port 112
Interrupt (DS5203) 315
parameters 315
port 315
Interrupt (DS6601) 147
parameters 148
port 148
Interrupt (DS6602) 193
parameters 194
port 194
Interrupt (FPGA1401TP1)
DS1552 359
DS1554 402
parameters
DS1552 359
DS1554 402
port
DS1552 359
DS1554 402
Interrupt (FPGA1403TP1)
DS1552 439
DS1554 481
parameters
DS1552 439
DS1554 481
port
DS1552 440
DS1554 481
Invert values
Digital InOut Class 1 (DS1302) 60
Digital InOut Class 2 (DS1302) 63
IOCNET access (DS2655)
buffer (32-bit) 93, 98
buffer (64-bit) 95, 100
register (32-bit) 114, 117
register (64-bit) 115, 118
IOCNET access (DS6601)
buffer (32-bit) 129, 133
buffer (64-bit) 131, 136
register (32-bit) 150, 153
register (64-bit) 151, 154
IOCNET access (DS6602)
buffer (32-bit) 165, 170
buffer (64-bit) 167, 172
register (32-bit) 196, 199
register (64-bit) 197, 201
IOCNET Global Time (DS2655) 112
parameters 112
port 112
IOCNET Global Time (DS6601) 148
parameters 148
port 148
IOCNET Global Time (DS6602) 194
parameters 194
port 195
iocnet_appl_status
CN App Status (DS2655) 103, 113
CN App Status (DS6601) 139, 149
CN App Status (DS6602) 175, 195
iocnet_glob_angle<ChannelNumber>_ctr
APU Slave (DS2655) 93
APU Slave (DS6601) 128, 165
iocnet_glob_angle<ChannelNumber>_en
APU Slave (DS2655) 93
APU Slave (DS6601) 128, 165
iocnet_glob_angle<ChannelNumber>_pos
APU Slave (DS2655) 92
APU Slave (DS6601) 128, 164
iocnet_glob_angle<ChannelNumber>_res
APU Slave (DS2655) 93
APU Slave (DS6601) 128, 165
iocnet_glob_angle<ChannelNumber>_rev
APU Slave (DS2655) 92
APU Slave (DS6601) 128, 164
iocnet_glob_angle<ChannelNumber>_vel
APU Slave (DS2655) 93
APU Slave (DS6601) 128, 165
iocnet_glob_master_angle<ChannelNumber>_busy
APU Master (DS2655) 90
APU Master (DS6601) 126
APU Master (DS6602) 162
iocnet_glob_master_angle<ChannelNumber>_ctr
APU Master (DS2655) 90
APU Master (DS6601) 125
APU Master (DS6602) 162
iocnet_glob_master_angle<ChannelNumber>_en
APU Master (DS2655) 90
APU Master (DS6601) 126
APU Master (DS6602) 162
iocnet_glob_master_angle<ChannelNumber>_pos
APU Master (DS2655) 90
APU Master (DS6601) 125
APU Master (DS6602) 162
iocnet_glob_master_angle<ChannelNumber>_res
APU Master (DS2655) 90
APU Master (DS6601) 126
APU Master (DS6602) 162
iocnet_glob_master_angle<ChannelNumber>_rev
APU Master (DS2655) 90
APU Master (DS6601) 126
APU Master (DS6602) 162

iocnet_glob_master_angle<ChannelNumber>_upd_trig
APU Master (DS2655) 91
APU Master (DS6601) 126
APU Master (DS6602) 163
iocnet_glob_master_angle<ChannelNumber>_upd_vel_deg_sec
APU Master (DS2655) 90
APU Master (DS6601) 126
APU Master (DS6602) 163
iocnet_glob_master_angle<ChannelNumber>_vel
APU Master (DS2655) 90
APU Master (DS6601) 126
APU Master (DS6602) 162

K

Knock Sensor (FPGA1401TP1) 403
I/O mapping 404
parameters 403
port 403
Knock Sensor (FPGA1403TP1)
DS1554 482
I/O mapping 483
parameters 482
port 482

L

LED Out
LED Out (DS2655) 113
LED Out (DS6601) 149
LED Out (DS6602) 195
LED Out (DS1302) 65
parameters 66
port 66
LED Out (DS2655) 113
parameters 113
port 113
LED Out (DS5203) 315
parameters 316
port 316
LED Out (DS6601) 149
parameters 149
port 149
LED Out (DS6602) 195
parameters 195
port 195
LED Out (FPGA1401TP1)
DS1552 360
DS1554 404
parameters
DS1552 360
DS1554 404
port
DS1552 360
DS1554 405
LED Out (FPGA1403TP1)
DS1552 440
DS1554 483
parameters
DS1552 440
DS1554 483
port
DS1552 440
DS1554 483
led_<ChannelNumber>_blue
LED Out (DS1302) 66
led_<ChannelNumber>_green
LED Out (DS1302) 66
led_<ChannelNumber>_red
LED Out (DS1302) 66
led_out
LED Out (DS2655) 113
LED Out (DS5203) 316
LED Out (DS6601) 149
LED Out (DS6602) 195
LED Out (FPGA1401TP1)
DS1552 360
DS1554 405
LED Out (FPGA1403TP1)
DS1552 440
DS1554 484
Load Config
Analog In-L (DS6651) 245
Local Program Data folder 12
Low threshold voltage
Digital Crank/Cam Sensor (FPGA1401TP1)
DS1552 349
DS1554 393
Digital Crank/Cam Sensor (FPGA1403TP1)
DS1552 429
DS1554 472
lq_adc_<ChannelNumber>_data
ADC (Type B) (FPGA1403TP1) 419
ADC Type B (FPGA1401TP1) 339

