XSG SPI I2C Solution

User Guide

Version 20.1.0 – 22-Jun-2020

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)
  - On the dSPACE Website (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.

Contents

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 Documentation

PDF File

After installing your dSPACE software, the documentation for the installed products is available as Adobe® PDF file. Keep in mind, that you have to decrypt the solution via the dSPACE Installation Manager first.

After decryption you can access the PDF file via the following approaches:

- Documentation root: <INSTALLDIR>\Doc\ XSG_SPI_I2C.pdf
- Help buttons of every Simulink library block

Related Documents

dSPACE Help

Below is a list of documents that are recommended to read when working with the XSG SPI I2C Solution:

- 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

Requirements

Important information

The following tools have to be installed for using a free programmable dSPACE FPGA Board:

- MATLAB & Simulink
- Xilinx Vivado including Xilinx System Generator (XSG)
- dSPACE RCPHIL Release

Below you find the compatibility matrix for dSPACE Release, Xilinx and MATLAB for the XSG SPI/I2C Programming Blockset for FPGA:

RTI FPGA Programming Blockset	dSPACE Release	Operating System	MATLAB	Xilinx Design Tools
3.9	2020-A (64 bit)	Windows 7 (64 bit) Windows 10 (64 bit)	R2018b R2019a R2019b (64 bit)	Vivado 2019.2 (64 bit)

In the following table you find the compatibility matrix for dSPACE Release and MATLAB for the XSG SPI/I2C Blockset for Processor:

dSPACE Release	Operating System	MATLAB
2020-A (64 bit)	Windows 7 (64 bit) Windows 10 (64 bit)	R2018b R2019a R2019b R2020a (64 bit)

Please note that the Solution is optimized for and only supports the following dSPACE FPGA platforms:

- DS2655 FPGA Base Board (Kintex 7K160, 7K410)
- DS5203 FPGA Board (Kintex 7K325, 7K410)
- DS6601 FPGA Base Board (Kintex Ultrascale KU035)
- DS6602 FPGA Base Board (Kintex Ultrascale+ KU15P)
- DS1202 MicroLabBox (Kintex 7K325)
- DS1403 MicroAutoBox III (DS1514 + DS1552)

XSG SPI I2C Solution

Library Contents

Description / Overview

The XSG SPI I2C Library in its current state is used for simulation of SPI and I^2C masters and slaves based on Xilinx Kintex-7 FPGAs.

The general physical structure is defined inside the FPGA using Xilinx System Generator in combination with RTI FPGA. The communication parameters can be adjusted in the processor application.

Important Remarks

Bus delay

Due to protection circuits there are different delays (signal runtime from the generation on the FPGA to the real digital output and vice versa) on the digital I/O boards. For low SPI clock frequencies, they are negligible. For higher clock frequencies (approximately above 4 MHz) the bus delay parameter can be set inside the mask of the SPI Slave Sequence and the SPI Master Sequence FPGA blocks.

I/O input/output frequency

Please consider the maximum input/output frequencies for the different I/O channel. For the MicroAutoBox III it is strongly recommended to use the Digital InOut 6 channel type. It provides the fastest frequency.

Preceding library

The XSG SPI I2C Solution is based on the SCALEXIO SPI Master/Slave Library. In order to use the XSG SPI I2C Library in a clean state and avoid any complications please uninstall the SCALEXIO SPI Master/Slave Library from your system.

The XSG SPI I2C Solution in its current state has an extended functional range compared to the SCALEXIO SPI Master/Slave Library V14.1p2.

FPGA clock

Some configuration parameters concerning the timing (e. g. the serial clock period) depend on the internal FPGA clock. Which means the adjustable periods are automatically multiples of the FPGA clock. The FPGA clocks of the used boards can be determined via the dSPACE Help. Some examples can be found below mentioned:

FPGA Base Board	Clock Period
DS2655 FPGA Base Board	8 ns
DS5203 FPGA Board	10 ns

Overview

Objective

The following section describes the general structure and features of the XSG SPI I2C Solution.

Library Structure

General information

There are two components of the XSG SPI I2C Solution, depending on the licenses available:

- Interface (processor) part
- FPGA part

Interface Part

This library only contains the processor (CN) interface of the XSG SPI I2C Solution. It can be used if a fixed FPGA configuration is used or the FPGA configuration has already been implemented in another environment. If required dSPACE can implement an FPGA application according to your requirements, which is fully controllable via the processor interface.
To open the basic library, type “xsg_spi_i2c_if_lib” in the MATLAB Command Window or access the library via the Simulink library browser.

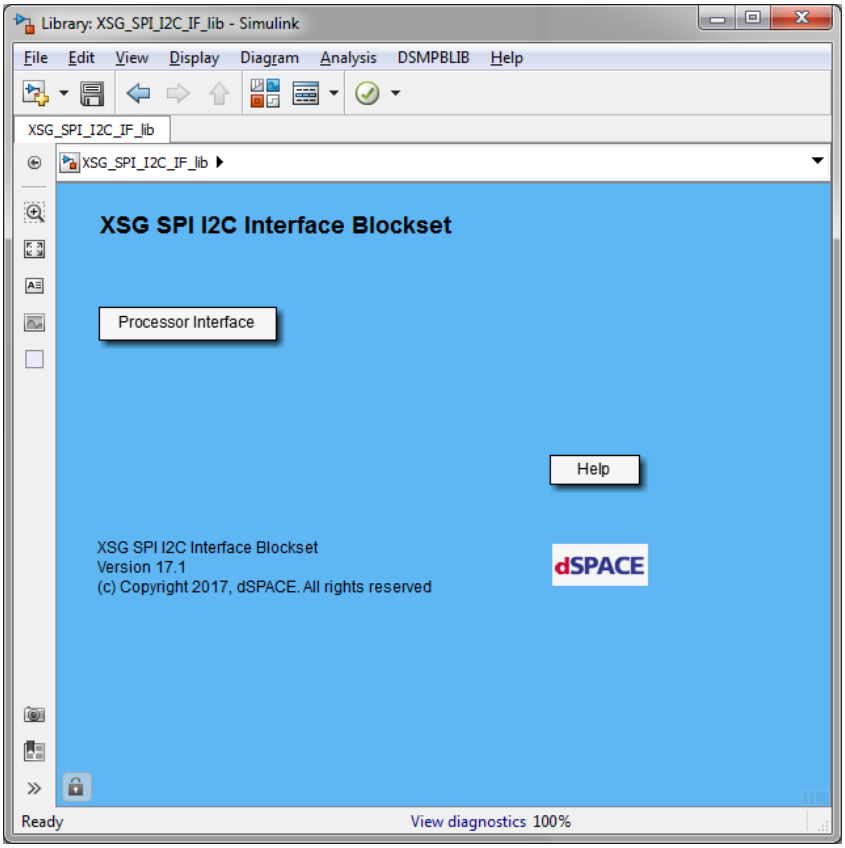

Figure 1: XSG SPI I2C Interface Blockset

FPGA Part

This library contains the FPGA part of the XSG SPI I2C Solution. It is required if a special FPGA configuration is desired. In this case, the FPGA configuration can be set up by the user on his own. A link to the processor (CN) interface is located in this library.
To open the FPGA part, type “xsg_spi_i2c_lib” in the MATLAB Command Window or access the library via the Simulink library browser.

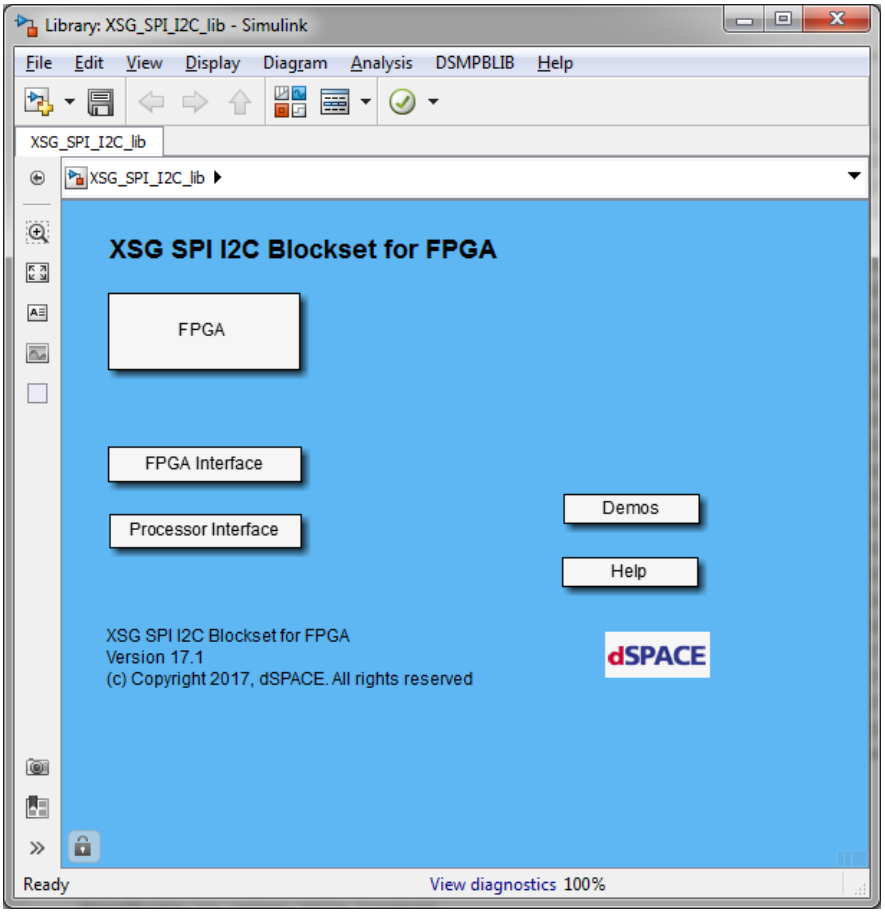

Figure 2: XSG SPI I2C FPGA Blockset

Workflow

Objective

The following section describes the purpose of each library block type and the way to use them.

Processor Parameterization

The Processor Parameterization blocks are used to set parameters of the single components. Some parameterization blocks have to be called at least 4 times before parameterization of the corresponding FPGA components are complete (time-driven multiplexing). Afterwards they can be disabled if desired. Do NOT place this type of blocks in an interrupt-driven task!
Via the register outputs some configuration data of the corresponding FPGA components can be read. The register outputs have to be connected to data out

ports from the DSMPBLIB if the target platform is a dSPACE SCALEXIO system.

Processor Output

The Processor Output blocks are used to write process data to the FPGA. The blocks can be placed in an interrupt-driven task if desired.
The register outputs have to be connected to data out ports from the DSMPBLIB if the target platform is a dSPACE SCALEXIO system.

Processor Input

The Processor Input blocks are used to read process data from the FPGA. The blocks can be placed in an interrupt-driven task if desired.
The register inputs have to be connected to data in ports from the DSMPBLIB if the target platform is a dSPACE SCALEXIO system.

Auto-Generate

Nearly all block masks of the FPGA main components have a tab called *Interface*. Using this tab, an automatic generation of all blocks required for a component can be started. The register numbers to be used can be set and a name differing from the default name can be specified.

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!

SPI Master Setup

Objective

The SPI Master Setup block is used to configure the SPI cycle settings of an SPI master. The SPI cycle can contain several sequences, which are defined in dedicated blocks.

