XSG AC Motor Control Solution

Version 21.1 – 5/2021

Embedded Success 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

There are different ways to contact dSPACE Support:

- Visit our Web site at http://www.dspace.com/goto?support
- Send an e-mail or phone:
  - General Technical Support:
    support@dspace.de
    +49 5251 1638-941
  - SystemDesk Support:
    support.systemdesk@dspace.de
    +49 5251 1638-996
  - TargetLink Support:
    support.tl@dspace.de
    +49 5251 1638-700
- Use the dSPACE Installation Manager:
  - On your dSPACE DVD at *\Tools\InstallationManager.exe*
  - Via Start – Programs – dSPACE Installation Manager (after installation of the dSPACE software)
  - At http://www.dspace.com/goto?im

  You can always find the latest version of the dSPACE Installation Manager here.

dSPACE recommends that you use the dSPACE Installation Manager to contact dSPACE Support.

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.de/goto?support for software updates and patches.

Important Notice

This document contains proprietary information that is protected by copyright. All rights are reserved. Neither the documentation nor software may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© Copyright 2018 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

This publication and the contents hereof are subject to change without notice.
AutomationDesk, CalDesk, ConfigurationDesk, ControlDesk, SCALEXIO, SystemDesk and TargetLink are registered trademarks of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations

Contents

About This Guide **7**
Document Symbols and Conventions *7*
Accessing PDF File *8*
Related Documents *9*
Requirements *10*

Target Hardware **11**
DS5203 FPGA Board 11
DS1514 FPGA Board + DS1552/DS1553 I/O Module 12
DS1302 FPGA Board (MicroLabBox) 14
DS2655 DS6601/DS6602 FPGA Board + DS2655Mx/DS6651 I/O Module 15
Compatibility Matrix 18

Simulink Model Structure **19**
General Description 19
Example of Interfacing 20
The Processor Setup Block 21
The FPGA Setup Block 21

XSG AC Motor Control Solution **23**
Library Contents 23
Description / Overview 25
Library Structure 25
Incremental Encoder Processing (TTL or Sine) 30
Processor Output 33
FPGA Main Component 38
Processor Input 41
Step/Direction Encoder Processing (TTL) 45
Processor Output 47
FPGA Main Component 52
Processor Input 55
Hall Sensor Processing 58
Processor Output 60
FPGA Main Component 67
Processor Input 70
EnDat 2.1 Processing 73
Processor Output 74
FPGA Main Component 77
Processor Input 79
Resolver & LVDT Processing 82
Processor Output 85
FPGA Main Component 90
Processor Input 94
SSI & SPI Sensor Processing 101

Processor Output ... 102
FPGA Main Component ... 106
Processor Input ... 109
SSI Sensor Processing ... 112
Processor Output ... 112
FPGA Main Component ... 118
Processor Input ... 121
BiSS Sensor Processing ... 125
Processor Output ... 125
FPGA Main Component ... 131
Processor Input ... 134
HIPERFACE® Sensor Processing ... 137
Processor Output ... 137
FPGA Main Component ... 148
Processor Input ... 150
Asymmetric H Bridge PWM Generator ... 157
Processor Output ... 159
FPGA Main Component ... 162
Processor Input ... 164
Drive Safety ... 166
Processor Output ... 167
FPGA Main Component ... 169
Processor Input ... 171
Space Vector Modulator ... 176
FPGA Main Component ... 177
Processor Input ... 178
Block Commutated PWM Generator ... 181
Processor Output ... 182
FPGA Main Component ... 190
Processor Input ... 194
Block Modulator ... 197
Processor Output ... 199
FPGA Main Component ... 205
Processor Input ... 208
PMSM Simple ... 211
Processor Output ... 212
FPGA Main Component ... 216
Processor Input ... 219
Miscellaneous ... 222
Debounce ... 222
Sync Generation ... 223
ROUTE PWM SIGNALS ... 224
Processor based model parts ... 226
Motor Inertia ... 226
Discontinued Blocks ... 228

Demos ... 230
Overview ... 230
Permanent Magnet Synchronous Machine Demo ... 233

DS1553 Framework ... 250
Introduction ... 250

I/O Read Functions 253
Status In 253
Digital In 254
ADC 255
RS485 In 257
Resolver IC 258
FPGA Temperature 261
I/O Write Functions 263
LED Out 263
Digital Out 263
DAC 265
RS485 Out 266
Pin-Out 268

About This Guide

Document Symbols and Conventions

Symbols

The following symbols may be used in this document:

	Indicates a general hazard that may cause personal injury of any kind if you do not avoid it by following the instructions given.
	Indicates the danger of electric shock which may cause death or serious injury if you do not avoid it by following the instructions given.
	Indicates a hazard that may cause material damage if you do not avoid it by following the instructions given.
	Indicates important information that should be kept in mind, for example, to avoid malfunctions.
	Indicates tips containing useful information to make your work easier.

Naming Conventions

The following abbreviations and formats are used in this document:

%name% Names enclosed in percent signs refer to environment variables for file and path names, for example, %DSPACE_PYTHON25% specifies the location of your dSPACE installation in the file system.

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

Precedes the document title in a link that refers to another document.

Indicates that a link refers to another document, which is available in dSPACE HelpDesk.

Accessing PDF File

Objective	After you install your dSPACE software, the documentation for the installed products is available as Adobe® PDF file.
PDF files	You can access the PDF files as follows: Documentation root: <%INSTALLDIR%>\Doc\XSG_ACMC.pdf

Related Documents

Below is a list of documents that you are recommended to read when working with the XSG ACMC Library.

- RTI Reference
- RTI FPGA Programming Blockset Guide
- RTI FPGA Programming Blockset - Processor Interface Reference
- RTI FPGA Programming Blockset - FPGA Interface Reference
- Hardware Installation and Configuration Reference
- DS5346 Data Sheet
- MABXII ACMC User Guide
- XSG Utils User Guide

Requirements

The following tools have to be installed for using the dSPACE FPGA Boards:

- MATLAB & Simulink
- Xilinx Vivado including System Generator (e.g. System Edition)
- dSPACE Release

Below you can find the compatibility matrix for dSPACE Release, Xilinx and MATLAB for the XSG AC Motor Control Blockset for FPGA:

RTI FPGA Programming Blockset	dSPACE Release	Operating System	MATLAB	Xilinx Vivado and XSG
3.11	2021-A	Windows 10 64-bit	R2019b R2020a R2020b (64 Bit)	2020.2 (64 Bit)

In the following table you find the compatibility matrix for dSPACE Release and MATLAB for the XSG AC Motor Control Blockset for Processor:

dSPACE Release	Operating System	MATLAB
2021-A (64 bit)	Windows 10 (64 bit)	R2019b R2020a R2020b R2021a (64 bit)

Target Hardware

DS5203 FPGA Board

Board description

With the DS5203 FPGA board (shown in the figure below), an integration of a FPGA application into a dSPACE modular PHS-Bus based system can be performed.

Figure 1: DS5203 Boards (PHS Based)

The board provides a Xilinx® Kintex-7 FPGA which can be programmed by the user by using the RTI FPGA Programming Blockset. The board features the following onboard I/O:

- 6 A/D channels with 10 MHz sample rate and adjustable voltage range
- 6 D/A channels with 10 MHz update rate and ±10 V voltage range
- 16 digital single-ended input or output channels with configurable input voltage threshold and output voltage of 3.3 V or 5 V

An additional piggy back module (M1) to double the I/O can also be added.

There are two FPGA types available:

- Kintex-7 325T (normal amount of recourses)
- Kintex-7 410T (larger amount of recourses)

The I/O of the DS5203 enables most of the features provided by the XSG AC Motor Control Solution. However, the processing of SSI sensors is not possible with the standard hardware, as it required RS485 or RS422 digital I/O drivers.

DS1514 FPGA Board + DS1552/DS1553 I/O Module

Board description

Using the DS1514 FPGA base board for MicroAutoBox II (DS1401) and MicroAutobox III (DS1403) in combination with the DS1552 Multi-I/O piggy-back module or the DS1553 ACMC piggy-back module (shown in the figure below), an integration of a FPGA application into a dSPACE MicroAutoBox can be performed.

Figure 2: MicroAutoBox II (DS1401) and MicroAutoBox III (DS1403) with DS1514 and DS1553

The board provides a Xilinx® Kintex-7 FPGA which can be programmed by the user, for example, by using the RTI FPGA Programming Blockset.

The DS1553 I/O module features the following I/O:

- 8 A/D channels with 10 MHz sample rate and adjustable voltage range
- 2 D/A channels with 10 MHz update rate and ±20 V voltage range
- 8 digital single-ended or differential 5V input channels
- 24 digital single-ended 5 V output channels
- 4 RS485 digital differential bi-directional channels
- 1 resolver IC with 16 bit position resolution and 500 kHz update rate

With the I/O provided by the DS1553, all features of the XSG AC Motor Control Solution can be applied.

The DS1552 I/O module features the following I/O:

- 8 A/D channels with a sample rate of 1 MHz and 0…5 V or ±10 V voltage range
- 16 A/D channels with a sample rate of 200 kHz and ±10 V voltage range
- 4 D/A channels with an update rate of 2.1 MHz and 0…5 V voltage range
- 16 digital single-ended 5 V input channels
- 16 digital single-ended output channels with configurable output voltage
- 8 digital single-ended input or output channels with configurable input voltage threshold and output voltage of 3.3 V or 5 V
- 3 crank/cam input channels with ±40 V voltage range

- 1 Inductive zero voltage detector (for zero-crossing detection)
- 4 UART interfaces

The I/O of DS1552 enables most features of the XSG AC Motor Control. However, the DAC range is not suitable for all kinds of resolvers and LVDT. Additionally, the processing of SSI and EnDat sensors is not possible with the standard framework.

DS1302 FPGA Board (MicroLabBox)

Board description

Using the DS1302 FPGA board for MicroLabBox (shown in the figure below) with on-board I/O, an integration of a FPGA application into a dSPACE MicroLabBox can be performed.

Figure 3: MicroLabBox with DS1302

The DS1302 FPGA board provides a Xilinx® Kintex-7 FPGA which can be programmed by the user by using the RTI FPGA Programming Blockset. It features the following I/O:

- 8 A/D channels with 10 MHz sample rate and ±10 V voltage range
- 24 A/D channels 1 MHz sample rate and ±10 V voltage range
- 16 D/A channels with 1 MHz update rate and ±10 V voltage range
- 48 digital single-ended input or output channels with configurable input voltage threshold and output voltage of 2.5V, 3.3 V or 5 V
- 12 digital differential RS485/RS422 channels

With the I/O provided by the MicroLabBox, all features of the XSG AC Motor Control Solution can be applied.

DS2655 DS6601/DS6602 FPGA Board + DS2655Mx/DS6651 I/O Module

Board description

With the DS2655/DS6601/DS6602 FPGA base module and the DS2655M1 I/O module / DS2655M2/ DS6651 I/O module (shown in the figure below), an integration of a FPGA application into a dSPACE modular IOCNET based system (SCALEXIO) can be performed.

Figure 4: DS2655 FPGA base module and the DS2655M1 I/O module / DS2655M2 I/O module (SCALEXIO)

The FPGA base modules (DS2655, DS6601, DS6602) provide a Xilinx® Kintex-7 FPGA which can be programmed by the user by using the RTI FPGA Programming Blockset. Up to 5 I/O modules can be connected to the base module. There are four FPGA base modules available:

- DS2655 Kintex7 K160
  (Xilinx Tool: Vivado + standard amount of resources)
- DS2655 Kintex7 K410
  (Xilinx Tool: Vivado + huge amount of resources)
- DS6601 Kintex Ultrascale KU035
  (Xilinx Tool: Vivado + huge amount of resources)
- DS6602 Kintex Ultrascale KU15P
  (Xilinx Tool: Vivado + very huge amount of resources)

	DS2655 (7K160)	DS2655 (7K410)	DS6601	DS6602
Release	RLS2013-A	RLS2017-B	Planned for RLS2019-A	Planned for RLS2019-A
Positioning	Allrounder	(interims) high-end	Allrounder Successor of the DS2655 7K160	High-end
FPGA technology	Kintex-7	Kintex-7	Kintex Ultrascale	Kintex Ultrascale+
Relative size	1	2.5	2	5
FPGA	Xilinx Kintex-7 160T • Logic cells: 162240 ◦ DSP slices: 600 • Dist. RAM: 2188 kBits • Block RAM: 11700 kBits	Xilinx Kintex-7 410T • Logic cells: 406720 ◦ DSP slices: 1540 • Dist. RAM: 5663 kBits • Block RAM: 28620 kBits	Xilinx Kintex KU035 • System Logic Cells: 444000 ◦ DSP slices: 1700 • Dist. RAM: 5908 kBits • Block RAM: 19000 kBits	Xilinx Kintex KU15P • System Logic cells: 1143000 ◦ DSP slices: 1968 • Dist. RAM: 9800 kBits • Block RAM: 34600 kBits • UltraRAM: 36000 kBits
Additional RAM onboard	--	--	--	4 GByte
Further options	--	--	Multi-gigabit transceivers are available on a dedicated connector	Multi-gigabit transceivers are available on a dedicated connector

Note: The counting of "logic cells" changed from Kintex-7 ("logic cells") to Kintex UltraScale ("system logic cells").

Figure 5: Feature Comparison DS2555 (7K160), DS2555 (7K410), DS6601 and DS6602 FPGA base boards

These base modules can each be configured with up to 5 of the below listed IO modules. A configuration of different types of IO modules for the same base board is support!

DS2655M1:

- 5 A/D channels with 2 MHz sample rate and adjustable voltage range
- 5 D/A channels with 7.8125 MHz update rate and ±10 V voltage range
- 10 digital single-ended input or output channels with configurable input voltage threshold and output voltage of 3.3 V or 5 V

The DS2655M2 I/O module features 32 digital I/O channels with the following features:

DS2655M2:

- Push-Pull drivers
- One output voltage can be selected for all channels: 3.3V or 5V
- 16 of the channels are extended with RS485/RS232 transceivers
- RS232: max. 250 kBaud
- RS485: max. 40 MBaud
- Max. of following functions are possible:
  - 32 x digital input
  - 32 x digital output (push/pull or push or pull)
  - 16 x digital output (push/pull/tristate)
  - 8 x RS232 RX (24 digital I/O or 8 x RS232 TX channels are free)
  - 8 x RS232 TX (24 digital I/O or 8 x RS232 RX channels are free)
  - 8 x RS485 RX (16 digital I/O channels are free)
  - 8 x RS485 TX (16 digital I/O channels are free)

- 8 x RS485 RX/TX (8 digital I/O channels are free)

DS6651:

- 16 digital single-ended input or output channels with configurable input voltage threshold and output voltage of 3.3 V or 5 V
  - Can be configured to up to 8 EIA 422 differential channel with max. 16 MBaud
- 6 A/D Channels with adjustable voltage range and 5 MHz update rate
- 6 D/A Channels with 10.4 MHz update rate and ±10V Voltage range
- 2 × power supply for external devices (+5V, 500mA)

Compatibility Matrix

Description

The following table shows the compatibility of XSG AC Motor Control Solution components in combination with the supported hardware.

Component	DS5203	DS1514 DS1552	DS1514 DS1553	DS1302	DS2655 M1/M2/DS 6651
Incr. encoder	✓	✓	✓	✓	✓
Step direction encoder	✓	✓	✓	✓	✓
Hall sensor	✓	✓	✓	✓	✓
Resolver & LVDT	✓	(✓)	✓	✓	✓
SSI	-	-	✓	✓	✓
EnDat 2.1	-	-	✓	✓	✓
SPI	✓	✓	✓	✓	✓
PWM generation*	✓	✓	✓	✓	✓
Miscellaneous	✓	✓	✓	✓	✓

Components marked with an asterisk (*) are located in the XSG Utils library.

Simulink Model Structure

General Description

Description

This chapter describes the user interface for accessing the FPGA board from Simulink. In general, there are two different ways to access the board, either via bus for communication between processor and FPGA or via digital or analog in channels on the board. The FPGA output is similar to its input: either information can be written from the FPGA to the processor via bus register, or information can be sent out via hardware channels.

Two interfaces are available for handling the communication.

- **Processor interface** (read and write on the processor side)
- **FPGA interface** (read and write on the FPGA side)

The processor interface comes with the regular RTI license, the FPGA interface comes with an extra RTI FPGA license. The figure below shows the library with FPGA and processor interface.

Figure 6: RTIFPGA library

	Separating these two libraries makes it possible to use precompiled FPGA applications without buying an additional license for FPGA modeling.

The XSG AC Motor Control Solution contents are designed to give you easy access to create the model that you require. Each main component blockset consists of five elements:

1. **Processor out interface** (<component>_out (Processor Interface))
   All the required parameters and actual values are merged into the smallest necessary number of registers or buffers.
2. **FPGA in interface** (<component>_in (FPGA Interface))
   The merged signals from the processor side are decoded on the FPGA side
3. **FPGA main component** (<component>_FPGA (FPGA))
   Provides the actual model functionality of the component
4. **FPGA out interface** (<component>_out (Processor Interface))
   All the required signals to be returned to the processor are merged here to utilize the smallest possible amount of registers
5. **Processer in interface** (<component>_in (Processor Interface))
   The merged signals from the FPGA side are decoded on the processor side

Example of Interfacing

Example

The illustration below shows a structural example of how to implement the interfaces. The red blocks (1-5) are components from the dSPACE XSG AC Motor Control Solution. The other blocks (6-9) are RTI FPGA interface blocks that define the basic communication between FPGA, processor and I/O.

Figure 7: Using library components

A detailed view of the RTI interface is shown in the next figure below. In this example, the values calculated by the processor are written via bus interface to the FPGA (on the left-hand side) and the values calculated on the FPGA are written back to the processor (right-hand side). The I/O access is also shown (ADC1, DAC1).

Figure 8: RTI interface usage

The Processor Setup Block

Features

	The Processor Setup block has to be located on the model root. All processor interface blocks can be placed anywhere inside the Simulink model except the FPGA subsystem. This block is not required for IOCNET platforms.

The number of FPGA boards which are used in the application has to be defined on the unit page, which is called double-clicking the processor setup block PROC_SETUP_BL1. Up to 16 FPGA boards can be handled. The subsystem, or precompiled binary file, has to be specified for each board. On the FPGA side the configuration can be stored in either the RAM or the flash memory.

The Interface lets you generate the interface blocks on the processor side automatically. Therefore only the interface blocks on the FPGA side have to be configured.

The Model Configuration page lets you switch between the FPGA Build and the Processor Build mode. For offline simulation, the FPGA build option must be used. Changes take effect when the Switch model mode button is clicked. If the Processor Build mode is selected, the FPGA subsystem is removed from the model file and stored inside a separate model file. When you switch back to FPGA Build mode, the FPGA subsystem is copied back to the application model.

	When performing the build process on the processor side, it is mandatory to activate the Processor Build mode.

On the Advanced page, precompiled FPGA applications (*.ini) can be added. After adding them there, you can select the application on the Unit page. The FPGA application is added to the generated processor files during the processor build process.

	Because of the long synthesis time and the huge resource consumption of the FPGA build process, it is recommended to use a separate PC. The synthesis result (*.ini file), can be linked to the application during a separate processor build on the modeling PC.

The FPGA Setup Block

Features

	All blocks belonging either to the Xilinx System Generator (XSG) or to the RTI FPGA Programming Blockset have to be placed in one Simulink subsystem if they are assigned to the same FPGA board.

	The root directory of the FPGA subsystem must contain an FPGA Setup block from the RTIFPGA Library

The Unit page, which is opened by double-clicking the FPGA setup block, displays the board number on which the application is stored. You must also select the framework and the piggyback.

The Parameter page shows the clock period of the FPGA. The parameters for the down-sample factor and the offline simulation period can be specified here. To start a timing analysis, an FPGA Build process or a HDL simulation, click the Execute button.

	Before starting an FPGA Build process (synthesis), it is recommended to perform a timing analysis in advance to ensure that the modeled design fits the timing constraints and the resources of the target FPGA. It is also recommended to verify changes in the offline simulation before starting the synthesis

	If inserting RTI FPGA interface blocks manually, ensure that all write/read registers are in an unsigned fixed format with a binary point of zero (UFix_32_0). Please note that the default setting is signed!

XSG AC Motor Control Solution

Library Contents

Description / Overview	The XSG AC Motor Control Solution is designed to provide easy access for rapid control prototyping of e-drive control or parts of it on Xilinx FPGAs. This includes commonly used functions like the processing of various position sensors, the generation of PWM pattern and components of the control algorithms.
Sensor Processing	Sensor processing for the following encoders: ▪ Incremental encoder (TTL or sine) ▪ Hall sensors ▪ Resolver and LVDT ▪ Step Direction Encoder ▪ EnDat encoder ▪ SSI encoders ▪ BiSS encoder ▪ HIPERFACE® encoder
Modulation	Modulation has the following modulation blocks: ▪ Block Modulator ▪ Space vector Modulator
Asym H-Bridge	Asym H-Bridge has the following PWM generators: ▪ 2 Phase ▪ 3 Phase ▪ 5 Phase
Safety	Safety has the following safety features: ▪ Drive Safety
Miscellaneous	Miscellaneous has the following functionality: ▪ Debounce ▪ Sync Generation ▪ Route PWM

eMotor Sim	eMotor simulation has the following motor simulations: ▪ PMSM Simple
Utils	The library also contains a separate library called XSG_Utils. This library collected frequently used elements, which are relevant for all e-drive use cases (e.g. PWM generation and measurement, PI controller, digital filters, Clarke/Park transformations, scaling and scope functionalities). All FPGA main components include interfaces to the processor model. For further information about these blocks, please refer to the XSG Utils documentation.
Demos	The XSG ACMC library also contains two separate demo models to illustrate an exemplary model structure of an FPGA application in combination with a control loop implemented on the processor platform. Therefore the following models are prepared: ▪ PMSM control for PHS based system ▪ PMSM control for Scalexio based system

Description / Overview

Objective

The following section describes the general structure and features of the XSG ACMC Library

Library Structure

General information

The XSG AC Motor Control Solution is divided into:

- XSG AC Motor Control Solution (FPGA-based blocks)
- XSG ACMC Interface Library (processor-based blocks)

The XSG AC Motor Control Solution contains the separate XSG Utils Library. The following figure illustrates the overall structure of the solution package.

Figure 9: XSG ACMC Library

To realize an easy user access to the different functionalities of the XSG Utils Library from the XSG ACMC Library hyperlinks are used. For further information about the blocks of the XSG Utils Library please find the XSG Utils Library documentation.

XSG ACMC Library

The XSG ACMC Library provides access to FPGA based position sensor processing. Combined with the PWM functionality of the XSG Utils Library and the I/O functions of the RTI FPGA Programming Blockset, closed-loop control for different kinds of e-drives can easily be realized. The library is divided into FPGA / Processor Interface and FPGA components as shown in the figure below. To open the library, type “XSG_ACMC_lib” in the MATLAB Command Window or access the library via Simulink browser.

Figure 10: XSG AC Motor Control Solution

The FPGA-based blocks are located in the FPGA subsystem. To simplify the realization of an interface for sending processor-based parameters to the FPGA and back, optimized FPGA and processor interface blocks are located in the subsystems FPGA Interface and Processor Interface.

	The XSG AC Motor Control Solution also contains the XSG ACMC Interface Library.

The following illustration gives you an overview of the internal structure of the library.

Figure 11: Overview of the XSG ACMC Library

The **FPGA**-based blocks are in the FPGA subsystem. To simplify the realization of an interface for sending processor-based parameters to the FPGA and back, optimized FPGA and processor interface blocks are in the subsystem FPGA Interface and Processor Interface. All the systems are organized in logical groups like “Sensor Processing”, “Modulators”, etc. All processor blocks are in the XSG ACMC Interface Library.

To simplify the overall timing of the library blocks in a customer model, all FPGA based library blocks have a block description. This description gives information about the overall block latency and the down sampling of the internal model structure as shown in the figure below.

Figure 12: Block information example (PMSM Simple): Latency: 23; internal model down sampling: 23

Some of the XSG_ACMC block accept Floating point arithmetic type and this can be identified with the logo ‘XF’ as shown below. Also, the input ports with color green signifies that the signal connected to this port can be of Signed, Unsigned or Floating arithmetic type; the color ‘red’ signifies that the signal should be of type Signed/Unsigned; the color blue signifies that the signal should be of Floating-point type.

Figure 13: Block information example (Space Vector Modulator): Floating point

For an example of the general structure of an FPGA programming model, refer to the Demos subsystem.

XSG ACMC Interface Library

If no user-defined adjustments have to be made in the FPGA code, the processor interface is separately located in the XSG ACMC Interface Library as shown in the figure below. To open the Processor Interface Library, type “XSG_ACMCInterface_lib” in the MATLAB Command Window or access the library via Simulink browser.

.

Figure 14: XSG ACMC Interface Library

The internal structure of the library is the same as the structure of the XSG ACMC Library, except that the interface library only contains the processor blocks.

Incremental Encoder Processing (TTL or Sine)

Objective

The Incremental Encoder Processing is used to derive position (angle) and velocity by decoding the output of incremental encoders. Different kinds of incremental encoders can be analyzed, dependent on the I/O connected. The most common sensors are TTL/HTL and sine encoders.
The incremental encoder processing outputs position and speed. For speed measurement, two different methods are implemented, which are described below.

TTL Encoders

Incremental TTL encoders provide angular information as rectangle pulses with a frequency dependent on the current velocity. Using two identical tracks which are shifted by 90° the direction can be determined as well. The third track is the so-called index track which provides only one pulse per revolution, which enables determination of the absolute mechanical position. Prior to detection of the index pulse (worst case after one mechanical revolution) only the relative position and the velocity can be determined.
Typical high level output voltages of TTL encoders are 5V or 3.3V. Both single-ended as well as differential signals are common. HTL encoders use higher voltages of usually 24V.

Figure 151: Incremental TTL encoder shape

Sine Encoders

In contrast to TTL encoders, sine encoders output analog sine wave signals. By counting the zero crossings of the signals a position measurement like using TTL encoders can be performed. By additionally calculating the arc tangent of A/B, a finer position interpolation can be achieved. Like TTL encoders, sine encoders usually provide an index (also called reference) track for absolute position measurement as well. The index can be a digital rectangle pulse or an analog half sine wave with a width of usually 90°.

A common voltage output of a sine encoder is an analog signal of 1 V_{PP} with a DC offset of 2.5 V_{DC}.

Figure 162: Incremental sine encoder shape

	At high motor speeds and/or high line count, it is possible that the A/D signal has too much noise and because of that sine/cosine processing can lose line information. To prevent that from happening it is advisable to filter the A/D inputs with PT1 filters to improve signal quality.

Speed Measurement

Two methods of velocity measurement are implemented for the incremental encoder processing:

- Position increment based measurement: The time elapsed between two identical edges of the input signals (360°) is evaluated. The measurement dead time is minimized, but dependent on the current speed. For high velocities, the dead time is negligible. However, the accuracy decreases, as jitter becomes significant for short measurement intervals.
- Sample time based measurement: The measurement time is defined by an external trigger, which is by default triggered with each sample of the processor output block. The number of edges during this period is counted, as well as the time elapsed between the first and the last edge considered. This method provides a relatively constant measurement dead time and more accurate results for high velocities.

The principle of the sample time based measurement is illustrated in the figure below.