M

m<ModuleNumber>_io2raw_n_n_n_ni
Digital In (DS2655M2) 219
RS232 Rx (DS2655M2) 227
RS485 Rx (DS2655M2) 230
RS485 RxTx (DS2655M2) 233
m<ModuleNumber>_raw2io_n_n_n_ni
Digital Out (DS2655M2) 222
Digital Out-Z (DS2655M2) 225
RS232 Tx (DS2655M2) 229
RS485 Tx (DS2655M2) 235
m<ModuleNumber>_raw2io_n+1_n+1_n+1_n+1i
Digital Out-Z (DS2655M2) 225
RS485 RxTx (DS2655M2) 232
RS485 Tx (DS2655M2) 235
m<ModuleNumber>_raw2io_n+2_n+2_n+2_n+2i
RS485 Tx (DS2655M2) 233
m<ModuleNumber>_rx<ChannelNumber>_data00
Analog In (DS2655M1) 207
Analog In (DS6651) 242
Analog In-L (DS6651) 246
m<ModuleNumber>_rx<ChannelNumber>_ready_pulse
Analog In (DS2655M1) 207
Analog In (DS6651) 242
Analog In-L (DS6651) 246
m<ModuleNumber>_rx<ChannelNumber>_trigger00
Digital In (DS2655M1) 210
Digital InOut (DS2655M1) 213
m<ModuleNumber>_rx<ChannelNumber>_trigger00 / Data
Digital In (DS6651) 250
RS485 Rx (DS6651) 261
m<ModuleNumber>_rx<ChannelNumber>_trigger00 / Data In
Digital In/Out-Z (DS6651) 253
m<ModuleNumber>_rx<ChannelNumber>_trigger00 / Rx Data
RS485 Rx/Tx (DS6651) 263
m<ModuleNumber>_tx<ChannelNumber>__data00
Analog In (DS2655M1) 207
m<ModuleNumber>_tx<ChannelNumber>_data00_14_0_14_0
Analog Out (DS2655M1) 209
m<ModuleNumber>_tx<ChannelNumber>_data00_16_0_16_0i
Analog Out (DS2655M1) 209
m<ModuleNumber>_tx<ChannelNumber>_data00_17_0_17_0i
Analog Out (DS6651) 247
Analog Out-T (DS6651) 249
m<ModuleNumber>_tx<ChannelNumber>_data01_0_0_0_0i
Digital InOut (DS2655M1) 213
m<ModuleNumber>_tx<ChannelNumber>_data01_0_0_1_1i
Digital InOut (DS2655M1) 213
m<ModuleNumber>_tx<ChannelNumber>_trigger00
Digital Out (DS2655M1) 215
m<ModuleNumber>_tx<ChannelNumber>_trigger00 / Data
Digital Out (DS6651) 256
Digital Out-Z (DS6651) 259
RS485 Tx (DS6651) 266
m<ModuleNumber>_tx<ChannelNumber>_trigger00 / Data Out
Digital In/Out-Z (DS6651) 253
m<ModuleNumber>_tx<ChannelNumber>_trigger00 / Enable
Digital In/Out-Z (DS6651) 253
Digital Out-Z (DS6651) 259
RS485 Tx (DS6651) 266
Trigger (DS6651) 268
m<ModuleNumber>_tx<ChannelNumber>_trigger00 / Tx Data
RS485 Rx/Tx (DS6651) 263
m<ModuleNumber>_tx<ChannelNumber>_trigger00 / Tx Enable
RS485 Rx/Tx (DS6651) 264

m1_adc<ChannelNumber>_data
ADC M1 (DS5203M1) 326
m1_adc<ChannelNumber>_valid
ADC M1 (DS5203M1) 327
m1_dac<ChannelNumber>_data
DAC M1 (DS5203M1) 328
m1_digio_<ChannelNumber>_in
Digital In M1 (DS5203M1) 330
m1_digio_<ChannelNumber>_oe
Digital Out M1 (DS5203M1) 332
m1_digio_<ChannelNumber>_out
Digital Out M1 (DS5203M1) 332
Maximum speed
Resolver (DS1302) 75
Mechanical Position
Resolver (DS1302) 75
MGT In
DS6601 277
DS6602 277
I/O mapping 278
parameters 277
port 277
MGT Out
DS6601 278
DS6602 278
I/O mapping 280
parameters 279
port 279
MGT reference clock frequency
MGT In 277
MGT_CLK_N
MGT In 278
MGT_CLK_P
MGT In 277
mgt_lane<LaneNumber>_rx_data
Aurora 64b66b 128 Bit In 274
Aurora 64b66b In 270
mgt_lane<LaneNumber>_rx_valid
Aurora 64b66b 128 Bit In 274
Aurora 64b66b In 271
mgt_lane<LaneNumber>_tx_data
Aurora 64b66b 128 Bit Out 276
Aurora 64b66b Out 272
mgt_lane<LaneNumber>_tx_ready
Aurora 64b66b 128 Bit Out 276
Aurora 64b66b Out 273
mgt_lane<LaneNumber>_tx_valid
Aurora 64b66b 128 Bit Out 276
Aurora 64b66b Out 272
MGT_RX_N
MGT In 278
MGT_RX_P
MGT In 278
MGT_TX_N
MGT Out 279
MGT_TX_P
MGT Out 279
Mode
Analog Out-T (DS6651) 249
I-FPGA In (SCALEXIO) 283
I-FPGA Master (DS5203) 310
I-FPGA Out (SCALEXIO) 287
I-FPGA Slave (DS5203) 312