Content

The blockset contains the following elements:

- Processor Interface: SPI_MASTER_SETUP_par (Processor Interface)
- FPGA Interface: SPI_MASTER_SETUP_in (FPGA Interface)
- FPGA: SPI_MASTER_SETUP (FPGA Main Component)

Processor Parameterization

Block

Merges the processor signals and writes them to the FPGA.

Figure 3: SPI_MASTER_SETUP_par block

Block Dialog

The processor parameterization block provides the following dialog:

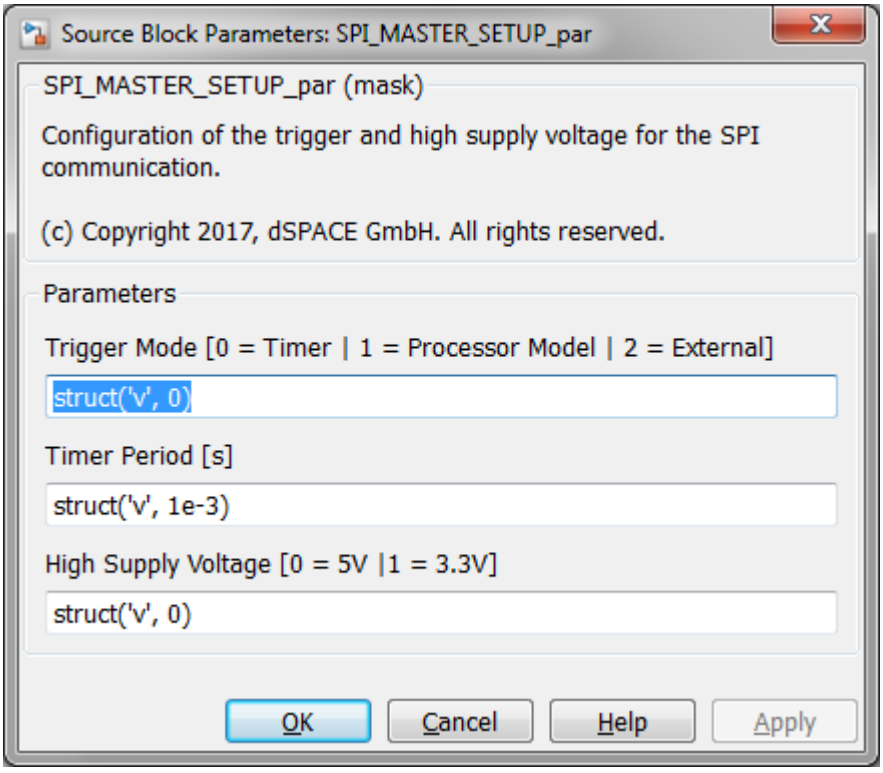

Figure 4: SPI_MASTER_SETUP_par dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Trigger Mode	-	Using this parameter, it can be selected on which event an SPI cycle is triggered: 0 = internal timer 1 = call of this block in the processor model 2 = rising edge at an external input	0|1|2
Timer Period	s	The timer period determine when the SPI cycle is triggered, if the timer trigger mode is selected.	Range: 0…2^{27}-1 * FPGA clock Resolution: FPGA clock
High Supply Voltage	-	Defines the supply voltage for the high level of the output signals.	0|1

Output

The Processor Parameterization block outputs one register. The sectioning is shown below:

Register 1

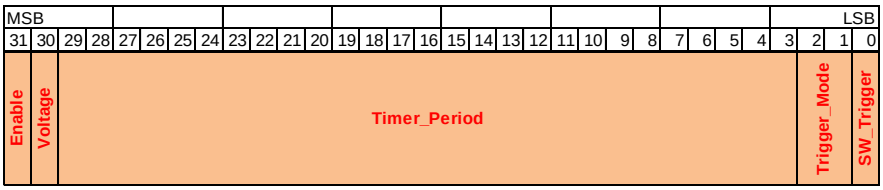

Figure 5: SPI_MASTER_SETUP_par Register 1

Name	Bits	Description
SW_Trigger	0	The cycle trigger from the processor model.
Trigger_Mode	2..1	The trigger mode selected by the processor output block.
Timer_Period	29..3	The period for the timer trigger.
Voltage	30	The high supply voltage.
Enable	31	Enables the trigger generation based on the internal timer.

FPGA Main Component

Block

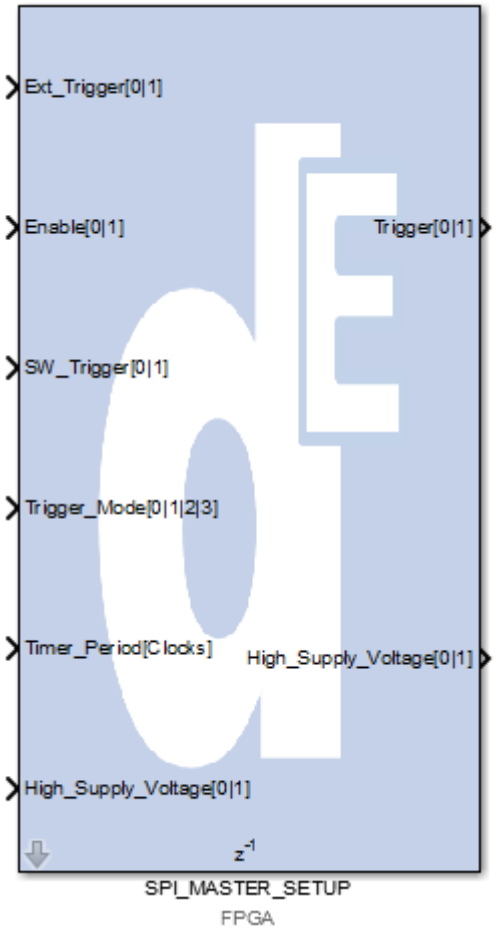

Figure 6: SPI_MASTER_SETUP FPGA Main Component

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Ext_Trigger	-	An external trigger to start an SPI cycle if the trigger mode is set accordingly.	Bool
Enable	-	The enable input for the timer.	Bool
SW_Trigger	-	Trigger from the processor model to start an SPI cycle if the trigger mode is set accordingly.	Bool
Trigger_Mode	-	The trigger mode as described in the processor output section.	UFix_2_0
Timer_Period	Clocks	The period of the timer to start an SPI cycle if the trigger mode is set accordingly.	UFix_27_0
High_Supply_Voltage	-	Defines the supply voltage for the high level of the output signals.	Bool

Output

The main block has the following outputs:

Name	Unit	Description	Format
Trigger	-	The generated trigger to start the SPI cycle.	Bool
High_Supply_Voltage	-	The high supply voltage can be forwarded to the I/O channels.	Bool

SPI Master Sequence

Objective

The SPI cycle generated by an SPI master can contain up to 32 sequences. Each sequence is configurable in terms of timing parameters, addressed chip selects and number of data bits. In each sequence, up to 32 data bits can be sent and received. For higher word width, multiple sequences can be connected.

Content

The blockset contains the following elements:

- Processor Interface: SPI_MASTER_SEQUENCE_par (Processor Interface)
- Processor Interface: SPI_MASTER_SEQUENCE_out (Processor Interface)
- FPGA Interface: SPI_MASTER_SEQUENCE_in (FPGA Interface)
- FPGA: SPI_MASTER_SEQUENCE (FPGA Main Component)
- FPGA Interface: SPI_MASTER_SEQUENCE_out (FPGA Interface)
- Processor Interface: SPI_MASTER_SEQUENCE_in (Processor Interface)

Processor Parameterization

Block

Merges the processor signals and writes them to the FPGA.

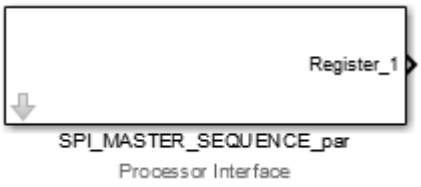

Figure 7: SPI_MASTER_SEQUENCE_par block

Block Dialog

The processor output block provides the following dialog:

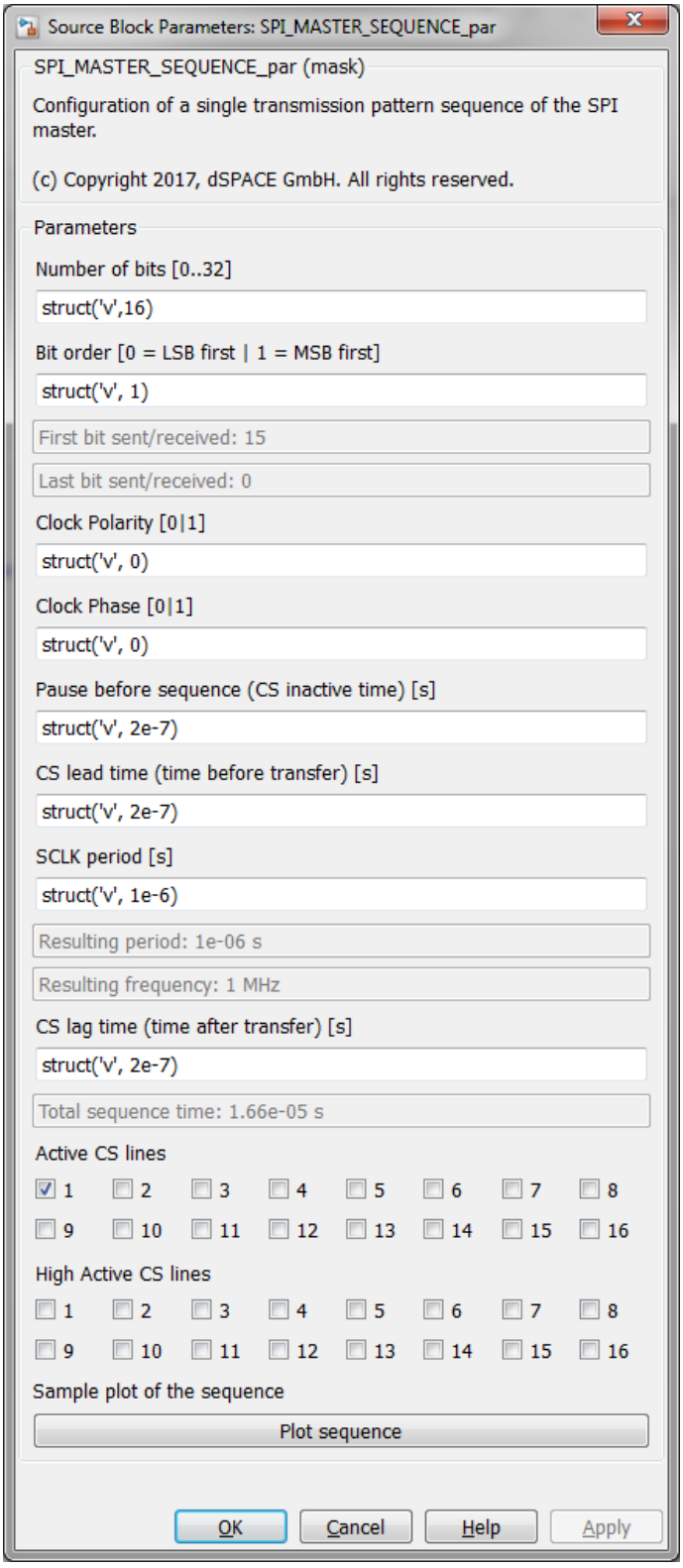

Figure 8: SPI_MASTER_SEQUENCE_par dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Number of bits	-	The number of bits to be sent / received within this sequence.	Range: 0…32 Resolution: 1
Bit order	-	Defines if the first bit to be sent / received is the LSB or MSB of the data word. 0 = LSB first 1 = MSB first	0|1
Clock polarity	-	The polarity (idle state) of the SCLK line (CPOL).	0|1
Clock phase	-	The SPI clock phase (CPHA).	0|1
CS lead time	s	The time the chip select line is active at the start of the cycle before data transmission starts.	Range: 0…2^{16}-1 * FPGA clock Resolution: FPGA clock
SCLK period	s	The period of the serial clock.	Range: 0…2^{16}-1 * FPGA clock Resolution: FPGA clock
CS lag time	s	The time the chip select line remains active after the transfer is finished.	Range: 0…2^{16}-1 * FPGA clock Resolution: FPGA clock
Pause before sequence	s	The time the CS line is inactive (low or high, depends on the logic level of the CS) between the preceding and the start of this sequence. If the pause time is set to 0, the chip select line will not be inactive between the two sequences.	Range: 0…2^{19}-1 * FPGA clock Resolution: FPGA clock
Active CS lines	-	The chip select lines addressed in this sequence. Only chip select lines addressed will be set active during data transfer.	0|1 (16x)
High Active CS lines	-	The chip select lines, which should be high active instead of low active.	0|1 (16x)

