

XSG Advanced Engine Control

User Guide

Version 20.1 – 06/2020

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.ll@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.....	6
<i>Document Symbols and Conventions</i>	6
<i>Accessing PDF File.....</i>	7
<i>Related Documents.....</i>	8
<i>Requirements.....</i>	9
Features	10
Target Hardware	11
DS1514 FPGA Board + DS1552 I/O Module.....	11
DS1514 FPGA Board + DS1554 I/O Module.....	13
DS5203 FPGA Board.....	14
Simulink Model Structure.....	15
General Description.....	15
Example of Interfacing	16
The Processor Setup Block.....	17
The FPGA Setup Block.....	18
XSG Advanced Engine Control Library.....	19
Library Contents.....	19
Library Structure	20
Application Example.....	23
Injection & Ignition	24
ANGLE_SERIALIZER	27
Processor Output.....	28
FPGA Main Component.....	35
Processor Input.....	39
INJECTION_IGNITION_CORE.....	43
Processor Output.....	44
FPGA Main Component.....	50
Angle Measurement	56
ACU.....	57
Processor Output.....	59
FPGA Main Component.....	75
Processor Input.....	78
ENGINE_ENCODER	81
Processor Output.....	83
FPGA Main Component.....	88
Processor Input.....	92
Pressure Indication	94
CPI_4CYL	95
Processor Output.....	102

FPGA Main Component.....	115
Processor Input.....	120
COMB_DETECT_4CYL.....	126
Knock Detection	130
KNOCK_DETECTION	131
Processor Output.....	135
FPGA Main Component.....	141
Processor Input	144
Auxiliary Components	147
ENGINE_2_CYLINDER	148
Processor Output.....	150
FPGA Main Component.....	153
Processor Input	155
CYLINDER_2_ENGINE	156
FPGA Main Component.....	157
Processor Input.....	160
ANGLE_BASED_PULSE.....	161
Processor Output.....	162
FPGA Main Component.....	165
AABP_SERIALIZER	168
Processor Output.....	169
FPGA Main Component.....	172
Processor Input	175
ANGLE_INTERRUPT	177
Processor Output.....	178
FPGA Main Component.....	180
THRESHOLD_INTERRUPT	182
Processor Output.....	183
FPGA Main Component.....	185
RAPID_PRO_INPUT	187
Processor Output.....	188
FPGA Main Component.....	192
Bus Definitions	194
Injection & Ignition.....	195
IICAngleBus.....	195
IICFaultBus	197
IICDiagnosisBus	199
Angle Measurement.....	201
ACUCommonBus	201
ACUExtendedBus	202
ACUDiagnosisBus	204
Pressure Indication	207
CPICHannel[x]Bus	207
RapidPro Interface	213
RPLambdaBus	213
RPKnockBus	213
Demos	214
Template Demo.....	215

I/O Mapping	217
MicroAutoBox III.....	219
MABXII CPI Solution.....	220
Scaling	221
Processor Output.....	221
Processor Input.....	223
Template Model.....	224
I/O Mapping	227

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_ENGCON.pdf

Related Documents

Below is a list of documents that you are recommended to read when working with the XSG Engine Control 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
- XSG Utils User Guide

Requirements

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

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

Below you can find the compatibility matrix for dSPACE Release, Xilinx and MATLAB for the XSG Advanced Engine Control Programming Blockset for FPGA:

RTI FPGA Programming Blockset	dSPACE Release	Operating System	MATLAB	Xilinx Vivado and XSG
3.9	2020-A	Windows 7 64-bit, Windows 10 64-bit	R2018b R2019a R2020a (all 64 Bit)	2019.2 (64 Bit)

Figure 1: Compatibility matrix for XSG Advanced Engine Control 20.1

In the following table you find the compatibility matrix for dSPACE Release and MATLAB for the XSG Advanced Engine Control 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)

Features

Overview

The following main features are supported by the XSG Advanced Engine Control Solution:

- FPGA library for high-speed combustion engine control
- Support of various types of engines:
 - Diesel,
 - Gasoline,
 - Gas,
 - Multi-fuel,
 - ...
- 2-, 4- and 6-stroke engines
- Up to 12 cylinders on a single platform
- Output up to 16 pulses per combustion cycle per channel
- Angular resolution of 0.1° (ACU) or even finer (encoder)
- Support of any crankshaft / camshaft pattern, including reverse crank
- Controlled intervention in combustion process within a few microseconds
- All library components are open and accessible to users
- Components of XSG Advanced Engine Control library:
 - Injection and ignition multi-pulse generation
 - Crank/cam based angular computation unit (ACU)
 - Incremental encoder processing
 - Cylinder pressure indication
 - Knock detection (only DS1554)
 - Auxiliary components (angle transformations, angle-based pulses, interrupts, RapidPro interfaces)
- XSG Utils library included

Target Hardware

DS1514 FPGA Board + DS1552 I/O Module

Board description

Using the DS1514 FPGA base board for MicroAutoBox II or MicroAutoBox III in combination with the DS1552 (B1) Multi-I/O piggy-back module (shown in the figure below), an integration of a FPGA application into a dSPACE MicroAutoBox II or MicroAutoBox III can be performed.

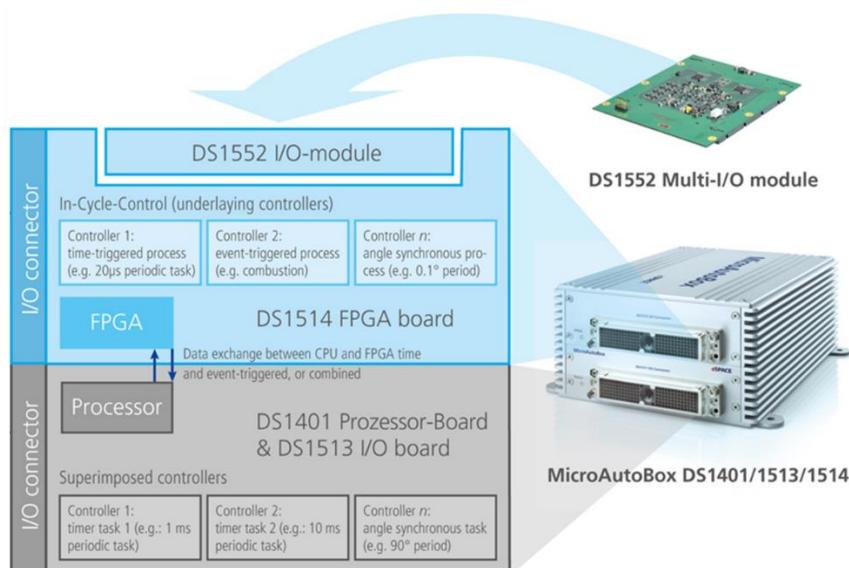


Figure 2: XSG Advanced Engine Control on MicroAutoBox II

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

- 8 A/D channels with a sample rate of 1 MHz and 0...+5 V (DS1552) or ±10 V (DS1552B1) 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

DS1514 FPGA Board + DS1554 I/O Module

Board description

Using the DS1514 FPGA base board for MicroAutoBox II or MicroAutoBox III in combination with the DS1554 Multi-I/O piggy-back module (shown in the figure below), an integration of a FPGA application into a dSPACE MicroAutoBox II or MicroAutoBox III can also be performed.



Figure 3: DS1554 Engine Control I/O Module

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 DS1554 I/O features the following:

- 5 Digital In channels with configurable thresholds dedicated for crank/cam measurement
- 1 crank/cam channel with zero crossing detection for crank/cam measurement
- 4 16-bit-resolution 1 MSPS Analog-In channels dedicated for knock measurement
- 2 Lambda channels, which support the *Bosch LSU 4.9*, the *Bosch LSU ADV gasoline & diesel* and the *Bosch LSU 5.1* lambda probes
- 40 digital out channels for general purpose actuator control
- 8 bidirectional in-/output channels
- 14 16-bit-resolution, 1MSPS Analog-In channels, e.g. for cylinder pressure indication (4 channels can be equipped with shunt resistors for current measurement)

DS5203 FPGA Board

Board description

Using 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.

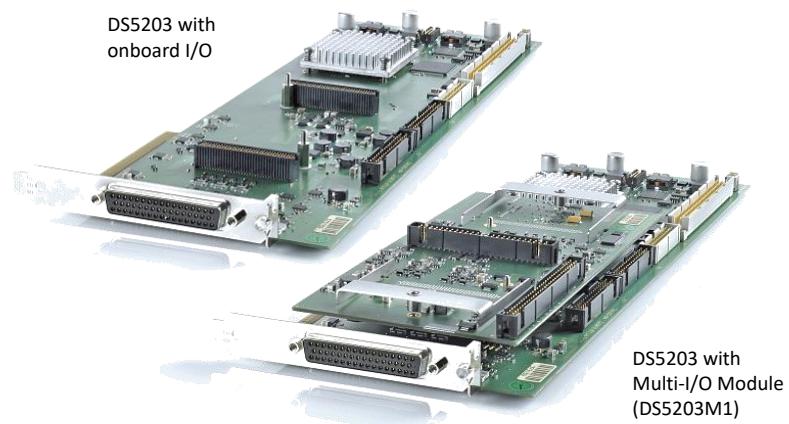


Figure 4: DS5203 Boards (PHS-based)

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 resources)
- Kintex-7 410T (larger amount of resources)

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.

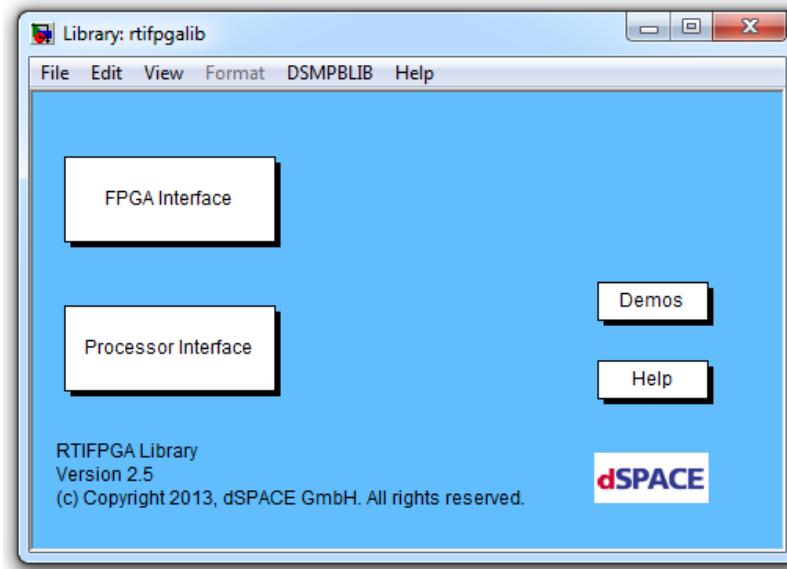


Figure 5: RTI FPGA library

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

The XSG Engine Control library 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. **Processor 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 Advanced Engine Control Library. The other blocks (6-9) are RTI FPGA interface blocks that define the basic communication between FPGA, processor and I/O.

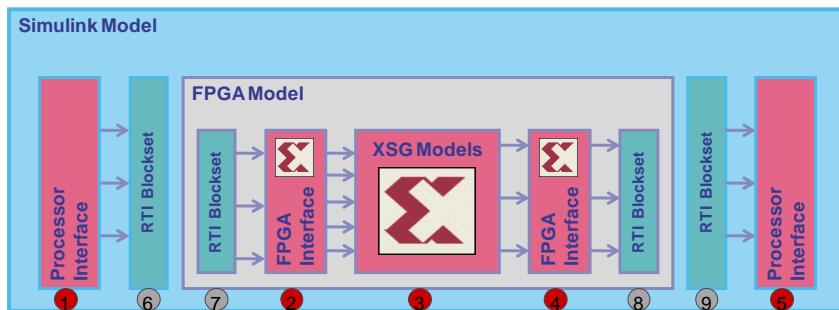


Figure 6: Using library components

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).

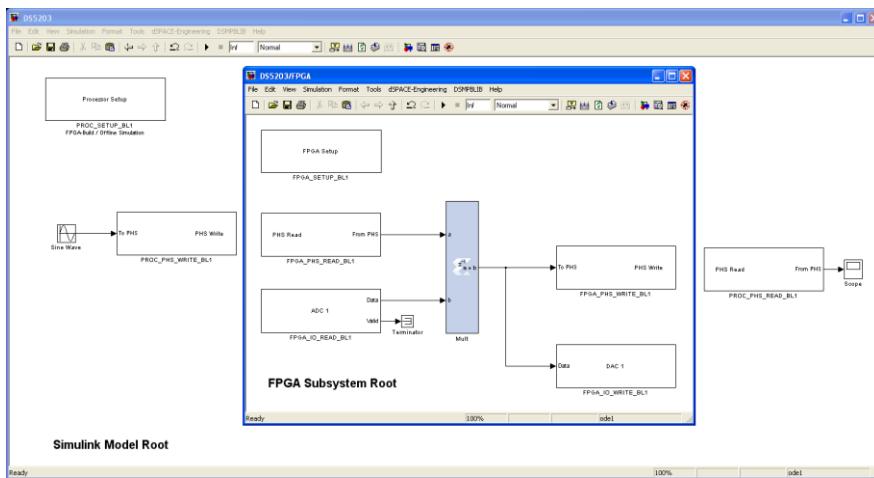


Figure 7: RTI interface usage

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 Advanced Engine Control Library

Library Contents

Description / Overview	The XSG Advanced Engine Control Library is designed to provide easy access for rapid control prototyping of advanced engine control or parts of it on Xilinx FPGAs. It contains the five sub-libraries described below.
Injection & Ignition	This sub-library is used to handle the generation of injection and ignition pulses out of user-defined start angles, end angles, or pulse durations. Up to 16 pulses per combustion cycles can be generated for each control signal. The pulses can be specified either angle-angle- or angle-duration-based. They can be updated even during active injection or ignition.
Angle Measurement	This sub-library contains the processing of various crankshaft position sensors for angle and speed measurement. Currently it contains a crank/cam input based angular processing unit and the processing of incremental encoders. The crank/cam based ECU determines the engine angle with a resolution of 0.1° and supports reverse crank as well. On top of that camshaft phase shift measurement is supported.
Pressure Indication	This sub-library contains algorithms to determine the in-cylinder pressure parameters during combustion. One CPI component supports up to 4 cylinder. For more cylinder support, add another CPI component.
Auxiliary Components	This sub-library contains several useful elements like angle transformations, generation of angle-based pulses, interrupt triggers and interfaces to RapidPro.
Utils	The main library also contains a separate library called XSG Utils. This library contains frequently used elements, which are relevant for various control use cases (e.g. PWM generation and measurement, digital filters, PI controller, scaling and scope functionalities). Most FPGA main components include interfaces to the processor model. For further information about these blocks, please refer to the XSG Utils documentation.

Library Structure

General information

The XSG Advanced Engine Control Library is divided into:

- XSG Advanced Engine Control Library (FPGA-based blocks)
- XSG Engine Control Interface Library (processor-based blocks)

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

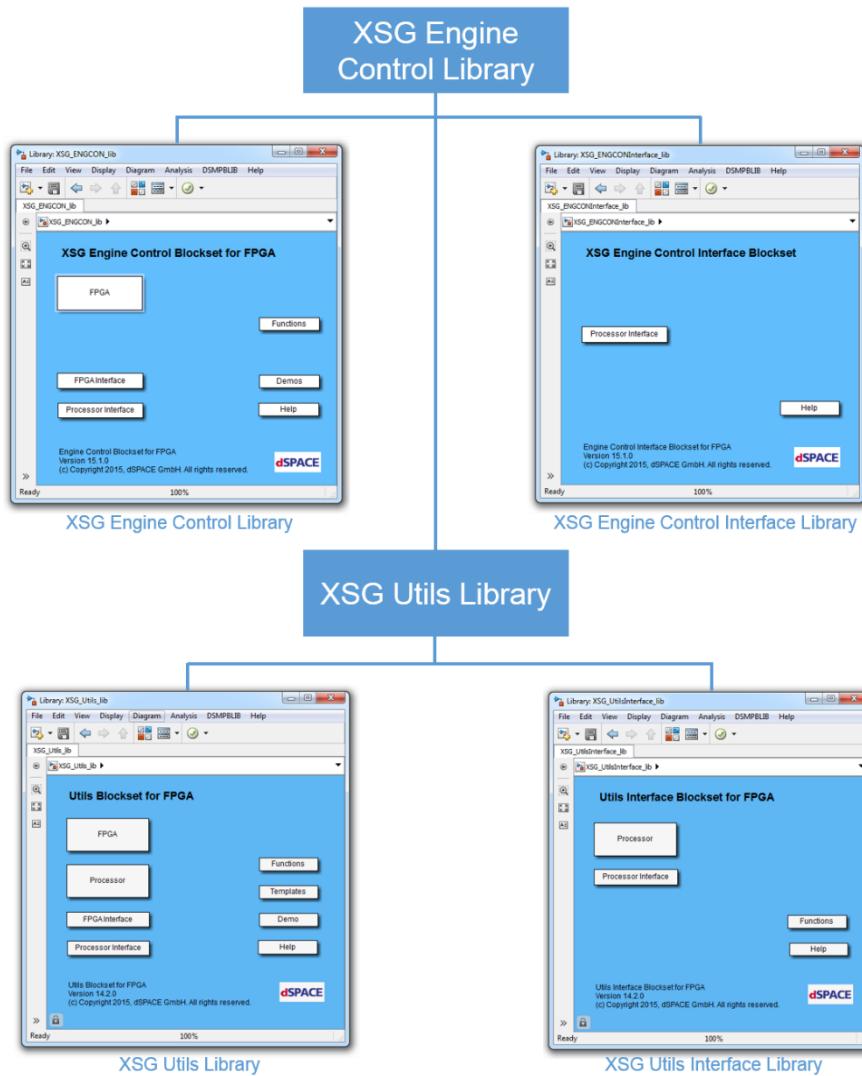


Figure 8: XSG Advanced Engine Control library structure

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

XSG Engine Control Library

The XSG Engine Control Library provides access to FPGA based engine control functions. It allows highly dynamic in-cycle control for various types of combustion engines. The library is divided into FPGA / Processor Interface and FPGA components as shown in the figure below. To open the library, type "XSG_ENGCON_lib" in the MATLAB Command Window or access the library via Simulink browser.

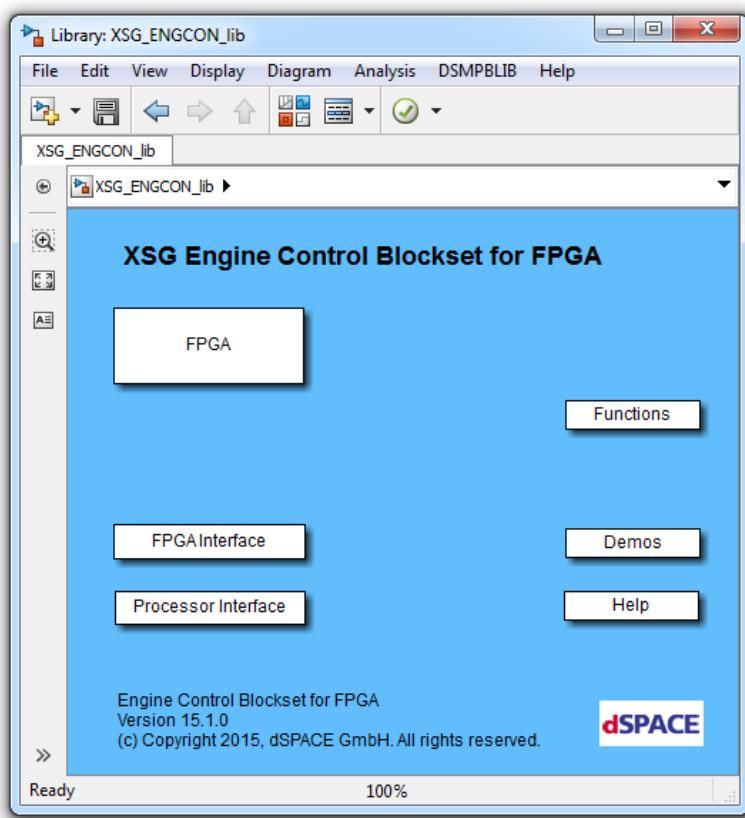


Figure 9: XSG Engine Control library

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.