O

Output mode
Digital InOut (DS2655M1) 212
Digital Out (DS2655M1) 214
Digital Out (DS2655M2) 221
Output Mode
Digital In/Out-Z (DS6651) 252
Digital Out (DS6651) 255
Digital Out-Z (DS6651) 258
Overflow
Buffer Out (DS1302) 54
Buffer Out (DS2655) 99
Buffer Out (DS5203) 301
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171
Buffer Out (FPGA1401TP1)
DS1552 345
DS1554 390
Buffer Out (FPGA1403TP1)
DS1552 425
DS1554 470
Buffer64 Out (DS1302) 56
Buffer64 Out (DS2655) 102
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 138
Buffer64 Out (DS6602) 174
Buffer64 Out (FPGA1401TP1)
DS1552 347
DS1554 392
Buffer64 Out (FPGA1403TP1)
DS1552 427
DS1554 472
I-FPGA Out (IOCNET) (DS2655) 109
I-FPGA Out (IOCNET) (DS6601) 145
I-FPGA Out (IOCNET) (DS6602) 191
I-FPGA64 Out (IOCNET) (DS2655) 111
I-FPGA64 Out (IOCNET) (DS6601) 147
I-FPGA64 Out (IOCNET) (DS6602) 193

P

Pause Duration
Buzzer (DS1302) 57
Phi New
APU Master (DS2655) 90
APU Master (DS6601) 126
APU Master (DS6602) 162
APU Slave (DS2655) 92
APU Slave (DS5203) 296
APU Slave (DS6601) 128, 164
Phi Read
APU Master (DS2655) 90
APU Master (DS6601) 125
APU Master (DS6602) 162
APU Slave (DS2655) 92
APU Slave (DS5203) 296
APU Slave (DS6601) 128, 164
Phi Read HD
APU Master (DS2655) 90
APU Master (DS6601) 125
APU Master (DS6602) 162
APU Slave (DS2655) 93
APU Slave (DS6601) 128, 165
Phi Write
APU Master (DS5203) 295
PHS bus access (DS5203)
buffer 297, 300
register 316, 320
register (64-bit) 318, 321
Proc App Status (DS1302) 66
parameters 67
port 67
Processor Application Status
Proc App Status (DS1302) 67

R

Read Data
UART 3 RS232 (FPGA1403TP1) 452
UART 3 RS422/485 (FPGA1403TP1) 457
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 85
UART RS422/485 (FPGA1401TP1) 378
Read Data Count
UART 3 RS232 (FPGA1403TP1) 452
UART 3 RS422/485 (FPGA1403TP1) 457
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 84
UART RS422/485 (FPGA1401TP1) 378
Read Enable
UART 3 RS232 (FPGA1403TP1) 452
UART 3 RS422/485 (FPGA1403TP1) 457
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 84
UART RS422/485 (FPGA1401TP1) 378
Read Fifo Empty
UART 3 RS232 (FPGA1403TP1) 452
UART 3 RS422/485 (FPGA1403TP1) 457
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 85
UART RS422/485 (FPGA1401TP1) 378
Read Request
Buffer In (DS2655) 95
Buffer In (DS6601) 130
Buffer In (DS6602) 167
Buffer64 In (DS2655) 97
Buffer64 In (DS6601) 133
Buffer64 In (DS6602) 169
Ready
Aurora 64b66b 128 Bit Out 276
Aurora 64b66b Out 273
Buffer Out (DS1302) 54
Buffer Out (DS2655) 99
Buffer Out (DS5203) 301
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171