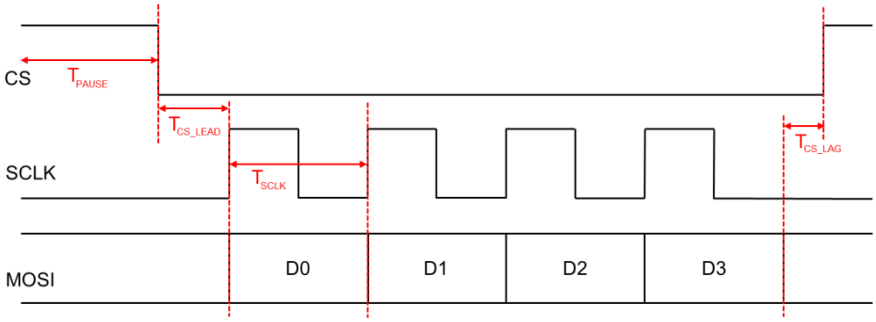

Figure 9: Timing parameters

The block mask also provides the function button *Plot Sequence*. When pressing this button, the sequence will be calculated according to the user settings and plotted in a MATLAB window.

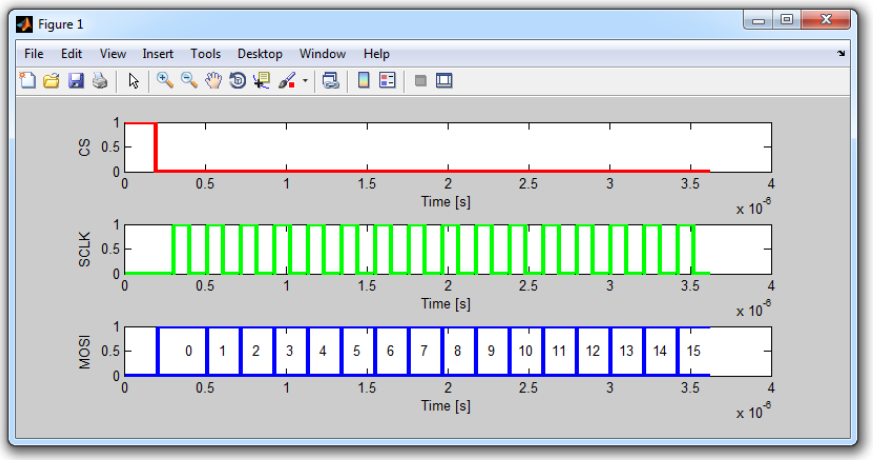

Figure 10: Sequence plot

Output

The Processor Parameterization block outputs 4 register contents, mapped to 1 register by time multiplexing. The sectioning is shown below:

Register 1.1

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		Enable	CPHA	CPOL	Bit_Order		High_Active_CS_1_9									CS_Lead_Time															

Figure 11: SPI_MASTER_SEQUENCE_par Register 1.1

Name	Bits	Description
CS_Lead_Time	15..0	The time before data transfer (chip select active)
High_Active_CS_1_9	16..24	Flags if the chip select signals 1 - 9 are high active
Bit_Order	26	Select between LSB first (0) or MSB first (1)
CPOL	27	The clock polarity
CPHA	28	The clock phase
Enable	29	The enable signal

Register 1.2

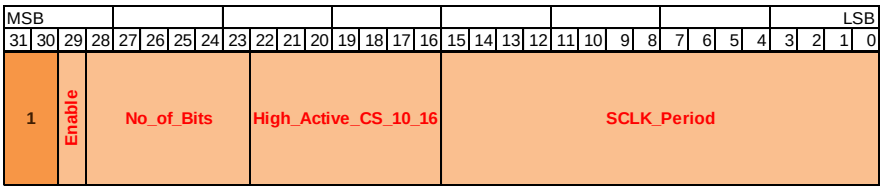

Figure 12: SPI_MASTER_SEQUENCE_par Register 1.2

Name	Bits	Description
SCLK_Period	15..0	The period of the serial clock
High_Active_CS_10_16	22..16	Flags if the chip select signals 10 - 16 are high active
No_of_Bits	28..23	The number of bits to be sent / received
Enable	29	The enable signal

Register 1.3

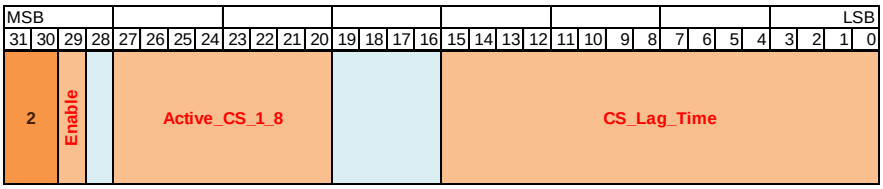

Figure 13: SPI_MASTER_SEQUENCE_par Register 1.3

Name	Bits	Description
CS_Lag_Time	15..0	The time after data transfer (chip select active)
Active_CS_1_8	27..20	Flags if the chip select signals 1 - 8 are addressed in this sequence
Enable	29	The enable signal

Register 1.4

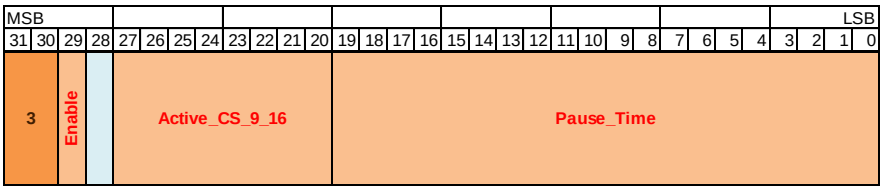

Figure 14: SPI_MASTER_SEQUENCE_par Register 1.4

Name	Bits	Description
Pause_Time	19..0	The time the CS line is inactive between the preceding and the start of this sequence
Active_CS_9_16	27..20	Flags if the chip select signals 9 - 16 are addressed in this sequence
Enable	29	The enable signal

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

Figure 15: SPI_MASTER_SEQUENCE_out block

Input

The SPI_MASTER_SEQUENCE_out block has the following inputs:

Name	Unit	Description	Range
TX_Data	-	The data to be sent via MOSI output.	Range: 0…2^{32}-1 Resolution: 1

Output

The Processor Output block outputs 1 register content, mapped to 1 register. The sectioning is shown below:

Register 2

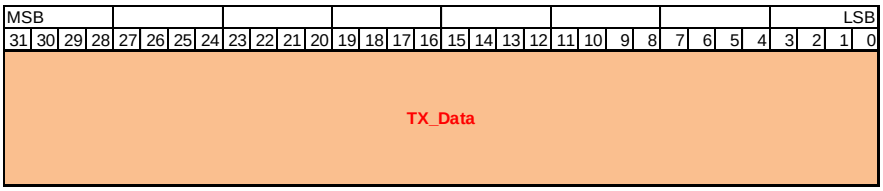

Figure 16: SPI_MASTER_SEQUENCE_out Register 2

Name	Bits	Description
TX Data	31..0	The data to be transmitted within this sequence

FPGA Main Component

Block

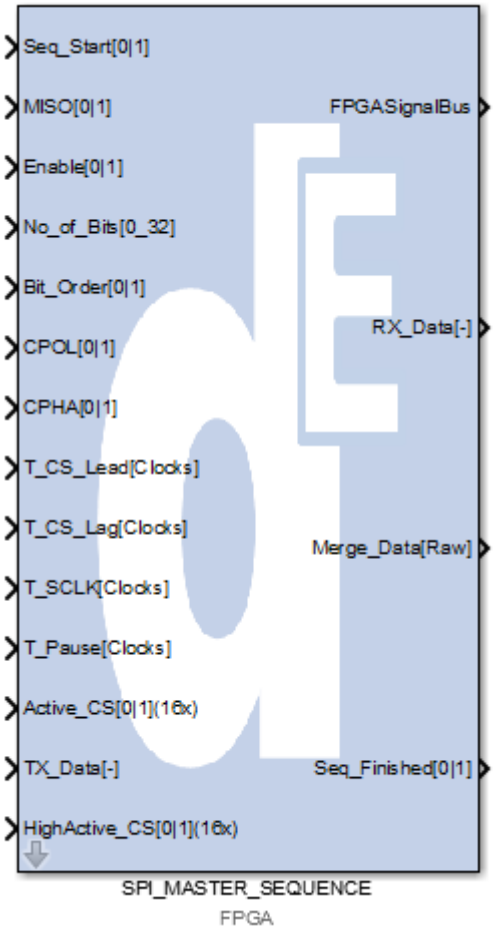

Figure 17: SPI_MASTER_SEQUENCE FPGA Main Component

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Seq_Start	-	Flag for starting the sequence. If it is the first sequence in the SPI cycle, connect it to the *Trigger* output of the master setup block. Otherwise connect it to the *Seq_Finished* output of the prior sequence.	Bool
MISO	-	The master-in-slave-out serial data input	Bool
Enable	-	Enables the sequence	Bool
No_of_Bits	-	The number of bits to be sent / received within this sequence.	UFix_6_0
Bit_Order	-	0 = LSB first 1 = MSB first	Bool
CPOL	-	The clock polarity	Bool
CPHA	-	The clock phase	Bool
T_CS_Lead	Clocks	The time before data transfer (chip select active)	UFix_16_0
T_SCLK	Clocks	The period of the serial clock	UFix_16_0
T_CS_Lag	Clocks	The time after data transfer (chip select active)	UFix_16_0
T_Pause	Clocks	The time the CS line is inactive between the preceding and the start of this sequence. If the pause time is 0, the chip select will not be inactive between the SPI sequences.	UFix_20_0
Active_CS	-	The chip select lines addressed in this sequence. Only chip select lines addressed will be set low during data transfer	Bool (16x)
HighActive_CS	-	The chip select lines, which should be high active	Bool (16x)
TX_Data	-	The data to be sent via MOSI output	UFix_32_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals bundled as bus interface.	Bus
RX Data	-	The data received via MISO input.	UFix_32_0
Merge_Data	Raw	Data port which has to be connected to the SPI Master Merge block.	UFix_19_0
Seq_Finished	-	Flag indicating that the sequence is finished. Can be used to trigger the next sequence.	Bool

Processor Input

Block

Adapts the FPGA signals for the processor side.