	<p>The XSG Engine Control Library also contains the XSG Engine Control Interface Library.</p>
---	---

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

XSG Engine Control Interface Library

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

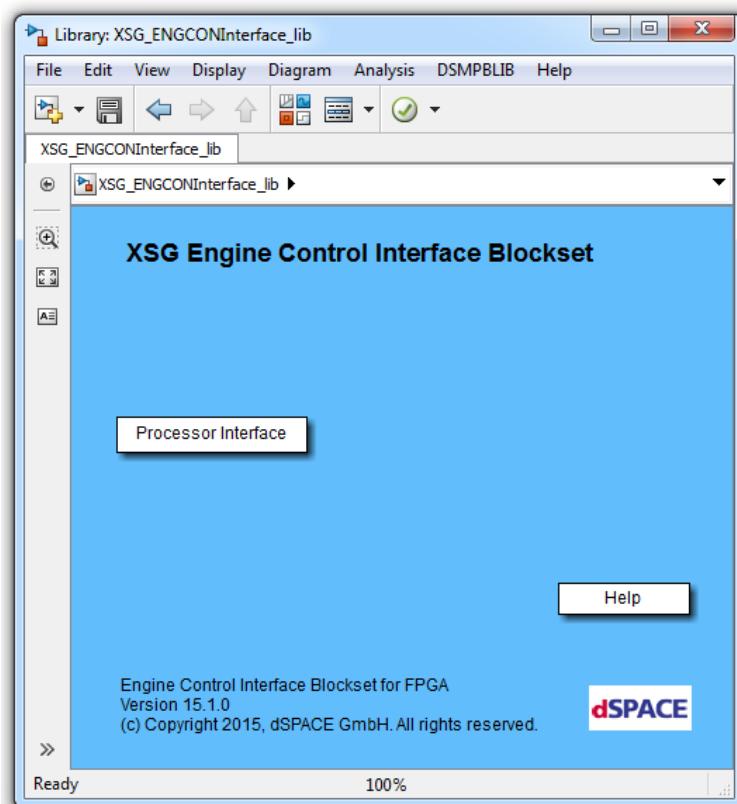


Figure 10: XSG Engine Control Interface library

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

Application Example

Description

Figure 11 shows an overview of the combination of XSG Advanced Engine Control components. The FPGA model contains the ACU, CPI and injection/ignition modules from the library as well as low-level control algorithms designed by the user. The processor model contains interface components of the library as well as high-level control algorithms designed by the user.

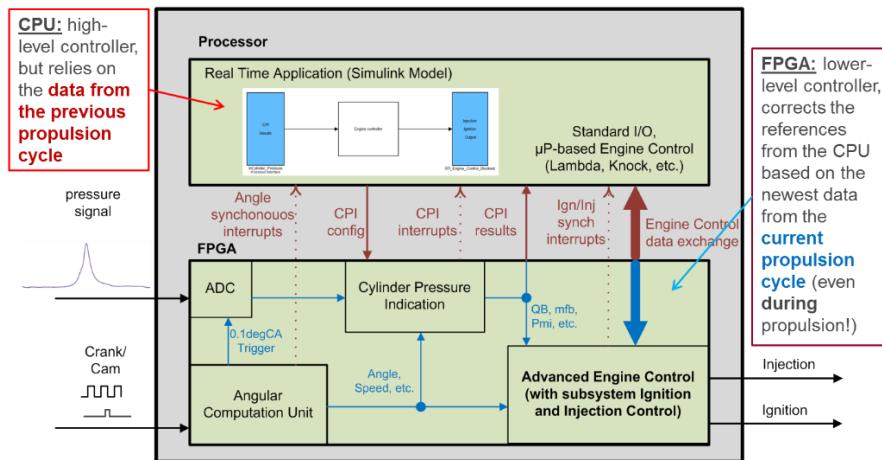


Figure 11: Combination of XSG Advanced Engine Control components

Injection & Ignition

Objective	The Injection & Ignition sub-library is used to handle the generation of injection and ignition pulses out of user-defined start angles, end angles, or pulse durations.
------------------	--

Features	The <i>Injection & Ignition</i> components provide the following main features:
-----------------	---

- Single- and double injection control signal mode supported. Double-injection control signal mode supports peaks and hold phases (e.g. with RapidPro).
- Angle-angle or angle-duration based pulse generation supported. For injection a start angle and duration can be specified, for ignition an end (spark) angle and duration can be specified.
- Pulses can be updated during active injection/ignition. Multiple checks are implemented within the FPGA component to avoid illegal angle changes.
- For each control signal, up to 16 pulses can be generated per combustion cycle. All pulses can be specified before or during the active cycle.

Content	The library contains the following components:
----------------	--

- ANGLE_SERIALIZER
- INJECTION_IGNITION_CORE

RapidPro Direct Injection Module	Injection signals generally contain up to three phases:
---	---

- Peak A
- Peak B (optional)
- Hold

The RapidPro direct injection module DS1664 (PS-DINJ 2/1) can be operated in two different injection modes to produce those three phases:

- Single control signal mode. In this mode a single injection pulse signal is generated by the controller platform (e.g. MicroAutoBox). The duration of peak A is set up by the configuration of the RapidPro Power Unit (via Configuration Desk). Peak B is not generated. Only the duration of the hold phase can be controlled by the injection pulse.
- Double control signal mode. Two injection pulses are generated by the controller. Using the two binary signals, four states (peak A, peak B, hold,

off) can be encoded. This way, the duration of all phases can be adjusted in real-time by the controller platform.

The XSG Engine Control library is able to generate both sorts of pulse pattern.

Alternatively, two independent injection pulses (A, B) can be generated to drive third-party hardware.

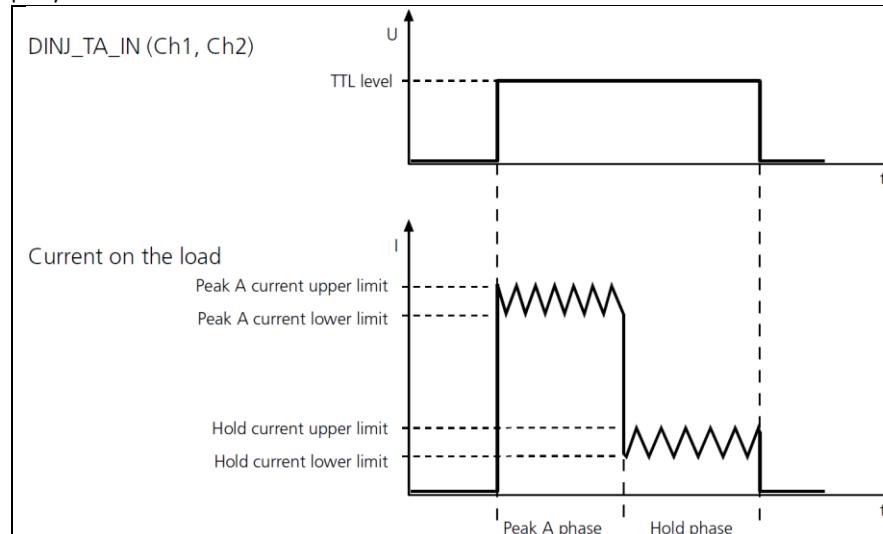


Figure 12: DS1664 single control signal mode

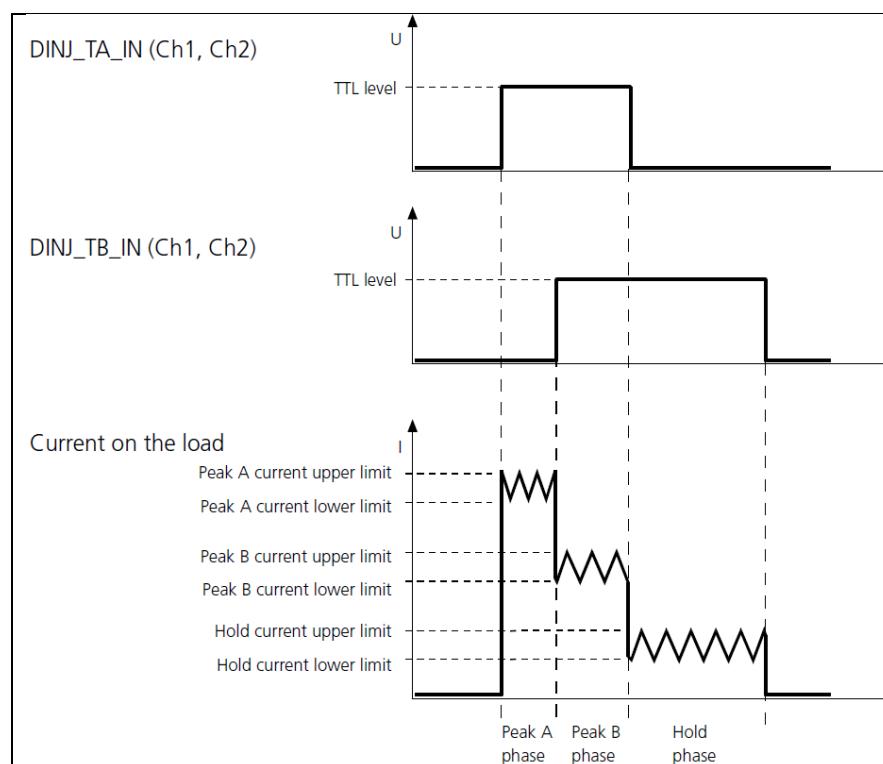


Figure 13: DS1664 double control signal mode

Overlapping pulses

If injection or ignition pulses overlap, there are generally two possible strategies to resolve this conflict:

- **Merge pulses.** In this case overlapping pulses will be merged to one single pulse
- **Remove pulses.** In this case the pulse starting while another pulse is still active will be completely removed.

The XSG Engine Control library is able to perform both strategies to resolve overlapping pulses.

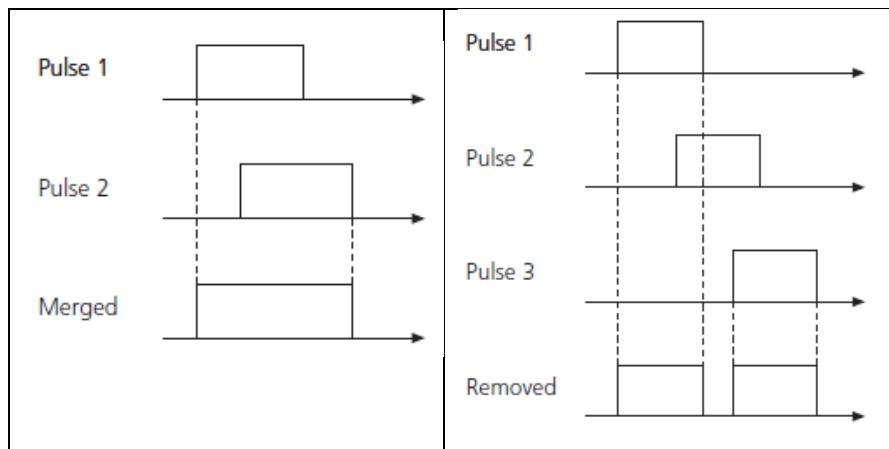


Figure 14: Merge and remove of overlapping pulses



If the DS1664 double control signal mode is activated, both control signals (INJ.A, INJ.B) are considered in conjunction for merge or remove. If remove is selected, a whole pulse (peak and hold) will be removed if any signal of this cycle overlaps with any signal of a previous cycle. If merge is selected, only the hold phase will be extended in the case of overlap.

ANGLE_SERIALIZER

Objective

This component selects the next ignition and injection start and end angles, according to the start and end angles defined by the user and the current cylinder angle.

	The angle input vectors for injection or ignition must be of ascending order. However the starting point does not matter. For example [-100°, 0°, 100°] as well as [0°, 100°, -100°] are valid angle inputs, while [0°, -100°, 100°] or [100°, 0°, -100°] are invalid inputs!
	If a pulse completely covers another pulse, this input is invalid as well (e.g. first pulse 0°-100°, second pulse 20°-80°).
	The minimum distance between two injection/ignition start or end angles is 0.1° CA.
	The generation of injection and ignition pulses are activated or deactivated during runtime using the inputs of the INJECTION_IGNITION_CORE.

Content

The blockset contains the following elements:

- Processor Interface: ANGLE_SERIALIZER_out (Processor Interface)
- FPGA Interface: ANGLE_SERIALIZER_in (FPGA Interface)
- FPGA: ANGLE_SERIALIZER (FPGA Main Component)
- FPGA Interface: ANGLE_SERIALIZER_out (FPGA Interface)
- Processor Interface: ANGLE_SERIALIZER_in (Processor Interface)

Processor Output

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

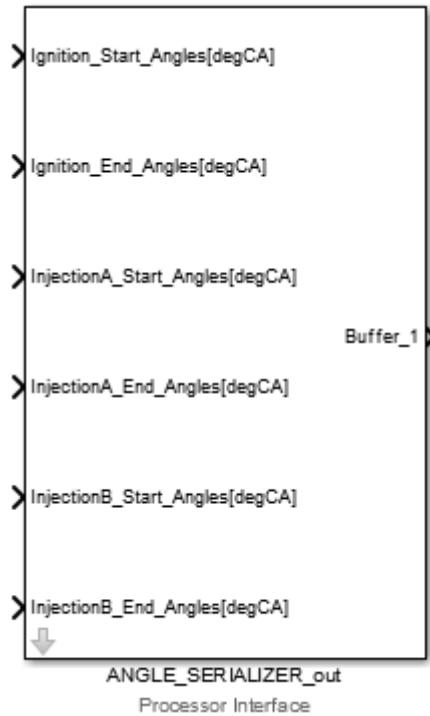


Figure 15: ANGLE_SERIALIZER_out block

Block Dialog

The processor output block provides the following dialog:

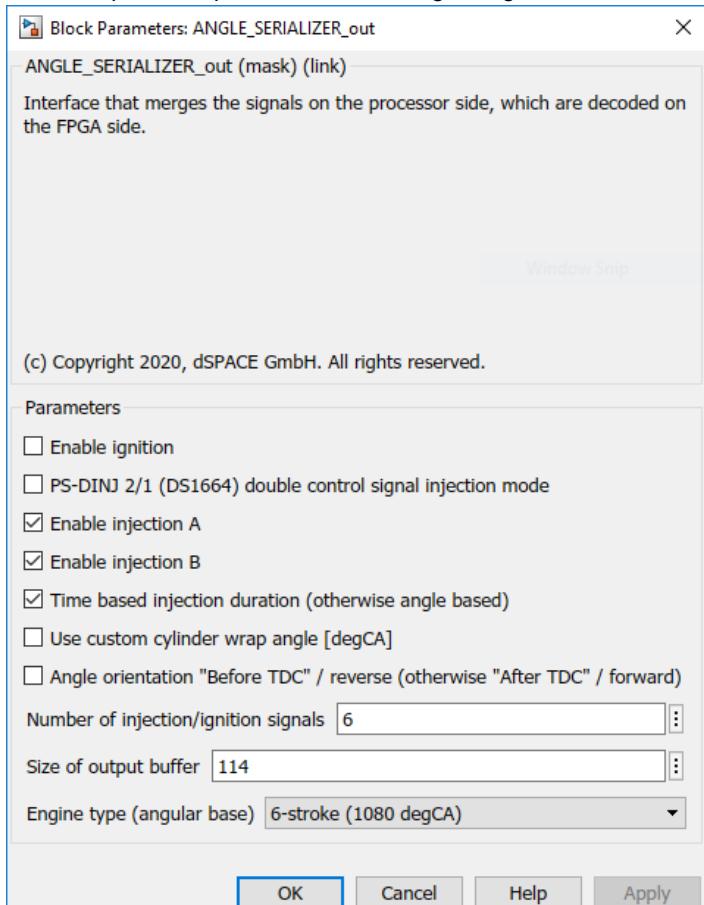


Figure 16: ANGLE_SERIALIZER_out dialog

The block dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Enable ignition		Enables the generation of ignition start and end angles.	on / off
Time based ignition duration	-	Ignition pulses are defined by ignition time and end (spark) angle, otherwise by start angle and end (spark) angle.	on / off
PS-DINJ 2/1 (DS1664) double control signal injection mode	-	If this option is set, the injection parameters can be set according to the dual control operation mode of the DS1664 RapidPro direct injection modules.	on / off
Enable injection A	-	Enables the generation of injection A control signal start and end angles	on / off
Enable injection B	-	Enables the generation of injection B control signal start and end angles	on / off
Time based injection duration	-	Injection pulses are defined by start angle and injection time, otherwise by start angle and end angle	on / off
Use custom cylinder wrap angle	°CA	If selected, a custom wrap angle can be specified for the cylinder angles. Otherwise the default wrap angle (e.g. 360° for 4-stroke engines) is applied.	Range: 0° ...1080° Resolution: 0.004°
Angle orientation "Before TDC"	-	If this option is checked the direction of the cylinder angles is reverse to the direction of the engine angle.	on / off
Number of injection/ignition signals	-	Number of injection/ignition signals	Range: 1...16
Size of output buffer	-	Size of the connected output buffer	Dependant on input. Maximum: 114
Engine type (angular base)	-	The engine type can be specified here. The angle ranges depend on the engine type (360°/720°/1080°)	2-stroke (360°CA) 4-stroke (720°CA) 6-stroke (1080°CA)



The information about engine and cylinder wrap angles are required by several components. As the wrap angles are usually the same for all components, it is recommended to use a variable for these parameters.

Input

The ANGLE_SERIALZER_out block has the following inputs:

Name	Unit	Description	Range
Ignition_Start_Angles	°CA	Start angles for ignition pulses (up to 16 elements)	Range: 0°...1080° Resolution: 0.004°
Ignition_Times	s	Durations for ignition pulses (up to 16 elements)	Range: 0s...<16s Resolution: 8ns
Ignition_End_Angles	°CA	Spark angles for ignition pulses (up to 16 elements)	Range: 0°...1080° Resolution: 0.004°
InjectionA_Start_Angles	°CA	Start angles for injection signal A pulses (up to 16 elements).	Range: 0°...1080° Resolution: 0.004°
InjectionA_End_Angles	°CA	End angles for injection signal A pulses (up to 16 elements).	Range: 0°...1080° Resolution: 0.004°
InjectionA_Times	s	Durations for injection signal A pulses (up to 16 elements).	Range: 0s...<16s Resolution: 8ns
InjectionB_Start_Angles	°CA	Start angles for injection signal B pulses (up to 16 elements).	Range: 0°...1080° Resolution: 0.004°
InjectionB_End_Angles	°CA	End angles for injection signal B pulses (up to 16 elements).	Range: 0°...1080° Resolution: 0.004°
InjectionB_Times	s	Durations for injection signal B pulses (up to 16 elements).	Range: 0s...<16s Resolution: 8ns
Injection_Start_Angles	°CA	If the PS-DINJ 2/1 mode is selected, this input represents the injection start angles (up to 16 elements).	Range: 0°...1080° Resolution: 0.004°
PeakA_Times	s	If the PS-DINJ 2/1 mode is selected, this input represents the durations of peak A (up to 16 elements).	Range: 0s...<16s Resolution: 8ns
PeakB_Times	s	If the PS-DINJ 2/1 mode is selected, this input represents the durations of peak B (up to 16 elements).	Range: 0s...<16s Resolution: 8ns
Injection_Times	s	If the PS-DINJ 2/1 mode is selected, this input represents the durations of the whole injection pulses (up to 16 elements).	Range: 0s...<16s Resolution: 8ns

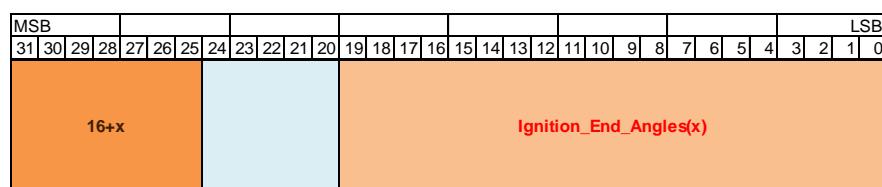
Output

The Processor Out block outputs up to 98 register contents, mapped to 1 buffer. The actual amount of data is dependent on the settings and the length of the input vectors. All data is transferred each simulation step.