Buffer Out (FPGA1401TP1)
DS1552 345
DS1554 390
Buffer Out (FPGA1403TP1)
DS1552 424
DS1554 469
Buffer64 Out (DS1302) 56
Buffer64 Out (DS2655) 102
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 137
Buffer64 Out (DS6602) 174
Buffer64 Out (FPGA1401TP1)
DS1552 347
DS1554 392
Buffer64 Out (FPGA1403TP1)
DS1552 426
DS1554 471
I-FPGA Master (SCALEXIO) 290
I-FPGA Out (IOCNET) (DS2655) 108
I-FPGA Out (IOCNET) (DS6601) 144
I-FPGA Out (IOCNET) (DS6602) 190
I-FPGA64 Out (IOCNET) (DS2655) 110
I-FPGA64 Out (IOCNET) (DS6601) 146
I-FPGA64 Out (IOCNET) (DS6602) 192
Red
LED Out (DS1302) 66
Register group ID
Register In (DS1302) 68
Register In (DS2655) 114
Register In (DS5203) 317
Register In (DS6601) 150
Register In (DS6602) 197
Register In (FPGA1401TP1)
DS1552 362
DS1554 406
Register In (FPGA1403TP1)
DS1552 442
DS1554 485
Register Out (DS1302) 72
Register Out (DS2655) 118
Register Out (DS5203) 320
Register Out (DS6601) 154
Register Out (DS6602) 200
Register Out (FPGA1401TP1)
DS1552 365
DS1554 410
Register Out (FPGA1403TP1)
DS1552 445
DS1554 488
Register64 In (DS1302) 70
Register64 In (DS2655) 116
Register64 In (DS5203) 319
Register64 In (DS6601) 152
Register64 In (DS6602) 198
Register64 In (FPGA1401TP1)
DS1552 363
DS1554 408
Register64 In (FPGA1403TP1)
DS1552 443
DS1554 487
Register64 Out (DS1302) 73
Register64 Out (DS2655) 119
Register64 Out (DS5203) 322
Register64 Out (DS6601) 155
Register64 Out (DS6602) 202
Register64 Out (FPGA1401TP1)
DS1552 367
DS1554 411
Register64 Out (FPGA1403TP1)
DS1552 447
DS1554 490
Register In (DS1302) 67
parameters 67
port 68
Register In (DS2655) 114
parameters 114
port 115
Register In (DS5203) 316
parameters 316
port 317
Register In (DS6601) 150
parameters 150
port 151
Register In (DS6602) 196
parameters 196
port 197
Register In (FPGA1401TP1)
DS1552 361
DS1554 405
parameters
DS1552 361
DS1554 405
port
DS1552 362
DS1554 406
Register In (FPGA1403TP1)
DS1552 441
DS1554 484
parameters
DS1552 441
DS1554 484
port
DS1552 442
DS1554 485
Register Out (DS1302) 71
parameters 71
port 72
Register Out (DS2655) 117
parameters 117
port 118
Register Out (DS5203) 320
parameters 320
port 321
Register Out (DS6601) 153
parameters 153
port 154
Register Out (DS6602) 199
parameters 199
port 200
Register Out (FPGA1401TP1)
DS1552 364
DS1554 409
parameters
DS1552 365
DS1554 409
port
DS1552 365
DS1554 410
Register Out (FPGA1403TP1)
DS1552 444
DS1554 487
parameters
DS1552 444
DS1554 488
port
DS1552 445
DS1554 488
Register64 In (DS1302) 69
parameters 69
port 70, 319
Register64 In (DS2655) 115
parameters 115
port 116
Register64 In (DS5203) 318
parameters 318
Register64 In (DS6601) 151
parameters 151
port 152
Register64 In (DS6602) 197
parameters 198
port 199
Register64 In (FPGA1401TP1)
DS1552 362
DS1554 407
parameters
DS1552 363
DS1554 407
port
DS1552 364
DS1554 408
Register64 In (FPGA1403TP1)
DS1552 442
DS1554 486
parameters
DS1552 443
DS1554 486
port
DS1552 444
DS1554 487
Register64 Out (DS1302) 72
parameters 72
port 73
Register64 Out (DS2655) 118
parameters 119
port 120
Register64 Out (DS5203) 321
parameters 321
port 322
Register64 Out (DS6601) 154
parameters 155
port 156
Register64 Out (DS6602) 201
parameters 201

port 202
Register64 Out (FPGA1401TP1)
DS1552 366
DS1554 410
parameters
DS1552 366
DS1554 410
port
DS1552 367
DS1554 411
Register64 Out (FPGA1403TP1)
DS1552 446
DS1554 489
parameters
DS1552 446
DS1554 489
port
DS1552 447
DS1554 490
Reset
Resolver (DS1302) 76
Resolver (DS1302) 74
I/O mapping 76
parameters 74
port 75
resolver_<ChannelNumber>_enable
Resolver (DS1302) 75
resolver_<ChannelNumber>_err_rst
Resolver (DS1302) 76
resolver_<ChannelNumber>_fault
Resolver (DS1302) 76
resolver_<ChannelNumber>_mech_pos
Resolver (DS1302) 75
resolver_<ChannelNumber>_update
Resolver (DS1302) 76
resolver_<ChannelNumber>_valid
Resolver (DS1302) 75
Rising edge delay
Digital InOut Class 1 (DS1302) 60
Digital InOut Class 2 (DS1302) 63
RS232 Rx (DS2655M2) 227
I/O mapping 228
parameters 227
port 227
RS232 Tx (DS2655M2) 228
I/O mapping 229
parameters 228
port 229
RS485 Rx (DS2655M2) 230
I/O mapping 231
parameters 230
port 230
RS485 Rx (DS6651) 260
I/O mapping 261
parameters 261
port 261
RS485 Rx/Tx (DS6651) 262
I/O mapping 264
parameters 263
ports 263
RS485 RxTx (DS2655M2) 232
I/O mapping 233
parameters 232
port 232
RS485 Termination
RS485 Rx (DS6651) 261
RS485 Rx/Tx (DS6651) 263
RS485 Tx (DS6651) 265
RS485 Termination config
RS485 Rx (DS2655M2) 230
RS485 RxTx (DS2655M2) 232
RS485 Tx (DS2655M2) 235
RS485 Tx (DS2655M2) 235
I/O mapping 236
parameters 235
port 235
RS485 Tx (DS6651) 265
I/O mapping 266
parameters 265
ports 266
RTS
UART 3 RS232 (FPGA1403TP1) 453
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374
Rx Data
RS485 Rx/Tx (DS6651) 263
RS485 RxTx (DS2655M2) 233
RX_N
MGT In 278
RX_P
MGT In 278