Figure 17: Principle of the sample time-based speed measurement

Content

The blockset contains the following elements:

- Processor Interface: INC_ENCODER_RCP_out (Processor Interface)
- FPGA Interface: INC_ENCODER_RCP_in (FPGA Interface)
- FPGA: INC_ENCODER_RCP (FPGA Main Component)
- FPGA Interface: INC_ENCODER_RCP_out (FPGA Interface)
- Processor Interface: INC_ENCODER_RCP_in (Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

Figure 18: INC_ENCODER_RCP_out block

Block Dialog

The processor output block provides the following dialog:

Figure 19: INC_ENCODER_RCP_out dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Input type of A and B	-	0 = digital input signals 1 = analog input signals	0|1
Debounce time	s	Optional debounce time for digital input signals	Dependent on FPGA clock rate (@100MHz: Range 0…10.23µs Resolution 10ns)
DC offset	LSB (Bit)	In case of analog signals (A, B) a DC offset (usually 2.5V) can be specified	Range: 0…65535 Resolution: 1
Symmetrical hysteresis	LSB (Bit)	In case of analog signals a hysteresis can be defined to avoid multiple zero crossing detection caused by noise. A positive zero crossing will be detected when the input exceeds the specified hysteresis value, a negative zero crossing will be detected when the inputs falls below the negation of the specified hysteresis value.	Range: 0…1023 Resolution: 1
Fine interpolation	-	In case of analog sine inputs, the position can be interpolated by calculating arctan(A/B) to increase accuracy	0|1
Input type of index	-	0 = digital index signal 1 = analog index signal	0|1
Threshold for analog index	LSB (Bit)	In case of an analog index, a threshold for detecting the index can be specified	Range: 0…65535 Resolution: 2
Edges used for speed measurement	-	Select which edges are used for speed measurement. Each measurement always uses a whole period of the signal. Each time a new valid measurement is completed, this measurement result is latched to the output.	0|1 (4x)

Use sample time based speed measurement if applicable	-	If this option is deactivated, the speed measurement will be performed by measuring the time elapsed between two identical edges. If this option is activated, the speed is measured over multiple all periods of the incremental encoder signals which are passed within the last simulation step of this block. This method is not applicable if less than 1 whole period is passed in 1 simulation step. In this case the standard method will be applied.	0|1
Resolution option for sample time based speed measurement	-	Using this option, the resolution of a sample time based speed measurement can be specified. The option should be set according to the maximum possible sample time of this block. The maximum sample time calculates as $T_{S\|FPGA} \cdot (2^{(16+x)} - 2)$ For DS5203 i.e. (100MHz) this means: 0: T < 0.655ms, res. 10ns 1: T < 1.310ms, res. 20ns 2: T < 2.621ms, res. 40ns 3: T < 5.242ms, res. 80ns 4: T < 10.48ms, res.160ns 5: T < 20.97ms, res.320ns 6: T < 41.94ms, res.640ns 7: T < 83.89ms,res.1,28µs	Range: 0…7 Resolution: 1
Index position	°	The absolute mechanical position of the index signal rising edge during forward movement.	Range: 0…360° Resolution: 0.005°
Position for Interrupt trigger	°	Specifies a position for an optional position interrupt on the FPGA	Range: 0…360° Resolution: 0.005°
Number of lines	-	Number of lines of the incremental encoder.	Range: 1…16384 Resolution: 1
Sample time FPGA	s	Sample time of the FPGA	-

Input

The INC_ENCODER_RCP_out block has the following inputs:

Name	Unit	Description	Range
Enable	-	Sensor processing enable	0|1

Output

The Processor Out block outputs four register contents, mapped to two registers by time multiplexing. The sectioning is shown below:

Register 1.1

Figure 20: INC_ENCODER_RCP_out Register 1.1

Name	Bits	Description
Enable	0	Enables the encoder processing
Use_A_Rising	1	Use rising edge of input signal A for speed measurement
Use_A_Falling	2	Use rising edge of input signal A for speed measurement
Use_B_Rising	3	Use rising edge of input signal B for speed measurement
Use_B_Falling	4	Use rising edge of input signal B for speed measurement
Resolution_Option	7..5	The resolution option described for the processor interface
Use_TS_Speed	8	Activate the sample time based speed measurement described for the processor interface
Position_Increment	30..9	The increment of the position at each edge of an input signal. The FPGA main component section for details.

Register 1.2

Figure 21: INC_ENCODER_RCP_out Register 1.2

Name	Bits	Description
Enable	0	Enables the encoder processing
No_of_Lines-1	14..1	The number of lines of the encoder - 1
Index_Position	30..15	The Position of the index signal scaled to 16 bit

Register 2.1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DC_Offset															Index_Threshold															

Figure 22: INC_ENCODER_RCP_out Register 2.1

Name	Bits	Description
Index_Threshold	14..0	The threshold for detecting an analog index. 1 LSB in the register corresponds to 2 LSB set in the processor interface GUI and supplied at the FPGA main component input.
DC_Offset	30..15	The DC offset of analog input signals (A, B).

Register 2.2

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1		Debounce_Time / Hysteresis											Use_Fine_Pos	Analog_Index	Analog_AB	Interrupt_Position															

Figure 23: INC_ENCODER_RCP_out Register 2.2

Name	Bits	Description
Interrupt_Position	15..0	Position for an optional position interrupt, scaled to 16 bit.
Analog_AB	16	Input signals A & B are analog (otherwise digital)
Analog_Index	17	Index signal is analog (otherwise digital)
Use_Fine_Position	18	Use interpolation for sin encoders by arctan(A/B)
Debounce_Time | Hystereis	28..19	Debounce time for input digital signals or hysteresis for analog input signals
SPARE	30..29	-

FPGA Main Component

Block

Figure 24: INC_ENCODER_RCP FPGA Main block

Block Dialog

You can define the parameters of the input and output registers for automatic interface generation. The dialog also contains a button to start interface generation. For more details about automatic interface generation, refer to the XSG Utils documentation chapter Automatic Interface Generation. The FPGA clock period can be configured too.

Input

The main block has the following inputs:

Name	Unit	Description	Format
A	- | Bit	Input signal A. The signal can either be digital (Boolean) or analog (signed integer, e.g. raw output from ADC)	Bool | Fix_16_0
B	- | Bit	Input signal B. The signal can either be digital (Boolean) or analog (signed integer, e.g. raw output from ADC)	Bool | Fix_16_0
Index	- | Bit	Index signal. The signal can either be digital (Boolean) or analog (signed integer, e.g. raw output from ADC)	Bool | Fix_16_0
Enable	-	Enables the encoder processing	Bool
Trig_Speed_Meas	-	Triggers the sample time based speed measurement	Bool
Speed_Signals	-	Vector containing the flags [A rising, A falling, B rising, B falling] for the signal edges considered for speed measurement	Bool (4x)
Index_Position	-	Mechanical position of the index (rising edge during forward movement), scaled to 16 bit	UFix_16_0
Interrupt_Position	-	Position for optional generation of a position interrupt, scaled to 16 bit	UFix_16_0
Position_Increment	-	The increment which has to be added to or subtracted from the current position (position scaled to 16 bit) in case of a signal edge	UFix_22_8
Resolution_Option	-	The resolution option described for the processor interface	UFix_3_0
Use_Ts_Based_Speed	-	Activates sample time based speed measurement	Bool
No_of_Lines-1	-	The number of lines of the encoder - 1	UFix_14_0
Debounce_Time	Clocks	The debounce time of signal inputs.	UFix_10_0
Analog_AB	-	Signals A&B are analog when 1 (otherwise digital)	Bool
Analog_Index	-	Index is analog when 1 (otherwise digital)	Bool
Analog_DC_Offset	Bit	DC offset for analog signals A&B	UFix_16_0
Index_Threshold	Bit	In case of an analog index, a threshold for detecting the index can be specified	UFix_16_0
Use_Fine_Position	-	Activates arctan interpolation for sine input signals	Bool

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals which are passed to the processor interface	Bus
Position	-	The calculated angle (0…360°), scaled to 0…1.	UFix_16_16
PositionHD		The calculated angle (0…360°), scaled to 0…1 in high definition.	UFix_32_32
Position_Valid	-	The valid flag is set to 1 after the index has been passed at least once and the position is referenced	Bool
Direction	-	The detected direction of the encoder movement. 0 = forward, 1 = backward	Bool
Pos_Delta_Based_Time	-	The number of clock cycles between two identical edges	UFix_28_0
Pos_Delta_Based_Overfl	-	Indicates that the time between two edges exceeds the maximum range	Bool
TS_Based_Time	Clocks	The number of clock cycles for the last speed sample period	UFix_23_0
TS_Based_Incr	Edges	The number of edges detected for the last speed sample period. A negative number of edges indicates backward movement	Fix_16_0
TS_Based_Overflow	-	Indicates that the measurement time exceeds the maximum range	Bool
Index_Crossing	-	Impulse indicating the detection of an index passing	Bool
Position_Interrupt	-	Impulse indicating that the specified interrupt position has been crossed. The output can be connected to an interrupt block of the RTI FPGA Programming Blockset if required.	Bool

Processor Input

Block

Adapts the FPGA signals for the processor side.

Figure 25: INC_ENCODER_RCP_in block

Block Dialog

The following parameters can be configured in the processor input dialog:

Figure 26: INC_ENCODER_RCP_in dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Sample time FPGA	s	Sample time of the FPGA	-

Input

The processor input block has the following inputs:

Register 1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Use_Ts_Speed	No_of_Lines-1														Position_Valid	Position															

Figure 27: INC_ENCODER_PROCESSING_in Register 1

Name	Bits	Description
Position	15..0	The calculated angle, scaled to 16 bit
Position_Valid	1	The position valid flag (see main component)
No_of_Lines-1	30..17	The number of lines is passed back to the processor application
Use_Ts_Speed	31	The flag for using the sample time based speed is passed back to the processor application

Register 2

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direction	Resolution_Opt	Pos_Delta_Based_Time																													

Figure 28: INC_ENCODER_RCP_in Register 2

Name	Bits	Description
Pos_Delta_Based_Time	27..0	The number of clock cycles between two identical edges
Resolution_Option	30..28	The resolution option is passed back to the processor application
Direction	31	The direction bit (0 = forward, 1 = backward)

Register 3

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ts_Based_Incr																Ts_Based_Time															

Figure 29: INC_ENCODER_RCP_in Register 3

Name	Bits	Description
Ts_Based_Time	15..0	Clock cycles for the last speed sample period
Ts_Based_Incr	31..16	Edges detected for the last speed sample period

Output

The INC_ENCODER_RCP_in block has the following outputs:

Name	Unit	Description	Range
Position	°	The angle calculated by the FPGA main component	Range: 0…360° Resolution: 0.005°
Speed	rpm	The speed calculated by the FPGA main component	Dependent on FPGA clock rate (@100MHz: Range: >±0.17rpm Resolution: 10ns)
Position_Valid	-	The valid flag, which is raised as soon as the position is referenced (index is passed at least disable)	0|1

Step/Direction Encoder Processing (TTL)

Objective

The Step/Direction Processing is used to derive position (angle) and velocity by decoding the output of step/direction encoders.
The Step/Direction encoder processing outputs position and speed. For speed measurement, two different methods are implemented, which are described below.

Step/Direction Encoders

Step/Direction encoders provide angular information as rectangle pulses with a frequency dependent on the current velocity. The second pulse, the so called direction track, indicates the direction of the circular movement. The third track is the so-called index track which provides only one pulse per revolution, which enables determination of the absolute mechanical position. Prior to detection of the index pulse (worst case after one mechanical revolution) only the relative position and the velocity can be determined.
Typical high level output voltages of this kind of encoders are 5V or 3.3V.

Figure 30: Step/Direction encoder signal shape

Speed Measurement

Two methods of velocity measurement are implemented for the incremental encoder processing:

- Position increment based measurement: The time elapsed between two identical edges of the input signals (360°) is evaluated. The measurement dead time is minimized, but dependent on the current speed. For high velocities, the dead time is negligible. However, the accuracy decreases, as jitter becomes significant for short measurement intervals.
- Sample time based measurement: The measurement time is defined by an external trigger, which is by default triggered with each sample of the processor output block. The number of edges during this period is counted, as well as the time elapsed between the first and the last edge considered. This method provides a relatively constant measurement dead time and more accurate results for high velocities.

The principle of the sample time based measurement is illustrated in the figure below.

Figure 31: Principle of the sample time-based speed measurement

Content

The blockset contains the following elements:

- Processor Interface: STEP_DIR_ENCODER_RCP_out (Processor Interface)
- FPGA Interface: STEP_DIR_ENCODER_RCP_in (FPGA Interface)
- FPGA: STEP_DIR_ENCODER_RCP (FPGA Main Component)
- FPGA Interface: STEP_DIR_ENCODER_RCP_out (FPGA Interface)
- Processor Interface: STEP_DIR_ENCODER_RCP_in (Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

Figure 32: STEP_DIR_ENCODER_RCP_out block

Block Dialog

The processor output block provides the following dialog:

Figure 33: STEP_DIR_ENCODER_RCP_out dialog

The blockset dialog has the following parameters:

ame	Unit	Description	Range
Debounce time	s	Optional debounce time for digital input signals	Dependent on FPGA clock rate (@100MHz: Range 0…10.23µs Resolution 10ns)
Use sample time based speed measurement if applicable	-	If this option is deactivated, the speed measurement will be performed by measuring the time elapsed between two identical edges. If this option is activated, the speed is measured over multiple all periods of the incremental encoder signals which are passed within the last simulation step of this block. This method is not applicable if less than 1 whole period is passed in 1 simulation step. In this case the standard method will be applied.	0|1
Resolution option for sample time based speed measurement	-	Using this option, the resolution of a sample time based speed measurement can be specified. The option should be set according to the maximum possible sample time of this block. The maximum sample time calculates as $T_{S\|FPGA} \cdot (2^{(16+x)} - 2)$ For DS5203 i.e. (100MHz) this means: 0: T < 0.655ms, res. 10ns 1: T < 1.310ms, res. 20ns 2: T < 2.621ms, res. 40ns 3: T < 5.242ms, res. 80ns 4: T < 10.48ms, res.160ns 5: T < 20.97ms, res.320ns 6: T < 41.94ms, res.640ns 7: T < 83.89ms,res.1,28µs	Range: 0…7 Resolution: 1
Index position	°	The absolute mechanical position of the index signal rising edge during forward movement.	Range: 0…360° Resolution: 0.005°
Position for Interrupt trigger	°	Specifies a position for an optional position interrupt on the FPGA	Range: 0…360° Resolution: 0.005°

Define direction bit		Specifies how the direction bit of the step/direction encoder has to be interpreted. 0 interprets a high signal as CCW (positive direction) and a low signal as CW (negative direction). 1 interprets a high signal as CW (negative direction) and a low signal as CCW (positive direction).	0|1
Index Zero Crossing Mode		Specifies on which signal edge of the index signal the zero crossing is detected. 0: direction dependent (CCW: rising edge, CW: falling edge) 1: always on rising edge 2: always on falling edge	0|1|2
Number of steps	-	Number of steps of the step/direction encoder.	Range: 1…32768 Resolution: 1
Sample time FPGA	s	Sample time of the FPGA	-

Input

The STEP_DIR_ENCODER_RCP_out block has the following inputs:

Name	Unit	Description	Range
Enable	-	Sensor processing enable	0|1

Output

The Processor Out block outputs three register contents, for register 1 two registers are processed by time multiplexing. The sectioning is shown below:

Register 1.1

Figure 34: STEP_DIR_ENCODER_RCP_out Register 1.1

Name	Bits	Description
Enable	0	Enables the encoder processing
Use_A_Rising	1	Use rising edge of input signal A for speed measurement
Use_A_Falling	2	Use rising edge of input signal A for speed measurement
Resolution_Option	7..5	The resolution option described for the processor interface
Use_TS_Speed	8	Activate the sample time based speed measurement described for the processor interface
Position_Increment	30..9	The increment of the position at each edge of an input signal. The FPGA main component section for details.

Register 1.2

Figure 35: STEP_DIR_ENCODER_RCP_out Register 1.2

Name	Bits	Description
Enable	0	Enables the encoder processing
No_of_Bits-1[13..0]	14..1	The lower 14 bits of the bit value of the edge counter - 1
Index_Position	30..15	The Position of the index signal scaled to 16 bit

Register 2

Figure 36: STEP_DIR_ENCODER_RCP_out Register 2

Name	Bits	Description
Interrupt_Position	15..0	Position for an optional position interrupt, scaled to 16 bit.
Zero_Crossing_Mode	17..16	Definition of the used zero crossing mode
No_of_Bits-1[14]	18	The 14th bit of the bit value of the edge counter - 1
Debounce_Time | Hystereis	28..19	Debounce time for input digital signals or hysteresis for analog input signals
CCW_CW_Toggle	29	Defines how the direction bit of the encoder is interpreted.
No_of_Bits-1[15]	30	The MSB of the bit value of the edge counter - 1

FPGA Main Component

Block

Figure 37: STEP_DIR_ENCODER_RCP FPGA Main block

Block Dialog

You can define the parameters of the input and output registers for automatic interface generation. The dialog also contains a button to start interface generation. For more details about automatic interface generation, refer to the XSG Utils documentation chapter Automatic Interface Generation. The FPGA clock period can be configured too.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Step	Bit	Step input signal of the encoder.	Bool
Dir	Bit	Direction input signal of the encoder.	Bool
Index	Bit	Index input signal of the encoder.	Bool
Enable	-	Enables the encoder processing	Bool
Trig_Speed_Meas	-	Triggers the sample time based speed measurement	Bool
Speed_Signals	-	Vector containing the flags [Step rising, Step falling] for the signal edges considered for speed measurement	Bool (2x)
Index_Position	-	Mechanical position of the index (rising edge during forward movement), scaled to 16 bit	UFix_16_0
Interrupt_Position	-	Position for optional generation of a position interrupt, scaled to 16 bit	UFix_16_0
Position_Increment	-	The increment which has to be added to or subtracted from the current position (position scaled to 16 bit) in case of a signal edge	UFix_22_0
Resolution_Option	-	The resolution option described for the processor interface	UFix_3_0
Use_Ts_Based_Speed	-	Activates sample time based speed measurement	Bool
No_of_Bits-1	-	The bit value needed for edge detection of the step signals – 1. (2*No_of_Steps)-1	UFix_15_0
Debounce_Time	Clocks	The debounce time of signal inputs.	UFix_10_0
CCW_CW_Toggle	Bit	Defines how the direction bit of the encoder is interpreted.	Bool
Zero_Crossing_Mode	-	Definition of the used zero crossing mode. 0: direction dependent (CCW: rising edge, CW: falling edge) 1: always on rising edge 2: always on falling edge	UFix_2_0

Output The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals which are passed to the processor interface	Bus
Position	-	The calculated angle (0…360°), scaled to 0…1.	UFix_16_16
Position_Valid	-	The valid flag is set to 1 after the index has been passed at least once and the position is referenced	Bool
Direction	-	The detected direction of the encoder movement. 0 = forward, 1 = backward	Bool
Pos_Delta_Based_Time	-	The number of clock cycles between two identical edges	UFix_28_0
Pos_Delta_Based_Overfl	-	Indicates that the time between two edges exceeds the maximum range	Bool
TS_Based_Time	Clocks	The number of clock cycles for the last speed sample period	UFix_23_0
TS_Based_Incr	Edges	The number of edges detected for the last speed sample period. A negative number of edges indicates backward movement	Fix_16_0
TS_Based_Overflow	-	Indicates that the measurement time exceeds the maximum range	Bool
Index_Crossing	-	Impulse indicating the detection of an index passing	Bool
Position_Interrupt	-	Impulse indicating that the specified interrupt position has been crossed. The output can be connected to an interrupt block of the RTI FPGA Programming Blockset if required.	Bool

Processor Input

Block

Adapts the FPGA signals for the processor side.

Figure 38: STEP_DIR_ENCODER_RCP_in block

Block Dialog

The following parameters can be configured in the processor input dialog:

Figure 39: STEP_DIR_ENCODER_RCP_in dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Sample time FPGA	s	Sample time of the FPGA	-

Input

The processor input block has the following inputs:

Register 1

Figure 40: STEP_DIR_ENCODER_PROCESSING_in Register 1

Name	Bits	Description
Position	15..0	The calculated angle, scaled to 16 bit
Position_Valid	1	The position valid flag (see main component)
Use_Ts_Speed	31	The flag for using the sample time based speed is passed back to the processor application

Register 2

Figure 41: STEP_DIR_ENCODER_RCP_in Register 2

Name	Bits	Description
Pos_Delta_Based_Time	27..0	The number of clock cycles between two identical edges
Resolution_Option	30..28	The resolution option is passed back to the processor application
Direction	31	The direction bit (0 = forward, 1 = backward)

Register 3

Figure 42: STEP_DIR_ENCODER_RCP_in Register 3

Name	Bits	Description
Ts_Based_Time	15..0	Clock cycles for the last speed sample period
Ts_Based_Incr	31..16	Edges detected for the last speed sample period

Register 4

Figure 43: STEP_DIR_ENCODER_RCP_in Register 3

Name	Bits	Description
No_of_Bits-1	15..0	The bit value needed for edge detection of the step signals – 1. (2*No_of_Steps)-1

Output

The INC_ENCODER_RCP_in block has the following outputs:

Name	Unit	Description	Range
Position	°	The angle calculated by the FPGA main component	Range: 0…360° Resolution: 0.005°
Speed	rpm	The speed calculated by the FPGA main component	Dependent on FPGA clock rate (@100MHz: Range: >±0.17rpm Resolution: 10ns)
Position_Valid	-	The valid flag, which is raised as soon as the position is referenced (index is passed at least disable)	0|1

Hall Sensor Processing

Objective

Hall sensors use the magnetic field of the rotor in order to determine its orientation. If three digital hall sensors are mounted around the rotor in steps of 120°, a segmentation of the electrical rotor position in steps of 60° (called sectors) can be made by the digital output of each sensor. Hall sensors always measure the electrical position of a motor. If the number of motor pole pairs is 1, the mechanical position is equal to the electrical position.

Figure 44: Hall sector mapping

Angle Extrapolation

To obtain more precise angle information, the angle can be extrapolated using the velocity information. As

$$\omega = \frac{d\varepsilon}{dt}$$

the angle can be calculated as

$$\varepsilon = \int \omega dt + \varepsilon_0$$

The start angle ε_0 is known when a hall edge occurs. As velocity is also determined the angle can be extrapolated. This procedure shows good results at constant velocities but fails on rapidly changing velocities.

Figure 45: Hall sensor processing with and without extrapolation

Content

The blockset contains the following elements:

- Processor Interface: HALL_SENSOR_RCP_out (Processor Interface)
- FPGA Interface: HALL_SENSOR_RCP_in (FPGA Interface)
- FPGA: HALL_SENSOR_RCP (FPGA Main Component)
- FPGA Interface: HALL_SENSOR_RCP_out (FPGA Interface)
- Processor Interface: HALL_SENSOR_RCP_in (Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

Figure 46: HALL_SENSOR_RCP_out block

Block Dialog

The processor output block provides the following dialog:

Figure 47: HALL_SENSOR_RCP_out dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Debounce time	s	Optional debounce time for analog or digital input signals	Dependent on FPGA clock rate (@100MHz: Range 0…10.23µs Resolution 10ns)
Use sector time for speed meas.	-	0 = The speed is measured by evaluating the time elapsed for a whole electrical revolution. This time measurement is the most accurate, but has a high measurement dead time. It is updated with every Hall edge 1 = The speed is measured by evaluating the time elapsed for a single Hall sector. This measurement is less accurate but infers less measurement dead time. It is updated with every Hall edge.	0|1
Minimum speed for extrapolation	rpm	For low speeds, angle extrapolation does not provide accurate results. According to this, a minimum speed for angle extrapolation can be defined. If this parameter is set to 0, angle extrapolation will be deactivated	Dependent on FPGA clock rate (@100MHz: Range: > 1.4 rpm Resolution: 40ns)
Sector start angles	°	The start angles for each Hall sector of the motor setup can be defined. The start angle is defined as the angle where the sector is entered in forward movement (left side in the diagram), All combinations of sectors are possible.	Range: 0…360° Resolution: 0.005°
Position for Interrupt trigger	°	Specifies a position for an optional position interrupt on the FPGA	Range: 0…360° Resolution: 0.005°
Sample time FPGA	s	Sample time of the FPGA	-

Input

The HALL_SENSOR_RCP_out block has the following inputs:

Name	Unit	Description	Range
Enable	-	Sensor processing enable	0|1

Output

The Processor Out block outputs eight register contents, mapped to four registers by time multiplexing. The sectioning is shown below:

Register 1.1

Figure 48: HALL_SENSOR_RCP_out Register 1.1

Name	Bits	Description
Enable	0	Enables the encoder processing
Debounce_Time	10..1	Debounce time for input signals
Speed_Option	11	0 = use revolution time 1 = use sector time
Interrupt_Position	27..12	Position for an optional position interrupt, scaled to 16 bit.
SPARE	30..28	Use rising edge of input signal B for speed measurement

Register 1.2

Figure 49: HALL_SENSOR_RCP_out Register 1.2

Name	Bits	Description
Enable	0	Enables the encoder processing
Max_Rev_Time	30..1	The maximum number of clocks per revolution to apply angle extrapolation. The value is derived from the minimum speed defined by the block mask of the processor output block

Register 2.1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Sector1_Hall			Sector1_Mult												Sector1_Start															

Figure 50: HALL_SENSOR_RCP_out Register 2.1

Name	Bits	Description
Sector1_Start	13..0	The sector start angle, scaled to 14 bit
Sector1_Mult	27..14	A multiplication factor which is calculated according to the sector length in order to derive revolution time from sector time
Sector1_Hall	30..28	The state of the Hall signals A, B, C which define the sector. LSB is Hall signal A

Register 2.2

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Sector2_Hall			Sector2_Mult														Sector2_Start													

Figure 51: HALL_SENSOR_RCP_out Register 2.2

Name	Bits	Description
Sector2_Start	13..0	The sector start angle, scaled to 14 bit
Sector2_Mult	27..14	A multiplication factor which is calculated according to the sector length in order to derive revolution time from sector time
Sector2_Hall	30..28	The state of the Hall signals A, B, C which define the sector. LSB is Hall signal A

Register 3.1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Sector3_Hall			Sector3_Mult														Sector3_Start													

Figure 52: HALL_SENSOR_RCP_out Register 3.1

Name	Bits	Description
Sector3_Start	13..0	The sector start angle, scaled to 14 bit
Sector3_Mult	27..14	A multiplication factor which is calculated according to the sector length in order to derive revolution time from sector time
Sector3_Hall	30..28	The state of the Hall signals A, B, C which define the sector. LSB is Hall signal A

Register 3.2

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Sector4_Hall							Sector4_Mult										Sector4_Start													

Figure 53: HALL_SENSOR_RCP_out Register 3.2

Name	Bits	Description
Sector4_Start	13..0	The sector start angle, scaled to 14 bit
Sector4_Mult	27..14	A multiplication factor which is calculated according to the sector length in order to derive revolution time from sector time
Sector4_Hall	30..28	The state of the Hall signals A, B, C which define the sector. LSB is Hall signal A

Register 4.1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Sector5_Hall							Sector5_Mult										Sector5_Start													

Figure 54: HALL_SENSOR_PROCESSING_out Register 4.1

Name	Bits	Description
Sector5_Start	13..0	The sector start angle, scaled to 14 bit
Sector5_Mult	27..14	A multiplication factor which is calculated according to the sector length in order to derive revolution time from sector time
Sector5_Hall	30..28	The state of the Hall signals A, B, C which define the sector. LSB is Hall signal A

Register 4.2

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Sector6_Hall							Sector6_Mult										Sector6_Start													

Figure 55: HALL_SENSOR_PROCESSING_out Register 4.2

Name	Bits	Description
Sector6_Start	13..0	The sector start angle, scaled to 14 bit
Sector6_Mult	27..14	A multiplication factor which is calculated according to the sector length in order to derive revolution time from sector time
Sector6_Hall	30..28	The state of the Hall signals A, B, C which define the sector. LSB is Hall signal A

FPGA Main Component

Block

Figure 56: HALL_SENSOR_RCP FPGA Main block

Block Dialog

You can define the parameters of the input and output registers for automatic interface generation. The dialog also contains a button to start interface generation. For more details about automatic interface generation, refer to the XSG Utils documentation chapter Automatic Interface Generation. The FPGA clock period can be configured too.