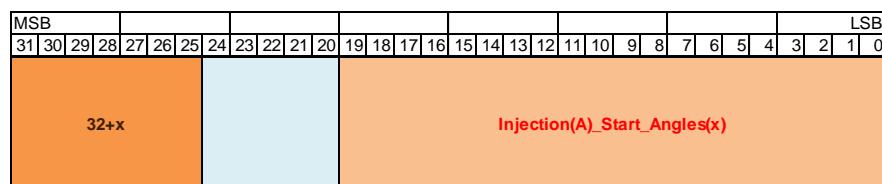
Register 1...16



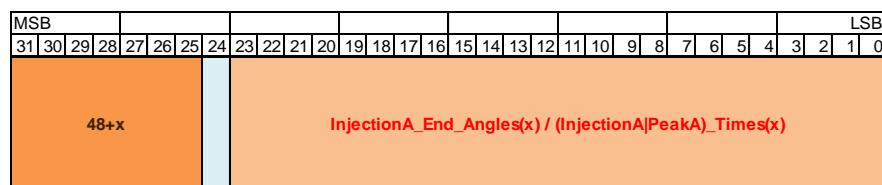
Register 17...32



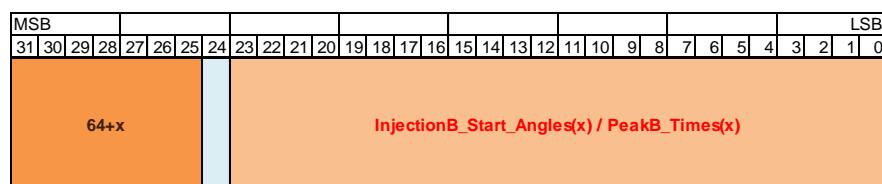
Register 33...48



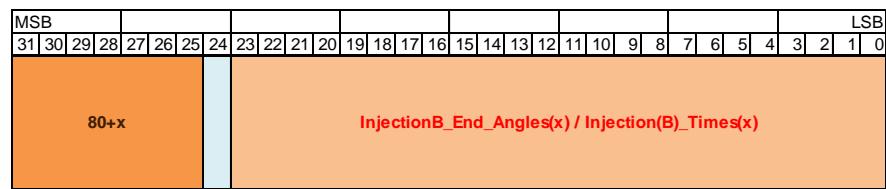
Register 49...64



Register 65...80



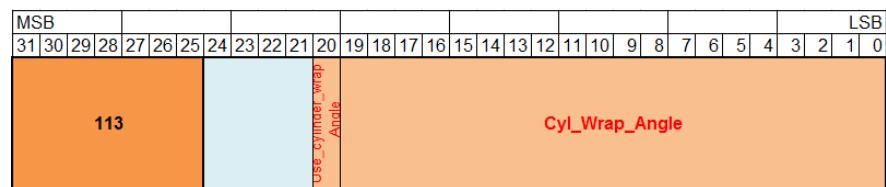
Register 81...96



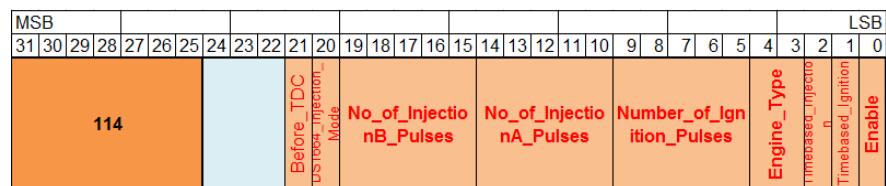
Register 97-112



Register 113



Register 114



FPGA Main Component

Block



Figure 17: ANGLE_SERIALIZER 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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Cylinder_Angle	°CA	The cylinder angle derived from the engine angle and the TDC.	Fix_20_8
Engine_Speed	rpm	The engine speed calculated by the ACU. Only required for time based injection or ignition pulses.	UFix_18_4
Direction	-	The direction of the engine motion.	Bool
Reset	-	Resets the angle selection.	Bool
Ignition_Start_Angles	°CA	Start angles for ignition pulses (16 elements).	Fix_20_8 (16x)
Ignition_Times	s	Durations for ignition pulses (16 elements).	UFix_31_27 (16x)
Ignition_End_Angles	°CA	Spark angles for ignition pulses (16 elements).	Fix_20_8 (16x)
No_of_Ignition_Pulses	-	Number of actually used ignition pulses.	UFix_5_0
DS1664_Injection_Mode	-	Activates the PS-DINJ 2/1 dual signal injection mode.	Bool
Injection(A)_Start_Angles	°CA	Start angles for injection signal A pulses (16 elements). In case of PS-DINJ 2/1 mode this input represents the injection start angles.	Fix_20_8 (16x)
InjectionA_End_Angles	°CA	End angles for injection signal A pulses (16 elements).	Fix_20_8 (16x)
(InjectionA PeakA)_Times	s	Durations for injection signal A pulses (16 elements). In case of PS-DINJ 2/1 mode this input represents the peak A durations.	UFix_31_27 (16x)
No_of_Injection(A)_Pulses	-	Number of actually used injection A pulses. In case of DS1664 double control signals mode, this input sets the number of injection pulses in general.	UFix_5_0
InjectionB_Start_Angles	°CA	Start angles for injection signal B pulses (16 elements).	Fix_20_8 (16x)
PeakB_Times	s	In case of PS-DINJ 2/1 mode this input represents the peak B durations.	UFix_31_27 (16x)
InjectionB_End_Angles	°CA	End angles for injection signal B pulses (16 elements).	Fix_20_8 (16x)

Injection(B)_Times	s	Durations for injection signal B pulses (16 elements). In case of PS-DINJ 2/1 mode this input represents the total injection durations.	UFix_31_27 (16x)
No_of_InjectionB_Pulses	-	Number of actually used injection B pulses. In case of DS1664 double control signals mode, this input has no function.	UFix_5_0
Cyl_Wrap_Angle	°CA	The wrap angle for the cylinder angle.	UFix_19_8
Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
IICAngleBus	-	Contains the next injection and ignition angles calculated by this unit (see IICAngleBus).	Bus
IICFaultBus	-	Contains validity checks for angles provided to or calculated by this unit (see IICFaultBus).	Bus

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	10455	2.56%
LUTs	9585	4.70%
Block RAM	2	0.44%
DSP Slices	4	0.48%

Processor Input

Block	Adapts the FPGA signals for the processor side.
--------------	---

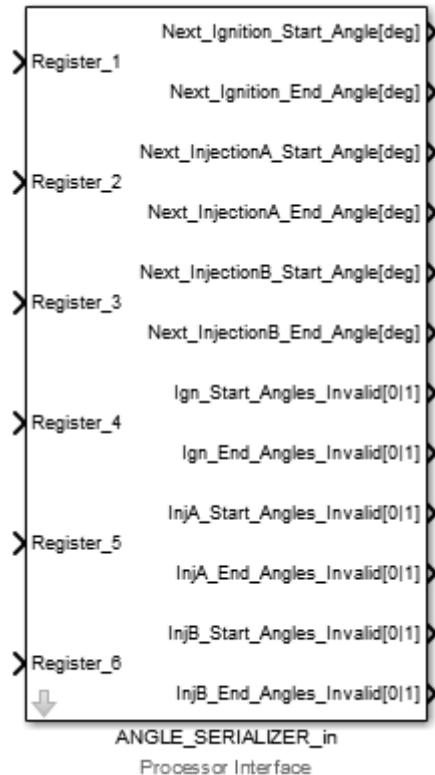


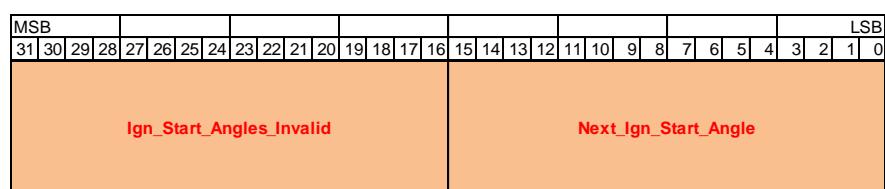
Figure 18: ANGLE_SERIALIZER_in block



This block is only used to provide output signals to the processor application. If the signals are not required in the processor application, computation resources can be saved by not using this block.

Input	The processor input block has 6 input register with a sectioning shown below.
--------------	---

Register 1



Register 2

MSB	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	LSB
Ign_End_Angles_Invalid															Next_Ign_End_Angle																		

Register 3

MSB	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	LSB
InjA_Start_Angles_Invalid															Next_InjA_Start_Angle																		

Register 4

MSB	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	LSB
InjA_End_Angles_Invalid															Next_InjA_End_Angle																		

Register 5

MSB	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	LSB
InjB_Start_Angles_Invalid															Next_InjB_Start_Angle																		

Register 6

MSB	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	LSB
InjB_End_Angles_Invalid															Next_InjB_End_Angle																		

Output

The ANGLE_SERIALIZER_in block has the following outputs:

Name	Unit	Description	Range
Next_Ignition_Start_Angle	°CA	The next ignition start angle, selected among all input angles, according to the current cylinder angle.	Range: ±1080° Resolution: 0.06°
Next_Ignition_End_Angle	°CA	The next ignition end angle, selected among all input angles, according to the current cylinder angle.	Range: ±1080° Resolution: 0.06°
Next_InjectionA_Start_Angle	°CA	The next injection A start angle, selected among all input angles, according to the current cylinder angle.	Range: ±1080° Resolution: 0.06°
Next_InjectionA_End_Angle	°CA	The next injection A end angle, selected among all input angles, according to the current cylinder angle.	Range: ±1080° Resolution: 0.06°
Next_InjectionB_Start_Angle	°CA	The next injection B start angle, selected among all input angles, according to the current cylinder angle.	Range: ±1080° Resolution: 0.06°
Next_InjectionB_End_Angle	°CA	The next injection B end angle, selected among all input angles, according to the current cylinder angle.	Range: ±1080° Resolution: 0.06°
Ign_Start_Angles_Invalid	-	Vector of flags indicating if the corresponding ignition start angle is invalid (according to the angle input rules).	0 1

Ign_End_Angles_Invalid	-	Vector of flags indicating if the corresponding ignition end angle is invalid (according to the angle input rules).	0 1
InjA_Start_Angles_Invalid	-	Vector of flags indicating if the corresponding injection A start angle is invalid (according to the angle input rules).	0 1
InjA_End_Angles_Invalid	-	Vector of flags indicating if the corresponding injection A end angle is invalid (according to the angle input rules).	0 1
InjB_Start_Angles_Invalid	-	Vector of flags indicating if the corresponding injection Bstart angle is invalid (according to the angle input rules).	0 1
InjB_End_Angles_Invalid	-	Vector of flags indicating if the corresponding injection Bend angle is invalid (according to the angle input rules).	0 1

INJECTION_IGNITION_CORE

Objective

The injection & ignition core generates injection and ignition pulses according to the calculated next start and end angles. If injection and ignition pulses overlap, the pulses can be merged or removed, according to the user's choice.



For detection of start or end of injection pulses, it is required that the next pulse angles cross the cylinder angle for at least one FPGA clock period. This is illustrated in the figure below. If the angle serializer is used for the next angle inputs, all angle inputs (including cylinder angle) can directly be connected without any delays to be inserted.

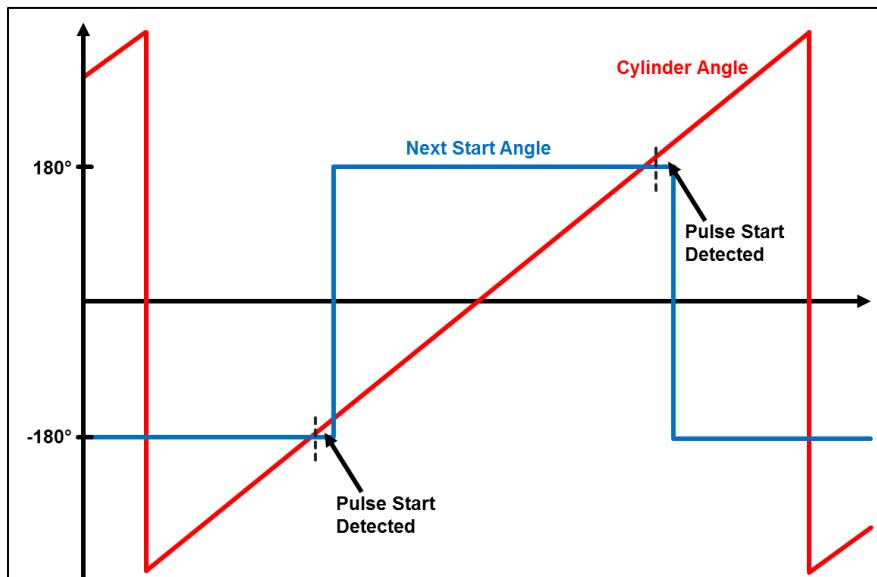


Figure 19: Example for valid next start or end angle input (start angles at 180° and -180° in this case)

Content

The blockset contains the following elements:

- Processor Interface: INJECTION_IGNITION_CORE_out (Processor Interface)
- FPGA Interface: INJECTION_IGNITION_CORE_in (FPGA Interface)
- FPGA: INJECTION_IGNITION_CORE (FPGA Main Component)

Processor Output

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

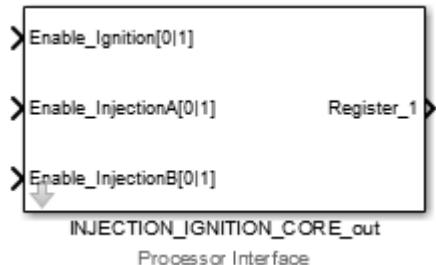


Figure 20: INJECTION_IGNITION_CORE_out block

Block Dialog	The processor output block provides the following dialog:
---------------------	---

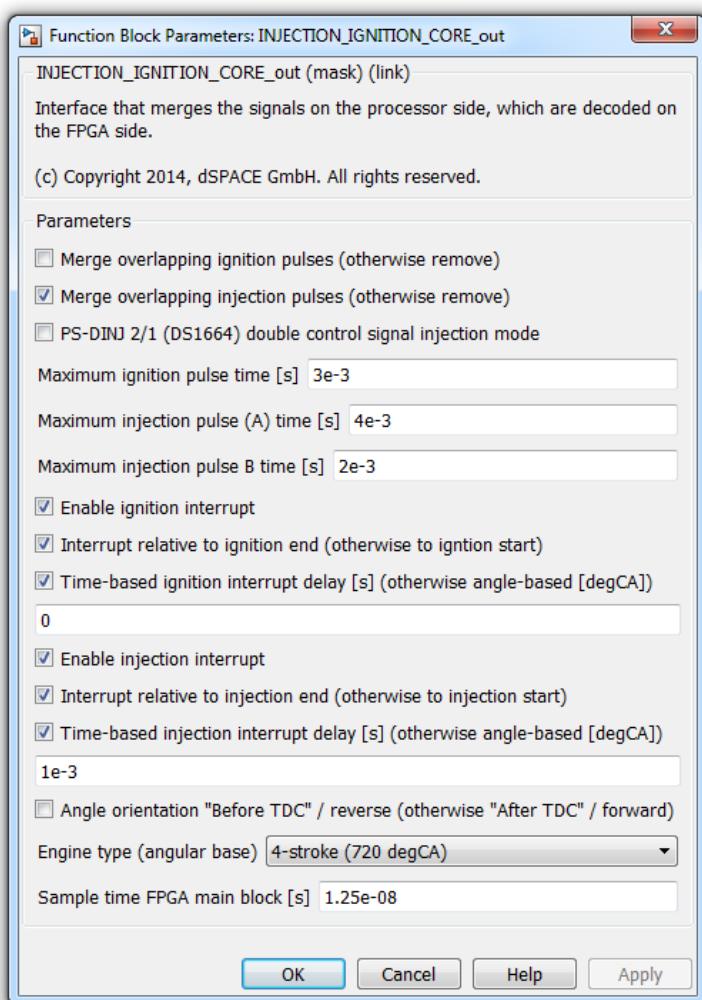


Figure 21: INJECTION_IGNITION_CORE_out dialog

The block dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Merge overlapping ignition pulses	-	If this option is set, overlapping ignition pulses will be merged. Otherwise overlapping pulses will be removed.	on / off
Merge overlapping injection pulses	-	If this option is set, overlapping ignition pulses will be merged. Otherwise overlapping pulses will be removed.	on / off
PS-DINJ 2/1 (DS1664) double control signal injection mode	-	If this option is set, the logic for removing or merging injection pulses is adapted to the double control signal logic of the DS1664 RapidPro direct injection modules.	on / off
Maximum ignition pulse time	s	Safety function to disable ignition pulses after a specified maximum duration.	Range: 0...17.17s Resolution: 8ns (*)
Maximum injection (A) pulse time	s	Safety function to disable injection A pulses after a specified maximum duration. In case of DS1664 double control signal mode this parameter specifies the maximum total duration of injection pulses.	Range: 0...17.17s Resolution: 8ns (*)
Maximum injection B pulse time	s	Safety function to disable injection B pulses after a specified maximum duration.	Range: 0...17.17s Resolution: 8ns (*)
Enable ignition interrupt		If this option is set, an interrupt will be triggered upon generated ignition pulses.	on / off
Interrupt relative to ignition end	-	If this option is set, the ignition interrupt is triggered after the end of the ignition pulse (and a specified delay). Otherwise it is triggered after the start of the ignition pulse.	on / off
Time based ignition interrupt delay	-	If this option is set, an interrupt time-based delay relative to the ignition pulse start or end can be specified. Otherwise, an angle-based delay can be specified.	on / off

Ignition interrupt delay	s °CA	The value of the ignition interrupt delay (either angle or time) can be specified here.	Range: 0°...1080° 0...0.13s(*) Resolution: 0.004° 8ns(*)
Enable injection interrupt		If this option is set, an interrupt will be triggered upon generated injection pulses. In normal operation the interrupt is referenced to injection signal A, in DS1664 double control signal mode, the interrupt is referenced to the whole injection pulse.	on / off
Interrupt relative to injection end	-	If this option is set, the ignition interrupt is triggered after the end of the injection (A) pulse (and a specified delay). Otherwise it is triggered after the start of the injection (A) pulse.	on / off
Time based injection interrupt delay	-	If this option is set, an interrupt time-based delay relative to the injection pulse start or end can be specified. Otherwise, an angle-based delay can be specified.	on / off
Injection interrupt delay	s °CA	The value of the injection interrupt delay (either angle or time) can be specified here.	Range: 0°...1080° 0...0.13s(*) Resolution: 0.004° 8ns(*)
Angle orientation "Before TDC"	-	If this option is checked the direction of the cylinder angles is reverse to the direction of the engine angle.	on / off
Engine type (angular base)	-	The engine type can be specified here. The angle ranges depend on the engine type (360°/720°/1080°)	2-stroke (360°CA) 4-stroke (720°CA) 6-stroke (1080°CA)
Sample time FPGA main block	s	The sample time the FPGA main component is clocked with. Usually this is the FPGA clock rate, however in case of downsampling of the FPGA component this parameter has to be adapted.	-

(*) Assuming FPGA sample time of 8ns. For other sample times, the ranges and resolutions scale accordingly.