S

Sample period
Analog In (DS6651) 242
Analog In-L (DS6651) 245
Scaling
ADC (DS5203) 293
ADC M1 (DS5203M1) 326
Analog In (DS2655M1) 207
Analog In (DS6651) 241
Analog In-L (DS6651) 244
Analog Out (DS2655M1) 208
Analog Out (DS6651) 247
Analog Out-T (DS6651) 248
DAC (DS5203) 304
DAC M1 (DS5203M1) 328
Send
I-FPGA Out (IOCNET) (DS2655) 108
I-FPGA Out (IOCNET) (DS6601) 144
I-FPGA Out (IOCNET) (DS6602) 191
I-FPGA64 Out (IOCNET) (DS2655) 111
I-FPGA64 Out (IOCNET) (DS6601) 147
I-FPGA64 Out (IOCNET) (DS6602) 193
Send Acknowledge
Buffer Out (DS2655) 99
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171
Buffer64 Out (DS2655) 102
Buffer64 Out (DS6601) 138
Buffer64 Out (DS6602) 174
Sensor Supply (FPGA1401TP1)
DS1552 368
I/O mapping
DS1552 368
parameters
DS1552 368
port
DS1552 368
Sensor Supply (FPGA1403TP1)
DS1552 447
I/O mapping
DS1552 448
parameters
DS1552 448
port
DS1552 448
Sensor Supply M1 (DS5203M1) 333
I/O mapping 334
parameters 334
port 334
Start
Buzzer (DS1302) 58
Start of conversion
ADC (Type A) (FPGA1403TP1)
DS1552 417
DS1554 463
ADC Type A (FPGA1401TP1)
DS1552 337
DS1554 383
Knock Sensor (FPGA1401TP1) 403
Knock Sensor (FPGA1403TP1) 482
Startbit
I-FPGA In (SCALEXIO) 283
I-FPGA Master (DS5203) 310
I-FPGA Out (SCALEXIO) 288
I-FPGA Slave (DS5203) 312
Status In (DS1302) 77
parameters 77
port 77
Status In (DS2655) 120
parameters 120
port 120
Status In (DS5203) 323
parameters 323
port 323
Status In (DS6601) 156
parameters 156
port 156
Status In (DS6602) 202
parameters 202
port 202
Status In (FPGA1401TP1)
DS1552 368
DS1554 412
parameters
DS1552 369
DS1554 412
port
DS1552 369
DS1554 412
Status In (FPGA1403TP1)
DS1552 448

DS1554 490
parameters
DS1552 449
DS1554 491
port
DS1552 449
DS1554 491
Stop bits
UART RS232 (DS1302) 79
UART RS232 (FPGA1403TP1) 372, 451
UART RS422/485 (DS1302) 83
UART RS422/485 (FPGA1401TP1) 377, 456
Supply voltage
Sensor Supply (FPGA1401TP1)
DS1552 368
Sensor Supply (FPGA1403TP1)
DS1552 448
Sensor Supply M1 (DS5203M1) 334

T

temperature
FPGA1401TP1 port 413
DS1552 369
FPGA1403TP1 port 492
DS1552 449
Temperature (FPGA1401TP1) 413
parameters 413
port 413
Temperature (FPGA1403TP1) 491
parameters 491
port 492
Termination
UART RS232 (DS1302) 79
UART RS232 (FPGA1401TP1) 372
UART RS232 (FPGA1403TP1) 451
UART RS422/485 (DS1302) 84
UART RS422/485 (FPGA1401TP1) 377
UART RS422/485 (FPGA1403TP1) 456
Threshold init voltage
Digital In (DS2655M1) 210
Digital In (DS2655M2) 219
Digital In (DS6651) 250
Digital In/Out-Z (DS6651) 252
Threshold voltage
Digital In (DS5203) 306
Digital In (Type B) (FPGA1403TP1)
DS1552 433
DS1554 474
Digital In M1 (DS5203M1) 330
Digital In Type B (FPGA1401TP1)
DS1552 353
DS1554 394
Trigger (DS6651) 267
I/O mapping 268
parameters 268
port 268
Trigger mode
Analog In (DS6651) 242
Analog In-L (DS6651) 245
Tx Data
RS485 Rx/Tx (DS6651) 263
RS485 RxTx (DS2655M2) 232
Tx Enable
RS485 Rx/Tx (DS6651) 264
RS485 Tx (DS2655M2) 233
TX_N
MGT Out 279
TX_P
MGT Out 279

U

UART (RS422/485) (DS1302)
I/O mapping 86
UART 3 RS232 (FPGA1403TP1)
I/O mapping 454
parameters 450
port 452
UART 3 RS422/485 (FPGA1403TP1)
I/O mapping 459
parameters 455
port 457
UART mode
UART RS422/485 (DS1302) 83
UART RS422/485 (FPGA1401TP1) 377
UART RS422/485 (FPGA1403TP1) 456
UART RS232 (DS1302) 78
I/O mapping 82
parameters 78
port 79
UART RS232 (FPGA1401TP1) 370
I/O mapping 375
parameters 370
port 372
UART RS232 (FPGA1403TP1) 450
UART RS422/485 (DS1302) 82
parameters 82
port 84
UART RS422/485 (FPGA1401TP1) 375
I/O mapping 380
parameters 376
port 378
UART RS422/485 (FPGA1403TP1) 455
UART type
UART RS232 (DS1302) 79
UART RS232 (FPGA1401TP1) 372, 451
UART RS422/485 (DS1302) 84
UART RS422/485 (FPGA1401TP1) 377, 456
uart_<Channel_Number>_driver_en
UART RS422/485 (DS1302) 86
uart_<ChannelNumber>_cts
UART 3 RS232 (FPGA1403TP1) 454
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374
uart_<ChannelNumber>_driver_en
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS422/485 (FPGA1401TP1) 379
uart_<ChannelNumber>_rd
UART 3 RS232 (FPGA1403TP1) 452
UART 3 RS422/485 (FPGA1403TP1) 457
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 84
UART RS422/485 (FPGA1401TP1) 378
uart_<ChannelNumber>_rd_data
UART 3 RS232 (FPGA1403TP1) 452
UART 3 RS422/485 (FPGA1403TP1) 457
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 85
UART RS422/485 (FPGA1401TP1) 378
uart_<ChannelNumber>_rd_data_count
UART 3 RS232 (FPGA1403TP1) 452
UART 3 RS422/485 (FPGA1403TP1) 457
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 84
UART RS422/485 (FPGA1401TP1) 378
uart_<ChannelNumber>_rd_fifo_empty
UART 3 RS232 (FPGA1403TP1) 452
UART 3 RS422/485 (FPGA1403TP1) 457
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 85
UART RS422/485 (FPGA1401TP1) 378
uart_<ChannelNumber>_rts
UART 3 RS232 (FPGA1403TP1) 453
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374
uart_<ChannelNumber>_wr
UART 3 RS232 (FPGA1403TP1) 453
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 85
UART RS422/485 (FPGA1401TP1) 379
uart_<ChannelNumber>_wr_data
UART 3 RS232 (FPGA1403TP1) 453
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374
UART RS422/485 (DS1302) 86
UART RS422/485 (FPGA1401TP1) 379
uart_<ChannelNumber>_wr_data_count
UART 3 RS232 (FPGA1403TP1) 453
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374
UART RS422/485 (DS1302) 85
UART RS422/485 (FPGA1401TP1) 379
uart_<ChannelNumber>_wr_fifo_full
UART 3 RS232 (FPGA1403TP1) 453
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374
UART RS422/485 (DS1302) 86
UART RS422/485 (FPGA1401TP1) 379
Update
Resolver (DS1302) 76
usr_<ChannelNumber>_interrupt
Interrupt (DS2655) 112
Interrupt (DS5203) 315
Interrupt (DS6601) 148
Interrupt (DS6602) 194