Figure 18: SPI_MASTER_SEQUENCE_in block

Input

The processor input block has the following inputs:

Register 1

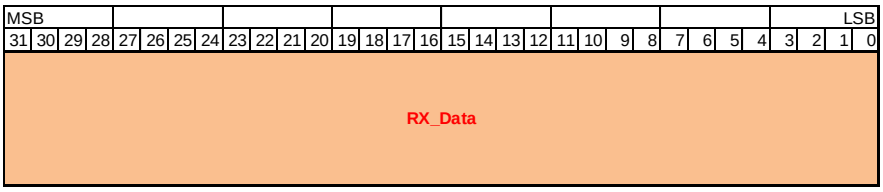

Figure 19: SPI_MASTER_SEQUENCE_in Register 1

Name	Bits	Description
RX_Data	31..0	The data received via MISO input

Output

The SPI_MASTER_SEQUENCE_in block has the following outputs:

Name	Unit	Description	Range
RX_Data	-	The data received via MISO input.	Range: 0…2^{32}-1 Resolution: 1

SPI Master Merge

Objective	The function of the Master Merge block is to take the generated output signals of all sequences and to merge them to one physical interface.
Content	The blockset contains the following elements: ▪ FPGA: SPI_MASTER_SEQUENCE (FPGA Main Component)

FPGA Main Component

Block

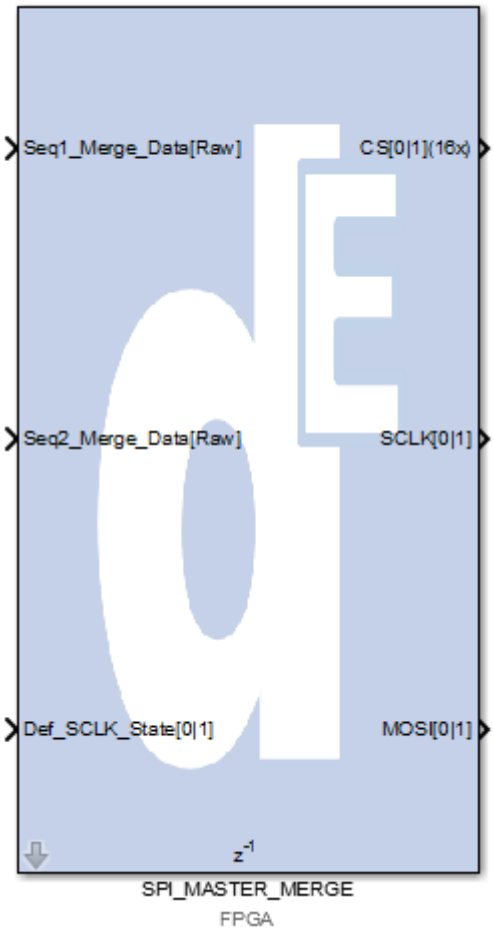

Figure 20: SPI_MASTER_MERGE FPGA Main Component

Block Dialog

The SPI_MASTER_MERGE block provides the following dialog:

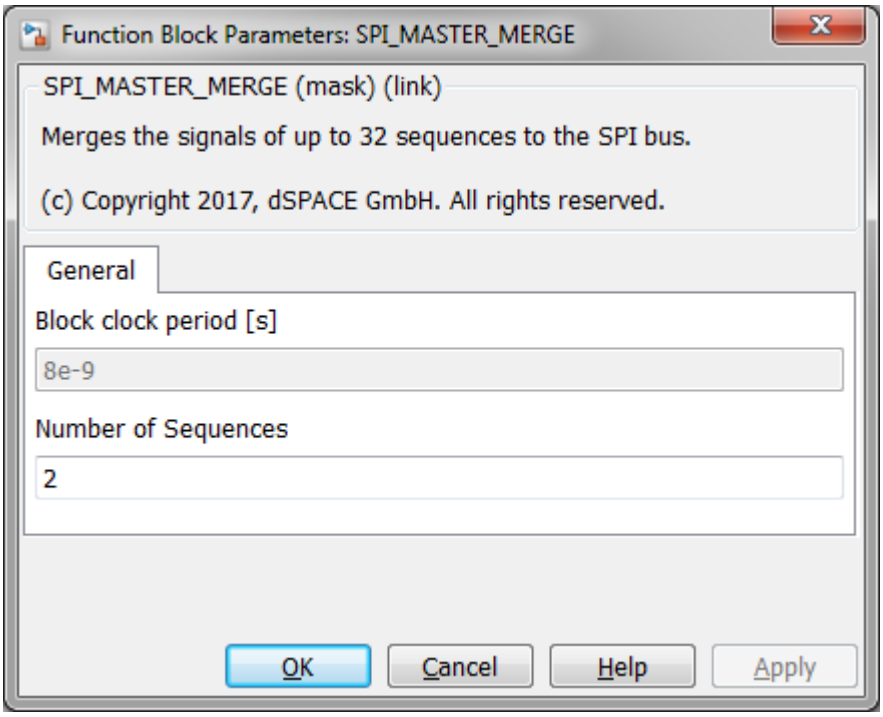

Figure 21: SPI_MASTER_MERGE dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Number of Sequences	-	The number of sequences to be merged.	Range: 1…32 Resolution: 1

Input

The main block has the following inputs:

Name	Unit	Description	Format
SeqX_Merge_Data	Raw	The merge data output of the SPI Master Sequence blocks.	UFix_19_0
Def_SCLK_State	-	The default signal level of the SCLK output when no sequence is active. This input can be connected to the CPOL signal of the SPI Master Sequence block.	Bool

Output

The main block has the following outputs:

Name	Unit	Description	Format
CS	-	The chip select output. To be connected to a digital out block.	Bool (16x)
SCLK	-	The serial clock output. To be connected to a digital out block.	Bool
MOSI	.	The master-out-slave-in. To be connected to a digital out block.	Bool

SPI Slave Sequence

Objective

For SPI slave applications, several sequences can be defined as well. Each sequence is configurable in terms of timing parameters and number of data bits. In each sequence, up to 32 data bits can be sent and received. For higher word width, multiple sequences can be connected. Splitting up SPI data transmission into multiple sequences enables for example in-frame-response behavior.

Content

The blockset contains the following elements:

- Processor Interface: SPI_SLAVE_SEQUENCE_par (Processor Interface)
- Processor Interface: SPI_SLAVE_SEQUENCE_out (Processor Interface)
- FPGA Interface: SPI_SLAVE_SEQUENCE_in (FPGA Interface)
- FPGA: SPI_SLAVE_SEQUENCE (FPGA Main Component)
- FPGA Interface: SPI_SLAVE_SEQUENCE_out (FPGA Interface)
- Processor Interface: SPI_SLAVE_SEQUENCE_in (Processor Interface)

Processor Parameterization

Block

Merges the processor signals and writes them to the FPGA.

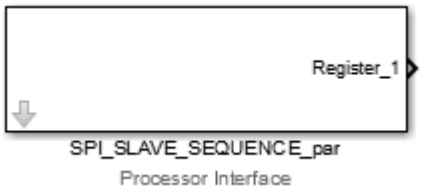

Figure 22: SPI_SLAVE_SEQUENCE_par block

Block Dialog

The processor parameterization block provides the following dialog:

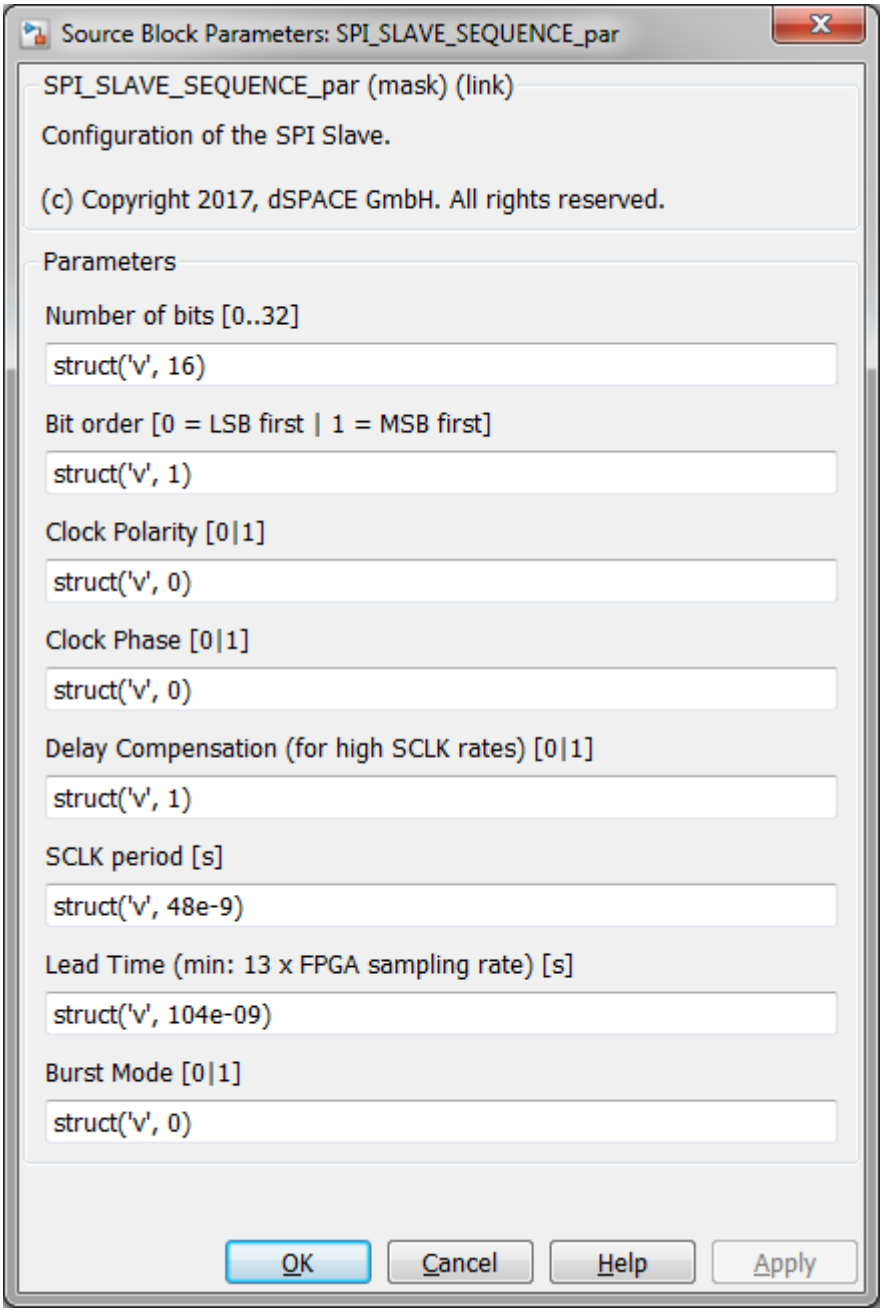

Figure 23: SPI_SLAVE_SEQUENCE_par dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Number of bits	-	The number of bits to be sent / received within this sequence.	Range: 0…32 Resolution: 1
Bit order	-	Defines if the first bit to be sent / received is the LSB or MSB of the data word. 0 = LSB first 1 = MSB first	0|1
Clock polarity	-	The polarity (idle state) of the SCLK line (CPOL).	0|1
Clock phase	-	The SPI clock phase (CPHA).	0|1
Delay Compensation	-	Activates hardware delay compensation. Using the DS2655M2, compensation is recommended for serial clock frequencies >4MHz.	0|1
SCLK period	s	The expected period of the serial clock (only required for delay compensation).	Range: 0…2^{16}-1 * FPGA clock Resolution: FPGA clock
Lead Time	s	The expected active time of the chip select signal before data transmission starts (only required for delay compensation). The minimum are 13 x the FPGA sampling rate (clock).	Range: 0…2^{16}-1 * FPGA clock Resolution: FPGA clock
Burst Mode	-	If activated the sequence will end after the specified number of bits. If more sequences with activated burst mode are chained together they can be processed without a chip select inactive time between them. If an inactive time of the chip select signal occurs during data transmission, the slave will stop the sequence.	0|1