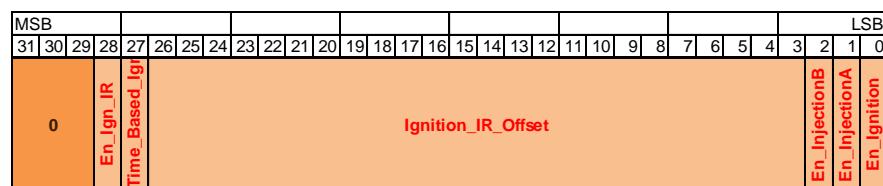
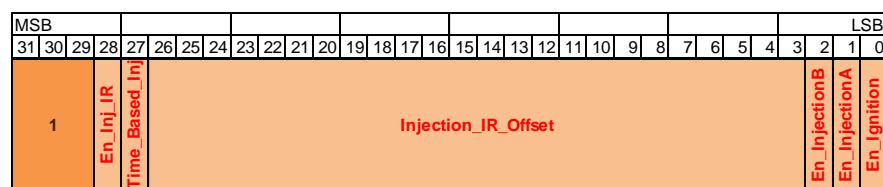
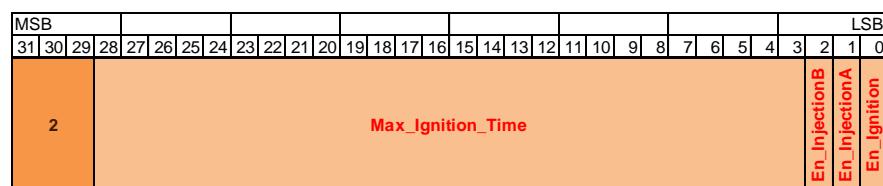
Input

The INJECTION_IGNITION_CORE_out block has the following inputs:

Name	Unit	Description	Range
Enable_Ignition	-	The generation of ignition pulses can be activated or deactivated during runtime using this input	0 1
Enable_InjectionA	-	The generation of injection A pulses can be activated or deactivated during runtime using this input	0 1
Enable_InjectionB	-	The generation of injection B pulses can be activated or deactivated during runtime using this input	0 1

Output

The Processor Out block outputs to 6 register contents, mapped to 1 register. The mapping is shown below.

Register 1.1**Register 1.2****Register 1.3****Register 1.4**

MSB	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	LSB	
3																																En_InjectionB	En_InjectionA	En_Ignition

Register 1.5

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
4		Max_InjectionB_Time																									En_InjectionB	En_InjectionA	En_Ignition		

Register 1.6

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
5					InjB_Max[30..26]	InjA_Max[30..26]	Ign_Max[30..26]								Before_TDC	Inj_End_IR	Ign_End_IR	MergeInjection	Merge_Ignition	DS1664_Injecti	Engine_Type	Enable(Reset)	En_InjectionB	En_InjectionA	En_Ignition						

FPGA Main Component

Block

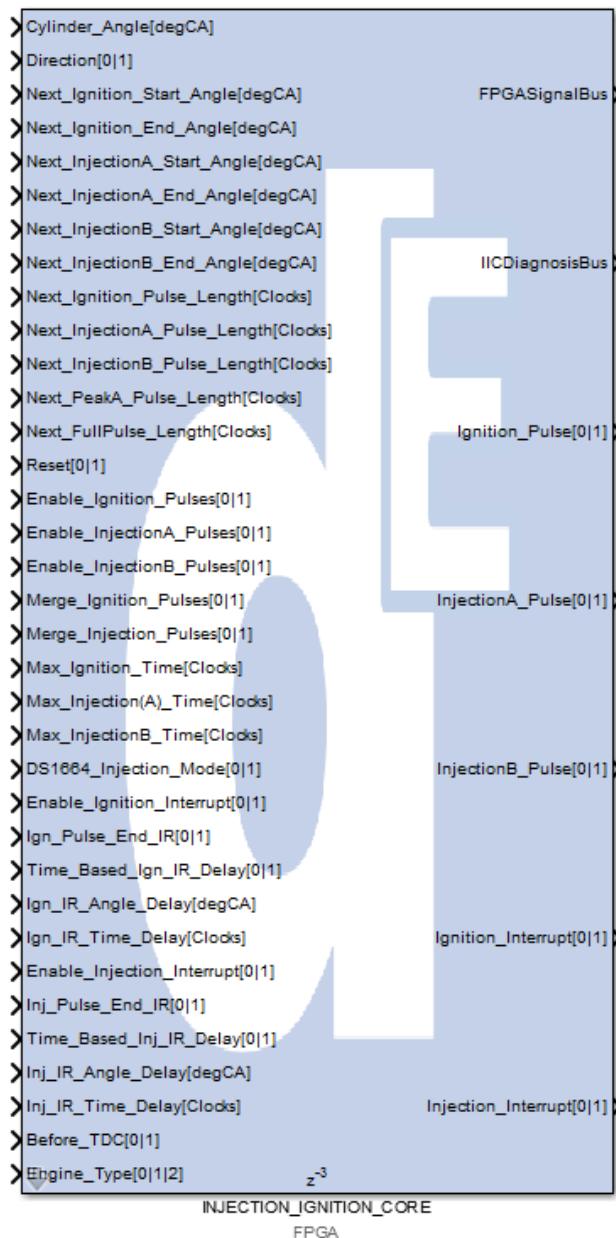


Figure 22: INJECTION_IGNITION_CORE FPGA Main block

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Cylinder_Angle	°CA	The cylinder angle derived from the engine angle and the TDC.	Fix_20_8
Direction	-	The direction of the engine motion.	Bool
Engine_Speed	rpm	The engine speed calculated by the ACU. Only required for time based injection or ignition pulses.	UFix_18_4
Next_Ignition_Start_Angle	°CA	The start angle of the next ignition pulse.	Fix_20_8
Next_Ignition_End_Angle	°CA	The end angle of the next ignition pulse.	Fix_20_8
Next_InjectionA_Start_Angle	°CA	The start angle of the next injection A pulse.	Fix_20_8
Next_InjectionA_End_Angle	°CA	The end angle of the next injection A pulse.	Fix_20_8
Next_InjectionB_Start_Angle	°CA	The start angle of the next injection B pulse.	Fix_20_8
Next_InjectionB_End_Angle	°CA	The end angle of the next injection B pulse.	Fix_20_8
Next_Ignition_Pulse_Length	Clocks	The pulse length of the next ignition pulse.	UFix_32_0
Next_InjectionA_Pulse_Length	Clocks	The pulse length of the next injection A pulse.	UFix_32_0
Next_InjectionB_Pulse_Length	Clocks	The pulse length of the next injection B pulse.	UFix_32_0
Next_PeakA_Pulse_Length	Clocks	The pulse length of the next PeakA pulse in double control signal injection mode.	UFix_32_0
Next_FullPulse_Length	Clocks	The pulse length of the next double control signal (time from start of pulse A to end of pulse B) in double control signal injection mode.	UFix_32_0
Reset	-	Resets the injection and ignition pulse generation.	Bool
Enable_Ignition_Pulses	-	Enables or disables the generation of ignition pulses during runtime.	Bool
Enable_InjectionA_Pulses	-	Enables or disables the generation of injection A pulses during runtime.	Bool

Enable_InjectionB_Pulses	-	Enables or disables the generation of injection B pulses during runtime.	Bool
Merge_Ignition_Pulses	-	Activates the merging of overlapping ignition pulses (otherwise overlapping pulses will be removed).	Bool
Merge_Injection_Pulses	-	Activates the merging of overlapping injection pulses (otherwise overlapping pulses will be removed).	Bool
Max_Ignition_Time	Clocks	The maximum valid ignition pulse duration. Pulses will be stopped after exceeding this time.	UFix_31_0
Max_Injection(A)_Time	Clocks	The maximum valid injection A pulse duration (or total injection pulse duration in case of DS1664 double control signal injection mode). Pulses will be stopped after exceeding this time.	UFix_31_0
Max_InjectionB_Time	Clocks	The maximum valid injection B pulse duration. Pulses will be stopped after exceeding this time. This port is unused in case of DS1664 double control signal injection mode.	UFix_31_0
DS1664_Injection_Mode	-	Activates the PS-DINJ 2/1 double control signal injection mode.	Bool
Enable_Ignition_Interrupt	-	Enables the generation of ignition interrupts.	Bool
Ign_Pulse_End_IR	-	Ignition interrupt is relative to pulse end (otherwise pulse start)	Bool
Time_Based_Ign_IR_Delay	-	The delay of the ignition interrupt is set as a time in s.	Bool
Ign_IR_Angle_Delay	°CA	The time delay for the ignition interrupt generation.	Fix_20_8

Ign_IR_Time_Delay	s	The angle delay for the ignition interrupt generation.	UFix_20_20
Enable_Injeciton Interrupt	-	Enables the generation of injection interrupts.	Bool
Inj_Pulse_End_IR	-	Injection interrupt is relative to pulse end (otherwise pulse start)	Bool
Time_Based_Inj_IR_Delay	-	The delay of the injection interrupt is set as a time in s.	Bool
Inj_IR_Angle_Delay	°CA	The time delay for the injection interrupt generation.	Fix_20_8
Inj_IR_Time_Delay	s	The angle delay for the injection interrupt generation.	UFix_20_20
Before_TDC	-	If this option is checked the direction of the cylinder angles is reverse to the direction of the engine angle.	Bool
Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
IICDiagnosisBus	-	Contains error signals and internal flags (see IICDiagnosisBus)	Bus
Ignition_Pulse	-	The generated ignition pulses.	Bool
InjectionA_Pulse	-	The generated injection A pulses.	Bool
InjectionB_Pulse	-	The generated injection B pulses.	Bool
Ignition Interrupt	-	The generated ignition interrupt impulse.	Bool
Injection Interrupt	-	The generated injection interrupt impulse.	Bool

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	564	0.14%
LUTs	1439	0.71%
Block RAM	0	0.00%
DSP Slices	0	0.00%

Angle Measurement

Objective

The *Angle Measurement* section contains the processing of crank/cam signals to determine engine angle and speed (angular computation unit) as well as the processing of incremental encoders.

Features

The *Angle Measurement* components provide the following main features:

- Processing of crank- and camshaft sensors with up to 360 teeth on crank
- Up to 4 digital camshaft signals
- Digital (Hall) or analog (inductive) crankshaft signal
- Crank/cam pattern configurable by the user
- Reverse crank supported, e.g. for start/stop
- 0.1° resolution for crank/cam-based ACU (angular processing unit)
- Incremental encoder processing for even finer angular resolutions
- Measurement of four camshaft phase shift values.

Content

The section contains the following components:

- ACU
- Engine Encoder

ACU

Objective

Processes crank/cam inputs and determines engine position, speed as well as cam phase shift angles. The ACU core is provided as a pre-synthesized netlist (IP-Core).



Offline simulation is not supported for the ACU. To get a simulated angle for offline simulation, use the APU from the XSG Utils library.



At the current state, downsampling is not supported for the ACU. Support of downsampling as planned for future releases.

Principle of Operation

The main task of the Angular Computation Unit (ACU) is the computation of the crank angle, based on the available crank and cam signals. As the crank and cam signals contain only limited angular resolution, the angular resolution of the crank angle is enhanced by means of interpolation and state estimation inside of the ACU unit. By the engine start, the angular position is not known. The position recognition and synchronization during engine start are further very important tasks covered by the ACU component. After the engine position is recognized and the synchronization is established, its position will be continuously tracked and high-resolution position and angular velocity information becomes available by the ACU unit. With this available information the cam phase shift angles are also computed and put out by the ACU. An overview of the functionality of the ACU component is displayed in Figure 23.

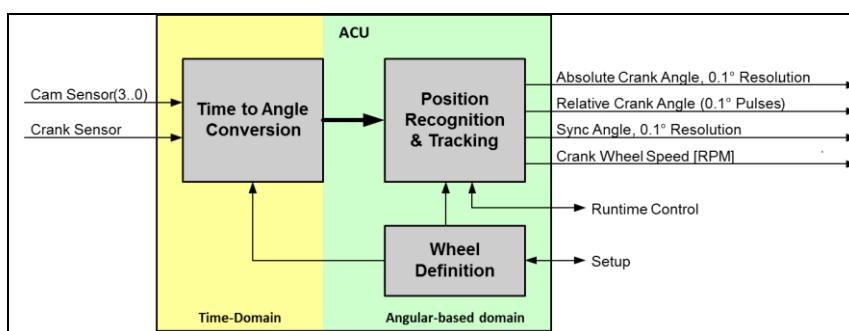


Figure 23: Simplified functional overview of the Angular Computation Unit

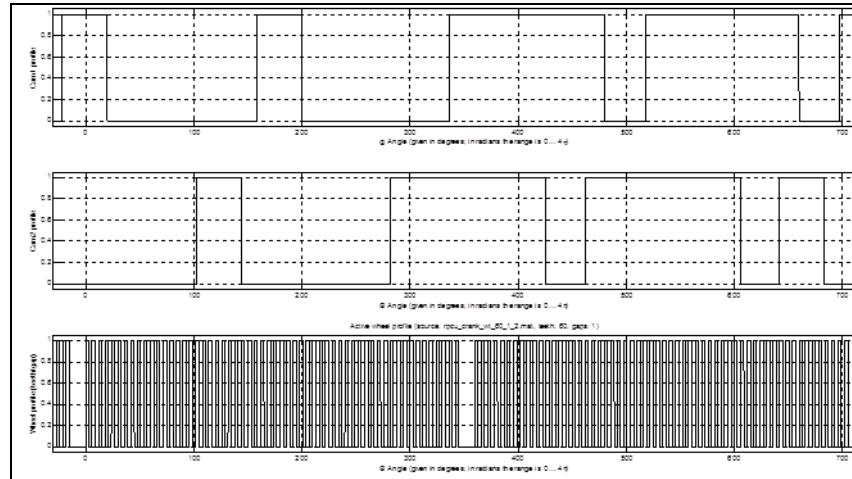


Figure 24: Crank/Cam wheel pattern

Content

The blockset contains the following elements:

- Processor Interface: ACU_out
(Processor Interface)
- FPGA Interface: ACU_in
(FPGA Interface)
- FPGA: ACU
(FPGA Main Component)
- FPGA Interface: ACU_out
(FPGA Interface)
- Processor Interface: ACU_in
(Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

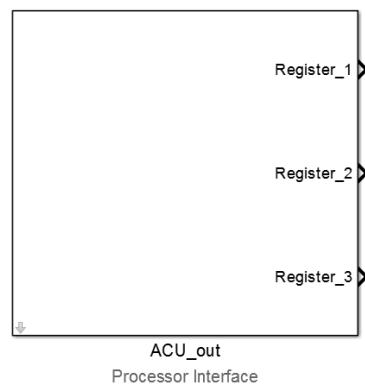


Figure 25: ACU_out block

General parameters tab The block dialog has the following general parameters:

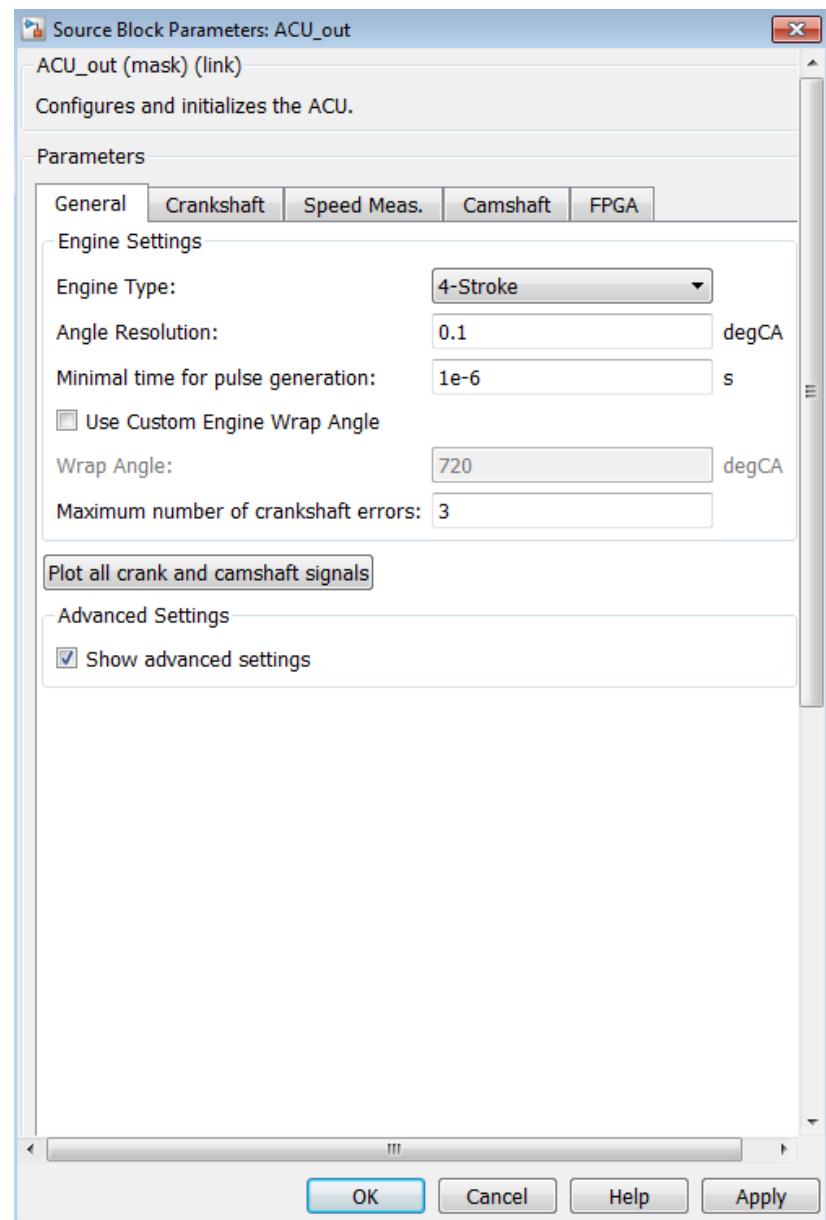


Figure 26: ACU_out dialog - General tab

Name	Unit	Description	Range/Resolution
Engine Type	-	The engine type and corresponding angular base of the engine cycle can be specified here.	2-stroke (360°) 4-stroke (720°) 6-stroke (1080°)
Angle Resolution	°CA	The angular resolution of the system. Currently this value is fixed to 0.1°.	
Minimal time for pulse generation (Advanced feature: Invisible by default)	s	Defines the minimal time between two adjacent 0.1° pulses. An appropriate minimum distance is essential to avoid TPU- and/or ADC overrun if the 0.1° pulse sequence is used for such purposes.	Range 0 ... 12.7875µs @ 80 MHz
Use Custom Engine Wrap Angle	°CA	If selected, a custom wrap angle can be specified for the engine angle. Otherwise the default wrap angle (e.g. 360° for 4-stroke engines) is applied.	Range: 0°...1080° Resolution: 0.004°
Maximum number of crankshaft errors		When more crankshaft signal errors are detected in a 720° interval than specified by this parameter, the engine control leaves the synchronized state and starts a resynchronization.	Range: 0 ...255 Resolution: 1
Show advanced settings		If enabled the advanced settings parameters are shown in their respective tabs. By default all of the advanced settings parameters are hidden.	On Off

Plot all crank and camshaft signals

The button **Plot all crank and camshaft signals** creates a figure containing the actual crank and cam configuration, taking into account the selected start angle (offset) of the crankshaft. For better overview, the base range of the cam (0° - 720°) is a little bit extended, as shown in the figure below. This function recognizes the number of configured cams and creates the figure accordingly. The example shown in the figure below contains two activated cams.