Input

The main block has the following inputs:

Name	Unit	Description	Format
A	-	Hall input signal A	Bool
B	-	Hall input signal B	Bool
C	-	Hall input signal C	Bool
Enable	-	Enables the sensor processing	Bool
Debounce_Time	Clocks	The debounce time of signal inputs.	UFix_10_0
Speed_Option	-	The time measurement for speed calculation, which will be provided for the processor application 0 = revolution time 1 = sector time	Bool
Interrupt_Position	-	Position for optional generation of a position interrupt, scaled to 16 bit	UFix_16_0
Max_Rev_Time	Clocks	The maximum number of clocks per revolution to apply angle extrapolation. The value is derived from the minimum speed defined by the block mask of the processor output block	UFix_32_0
Sector_Start	-	The start angles for all sectors, sorted in ascending order	UFix_16_0 (6x)
Sector_Hall	-	The state of the Hall signals A, B, C which define the sectors. LSB is Hall signal A	UFix_3_0 (6x)
Speed_Mult	-	A multiplication factor which is calculated according to the sector length in order to derive revolution time from sector time (if all sectors have the same length, all factors have the value 6.0)	UFix_14_10 (6x)

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals which are passed to the processor interface	Bus
Position	-	The calculated angle (0…360°), scaled to 0…1	UFix_16_16
Direction	-	The detected direction of the encoder movement. 0 = forward, 1 = backward	Bool
Revolution_Time	Clocks	The number of clock cycles for an electrical revolution. The measurement is based on a whole mechanical revolution and is updated with each Hall edge.	UFix_35_0
Sector_Time	Clocks	The number of clock cycles for an electrical revolution. The measurement is based on the last sector completed and is updated with each Hall edge.	UFix_35_0
Sector_Number	-	The number of the current Hall sector. Sector 1 is the sector with the lowest start angle, while sector 6 is the sector with the highest start angle. If the output is 0, the Hall input signals are invalid (all 0 or all 1).	UFix_3_0
Position_Interrupt	-	Impulse indicating that the specified interrupt position has been crossed. The output can be connected to an interrupt block of the RTI FPGA Programming Blockset if required.	Bool

Processor Input

Block

Adapts the FPGA signals for the processor side.

Figure 57: HALL_SENSOR_RCP_in block

Block Dialog

The following parameters can be configured in the processor input dialog:

Figure 58: HALL_SENSOR_RCP_in dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Sample time FPGA	s	Sample time of the FPGA	-

Input

The processor input block has the following inputs:

Register 1

Figure 59: HALL_SENSOR_RCP_in Register 1

Name	Bits	Description
Position	15..0	The calculated angle, scaled to 16 bit
Sector_Number	18..16	The number of the current sector
Hall_A_Deb	19	The debounced Hall input signal A
Hall_B_Deb	20	The debounced Hall input signal B
Hall_C_Deb	20	The debounced Hall input signal C

Register 2

Figure 60: HALL_SENSOR_RCPG_in Register 2

Name	Bits	Description
Revolution_Time	29..0	The number of clock cycles for 1 electrical revolution
Downscale	30	Flag indicating, that the revolution time is downscaled by a factor of 32. This is used when the 30 bit register space is too small for the number of clocks per revolution. This way, lower revolution times (< 1 rpm) can be measured
Direction	31	The direction bit (0 = forward, 1 = backward)

Output

The HALL_SENSOR_RCP_in block has the following outputs:

Name	Unit	Description	Range
Position	°	The angle calculated by the FPGA main component	Range: 0…360° Resolution: 0.005°
Speed	rpm	The speed calculated by the FPGA main component	Dependent on FPGA clock rate (@100MHz: Range: >±0.17rpm Resolution: 10ns)
Sector_Number	-	The number of the current Hall sector. Sector 1 is the sector with the lowest start angle, while sector 6 is the sector with the highest start angle. If the output is 0, the Hall input signals are invalid (all 0 or all 1).	1…6
Hall State	-	Bus containing the debounced Hall input signals A, B, C	0|1 (3x)

EnDat 2.1 Processing

Objective

The EnDat interface from HEIDENHAIN is a digital, bidirectional interface for encoders. It is capable of transmitting position values from incremental and absolute encoders as well as transmitting or updating information stored in the encoder, or saving new information. Due to the serial transmission method only four signal lines are required.

Content

The blockset contains the following elements:

- Processor Interface: ENDAT_RCP_out (Processor Interface)
- FPGA Interface: ENDAT_RCP_in (FPGA Interface)
- FPGA: ENDAT_RCP (FPGA Main Component)
- FPGA Interface: ENDAT_RCP_out (FPGA Interface)
- Processor Interface: ENDAT_RCP_in (Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

Figure 61: ENDAT_RCP_out block

Block Dialog

The processor output block provides the following dialog:

Figure 62: ENDAT_RCP_out dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Counter-clockwise Rotation	-	The default orientation is defined as follows: Clockwise rotation (front view of the motor shaft) means forward rotation. Setting this value to one inverts rotor position determination.	0|1
Singleturn Resolution	-	Specifies the angular resolution of the encoder used in bit.	Range: 1..32
Multiturn Resolution	-	Specifies the resolution for the revolution number of the encoder used in bit.	Range: 0..31
Serial Frequency	Hz	The clock frequency of the serial data transfer. The frequency should be chosen according to the sensor data sheet and cable length.	Range:100,000… 5,000,000 1/FPGA_clock
Synchronize to Real-Time Application		If this option is set the data transfer of a new position sample will be triggered with the call of this block. Otherwise the data transfer will be triggered as often as possible. A synchronous data transfer can have advantages for velocity processing.	0|1
Sample time FPGA	s	Sample time of the FPGA	-
Processor fixed step size		The processor sample time for offline simulation.	-

Input

The ENDAT_RCP_out block has the following inputs:

Name	Unit	Description	Range
Enable	-	Sensor processing enable	0|1

Output

The Processor Out block outputs two registers. The sectioning is shown below:

Register 1:

Figure 63: ENDAT_RCP_out Register 1

Name	Bits	Description
Enable	0	Enables the sensor processing
SingleTurnRes-1	5..1	Singleturn Resolution -1 for the Endat Proccessing.
MultiTurnRes	6..10	Multiturn Resolution for the EnDat Processing.
SerialFrequency	23..11	Clock frequency of the EnDat core in clocks.
Trigger_Mode	24	Enable/disable triggered mode. *0: Continuous mode* *1: Triggered by processor application*
Trigger	25	Trigger bit from the processor, if trigger mode is enabled.
CounterClockwise	26	If '1', it reverses the interpretation of the position sent by the sensor.

Register 2:

Figure 64: ENDAT_RCP_out Register 2

Name	Bits	Description
Delay_Error	0..15	The amount of clock counts that will be waited in presence of a communication error.
Delay_Standard	16..31	The amount of clock counts that will be waited in standard cases before transmitting.

FPGA Main Component

Block

Figure 65: ENDAT_RCP FPGA Main block

Block Dialog

You can define the parameters of the input and output registers for automatic interface generation. The dialog also contains a button to start interface generation. For more details about automatic interface generation, refer to the XSG Utils documentation chapter Automatic Interface Generation. The FPGA clock period can be configured too.

Input

Name	Unit	Description	Format
Input_Data	-	Serial data bit received by an EnDat sensor.	Bool
Trigger	-	Triggers an EnDat cycle.	Bool
Multiturn	-	Enables or disables multiturn mode.	Bool
Serial_Clk_Period	Clocks	Length of an EnDat clock in clocks.	UFix_13_0
Singleturn_Resolution	-	The EnDat sensor's singleturn resolution	UFix_5_0
Multiturn_Resolution	-	The EnDat sensor's multiturn resolution	UFix_5_0
Delay_Error	Clocks	The amount of clock counts that will be waited in presence of a communication error.	UFix_16_0
Delay_Standard	Clocks	The amount of clock counts that will be waited in standard cases before transmitting.	UFix_16_0
Enable	-	Enables the XSG block.	Bool

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals which are passed to the processor interface	Bus
Output_Data	-	Serial bits that should be transmitted to the EnDat sensor	Bool
Drive_Enable	-	Flag that indicates, that the Data line of the EnDat interface should be driven with Output_Data (1). Otherwise Data will be received (0).	Bool
Position	-	Current position transmitted by the EnDat sensor. The position is normalized from 0 .. 1.	UFix_32_32
Multiturn_Revolutions	-	Current revolution transmitted by the EnDat sensor	UFix_31_0
Error	-	Flag that indicates a communication error	Bool
Update	-	Indicates refreshed output values	Bool
Clock	-	The generated clock signal bit. Has to be connected to the EnDat clock line.	Bool

Processor Input

Block Adapts the FPGA signals for the processor side.

Figure 66: ENDAT_ RCP_in block

Input

The processor input block has the following inputs:

Register 1

Figure 67: ENDAT_RCP_in Register 1

Name	Bits	Description
Position	31..0	The calculated, normalized position angle, scaled to 32 bit which correspondents to 0 to 1.

Register 2

Figure 68: ENDAT_RCP_in Register 2

Name	Bits	Description
Multiturn_Revolutions	30..0	The number of revolutions sent by the EnDat sensor. The number depends on the Multiturn resolution of the sensor.
Error	31	Flag that indicates a communication error.

Output

The ENDAT_RCP_in block has the following outputs:

Name	Unit	Description	Range
Position	°	This output provides the current mechanical angle (single-turn) in degrees. The angle provided is absolute as soon as the index signal is detected.	Singleturn: 0° .. <360° Multiturn: 0° .. (MultiturnRevolution+1)*360°
Error	-	Indicates that the sensor sets the error bit in the position frame or time-out while waiting for the start bit.	0|1

Resolver & LVDT Processing

Objective

The Resolver and Processing is used to derive position and velocity by decoding the output of both resolvers and LVDT sensors.
It is able to generate an excitation signal of variable frequency and amplitude. The resolver processing is capable of working with the self-generated excitation, as well as with an externally generated excitation. In addition to the position and speed processing, a signal analysis is performed, detecting amplitudes, offsets and phase shifts between the signals. Offsets and phase shifts can automatically be compensated to increase accuracy.

Resolver Processing

Resolvers provide precise information about the absolute mechanical rotor position. A resolver mainly consists of three coils, a rotor and two stator coils. Either the rotor or the stator coils are excited with a sinusoidal signal which causes a position-dependent induction signal in the remaining coils.
The angle is determined by calculating the arc tangent of S_1 by S_2 (also called “sine” and “cosine”). The best moment to calculate the arc tangent is during the extremes of the excitation signal, while it is undefined in the zero crossings of the excitation.

Figure 69: Resolver shape over 360°

LVDT Processing

LVDTs (Linear Variable Differential Transformer) consist of a primary coil and one or two secondary coils. LVDTs are used to measure linear displacement. There are two general types: 4-wire and 5-wire LVDT. 4-wire LVDTs have only one secondary coil (A). 5-wire LVDTs have two secondary coils (A, B), which are connected in serial.
For 4-wire LVDT the position is calculated as A / EXC. For 5-wire LVDT the position is calculated as (A-B) / (A+B). The best moment to calculate the position is during the extremes of the excitation signal, while it is undefined in the zero crossings of the excitation.

Amplifier Module for Resolver & LVDT

To provide an excitation signal to a resolver or LVDT, additional signal conditioning is required. This can be achieved by connecting a DS5346 amplifier module to the DAC output of the FPGA board or I/O module used.

Figure 70: DS5346 amplifier module

The most important electrical characteristics of the DS5346 are listed in the table below. For more detailed information, please refer to the DS5346 data sheet.

Property	Value	Unit
Number of channels	2	-
Input voltage range	±10	V
Gain (optional)	1.2	-
Maximum supply current output	60	mA
Input resistance	20	kΩ
Over-voltage protection	±50	V
Supply voltage	±15	V

Content

The blockset contains the following elements:

- Processor Interface: RESOLVER_LVDT_RCP_out (Processor Interface)
- FPGA Interface: RESOLVER_LVDT_RCP_in (FPGA Interface)
- FPGA: RESOLVER_LVDT_RCP (FPGA Main Component)
- FPGA Interface: RESOLVER_LVDT_RCP_out (FPGA Interface)
- Processor Interface: RESOLVER_LVDT_RCP_in (Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA

Figure 71: RESOLVER_LVDT_RCP_out block

Block Dialog

The processor output block provides the following dialog:

Figure 72: RESOLVER_LVDT_RCP_out dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Minimum excitation magnitude for position sampling	Bit (LSBs)	Set the threshold sampling the position. The position will be sampled when the current magnitude of the excitation exceeds the specified value. If this parameter is set to 0, the position will only be sampled at the maxima and minima of the excitation.	Range: 0…32767 Resolution: 1
Minimum signal amplitude for phase shift measurement	Bit (LSBs)	The phase shift measurement is not possible in position ranges where the amplitude of the secondary coils is nearly zero. The higher the position-dependent amplitude of the sine/cosine or A/B signals, the more accurate the phase shift measurement. A minimum signal amplitude can be defined, for which a phase shift measurement is valid. If the amplitude of a signal does not exceed this threshold, no measurement will be performed and the output stays unchanged.	Range: 0…32767 Resolution: 1
Extrapolation	-	Setting this parameter to 1 enables extrapolation of the position at the times when no valid arctan measurement can be performed (e.g. near the zero crossing of the excitation)	0|1
Phase shift compensation	-	Setting this parameter to 1 enables automatic phase shift compensation between sine/A and cosine/B as well as excitation. Phase shift compensation is only valid in a phase shift range of 1000 FPGA clocks.	0|1
Sensor type	-	0 = Resolver 1 = LVDT	0|1

LVDT type	-	In case of LVDT processing, the type of LVDT sensor can be specified: 0 = 4-wire 1 = 5-wire	0|1
Input signal filter time constant	-	All analog signals are filtered with a discrete PT1 to reduce noise. The filter time constant can be selected by the user.	Dependent on FPGA clock rate (@100MHz: Range:160ns..10µs Resolution: 10ns)
Speed averaging option		The speed measurement is performed twice per excitation period. To receive higher accuracy multiple speed samples can be averaged on the FPGA. The number of samples to average calculates to 2^x, where x is the speed averaging option in the block mask.	Range: 0…7 Resolution: 1
Frequency of generated excitation	Hz	The frequency of the excitation signal to be generated	Dependent on FPGA clock rate (@100MHz: Range:1kHz…30kHz Resolution: 10ns)
Amplitude of generated excitation	Bit (LSBs)	The Amplitude of the generated excitation signal, scaled in LSBs of the DAC raw output.	Range: 0…32767 Resolution: 1
DC offset of generated excitation	Bit (LSBs)	Optional DC offset for the generated excitation, e.g. for external offset compensation	Range:-8192..8191 Resolution: 1
Sample time FPGA	s	Sample time of the FPGA	-

Input

The RESOLVER_LVDT_RCP_out block has the following inputs:

Name	Unit	Description	Range
Enable	-	Sensor processing enable	0|1

Output

The Processor Out block outputs four register contents, mapped to two registers by time multiplexing. The sectioning is shown below:

Register 1.1

Figure 73: RESOLVER_LVDT_RCP_out Register 1.1

Name	Bits	Description
Enable	0	Enables the encoder processing
Gen_Exc_Frequency	17..1	The excitation frequency, scaled to the FPGA raw value
SPARE	30..18	-

Register 1.2

Figure 74: RESOLVER_LVDT_RCP_out Register 1.2

Name	Bits	Description
Enable	0	Enables the encoder processing
Gen_Exc_Offset	14..1	The offset of the excitation output signal, scaled with a bias of 8192
Averaging_Option	18..15	The option for averaging for speed measurement samples.
SPARE	30..19	-

Register 2.1

Figure 75: RESOLVER_LVDT_RCP_out Register 2.1

Name	Bits	Description
Exc_Out_Amplitude	14..0	The amplitude of the generated excitation signal
Delay_Compens	15	Phase shift compensation enable bit
Sensor_Type	16	0 = Resolver, 1 = LVDT
LVDT_Type	17	0 = 4-wire, 1 = 5-wire
F_Filter	30..18	The PT1 filter frequency, scaled to 13 bit

Register 2.2

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Extrapolation	Position_Sample_Threshold														Delay_Measure_Threshold															

Figure 76: RESOLVER_LVDT_RCP_out Register 2.2

Name	Bits	Description
Delay_Measure_ Threshold	14..0	Signal amplitude threshold for measuring phase shifts
Position_Sample_ Threshold	29..15	Excitation magnitude threshold for sampling the resolver position
Extrapolation	30	Extrapolation enable bit

FPGA Main Component

Block

Figure 77: RESOLVER_LVDT_RCP FPGA Main block

Block Dialog

You can define the parameters of the input and output registers for automatic interface generation. The dialog also contains a button to start interface generation. For more details about automatic interface generation, refer to the XSG Utils documentation chapter Automatic Interface Generation. The FPGA clock period can be configured too.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Sine|A	Bit	Sine input (resolver) or input signal A (LVDT)	Fix_16_0
Cosine|B	Bit	Cosine input (resolver) or input signal B (LVDT)	Fix_16_0
Excitation_In	Bit	Input for excitation. If self-generated excitation is used, connect this input to the excitation output (either directly or across analog I/O)	Fix_16_0
Enable	-	Enables the sensor processing	Bool
Sensor_Type	-	0 = Resolver, 1 = LVDT	Bool
LVDT_Type	-	0 = 4-wire, 1 = 5-wire In case of resolver processing this input will be ignored	Bool
Delay_Measure_ Threshold	-	The threshold for phase shift measurement as described in the processor interface section	UFix_15_0
Position_Sample_ Threshold	-	The position sampling threshold as described in the processor interface section	UFix_15_0
Extrapolation	-	Extrapolation enable	Bool
Delay_Compensation	-	Phase shift compensation enable	Bool
f_Filter	-	The input signal PT1 filter frequency in the unit 1/clocks	UFix_18_18
Gen_Exc_Frequency	-	The frequency for the generated excitation, scaled to the FPGA raw input value	UFix_17_14
Gen_Exc_Amplitude	Bit	The amplitude for the generated excitation	UFix_15_0
Gen_Exc_Offset	Bit	The DC offset for the generated excitation	Fix_14_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals which are passed to the processor interface	Bus
Output_Valid	-	Flag indicating that the output are valid, which is the case after 2 valid excitation periods	Bool
Position	-	The calculated position. In case of resolver the position is scaled in the full range of 16 bit (0…65535), in case of LVDT the position is in the range -1…1	UFix_16_0
Delta_Pos	-	The position change between 2 equal edges of the excitation signal	Fix_16_0
Delta_T	Clocks	The number of clock cycles between 2 equal edges of the excitation signal	UFix_17_0
Sin|A_Delay	Clocks	The delay between the excitation and sine/A. If the excitation is later than sine/A, the delay is negative.	Fix_17_0
Cos|B_Delay	Clocks	The delay between the excitation and cosine/B. If the excitation is later than cosine/B, the delay is negative.	Fix_17_0
Sin|A_Amplitude	Bit	The position-dependent amplitude of sine/A	UFix_15_0
Cos|B_Amplitude	Bit	The position-dependent amplitude of cosine/B	UFix_15_0
Exc_in_Amplitude	Bit	The measured amplitude of the excitation input	UFix_15_0
Sin|A_Offset	Bit	The DC offset measured for sine/A input	Fix_14_2
Cos|B_Offset	Bit	The DC offset measured for cosine/B input	Fix_14_2
Exc_In_Offset	Bit	The DC offset measured for excitation input	Fix_14_2
Exc_In_Period	Clocks	The measured number of clock cycles of the excitation input period	UFix_17_0

Phase_Shift_Flags	-	A Bus containing the following status signals: *(1) Sin|A_Meas_Active* This flag indicates that the phase shift measurement for sine/A is currently performed *(2) Cos|B_Meas_Active* This flag indicates that the phase shift measurement for cosine/B is currently performed *(3) Delay_Meas_Valid* This flag indicates that phase shift measurements for all relevant signals have been performed at least once *(4) Phase_Shift_Compens.* This flag indicates that the phase shifts are within a compensable range (< 1000 clock cycles)	Bool (4x)
Exc_Extremum	-	Impulse occurring at maxima and minima of excitation input	Bool
Exc_Zero_Cross	-	Impulse occurring at zero crossings of excitation input	Bool
Generated_Excitation	Bit	The generated excitation signal.	Fix_14_0

Processor Input

Block

Adapts the FPGA signals for the processor side.

Figure 78: RESOLVER_LVDT_RCP_in block

Block Dialog

The following parameters can be configured in the processor input dialog:

Figure 79. RESOLVER_LVDT_RCP_in dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Sample time FPGA	s	Sample time of the FPGA	-

Input

The processor input block has the following inputs:

Register 1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Output_Valid	Sensor_Type	Exc_In_Offset														Position															

Figure 80: RESOLVER_LVDT_RCP_in Register 1

Name	Bits	Description
Position	15..0	The calculated angle, scaled to 0…65535
Exc_In_Offset	29..16	The DC offset measured for the excitation input
Sensor_Type	30	The sensor type is forwarded to the processor input for scaling
Output_Valid	31	The output valid flag

Register 2

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Delta_Pos																Delta_T															

Figure 81: RESOLVER_LVDT_RCP_in Register 2

Name	Bits	Description
Delta_T	16..0	The number of clock cycles between 2 equal edges of the excitation signal
Delta_Pos	31..17	The position change between 2 equal edges of the excitation signal

Register 3

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sin|A_Delay																	Sin|A_Meas_Ac	Sin|A_Offset													

Figure 82: RESOLVER_LVDT_RCP_in Register 3

Name	Bits	Description
Sin|A_Offset	13..0	The DC offset measured for sine/A input
Sin|A_Meas_Act	14	Status bit indicating that a phase shift measurement is currently performed for the sine/A input
Sin|A_Delay	31..15	The delay between the excitation and sine/A

Register 4

Figure 83: RESOLVER_LVDT_RCP_in Register 4

Name	Bits	Description
Cos|B_Offset	13..0	The DC offset measured for cosine/B input
Cos|B_Meas_Act	14	Status bit indicating that a phase shift measurement is currently performed for the cosine/B input
Cos|B_Delay	31..15	The delay between the excitation and cosine/B

Register 5

Figure 84. RESOLVER_LVDT_RCP_in Register 5

Name	Bits	Description
Exc_In_Period	16..0	The measured period of the excitation input
Exc_In_Amplitude	31..17	The measured amplitude of the excitation input

Register 6

Figure 85: RESOLVER_LVDT_RCP_in Register 6

Name	Bits	Description
Sin|A_Amplitude	13..0	The position-dependent amplitude of sine/A
Cos|A_Amplitude	29..14	The position-dependent amplitude of cosine/B
Delay_Meas_Val	30	Status bit indicating that all phase shift measurements have been performed at least once
Phase_Shift_Val	31	Status bit indicating that phase shifts are within a compensable range (< 1000 clock cycles)

Output

The RESOLVER_LVDT_RCP_in block has the following outputs:

Name	Unit	Description	Range
Position	°|-	The angle calculated by the FPGA main component	For Resolver: Range: 0…360° Resolution: 0.005° For LVDT: Range: -1…1 Resolution: 3e-5
Speed	1/min	The speed calculated by the FPGA main component	Range:0...(1/2T_{EXC}) Resolution: $T_{S\|FPGA}$
Phase_Shifts	°	The phase shift between the excitation and sine/A (1st element) and the phase shift between the excitation and cosine/B (2nd element)..	Range: -90°…90° Resolution: $T_{S\|FPGA}$
Amplitudes		The current amplitude of sine/A (1st element), cosine/B (2nd element) and excitation input (3rd element).	Range: 0…32767 Resolution: 1
Offsets	Bit (LSBs)	The DC offset measured for sine/A (1st element), cosine/B (2nd element) and excitation input (3rd element).	Range: -2048...2047 Resolution: 0.25
Exc_In_ Frequency	Hz	The measured frequency of the excitation input	Dependent on FPGA clock rate (@100MHz: Range: 763Hz…32kHz Resolution: 10ns)
Output_Valid	-	Flag indicating that the output are valid, which is the case after 2 valid excitation periods	0|1

Phase_Shift_ Flags	-	A Bus containing the following status signals: *(1) Sin|A_Meas_Active* This flag indicates that the phase shift measurement for sine/A is currently performed *(2) Cos|B_Meas_Active* This flag indicates that the phase shift measurement for cosine/B is currently performed *(3) Delay_Meas_Valid* This flag indicates that phase shift measurements for all relevant signals have been performed at least once *(4) Phase_Shift_Compens.* This flag indicates that the phase shifts are within a compensable range (< 1000 clock cycles)	0|1 (4x)

SSI & SPI Sensor Processing

Objective

SSI (Synchronous Serial Interface) encoders are optical absolute encoders which contain integrated signal processing and send position information via a serial protocol based on the physical differential RS422 standard. Beside singleturn encoders, which provide position information, there are multiturn encoders, which provide the revolution number as well. Additional check or status bits (e.g. a parity bit) are sometimes transmitted as well. The output data is usually Gray coded.
The SSI interface handles the SSI protocol with variable parameters and provides received data to the output block.

SPI (Serial Peripheral Interface) is used for various types of sensors. In contrast to SSI, communication in both directions is possible. SPI usually uses a single-ended interface.

Content

The blockset contains the following elements:

- Processor Interface: SSI_SPI_RCP_out (Processor Interface)
- FPGA Interface: SSI_SPI_RCP_in (FPGA Interface)
- FPGA: SSI_SPI_RCP (FPGA Main Component)
- FPGA Interface: SSI_SPI_RCP_out (FPGA Interface)
- Processor Interface: SSI_SPI_RCP_in (Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA

Figure 86: SSI_SPI_RCP_out block

Block Dialog

The processor output block provides the following dialog:

Figure 87. SSI_SPI_RCP_out dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Protocol Selection	-	Select between SSI (0) or SPI (1) protocol	0|1
Trigger Mode	-	Set the mode for receiving new data words: 0 = As soon as the sensor is ready to transmit (continuous transmission / SSI only) 1 = triggered by the processor model (call of this block) 2 = triggered by an external input of the FPGA main component	Range: 0…2 Resolution: 1
Number of bits transmitted	-	The total number of bits to be transmitted each cycle	Range: 0…48 Resolution: 1
Serial clock period	s	The period of the SSI or SPI serial clock (SCLK)	Dependent on FPGA clock rate (@100MHz: Range:0.02…163µs Resolution: 20ns)
Maximum SSI monoflop time	s	The monoflop time is the minimum time between consecutive SSI data transmissions. After the monoflop time, the sensor sets the data line high. A maximum monoflop time must be specified to recognize communication timeouts	Dependent on FPGA clock rate (@100MHz: Range:0.02…163µs Resolution: 20ns)
Gray encoded data transmission	-	If this parameter is set to 1, the received serial data will be Gray decoded (usual for SSI, but not for SPI)	0|1
Bit order	-	Select between LSB (0) or MSB (1) first. SSI usually uses MSB first	0|1
Clock polarity	-	The idle state of the serial clock line. For SSI usually 1 (high)	0|1
Clock phase	-	Describes if the first data bit is latched with the first (0) or second (1) serial clock edge. For SSI, clock phase is usually 0 (after one empty bit)	0|1
Sample time FPGA	s	Sample time of the FPGA	-

SPI communication contains no *ready* feedback from the senor. The trigger option has to be set to *processor model* or *external* for SPI.

Input

The SSI_SPI_RCP_out block has the following inputs:

Name	Unit	Description	Range
Enable	-	SSI or SPI data transmission enable	0|1
SPI_TX_Data	Bit	Data to send in case of SPI	Range: 0…(2^{48}-1) Resolution: 1

Output

The Processor Out block provides four register outputs, mapped to two registers by time multiplexing. The sectioning is shown below:

Register 1

Figure 88: SSI_SPI_RCP_out Register 1

Name	Bits	Description
SPI_TX_Data (31:0)	31..0	The lower bits of data to send via SPI

Register 2.1

Figure 89: SSI_SPI_RCP_out Register 2.1

Name	Bits	Description
SPI_TX_Data(47:32)	15..0	The higher bits of data to send via SPI
CN_Trigger	16	Trigger from processor model
SCLK_Period	29..17	Period of the SSI or SPI serial clock. 1 LSB in the register corresponds to 2 FPGA clock cycles.

Register 2.2

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	SSI_Max_Monoflop_Time														CN_Trigger	SPI_TX_Data (47:32)															

Figure 90: SSI_SPI_RCP_out Register 2.2

Name	Bits	Description
SPI_TX_Data(47:32)	15..0	The higher bits of data to send via SPI
CN_Trigger	16	Trigger from processor model
Max_Monoflop_Time	29..17	The maximum monoflop time for SSI sensors as described in the processor output parameter table. 1 LSB in the register corresponds to 2 FPGA clock cycles

Register 2.3

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2	Enable	Protocol_Type	CPHA	CPOL	Bit_Order	Gray_Decode	No_of_Bits							En_Ext_Trig	CN_Trigger	SPI_TX_Data (47:32)															

Figure 91: SSI_SPI_RCP_out Register 2.3

Name	Bits	Description
SPI_TX_Data(47:32)	15..0	The higher bits of data to send via SPI
CN_Trigger	16	Trigger from processor model
En_Ext_Trig	17	Enables the external trigger input
No_of_Bits	23..18	Number of data bits to be transmitted
Gray_Decode	24	Activates Gray decoding of serial input data
Bit_Order	25	Selects between LSB (0) or MSB (1) first
CPOL	26	Clock polarity (idle state of the SCLK line)
CPHA	27	Clock phase, determines if the first data bit is latched in with the first (0) or second (1) serial clock edge
Protocol_Type	28	Select if SSI (0) or SPI (1) protocol is used
Enable	29	Enables the SSI or SPI master

FPGA Main Component

Block

Figure 92: SSI_SPI_RCP FPGA Main block

Block Dialog

You can define the parameters of the input and output registers for automatic interface generation. The dialog also contains a button to start interface generation. For more details about automatic interface generation, refer to the XSG Utils documentation chapter Automatic Interface Generation. The FPGA clock period can be configured too.