Output

The Processor Parameterization block outputs 2 register contents, mapped to 1 register by time multiplexing. The sectioning is shown below:

Register 1.1

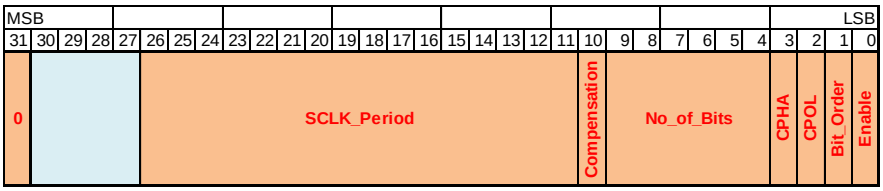

Figure 24: SPI_SLAVE_SEQUENCE_par Register 1.1

Name	Bits	Description
Enable	0	The enable signal
Bit_Order	1	Select between LSB first (0) or MSB first (1)
CPOL	2	The clock polarity
CPHA	3	The clock phase
No_of_Bits	9..4	The maximum number of bits sent / received within this sequence
Compensation	10	Activates delay compensation
SCLK_Period	26..11	The expected serial clock period

Register 1.2

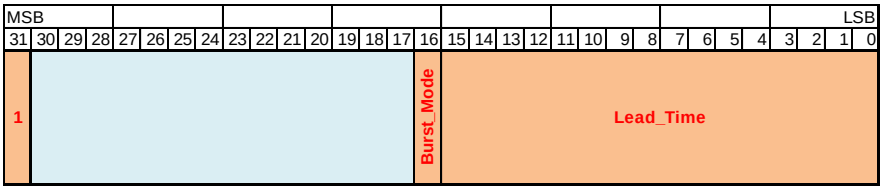

Figure 25: SPI_SLAVE_SEQUENCE_par Register 1.2

Name	Bits	Description
Lead_Time	15..0	The lead time (chip select active time before start of data transmission) necessary for correct delay compensation
Burst_Mode	16	The burst mode for processing multiple sequences without chip select inactive time between them

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

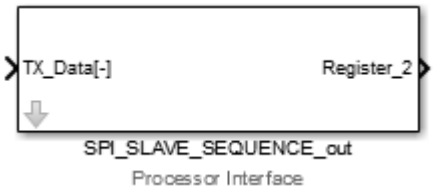

Figure 26: SPI_SLAVE_SEQUENCE_out block

Input

The SPI_SLAVE_SEQUENCE_out block has the following inputs:

Name	Unit	Description	Range
TX_Data	-	The data to be sent via MISO output.	Range: 0…2^{32}-1 Resolution: 1

Output

The Processor Out block outputs 1 register. The sectioning is shown below:

Register 1

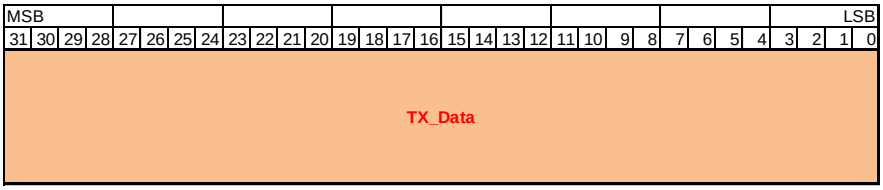

Figure 27: SPI_SLAVE_SEQUENCE_out Register 1

Name	Bits	Description
TX_Data	31..0	The data to be transmitted within this sequence

FPGA Main Component

Block

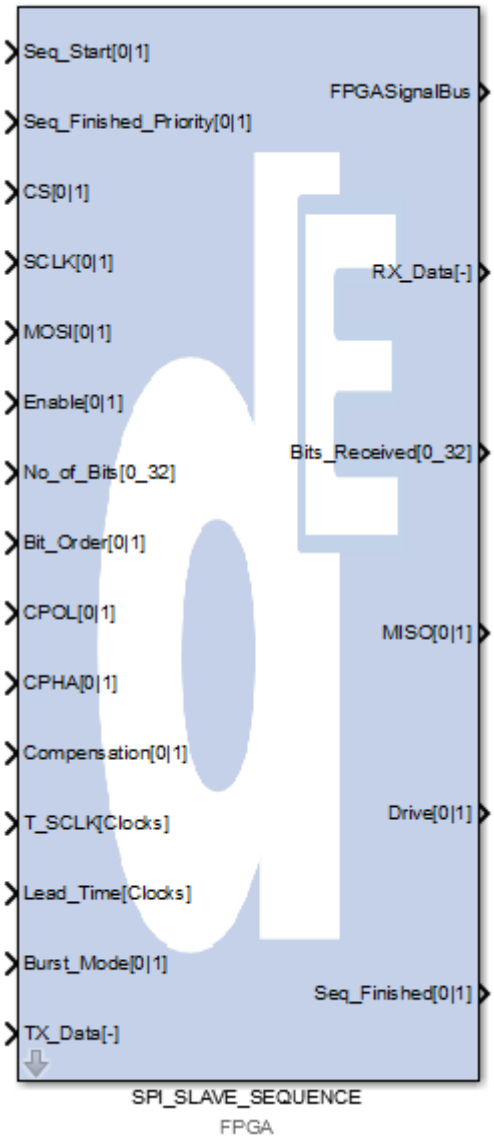

Figure 28: SPI_SLAVE_SEQUENCE FPGA Main Component

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Seq_Start	-	Flag for starting the sequence. Connect it to the *Seq_Finished* output of the prior sequence.	Bool
Seq_Finished_Priority	-	Flag to determine the priority of the sequence finish signal. If activated the sequence finish signal overrides the sequence start signal, when both are present at the same point in time.	Bool
CS	-	The chip select input.	Bool
SCLK	-	The serial clock input.	Bool
MOSI	-	The master-out-slave-in serial data input	Bool
Enable	-	Enables the sequence.	Bool
No_of_Bits	-	The maximum number of bits to be sent / received within this sequence.	UFix_6_0
Bit_Order	-	0 = LSB first 1 = MSB first	Bool
CPOL	-	The clock polarity	Bool
CPHA	-	The clock phase	Bool
Compensation	-	Activates delay compensation for this sequence.	Bool
T_SCLK	Clocks	The expected period of the serial clock for delay compensation.	UFix_16_0
Lead_Time	Clocks	The expected active time of the chip select signal before data transmission starts (only required for delay compensation).	UFix_16_0
Burst_Mode	-	If activated the sequence will end after the specified number of bits. If more sequences with activated burst mode are chained together they can be processed without a chip select inactive time between them.	Bool
TX_Data	-	The data to be sent via MISO output.	UFix_32_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 via MOSI input.	UFix_32_0
Bits_Received	-	The number of bits transferred within this sequence.	UFix_6_0
MISO	-	The master-in-slave-out. To be connected to a digital out block.	Bool
Drive	-	Flag indicating that the sequence and the chip select is active, so the MISO line can be driven. Connect it to the drive port of a digital out block. (Only DS2655M2. For DS2655M1 the driving is performed by hardware modification.)	Bool
Seq_Finished	-	Flag indicating that the sequence is finished. A sequence is finished if the number of bits specified are transmitted or the chip select signal goes inactive. Can be used to trigger the next sequence.	Bool

If a “sequence finished” event (rising edge of CS or number of bits transmitted) and a “sequence start” event occur at the same time, no new sequence is triggered. The *Seq_Start* input has to be raised **after** the sequence is finished. If for example a sequence shall be started right after the rising edge of CS, the input *Seq_Start* must be the rising edge of CS delayed by at least one clock cycle.
This behavior can be controlled via the Seq_Finished_Priority flag.

Processor Input

Block

Adapts the FPGA signals for the processor side.

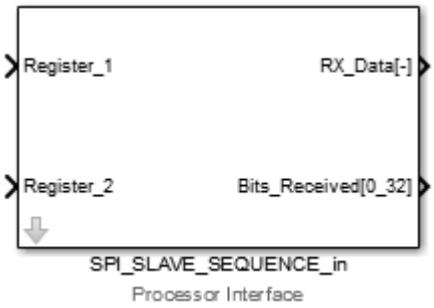

Figure 29: SPI_SLAVE_SEQUENCE_in block

Input

The processor input block has the following inputs:

Register 1

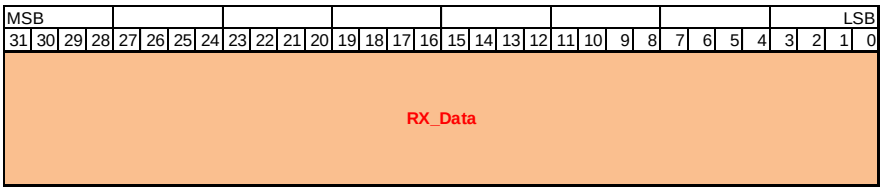

Figure 30: SPI_SLAVE_SEQUENCE_in Register 1

Name	Bits	Description
RX_Data	31..0	The data received via MOSI input

Register 2

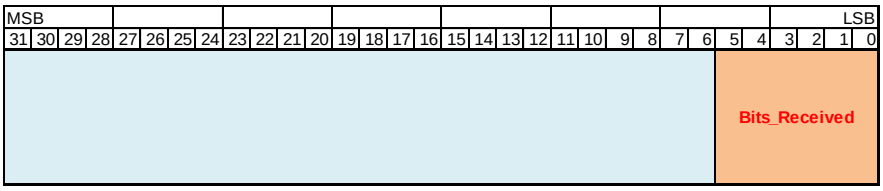

Figure 31: SPI_SLAVE_SEQUENCE_in Register 2

Name	Bits	Description
Bits_Received	5..0	The data received via MOSI input

Output

The SPI_SLAVE_SEQUENCE_in block has the following outputs:

Name	Unit	Description	Range
RX_Data	-	The data received via MOSI input.	Range: 0…2^{32}-1 Resolution: 1
Bits_Received	-	The number of data bits transferred within this sequence.	Range: 0..32 Resolution: 1

SPI Slave Lookup

Objective

Optionally, SPI slave responses to SPI master requests can be selected using look-ups. The SPI Slave Lookup block implements a single lookup pattern. If the request matches the pattern configured by the user, a match flag will be set at the output.

Content

The blockset contains the following elements:

- Processor Interface: SPI_SLAVE_LOOKUP_par (Processor Interface)
- Processor Interface: SPI_SLAVE_LOOKUP_out (Processor Interface)
- FPGA Interface: SPI_SLAVE_LOOKUP_in (FPGA Interface)
- FPGA: SPI_SLAVE_LOOKUP (FPGA Main Component)

Processor Parameterization

Block

Merges the processor signals and writes them to the FPGA.