Interrupt (FPGA1401TP1)
DS1552 360
DS1554 402
Interrupt (FPGA1403TP1)
DS1552 440
DS1554 481
usr_interrupt_<ChannelNumber>
Interrupt (DS1302) 65

V

Valid
ADC (DS5203) 294
ADC M1 (DS5203M1) 327
Resolver (DS1302) 75

W

Word length
UART RS232 (DS1302) 79
UART RS232 (FPGA1401TP1) 372
UART RS232 (FPGA1403TP1) 451
UART RS422/485 (DS1302) 83
UART RS422/485 (FPGA1401TP1) 377
UART RS422/485 (FPGA1403TP1) 456
Write Data
UART 3 RS232 (FPGA1403TP1) 453
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374
UART RS422/485 (DS1302) 86
UART RS422/485 (FPGA1401TP1) 379
Write Data Count
UART 3 RS232 (FPGA1403TP1) 453
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374
UART RS422/485 (DS1302) 85
UART RS422/485 (FPGA1401TP1) 379
Write Enable
UART 3 RS232 (FPGA1403TP1) 453
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS232 (DS1302) 80
UART RS232 (FPGA1401TP1) 373
UART RS422/485 (DS1302) 85
UART RS422/485 (FPGA1401TP1) 379
Write Fifo Full
UART 3 RS232 (FPGA1403TP1) 453
UART 3 RS422/485 (FPGA1403TP1) 458
UART RS232 (DS1302) 81
UART RS232 (FPGA1401TP1) 374
UART RS422/485 (DS1302) 86
UART RS422/485 (FPGA1401TP1) 379