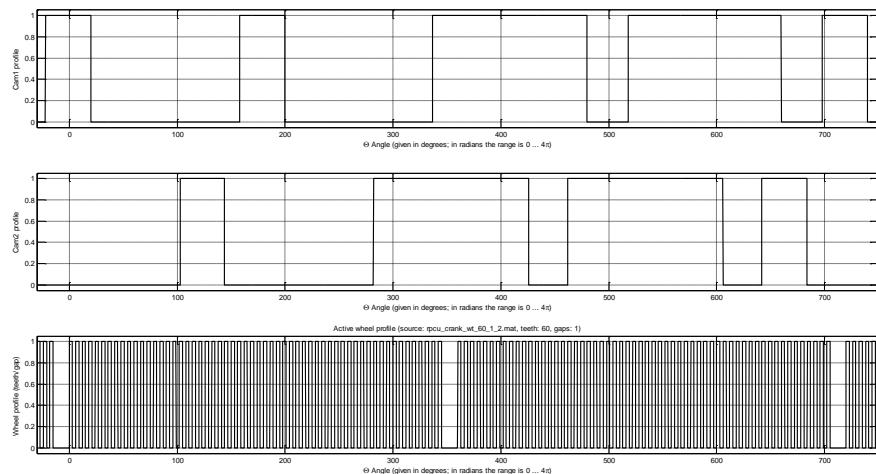


Figure 27: Example of figure created using the *Plot all* button



The information about engine and cylinder wrap angles are required by several components. As the wrap angles are usually the same for all components, it is recommended to use a variable for these parameters.

Crankshaft parameters tab

The block dialog has the following parameters:

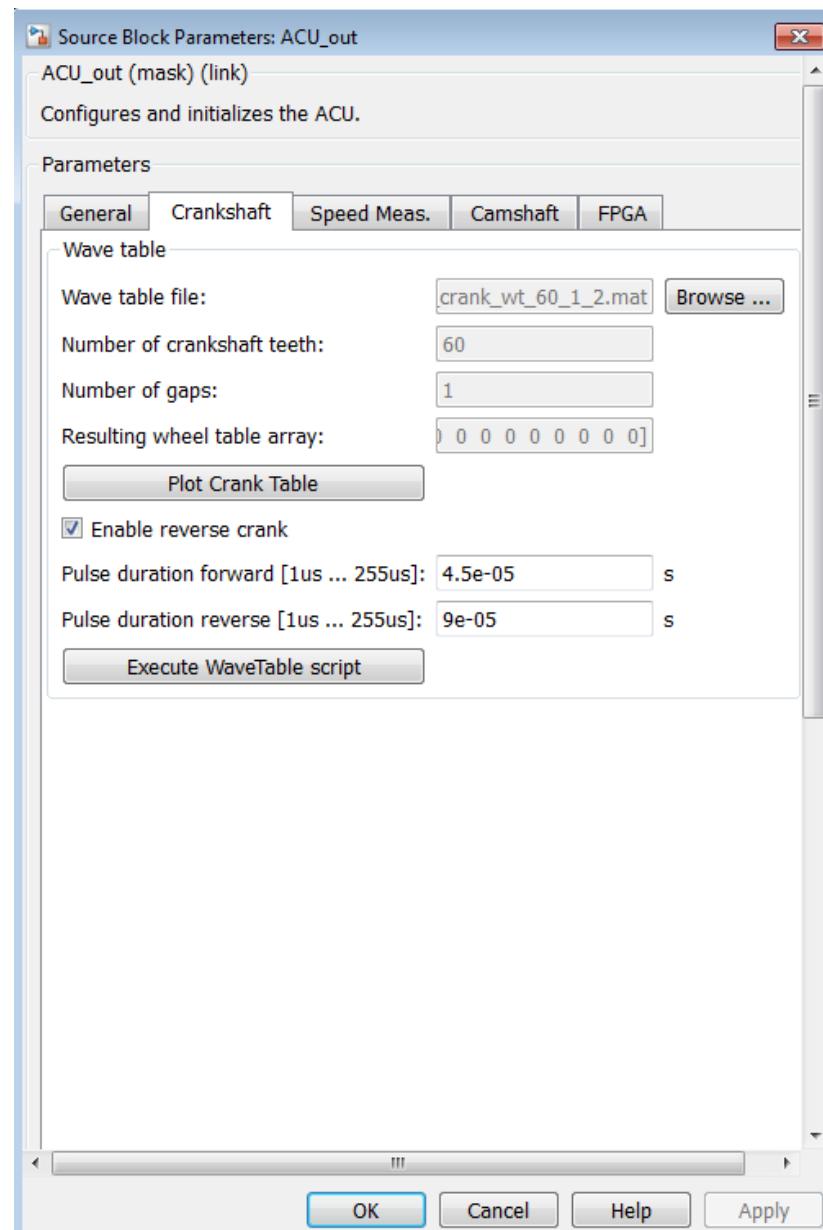


Figure 28: ACU_out dialog - Crankshaft tab

Name	Unit	Description	Range/Resolution
Enable reverse crank	-	Enable / disable reverse crank detection.	On Off
Pulse duration forward	s	Crank sensor pulse duration forward.	Range:1μs...255μs Resolution: 1μs
Pulse duration reverse	s	Crank sensor pulse duration backward.	Range:1μs...255μs Resolution: 1μs

Wave table

Using the button **[Browse ...]** a *.mat file containing the wheel table description can be loaded. After the file is loaded, basic plausibility checks are performed like the check for appropriate variables and check of the wheel table size. In order to be loadable, the *.mat file must contain the structure variable **crank** having following mandatory fields:

Data (vector of 7200 elements)

NumOfGaps (scalar value)

NumOfTeeth (scalar value)

The vector **Data** must contain the wheel table in range of 0 ... 360° with table data resolution of 0.05°. Allowed element values of **Data** are 0 and 1. The variable **NumOfGaps** indicates the number of gaps whereas the element **NumOfTeeth** indicates the number of teeth in the crankshaft. These two variables must be in agreement with the wheel table stored in Data.



The first element in the wheel table is automatically assumed as the global zero (0)-degree position. If you need to change this, please use the parameter *start angle*.

After the wheel table is successfully loaded and interpreted, the information about the wheel table status is updated in the associated text field of the dialog, namely *Wave table file*, *Number of crankshaft teeth*, *Number of gaps* and *Resulting wheel table array*. All of those fields are read only. To visually check the loaded wheel table, the button **[Plot Crank Table]** may be used as shown in the figure below.

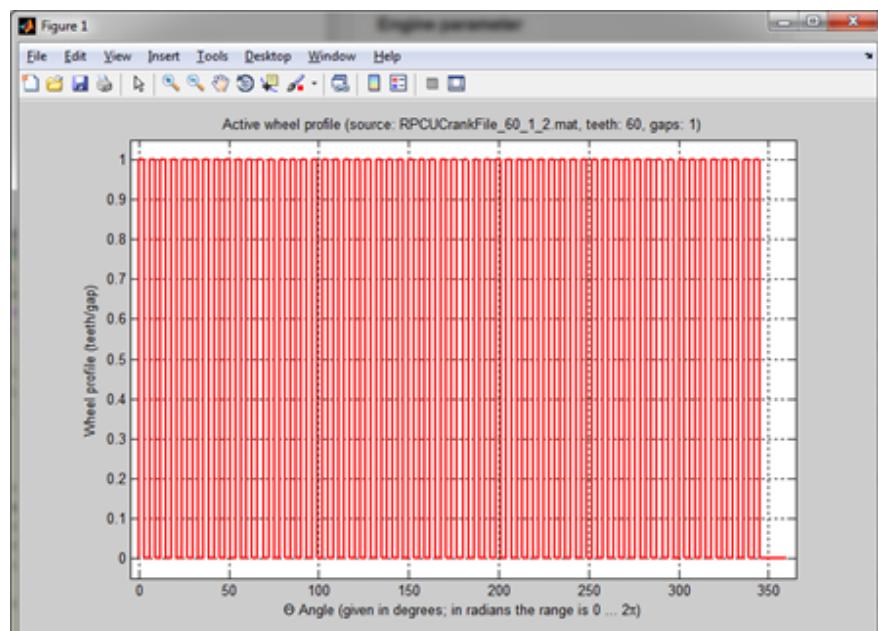


Figure 29: Plot of the wheel table created

For creation of an own specific wheel table file the example Matlab script `xsg_engcon_crank_wt_create` can be used. The script can be executed with the button **Execute WaveTable script**.

Speed Measurement tab

The block dialog has the following parameters:

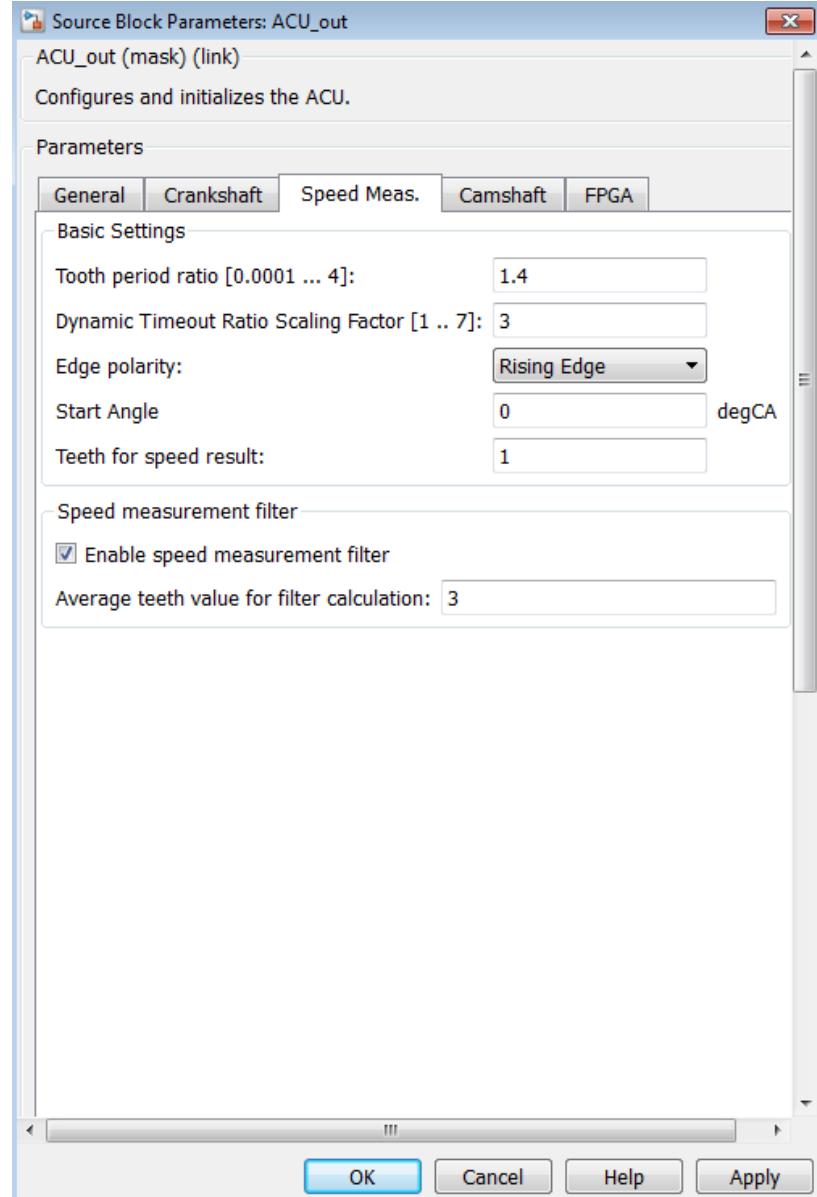


Figure 30: ACU_out dialog - Speed Measurement tab

Name	Unit	Description	Range/Resolution
Tooth period ratio		Configures the tooth period ratio. Enables to specify the region of valid tooth periods. Tooth periods are valid if they are either smaller than "PreviousPeriod" *	Range: 0.00001...4

		PeriodRatio, or greater than "PreviousPeriod" / PeriodRatio.	
Dynamic Timeout Ratio Scaling Factor (Advanced feature: Invisible by default)		Enables the user to set a scaling factor on when the ACU is detecting a crank timeout. The recommended range is from 2 to 4 and the default value is 3.	Range: 1 ... 7 Resolution: 1
Edge polarity	-	The matching edge of the crankshaft signal. Possible values are rising edge or falling edge.	Rising edge Falling edge
Start angle	$^{\circ}$ CA	Defines the offset of the angle counter. Requires matching settings in camshaft vector.	
Teeth for speed result	-	The number of crankshaft teeth after which a new average speed measurement result is calculated.	
Enable Speed measurement filter		Choose if speed measurement filter is enabled, and if yes, let the user specify the number of crankshaft teeth used for calculation of the average speed.	On Off

Camshaft parameters tab

The 4 cam configuration sections contain the following parameters:

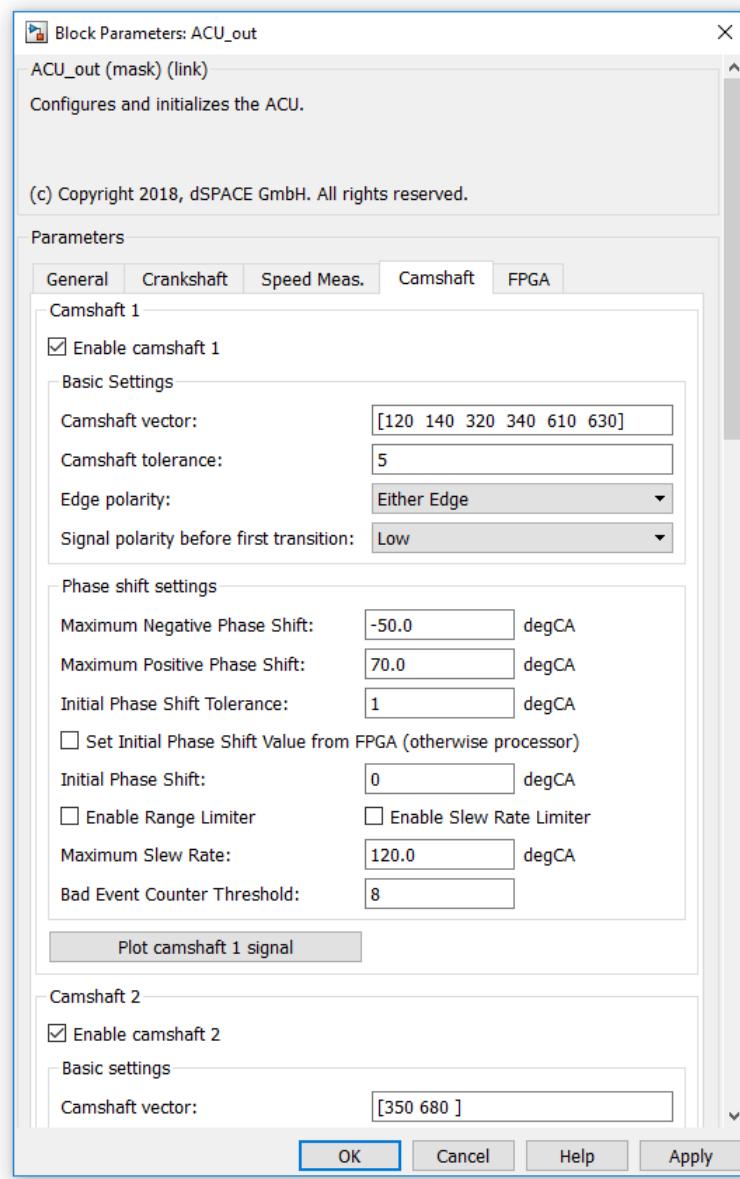


Figure 31: ACU_out dialog - Camshaft tab

Name	Unit	Description	Range/Resolution
Enable camshaft x	-	Enable or disable the use of Camshaft 1,2,3 or 4. When disabled, all other fields of the appropriate Camshaft are not enabled.	On Off
Camshaft vector	-	Camshaft vector. Corresponding to the <i>Edge polarity</i> parameter each transition must be specified in this vector. Angle values must be entered in ascending order.	Range: 0°...719.9° Resolution: 0.1°
Camshaft tolerance	°CA	Defines the standard tolerance of all angles specified in <i>Camshaft vector</i> .	Range: 0°...180° Resolution: 0.1°
Edge polarity	-	Specifies the edge polarity. Falling Edge: detects falling edges. Rising Edge: detects rising edges. Either Edge: detects both rising and falling edges.	Rising Edge Falling Edge Either Edge
Signal polarity before first transition	-	Specifies the signal polarity before the first transition. Low: camshaft signal starts low. High: camshaft signal starts high.	Low High
Maximum negative phase shift	°CA	Specifies the maximum allowed negative phase shift value. This value has to be smaller than the <i>Maximum positive phase shift</i> value.	Range: -120.0° ... 120.0° Resolution: 0.1°
Maximum positive phase shift	°CA	Specifies the maximum allowed positive phase shift value. This value has to be smaller than the <i>Maximum negative phase shift</i> value.	Range: -120.0° ... 120.0° Resolution: 0.1°
Enable Range Limiter (Advanced feature: Invisible by default)		Enables the Phase Range Limiter. If enabled the range limiter performs a clipping of the phase measurements values. If enabled, undesirable phase range violations will be avoided.	On Off

Enable Slew Rate Limiter (Advanced feature: Invisible by default)		Enables the Phase Slew Rate Limiter. If enabled the slew rate limiter performs a limitation of the slew rate of the phase measurements values. The slew rate is defined as the difference between two consecutive phase measurement values.	On Off
Maximum Slew Rate (Advanced feature: Invisible by default)	°CA	Defines the maximal allowed slew rate.	Range: 0.0° ... 240.0 ° Resolution: 0.1°
Bad Event Counter Threshold (Advanced feature: Invisible by default)		Defines how many bad events can occur in the phase measurement unit until it is reset. If the value is set to 0, no automatic reset will be performed.	Range: 0 ... 15 Resolution: 1
Initial Phase Shift Tolerance (Advanced feature: Invisible by default)	°CA	Defines the tolerance of the initial phase shift. Small values increase the possibility of fast synchronization.	Range 0.0° ... 120.0° Resolution: 0.1°
Initial Phase Shift (Advanced feature: Invisible by default)	°CA	Defines the initial camshaft phase shift, so the ACU is able to synchronize even if the camshafts are shifted.	Range: -120.0° ... 120.0° Resolution: 0.1°
Set Initial Phase Shift value from FPGA (Advanced feature: Invisible by default)		If enabled the initial phase shift value can be set by the FPGA application. Disabled the definition in this block GUI is taken.	On Off

Using the button **Plot Camshaft x signal**, a wave table diagram of the entered cam data for cam x can be shown.