Figure 32: SPI_SLAVE_LOOKUP_par block

Block Dialog

The processor parameterization block provides the following dialog:

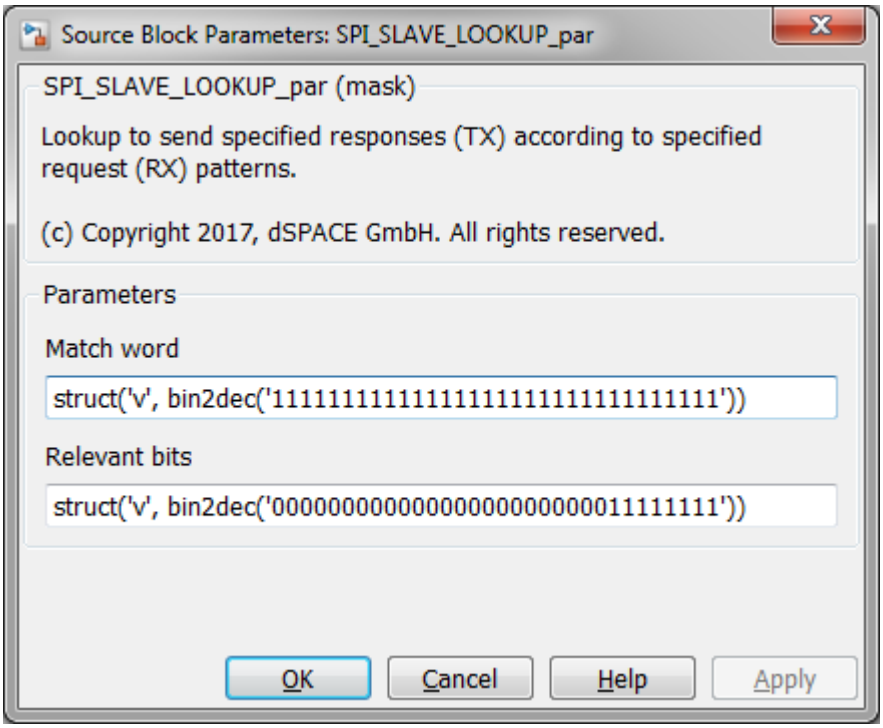

Figure 33: SPI_SLAVE_LOOKUP_par dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Match word	-	Defines the request to be matched.	Range: 0…32 Resolution: 1
Relevant bits	-	Set the bits which are relevant for matching the request. Bits set to 0 in this mask will be ignored during comparison. So setting all bits to 0 will result in the comparison to be always true.	0|1

Output

The Processor Parameterization block outputs 4 register contents, mapped to 1 register by time multiplexing. The sectioning is shown below:

Register 1.1

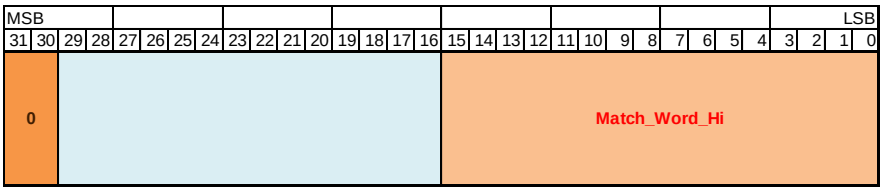

Figure 34: SPI_SLAVE_LOOKUP_par Register 1.1

Name	Bits	Description
Match_Word_Hi	15..0	High part of the match word

Register 1.2

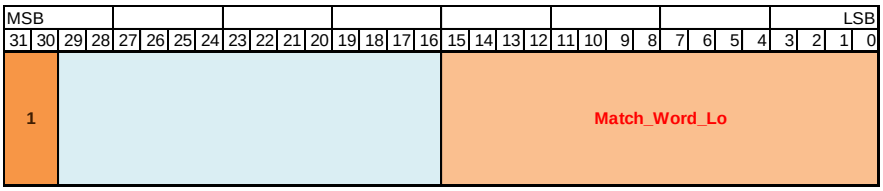

Figure 35: SPI_SLAVE_LOOKUP_par Register 1.2

Name	Bits	Description
Match_Word_Lo	15..0	Low part of the match word

Register 1.3

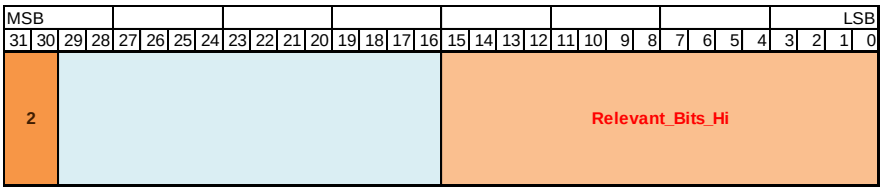

Figure 36: SPI_SLAVE_LOOKUP_par Register 1.3

Name	Bits	Description
Relevant_Bits_Hi	15..0	High part of the relevant bits

Register 1.4

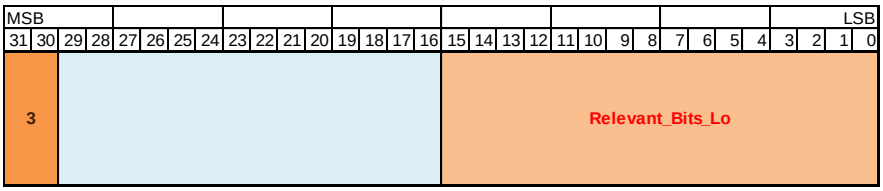

Figure 37: SPI_SLAVE_LOOKUP_par Register 1.4

Name	Bits	Description
Relevant_Bits_Lo	15..0	Low part of the relevant bits

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

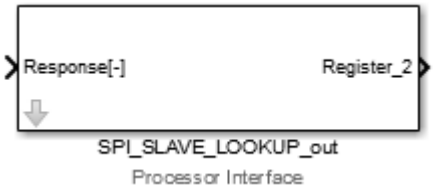

Figure 38: SPI_SLAVE_LOOKUP_out block

Input

The SPI_SLAVE_LOOKUP_out block has the following inputs:

Name	Unit	Description	Range
Response	-	The response to be sent if the request matches.	Range: 0…2^{32}-1 Resolution: 1

Output

The Processor Out block outputs 1 register. The sectioning is shown below:

Register 2

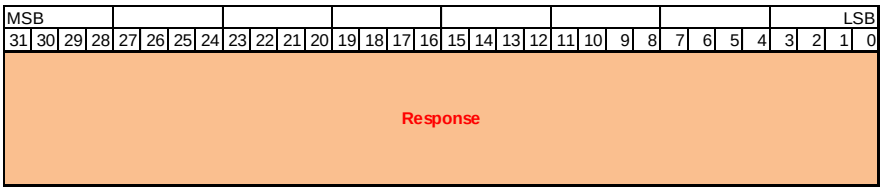

Figure 39: SPI_SLAVE_LOOKUP_out Register 2

Name	Bits	Description
Response	31..0	The response to be sent if the request matches.

FPGA Main Component

Block

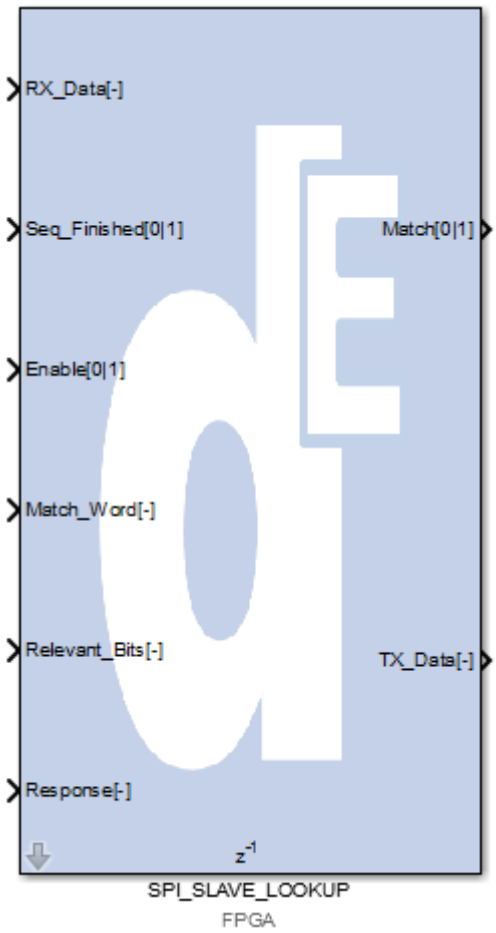

Figure 40: SPI_SLAVE_LOOKUP FPGA Main Component

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
RX_Data	-	The request received (output of a SPI Slave Sequence block).	UFix_32_0
Seq_Finished	-	Connect to *Seq_Finished* output of the SPI Slave Sequence block which receives the request.	Bool
Enable	-	The enable signal.	Bool
Match_Word	-	The word which has to be matched by the request.	UFix_32_0
Relevant_Bits	-	The relevant bits for the comparison.	UFix_32_0
Response	-	The response to send if the request matches.	UFix_32_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
Match	-	Flag indicating that the request has matched.	Bool
TX_Data	-	If the request has matched, the response is forwarded as TX data for a later sequence.	UFix_32_0

SPI Lookup Merge

Objective	This block can be used to merge multiple slave lookups.
Content	The blockset contains the following elements:

- FPGA: SPI_ LOOKUP_MERGE (FPGA Main Component)
- FPGA Interface: SPI_ LOOKUP_MERGE_out (FPGA Interface)
- Processor Interface: SPI_ LOOKUP_MERGE_in (Processor Interface)

FPGA Main Component

Block

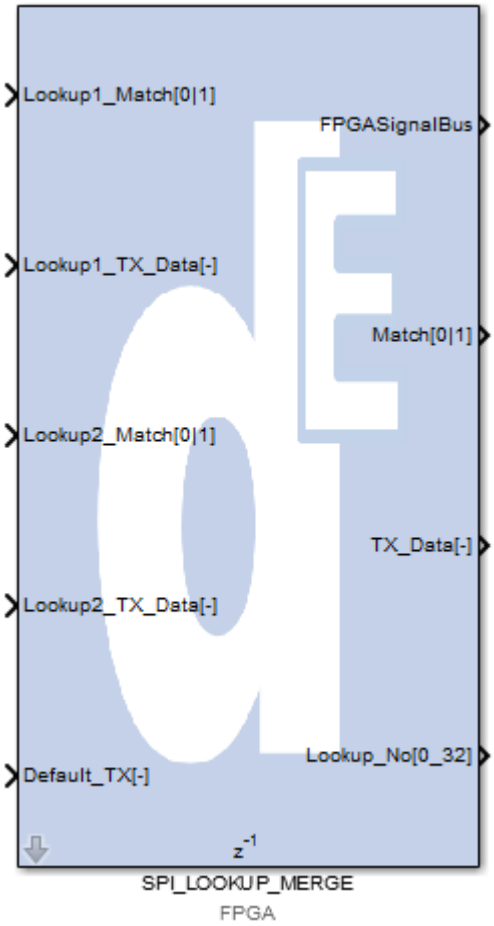

Figure 41: SPI_ LOOKUP_MERGE FPGA Main Component

Block Dialog

The SPI_LOOKUP_MERGE block provides the following dialog:

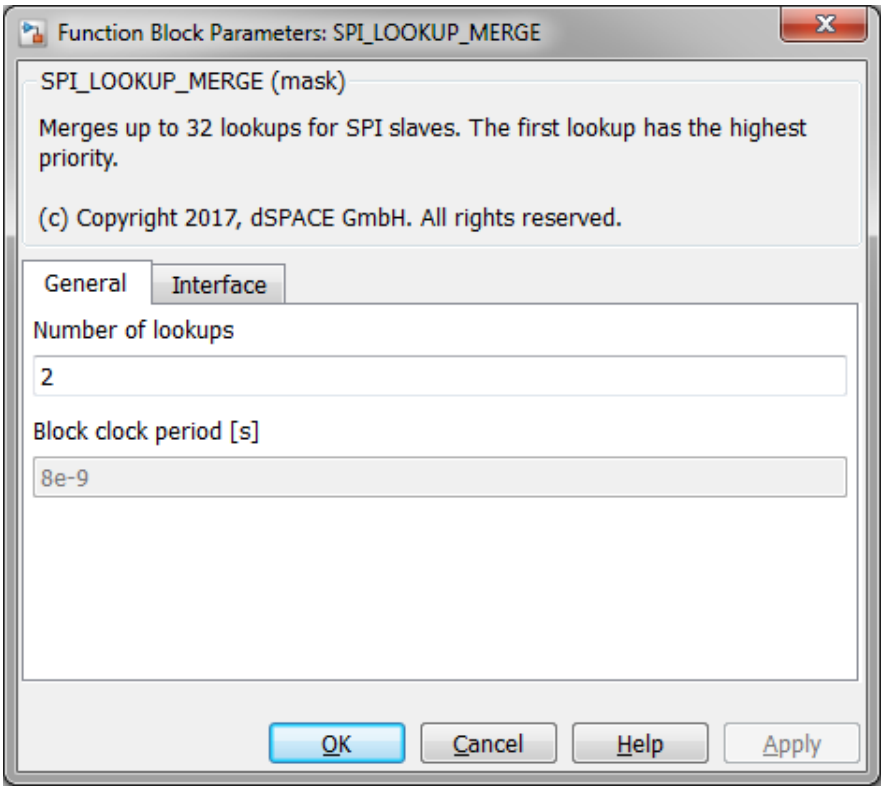

Figure 42: SPI_LOOKUP_MERGE dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Number of lookups	-	The number of lookups to be merged.	Range: 1…32 Resolution: 1

You can define the parameters of the input and output registers for automatic interface generation (on the interface tab). The dialog also contains a button to start interface generation.

Input

The main block has the following inputs:

Name	Unit	Description	Format
LookupX_Match	-	Match flag of a SPI Slave Lookup block.	Bool
LookupX_TX_Data	-	TX data output of a SPI Slave Lookup block.	UFix_32_0
Default_TX	-	The default response to be sent if there is no match on any of the lookups connected.	UFix_32_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
Match	-	Flag indicating that at least 1 request has matched.	Bool
TX_Data	-	The TX data of the lookup which matches the request. If multiple lookups match, the one with the lowest number get highest priority.	UFix_32_0
Lookup_No		The number of the lookup, which matches the request. If no lookup matches, the output is 0.	UFix_6_0

Processor Input

Block

Adapts the FPGA signals for the processor side.

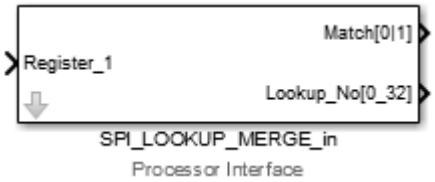

Figure 43: SPI_LOOKUP_MERGE_in block

Input

The processor input block has the following inputs:

Register 1

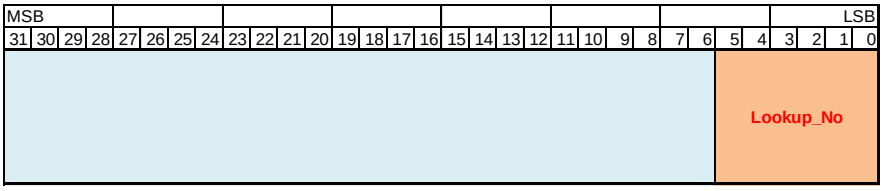

Figure 44: SPI_LOOKUP_MERGE_in Register 1

Name	Bits	Description
Lookup_No	5..0	The number of the lookup, which matches the request.

Output

The SPI_LOOKUP_MERGE_in block has the following outputs:

Name	Unit	Description	Range
Match	-	Flag indicating that at least 1 request has matched.	0|1
Lookup_No	-	The number of the lookup, which matches the request. If no lookup matches, the output is 0.	Range: 0..32 Resolution: 1

SPI CS Interrupt

Objective

The chip select interrupt can be used to trigger an interrupt to the processor model if a chip select line has been inactive for a certain time.

Content

The blockset contains the following elements:

- Processor Interface: SPI_CS_INTERRUPT_par (Processor Interface)
- FPGA Interface: SPI_CS_INTERRUPT_in (FPGA Interface)
- FPGA: SPI_CS_INTERRUPT (FPGA Main Component)

Processor Parameterization

Block

Merges the processor signals and writes them to the FPGA.

Figure 45: SPI_CS_INTERRUPT_par block

Block Dialog

The processor parameterization block provides the following dialog:

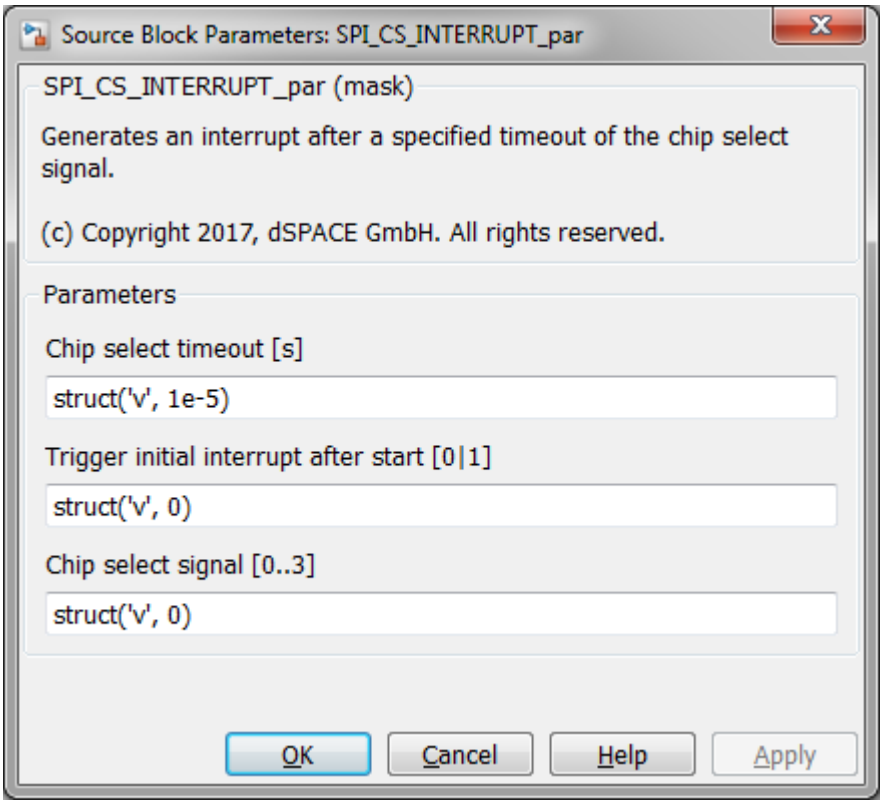

Figure 46: SPI_CS_INTERRUPT_par dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Chip select timeout	s	The time the chip select line must be inactive to trigger an interrupt.	Range: 0…2^{16}-1 * FPGA clock Resolution: FPGA clock
Trigger initial interrupt after start	-	If this option is set, an initial interrupt will be triggered on startup (for example to load initial data)	0|1
Chip select signal	-	With this parameter it is possible to select different chip select signals on the FPGA from the processor application.	0|1|2|3

Output

The Processor Parameterization block outputs 1 register. The sectioning is shown below:

Register 1

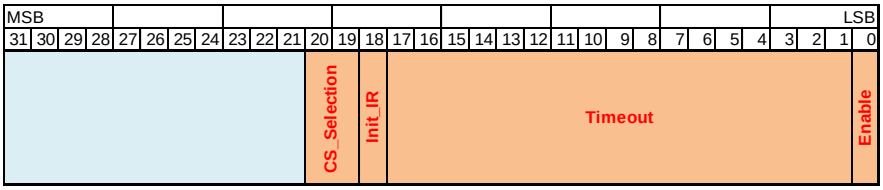

Figure 47: SPI_CS_INTERRUPT_par Register 1

Name	Bits	Description
Enable	0	The enable signal
Timeout	17..1	The timeout to trigger the interrupt
Init_IR	18	Flag to trigger initial interrupt
CS_Selection	20..19	Select different chip select signals

FPGA Main Component

Block

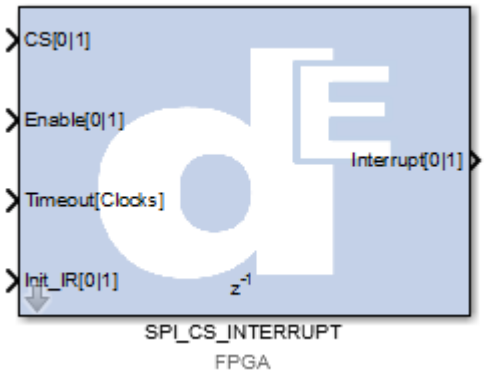

Figure 48: SPI_CS_INTERRUPT FPGA Main Component

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
CS	-	The chip select input.	Bool
Enable	-	The enable signal.	Bool
Timeout	Clocks	The time the chip select line must be inactive to trigger an interrupt.	UFix_17_0
Init_IR	-	If this option is set, an initial interrupt will be triggered on rising edge of Enable input	Bool

Output

The main block has the following outputs:

Name	Unit	Description	Format
Interrupt	-	The interrupt flag which will be generated after chip select timeout. Can be connected to an RTI FPGA interrupt block.	Bool

SPI SEQ Interrupt

Objective	The sequence interrupt can be used to trigger an interrupt to the processor model if a specified sequence finished its execution.
Content	The blockset contains the following elements:

- Processor Interface: SPI_SEQ_INTERRUPT_par (Processor Interface)
- FPGA Interface: SPI_SEQ_INTERRUPT_in (FPGA Interface)
- FPGA: SPI_SEQ_INTERRUPT (FPGA Main Component)

Processor Parameterization

Block Merges the processor signals and writes them to the FPGA.