X

xmem64f_<ChannelNumber>_addr
Buffer64 In (DS1302) 52
Buffer64 In (DS2655) 97
Buffer64 In (DS5203) 300
Buffer64 In (DS6601) 132
Buffer64 In (DS6602) 169
xmem64f_<ChannelNumber>_count
Buffer64 In (DS1302) 52
Buffer64 In (DS2655) 97
Buffer64 In (DS5203) 299
Buffer64 In (DS6601) 132
Buffer64 In (DS6602) 169
xmem64f_<ChannelNumber>_dout
Buffer64 In (DS1302) 52
Buffer64 In (DS2655) 97
Buffer64 In (DS5203) 299
Buffer64 In (DS6601) 132
Buffer64 In (DS6602) 169
xmem64f_<ChannelNumber>_new_data
Buffer64 In (DS1302) 52
Buffer64 In (DS2655) 97
Buffer64 In (DS5203) 300
Buffer64 In (DS6601) 133
Buffer64 In (DS6602) 169
xmem64f_<ChannelNumber>_read_req
Buffer64 In (DS2655) 97
Buffer64 In (DS6601) 133
Buffer64 In (DS6602) 169
xmem64f_inter_<ChannelNumber>_addr
I-FPGA64 In (IOCNET) (DS2655) 106
I-FPGA64 In (IOCNET) (DS6601) 142
I-FPGA64 In (IOCNET) (DS6602) 188
xmem64f_inter_<ChannelNumber>_count
I-FPGA64 In (IOCNET) (DS2655) 106
I-FPGA64 In (IOCNET) (DS6601) 142
I-FPGA64 In (IOCNET) (DS6602) 189
xmem64f_inter_<ChannelNumber>_dout
I-FPGA64 In (IOCNET) (DS2655) 106
I-FPGA64 In (IOCNET) (DS6601) 142
I-FPGA64 In (IOCNET) (DS6602) 189
xmem64f_inter_<ChannelNumber>_new_data
I-FPGA64 In (IOCNET) (DS2655) 107
I-FPGA64 In (IOCNET) (DS6601) 143
I-FPGA64 In (IOCNET) (DS6602) 189
xmem64p_<ChannelNumber>_din
Buffer64 Out (DS1302) 56
Buffer64 Out (DS2655) 101
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 137
Buffer64 Out (DS6602) 173
xmem64p_<ChannelNumber>_finished
Buffer64 Out (DS1302) 56
Buffer64 Out (DS2655) 102
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 137
Buffer64 Out (DS6602) 174
xmem64p_<ChannelNumber>_overflow
Buffer64 Out (DS1302) 56
Buffer64 Out (DS2655) 102
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 138
Buffer64 Out (DS6602) 174
xmem64p_<ChannelNumber>_send_ack
Buffer64 Out (DS2655) 102
Buffer64 Out (DS6601) 138
Buffer64 Out (DS6602) 174
xmem64p_<ChannelNumber>_write
Buffer64 Out (DS1302) 56
Buffer64 Out (DS2655) 102
Buffer64 Out (DS5203) 303
Buffer64 Out (DS6601) 137
Buffer64 Out (DS6602) 174
xmem64p_inter_<ChannelNumber>_din
I-FPGA64 Out (IOCNET) (DS2655) 110
I-FPGA64 Out (IOCNET) (DS6601) 146
I-FPGA64 Out (IOCNET) (DS6602) 192
xmem64p_inter_<ChannelNumber>_finished
I-FPGA64 Out (IOCNET) (DS2655) 110
I-FPGA64 Out (IOCNET) (DS6601) 146
I-FPGA64 Out (IOCNET) (DS6602) 192
xmem64p_inter_<ChannelNumber>_overflow
I-FPGA64 Out (IOCNET) (DS2655) 111
I-FPGA64 Out (IOCNET) (DS6601) 147
I-FPGA64 Out (IOCNET) (DS6602) 193
xmem64p_inter_<ChannelNumber>_strobe
I-FPGA64 Out (IOCNET) (DS2655) 110
I-FPGA64 Out (IOCNET) (DS6601) 146
I-FPGA64 Out (IOCNET) (DS6602) 192
xmem64p_inter_<ChannelNumber>_write
I-FPGA64 Out (IOCNET) (DS2655) 111
I-FPGA64 Out (IOCNET) (DS6601) 147
I-FPGA64 Out (IOCNET) (DS6602) 193
xmemf_<ChannelNumber>_addr
Buffer In (DS1302) 51
Buffer In (DS2655) 94
Buffer In (DS5203) 298
Buffer In (DS6601) 130
Buffer In (DS6602) 166
Buffer In (FPGA1401TP1)
DS1552 341
DS1554 386
Buffer In (FPGA1403TP1)
DS1552 421
DS1554 466
xmemf_<ChannelNumber>_count
Buffer In (DS1302) 50
Buffer In (DS2655) 94
Buffer In (DS5203) 298
Buffer In (DS6601) 130
Buffer In (DS6602) 166
Buffer In (FPGA1401TP1)
DS1552 341
DS1554 386
Buffer In (FPGA1403TP1)
DS1552 421
DS1554 466
xmemf_<ChannelNumber>_dout
Buffer In (DS1302) 50
Buffer In (DS2655) 94
Buffer In (DS5203) 298
Buffer In (DS6601) 130
Buffer In (DS6602) 166
Buffer In (FPGA1401TP1)
DS1552 341
DS1554 386
Buffer In (FPGA1403TP1)
DS1552 421
DS1554 466
xmemf_<ChannelNumber>_new_data