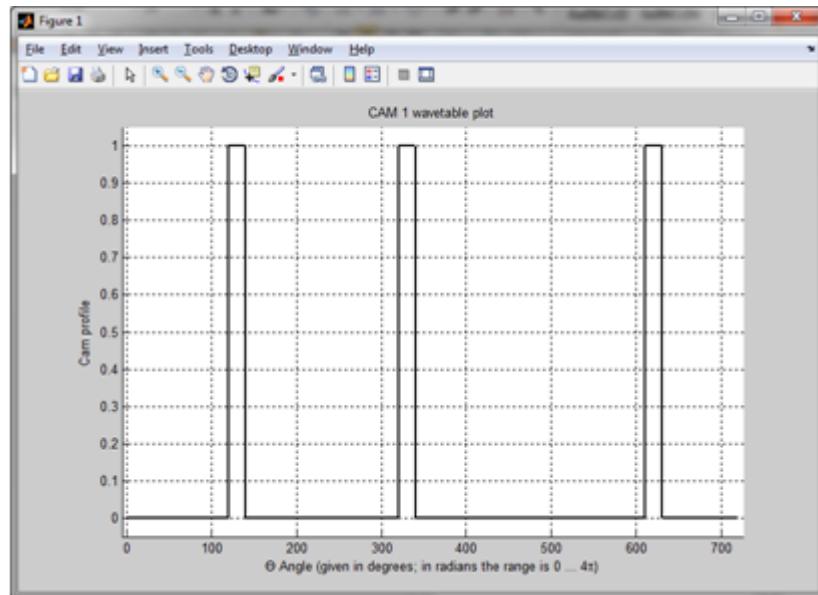


Figure 32: Plot of the cam1 wave table

Camshaft phase shift measurement

This feature measures the phase shift of the real camshaft signal to the defined one in the ACU. The ACU has to be synced to calculate and output a phase shift value. The following formula shows the way the phase shift is calculated by the ACU:

$$\Phi = \text{Angle}_{\text{Meas}} - \text{Angle}_{\text{Def}}$$

Where Φ is the calculated phase shift, $\text{Angle}_{\text{Meas}}$ the measured camshaft angle and $\text{Angle}_{\text{Def}}$ the defined camshaft angle. Below figure shows how a positive phase shift may look like:

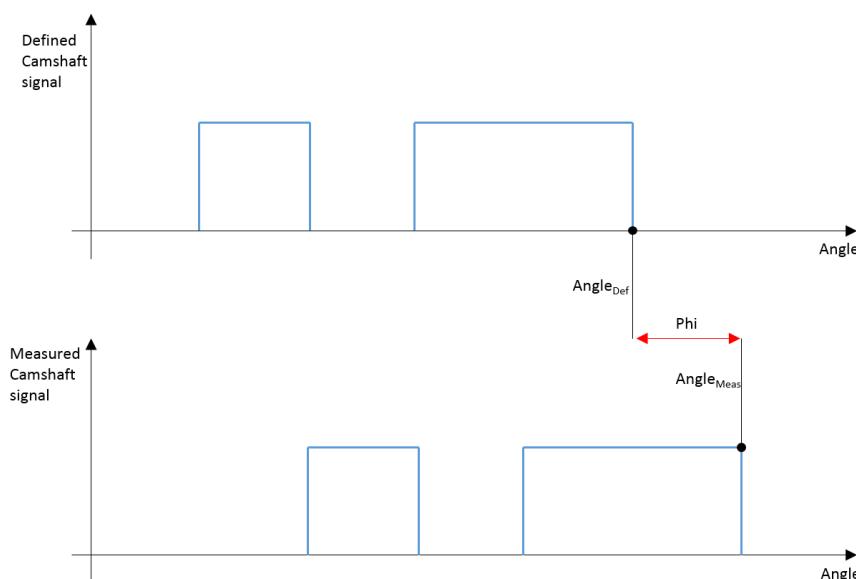


Figure 33 Definition of phase shift calculation

Interface tab

The interface tab has the following dialog options:

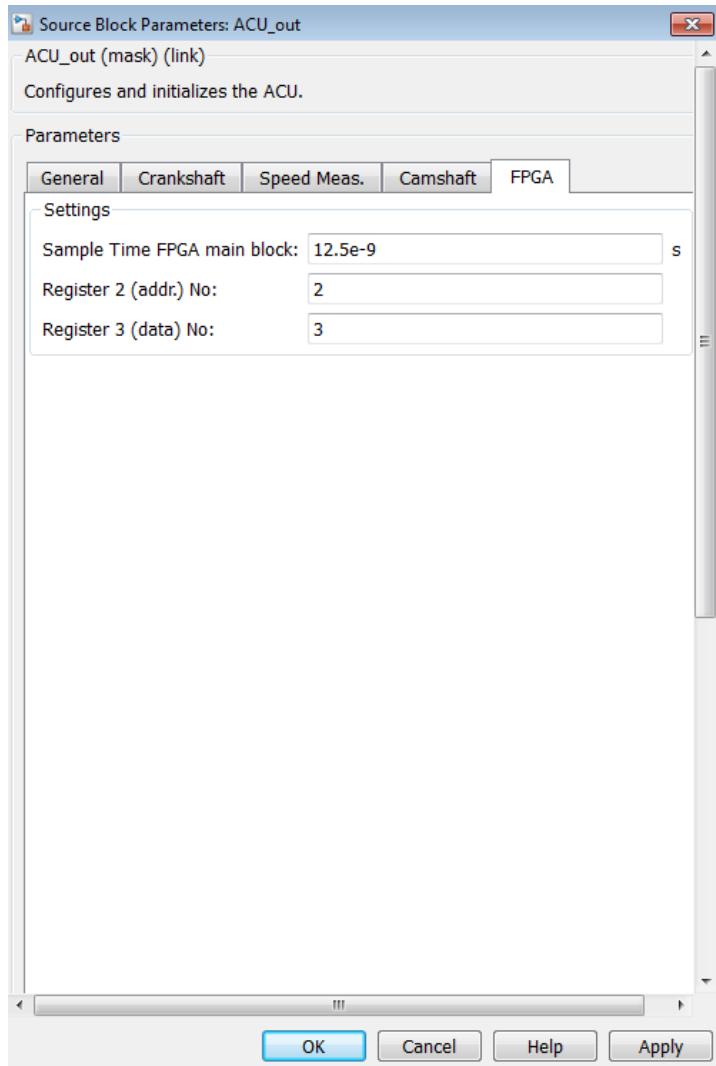


Figure 34: ACU_out dialog - Interface tab

Name	Unit	Description	Range/Resolution
Sample time FPGA main block	s	The sample time the FPGA main component is clocked with. Usually this is the FPGA clock rate, however in case of downsampling of the FPGA component this parameter has to be adapted.	
Register 2 (addr) no.	-	When modelling the interface, the ACU_out block contains 3 register outputs which are to be connected to registers of the RTI FPGA Programming Blockset. After initializing, the	Range: 1...256 Resolution: 1

		<p>channel numbers of the 2nd and 3rd register are stored in the ACU_out_block and the corresponding output registers will be removed. The reason is that these registers are not used in cyclic operation, but for initializing the ACU using a dedicated software function.</p> <p>After the register blocks have been automatically removed, this parameter will appear in the block mask, enabling the user to propagate manual interface changes on the FPGA side to the processor later on.</p>	
Register 3 (data) no.	-	See Register 2 (addr) no.	Range: 1...256 Resolution: 1

Range Checks

Basic range checks are performed for the most parameter by the GUI when the **Apply** or the **OK** buttons are pressed.

When a range error is detected (e.g. the entered data is out of bounds), the background of the affected input is changed to red color and the user input is either replaced with a default value or rejected according to the *Error handling* setting. Depending on the *GUI mode* settings, additionally an error dialog may be triggered or a text entry into the Matlab command window can be generated.

	<p>It is necessary to close the ACU configuration GUI before closing of the associated Simulink model.</p> <p>It is also recommended to close the ACU configuration GUI before a RTW build or code generation is performed.</p> <p>Disregarding this may lead to a lost block reference, error message and lost settings. If such an error is encountered, the only way to close the GUI is by pressing the <i>Cancel</i> button. In this case all not yet applied settings will be lost.</p>
---	---

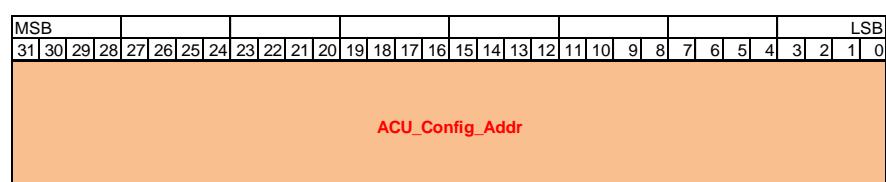
Output

The Processor Out block outputs 3 registers. The mapping is shown below.

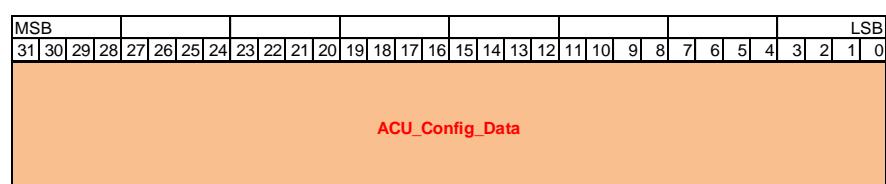
Register 1.1



Register 2



Register 3



FPGA Main Component

Block

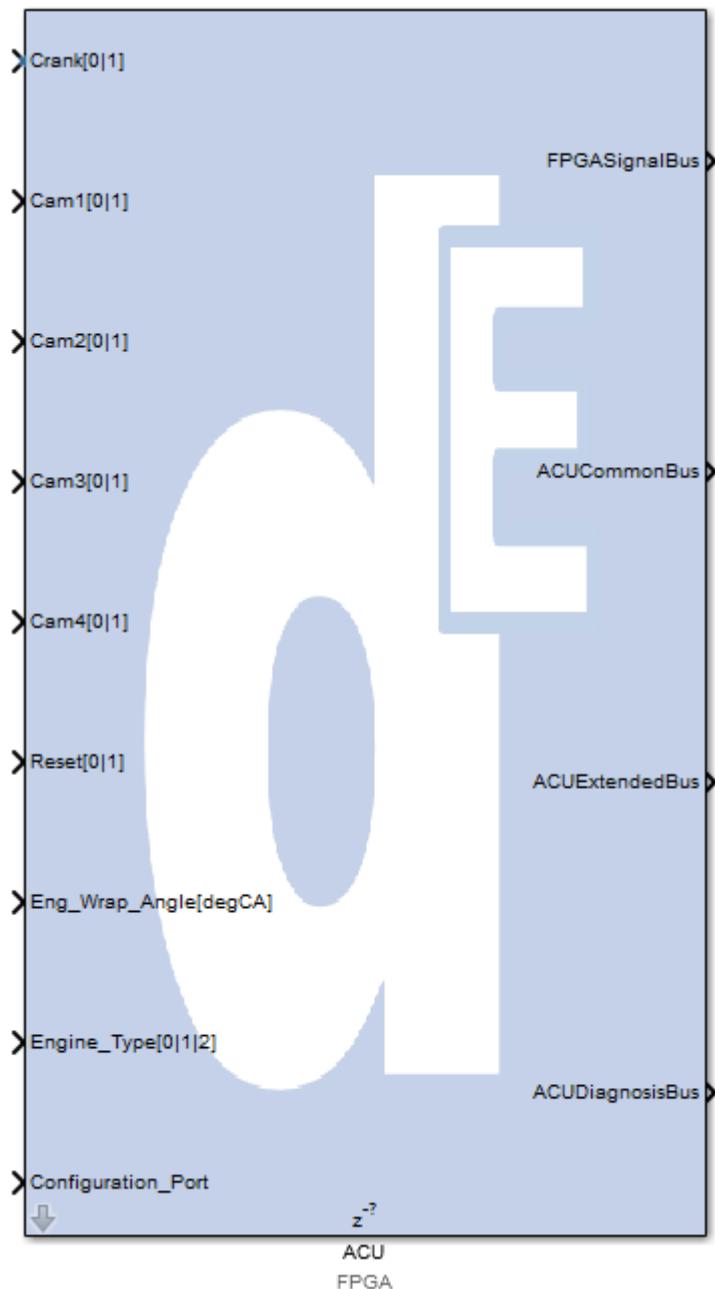


Figure 35: ACU FPGA Main block

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 this component interface generation with buffer access is not supported. Please use register access.** 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
Crank	-	The crank shaft input signal.	Bool
Cam1	-	The 1 st cam shaft input signal.	Bool
Cam2	-	The 2 nd cam shaft input signal.	Bool
Cam3	-	The 3 rd cam shaft input signal.	Bool
Cam4	-	The 4 th cam shaft input signal.	Bool
Reset	-	Resets the angle processing.	Bool
Eng_Wrap_Angle	-	The wrap angle the engine angle output.	UFix_19_8
Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0
Configuration_Port	-	Internal configuration port. Has to be connected to the ACU_in component.	UFix_66_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
ACUCommonBus	-	Contains a normed output signal bus equal for all angle-processing components (see ACUCommonBus).	Bus
ACUExtendedBus	-	Contains signals which are only available for the crank/cam based ACU (see ACUExtendedBus).	Bus
ACUDiagnosisBus	-	Contains additional debugging signals (see ACUDiagnosisBus).	Bus

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	9578	2.35%
LUTs	16976	8.33%
Block RAM	2	0.44%
DSP Slices	26	3.15%

Processor Input

Block Adapts the FPGA signals for the processor side.

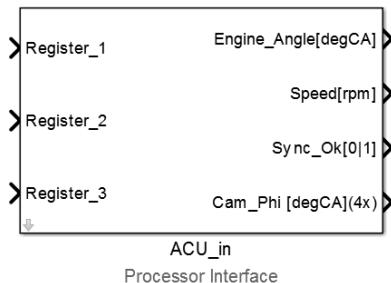


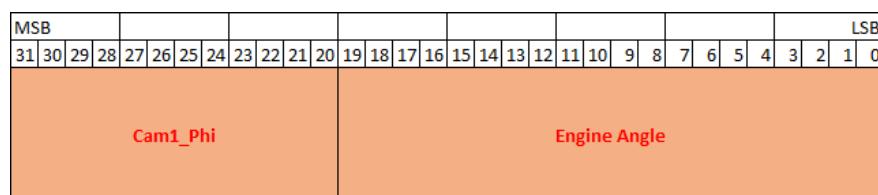
Figure 36: ACU_in block



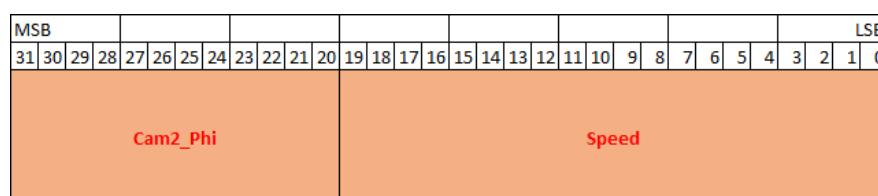
This block is only used to provide output signals to the processor application. If the signals are not required in the processor application, computation resources can be saved by not using this block.

Input The processor input block has 3 input register with a sectioning shown below.

Register 1



Register 2



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
ACU_Init_Done	ACU_Cam_Init_Error	Crank_Filt	Cam1_Filt	Cam2_Filt	Cam3_Filt	Cam4_Filt	Sync_Ok																								
Cam4_Phi												Cam3_Phi																			

Output

The ACU_in block has the following outputs:

Name	Unit	Description	Range
Position	°CA	The engine angle calculated by the FPGA main component.	Range: $\pm 720^\circ$ Resolution: 0.004°
Speed	rpm	The engine speed calculated by the FPGA main component.	Range: ± 20000 rpm Resolution: 0.06 rpm
Sync_Ok	-	The valid flag, which is raised as soon as the position is referenced (synchronization established).	0 1
Cam_Phi(4x)	°CA	All 4 camshaft phase shift measurement values calculated by the FPGA main component.	Range $\pm 120^\circ$ Resolution 0.1°



The lowest possible speed for the ACU to remain synchronization is FPGA clock time and crankshaft signal dependent. It is calculated with the following formula:

$$speed_{\min} = \frac{CrankshaftGap_{\max}}{2^{28} \cdot t_{FPGA} \cdot 6}$$

So for the DS1554 the lowest possible speed with a crankshaft with 60 teeth and a gap of two teeth would be:

$$speed_{\min} = \frac{18^\circ}{2^{28} \cdot 12.5 \text{ ns} \cdot 6}$$

$$speed_{\min} \approx 0.89 \text{ rpm}$$



Maximum possible speed of the ACU is 20,000 rpm. Please note that the minimal time for pulse generation should be set to at least 500 ns to achieve that speed. The parameter is an advanced setting of the *General* tab.

ENGINE_ENCODER

Objective	Processes the input of incremental encoders for combustion engines and provides engine angle and speed.
------------------	---

Principle of Operation	<p>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.</p> <p>The third track is the so-called index track which provides only one pulse per revolution, which enables determination of the absolute mechanical position. For engine control applications, special incremental encoders provide index tracks which are only passed once per combustion cycle (e.g. every 720° for 4-stroke engines).</p> <p>Prior to detection of the index pulse (worst case after one mechanical revolution) only the relative position and the velocity can be determined.</p> <p>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.</p>
-------------------------------	--

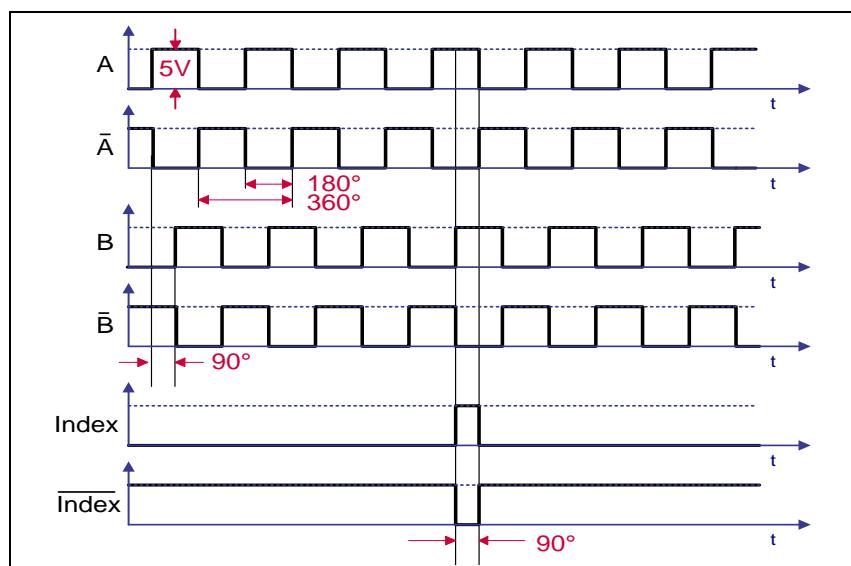


Figure 37: Incremental TTL encoder shape

Content

The blockset contains the following elements:

- Processor Interface: ENGINE_ENCODER_out
(Processor Interface)
- FPGA Interface: ENGINE_ENCODER_in
(FPGA Interface)
- FPGA: ENGINE_ENCODER
(FPGA Main Component)
- FPGA Interface: ENGINE_ENCODER_out
(FPGA Interface)
- Processor Interface: ENGINE_ENCODER_in
(Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

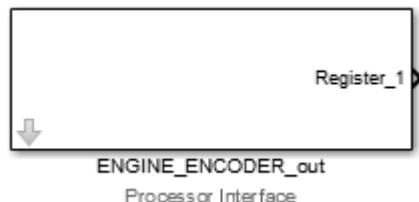


Figure 38: ENGINE_ENCODER_out block

Block Dialog

The processor output block provides the following dialog:

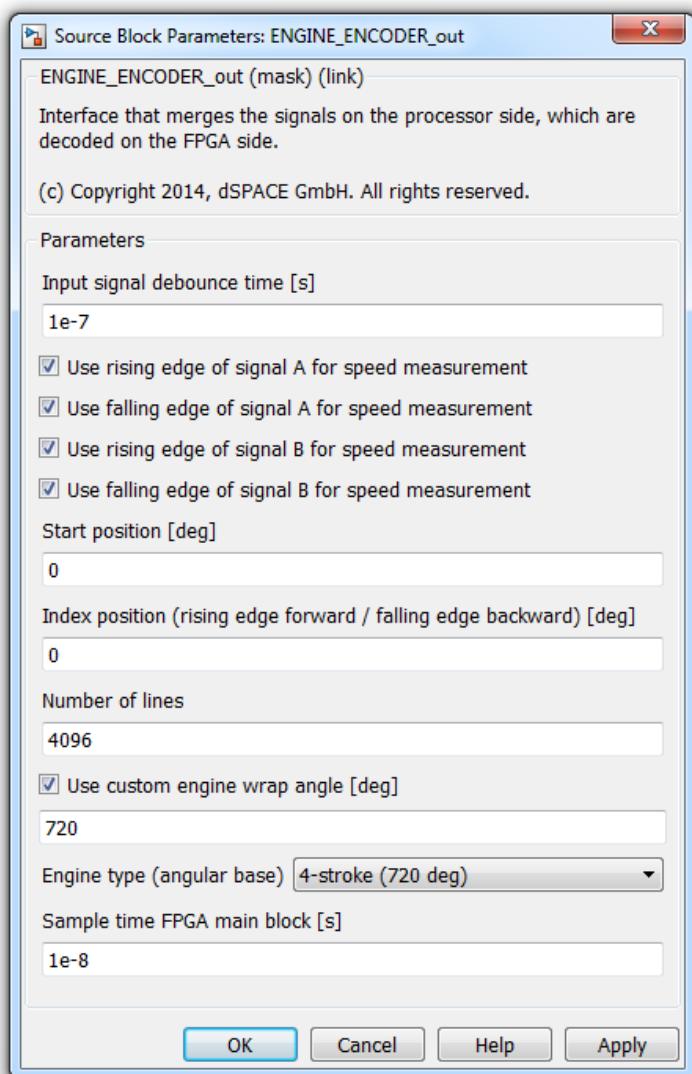


Figure 39: ENGINE_ENCODER_out dialog

The block dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Debounce Time	s	Debouncing for the incremental encoder input signals can optionally be enabled with a specified debounce time.	Range: 0...8.184µs Resolution: 8ns (*)
Use rising edge of signal A for speed measurement	-	Speed is always measured as the time between two identical edges. If this option is set, the time between two rising edges of input signal A is used for speed measurement.	on / off
Use falling edge of signal A for speed measurement	-	If this option is set, the time between two falling edges of input signal A is used for speed measurement.	on / off
Use rising edge of signal B for speed measurement	-	If this option is set, the time between two rising edges of input signal B is used for speed measurement.	on / off
Use falling edge of signal B for speed measurement	-	If this option is set, the time between two falling edges of input signal B is used for speed measurement.	on / off
Start angle	°CA	Specifies which start angle shall be assumed as long as the index (reference) has not been detected	Range: 0°...1080° Resolution: 0.004°
Index angle	°CA	Specifies the angle of the index (reference). The index is supposed to be passed once per engine cycle.	Range: 0°...1080° Resolution: 0.004°
No of Lines	-	Specifies the number of encoder lines per engine cycle.	Range: 4...inf Resolution: 1
Use custom engine wrap angle	°CA	If selected, a custom wrap angle can be specified for the engine angle. Otherwise the default wrap angle (e.g. 360° for 4-stroke engines) is applied.	Range: 0°...1080° Resolution: 0.004°
Engine type (angular base)	-	The engine type and corresponding angular base of the engine cycle can be specified here.	2-stroke (360°CA) 4-stroke (720°CA) 6-stroke (1080°CA)

Sample time FPGA main block	s	The sample time the FPGA main component is clocked with. Usually this is the FPGA clock rate, however in case of downsampling of the FPGA component this parameter has to be adapted.	-
-----------------------------------	---	---	---

(*) Assuming FPGA sample time of 8 ns. For other sample times, the ranges and resolutions scale accordingly.

	The information about engine and cylinder wrap angles are required by several components. As the wrap angles are usually the same for all components, it is recommended to use a variable for these parameters.
--	---

Output

The Processor Out block outputs to 4 register contents, mapped to 1 register. The mapping is shown below.

Register 1.1

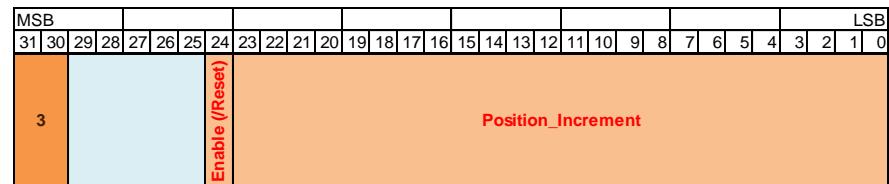


Register 1.2



Register 1.3



Register 1.4

FPGA Main Component

Block

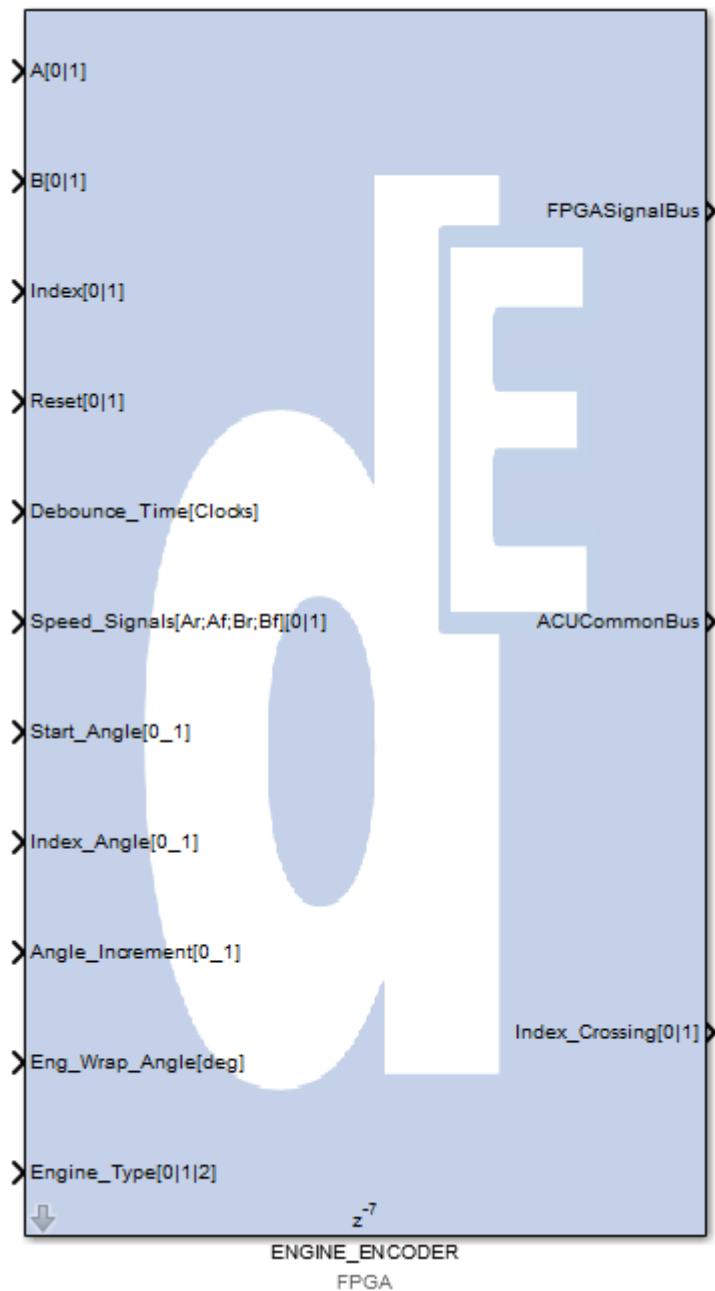


Figure 40: ENGINE_ENCODER FPGA Main block

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	-	The track A input of the connected incremental encoder.	Bool
B	-	The track B input of the connected incremental encoder.	Bool
Index	-	The index (reference) signal. Possibilities for this input are e.g. a reference track of the incremental encoder or an FPGA-internal synchronization signal.	Bool
Reset	-	Resets the angle processing.	Bool
Debounce Time	Clocks	The number of clocks for the signal debounce time.	UFix_10_0
Speed_Signals	-	Flags determining which edges are to be used for speed measurement (A rising, A falling, B rising, B falling)	Bool (4x)
Start_Angle	-	The angle assumed at startup. The start angle input is scaled in the range 0...1, which corresponds to a whole engine cycle (e.g. 1 = 720° for 4-stroke engines)!	UFix_24_24
Start_Angle	-	The angle when passing the index (rising edge during forward movement, negative edge during reverse movement). The index angle input is scaled in the range 0...1, which corresponds to a whole engine cycle (e.g. 1 = 720° for 4-stroke engines)!	UFix_24_24
Angle_Increment	-	The angle increment of 1 period on the incremental encoder tracks relative to a whole engine cycle (scaling 0...1). Effectively, this is the reciprocal of the number of encoder lines per engine cycle.	UFix_24_24
Eng_Wrap_Angle	-	The wrap angle the engine angle output.	UFix_19_8

Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0
-------------	---	---	----------

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
ACUCommonBus	-	Contains a normed output signal bus equal for all angle-processing components (see ACUCommonBus)	Bus
Index_Crossing	-	Impulse indicating that the index has just been passed. Can for example be used as an interrupt trigger.	Bool

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	738	0.18%
LUTs	710	0.35%
Block RAM	2.5	0.56%
DSP Slices	1	0.11%

Processor Input

Block Adapts the FPGA signals for the processor side.

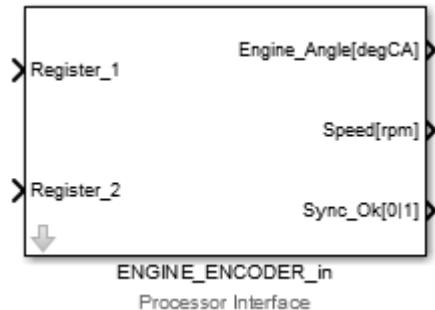


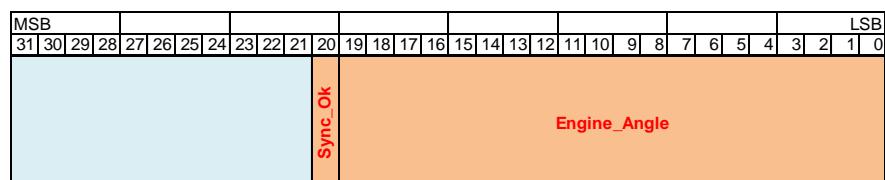
Figure 41: ENGINE_ENCODER_in block



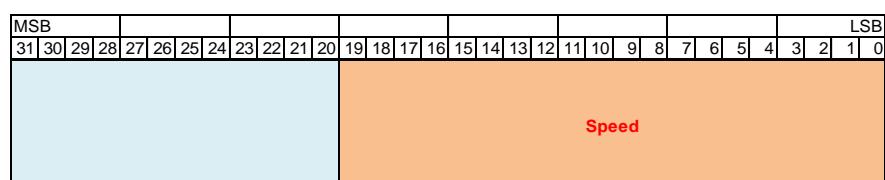
This block is only used to provide output signals to the processor application. If the signals are not required in the processor application, computation resources can be saved by not using this block.

Input The processor input block has 2 input register with a sectioning shown below.

Register 1



Register 2



Output

The ENGINE_ENCODER_in block has the following outputs:

Name	Unit	Description	Range
Position	°CA	The engine angle calculated by the FPGA main component.	Range: $\pm 1080^\circ$ Resolution: 0.004°
Speed	rpm	The engine speed calculated by the FPGA main component.	Range: ± 32767 rpm Resolution: 0.06rpm
Sync_Ok	-	The valid flag, which is raised as soon as the position is referenced (index passed).	0 1

Pressure Indication

Objective

This sub-library contains algorithms to determine the in-cylinder pressure parameters during combustion. Each CPI component is able to handle up to four cylinders. If the engine consists of more than four cylinders, use 2 CPI components.

Features

The *Pressure Indication* components provide the following main features:

- Automatic drift compensation of piezo sensors
- Real-time (in-cycle) combustion detection based on pressure gradients
- Calculation of indicated mean effective pressure (IMEP, IMEPcc, IMEPwc)
- In-cycle calculation of incremental and integrated released heat (dQB, QB)
- Calculation of mass fraction burned (MFB10/50/90)
- Calculation of maximum pressure / pressure gradient and related angles (p_{max} / dp_{max})
- Indication is performed in hard real-time, means the significant results are available and updated continuously, not only at the end of the combustion cycle
- Support of 4 cylinders with a single component, 8 or more cylinders are possible by using multiple components



Currently cylinder pressure indication (CPI) is only available for 4-stroke engines.

Content

The section contains the following components:

- CPI_4CYL
- COMBUST_DETECT

CPI_4CYL

Objective	This component contains the cylinder pressure indication for engines with up to 4 cylinders.
------------------	--

Content	The blockset contains the following elements:
----------------	---

- Processor Interface: CPI_4CYL_out
(Processor Interface)
- FPGA Interface: CPI_4CYL_in
(FPGA Interface)
- FPGA: CPI_4CYL
(FPGA Main Component)
- FPGA Interface: CPI_4CYL_out
(FPGA Interface)
- Processor Interface: CPI_4CYL_in
(Processor Interface)

Principle of Operation	Cylinder pressure indication detects angles of mass fraction burned as other characteristics relevant for control of combustion. The inputs required are the in-cylinder pressures as well as the engine angle determined by ACU or encoder.
Pressure Filters	The pressure input signals are filtered by an IIR-filter. The cut-off frequency of the filter can be set by the input of the mask.

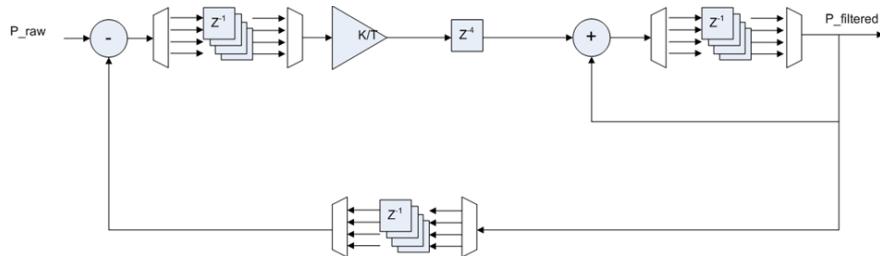


Figure 42: Structure of the IIR-filter (4 channels in this example)

The IIR-filter is a modification of a PT1-filter. Due to FPGA-related performance reason, additional 4 delay elements are added into the Multiplier. One Pole remains the same as the original PT1-filter. This pole determines the cut-off frequency. The other new poles are placed outside of the relevant frequency range and therefore have no practical effect.

The transfer function of the filter is:

$$G(z) = \frac{Kz}{z^3 - z^2 + K} \quad \text{with} \quad K = \frac{1}{1 + \frac{1}{f_c \cdot T_s}}$$

f_c : cutoff frequency in rad

With $f_c = 15000$ rad = 2387 Hz, the frequency response and pole-zero plain look as followed:

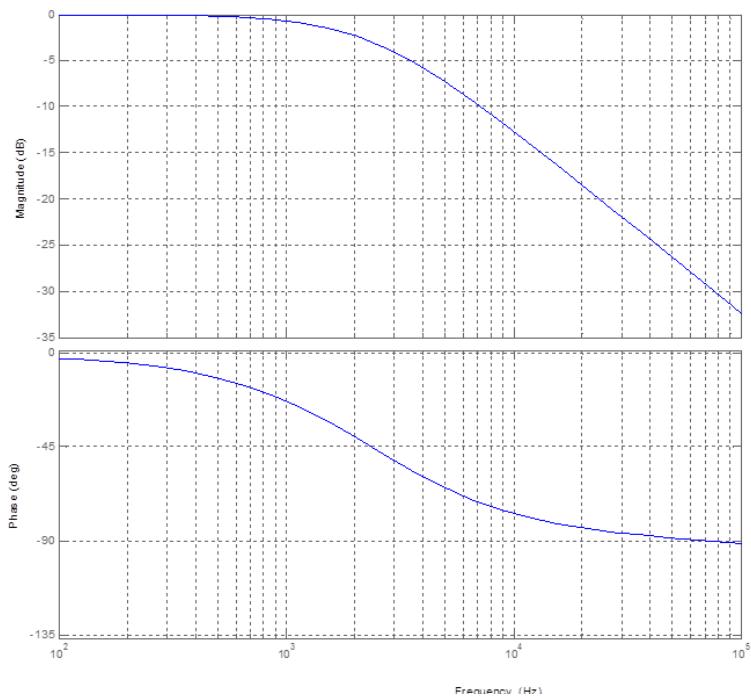


Figure 43: Bode diagram for example filter characteristic

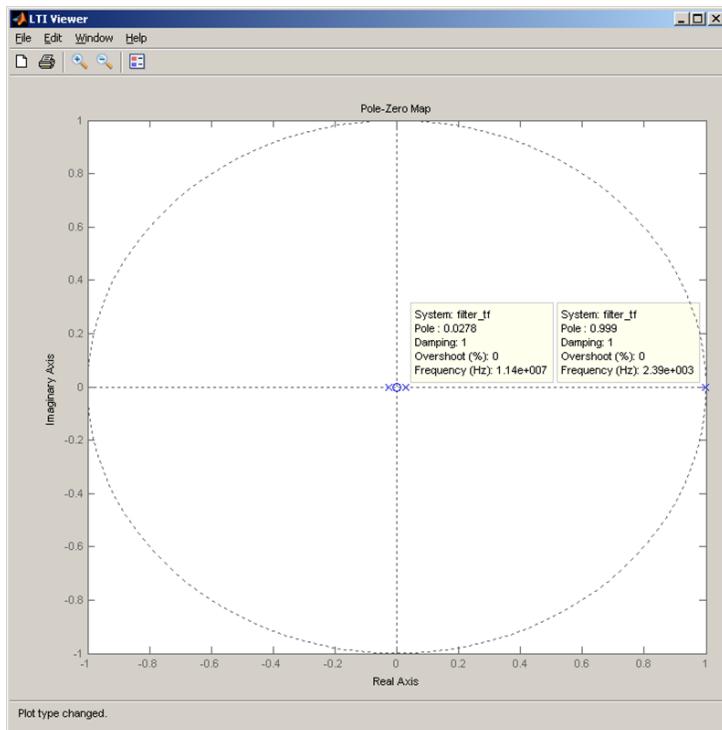


Figure 44: Unit circle with poles for example filter characteristic

Drift Compensation

Generally piezoelectric pressure sensors are used to measure the pressure in the cylinder and they introduce an offset error due to its physical nature. In other words the measured value drifts gradually away. This offset needs to be corrected.

The underlying formula for the calculation used is:

$$p_1 = \frac{\Delta p_{meas}}{\left[\left(\frac{V_1}{V_2} \right)^n - 1 \right]} \quad n = 1.32 \text{ for Otto engines}$$

$$n = 1.37 \text{ for Diesel engines}$$

For the calculation a start angle, an end angle and kappa (polytropic exponent) is needed. The start angle and the end angle must be in the range of the compression phase of the engine, where no combustion is occurred yet. Therefore kappa is called 'kappa compression'.

For simplification, isentropic exponent can be assumed for kappa. V_1 is the volume of the cylinder at start angle. V_2 is the volume of the cylinder at end angle.

The same is true is pressure p_1 .

The setup block calculates a drift compensation factor which is used by the FPGA algorithm.

Combustion Detection

For good results of the calculation of mass fraction burned (MFB10/MFB50/MFB90) and integrated released heat caused by combustion (QB) it is important to detect the time of combustion correctly.

The CPI solution is delivered with an automatic detection algorithm. This algorithm works in real time and detects combustion in the current engine cycle. However this algorithm relies on high quality of input signal (the measured pressure). In field applications it is not always possible to meet this requirement.

A possibility more robust method can be applied to detect the combustion by user. This method uses the result of an additional "QB for calibration" algorithm which is calculated within a special user-definable detection window.

More details can be found below under section Manual mode.

Automatic Mode

In the automatic mode combustion start and end are determined by observation of the pressure alternation in the cylinder. When the pressure increment, after neglected the rise caused by volume compression, is bigger than a certain user defined value for a certain user defined duration, then a combustion start has occurred. The opposite way around, the pressure decrement, after neglected the fall caused by volume expansion is bigger than a certain user defined value for a certain user defined duration, then a combustion end has occurred. Furthermore filters are applied on the derivative pressure signal (Δp_{raw}) in order to filter out noise and measurement error.