Input

The main block has the following inputs:

Name	Unit	Description	Format
SDI	-	The serial data received from the sensor (in case of SPI often called MISO)	Bool
Ext_Trigger	-	External trigger to start data transmission (continuous transmission if trigger remains high)	Bool
Enable	-	Enables the sensor processing	Bool
En_Ext_Trigger	-	Enables the external trigger input	Bool
CN_Trigger	-	Transmission trigger from processor model	Bool
No_of_Bits	-	Number of data bits to be transmitted	UFix_6_0
SCLK_Period	Clocks	The period of the SSI or SPI serial clock	UFix_14_0
Bit_Order	-	Select between LSB (0) or MSB (1) first	Bool
Gray_Decode	-	Activates Gray decoding of serial input data	Bool
SSI_Max_Monoflop_Time	Clocks	The maximum monoflop time of the SSI sensor as described in the processor output parameter table	UFix_14_0
CPOL	-	Clock polarity (idle state of the SCLK line)	Bool
CPHA		Clock phase, selects if the first data bit is latched in with the first (0) or second (1) serial clock edge	Bool
Protocol_Type	-	Selects between SSI (0) or SPI (1)	Bool
SPI_TX_Data	-	The data to send in case of SPI	UFix_48_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals which are passed to the processor interface	Bus
RX_Data	-	The data received during the last data transmission	UFix_48_0
Data_Ready	-	Impulse indicating that a data transmission has been completed. Can e.g. be used as an interrupt trigger	Bool
Parity_Odd	-	The parity of the bits received is calculated. This output represents the parity value	Bool
SCLK	-	The serial clock to be provided for the sensor connected	Bool
SPI_CS	-	The chip select signal. Only required for SPI	Bool
SPI_SDO	-	The serial data output (MOSI). Only required for SPI	Bool

Processor Input

Block

Adapts the FPGA signals for the processor side.

Figure 93: SSI_SPI_RCP_in block

Block Dialog

The processor input does not provide any block dialog or mask parameters.

Input

The processor input block has the following inputs:

Register 1

Figure 94: SSI_SPI_RCP_in Register 1

Name	Bits	Description
RX_Data (31:0)	31..0	Bits 31..0 of the received SSI or SPI data

Register 2

Figure 95:SSI_SPI_RCP_in Register 2

Name	Bits	Description
RX_Data (47:32)	15..0	Bits 47..32 of the received SSI or SPI data
SSI_Com_Error	16	Signals a communication timeout at the SSI interface
Parity_Odd	17	Parity of received data is odd
SPARE	31..18	-

Output

The processor input block has the following outputs:

Name	Unit	Description	Range
RX_Data	Bit	The received SSI or SPI raw data in LSBs.	Range: 0…(2^{48}-1) Resolution: 1
SSI_Com_Error	-	A communication error flag, which is raised when the SSI sensor does not set the data line high within the specified maximum monoflop time	0|1
Parity_Odd	-	The parity of the bits received is calculated. This output represents the parity value.	0|1

SSI Sensor Processing

Objective

SSI (Synchronous Serial Interface) encoders are optical absolute encoders which contain integrated signal processing and send position information via a serial protocol based on the physical differential RS422 standard. Beside single turn encoders, which provide position information, there are multiturn encoders, which also provide the revolution number (absolute position). Additional check or status bits (e.g. parity bit) can be also received. The output data is usually Gray coded. The SSI interface handles the SSI protocol and based on the specified frame settings, the received data is decoded and separated into several groups which are provided at the block output. Two communication error states can be detected: Violation of specified maximum timeout and parity error.

Content

The block set contains the following elements:

- Processor Interface: SSI_RCP_out (Processor Interface)
- FPGA Interface: SSI_RCP_in (FPGA Interface)
- FPGA: SSI_RCP (FPGA Main Component)
- FPGA Interface: SSI_RCP_out (FPGA Interface)
- Processor Interface: SSI_RCP_in (Processor Interface)

Processor Output

Block

Merges the processor signals from General settings (Trigger mode, Data coding type, Transmission reference, Maximum SSI monoflop time, Serial clock period, etc.) and Frame settings and writes them to the FPGA. Based on the specified frame settings StatusBus output is updated and shared with SSI_RCP_in interface block for decoding of the received data.

Figure 96: SSI_RCP_out block

Block Dialog

The processor output block contains two dialogs. In the first one, the general parameters can be configured.

Figure 97: SSI_RCP_out dialog; page “General”

The first dialog has the following parameters:

Name	Unit	Description	Range
Trigger Mode	[-]	Set the mode for triggering new communication cycle: 0: Continuous transmission (as soon as the sensor is ready to transmit) 1: Processor trigger (triggered by the processor model - call of this block) 2: External trigger (triggered by an external input of the FPGA main component)	0…2
Data coding type	[-]	Type of data coding in the transmission frame: 0: Binary 1: Gray – Position data (only position bits are Gray coded) 2: Gray – Data frame (whole data frame is Gray coded)	0…2
Transmission reference	[-]	Reference for the sampling of serial data line: 0: First Edge (slave updates output on the clock rising edge) 1: Following Edge (slave updates output on the clock falling edge)	0|1
Clock polarity	[-]	The idle state of the serial clock line (low / high)	0|1

Parity check	[-]	Specify type of parity check: 0: Disabled 1: EVEN parity 2: ODD parity	0|1|2
Serial clock period	[s]	The period of the SSI serial clock (SCLK)	Dependent on FPGA clock rate (@100MHz: Range:0.02…163µs Resolution: 20ns)
Maximum SSI monoflop time	[s]	The monoflop time is the minimum time between consecutive SSI data transmissions. After the monoflop time, the sensor sets the data line high. A maximum monoflop time must be specified to recognize communication timeouts.	Dependent on FPGA clock rate (@100MHz: Range:0.02…163µs Resolution: 20ns)
SSI sample delay	[s]	This parameter should be specified in cases when total delay of the SSI interface signal chain (RCP I/O stage -> clock line -> encoder -> data line -> RCP I/O stage) is higher then ½ of the clock period. This cases are common when high clock frequency is applied or long cables are used. In this cases total delay of the signal chain must be calculated and entered as sample delay parameter.	Dependent on FPGA clock rate (@100MHz: Range:0.…1.27e-6s Resolution: 10ns)
Sample time FPGA main block	[s]	Sample time of the FPGA	-
Processor fixed step time	[s]	Step time of the processor task; a different sample time can be set in offline simulation to simulate the processor interface communication behaviour.	-

Total delay of the SSI interface signal chain is measured from the moment of the updating edge of clock up to arrival of new data. Figure below shows example where data is updated on the rising edge of clock and sampled on the falling edge. In this case if data arrives before the falling edge of clock data delay compensation is not needed (Delay 1 case), otherwise (Delay 2 case) sampling delay must be specified in order to compensate data delay.

Figure 98: SSI Interface data delay diagram

On the second page configuration of SSI data frame can be specified.

Figure 99: SSI_RCP_out dialog; page “Frame Settings”

In the GUI the actual used bit settings of the data frame is shown (marked in blue in the figure above). To configure other data frames the actual bit settings can be

configured with the edit ports which are marked with red. After configuration of the data frame the settings can be applied with the button “APPLY USED BIT SETTINGS”. The block GUI will be modified and the actual possible popups for the bit configuration will be activated. With the button “REFRESH/APPLY BIT ORDER” the actual bit settings will be updated, and the default bit configuration will be set. After applying the changes of the GUI with the button “Apply“the internal Simulink block configuration will be modified.

Input

The SSI_RCP_out block has the following inputs:

Name	Unit	Description	Range
Enable	[-]	Transmission enable	0|1

Output

The Processor out block provides StatusBus output which is composed of vector signals used for decoding of received data frame and two output registers which sectioning is shown below:

Register 1

Figure 100: SSI_SPI_RCP_out Register 1

Name	Bits	Description
SCLK_Period	12..0	Serial clock period
Max_Monoflop_Time	25..13	The maximum monoflop time for SSI sensors as described in the processor output parameter table
Proc_Trigger	26	Trigger from the processor model
En_Ext_Trig	27	Enables the external trigger input
Gray_Decode_Type	29…28	Type of Gray coding in the data frame
CPOL	30	Clock polarity (idle state of the SCLK line)
Enable	31	SSI data transmission enable

Register 2

Figure 101: SSI_RCP_out Register 2

Name	Bits	Description
No_of_Bits	5..0	Number of data bits to be transmitted
Table_Data	6	Bit configuration of data frame for Gray decoding with the respect to actual address in frame
Address	12..7	Actual address of committed bit
New_LUT_Data	13	Flag which starts the update of the internal bit configuration of the data frame for Gray decoding
Transmission_Reference	14	Reference for the sampling of the serial data
Sample_Delay	21…15	Configurable delay value used for compensation of the signal chain delay as described in the processor output parameter table

FPGA Main Component

Block

Figure 102: SSI_RCP FPGA Main block

Block Dialog

The FPGA main block contains the following dialog:

Figure 103: SSI_RCP FPGA Main block

On the page “Interface” you can define the parameters of the input and output registers for automatic interface generation. The dialog also contains a button to start interface generation. For more details about automatic interface generation, refer to the XSG Utils documentation chapter Automatic Interface Generation. The FPGA clock period can be configured too.

Input

The main block has the following inputs:

Name	Unit	Description	Format
SDI	[-]	The serial data received from the sensor	Bool or UFix_1_0
Ext_Trigger	[-]	External trigger to start data transmission (continuous transmission if trigger remains high)	Bool or UFix_1_0
Enable	[-]	Enables the sensor processing	Bool
En_Ext_Trigger	[-]	Enables the external trigger input	Bool
Proc_Trigger	[-]	Transmission trigger from the processor model	Bool
No_of_Bits	[-]	Number of data bits to be transmitted	UFix_6_0
SCLK_Period	[Clocks]	Period of the serial clock	UFix_14_0
Gray_Decode_Type	[-]	Enables/disables Gray decoding based on specified coding type	UFix_2_0
SSI_Max_Monoflop_Time	[Clocks]	The maximum monoflop time of the SSI sensor as described in the processor output parameter table	UFix_14_0
CPOL	[-]	Clock polarity (idle state of the SCLK line)	Bool
Transmission_Ref	[-]	Reference for the sampling of the serial data	Bool
Table_Data	[-]	Bit configuration of data frame for Gray decoding with the respect to actual address in frame	Ufix_1_0
Adress	[-]	Actual address of committed bit	Ufix_6_0
New_LUT_Data	[-]	Flag which starts the update of the internal bit configuration of the data frame for Gray decoding	Ufix_1_0
Sample_Delay	[Clocks]	Sample delay for compensation of the data delay as described in the processor output parameter table	Ufix_7_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	[-]	Contains all output signals which are passed to the processor interface	Bus
RX_Data	[-]	The data received during the last data transmission	UFix_51_0
Parity_Odd	[-]	The parity of the bits received is calculated. This output represents the parity value.	Bool
SSI_Comm_Error	[-]	Signals violation of a maximum specified timeout	Bool
Data_Ready	[-]	Impulse indicating that a data transmission has been completed. Can e.g. be used as an interrupt trigger.	Bool
SCLK	[-]	The serial clock to be provided for the sensor connected	UFix_1_0

Processor Input

Block

Merges the FPGA signals for the postprocessing on the processor side. Received raw data is decoded according to actual data frame configuration. Postprocessed data is separated and provided at the block output ports.

Figure 104: SSI_RCP_in block

Block Dialog	The processor input does not provide any block dialog or mask parameters.
Input	The processor in block has two register inputs and one signal bus input: StatusBus provides vector signals based on the actual data frame settings which are used for decoding of the received raw data.

	Input registers must be grouped under the same Register Group ID number in order to ensure synchronous update of the data which is transferred over the several registers!

The processor input block has the following register inputs:

Register 1

Figure 105: SSI_RCP_in Register 1

Name	Bits	Description
RX_Data (31:0)	31..0	Bits 31..0 of the received SSI data frame

Register 2

Figure 106: SSI_RCP_in Register 2

Name	Bits	Description
RX_Data (50:32)	18..0	Bits 50..32 of the received SSI data frame
SSI_Com_Error	19	Signals violation of a maximum specified timeout
Parity_Error	20	Signals parity mismatch
SPARE	31..21	-

Input

The processor input block has the following outputs:

Name	Unit	Description	Range
RX_Data	[Raw]	Received SSI raw data	Range: 0…(2^{51}-1) Resolution: 1
Angle_Phi	[deg]	Received position in degrees (360 base)	Range: 0°…360°
Angle_Rev	[-]	Received number of full revolutions	Range: +(2^{17}-1)…- 2^{17} Resolution: 1
Angle_Absolute	[deg]	Received absolute position in degrees	Range: 0…2^{18}*360
Error_Bits_Vector	[-]	Received error bits vector	0|1
Special_Bits_Vector	[-]	Received special bits vector	0|1
Alarm_Bit	[-]	Received alarm bit	0|1
Parity_Error	[-]	Signals parity error if parity in received data frame mismatches specified parity options	0|1
SSI_Com_Error	[-]	Signals violation of a maximum specified timeout	0|1
MT_Data	[Raw]	Received multiturn data	Range: 0…(2^{18}-1) Resolution: 1
ST_Data	[Raw]	Received singleturn data	Range: 0…(2^{18}-1) Resolution: 1

BiSS Sensor Processing

Objective

BiSS (Bidirectional Serial Synchronous) is an open source interface which is commonly used by position sensors for providing position data which is sent serially over the differential line according to the physical differential standards (e.g. RS422, RS458, etc.). Beside position data, up to 8 additional diagnosis bits (e.g. error, warning, etc.) can be transmitted as well. The data frame is protected with Cyclical Redundancy Check (CRC) bits which are transmitted inversely at the end of the frame.

The BiSS interface block handles the BiSS-C (unidirectional) protocol in point-to-point configuration (position data read-out from single slave with single data channel). Based on the specified frame settings the received data is decoded and separated into several groups which are provided at the block output. BiSS communication error states are detected and signalized at the block output, which can be used for the purpose of debugging.

Content

The blockset contains the following elements:

- Processor Interface: BISS_RCP_out (Processor Interface)
- FPGA Interface: BISS_RCP_in (FPGA Interface)
- FPGA: BISS_RCP (FPGA Main Component)
- FPGA Interface: BISS_RCP_out (FPGA Interface)
- Processor Interface: BISS_RCP_in (Processor Interface)

Processor Output

Block

Merges the processor signals from General settings (Trigger mode, Clock polarity, Serial clock period, Maximum BiSS timeout period, Maximum SCD processing time, etc.) and Frame settings and writes them to the FPGA. Based on the specified frame settings StatusBus output is updated and shared with BISS_RCP_in interface block for decoding of the received data.

Figure 107: BISS_RCP_out block

Block Dialog

The processor output block contains two dialogs. In the first one, the general parameters can be configured.

Figure 108: BISS_RCP_out dialog; page “General”

The first dialog has the following parameters:

Name	Unit	Description	Range
Trigger Mode	[-]	Set the mode for triggering new communication cycle: 0: Continuous transmission (triggered as soon as the sensor is ready to transmit) 1: Processor trigger (triggered by the processor model - call of this block) 2: External trigger (triggered by an external input of the FPGA main component)	Range: 0…2 Resolution: 1
Clock polarity	[-]	Idle state of the serial clock line: 0: Low 1: High	0|1
Serial clock period	[s]	Period of the BiSS serial clock	Dependent on FPGA clock rate (@100MHz: Range:0.01…12.5µs Resolution: 10ns)
Maximum BiSS timeout period	[s]	Maximum expected timeout duration. If during communication, at the end of communication cycle slave signalise timeout (holds data line low) for a time period which is longer then the specified one this will be detected as communication error (ID=1).	Dependent on FPGA clock rate (@100MHz: Range:12.5…40µs Resolution: 10ns)

Maximum SCD processing time	[s]	Maximum expected Single Cycle Data (SCD) processing time. If during communication, at the beginning of the frame slave delays START bit for a time period longer then specified one this will be detected as processing time error (ID=3).	Dependent on FPGA clock rate (@100MHz: Range:0….40e-6s Resolution: 10ns)
Sample time FPGA main block	[s]	Sample time of the FPGA	-
Processor fixed step time	[s]	Step time of the processor task; a different sample time can be set in offline simulation to simulate the processor interface communication behaviour.	-

On the second page data frame settings can be specified.

Figure 109: BISS_RCP_out dialog; page “Frame Settings”

The second dialog has the following parameters:

Name	Unit	Description	Range
No. multi turn bits	[-]	Number of multi turn bits	Range: 0…55 Resolution: 1
No. single turn bits	[-]	Number of single turn bits	Range: 0...55 Resolution: 1
No. diagnosis bits	[-]	Number of diagnosis bits	Range: 0…8 Resolution: 1
Multi turn resolution	[-]	Number of multiturn bits, starting from the LSB bit, which will be taken into account for the postprocessing of the RX data inside BISS_RCP_in block (e.g. it can be used to limit the range of the absolute angle).	Range: 0…55 Resolution: 1
Single turn resolution	[-]	Number of singleturn bits, starting from the MSB bit, which will be taken into account for the postprocessing of the RX data inside BISS_RCP_in block (e.g. it can be used to limit the resolution of the calculated angle).	Range: 0…55 Resolution: 1
Data alignment	[-]	Alignment of the position bits regards to position in the data frame: 0: Left alignment - Position bits are the most significant bits in the frame; diagnosis bits follow. 1: Right alignment – Diagnosis bits are the most significant bits in the frame. Position bits follow.	0|1
Position bits coding type	[-]	Coding of the position bits: 0: Binary 1: Gray	0|1
Diagnosis bits coding type	[-]	Coding of the diagnosis bits: 0: Binary 1: Gray	0|1
CRC polynomial	[-]	CRC polynomial in hexadecimal format	0x00000...0x1FFFF

CRC start value	[-]	CRC start value in hexadecimal format	0x0000...0xFFFF

Input

The BISS_RCP_out block has the following inputs:

Name	Unit	Description	Range
Enable	[-]	Transmission enable	0|1
Error_Reset	[-]	Reset the error state	0|1

Output

The Processor out block provides StatusBus output which is composed of vector signals used for decoding of the received data frame and four output registers which sectioning is shown below:

Register 1

Figure 110: BISS_RCP_out Register 1

Name	Bits	Description
SCLK_Period	10..0	Serial clock period
BiSS_Max_Timeout	23..11	The maximum BiSS timeout period
Proc_Trigger	24	Trigger from the processor model
Trigger_Mode	26…25	Trigger mode
Spare	31…27	-

Register 2

Figure 111: BISS_RCP_out Register 2

Name	Bits	Description
Enable	0	BiSS data transmission enable
Error reset	1	Reset the error state
No_of_Bits	2..8	Number of bits to be transmitted
Gray_Start_Adress	9…15	Start address of Gray coded bits in the data frame
Gray_End_Adress	16..22	End address of Gray coded bits in the data frame
Bit_Config_Error	23	Flag bit which stops communication when wrong data frame settings are specified
CRC_Poly_Width	24…28	CRC polynomial width

Register 3

Figure 112: BISS_RCP_out Register 3

Name	Bits	Description
CRC_Polynomial	15…0	CRC polynomial
CRC_Start_Value	31…16	CRC start value

Register 4

Figure 113: BISS_RCP_out Register 4

Name	Bits	Description
Max_SCD_Proc_Time	13…0	Maximum SCD processing time
CRC_Start_Value	14	CLK_Polarity

FPGA Main Component

Block

Figure 114: BISS_RCP FPGA Main block

Block Dialog

The FPGA main block contains the following dialog:

Figure 115: BISS_RCP FPGA Main block

On the page “Interface” you can define the parameters of the input and output registers for automatic interface generation. The dialog also contains a button to start interface generation. For more details about automatic interface generation, refer to the XSG Utils documentation chapter Automatic Interface Generation. The FPGA clock period can be configured too.

Input

The main block has the following inputs:

Name	Unit	Description	Format
SDI	[-]	The serial data received from the sensor	Bool or UFix_1_0
Ext_Trigger	[-]	External trigger to start data transmission	Bool or UFix_1_0
Enable	[-]	Enables the sensor processing	Bool
Trig_Mode	[-]	The communication triggering mode	Ufix_2_0
Proc_Trigger	[-]	Transmission trigger from the processor model	Bool
No_of_Bits	[-]	Number of data bits to be transmitted	UFix_7_0
SCLK_Period	[Clocks]	Period of the serial clock	UFix_11_0
BiSS_Max_Timeout	[Clocks]	Maximum length of the timeout period	UFix_13_0
CPOL	[-]	Clock polarity (idle state of the SCLK line)	Bool
Gray_Decode_Start_Adress	[-]	Start address of Gray coded bits inside the data frame	Ufix_7_0
Gray_Decode_End_Adress	[-]	End address of Gray coded bits inside the data frame	Ufix_7_0
CRC_Poly_Width	[-]	CRC polynomial width	Ufix_5_0
Max_SCD_Proc_Time	[Clocks]	Maximum SCD processing time	Ufix_13_0
Error_Reset	[-]	Reset the error state	Ufix_7_0
CRC_Polynomial	[-]	CRC polynomial equivalent decimal value	Ufix_16_0
CRC_Start_Value	[-]	CRC equivalent decimal start value	Ufix_16_0
Bit_Config_Error	[-]	Wrong data frame settings flag bit	Bool

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	[-]	Contains all output signals which are passed to the processor interface	Bus
SCLK	[-]	The serial clock to be provided for the sensor connected	UFix_1_0
RX_Data	[-]	The data received during the last data transmission	UFix_80_0
Data_Ready	[-]	Impulse indicating that a data transmission has been completed. Can e.g. be used as an interrupt trigger.	Ufix_1_0

Processor Input

Block

Merges the FPGA signals for the postprocessing on the processor side. Received raw data is decoded according to actual data frame configuration. Postprocessed data is separated and provided at the block output ports.

Figure 116: BISS_RCP_in block

Block Dialog

The processor input does not provide any block dialog or mask parameters.

Input

The processor in block has three register and one status bus input: StatusBus provides vector signals based on the actual data frame settings which are used for decoding of the received raw data.

	Input registers must be grouped under the same Register Group ID number in order to ensure synchronous update of the data which is transferred over the several registers!

The processor input block has the following register inputs:

Register 1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_Data(31:0)																															

Figure 117: BISS_RCP_in Register 1

Name	Bits	Description
RX_Data (31:0)	31..0	Bits 31..0 of the received BiSS data frame

Register 2

Figure 118: BISS_RCP_in Register 2

Name	Bits	Description
RX_Data (63:32)	63..32	Bits 63..32 of the received BiSS data frame

Register 3

Figure 119: BISS_RCP_in Register 3

Name	Bits	Description
RX_Data (79:64)	15..0	Bits 79..64 of the received BiSS data frame
Error_ID	18…16	BiSS communication error ID

Input

The processor input block has the following outputs:

Name	Unit	Description	Range
RX_Data	[Raw]	Received raw data	Range: 0…(2^{80}-1) Resolution: 1
Angle_Phi	[deg]	Calculated position in degrees (360 base)	Range: 0°…360° Resolution: 0…2^{-55}*360
Angle_Rev	[-]	Calculated number of full revolutions	Range: +(2^{54}-1)…- 2^{54} Resolution: 1
Angle_Absolute	[deg]	Calculated absolute position in degrees	Range: 0…(2^{55} -1)*360 Resolution: 0…2^{-55}*360
Diagnosis_Bit_Vector	[-]	Received diagnosis bits: Bit1(MSB)…Bit8(LSB)	0|1
MT_Data	[Raw]	Received multiturn data	Range: 0…(2^{55}-1) Resolution: 1
ST_Data	[Raw]	Received singleturn data	Range: 0…(2^{55}-1) Resolution: 1
CRC_Data	[Raw]	Received CRC data	Range: 0…(2^{16}-1) Resolution: 1
BiSS_Comm_Error_ID	[-]	Active error state identifier: ID = 0: No error ID = 1: Timeout error ID = 2: Maximum data line delay error ID = 3: Maximum SCD processing time error ID = 4: Data line low or BiSS frame has not ended yet ID = 5: CRC error ID = 6: Wrong frame setting error (Number of data bits is greater then 64 or CRC bits are selected without data bits)	Range: 0…6 Resolution: 1

HIPERFACE® Sensor Processing

Objective

HIPERFACE® interface is commonly used by position sensors for providing position data. This is a hybrid interface which includes analog process channel and bidirectional parameter channel according to EIA-485 specification and UART standard. HIPERFACE_RCP block handles both process and parameter channel; sine/cosine signals on the analog process channel are used for counting of incremental position changes and fine angle interpolation, while parameter channel can be used for read out of absolute position information and various other parameters from the encoder.

Content

The blockset contains the following elements:

- Processor Interface: HIPERFACE_RCP_out (Processor Interface)
- FPGA Interface: HIPERFACE_RCP_in (FPGA Interface)
- FPGA: HIPERFACE_RCP (FPGA Main Component)
- FPGA Interface: HIPERFACE_RCP_out (FPGA Interface)
- Processor Interface: HIPERFACE_RCP_in (Processor Interface)

Processor Output

Block

Merges the processor signals from General settings (number of multiturn and singleturn bits), UART settings (datarate, parity bit settings, timeout, etc.) and HIPERFACE commands related settings and writes them to the FPGA. Based on the specified settings StatusBus output is updated and shared with HIPERFACE_RCP_in interface.

Figure 120: HIPERFACE_RCP_out block

Block Dialog

The processor output block contains three dialogs. In the first one, the general parameters can be configured.

Figure 121: HIPERFACE_RCP_out dialog; page “General”

The first dialog has the following parameters:

Name	Unit	Description	Range
No. MT bits	[-]	Number of multi turn bits	Range: 0…22 Resolution: 1
No. ST bits	[-]	Number of single turn bits	Range: 7…19 Resolution: 1
Sample time FPGA main block	[s]	Sample time of the FPGA	-

On the second page UART settings can be specified.

Figure 122: HIPERFACE_RCP_out dialog; page “UART Settings”

The second dialog has the following parameters:

Name	Unit	Description	Range
Datarate	[Baud]	Data rate of the UART communication (reception and transmission): 0: 600 1: 1200 2: 2400 3: 4800 4: 9600 5: 19200 6: 38400	Range: 0…6 Resolution: 1
Parity bit settings	[-]	Parity of the UART communication: 1: No parity bit is used 2: Even parity 3: Odd parity	Range: 1|2|3
Timeout condition	[s]	Timeout condition which specifies end of the slave data transmission: 1: 22/data rate 2: 55/data rate	Range: 1|2
TX driver ON delay	[s]	Period between activation of TX driver and start of the master transmission frame. NOTE: It is important that TX driver is enabled before start of the transmission; this will ensure that Start bit will be properly signalized with transition from high to low logic state. In order to fulfill that requirement use default or adjusted non zero value for this parameter.	Range: 0…(2^16-1)*Ts_FPGA Resolution: Ts_FPGA
TX driver OFF delay	[s]	Period between end of the master transmission frame and deactivation of TX driver. NOTE: It is important that this parameter is not set to value greater then sum of timeout and response time.	Range: 0…(2^16-1)*Ts_FPGA Resolution: Ts_FPGA

Maximum response time	[s]	Maximum expected slave response time. If slave does not respond within specified time response time error will be detected (Comm_Error_ID=3).	Range: 0….0.25s Resolution: Ts_FPGA
Length of valid pulse	[s]	Time period in which input data should stay on the same logic level after transition in order to detect valid edge. Prevents false edge triggering in cases when parameter channel is not properly terminated (Please refer to HIPERFACE documentation for more details).	Range:0….(2^18-1)*Ts_FPGA Resolution: Ts_FPGA

On the third page HIPERFACE Commands specific settings can be specified.