Figure 49: SPI_SEQ_INTERRUPT_par block

Block Dialog

The processor parameterization block provides the following dialog:

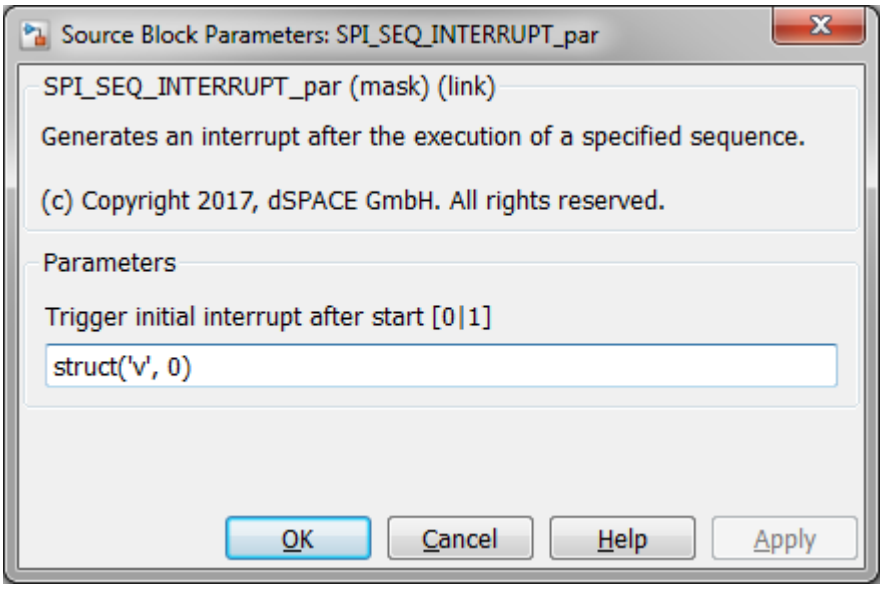

Figure 50: SPI_SEQ_INTERRUPT_par dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Trigger initial interrupt after start	-	If this option is set, an initial interrupt will be triggered on startup (for example to load initial data)	0|1

Output

The Processor Parameterization block outputs 1 register. The sectioning is shown below:

Register 1

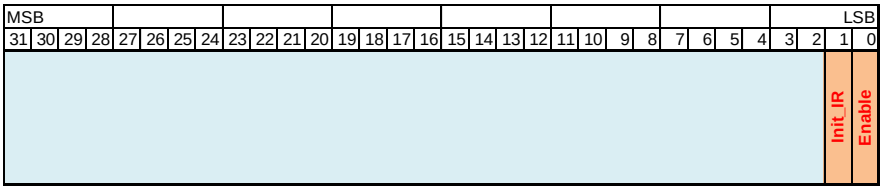

Figure 51: SPI_SEQ_INTERRUPT_par Register 1

Name	Bits	Description
Enable	0	The enable signal
Init_IR	1	Flag to trigger initial interrupt

FPGA Main Component

Block

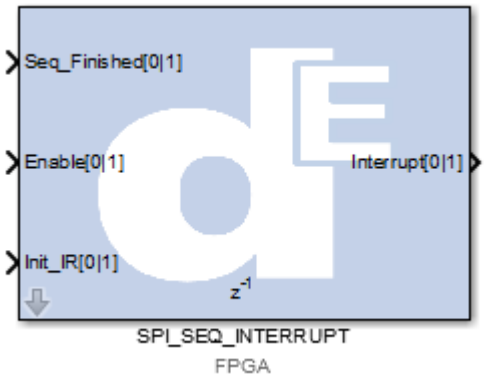

Figure 52: SPI_SEQ_INTERRUPT FPGA Main Component

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Seq_Finished	-	The sequence finished signal of an user-defined sequence	Bool
Enable	-	The enable signal.	Bool
Init_IR	-	If this option is set, an initial interrupt will be triggered on rising edge of Enable input	Bool

Output

The main block has the following outputs:

Name	Unit	Description	Format
Interrupt	-	The interrupt flag which will be generated after the connected sequence is finished. Can be connected to an RTI FPGA interrupt block.	Bool

CRC Calculation

Objective

The CRC Calculation block computes the checksum of the incoming data and provides the result after one FPGA clock cycle at the output. The block logic adapts to the user settings covering data width, seed and polynom.

Content

The blockset contains the following elements:

- FPGA: CRC_CALCULATION (FPGA Main Component)

FPGA Main Component

Block

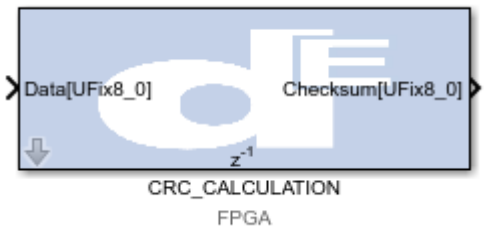

Figure 53: CRC_CALCULATION FPGA Main Component

Block Dialog

The CRC_CALCULATION block provides the following dialog:

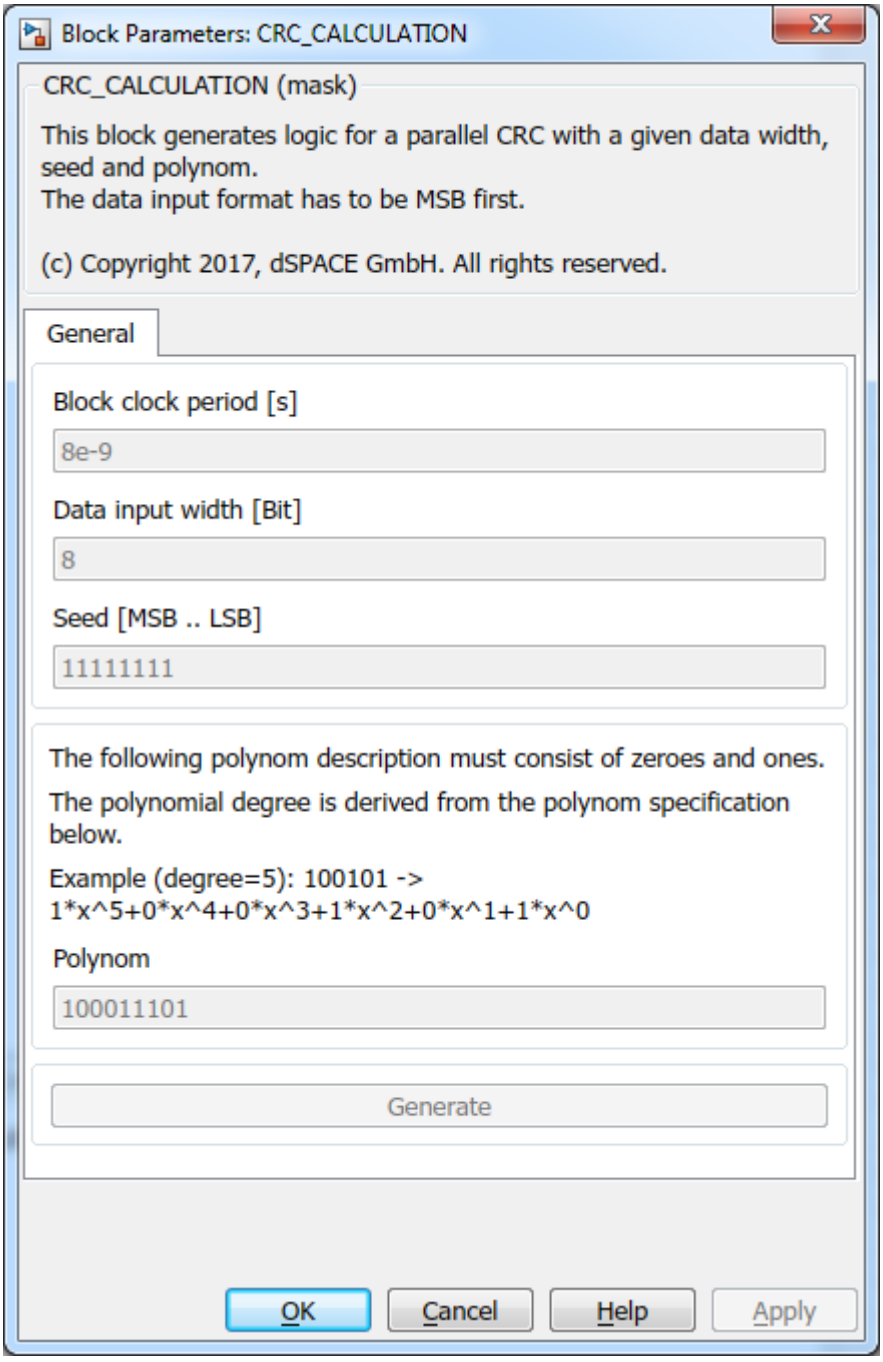

Figure 54: CRC_CALCULATION dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Block clock period	s	The clock period of the used FPGA platform	8e-9 | 10e-9
Data input width	bit	Bit width of the input data	1 - 384
Seed	digits	Initial value for the first xor operation with the input data	2 - 255
Polynom	digits	The divisor for the input data	3 - 256

With the Generate button the logic fitting to the corresponding block settings will be generated. The progress can be observed via the progress bar.

Before generating new block logic, the library link of the block has to be disabled or broken. Otherwise an error will be thrown.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Data	-	The input data for which the checksum will be generated	UFix

Output

The main block has the following outputs:

Name	Unit	Description	Format
Checksum	-	The generated checksum	UFix

BUS_FCN

Description / Overview

Sometimes it is useful or necessary to delay all signals of a Simulink bus in a customized model on the FPGA. In this case the BUS_FCN block can be used. Only scalar and muxed signals are supported.

Block

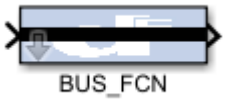

Figure 55: BUS_FCN block (direct feedthrough activated)

Block Dialog

The block provides the following dialog:

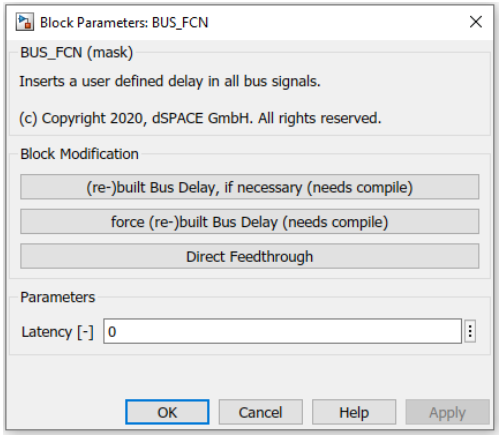

Figure 56: BUS_FCN dialog

The block dialog has the following parameters:

Name	Unit	Description	Range
Button: (re-) build Bus Delay, if necessary	-	Build the internal bus structure with delay blocks in each signal; (re-) build of the internal structure will proceed if necessary; if a (re-) build is necessary the model must be able to compile (successful CTRL-D)	-
Button: Force (re-) build Bus Delay	-	Forces a (re-) build of the internal structure; model must be able to compile (successful CTRL-D)	-
Direct Feedthrough	-	Internal bus delay block will be deleted and the in- and output are directly connected	
Latency	-	Latency of the overall delay	[0 .. x]

Input

The BUS_FCN block has the following input:

Name	Unit	Description	Format
In	-	Simulink input bus	-

Output

The BUS_FCN block has the following output:

Name	Unit	Description	Format
Out	-	Delayed output bus	-

SCALEXIO: Connecting FPGA Interface

Objective	The following section describes how to connect the pre-built FPGA application to the CN model in Configuration Desk.
Load FPGA Application	After the FPGA build process is finished a subfolder called *customio* is created in the *rtiFPGA* folder. This folder contains custom functions which have to be copied to the ConfigurationDesk application directory (\<project_name>\<application_name>\CustomFunctions). Reload the project. A custom function will be available in the functions view under Custom Functions / FPGA Blockset. Drag and drop this function to the Functions tab in your main window.
Connect FPGA Application	Right-click on the FPGA application in the main window, select *Hardware Assignment* and choose a suitable board connected (e. g. DS2655). Then connect all register inputs and outputs of the FPGA application to the model ports. If the auto-generate function has been used to create the interfaces, the numbering of all ports will fit to each other. So that the *Register In 1* port of the custom function block has to be connected to the *Register_Out_1* model port and vice versa the *Register Out 1* port of the custom function to the *Register_In_1* model port.