Buffer In (DS1302) 51
Buffer In (DS2655) 95
Buffer In (DS5203) 298
Buffer In (DS6601) 130
Buffer In (DS6602) 167
Buffer In (FPGA1401TP1)
DS1552 341
DS1554 386
Buffer In (FPGA1403TP1)
DS1552 421
DS1554 466
xmemf_<ChannelNumber>_read_req
Buffer In (DS2655) 95
Buffer In (DS6601) 130
Buffer In (DS6602) 167
xmemf_inter_<ChannelNumber>_addr
I-FPGA In (IOCNET) (DS2655) 104
I-FPGA In (IOCNET) (DS6601) 140
I-FPGA In (IOCNET) (DS6602) 187
xmemf_inter_<ChannelNumber>_count
I-FPGA In (IOCNET) (DS2655) 105
I-FPGA In (IOCNET) (DS6601) 140
I-FPGA In (IOCNET) (DS6602) 187
xmemf_inter_<ChannelNumber>_dout
I-FPGA In (IOCNET) (DS2655) 105
I-FPGA In (IOCNET) (DS6601) 140
I-FPGA In (IOCNET) (DS6602) 187
xmemf_inter_<ChannelNumber>_new_data
I-FPGA In (IOCNET) (DS2655) 105
I-FPGA In (IOCNET) (DS6601) 141
I-FPGA In (IOCNET) (DS6602) 187
xmemf64_<ChannelNumber>_addr
Buffer64 In (FPGA1401TP1)
DS1552 343
DS1554 388
Buffer64 In (FPGA1403TP1)
DS1552 423
DS1554 468
xmemf64_<ChannelNumber>_count
Buffer64 In (FPGA1401TP1)
DS1552 343
DS1554 388
Buffer64 In (FPGA1403TP1)
DS1552 422
DS1554 467
xmemf64_<ChannelNumber>_dout
Buffer64 In (FPGA1401TP1)
DS1552 343
DS1554 388
Buffer64 In (FPGA1403TP1)
DS1552 423
DS1554 468
xmemf64_<ChannelNumber>_new_data
Buffer64 In (FPGA1401TP1)
DS1552 343
DS1554 388
Buffer64 In (FPGA1403TP1)
DS1552 423
DS1554 468
xmemp_<ChannelNumber>_din
Buffer Out (DS1302) 54
Buffer Out (DS2655) 99
Buffer Out (DS5203) 301
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171
Buffer Out (FPGA1401TP1)
DS1552 344
DS1554 389
Buffer Out (FPGA1403TP1)
DS1552 424
DS1554 469
xmemp_<ChannelNumber>_finished
Buffer Out (DS1302) 54
Buffer Out (DS2655) 99
Buffer Out (DS5203) 301
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171
Buffer Out (FPGA1401TP1)
DS1552 345
DS1554 390
Buffer Out (FPGA1403TP1)
DS1552 424
DS1554 469
xmemp_<ChannelNumber>_overflow
Buffer Out (DS1302) 54
Buffer Out (DS2655) 99
Buffer Out (DS5203) 301
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171
Buffer Out (FPGA1401TP1)
DS1552 345
DS1554 390
Buffer Out (FPGA1403TP1)
DS1552 425
DS1554 470
xmemp_<ChannelNumber>_send_ack
Buffer Out (DS2655) 99
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171
xmemp_<ChannelNumber>_write
Buffer Out (DS1302) 54
Buffer Out (DS2655) 99
Buffer Out (DS5203) 301
Buffer Out (DS6601) 135
Buffer Out (DS6602) 171
Buffer Out (FPGA1401TP1)
DS1552 345
DS1554 389
Buffer Out (FPGA1403TP1)
DS1552 424
DS1554 469
xmemp_inter_<ChannelNumber>_din
I-FPGA Out (IOCNET) (DS2655) 108
I-FPGA Out (IOCNET) (DS6601) 144
I-FPGA Out (IOCNET) (DS6602) 190
xmemp_inter_<ChannelNumber>_finished
I-FPGA Out (IOCNET) (DS2655) 108
I-FPGA Out (IOCNET) (DS6601) 144
I-FPGA Out (IOCNET) (DS6602) 190
xmemp_inter_<ChannelNumber>_overflow
I-FPGA Out (IOCNET) (DS2655) 109
I-FPGA Out (IOCNET) (DS6601) 145
I-FPGA Out (IOCNET) (DS6602) 191
xmemp_inter_<ChannelNumber>_strobe
I-FPGA Out (IOCNET) (DS2655) 108
I-FPGA Out (IOCNET) (DS6601) 144
I-FPGA Out (IOCNET) (DS6602) 190
xmemp_inter_<ChannelNumber>_write
I-FPGA Out (IOCNET) (DS2655) 108
I-FPGA Out (IOCNET) (DS6601) 144
I-FPGA Out (IOCNET) (DS6602) 191
xmemp64_<ChannelNumber>_din
Buffer64 Out (FPGA1401TP1)
DS1552 346
DS1554 391
Buffer64 Out (FPGA1403TP1)
DS1552 426
DS1554 471
xmemp64_<ChannelNumber>_finished
Buffer64 Out (FPGA1401TP1)
DS1552 347
DS1554 392
Buffer64 Out (FPGA1403TP1)
DS1552 426
DS1554 471
xmemp64_<ChannelNumber>_overflow
Buffer64 Out (FPGA1401TP1)
DS1552 347
DS1554 392
Buffer64 Out (FPGA1403TP1)
DS1552 427
DS1554 472
xmemp64_<ChannelNumber>_write
Buffer64 Out (FPGA1401TP1)
DS1552 347
DS1554 391
Buffer64 Out (FPGA1403TP1)
DS1552 426
DS1554 471
xreg_<ChannelNumber>_din
Register Out (DS1302) 72
Register Out (DS2655) 118
Register Out (DS5203) 321
Register Out (DS6601) 154
Register Out (DS6602) 200
Register Out (FPGA1401TP1)
DS1552 366
DS1554 410
Register Out (FPGA1403TP1)
DS1552 445
DS1554 489
xreg_<ChannelNumber>_dout
Register In (DS1302) 69
Register In (DS2655) 115
Register In (DS5203) 317
Register In (DS6601) 151
Register In (DS6602) 197
Register In (FPGA1401TP1)
DS1552 362
DS1554 406
Register In (FPGA1403TP1)
DS1552 442
DS1554 485

xreg_<ChannelNumber>_dout_wr
Register In (DS2655) 115
Register In (DS5203) 317
Register In (DS6601) 151
Register In (DS6602) 197
Register In (FPGA1401TP1)
DS1552 362
DS1554 406
Register In (FPGA1403TP1)
DS1552 442
DS1554 485
xreg_<ChannelNumber>_wr
Register In (DS1302) 69
xreg64_<ChannelNumber>_din
Register64 Out (DS1302) 74
Register64 Out (DS2655) 120
Register64 Out (DS5203) 323
Register64 Out (DS6601) 156
Register64 Out (DS6602) 202
Register64 Out (FPGA1401TP1)
DS1552 367
DS1554 412
Register64 Out (FPGA1403TP1)
DS1552 447
DS1554 490
xreg64_<ChannelNumber>_dout
Register64 In (DS1302) 70
Register64 In (DS2655) 117
Register64 In (DS5203) 319
Register64 In (DS6601) 153
Register64 In (DS6602) 199
Register64 In (FPGA1401TP1)
DS1552 364
DS1554 408
Register64 In (FPGA1403TP1)
DS1552 444
DS1554 487
xreg64_<ChannelNumber>_dout_wr
Register64 In (DS1302) 70, 319
Register64 In (DS2655) 117
Register64 In (DS6601) 153
Register64 In (DS6602) 199
Register64 In (FPGA1401TP1)
DS1552 364
DS1554 408
Register64 In (FPGA1403TP1)
DS1552 444
DS1554 487