Figure 123: HIPERFACE_RCP_out dialog; page “HIPERFACE Commands”

The third dialog has the following parameters:

Name	Unit	Description	Range
Slave address	[-]	Slave address entry in hexadecimal format. NOTE: By default broadcast address ‘FF’ is used.	Range: 0x00…0xFF Resolution: 0x01
Access code 0…3	[-]	Access code 0...3 entry in hexadecimal format. NOTE: By default all access codes are set to ‘55’ which is access code 0 default value.	Range: 0x00…0xFF Resolution: 0x01
Datarate	[Baud]	Data rate of the UART communication to be set when “Set Serial Interface” command is used: 0: 600 1: 1200 2: 2400 3: 4800 4: 9600 5: 19200 6: 38400	Range: 0…6 Resolution: 1
Parity bit settings	[-]	Parity of the UART communication to be set when “Set Serial Interface” command is used: 0: No parity bit is used 2: Even parity 3: Odd parity	Range: 0|2|3 Resolution: 1
Timeout condition	[-]	Timeout condition of the UART communication to be set when “Set Serial Interface” command is used: 1: 22/data rate 2: 55/data rate	Range: 0|1 Resolution: 1

Select analog channel	[-]	Selects analog channel for read out when “Read Analog Value” command is used: 1: 48h – Encoder Temperature 2: F0h – Legacy Encoder Temperature 3: xxh – Other Analog Channel (Enables freely selectable analog channel address inside entry window below which opens when this option is selected)	Range: 1|2|3 Resolution: 1
Analog channel address	[-]	Analog channel address for read out when “Read Analog Value” command is used	Range: 0x00…0xFF Resolution: 0x01

Input

The HIPERFACE_RCP_out block has the following inputs:

Name	Unit	Description	Range
Enable	[-]	Enables the encoder processing	0|1
Reset	[-]	Resets the encoder processing	0|1
Command_Sel	[-]	Select HIPERFACE command: 0: Read Position 1: Read Analog Value 2: Read Counter 3: Increment Counter 4: Reset Counter 5: Read Data 6: Store Data 7: Data Field Status 8: Memory Status 9: Read Encoder Status 10: Read Type Label 11: Encoder Reset 12: Read Version 13: Set Serial Interface	Range: 0…13 Resolution: 1
Set_Command	[-]	Triggers transmission of the selected command	0|1
Access_Code	[-]	Select one of four access codes defined inside the block mask. Selected access code will be used when command which requires access code is send (e.g. Command “Reset Counter”).	0|1|2|3
Pos_Sync_Trig	[-]	Triggers absolute position synchronization; read position command is send and internal counting logic is updated with absolute position received from the encoder. NOTE: This function must be called at standstill of the motor!	0|1
Reset_Error	[-]	Resets the logic after entering in the error state	0|1
Reset_Error_Code	[-]	Resets received error codes (“Error_Resp_Code” and “Error_Code_Vector”) which are available at the HIPERFACE_RCP_in block output.	0|1
Store_Data	[-]	Decimal value of byte which will be stored inside encoder memory when “Store Data” command is used	Range: 0…255 Resolution: 1

Data_Field_Sel	[-]	Select data field for reading and storing data inside the encoder memory	Range: 0…255 Resolution: 1
Data_Field_Offset	[-]	Offset within selected data field used for reading and storing data inside encoder memory	Range: 0…127 Resolution: 1
Num_of_Bytes	[-]	Number of bytes for read out when command “Read Data” is used	Range: 1…128 Resolution: 1

	Absolute position synchronization must be called at standstill of the motor! Also by default it is asumed that absolute position is synchonized with the start of each sine/cosine period. If this is not the case then synchronization error is possible. If there is known alignment offset between absolute position and sine/cosine signals it should be specified under the mask with constant „Abs_Pos_Alignment[0_31]“!

Output

The Processor out block has two outputs: StatusBus (signals used for postprocessing of the received data inside HIPERFACE_RCP_in block) and Buffer_1 output which sectioning is shown below:

Register 1

Figure 124: HIPERFACE_RCP_out Register 1

Name	Bits	Description
Enable	0	Enables the encoder processing
Reset	1	Resets the encoder processing
Error_Reset	2	Resets the logic after entering in the error state
Cmd_Trigger	3	Triggers command transmission
Pos_Sync_Trig	4	Triggers absolute position synchronization
Parity	6…5	Parity bit settings
Timeout	30…7	Timeout condition expressed as number of FPGA clocks
Spare	31	-

Register 2

Figure 125: HIPERFACE_RCP_out Register 2

Name	Bits	Description
Datarate	17…0	Selected datarate or time length of the single bit inside transmission byte, expressed as number of FPGA clocks
ST_Bits	22…18	Number of single turn bits
Spare	31…23	-

Register 3

Figure 126: HIPERFACE_RCP_out Register 3

Name	Bits	Description
Set_Position	31…0	Calculated raw value of set position

Register 4

Figure 127: HIPERFACE_RCP_out Register 4

Name	Bits	Description
Command_ID	4…0	Selected command ID number
Access_Code	12…5	Selected access code
Analog_Ch_Address	20…13	Address of the selected analog channel
Analog_Ch_Sel	22…21	Selected analog channel ID number
Slave_Address	30…23	Selected slave address

Register 5

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Abs_Pos_Alignment							Max_Resp_Time																							

Figure 128: HIPERFACE_RCP_out Register 5

Name	Bits	Description
Max_Resp_Time	24…0	Slave maximum expected response time
Abs_Pos_Alignment	29…25	Absolute position alignment shift

Register 6

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Store_Data								Num_of_Bytes								Data_Field_Offset								Data_Field_Sel							

Figure 129: HIPERFACE_RCP_out Register 6

Name	Bits	Description
Data_Field_Sel	7…0	Selected data field
Data_Field_Offset	15…8	Selected data field offset
Num_of_Bytes	23…16	Selected number of bytes for data read out
Store_Data	31…24	Store data byte

Register 7

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						Valid_Edge																		UART_Set							

Figure 130: HIPERFACE_RCP_out Register 7

Name	Bits	Description
UART_Set	7…0	UART settings byte
Valid_Edge	25…8	Valid edge detection period as number of FPGA clock cycles

Register 8

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_OFF_Delay																TX_ON_Delay															

Figure 131: HIPERFACE_RCP_out Register 8

Name	Bits	Description
TX_ON_Delay	15…0	TX driver ON delay
TX_OFF_Delay	31…16	TX driver OFF delay

FPGA Main Component

Block

Figure 132: HIPERFACE_RCP FPGA Main block

Block Dialog

The FPGA main block contains the following dialog:

Figure 133: HIPERFACE_RCP FPGA Main block

On the page “Interface” you can define the parameters of the input and output registers for automatic interface generation. The dialog also contains a button to start interface generation. For more details about automatic interface generation, refer to the XSG Utils documentation chapter Automatic Interface Generation. The FPGA clock period can be configured too.

Input

The main block has the following inputs:

Name	Unit	Description	Format
RX_Data	[-]	Slave TX data send on the parameter channel	Bool or UFix_1_0
Sine	[-]	Sine signal received on the analog process channel, scaled inside range [-1,1]	Max UFix32_31
Cosine	[-]	Cosine signal received on the analog process channel, scaled inside the range [-1,1]	Max UFix32_31
From_Interface_in	[-]	SignalBus which is connected to the HIPERFACE_RCP_in block	

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	[-]	Contains all output signals which are passed to the processor interface	Bus
Angle_Absolute	[Raw]	Current absolute angle: 31…0 Bit: Single turn bits 53…32 Bit: Multi turn bits	UFix_54_0
Angle_Phi	[Raw]	Current single turn angle	UFix_32_0
Angle_Rev	[-]	Current number of multiturn revolutions	Fix_22_0

Processor Input

Block

Interface which decodes received signals from FPGA side. Absolute position information together with data received on the parameter channel are provided at the output.

Figure 134: HIPERFACE_RCP_in block

Block Dialog

The processor input does not provide any block dialog or mask parameters.

Input

The processor in block has two register, one buffer and one status bus input: Absolute position information is updated over the two registers, while buffer is used for transmission of data which is received on the parameter channel. StatusBus provides signals which are used for the postprocessing of the received data.

	Input registers must be grouped under the same Register Group ID number in order to ensure synchronous update of the data which is transferred over the several registers!

The processor input block register sectioning is shown below:

Register 1

Figure 135: HIPERFACE_RCP_in Register 1

Name	Bits	Description
ST_Bits(31:0)	31...0	Absolute angle single turn bits

Register 2

Figure 136: HIPERFACE_RCP_in Register 2

Name	Bits	Description
MT_Bits(21:0)	21…0	Absolute angle multi turn bits

The processor input block buffer has a total of 33 registers: First 32 registers are reserved for update of the received data bytes on the parameter channel, while last register holds diagnostics signals. This buffer is updated at the end of the parameter channel communication cycle. Sectioning of data inside the buffer is shown below:

Buffer in 1

Figure 137: HIPERFACE_RCP_in Buffer in 1

Name	Bits	Description
Comm_Error_ID	2..0	Communication error ID number
Cmd_Fdbk	7…3	Command feedback from the slave
Error_Resp_Flag	8	Error response flag
Warning_Bit	9	Warning bit in the slave response
Slave_Adress	17…10	Address feedback from the slave
Data_Field_Fdbk	25…18	Data field feedback from the slave
Spare	31…26	-

Input

The processor input block has the following outputs:

Name	Unit	Description	Range
Busy	[-]	Busy flag bit which is active during the command processing. While active new commands will not be accepted.	Range: 0|1
Angle_Absolute	[Raw]	Received raw absolute angle	Range: 0…(2^{54}-1) Resolution: 1
Angle_Absolute	[deg]	Calculated absolute angle in degrees	Range: (+2^{21}…-2^{21})*360 Resolution: 2^{-32}*360
Angle_Phi	[deg]	Calculated angle in degrees (360 base)	Range: 0°…360° Resolution: 2^{-32} *360
Angle_Rev	[-]	Calculated number of full revolutions	Range: +2^{21}-1… -2^{21} Resolution: 1
Read_Data_Array	[-]	Data array which is updated with data bytes received when command “Read Data” is triggered	Range: 0…255 Resolution: 1
Error_Code_Vector	[-]	Vector which holds last 4 error codes received	Range: 0…255 Resolution: 1
Error_Response_Code	[-]	Received error code during error response	Range: 0…255 Resolution: 1
Warning_Bit	[-]	Indicates that warning bit was active during the last slave response	Range: 0|1
Comm_Error_ID	[-]	Active error state identifier: ID = 0: No Error ID = 1: Checksum Error ID = 2: Parity Error ID = 3: Response Time Error ID = 4: RX Message Length Error ID = 5: Position Synchronization Error	Range: 0…5 Resolution: 1
Parameter_Bus - Contains parameters which are extracted and grouped based on the slave response. Description of each parameter is given below:			

Temperature	[°C]	Calculated temperature value when “Read Analog Value” command is used for reading of the encoder temperature from analog channel 48h or F0h	Range (48h): $-2^{15}\ldots+2^{15}-1$ Resolution (48h): 1 Range (F0h): $(-2^{15}\ldots+2^{15}-1)/2.048$ Resolution (F0h): 1/2.048
Counter_Value	[-]	Counter value updated when “Read Command” is triggered	Range: $0\ldots2^{24}-1$ Resolution: 1
Slave_Address	[-]	Address feedback from the slave updated on any command call	Range: 0…255 Resolution: 1
Absolute_Position	[Raw]	Absolute position raw data updated when “Read Position” command is triggered	Range: $0\ldots2^{32}-1$ Resolution: 1
Analog_Value	[Raw]	Analog channel raw value updated when command “Read Analog Value” is triggered	Range: $0\ldots2^{16}-1$ Resolution: 1
Data_Field_No	[-]	Data field number updated when command “Data Field Status” is triggered	Range: 0…255 Resolution: 1
Data_Field_Size	[-]	Data field size in 16 byte blocks, updated when command “Data Field Status” is triggered	Range: 0…7 Resolution: 1
Code_Enabled	[-]	Code enabled/disabled flag bit, updated when command “Data Field Status” is triggered	Range: 0|1
Access_Code	[-]	Access code for selected data field, updated when command “Data Field Status” is triggered	Range: 0|1|2|3
Write_En	[-]	Write enable flag bit, updated when command “Data Field Status” is triggered: 0: Read only 1: Write enabled	Range: 0|1
Data_Field_Exists	[-]	Flag bit which is updated when command “Data Field Status” is triggered	Range: 0|1

Free_User_Memory	[-]	Free user memory as number of 16 byte block, updated when command “Memory Status” is triggered	Range: 0…255 Resolution: 1
Existing_Data_Fields_No	[-]	Number of existing data fields, updated when command “Memory Status” is triggered	Range: 0…255 Resolution: 1
UART_Settings	[-]	UART settings byte updated when command “Read Type Label” is triggered	Range: 0…255 Resolution: 1
Encoder_Type	[-]	Encoder type byte updated when command “Read Type Label” is triggered	Range: 0…255 Resolution: 1
EEPROM_Size	[-]	Memory size in 16 byte blocks, updated when command “Read Type Label” is triggered	Range: 0…255 Resolution: 1
Option_Code	[-]	Updated when command “Read Type Label” is triggered	Range: 0…255 Resolution: 1
Serial_Number	[-]	ASSCII code array, updated when command “Read Version” is triggered	Range: 0…255 Resolution: 1
Firmware_Version	[-]	ASSCII code array, updated when command “Read Version” is triggered	Range: 0…255 Resolution: 1
Firmware_Date	[-]	ASSCII code array, updated when command “Read Version” is triggered	Range: 0…255 Resolution: 1
Encoder_Type	[-]	Encoder type parameter, updated when at least 2 data bytes are read from virtual data field ‘FF’: 0: Rotary Encoder 1: Linear Encoder	Range: 0|1
Counting_Type	[-]	Counting type parameter, updated when at least 2 data bytes are read from virtual data field ‘FF’: 0: Unipolar 1: Bipolar	Range: 0|1

No_of_Periods_per_Turn	[-]	Encoder resolution; Number of periods/turn for rotary encoder or period length in 1nm steps for linear encoder. This parameter is updated when at least 6 data bytes are read from the virtual data field ‘FF’.	Range: $0…2^{32}-1$ Resolution: 1
No_of_Revolutions	[-]	Number of coded revolutions for rotary encoder or number of periods for linear encoder. This parameter is updated when at least 6 data bytes are read from the virtual data field ‘FF’.	Range: $0…2^{32}-1$ Resolution: 1

Asymmetric H Bridge PWM Generator

Overview

The SRM PWM generates PWM for control an Asymmetric H Bridge inverter used to contrl a Switched reluctance machine. The SRM PWM block set can generate a rising edge or a pulse center synchronous PWM signal. A pulse center-synchronous interrupt can also be raised on the FPGA and caught on the processor side.

Asymmetric H-Bridge PWM

SRM PWM is used to supply a Switched reluctance motor. The switched reluctance machine is a doubly salient machine, i.e. both rotor and stator have poles on them. The inductance of the motor is a function of position and current (if the motor is operating in saturation region). The standard inverter topology used to control a Switched reluctance machine is called Asymmetric H Bridge and it can be seen in Figure 1. The "single phase leg circuit" as shown in Figure 1 is the smallest unit of the inverter. There are two switches needed per phase and two anti-parallel diodes for freewheeling

Figure 138: Asymmetric H-Bridge Inverter Topology

For higher number of phases the legs are connected in parallel to the DC link and can be controlled independently with PWM signals. Each leg of the inverter has have three states namely;

- Conducting
- Free wheeling
- Open

These three stages can be seen in Figure 2 a, b and c respectively. During conducting stage both the switches on the high side (HSD) and the Low side (LSD) are simultaneously conducting ang and putting the DC link voltage across the motor winding. During the freewheeling stage only one of the switches are

conducting while the other is off, this allows the current to pass through the conducting switch and anti-parallel diode. In this stage the voltage across the phase is due to the energy stored in the inductor. And finally, in the open stage the voltage across the phase is 0 as the antiparallel diodes are open in this condition

Figure 139: States of an Asymmetric H Bridge inverter

It is important to note that the switch in conducting state during the freewheeling is alternated between S1 and S2 in order to uniformly distribute the current over both switches and thereby uniformly distributing the heat losses. The following is a description of two-phase Asymmetric H Bridge PWM generator

Content

The blockset contains the following elements:

- Processor Interface: TWO_PHASE_ASYM_H_BRIDGE_PWM_GENERATOR_RCP_out (Processor Interface)
- FPGA Interface: TWO_PHASE_ASYM_H_BRIDGE_PWM_GENERATOR_RCP_in (FPGA Interface)
- FPGA: TWO_PHASE_ASYM_H_BRIDGE_PWM_GENERATOR_RCP (FPGA Main Component)
- FPGA Interface: TWO_PHASE_ASYM_H_BRIDGE_PWM_GENERATOR_RCP_out (FPGA Interface)
- Processor Interface: TWO_PHASE_ASYM_H_BRIDGE_PWM_GENERATOR_RCP_in (FPGA Interface)

Processor Output

Block

Figure 140. TWO_PHASE_ASYM_H_BRIDGE_PWM out block

Block Dialog

The processor output of the block-set contains the following dialog elements:

Figure 141: TWO_PHASE_ASYM_H_BRIDGE_PWM Parameter dialog

The Parameter dialog of the Block Commutated blockset has the following parameters:

Name	Unit	Description	Range
PWM Aligning Mode	[-]	Specify the modulation and the update algorithm of the duty cycle data from the inputs 0: rising-edge modulation and update 1: pulse-center modulation and update	0 … 3

		2: pause-center modulation and update 3: pulse-center modulation with an update of the duty cycles every pulse- and pause center of the pwm	
Downsampling update trigger	[-]	If the duty cycle data is updated in the FPGA main block an update trigger flag is generated. The trigger event can be down sampled with this factor	1 … 1023
Sample time FPGA	[s]	Sample time of the FPGA	1…..

Input

The processor output of the block-set has the following inputs:

Name	Unit	Description	Range
Enable	[-]	Enable of the output	0|1
PWM Frequency	[Hz]	PWM Frequency at disabled external trigger	0 … 5 MHz
Duty_Cycle A[-1_1]	[-]	Duty cycle (scalar)	-1 ……1
Duty_Cycle B[-1_1]	[-]	Duty cycle (scalar)	-1 ……1
Enable_Ext_ Trigger	[-]	Enable external pulse center synchronization signal	0|1

⚠	Duty Cycles for the same phase cannot be High or PWM at the same time. For example, A+ Sector 1: High and A- Sector 1: High or PWM. This configuration would Short circuit the Inverter.

Output

The processor output block provides 4 registers whose sectioning is shown below:

Register 1 and 2

Figure 142: TWO_PHASE_ASYM_H_BRIDGE_PWM _out Register 1

Name	Bits Used	Description	Range
Duty_Cycle	16…0	Duty cycle is a scalar between 0 and 1	-1…1

Register 3

Figure 143:TWO_PHASE_ASYM_H_BRIDGE_PWM _out Register 3

Name	Bits Used	Description	Range
PWM Period Time	23…0	Period time of the PWM	0 … 2^24-1 represents 0 … 0.1678 s
SPARE	28…24		
Enable	29	Output Enable	0|1
Aligning Mode	31…30	PWM aligning mode	0…3

Register 4

Figure 144: TWO_PHASE_ASYM_H_BRIDGE_PWM _out Register 4

Name	Bits Used	Description	Range
SPARE	19…0		
Enable ext Trigger	20	Enable external Trigger	0|1
Downsampling	30…21	Down sampling of the update trigger flag	1…1023
SPARE	31		

	The parameters of the input and output registers can be defined for automatic interface generation (When checkbox is enabled) on the page Interface. The dialog also contains an additional button to start interface generation. For more details about automatic interface generation, refer to Automatic Interface Generation chapter of XSG_Utils Documentation.

FPGA Main Component

Block

Figure 145: TWO_PHASE_ASYMMETRIC_H_BRIDGE_PWM_GENERATOR Main block

Block Dialog

The FPGA main blockset contains the following dialog.

Figure 146: TWO_PHASE_ASYMMETRIC_H_BRIDGE_PWM_GENERATOR Main block dialog

Input

The main block has the following inputs:

Name	Unit	Description	Format
Ext_Trigger	[-]	External pulse center trigger of an external PWM.	Bool
Enable	[-]	Enable of the output	Bool
Enable_Ext_ Trigger	[-]	Enable of the external tirgger	Bool
Period_Time	[Raw]	Period time of the PWM at disable external trigger	UFix_24_0
Downsampling_ Up_Tr	[-]	Downsampling of the update trigger flag	UFix_10_0
Duty_Cycle_A_[-1_1]	[-]	Duty cycle (scalar)	Fix_17_16
Duty_Cycle_B_[-1_1]	[-]	Duty cycle (scalar)	Fix_17_16

Output

The blockset contains the following elements:

Name	Unit	Description	Format
FPGASignalBus	Bus	Bus containing the blocks most significant signals	-
UpdateTrigger	[-]	Update trigger flag	Bool

En	[-]	Enable Output	Bool
PWM_HSD_A	[-]	High side of gate signal	Bool
PWM_LSD_A	[-]	Low side of gate signal	Bool
PWM_HSD_B	[-]	High side of gate signal	Bool
PWM_LSD_B	[-]	Low side of gate signal	Bool

Processor Input

Block

Adapts the FPGA signals for the processor side

Figure 147: TWO_PHASE_ASYM_H_BRIDGE_PWM_in block

Block Dialog

The following parameters can be configured in the processor input dialog:

Figure 148: TWO_PHASE_ASYM_H_BRIDGE_PWM_in dialog

The dialog blockset has the following parameters:

Name	Unit	Description	Range
Sample time FPGA	[s]	Sample time of the FPGA.	-

Input

The processor input block has the following inputs:

Register 1

Figure 149: TWO_PHASE_ASYM_H_BRIDGE_PWM_in Register_1

Name	Bits Used	Description	Range
PWM Period Time	23…0	Period time of the PWM	0 … 2^24-1
Valid Output	25	Valid flag for the PWM output	0|1

Output

The processor input block has the following outputs:

Name	Units	Description	Range
PWM Period Time	[s]	Period time of the PWM	0 … 83,9 ms
Valid Output	[-]	Valid flag for the PWM output	0|1

Drive Safety

Overview

The drive safety block protects a three phase motor in case of over current or current sensor failure. The block continuously monitors the current levels across all the three phases of the motor and in case of a failure, the motor power supply (gate signals of semiconductor) can be switched off. This block offers threes types of failure monitoring: Zero current, maximum phase current and maximum phase current over time.

Drive Safety

The current waveforms of a balanced load such as a three-phase motor are symmetric with 120 degrees phase lag. The power transfer of this system is dependent on the magnitude of current. When an alpha beta transformation is performed on the three phase currents then the output is a two-phase current waveform which has a 90-degree phase lag and a zero-sequence current. As long as the inputs are symmetric with 120-degree phase shift the zero-sequence current is negligible (round about zero).

In case of unbalance or a sensor failure, the zero-sequence current is no longer negligible. When this value exceeds a user specified threshold, this triggers a failure flag which can be used tfor various protection scenarios (e.g. switch off the controller output voltage).
The squared sum of the alpha and beta currents (I2) is the same as the square of three phase currents. This value should be within a predefined range for safe motor operation. In case the value exceeds a maximum limit, an error flag is generated.
Additionally, the I2 current values are passed through a first order low pass filter (PT1). If the filter value exceeds the user specified threshold an error flag is generated.

During set point changes and the start of a motor there are current peaks fora short time. This low duration of high current is normal for any motor operation, so that the I2 threshold should be bigger than these peak values and the filtered value (I2_Fi) is typically related to the rated current of the motor (samller than the I2)

Finally a global error flag is generated. This is a configurable error flag which is dependent on each of the three errors generated. If one error occurs the global error changes it state to high until an error reset is performed. All three safety functions can be activated individually.

The inputs to the drive safety block are the measured three phase current of the emotor to be controlled. The output of the drive safety block are the error flags and its corresponding values.

Content

The blockset contains the following elements:

- Processor Interface: DRIVE_SAETY_out (Processor Interface)
- FPGA Interface: DRIVE_SAFETY_in (FPGA Interface)
- FPGA: DRIVE_SAFETY(FPGA Main Component)
- FPGA Interface: DRIVE_SAFETY_out (FPGA Interface)
- Processor Interface: DRIVE_SAFETY _in (Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to FPGA.

Figure 150: DRIVE_SAFETY_out block

Block Dialog

The processor output of the block-set contains the following dialog elements:

Figure 151: DRIVE_SAFETY Parameter dialog

The Parameter dialog of the drive safety blockset has the following parameters:

Name	Unit	Description	Range
Sample time FPGA	[s]	Sample time of the FPGA	-

Input

The processor output of the block-set has the following inputs:

Name	Unit	Description	Range
Reset_All_Error[0|1]	[0|1]	Resets all error flags	0|1
i_Zero_En[0|1]	[0|1]	I Zero protection enable	0|1
i_Zero_Max[A]	[A]	Absolute maximum allowable zero Current	0…2048
i_Zero_Rst[0|1]	[0|1]	Reset I Zero protection Flag	0|1
i2_En[0|1]	[0|1]	I squared protection enable	0|1
i2_Max[A2]	[A]	Maximum allowable I squared value	0…262144
i2_Rst[0|1]	[0|1]	Reset I squared protection flag	0|1
i2_Fi_En[0|1]	[0|1]	I squared filtered protection enable	0|1
i2_Fi_Max[A2]	[A]	Maximum allowable filtered I squared value	0…262144
i2_Fi_Rst[0|1]	[0|1]	Reset of the integrator used inside the PT1 element	0|1
i2_Tfi[s]	[s]	Filter time of the first-order lag	[0 …~10] s (depends on FPGA sample time)

Output

The processor output block provides 4 registers whose sectioning is shown below. Resister 1 to 3 have similar structure. Registers 1, 2 and 3 are for Zero current, I2 and i2 Filtered Protection respectively.

Register 1 to 3

Figure 152: DRIVE_SAFETY _out Register 1_3

Name	Bits Used	Description	Range
Max	24…0	Maximum allowable threshold value	0…327786
SPARE	29..25		
Enable	30	Enable Protection	0|1
Reset	31	Reset Error Flag	0|1

Register 4

Figure 153. DRIVE_SAFETY _out Register 4

Name	Bits Used	Description	Range
I2_Tf	30…0	Filter time of the first-order low pass current squared filter	[0 …~10] s (depends on FPGA sample time)
Reset	31	Reset of the FPGA integrator used	0|1

FPGA Main Component

Block

Figure 154: DRIVE_SAFETY Main block

Block Dialog

The FPGA main blockset contains the following dialog.

Figure 155: DRIVE_SAFETY Main block dialog

Input

The main block has the following inputs:

Name	Unit	Description	Format
ia	[A]	Phase A current	User Defined
ib	[A]	Phase B current	User Defined
ic	[A]	Phase C current	User Defined
Reset_All_Error	[-]	Resets all error states	Bool

i_Zero_En	[-]	Enables zero current error checking	Bool
i_Zero_Max	[A]	Maximum allowable zero current	UFix_18_7
i_Zero_Rst	[-]	Reset zero current error State	Bool
i2_En	[-]	Enables i2 current error checking	Bool
i2_Max	[A2]	Maximum allowable i2 current	UFix_25_7
i2_Rst	[-]	Reset i2 filtered current error state	Bool
i2_En	[-]	Enables i2 filtered current error checking	Bool
i2_Fi_Max	[A2]	Maximum allowable i2 filtered current	UFix_25_7
i2_Rst	[-]	Reset i2 filtered current error state	Bool
I2_Ts|Tfi	[RAW]	Time constant for low pass filter	UFix_31_31

Output

The blockset has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	Bus	Bus containing the blocks most significant signals	-
i_Zero_Abs_Error	[-]	Zero current error flag	Bool
i2_Error	[-]	Current squared error flag	Bool
i2_Fi_Error	[-]	Current squared filtered error flag	Bool
Global_Error	[-]	Global error flag	Bool

Processor Input

Block

Adapts the FPGA signals for the processor side

Figure 156. DRIVE_SAFETY_in block

Input

The processor input block has the following inputs:

Register 1, 3

Figure 157: DRIVE_SAFETY _in Register_1 and 3

In case of register 1

Name	Bits Used	Description	Range
i_Abs_Zero	17…0	Zero Sequence Current Free running	0 … 2048

In case of register 3

Name	Bits Used	Description	Range
i_Abs_Zero_Hold	17…0	Zero Sequence Current during error	0 … 2048

Register 2, 4, 6

Figure 10: DRIVE_SAFETY _in Register_2, 4 and 5

In case of register 2

Name	Bits Used	Description	Range
i2	25…0	Actual value i2 current	0 … 262144

In case of register 4

Name	Bits Used	Description	Range
i2_hold	25…0	i2 current in case of Error	0 … 262144

In case of register 5

Name	Bits Used	Description	Range
i2_Fi_Hold	25…0	i2 Current Filtered in case of Error	0 … 262144

Register 6

Figure 158: DRIVE_SAFETY _in Register_6

Name	Bits Used	Description	Range
Global Error	0	Atleast one of the error flag is set	0|1
i2_Fi Error	1	i2 current filtered error flag	0|1

i2 Error	2	i2 current error flag	0|1
I_Zero_Error	3	Zero current error flag	0|1
Down sampling Factor	4..8	Down sampling Factor of the Drive safety Block	1..31

Register 7

Figure 159: DRIVE_SAFETY _in Register_7

Name	Bits Used	Description	Range
i2_Fi	30…0	i2 current filtered Free running	0 … 262143
Data Type	31	Type of Data coming back from FPGA	0|1

Output

The processor input block has the following outputs:

Name	Units	Description	Range
Global Error	[-]	Any of the Error Flags are set	0|1
i_Zero_Abs	[A]	Actual absolute value of zero current	0 …. 2048
i_Zero_Abs_Hold	[A]	Absolute value of zero current which is stored when error flag is set	0 …. 2048
i_Zero_Abs_Error	[-]	Zero current error flag is set	0|1
i2	[A2]	Actual absolut value of I square current	0 …. 262144

i2_Hold	[A2]	Absolut value of I square current which is stored when error flag is set	0 …. 262144
i2_Error	[-]	I2 Error Flag	0|1
i2_Fi	[A2]	Actual I2 filtered value	0 …. 262144
i2_Fi_Hold	[A2]	I2 Filtered in case of Error	0 …. 262144
i2_Fi_Error	[-]	I2 Filtered Error Flag	0|1

Space Vector Modulator

Overview

Voltage to a three-phase motor is delivered by a three-phase inverter. The input to the inverter is a DC voltage supplied by two wires one at higher DC potential than the other. Each phase of the inverter has two switches one which connects one motor phase to the higher DC voltage, referred to as High side switch, and other to the lower DC voltage, referred to as Low side Switch.

The three-phase space vector modulator block generates three phase duty cycles from three phase voltage setpoints. The calculated duty cycles serve as inputs to the three phase PWM generator, which generates the gate control signals for the inverter switches namely the high and Low side switches for each phase. Input to the space vector modulator are the three-phase voltage setpoints that are to be generated by the inverter.

Space vector modulation allows for better usage of DC link capacity as opposed to conventional sine commutation. The maximum voltage of space vector PWM is 15% higher than sine commutation.

Space Vector

A three phase Permanent magnet machine or an induction machine is usually driven by a three-phase sinusoidal voltage, similar to the three-phase power widely available from the grid. The three phase voltages have the same magnitude and lag each other by 120 degrees. In a standard control application, the motor controller generates a three-phase voltage set point. Then a three-phase modulator generates the corresponding Three phase duty cycle. The PWM generator converts the three phase duty cycles into high side and low side switching signals for each phase. When the gate signals are applied to the gates of the inverter it generates a switched three phase voltage across the motor phase windings. Since the motor is an inductive load the switched voltage waveform gets filtered sinusoidal voltage waveform. To summarize the Modulator PWM generator and the inverter amplify the voltage set points produced by the controller to power level.

In a three-phase inverter there are six switches, two per phase. Hence there are eight possible switching states (2^3). The details of each switching state can be seen in the table below;

Vector	A HSD	B HSD	C HSD	A LSD	B LSD	C LSD	VAB	VBC	VCA
V0	OFF	OFF	OFF	ON	ON	ON	0	0	0
V1	ON	OFF	OFF	OFF	ON	ON	$+V_{dc}$	0	$-V_{dc}$
V2	ON	ON	OFF	OFF	OFF	ON	0	$+V_{dc}$	$-V_{dc}$
V3	OFF	ON	OFF	ON	OFF	ON	$-V_{dc}$	$+V_{dc}$	0
V4	OFF	ON	ON	ON	OFF	OFF	$-V_{dc}$	0	$+V_{dc}$
V5	OFF	OFF	ON	ON	ON	OFF	0	$-V_{dc}$	$+V_{dc}$
V6	ON	OFF	ON	OFF	ON	OFF	$+V_{dc}$	$-V_{dc}$	0
V7	ON	ON	ON	OFF	OFF	OFF	0	0	0

Here HSD refers to the high side switch and LSD refers to the low side switch. The corresponding phase voltages hence generated are shown in the last three columns. Note that the HSD and LSD of a phase cannot be in ON state simultaneously as it would short circuit the inverter leg.

Content

The blockset contains the following elements:

- FPGA: SPACE_VECTOR(FPGA Main Component)
- FPGA Interface: SPACE_VECTOR_out (FPGA Interface)
- Processor Interface: SPACE_VECTOR _in (Processor Interface)

FPGA Main Component

Block

Figure 160: SPACE_VECTOR Main block Floating Point implementation (left) and Fixed Point implementation (right)

Block Dialog

The FPGA main block set contains the following dialog.

Figure 161: SPACE_VECTOR Main block dialog

Input

The main block has the following inputs:

Name	Unit	Description	Format
Update	[-]	Update trigger for the space vector block	Bool
Reset	[-]	Resets all outputs	Bool
V_a_Set	[V]	Phase A Voltage	Fix_25_x/ xFloat_8_24
V_b_Set	[V]	Phase B Voltage	Fix_25_x/ xFloat_8_24
V_c_Set	[V]	Phase C Voltage	Fix_25_x/ xFloat_8_24
Abs_V_DC_Set	[V]	Absolute value of DC link voltage	UFix_25_x/ xFloat_8_24

Output

The blockset has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	Bus	Bus containing the blocks most significant signals	-
HSD_A	[-]	High side duty cycle	UFix_16_15
LSD_A	[-]	Low side duty cycle	UFix_16_15
HSD_B	[-]	High side duty cycle	UFix_16_15
LSD_B	[-]	Low side duty cycle	UFix_16_15
HSD_C	[-]	High side duty cycle	UFix_16_15
LSD_C	[-]	Low side duty cycle	UFix_16_15

Processor Input

Block

Adapts the FPGA signals for the processor side

Figure 162: SPACE_VECTOR_in block

Input

The processor input block has the following inputs:

Register 1, 2, 3

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Duty_Cycle_LSD_Phase_X																Duty_Cycle_HSD_Phase_X															

Figure 163: SPACE_VECTOR _in Register_1, 2and 3

Name	Bits Used	Description	Range
Duty_Cycle_HSD_Phase_X	15…0	Duty cycle set for the high side switch of Phase X	0 … 1
Duty_Cycle_LSD_Phase_X	31…16	Duty cycle set for the low side switch of Phase X	0 … 1

Output

The processor input block has the following outputs:

Name	Units	Description	Range
Duty_Cycle_HSD[a;b;c][0_1]	[-]	Duty cycle set for the high side switch of all Phases	0_1
Duty_Cycle_LSD[a;b;c][0_1]	[-]	Duty cycle set for the low side switch of all Phases	0_1

Schematic

The block modulation can be interfaces with the XSG_Utils PWM generator with the help of the help of a ROUTE_PWM signals block. A schematic can be seen below.

Figure 164: Space Vector Modulator and route PWM schematic

For further information, and screen shot of an example implementation please see ROUTE_PWM_SIGNALS sub section in the Miscellaneous Chapter.

Block Commutated PWM Generator

Overview

The Block Commutation block configures the block commutator that is implemented on the FPGA. It allows controlling one BLDC via block commutation. The Block Commuted PWM generator blockset can generate a rising edge or a pulse center synchronous PWM signal (single phase and three phase). The dead time and the duty cycle can be set on the processor side (online tuneable). A pulse center-synchronous interrupt can also be raised on the FPGA and caught on the processor side.

on the FPGA and caught on the processor side.

Block Commutation

Block commutated PWM is used to supply a BLDC motor. One complete electrical rotation is divided into many sectors. The sectors (changing for multiple times in every electric cycle) is used to change the phase current flow direction. However, the current direction will not change within the same sectors. At any moment, only the upper bridge of one phase can be conducted with the lower bridge of the other phase.

Figure 165: Two Side Modulated Block Commutated PWM

In this Block set it is possible to do high side modulation and a low side modulation. In high side modulation only the upper half of the H-bridges are provided with PWM duty cycle and the lower half duty cycles are fixed at 100%. In low side modulation PWM is given to the low side switches only (similar to high side modulation).

Figure 166:High Side Modulated Block Commutated PWM

Fourth and final possibility is to provide power switches on the high and low side are provided with the PWM signal alternately. This is called “symmetric modulation” and provokes a uniform load on all switches

.

Figure 167: Symmetric Modulation Block Commutated PWM

Content

The blockset contains the following elements:

- Processor Interface: BLOCK_COMMUTATED_PWM_GENERATOR_RCP_out (Processor Interface)
- FPGA Interface: BLOCK_COMMUTATED_PWM_GENERATOR_RCP_in (FPGA Interface)
- FPGA: BLOCK_COMMUTATED_PWM_GENERATOR_RCP (FPGA Main Component)
- FPGA Interface: BLOCK_COMMUTATED_PWM_GENERATOR_RCP_out (FPGA Interface)
- Processor Interface: BLOCK_COMMUTATED_PWM_GENERATOR_RCP_in (FPGA Interface)

Processor Output

Block

Figure 168: BLOCK_COMMUTATED_PWM_out block

Block Dialog

The processor output of the block-set contains the following dialog elements:

Figure 169: BLOCK_COMMUTATED_PWM_out Parameter dialog

The Parameter dialog of the Block Commutated blockset has the following parameters:

Name	Unit	Description	Range
PWM Aligning Mode	[-]	Specify the modulation and the update algorithm of the duty cycle data from the inputs 0: rising-edge modulation and update 1: pulse-center modulation and update 2: pause-center modulation and update 3: pulse-center modulation with an update of the duty cycles every pulse- and pause center of the pwm	0 … 3
Deadtime	[s]	Deadtime between the generated low and high side gate signal	0 … 52.429 µs resolution: 10e-9 s (FPGA Sample time: 10e-9 s)
Downsampling update trigger	[-]	If the duty cycle data is updated in the FPGA main block an update trigger flag is generated. The trigger event can be down sampled with this factor	1 … 1023
Sample time FPGA	[s]	Sample time of the FPGA	1…..

The next tab is the Sector definition dialog of the Block Commutated blockset and it has the following parameters

Figure 170. BLOCK_COMMUTATED_PWM_out Sector Definition dialog

Name	Unit	Description	Range
Number of sectors	[-]	Specify the Number of sectors for the Block commutated PWM	0 … 12
Sector N End	[deg]	Specifies the degrees at which the Nth sector ends	0 … 360 Deg

⚠	The sectors must not overlap i.e. Sector N end must be numerically less than Sector N+1 End. This would cause the PWM block to malfunction. Can also short circuit the inverter

Figure 171:BLOCK_COMMUTATED_PWM_out Switching Pattern Definition dialog

The next tab is the Switching pattern definition dialog of the Block Commutated blockset and it has the following parameters

Name	Unit	Description	Range
Leg A+ Sector 1 …… Leg C- Sector N	[-]	Specify the Switching pattern for a given phase in a given sector Low: No switching High: On throughout the sector PWM: Modulated with given duty Cycle.	Low, High, PWM

⚠	Duty Cycles for the same phase cannot be High or PWM at the same time. For example, A+ Sector 1: High and A- Sector 1: High or PWM. This configuration would Short circuit the Inverter.

Figure 172: BLOCK_COMMUTATED_PWM_out Position Source dialog

The final tab is the Position Source dialog of the Block Commutated blockset and it has the following parameters:

Name	Unit	Description	Range
Position Signal Source	[-]	Specify the position signal used for block commutation 0: From the Processor 1: Hall Sensor position signal directly from the FPGA 2: Incremental Encoder position signal directly from FPGA	0 … 2
Pole Pairs	[-]	Specifies the number of pole pairs of the BLDC motor	0 …255

Input

The processor output of the block-set has the following inputs:

Name	Unit	Description	Range
Enable	[-]	Enable of the output	0|1
PWM Frequency	[Hz]	PWM Frequency at disabled external trigger	0 … 5 MHz
Duty_Cycle	[-]	Duty cycle (scalar)	0 ……1
DIR	[-]		
Offset	[Deg]		
Enable_Ext_ Trigger	[-]	Enable external pulse center synchronization signal	0|1

Output

The processor output block provides 15 registers whose sectioning is shown below:

Register 1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Duty_Cycle																Position															

Figure 173: BLOCK_COMMUTATED_PWM_out Register 1

Name	Bits Used	Description	Range
Position	15…0	Position Signal Sent from the Processor 0 to 360 degree is scaled to an Unsigned value between 0 and 1.	0…1
Duty_Cycle	31…16	Duty cycle is a scalar between 0 and 1	0…1

Register 2 ... 5

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+1		Const Leg X Plus or Minus Sector N	

Figure 174: BLOCK_COMMUTATED_PWM_out Register 2 … 5

Name	Bits Used	Description	Range
Const Leg X Plus or Minus Sector N	1…0	Switching pattern for phase X and the sector N. For example, Register 2 starts with Phase A sector 1	0…2
Const Leg X Plus or Minus Sector N+…	3...2/5…4/ … /30…31	Switching pattern for phase X and the sector N+….	0…2

Register 6

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dir				Const_No_of_Sectors												Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+1		Const Leg X Plus or Minus Sector N	

Figure 175: BLOCK_COMMUTATED_PWM_out Register 6

Name	Bits Used	Description	Range
Const Leg X Plus or Minus Sector N	1…0/3…2/ …/15…14	Switching pattern for phase X and the sector N+….	0…2
Const_No_of_Sectors	16…30	Number of sectors per phase	0 … 2^10-1 represents 12 Sectors
Dir	31	Direction of rotation	0|1

Register 7…12

Figure 176: BLOCK_COMMUTATED_PWM_out Register 7 … 12

Name	Bits Used	Description	Range
Const_Sector_X_End	15…0	Position of the send of Sector X scaled to number between 0 and 1.	0…1
Const_Sector_X+1_End	31…16	Position of the send of Sector X+1 scaled to number between 0 and 1.	0…1

Register 13

Figure 177: BLOCK_COMMUTATED_PWM_out Register 7 … 12

Name	Bits Used	Description	Range
PWM Period Time	23…0	Period time of the PWM	0 … 2^24-1 represents 0 … 0.1678 s
SPARE	28…24		
Enable	29	Output Enable	0|1
Aligning Mode	31…30	PWM aligning mode	0…3

Register 14

Name	Bits Used	Description	Range

Deadtime	19…0	Deadtime of the high and low side signals	0 … 2^19-1 represents 0 … 52.429 µs
Enable ext Trigger	20	Enable external Trigger	0|1
Downsampling	30…21	Down sampling of the update trigger flag	1…1023
SPARE	31		

Figure 178: BLOCK_COMMUTATED_PWM_out Register 14

Register 15

Figure 179: BLOCK_COMMUTATED_PWM_out Register 15

Name	Bits Used	Description	Range
Const_Position_Source	1…0	Select the source of position signal.	0…2
Const_Pole_Pairs	31…16	Pole pairs of the machine being controlled	1…255

FPGA Main Component

Block

Figure 180: BLOCK_COMMUTATED_PWM Main block

Block Dialog

The FPGA main blockset contains the following dialog.

Figure 181: BLOCK_COMMUTATED_PWM Main block dialog

	The parameters of the input and output registers can be defined for automatic interface generation (When checkbox is enabled) on the page Interface. The dialog also contains an additional button to start interface generation. For more details about automatic interface generation, refer to Automatic Interface Generation chapter of XSG_Utils Documentation.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Ext_Trigger	[-]	External pulse center trigger of an external PWM.	Bool
Hall_Position	[-]	Position signal from Hall Sensor	UFix_16_16
Incremental Position	[-]	Position signal from Incremental Sensor	UFix_16_16
Incremental Valid	[-]	Position signal from Incremental Sensor is valid	Bool
Hall Direction	[-]	Direction from Hall sensor	Bool
Incremental Direction	[-]	Direction from Incremental Direction	Bool
Enable	[-]	Enable the Output	UFix_1_0
Enable_Ext_Trigger	[-]	Enable of the external trigger	UFix_1_0
PWM_Mode	[-]	PWM aligning mode	UFix_2_0
Period_time	[Raw]	Period time of the PWM at disable external trigger	UFix_24_0
Dead_Time	[Raw]	Deadtime of PWM	UFix_20_0
Downsampling_Up_Tr	[-]	Downsampling Update trigger flag	UFix_10_0
Duty_Cycle	[-]	Duty Cycle for the current PWM time period	UFix_16_16
A_Pos_Sector_Vector	[Raw]	Switching Pattern definition for phase A positive leg	UFix_24_0
B_Pos_Sector_Vector	[Raw]	Switching Pattern definition for phase B positive leg	UFix_24_0
C_Pos_Sector_Vector	[Raw]	Switching Pattern definition for phase C positive leg	UFix_24_0
A_Neg_Sector_Vector	[Raw]	Switching Pattern definition for phase A negative leg	UFix_24_0

B_Neg_Sector_Vector	[Raw]	Switching Pattern definition for phase B negative leg	UFix_24_0
C_Neg_Sector_Vector	[Raw]	Switching Pattern definition for phase C negative leg	UFix_24_0
Position_Signal_Proc	[-]	Position signal from processor	UFix_16_16
Sector_Select	[Raw]	Selects number of sectors for block commutation	UFix_15_0
Sector_1_End	[-]	End of sector 1	UFix_16_16
Sector_2_End	[-]	End of sector 2	UFix_16_16
Sector_3_End	[-]	End of sector 3	UFix_16_16
Sector_4_End	[-]	End of sector 4	UFix_16_16
Sector_5_End	[-]	End of sector 5	UFix_16_16
Sector_6_End	[-]	End of sector 6	UFix_16_16
Sector_7_End	[-]	End of sector 7	UFix_16_16
Sector_8_End	[-]	End of sector 8	UFix_16_16
Sector_9_End	[-]	End of sector 9	UFix_16_16
Sector_10_End	[-]	End of sector 10	UFix_16_16
Sector_11_End	[-]	End of sector 11	UFix_16_16
Position_Offset	[-]	Offset added to the position signal	UFix_16_16
DIR	[-]	Direction of rotation from the processor	Bool
Position_Source_Select	[-]	Select source of the position signal	UFix_2_0
Pole_Pairs	[-]	Number of pole pairs	UFix_10_0

Content

The blockset contains the following elements:

Name	Unit	Description	Format
FPGASignalBus	Bus	Bus containing the blocks most significant signals	-

PWM_HSD_A	[-]	High side of gate signal	Bool
PWM_LSD_A	[-]	Low side of gate signal	Bool
PWM_HSD_B	[-]	High side of gate signal	Bool
PWM_LSD_B	[-]	Low side of gate signal	Bool
PWM_HSD_C	[-]	High side of gate signal	Bool
PWM_LSD_C	[-]	Low side of gate signal	Bool
UpdateTrigger	[-]	Update trigger flag	Bool
En	[-]	Enable Output	Bool

Processor Input

Block

Adapts the FPGA signals for the processor side

Figure 182: BLOCK_COMMUTATED_PWM_in block

.

Block Dialog

The following parameters can be configured in the processor input dialog:

Figure 183: BLOCK_COMMUTATED_PWM_in dialog

The dialog blockset has the following parameters:

Name	Unit	Description	Range
Sample time FPGA	[s]	Sample time of the FPGA.	-

Input

The processor input block has the following inputs:

Register 1

Figure 184: BLOCK_COMMUTATED_PWM_in Register_1

Name	Bits Used	Description	Range
PWM Period Time	23…0	Period time of the PWM	0 … 2^24-1
Valid Output	25	Valid flag for the PWM output	0|1

Register 2:

Figure 185: BLOCK_COMMUTATED_PWM_in Register_2

Name	Bits Used	Description	Range
Sector	3 … 0	Current Sector number of the rotor position	0…12
Position	19 … 4	Current Rotor position	0 …. 1

Output

The processor input block has the following outputs:

Name	Units	Description	Range
PWM Period Time	[s]	Period time of the PWM	0 … 83,9 ms
Valid Output	[-]	Valid flag for the PWM output	0|1
Sector	[-]	Current Sector number of the rotor position	0…12
Position	[Deg]	Current Rotor position	0 …. 360

Block Modulator

Overview

The voltage to three phase brushless DC machine (BLDC) is supplied by a three-phase inverter. The input to the inverter is a DC voltage supplied by two wires one at higher DC potential than the other. Each phase of the inverter has two switches one which connects one motor phase to the higher DC voltage, referred to as high side driver (HSD), and other to the lower DC voltage, referred to as low side driver ‘(LSD).

The three-phase block modulator block generates three phase duty cycle gates signals (for all HSD and LSD) according to the desired voltage setpoint. The calculated duty cycles serve as inputs to the three phase PWM generator, which generates the gate control signals for the inverter switches (HSD & LSD) Input to the block modulator is the voltage setpoint that is to be generated by the control algorithm and the actual absolute value of the DC link voltage.

The block modulator interfaces with a standard XSG_Utils PWM generator via a ROUTE_PWM block (like shown below) for more information refer to the ROUTE_PWM_SIGNALS section in the Miscellaneous chapter.

Schematic

The block modulation can be interfaces with the XSG_Utils PWM generator with the help of the help of a ROUTE_PWM signals block. A schematic can be seen below.

Figure 186: Schematic Block modulation and PWM generation interface

For further information, and screen shot of an example implementation please see ROUTE_PWM_SIGNALS sub section in the Miscellaneous chapter.

Block Commutation

Block commutated PWM is used to supply a BLDC motor. One complete electrical rotation is divided into several sectors. In each sector a certain voltage pattern is defined to supply the connected machine. The current direction will not change within the same sectors. At any sector, only the high side driver of one phase must be conducted with a low side driver of another phase. Below 3 examples of different commutation technique are displayed:

Figure 187: Two Side Modulated Block Commutated PWM

In this block set it is possible to do high side modulation and a low side modulation. In high side modulation only the upper half of the H-bridges are provided with PWM duty cycle and the lower half duty cycles are fixed at 100%. In low side modulation the PWM low side switches generate PWM signals and the high side is fixed at a 100% (like high side modulation).

Figure 188: High Side Modulated Block Commutated PWM

Alternative possibility is to provide power switches on the high and low side are provided with the PWM signal alternately. This is called “symmetric modulation” and creates a uniform load on all switches.

Figure 189: Symmetric Modulation Block Commutated PWM

Content

The block set contains the following elements:

- Processor Interface: BLOCK_MODULATOR_out (Processor Interface)
  - FPGA Interface: BLOCK_MODULATOR _in (FPGA Interface)
  - FPGA: BLOCK_MODULATOR (FPGA Main Component)
  - FPGA Interface: BLOCK_MODULATOR _out (FPGA Interface)
  - Processor Interface: BLOCK_MODULATOR _in (Processor Interface)

Processor Output

Block

Figure 190: BLOCK_MODULATOR_out block

Block Dialog

The processor output of the block-set contains the following dialog elements:

Figure 191: BLOCK_MODULATION_out Mask

Name	Unit	Description	Range
Activate Input Port: Processor Position	[-]	Adds an inport for position input from the processor side and allows position signal to be sent to the FPGA from processor	0|1
Activate Input Port: Processor Duty Cycle	[deg]	Adds an inport for position input from the processor side	0|1
Invert rotation direction	[-]	Inverts the direction of rotation of the motor	0|1
Digital spike filter for sector transition	[s]	Wait time before change of sector	[0 … 6.4 µs] (depends on FPGA sample time)
Sample time of the FPGA	[s]	Step time of the FPGA	-

Note that when Activate Duty Cycle is enabled. The duty cycle used by the block modulator, to generate PWM signals, are provided by the processor application. Similarly, when activate Position signal is enabled then the position signal, used to calculate the current sector, is provided by the processor application.

Figure 192: BLOCK_MODULATOR_out Sector definition

The Sector definition dialog of the block and it has the following parameters:

Name	Unit	Description	Range
Number of sectors	[-]	Specify the Number of sectors for the Block commutated PWM	1 … 12
Sector N End	[deg]	Specifies the degrees at which the Nth sector ends	0 … 360 Deg

The Switching Pattern Definition tab of the Block Modulator block has the following parameters layout which allows to define a sector wise switching definition for all 6 switches (3 phase HSD and LSD). Open option translates to the switch being open for the entire time rotor is in the sector. Closed option means the switch is closed for the entire time rotor is in this sector and with PWM (duty cycle) option the user can generate PWM signal for the current sector.

Figure 193: BLOCK_MODULATOR_out Switching Pattern Definition dialog

There is also the possibility to visualize the switching pattern hence defined graphically using the Plot switching pattern button on the top. This generates a matlab figure with sector wise switching pattern definition as shown in figure below.

Figure 194. switching pattern definition graphic generated via the mask

Name	Unit	Description	Range
HSD A (+) Sector 1 …… LSD C (-) Sector N	[-]	Specify the Switching pattern for a given phase in each sector Open: No switching Close: ON throughout the sector PWM: Modulated with given duty Cycle.	Open, Close, PWM

⚠	High side and low side switches for the same phase cannot be Close or PWM at the same time. For example, HSD A(+) Sector 1: close and LSD A (-) Sector 1: Close or PWM. This configuration would short circuit the inverter.

Input

The processor output of the block-set has the following inputs:

Name	Unit	Description	Range
Offset_Pos_el	[deg]	Electrical offset angle with reference to Processor position or position obtained from FPGA	-
Proc_Postion_el	[deg]	Electrical position of the motor generated from the processor. If processor position is activated	-
Proc_Duty_Cycle	[-1…1]	Duty cycle (scalar)	-1 ……. 1

Output

The processor output block provides 15 registers whose sectioning is shown below:

Register 1

Figure 195: BLOCK_MODULATOR_out Register 1

Name	Bits Used	Description	Range

Proc_Postion_el	15…0	Position Signal Sent from the Processor 0 to 360 degree is scaled to an Unsigned value between 0 and 1.	0…1
Proc_Duty_Cycle	31…16	Duty cycle is a scalar between 0 and 1	0…1

Register 2…5

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+1		Const Leg X Plus or Minus Sector N	

Figure 196: BLOCK_MODULATOR_out Register 2 … 5

Name	Bits Used	Description	Range
Const Leg HSD or LSD Phase X Sector N	1…0/3...2/5…4/ … /30…31	Switching pattern for phase X and the sector N. For example, register 2 starts with Phase A sector 1	0…2

Register 6

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Invert_Direction	Const_Valid_Edges									Const_Proc_Duty_Cycle	Const_Proc_Position	Const_No_of_Sectors				Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+…		Const Leg X Plus or Minus Sector N+1		Const Leg X Plus or Minus Sector N	

Figure 197:BLOCK_MODULATOR_out Register 6

Name	Bits Used	Description	Range
Const Leg HSD or LSD Phase X Sector N	1…0/3…2/ …/15…14	Switching pattern for phase X and the sector N+….	0…2
Const_No_of_Sectors	16…19	Number of sectors per phase	1…12
Const_Proc_Position	20	Enable position from Processor otherwise use FPGA position	0|1
Const_Proc_Duty_Cycle	21	Enable Duty Cycle from Processor otherwise ue duty cycle calculated on FPGA	0|1
Const_Valid_Edges	22…30	Number of clock cycles a sector change has to be valid	0…512
Invert Direction	31	Chane direction of rotation of the motor	0|1

Register 7…12

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Const_Sector_X+1_End																Const_Sector_X_End															

Figure 198:BLOCK_MODULATOR_out Register 7 … 12

Name	Bits Used	Description	Range
Const_Sector_X_End	15…0	Position of the send of Sector X scaled to number between 0 and 1.	0…1
Const_Sector_X+1_End	31…16	Position of the send of Sector X+1 scaled to number between 0 and 1.	0…1

FPGA Main Component

Block

Figure 199: Figure 151:BLOCK_MODULATOR Main block Fix Point implementation (left) and Floating-Point implementation (right)

Block Dialog

The FPGA main block contains the following dialog.

Figure 200: BLOCK_MODULATOR Main block dialog

Input

The main block has the following inputs:

Name	Unit	Description	Format
Update	[0|1]	Update trigger for block for sample and hold of new duty cycle value or voltage set points	Bool
Reset	[0|1]	Reset trigger for the Block	Bool
V_Set	[V]	Voltage Setpoint for Block Modulation	Fix_25_x/ xFloat_8_24
Abs_V_DC	[V]	Absolute Value of DC link Voltage	Fix_25_x/ xFloat_8_24
Position_el	[0…1]	Electrical Position signal	UFix_16_16
Proc_Duty_Cycle	[-1…1]	Duty cycle set from processor	Fix_16_15
HSD_A_Sector_type		Switching Pattern definition for HSD A	-
LSD_A_Sector_type		Switching Pattern definition for LSD A	-
HSD_B_Sector_type		Switching Pattern definition for HSD B	-
LSD_B_Sector_type		Switching Pattern definition for LSD B	-
HSD_C_Sector_type		Switching Pattern definition for HSD C	-
LSD_C_Sector_type		Switching Pattern definition for LSD C	-
Proc_Position_el		Position signal from processor	UFix_16_16

Num_Sector	[3…12]	Selects number of sectors for block commutation	UFix_4_0
Sector_1_End	[0…1]	End of sector 1	UFix_16_16
Sector_2_End	[0…1]	End of sector 2	UFix_16_16
Sector_3_End	[0…1]	End of sector 3	UFix_16_16
Sector_4_End	[0…1]	End of sector 4	UFix_16_16
Sector_5_End	[0…1]	End of sector 5	UFix_16_16
Sector_6_End	[0…1]	End of sector 6	UFix_16_16
Sector_7_End	[0…1]	End of sector 7	UFix_16_16
Sector_8_End	[0…1]	End of sector 8	UFix_16_16
Sector_9_End	[0…1]	End of sector 9	UFix_16_16
Sector_10_End	[0…1]	End of sector 10	UFix_16_16
Sector_11_End	[0…1]	End of sector 11	UFix_16_16
Offset_Position_el	[0…1]	Offset added to the position signal	UFix_16_16
Activate_Proc_Position	[0|1]	Activate position signal from Processor	Bool
Activate_Proc_Duty_Cycle	[0|1]	Activate Duty Cycle from Processor	Bool
Invert_Direction	[0|1]	Invert Direction of Rotation	Bool
Valid_Edge	[Clocks]	Number of clocks wait for a sector transition	UFix_9_0

Output

The blockset contains the following elements:

Name	Unit	Description	Format
FPGASignalBus	Bus	Bus containing the blocks most significant signals	-
PWM_HSD_A	[0…1]	High side of gate signal	UFix_16_16
PWM_LSD_A	[0…1]	Low side of gate signal	UFix_16_16

PWM_HSD_B	[0…1]	High side of gate signal	UFix_16_16
PWM_LSD_B	[0…1]	Low side of gate signal	UFix_16_16
PWM_HSD_C	[0…1]	High side of gate signal	UFix_16_16
PWM_LSD_C	[0…1]	Low side of gate signal	UFix_16_16

Processor Input

Block

Adapts the FPGA signals for the processor side

Figure 201: BLOCK_MODULATOR_in block

Block Dialog

The following parameters can be configured in the processor input dialog:

Figure 202: BLOCK_MODULATOR_in dialog

Input

The processor input block has the following inputs:

Register 1

Figure 203: BLOCK_MODULATOR_in Register_1

Name	Bits Used	Description	Range
Sector	3 … 0	Current Sector number of the rotor position	1…12
Position	19 … 4	Current Rotor position	0 …. 360

Register 2…4

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Duty_Cycle_LSD_Phase_X																Duty_Cycle_HSD_Phase_X															

Figure 204: BLOCK_MODULATOR_in Register_2...4

Name	Bits Used	Description	Range
Duty_Cycle_HSD_Phase_X	15…0	Duty cycle set for the high side switch of Phase X	0 … 1
Duty_Cycle_LSD_Phase_X	31…16	Duty cycle set for the low side switch of Phase X	0 … 1

Register 5

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ref_Duty_Cycle																Abs_V_DC															

Figure 205: BLOCK_MODULATOR_in Register_5

Name	Bits Used	Description	Range
Ref_Duty_Cycle	15…0	Duty cycle set	-1 … 1
Abs_V_DC	31…16	Absolute Value of DC link Voltage	0 … 511.99

Output

The processor input block has the following outputs:

Name	Units	Description	Range

Sector	[-]	Current Sector number of the rotor position	0…12
Position	[Deg]	Current Rotor position	0 …. 360
Duty_Cycle_HSD[a;b;c][0_1]	[-]	Duty cycle set for the high side switch of all Phases	0…1
Duty_Cycle_LSD[a;b;c][0_1]	[-]	Duty cycle set for the low side switch of all Phases	0…1
Ref_Duty_Cycle	[-]	Duty cycle set or calculated	0…1
Abs_V_DC	[V]	Absolute value of DC link volatge	0…511.99

PMSM Simple

Overview

Permanent magnet synchronous machine (PMSM) is modelled inside the field orientated reference frame. The generated back EMF from the PMSM has sinusoidal shape, therefore that the machine is modelled in dq-coordinates. Motor specific parameters such as the permanent flux, resistances and inductances are available as mask parameters of the processor interface, which can also be tuned during runtime (ControlDesk).

PMSM Simple

A simple three phase permanent magnet machine can be represented by writing out the Kirchhoff's voltage equation for all three phases. The three-phase system of differential equation can be reducing to a system of two-phase liner differential equation using the DQ reference frame transformation. The DQ axis equations are shown below:

$$L_d \frac{di_d}{dt} = v_d - R_s i_d + L_q \omega_e i_q$$

$$L_q \frac{di_q}{dt} = v_d - R_s i_q - L_d \omega_e i_d - \psi_P \omega_e$$

Not that the system of differential equations is interdependent i.e. the d axis has terms containing the q axis current and vice versa. The torque produced by the machine is calculated by the following torque equation.

$$Trq = \frac{3}{2} p \, [\, \psi_P i_q + (L_d - L_q) i_d i_q]$$

L_d	Direct axis inductance
L_q	Quadrature axis inductance
R_s	Stator winding resistance
ψ_p	Rotor permanent flux
p	Number of pole pair
Trq	Torque generated by the motor

A simplified schematic diagram can be found in the following figure. The phase voltage equation is expressed in block diagram format the cross coupling block is used to calculate the cross coupling terms such as $L_d \omega_e i_d + \psi_P \omega_e$.

Figure 206: Block Diagram for PMSM SIMPLE

	Important to note that in order to perform the DQ axis transformation electrical rotor position information is needed. This can be calculated by considering the mechanical equation of rotation of motor which calculated the angles from the torque input to the motor.

Content

The block set contains the following elements:

- Processor Interface: PMSM_SIMPLE_out (Processor Interface)
- FPGA Interface: PMSM_SIMPLE_in (FPGA Interface)
- FPGA: PMSM_SIMPLE (FPGA Main Component)
- FPGA Interface: PMSM_SIMPLE_out (FPGA Interface)
- Processor Interface: PMSM_SIMPLE_in (Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA

Figure 207: PMSM_SIMPLE_out block

Block Dialog

The processor output block contains two tabs. All general parameters of the block can be configured in the page “General”.

Figure 208: PMSM_SIMPLE_out dialog; page 'General'

The page has the following parameters:

Name	Unit	Description	Range
Sample Time FPGA	[s]	Sample time of the FPGA	-

All specific motor parameters can be configured in motor parameters tab

Figure 209: PMSM_SIMPLE_out dialog; page 'Motor Parameters'

The page has the following configuration:

Name	Unit	Description	Range
Stator Inductance direct axis Lsd	[H]	Stator inductance in the direct axis	0 … 0.25 H; resolution 3.81E-6
Stator Inductance direct axis Lsq	[H]	Stator inductance in the quadrature axis	0 … 0.25 H; resolution 3.81E-6
Permanent magnet rotor flux	[Wb]	Magnetic flux of motor magnets	0 … 1 Wb; resolution: 15.2E-6
Stator resistance Rs	[Ohm]	Stator resistance	244E-6 … 16 Ohm; resolution: 244E-6
Number of pole pairs	[-]	Number of motor pole pairs	0 … 16383; resolution 1

Input

The processor input block has the following inputs:

Name	Unit	Description	Range
Reset_Motor	[-]	Resets the function of motor model on FPGA	0|1

Output

The processor output block provides 5 registers whose sectioning is shown below:

Register 1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1|Const_R_Stator																Const_Psi															

Figure 210: PMSM_SIMPLE_out Register 1

Name	Bits Used	Description	Range
Const_Psi	15…0	Permanent rotor flux	0…1
1|Const_R_Stator	31…16	Inverse of stator resistance (scalar).	0.0625…4098

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Const_L_q_Stator																Const_L_d_Stator															

Figure 211: PMSM_SIMPLE_out Register 2

Name	Bits Used	Description	Range
Const_Ld_Stator	15…0	Stator inductance in the direct axis	0…0.25
Const_Lq_Stator	31…16	Stator inductance in the quadrature axis	0…0.25

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Const_Ts_FPGA . R | Ld																															

Figure 212: PMSM_SIMPLE_out Register 3

Name	Bits Used	Description	Range
Const_Ts_FPGA .R | Ld	31…0	Time constant in the direct axis	0…1

Figure 213: PMSM_SIMPLE_out Register 4

Name	Bits Used	Description	Range
Const_Ts_FPGA .R | Lq	31…0	Time constant in the quadrature axis	0…1

Figure 214: PMSM_SIMPLE_out Register 5

Name	Bits Used	Description	Range
Const_PolePairs	13…0	Number of pole pairs of the machine	1…16383
Reset_MotorMDL	16	Resets the motor model on FPGA	0|1

FPGA Main Component

Block

Figure 215: PMSM_SIMPLE FPGA Main block

Block Dialog

The FPGA main block contains the following dialog.

Figure 216: PMSM_SIMPLE FPGA Main block dialog

Input

The main block has the following inputs:

Name	Unit	Description	Format
V[a;b;c]	[V]	Three phase voltage input	Fix_18_7
APU_Speed	[Raw]	Angular processing unit input speed	Fix_30_0
APU_Reset	[0|1]	Reset angular processing unit	Bool
APU_Start	[V]	Start angular processing unit	Bool
Psi	[Wb]	Permanent Flux of the machine	UFix_16_16
R_inv	[1|Ohms]	Reciprcal of stator phase resistance	Fix_16_4
Ld	[H]	Direct axis stator inductance	UFix_18_18
Lq	[H]	Quadrature axis stator inductance	UFix_18_18
Pole pairs	[-]	Number of pole pairs of the machine	UFix_14_0
Ts|(Ld|R)	[-]	Electrical motor time constant in direct axis with reference to the FPGA sample time	UFix_32_32
Ts|(Lq|R)	[-]	Electrical motor time constant in quadrature axis with reference to the FPGA sample time	UFix_32_32
Reset	[0|1]	Reset Flag for Motor Model	Bool

Output

The block set contains the following elements:

Name	Unit	Description	Format
FPGASignalBus	Bus	Bus containing the blocks most significant signals	-
I_a	[A]	Current in phase a	Fix_18_7
I_b	[A]	Current in phase b	Fix_18_7
I_c	[A]	Current in phase c	Fix_18_7

I_d_Stator	[A]	Current in phase along direct axis	Fix_18_7
I_q_Stator	[A]	Current in phase along quadrature axis	Fix_18_7
v_d_Stator	[V]	Voltage in direct axis	Fix_27_16
v_q_Stator	[V]	Voltage in quadrature axis	Fix_27_16
v_emf_a	[V]	Back EMF Voltage in phase a	Fix_18_7
v_emf_b	[V]	Back EMF Voltage in phase b	Fix_18_7
v_emf_c	[V]	Back EMF Voltage in phase c	Fix_18_7
Trq	[Nm]	Torque generated	Fix_25_14

Processor Input

Block

Adapts the FPGA signals for the processor side

Figure 217: PMSMS_SIMPLE_in block

Block Dialog

The following parameters can be configured in the processor input dialog:

Figure 218: PMSM_SIMPLE_in block dialog

Input

The processor input block has the following inputs:

Register 1

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
i_a																i_b															

Figure 219: PMSM_SIMPLE_in Register_1

Name	Bits Used	Description	Range
I_b	15…0	Current in phase B	-1024… 1023
I_a	31…16	Current in phase A	-1024… 1023

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
i_c																i_d_Stator															

Figure 220: PMSM_SIMPLE_in Register_2

Name	Bits Used	Description	Range
I_d_Stator	15…0	D- Axis stator Current	-1024… 1023
I_c	31…16	Current in phase C	-1024… 1023

MSB																															LSB
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
i_q_Stator																Trq															

Figure 221: PMSM_SIMPLE_in Register_3

Name	Bits Used	Description	Range
Trq	15…0	Torque Generated	-1024… 1023
I_q_Stator	31…16	Q- Axis stator Current	-1024… 1023

Output

The processor input block has the following outputs:

Name	Units	Description	Range
i[a;b;c]	[A]	Current in the stator phases	-1024_1023
I_d_Stator	[A]	Current in the direct axis of the stator	-1024_1023
I_q_Stator	[A]	Current in the quadrature axis of the stator	-1024_1023
Trq	[Nm]	Torque generated	-1024_1023

Miscellaneous

Objective

The XSG AC Motor Control Solution contains a few smaller function blocks which can be useful typical measurement tasks. The small applications do not provide a processor interface.

Debounce

Description / Overview

This block handles bouncing digital inputs by ignoring successive input edges for a specified time.

Block

Figure 222: DEBOUNCE block

Input

The DEBOUNCE block has the following inputs:

Name	Unit	Description	Format
Signal In	-	The digital input signal	Bool
Debounce Time	Clocks	The debounce time, during which all following edges will be ignored. The maximum range of the debounce time can be specified in the block dialog (default: 1023 FPGA clock cycles).	UFix_x_0

Output

The DEBOUNCE block has the following outputs:

Name	Unit	Description	Format
Deb_Signal	-	The debounced output signal	Bool

Sync Generation

Description / Overview

This block detects a pulse-mid synchronous latch trigger and generates a synchronization signal of variable length and delay from it. If the period is known, it generates a pause-mid synchronous signal as well.

Block

Figure 223: SYNC_GENERATION block

Input

The SYNC_GENERATION block has the following inputs:

Name	Unit	Description	Format
Latch_Trigger	-	The PWM pulse-mid synchronous latch trigger	Bool
Sync_Delay	Clocks	The delay of the output sync pulse. Can be varied during runtime	UFix_24_0
Sync_Width	Clocks	The length of the output sync pulse. Can be varied during runtime	UFix_24_0

Output

The SYNC_GENERATION block has the following outputs:

Name	Unit	Description	Format
Pulse_Mid_Sync	-	PWM pulse-mid synchronous signal	Bool
Pause_Mid_Sync	-	PWM pause-mid synchronous signal	Bool

ROUTE PWM SIGNALS

Description / Overview

This block merges the signals from XSG_Utils PWM input bus from the Processor with Space Vector blocks FPGASignalBus. It affords the opportunity to switch between FPGA based Space vector modulation and Processor based modulation

Block

Figure 224: ROUTE_PWM_SIGNAL block

Input

The ROUTE_PWM_SIGNAL block has the following inputs:

Name	Unit	Description	Format
En_MODULATOR	-	Enables the Modulator bus inputs	Bool
MODULATOR_BUS	-	Inport for bus from Modulator signal collector	-
PWM_INTERFACE_BUS	-	Inport for bus from PWM Interface	-

Output

The ROUTE_PWM_SIGNAL block has the following outputs:

Name	Unit	Description	Format
Out	[-]	Combined Bus signal of Modulator and PWM interface bus	-

Interface Example

The ROUTE_PWM_SIGNAL can be used with XSG_Utils_lib Three phase PWM generator as shown in figure below.

Figure 225:ROUTE_PWM_SIGNAL Interface Example

Processor based model parts

Overview

The XSG AC Motor Control Solution contains following processor-based model parts for full support of FPGA based rapid control prototyping for e-motors.

- Motor Inertia

The following section describes different functionalities in detail.

Motor Inertia

Objective

The model simulates the mechanic of a motor shaft.

The following image shows the Simulink block found in the ACMC interface library.

Figure 226: MOTOR_INERTIA block

The equation of the system can be written as

$$\omega_{Motor} = \frac{1}{J_{Motor} + J_{Load}} \int (Trq_{Motor} - Trq_{Load}) dt$$

Here;

ω_{Motor}	Motor speed
Trq_{Motor}	Torque generated by the motor
Trq_{Load}	Torque demanded by the load
J_{Motor}	Moment of inertia of the motor

J_{Load} Moment of inertia of the load attached to the motor shaft

Block Dialog

The following parameters can be configured in the processor input dialog:

Figure 227: MOTOR_INERTIA dialog

The dialog has the following parameters

Name	Unit	Description
Motor inertia	[kgm2]	Motor inertia
Damping coefficient	[Nm|(rad|s)]	Damping coefficient

Input

The MOTOR_INERTIA block has the following inputs:

Name	Unit	Description
Reset	[-]	Reset functionality
Torque_Motor	[Nm]	Motor torque
Torque_Load	[Nm]	Load torque
Inertia_Load	[kgm2]	Load inertia
n_Init_Engine	[rpm]	Initial engine speed
T_Sample	[s]	Sample time
Open_Loop_Mode	[-]	Open Loop speed as output switch
Open_Loop_Speed	[rpm]	Open loop speed

Output

The MOTOR_INERTIA block has the following outputs:

Name	Unit	Description
SignalBus	Bus	Bus containing the most significant signals of this block
Speed	[rpm]	Actual motor speed in rpm
Speed	[rad|s]	Actual motor speed in rad|s

Discontinued Blocks

Overview

The following sections provide information about the blocks that are discontinued and therefor taken out from the XSG_ACMC_lib (starting from release 20.2.0).

General Information

Certain blocks from XSG_ACMC_lib and XSG_ACMCInterface_lib have been made obsolete. These blocks have been marked with a '!' with a red background as shown in figure below.

Figure 228: Discontinued block

If a model contains any of the discontinued blocks, the function in XSG_ACMC_lib identifies it and breaks the library link and therefore the block will not be migrated in the future versions. A warning also appears on the 'Diagnostic viewer', when opening or updating a model containing any of the discontinued blocks as shown in figure below.

Figure 229: Warning about discontinued block in Diagnostic Viewer

Information on replacement block (if exists) is given in the description as shown below.

Figure 230: Replacement block (if any) is suggested in the description

The blocks that are discontinued and removed from XSG_Utils_lib (including interfaces) are listed below:

Name	Interface on FPGA & Processor
SSI_SPI_RCP	SSI_SPI_RCP _in SSI_SPI_RCP _out
BLOCK_COMMUTATED_PWM_GENERATOR_RCP	BLOCK_COMMUTATED_PWM _GENERATOR_RCP _in BLOCK_COMMUTATED_PWM _GENERATOR_RCP _out

Demos

Overview

Overview

The XSG ACMC Library comes with demonstration projects, to control a 3 phase PMSM in current and speed control mode. It consists of MATLAB Simulink Model (including parameterization), a ControlDesk project and in case of Scalexio a ConfigurationDesk project.

	The XSG ACMC Demos require the license for: XSG_ACMC_FGPA, XSG_ACMC_PROC, XSG_ACMC_RTV plus XSG_Utils_FPGA, XSG_Utils_PROC, XSG_Utils_RTV

The Demos are precompiled for following hardware configurations

Platform	Processor Model	FPGA Model
MicroLabBox	DS1202	DS1202 FPGA
MABX III	DS1403	DS1514 (7K325) and DS1553

In case that a different hardware configuration should be used e.g. DS1401 with DS1514 (7K325) and DS1552, it is necessary to rebuild the FPGA & processor model

Initial Installation

The Demo Model Setup is separated from XSG ACMC installation. To install the Demos necessary open the ACMC Library as shown in Figure below and double click the demos block.

Figure 231: Demos in the XSG ACMC library Figure

When the demo link is double clicked then the following pop up dialog appears.

Figure 232: Dialog when opening a demo for the first time

The demos will be installed inside the folder with the current version name under the path: My Documents\ dSPACE\ XSG_ACMC\. Hence, the demo folder installation path, shown in the demo installation dialog will adjust, according to the user's login name.

Folder Structure

The Demos are copied using the following folder structure:

Figure 233: Dialog for selecting a demo model

The pdf document, Startup_Guide, contains a brief start up guide that can be used to step through the various steps needed to get the demo up and running on the real time hardware. The simulation folder contains the Simulink model and the relevant scripts to parametrize the model. It also contains build results in case of the PHS demo. The Instrumentation folder contains the ControlDesk project as a back up and the Configuration folder contains the Configuration Desk project as back up for the Scalexio based Demo.

The selection between PHS and Scalexio system is provided by a pop up window immediately after the installation process is completed. This can be seen in the figure below

Figure 234: Dialog for selecting a demo model

After the model type has been chosen, MATLAB will change the working directory to the corresponding folder, run a startup script to initialize variables and finally open the Simulink model. The startup script can be modified by the user to change sample times or motor parameters. Run the script again to apply (overwrite the parameter structure inside the workspace) changes. The PHS bus demo is only available for rti1202 (MicroLabBox) environment. In order to open the demo Matlab needs to be configured for rti1202. If this is not the case, the following error message will appear.

Figure 235: Error Warning for wrong RTI Environment

The user can switch to the desired platform by entering the command “rti1202” inside the command window.

Reinstallation

After the initial demo installation is done, the user can choose to open the available demo or to reinstall and open the demo to receive a not modified version of the demo:

Figure 236:Reinstallation pop up Block

	Any changes, on the demo model and experiment, will be lost after reinstalling the Demos

Permanent Magnet Synchronous Machine Demo

Overview

The control of synchronous machines is achieved by sinusoidal phase currents which are generated using (sine commutation) from the control algorithm.

The PMSM motor control typically uses a field-oriented control algorithm. This means, the coordinate system of the control loop is rotated with the rotor position. This way, the sinusoidal set points for the currents transform into constant set points, which are easier to control.

It is implemented as a cascaded velocity controller with subordinate current controller as shown in the figure below. The torque inducing current component i_q and the field weakening current component i_d are controlled separately.

The commutation is performed using the generic PWM generator block of the XSG Utils library. The motor position is determined using the Hall Sensor and Incremental Encoder processing of the XSG ACMC library. Velocity is provided by using the incremental encoder processing. Currents are measured by connecting shunts to the FPGA board ADCs and latching new values at the pulse mid of the PWM. The controller is triggered synchronously to the PWM pulse mid by using the RTI FPGA Programming Blockset interrupts.

The Demo has possibility of switching between a simulated PMSM motor and a real PMSM motor. The simulated motor parametrizations masks are present under the MOTOR_SIM subsystem in the root of the Simulink application. Any inputs that can be given to a real motor can also be given to the simulated motor. The PMSM motor and inverter is simulated on the FPGA side with PMSM_Simple and Inverter_Simple blocks. The generated torque of the motor is the input of a simple mechanic model, located on the processor side, which calculates the actual speed of the rotor. This speed is the input of an Angular Position Unit (APU) which calculates the actual position on the FPGA side. A more detailed description is available in the following sections.

The Demo also has the possibility of switching between Processor and FPGA based current control.

The processor-based control is located inside the CONTROLLER_TASK, this is a triggered subsystem and is called at every PWM center. This structure affords the demo application the opportunity to switch between the four different modes, namely:

- Speed Control
- Current Control
- Open_Loop (V_abc)
- Open_Loop (V_dq)

Speed Control: In this mode the processor-based Speed controller receives the actual speed from the position processing and the speed setpoint from the environmental control. According to the difference the setpoints for the current controller id and iq (direct and quadrature axis current) are calculated.

Current Control: This mode takes the set point values from the environmental control, for example from ControlDesk during online simulation. There are two setpoint required for current control, direct axis current setpoint (i_d_Set[A]) and quadrature axis current setpoint (i_q_Set[A]).

Open Loop (V_abc): In this mode direct axis voltage (V_d_Set[v]) and quadrature axis voltage (V_q_Set[V]) along with open loop electrical frequency (OL_f_Set[Hz]) can be directly specified by the user. Out of theses settings ideal sinusoidal three phase voltages (V_a, V_b, V_c) are generated and applied to the space vector modulator. These **voltages are not aligned to the rotor position**. The space vector modulator block calculates the corresponding duty cycles for the 3 phase PWM generator stage. Its outputs are routed via digital outputs to the inverter stage to apply the calculated voltage to the device under test (e-motor).

Open Loop (V_dq): In this mode direct axis voltage (V_d_Set[v]) and quadrature axis voltage (V_q_Set[V]) can be specified. With the **position information coming from the position encoder**, corresponding phase voltages (V_a, V_b, V_c) are calculated. The space vector modulator block calculates the corresponding duty cycles for the PWM generator stage. Its outputs are routed via digital outputs to the inverter stage to apply the calculated voltage to the device under test (e-motor).

Figure 237:Principle of the PMSM controller

Structure

This section provides a general description of the Simulink model. The figure below shows the Simulink root model. The FPGA containing part is placed on the top left corner, inside here everything is modelled with the use of the Xilinx System Generator. This subsystem contains all the calculation which later run on the FPGA platform.

The processor application, here everything is modelled with standard Simulink blocks, is placed in the centre of the model. The CONTROLLER_TASK subsystem (top of the processor-based blocks) is an asynchronous called function which is aligned to the PWM generation. It contains the processor speed- and current-controller als well as interfaces to the FPGA part.

The communication between the FPGA and the processor application for a Scalexio based system is carried out using buffer interface (for PHS it is register interface). Like shown on the right side of the model. The input to the processor part (output of the FPGA part) is marked in yellow and the output of the processor part (input of the FPGA part) is marked in magenta. These subsystems contain Buffer 2 Register and Register 2 Buffer blocks from XSG Utils. For more information please refer to the XSG Utils Documentation.

Figure 238:PMSM Controller Demo Root

CONTROLLER_TASK

The controller task has a cascaded control architecture. The **SENSOR** subsystem collects all the sensor signals coming from the FPGA: phase currents, d&q currents, DC link voltage and position sensor signals from Hall and Incremental encoders. The communication between the FPGA and this asynchronous task is realised within the B1D2 (board = 1, TaskID =2) subsystems (yellow and Magenta blocks).

Figure 239:PMSM processor controller model

The **SPEED_CONTROL** subsystem contains a PI controller for the speed control loop. The PI controller takes the set points from the ENVIRONMENTAL CONTROL (e.g.via ControlDesk during online simulation) and generates a quadrature axis current set point, the direct axis current setpoint is fix set to zero. These currents are routed via a switch, to provide a pure current control functionality without speed controller, directly to the input of the current controller.

Either the current setpoints are given to the **CURRENT_CONTROL** subsystem from the speed controller (if activated) or from the environmental control. There are two current controllers implemented, one processor- and the second FPGA based. Both take the same setpoints and the feedback from the phase current sensors. The output of this system are two voltage set points (V_d &V_q). Via an open loop switch, these calculated values can be overwritten by user defined signals coming from environmental control. The DQ voltages are converted to three phase voltages (using inverse Clark park transformation), with help of the actual rotor position. These voltages are fed into the two-level space vector modulator block. Another open loop switch allows to apply a certain three phase voltage independently from the rotor position. Here a user defined (constant) voltage (d & q) is transformed with a user defined frequency into phase voltages (abc)

An automatic position encoder offset calibration functionality is part of this system. It is necessary that the position sensor is calibrated, since otherwise the control algorithm cannot perform the required transformation. In case the offset position of the motor is not calibrated then this function prevents the motor from being switched on. The position calibration applies predefined constant voltages to the motor winding. These voltages lead to a rotation of the rotor, when rotor stops rotating it is located inside the electrical zero-degree position of the motor. The corresponding angular position value from the position sensor is recorded and used in the algorithm as position sensor offset.

⚠	During offset calibration the shaft is rotating. Ensure safety during calibration process.
⚠	The motor shaft should not be loaded or blocked during position sensor calibration. The offset calibration is possible when the rotor can rotate freely.

The **space vector modulator** takes the three phase voltages and the DC link voltage and converts them three-phase duty cycles. This three-phase duty cycles are given to a PWM modulator block which generates the corresponding switching signal for the power electronic. These switching signals are wired to the inverter, so that the inverter outputs the desired voltages to the connected motor.

ACTUATOR subsystem: In case of processor based current control, the gate duty cycles are taken from the space vector modulator located on the processor side. These duty cycles are provided as input to the PWM generator. The PWM generator is a standard coming from XSG_Utils, it provides the parameters for dead time, aligning mode and frequency. For more details refer to the XSG Utils documentation. In case of the FPGA based current control is activated, only the frequency, dead time and aligning mode are taken from here. The duty cycle for the FPGA based current controller are directly calculated on the FPGA.

MOTOR_SIM

Inside this subsystem the different components, which are required for the simulation of an e-motor are shown in the figure below.

Figure 240: MOTOR_SIM processor model

Inside the **INVERTER_SIMPLE** the parameters for the inverter, like DC_Link voltage, are sent to the FPGA. the **PMSM_SIMPLE** subsystem the motor parameters (e.g. R & L) are sent to and the calculated motor currents and torque are received from the FPGA. The motor torque is an input to the motor inertia model (**Mechanic**). The output of the mechanic model is the rotor speed which is sent to the FPGA to calculate the actual rotor position inside the **APU** on the FPGA side.

ENVIRONMENTAL_CONTROL

The processor application generated from the Simulink model is interfaced with ControlDesk using the Environment Control subsystem. This subsystem contains those signals and parameters which mainly are displayed and tuned inside ControlDesk. Like setpoints, control modes and other parameters

The display subsystem is connected to various LEDs and warning messages on the ControlDesk experiment that monitor the state of the motor and reports the current operating condition.

The parameter subsystem houses the standard motor and inverter parameters next to other environmental information needed by the controller. It contains also the motor safety relevant parameters.

	The setpoint subsystem contains the setpoints for all controller and open loop modes. Next to static- also dynamic-setpoints (1D LUT) can be activated. .
LOAD_MACHINE	The demo setup comes with a magnetic brake where the load torque can be applied to the device under test via PWM signal. The strength is in this configuration a function of the PWM duty cycle.
CURRENT_CONTROLLER	Interfaces for FPGA based field orientated current controller. Parameters coming from the environmental control such as PI controller parameter, control mode and filter time constants are sent from here to the FPGA based control algorithm. Next to this, motor specific parameter like Ld, Lq, pole pairs, phase resistance, etc. are also transferred.
POSITION_SENSOR	Parameterization interface for the FPGA based position sensors Hall and Incremental Encoder.
SERVICES	Contains the XSG Utils multiscope block, which can visualize signals from the FPGA directly to Control Desk with a time resolution of up to the FPGA sampling rate. A drive Safety block which protects to the motor for over current and thermal overheating and a library version block that verifies the correctness of the usedXSG solutions versions provided by dSPACE. For more information on Multiscope please refer to theXSG_Utils documentation.
SCALE_IO	Parametrization interface for linear scaling of all analog to digital converter (ADC) and digital to analog converter (DAC) available on the hardware. Next to the scaling (gain, offset, min, max) PT1 filters for the analog inputs (ADC) can be configured.. These standard components are located inside the XSG_Utils_lib. Please refer to the XSG_Utils documentation for more information.
FPGA Model Overview	A view inside the FPGA subsystem shows the FPGA based implementation, inside here everything is modeled by using the Xilinx System Generator. An overview of the FPGA implementation is shown in the figure below.

Figure 241:PMSM Demo FPGA Model

The FPGA setup block is on the left side of the FPGA model. Here the specification of desired the hardware platform (framework) is made. The FPGA build actions block is used to perform automated actions after a successful FPGA Build, e.g. backup of the current model and its settings. The analyse model block provides numerous useful functionalities for ease of FPGA programming with the dSPACE XSG based solutions. For more information refer to XSG Utils documentation. The yellow and magenta coloured subsystems contain the buffer communication between the processor and FPGA application.

MOTOR_SIM

The gate signals, which are the outputs of the motor controller, are the inputs of the **INVERTER_SIMPLE** model, which calculates with the applied DC_Link voltage the line voltages which are applied to the inputs of the **PMSM_SIMPLE** motor model. According to its parameterization and the input voltages the actual currents and motor torque is calculated. The generated torque is an input to the motor inertia model (**Mechanic**), which is located on the processor side. The output of the mechanic model is the rotor speed which is used to calculate the actual rotor position inside the APU on the FPGA side.

Figure 242: MOTOR_SIM FPGA

When motor simulation is enabled the outputs of the MOTOR_SIM are routed, inside the IO subsystem, to the FPGA and Processor based current- and the processor-based speed-controller, instead the once coming from the real sensors.

IO

All hardware in- and outputs are inside this subsystem. The IO subsystem is used to configure sensors and read sensor output, as well as phase current and battery supply voltage. This subsystem also has Digital Out Blocks which can be configured for different applications such as PWM. In here, scale ADC and scale DAC as well as the PT1 filters are located.

POSITION_SENSOR_PROCESSING

Contains the hall and incremental encoder processing from the XSG_ACMC library. They receive the digital inputs from the IO subsystem and calculate the actual position which is used for the controller algorithms.

CURRENT_CONTROLLER

The measured currents, rotor position and DC link voltages are input to the current controller subsystem and the output are three phase voltage setpoints. Next to the standard closed loop behavior of the controller, it comes with two open loop modes which may be useful during the put into operation phase.

Figure 243: FPGA based Current Controller

The figure above shows the basic structure of the current controller. The current sensor inputs for all three phases are taken in along with the mechanical sensor position and transformed into d-q coordinate frame. This is further given to two PT1

filters which then serves as inputs to two different PI controller each for d and q axis currents. The set points for the same are obtained from the processor. After the PI controller there is voltage decoupling network with back EMF feedforward. The output of the decoupling network is the d and q axis voltage setpoints which are provided to an inverse Clark Park transformation block and the three phase voltages hence obtained are the sent an inputs to Space Vector modulation blocks which take in the three phase voltages along with DC Link voltage to generate the duty cycles for all three legs of the inverter. Finally, the duty cycles are converted to corresponding switching signals in the PWM generator block.

The functionality of the FPGA based current controller is same as the processor based current controller as it affords the user the opportunity to select between the four different control mode available. In case the **Speed control** mode is selected the current set, points generated by the processor-based speed controller are received from the processor application and are fed as set point to the PI controller on the FPGA side. In case pure **Current Control** mode is selected then the setpoints obtained from the ENVIRONMENTAL_CONTROL on the processor application provided to set points to the current PI controllers. In case **open loop mode (V_abc)** is selected the corresponding voltages for the three phase V_a, V_b and V_c are calculated and sent over to the current controller. The current controller takes these voltages and sends them over directly to space vector modulator for generation of duty cycles. If the **open loop mode (V_dq)** is selected then the setpoints V_d_Set[v] and V_q_Set[v] originating from the ENVIRONMENTAL_CONTROL subsystem are provided to the current controller on the FPGA. The current controller coverts these setpoints to three phase voltage setpoints using the inverse clark park transformation block and the sends the value to the space vector modulator block.

SPACE_VECTOR_MODULATOR

Space vector modulation (SVM) is an algorithm for the control of pulse width modulation (PWM). It is used for the creation of alternating current (AC) waveforms; most commonly to drive 3 phase AC powered motors at varying speeds. An inverter converts a DC supply, via a series of switches, to three output legs which could be connected to a three-phase motor.

These switches must be controlled so that at no time both switches of a leg are activated at the same time, because this would result a shorten DC sypply voltage and destroy the power amplifier. This requirement is met by the complementary operation of the switches within a leg. i.e. if A+ is on the A− is off and vice versa. This leads to eight possible switching vectors for the inverter, V0 through V7 with six active switching vectors and two zero vectors in the case of a three-phase inverter.

Vector	A+	B+	C+	A-	B-	C-	VAB	VBC	VCA
V0	OFF	OFF	OFF	ON	ON	ON	0	0	0
V1	ON	OFF	OFF	OFF	ON	ON	$+V_{dc}$	0	$-V_{dc}$
V2	ON	ON	OFF	OFF	OFF	ON	0	$+V_{dc}$	$-V_{dc}$
V3	OFF	ON	OFF	ON	OFF	ON	$-V_{dc}$	$+V_{dc}$	0
V4	OFF	ON	ON	ON	OFF	OFF	$-V_{dc}$	0	$+V_{dc}$
V5	OFF	OFF	ON	ON	ON	OFF	0	$-V_{dc}$	$+V_{dc}$
V6	ON	OFF	ON	OFF	ON	OFF	$+V_{dc}$	$-V_{dc}$	0
V7	ON	ON	ON	OFF	OFF	OFF	0	0	0

In the table above V_{dc} is the DC link voltage. The space vector modulator block takes in the three phase reference phase voltages and converts them to their respective duty cycles for each of the three legs of the inverter. These duty cycles are the inputs of a 3 phase PWM generator block.

THREE_PHASE_PWM_GENERATOR

Its inputs are the duty cycles that come either from the FPGA- or processor-based current controller. A switch, which can be activated on the processor side, routes either the signals coming from the processor- or the FPGA based controller to the generator.

DRIVE_SAFETY

The drive safety block which provides long term (and with this also thermal protection) and short term over current protection along with zero current protection for the motor. The inputs of the system are three phase currents. It observes the current waveform and disables the digital outputs responsible for switching the inverter if the allowable limits are exceeded.

LOAD_MACHINE

This subsystem contains a single phase PWM generator. The output of this system is fed to a magnetic brake which is built in the demonstration setup. The amount of braking load is determined by the voltage (duty cycle) supplied to the brake.

SERVICES

Contains utilities such as the FPGA MultiScope and version information block from XSG_Utils library. The FPGA multiscope can take up to sixteen signals from the FPGA at the FPGA clock rate and send the data to the processor application. For more details on the scope refer to the XSG Utils documentation. The version information block transmits the version information of the XSG solutions located on the FPGA side to the processor application.

Control Desk Environment

To open the prepared ControlDesk backup, first start ControlDesk. Then, select File - Open – Project + Experiment from Backup.

Figure 244: Open Control Desk Backup Project

Navigate to the demo installation directory at:

C:\Users\<UserName>\Documents\dSPACE\XSG_ACMC\<Version_Number>\Demos\<Platform>\Instrumentation

For Example:
C:\Users\Admin\Documents\dSPACE\XSG_ACMC\20.1\Demos\Scalexio\Instrumentation

After the backup is loaded, the following overview layout can be seen, as in figure below. Now, register the processor board and start the online calibration/measurement. The real-time application will be downloaded. The entire ControlDesk experiment comes with 6 different layouts: Overview, Controller, Motor Sim, Dynamic Setpoints, ADC and System Information.

Overview Layout

This layout gives an overview of the demonstration model functionality.

Figure 245: ControlDesk Overview

The components on this layout have links on them which when clicked takes the user to the corresponding layout. Primary function of this layout is to provide the user the opportunity to navigate easily across the different functionalities of the

Demo. According to the activated Setups (simulated or real e-motor; processor or FPGA based current control), his layout adapts itself automatically.

Controller Layout

The controller layout can be seen organized in three distinct rows. The top row has a standard plotter to the left and and XSG_Utils FPGA Scope instrument to the right. Via the setting instrument, the measurement can be configured, like e.g. the trigger condition or a pre- or post-trigger time. The XY-Plotter in the centre displays the signals recorded on the FPGA side, here signals with a time resolution of up to FPGA Clock Speed (e.g. 12.5ns) can be visualized! These instruments are used to visualize the speed, current, voltage, and position signals used in the demo.

Figure 246: Control Desk Controller Layout

The lower row has four distinct instruments. On the left-hand side are the control instruments with which several setpoints and control modes can be selected. In the centre there is a schematic diagram which displays the most relevant actual values for the speed and current controller. On the right end of the row we have the online encoder offset calibration instrument and the Drive safety instrument.

The drive safety instrument as seen in the figure below, it provides protection of the drive against over current respectively over temperature. It can protect in case of a sensor failure with a zero current detection along with long term and short term over current with i2 and i2Fi thresholds.

Figure 247:Drive Safety Instrument

Online offset calibration is needed to find the electrical zero position of the motor. By using this instrument, it is very easy to identify this position. After pushing the start button, a rotating electrical field is applied to the motor as long as the position sensor can detect a valid position (index pulse of the TTL) In the second step a constant voltage in the d axis is applied which lets the rotor move into the electrical zero position.

Figure 248:Position Offset Control Desk Instrument

	The offset calibration is not required (Calibration NA) when MOTOR_SIM is activated.

The final row on this layout has the PI controller parameters for speed & current control (FPGA & Processor). The controller parameters can be adjusted individually for the cascaded control architecture.

Figure 249: Controller Parameter Instrument

Motor Sim Layout

In this layout it is possible to paramterize the simulated motor, inverter and mechanic and its operation points. The simulated feedback signals are visualized inside several display and time plotters. The left part of the motor sim instrument can be used to parameterize the different components of the motor simulation for example the PM machine charecteristics such as stator resistance. The start stop

button on the very top of the simulation instrument starts the PMSM_SIMPLE component on FPGA.

	Controller gains and limits are individually adjustable for simulated and real e-motor.

Figure 250: ControlDesk Motor Sim Layout

Dynamic Setpoints Layout

Inside the dynamic setpoint layout. the user can specify repeating sequences of time depending setpoints. In keeping with the theme of the demo it is possible to select between the four different control modes and provide the setpoints using one dimensional look up (LUT) tables. Each setpoint LUT has 10 discrete values that repeats after a specified period length. The period length can be altered from the intput field on the top and the table content can be modified during runtime.

Figure 251:Dynamic Setpoint Layout

ADC Layout

The ADC Scaling layout allows the user to adjust the sensor scaling (within a linear region, inclusive saturation) and filter times for analog input channels. For more information refer to XSG_Utils documentation.

Figure 252: ADC-DAC Scaling Layout

System Information Layout

The system information layout shows the version information of different FPGA and processor based XSG libraries provided by dSPACE. There are two plotters that display the periodic - and interrupt - task turnaround time. After the real-time application is downloaded and the real-time process is executed, the XSG library version info will show the recognized versions numbers for the FPGA and Processor model. These numbers must be equal!

Figure 253: Tasks and Versions Layout

Configuration Desk Environment

In case of using an SCALEXIO based platform, like e.g. MABX III a ConfigurationDesk project comes with the demo. To open the prepared ConfigurationDesk backup project, first start ConfigurationDesk. Then, select File - Open – Project + Experiment from Backup.

Figure 254: Open Configuration Desk Backup Project

Navigate to the demo installation directory at:

C:\Users\<UserName>\Documents\dSPACE\XSG_ACMC\<Version_Number>\Demos\<Platform>\Configuration

For Example:
C:\Users\Admin\Documents\dSPACE\XSG_ACMC\20.1\Demos\Scalexio\Configuration

DS1553 Framework

Introduction

Overview

Together with the XSG AC Motor Control Solution, a FPGA framework for the DS1553 AC Motor Control I/O-Board in conjunction with the DS1514 FPGA base board is delivered.

	Detailed information about the DS1553 hardware properties can be found in the MABXII ACMC User Guide.

Installation

The DS1553 framework is automatically installed when the XSG ACMC Solution is installed. The frameworks are located in:

C:\Users\<UserName>\Documents\dSPACE\RTIFPGA\Frameworks\

To reinstall the framework for the currently active dSPACE release and the therefore belonging XSG ACMC solution, click on the “Inst. Framework” button on the top level of the XSG AC Motor library.

Figure 255: Button to install the DS1553 FPGA framework

A dialog prompt will appear, asking to confirm that the framework will be installed. After the framework has been installed, the framework will be available in the framework selection window of the FPGA setup block. If a framework folder already exists a reinstall window will appear which can be used to over write the existing installation.

Figure 256: Button to install the DS1553 FPGA framework

Figure 257: Dialog to select DS1514 with DS1553 as target platform

I/O Read Functions

Overview

The DS1553 provides the following I/O read functions:

- 1x Status In
- 8x Digital In
- 8x ADC
- 4x RS485 In
- 1x Resolver IC
- 1x FPGA Temperature

Status In

Block

This block provides the internal status of the hardware initialization to the model.

Figure 258: FPGA_IO_READ block (Status In)

Output

The main block has the following outputs:

Name	Unit	Description	Format
Init Done	-	Signals that the hardware is initialized	UFix_1_0

Digital In

Block

The 8 digital inputs provide a signal level of V_H=5V and are configurable between single-ended or differential input signal in groups.

Figure 259: FPGA_IO_READ block (Digital In)

Block Dialog

The block provides the following dialog:

Figure 260: FPGA_IO_READ dialog (Digital In)

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Input type	-	Set the digital input group to single ended or differential input. For the 1st group (1,2,3) this option is set in the block mask of Digital In 1, for the 2nd group (4,5,6) this option is set in the block mask of Digital In 4 and for the 3rd group (7,8) this option is set in the block mask of Digital In 7.	single ended | differential
Enable simulation port	-	Creates a stimulus input port for offline simulation of the I/O. The stimulus data has to be a Simulink input scaled in V.	on | off

Output

The main block has the following outputs:

Name	Unit	Description	Format
Data	-	The logical signal level at the digital input	UFix_1_0

For single-ended digital inputs, please connect the corresponding DigIn (-) pin to ground.

ADC

Block

The 8 differential ADCs provide a resolution of 14 bit, a sample rate of 10 MHz and a configurable voltage range.

Figure 261: FPGA_IO_READ block (ADC)

Block Dialog

The block provides the following dialog:

Figure 262: FPGA_IO_READ dialog (ADC)

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Input voltage range	V	Set the ADC input voltage range to ±5V, ±15V or ±30V.	±5V | ±15V | ±30V
Scaling	-	Select if the output is scaled in ADC LSBs (bit) or mV.	Bit | mV
Enable simulation port	-	Creates a stimulus input port for offline simulation of the I/O. The stimulus data has to be a Simulink input scaled in V.	on | off

Output

The main block has the following outputs:

Name	Unit	Description	Format
Data	Bit | mV	The signal level at the analog input scaled in ADC LSBs (-8192…+8191) or mV.	Fix_16_0

RS485 In

Block

RS485 ports are bidirectional digital differential drivers with a signal low level of -2.5V and a signal high level of +2.5V. The ports RS485 In and RS485 out share the same physical driver.

Figure 263: FPGA_IO_READ block (RS485 In)

Block Dialog

The block provides the following dialog:

Figure 264: FPGA_IO_READ dialog (RS485 In)

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Enable simulation port	-	Creates a stimulus input port for offline simulation of the I/O. The stimulus data has to be a Simulink input scaled in V.	on | off

Output

The main block has the following outputs:

Name	Unit	Description	Format
Data	-	The digital signal level at the RS485 input.	UFix_1_0

Resolver IC

Block

The DS1553 I/O board provides an integrated resolver processing IC. The IC provides position information of a resolver connected the FPGA model. It also provides warnings concerning signal quality. The update rate of the resolver IC is 500 kHz.

Figure 265: FPGA_IO_READ block (Resolver IC)

Block Dialog

The block provides the following dialog:

Figure 266: FPGA_IO_READ dialog (Resolver IC)

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Sine / cosine RMS voltage	V	Selection of the input voltage range of the sine and cosine signal.	1.5V | 3.5V | 5V
Excitation RMS voltage	V	Selection of the RMS voltage of the generated excitation.	3V | 7V | 10V
Excitation Frequency	Hz	Selection of the frequency for the generated excitation.	Range:2kHz..20kHz Resolution: 250Hz
Resolution	Bit	Resolution of the resolver processing. The resolution effects the maximum trackable velocity as followed: 10 bit: 150000 rpm 12 bit: 60000 rpm 14 bit: 30000 rpm 16 bit: 7500 rpm	10bit | 12bit | 14 bit | 16 bit
Enable simulation port	-	Creates a stimulus input port for offline simulation of the I/O. The stimulus data (position) has to be a Simulink input scaled in degrees.	on | off

Input

The main block has the following inputs:

Name	Unit	Description	Format
Cl.Warn.	-	Resets warnings raised by the resolver IC.	UFix_1_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
Position	-	The position output of the resolver IC. The position range (0…360°) is scaled to 0…1.	UFix_16_16
Warnings	-	The warnings raised by the resolver IC. The warnings are as followed: *Bit 0 – Input amplitude low:* The sine/cosine input amplitude is more than ~20% lower than specified. Once this warning occurs it can only be cleared manually. *Bit 1 – Input amplitude high:* The sine/cosine input amplitude is more than ~20% higher than specified. Once this warning occurred it can only be cleared manually. *Bit 2 – Loss of tracking:* The position could not be tracked. This warning will disappear if the position tracking could be successfully resynchronized. *Bit 3 – Velocity over-range:* The current velocity is higher than the resolver processing is able to track. This warning will disappear if the velocity is low enough again to be tracked. *Bit 4 – Phase lock error:* The phase of the sine/cosine input does not match the phase of the excitation output. Check if the resolver is plugged and correctly working. This warning will disappear once the sine/cosine phase matches the excitation phase again.	UFix_5_0
Data New	-	Flag indicating that the position data is updated.	UFix_1_0

FPGA Temperature

Block

The DS1514 base board contains an integrated FPGA temperature measurement. The raw value of the temperature measurement is provided to the user model.

Figure 267: FPGA_IO_READ block (FPGA Temperature)

Block Dialog

The block provides the following dialog:

Figure 268: FPGA_IO_READ dialog (FPGA Temperature)

The blockset dialog has the following parameters:

Enable simulation port	-	Creates a stimulus input port for offline simulation of the I/O. The stimulus data (temperature) has to be a Simulink input scaled in °C.	on | off

Output

The main block has the following outputs:

Name	Unit	Description	Format
RAW_Temp	-	The raw temperature sensor output. The scaling is 130 LSB/K.	UFix_16_0
High	-	Flag indicating that the FPGA temperature is >125°C.	UFix_1_0

I/O Write Functions

Overview

The DS1553 provides the following I/O write functions:

- 1x LED Out
- 24x Digital Out
- 2x DAC
- 4x RS485 Out

LED Out

Block

This block controls the status LED of the DS1514 FPGA board.

Figure 269: FPGA_IO_WRITE block (LED Out)

Input

The main block has the following Inputs:

Name	Unit	Description	Format
Data	-	Sets the color of the FPGA LED to orange (1) or green (0).	UFix_1_0

Digital Out

Block

The 24 digital outputs provide a signal level of V_H=5V.

Figure 270. FPGA_IO_WRITE block (Digital Out)

Block Dialog

The block provides the following dialog:

Figure 271: FPGA_IO_WRITE dialog (Digital Out)

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Enable simulation port	-	Creates a stimulus output port for offline simulation of the I/O. The stimulus data is a Simulink output scaled in V.	on | off

Input

The main block has the following inputs:

Name	Unit	Description	Format
Data	-	The logical signal level to set at the digital output.	UFix_1_0

DAC

Block

The 2 differential DACs (referenced to ground) provide a resolution of 12 bit, an update rate of 20MHz and voltage range of -20V…+20.

Figure 272: FPGA_IO_WRITE block (DAC)

Block Dialog

The block provides the following dialog:

Figure 273: FPGA_IO_WRITE dialog (DAC)

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Scaling	-	Select if the input is scaled in DAC LSBs (bit) or mV.	Bit | mV
Enable simulation port	-	Creates a stimulus output port for offline simulation of the I/O. The stimulus data is a Simulink output scaled in V.	on | off

Input

The main block has the following Inputs:

Name	Unit	Description	Format
Data	Bit | mV	The signal level set at the analog output scaled in DAC LSBs (-2048…+2047) or mV.	Fix_16_0

	The DAC (+) and DAC (-) pins are both referenced to ground. To change the voltage range to -10V…+10V connect to ground instead of DAC (-). The scaling in mV is not valid in this case, so it is recommended to use scaling in bit when applying the -10V...+10V range.

RS485 Out

Block

RS485 ports are bidirectional digital differential drivers with a signal low level of -2.5V and a signal high level of +2.5V. The ports RS485 In and RS485 out share the same physical driver.

Figure 274: FPGA_IO_WRITE block (RS485 Out)

Block Dialog

The block provides the following dialog:

Figure 275: FPGA_IO_WRITE dialog (RS485 Out)

The blockset dialog has the following parameters:

Name	Unit	Description	Range
Enable simulation port	-	Creates a stimulus output port for offline simulation of the I/O. The stimulus data is a Simulink output scaled in V.	on | off

Input

The main block has the following inputs:

Name	Unit	Description	Format
Data	-	The digital signal level set at the RS485 output.	UFix_1_0
Enable	-	Sets the direction of the RS485 driver to output (1) or input (0).	UFix_1_0

Pin-Out

Layout

The pin-out of ZIF connector on the DS1553 ACMC I/O module is shown in the table on the next page. The layout of the ZIF connector (upper connector of the MicroAutoBox) is illustrated in the figure below. The location of ground pins can be seen in this figure as well.

Figure 276: Layout of the DS1553 ZIF connector

Group	Pin	Signal	Dir.	Group	Pin	Signal	Dir.
Digital Input	D3	DigIn 1 (+)	in	Analog Input	A6	ADC 1 (+)	in
	E6	DigIn 1 (-)	in		A2	ADC 1 (-)	in
	D4	DigIn 2 (+)	in		A5	ADC 2 (+)	in
	D5	DigIn 2 (-)	in		A4	ADC 2 (-)	in
	E5	DigIn 3 (+)	in		A3	ADC 3 (+)	in
	E4	DigIn 3 (-)	in		B6	ADC 3 (-)	in
	U3	DigIn 4 (+)	in		B5	ADC 4 (+)	in
	U5	DigIn 4 (-)	in		B4	ADC 4 (-)	in
	T3	DigIn 5 (+)	in		c2	ADC 5 (+)	in
	U4	DigIn 5 (-)	in		c3	ADC 5 (-)	in
	V6	DigIn 6 (+)	in		b2	ADC 6 (+)	in
	V5	DigIn 6 (-)	in		c4	ADC 6 (-)	in
	T5	DigIn 7 (+)	in		a2	ADC 7 (+)	in
	U6	DigIn 7 (-)	in		Z2	ADC 7 (-)	in
	F6	DigIn 8 (+)	in		c5	ADC 8 (+)	in
	E3	DigIn 8 (-)	in		Y2	ADC 8 (-)	in
Digital Output	R3	DigOut 1	out	Resolver IC	a4	Excitation (+)	out
	R5	DigOut 2	out		b6	Excitation (-)	out
	S5	DigOut 3	out		a5	Sine (+)	in
	R4	DigOut 4	out		a6	Sine (-)	in
	R6	DigOut 5	out		c6	Cosine (+)	in
	S4	DigOut 6	out		b5	Cosine (-)	in
	H3	DigOut 7	out	RS485	W3	RS485 1 (+)	in/out
	H4	DigOut 8	out		X4	RS485 1 (-)	in/out
	G3	DigOut 9	out		V4	RS485 2 (+)	in/out
	G4	DigOut 10	out		V3	RS485 2 (-)	in/out
	S3	DigOut 11	out		W6	RS485 3 (+)	in/out
	F5	DigOut 12	out		W4	RS485 3 (-)	in/out
	F3	DigOut 13	out		X3	RS485 4 (+)	in/out
	T4	DigOut 14	out		Y4	RS485 4 (-)	in/out
	F4	DigOut 15	out	Analog Output	C3	DAC 1 (+)	out
	G6	DigOut 16	out		D6	DAC 1 (-)	out
	X6	DigOut 17 *	out		C5	DAC 2 (+)	out
	C6	DigOut 18 *	out		C4	DAC 2 (-)	out
	H6	DigOut 19 *	out	Sensor Supply	E2	Supply +5V	out
	X5	DigOut 20 *	out		F2	Supply +5V	out
	G5	DigOut 21 *	out		C2	Supply +12V	out
	H5	DigOut 22 *	out		D2	Supply +12V	out
	S6	DigOut 23 *	out				
	W5	DigOut 24 *	out				

The pins marked with an asterisk (*) are only available if the DS1514 base board hardware is reconfigured. Please inquire about hardware reconfiguration if these digital outputs are required. The use of additional IP modules will not be possible after modification!

XSG_UTILS.docx