Manual Mode

In the manual mode the user defines start angle and end angle for the combustion measurement window.

To determine the angles, use a wider combustion detection window in combination with QB calibration. One possibility is to make use of the characteristic undershot and overshoot of QB at the beginning and the end of the Combustion. Two QB values are provided by the blockset: “QB for Calibration” (QB_Cal) and “QB for Calculation” (QB).

A rather wide window (combustion detection window) is defined for “QB for Calibration” so that “QB for Calibration” includes the time before and after combustion with certainty. In this way “QB for Calibration” shows the characteristic undershot and overshoot. Then the smaller combustion measurement window should be tuned for calculation of “QB for Calculation” in the next engine cycle. The smaller combustion measurement window should be set at the angles of the undershot and overshoot. A user-defined algorithm to detect the undershot and overshoot points can be implemented to accomplish this task automatically.

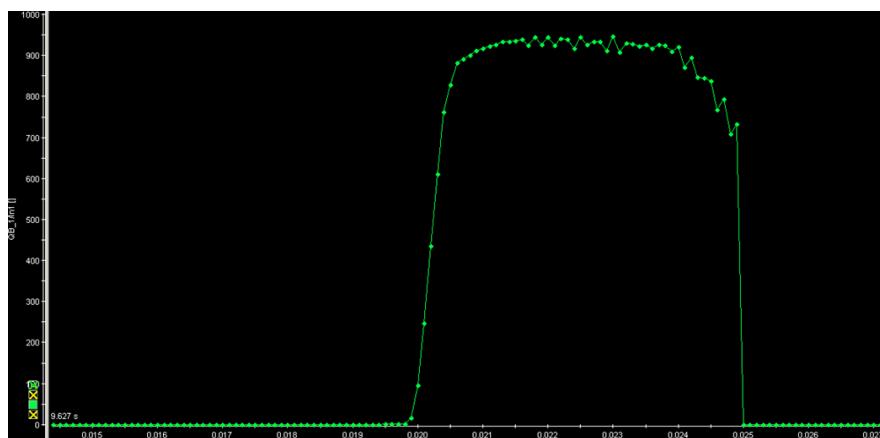


Figure 45: QB for calibration

Figure 45 shows QB calculated over the range of combustion detection window. The curve shows an undershot at the start and overshoot at the end of combustion.

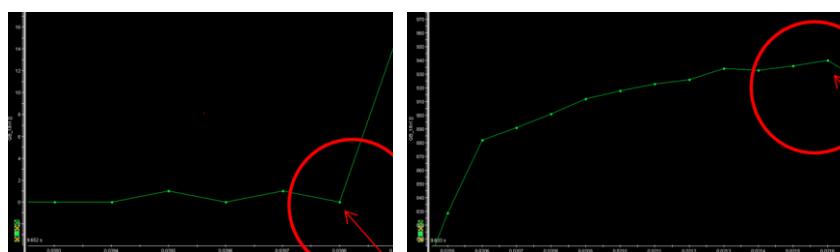


Figure 46: Over- and undershot of QB for calibration

The last falling point before the characteristic incline would define the starting angle of the combustion measurement window of the next cycle. The first falling point after the characteristic incline would define the ending angle of the combustion measurement window of the next cycle.

With the correct window setting, the finally QB, so called "QB for Calculation", should be a monotonic increasing curve

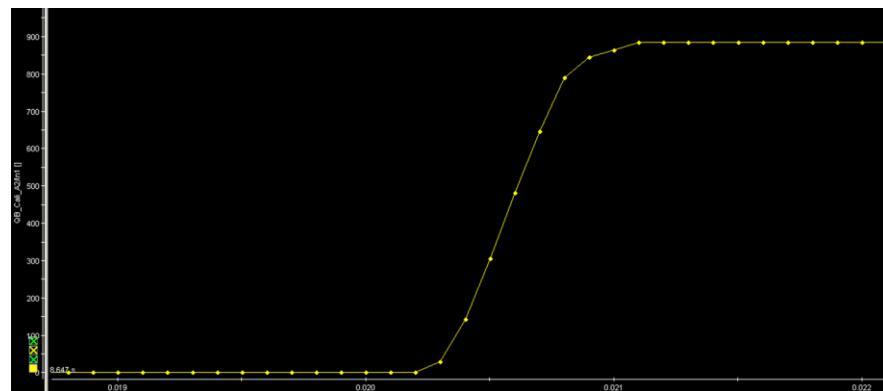


Figure 47: QB for calculation

Since information of the last engine cycle are used for setting of the combustion measurement window of the current cycle, discrepancy can be happen depending on the current dynamic of the process.



If the combustion measurement window is too wide, several CPI result values become wrong, e.g. MFB, QB and so on.

IMEP

The cylinder pressure indexing calculates indicated mean effective pressure (IMEP) according to the following formula:

$$IMEP = \frac{1}{V_h} \cdot \int p \cdot dV$$

V_h: swept volume
p: pressure measured in cylinder
dV: change of volume of the cylinder in
0.1 °CA

Additionally, the indicated mean effective pressure of the charge cycle is calculated for each cylinder. The mathematical formula used for the calculation of IMEP_{CC} is:

$$IMEP_{CC} = \frac{1}{V_h} \cdot \int_{GX_{start}}^{GX_{end}} p \cdot dV$$

V_h: swept volume
p: pressure measured in cylinder

dV:	change of volume of the cylinder in step of 0.1 °CA
GX _{start} :	IMEP charge cycle start angle (GX = gas exchange)
GX _{end} :	IMEP charge cycle end angle (GX = gas exchange)

Finally, the indicated mean effective pressure of the working cycle is calculated for each cylinder. The mathematical formula used for the calculation of IMEP_WC (p_{mi_WC}) is:

$$IMEP_{WC} + IMEP_{CC} = IMEP$$

dQB

dQB is calculated according to the following formula:

$$QB = \sum_{combust_start}^{combust_end} dQB$$

The calculation is executed in discrete manner (instead of continuous). Therefore the output can have strong oscillation, especially when the applied resolution (0.1°CA) in FPGA is much higher than the sample rate for the input pressure values. The oscillation will be eliminated in the accumulation (discrete integration) at QB.

The following formula is used as basis for the calculation of dQB:

$$dQB = \frac{k}{k-1} \cdot p \cdot dV + \frac{1}{k-1} \cdot V \cdot dp + \alpha_i \cdot A_w \cdot (T_i - T_w) \cdot dt$$

The last term is dropped, since the effect of heat loss through combustion chamber's wall is neglected. Then the formula remains to this:

$$dQB = \frac{1}{k-1} (kp dV + V dp)$$

κ :	polytropic exponent
p:	pressure in Cylinder
dp:	change of p
V:	current volume of the cylinder
dV:	change of V

This formula in discrete format with a resolution of 0.1 °CA is implemented on the FPGA.

Processor Output

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

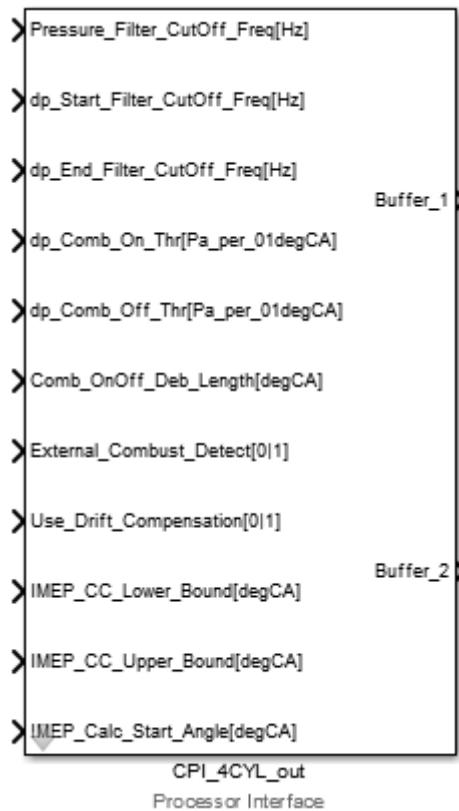


Figure 48: CPI_4CYL_out block

Block Dialog	The processor output block provides the following dialog:
---------------------	---

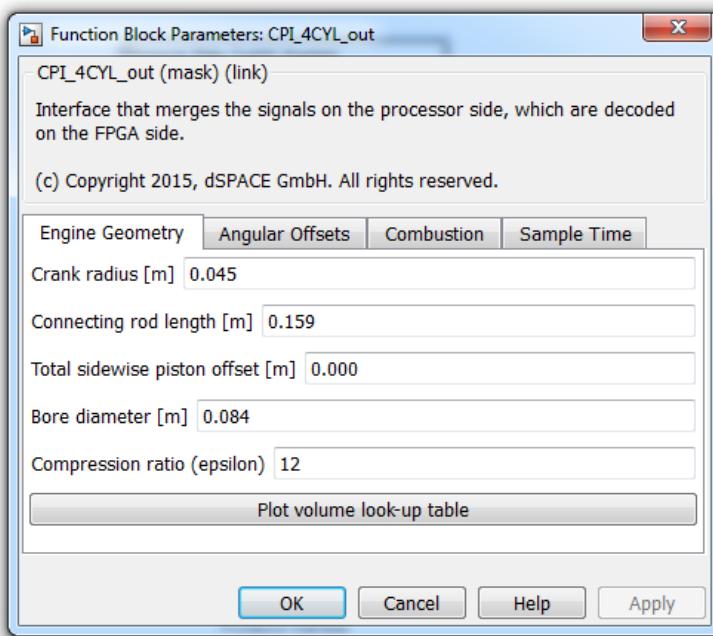
Engine Geometry

Figure 49: CPI_4CYL_out dialog (Engine Geometry)

The *Engine Geometry* tab has the following parameters:

Name	Unit	Description
Crank radius	m	The distance starting from the crank axis to the middle of crank pin.
Connecting rod length	m	The length of the con-rod starting from the middle of fixing point on the piston and ending at the middle of the fixing point on the crankshaft pin.
Total sidewise piston offset	m	<p>The horizontal distance from the joint of the piston and con-rod (piston pin) to crank axis. When looking from the direction so that crank is rotating clockwise, then Total Sidewise Piston Offset is defined as positive, if crank axis is to the left of the piston pin. Reversely Total Sidewise Piston Offset is defined as negative, if crank axis is to the right of the piston pin.</p> <p>Example:</p> <p>In the case the skewing and piston pin offset both are offset to cylinder axis into the same direction (like Figure 49), then:</p> $ Total Sidewise Piston Offset = Skewing - piston pin offset $ <p>In the case the skewing and piston pin offset both are offset to cylinder axis into different directions then:</p> $ Total Sidewise Piston Offset = Skewing + piston pin offset $ <p>Skewing is the offset between cylinder axis and crank axis.</p> <p>Piston Pin Offset is the offset between cylinder axis and piston pin.</p>
Bore diameter	m	The diameter of the cylinder bore.
Compression ratio	-	<p>The compression ratio (ϵ) determined by:</p> $\epsilon = (V_d + V_c) / V_c$ <p>V_c : clearance volume V_d : displacement volume of the cylinder</p>

 Pressing the checkbox button *Plot volume look-up table* will plot the calculated volume in relation to the crank angle in MATLAB.

 The information about engine and cylinder wrap angles are required by several components. As the wrap angles are usually the same for all components, it is recommended to use a variable for these parameters.

 Please note that the maximum cylinder volume supported by the FPGA application is $< 1 \text{ m}^3$. The plotter in the interface does not have these limitations, but volume values over 1 m^3 will be saturated to 1 m^3 when transferred to the FPGA.

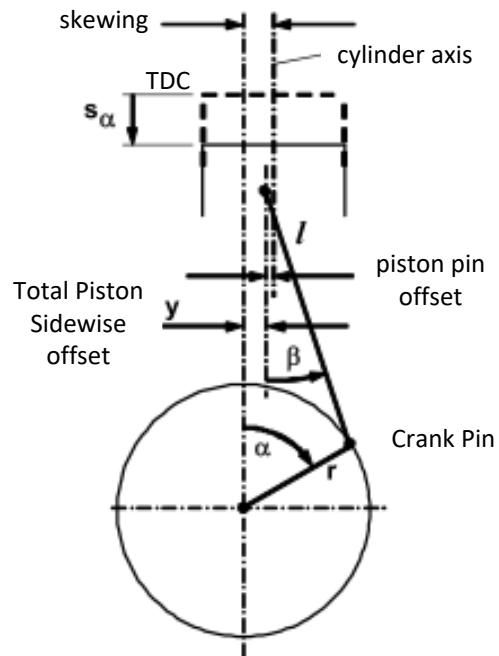


Figure 50: Engine geometry parameter definitions

Angular Offsets

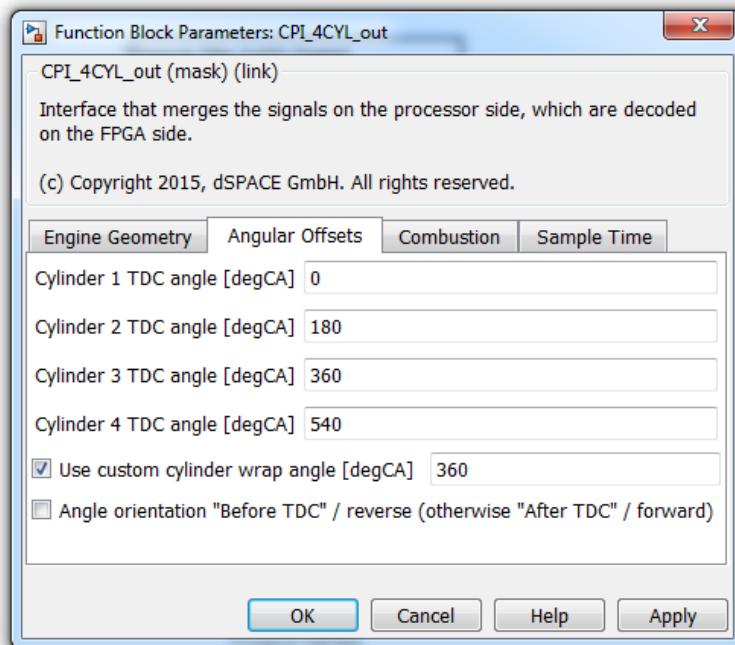


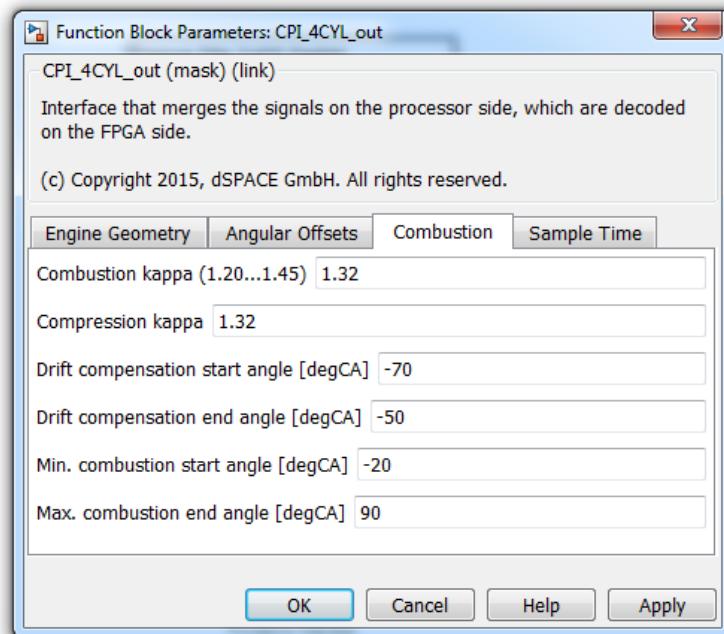
Figure 51: CPI_4CYL_out dialog (Angular Offsets)

The *Angular Offsets* tab has the following parameters:

Name	Unit	Description	Range/Resolution
Cylinder 1 TDC angle offset	°CA	The offset of the TDC angle of the 1 st cylinder relative to the engine angle	Range: 0°...720° Resolution: 0.004°
Cylinder 2 TDC angle offset	°CA	The offset of the TDC angle of the 2 nd cylinder relative to the engine angle	Range: 0°...720° Resolution: 0.004°
Cylinder 3 TDC angle offset	°CA	The offset of the TDC angle of the 3 rd cylinder relative to the engine angle	Range: 0°...720° Resolution: 0.004°
Cylinder 4 TDC angle offset	°CA	The offset of the TDC angle of the 4 th cylinder relative to the engine angle	Range: 0°...720° Resolution: 0.004°
Use custom engine wrap angle	°CA	If selected, a custom wrap angle can be specified for the engine angle. Otherwise the default wrap angle (e.g. 360° for 4-stroke engines) is applied.	Range: 0°...720° Resolution: 0.004°
Angle orientation "Before TDC"	-	If this option is checked the direction of the cylinder angles is reverse to the direction of the engine angle.	on / off



The TDC angle distance between cylinder 1 and 3 as well as between cylinder 2 and 4 must be at least 180°.
 For an 8-cylinder engine for example, cylinders 1, 3, 5 and 7 can be mapped to the 1st CPI component, while cylinders 2, 4, 6 and 8 can be mapped to the 2nd CPI component.
 If not all cylinders of a CPI component are required (e.g. 3- or 6-cylinder engines), use dummy TDC angles for the unused cylinders which meet this requirement.

Combustion**Figure 52: CPI_4CYL_out dialog (Combustion)**

The *Combustion* tab has the following parameters:

Name	Unit	Description	Range/Resolution
Combustion kappa	-	The polytropic exponent κ during the combustion phase.	Range: -2...2 Resolution: 0.0001
Compression kappa	-	The polytropic exponent κ during the compression phase of the engine. It is used for the calculation of drift compensation.	Range: -2...2 Resolution: 0.0001
Drift compensation start angle	°CA	The angle where the drift compensation calculation starts.	Range: ±720° Resolution: 0.004°
Drift compensation end angle	°CA	The angle where the drift compensation calculation ends.	Range: ±720° Resolution: 0.004°
Min. combustion start angle	°CA	The minimum possible angle for start of combustion.	Range: ±720° Resolution: 0.004°
Max. combustion end angle	°CA	The maximum possible angle for end of combustion.	Range: ±720° Resolution: 0.004°

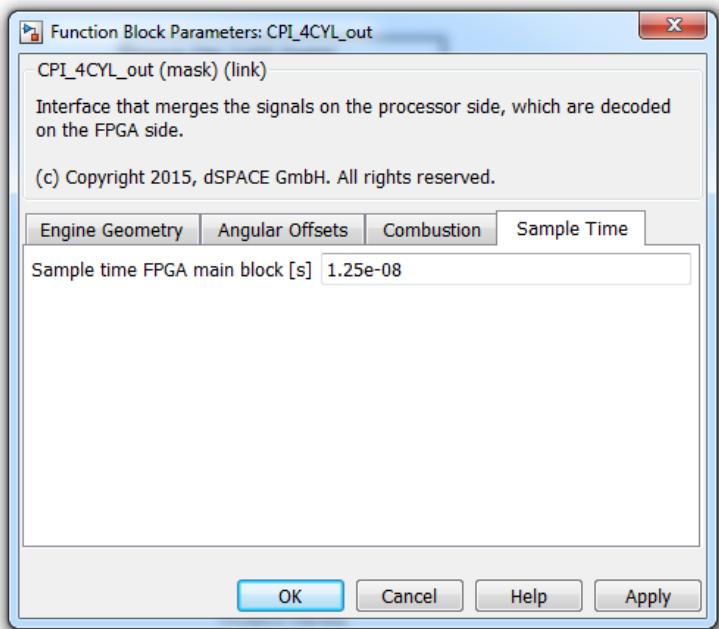
Sample Times

Figure 53: CPI_4CYL_out dialog (Sample Times)

The *Sample Times* tab has the following parameters:

Name	Unit	Description
Sample Time FPGA main block	s	The sample time the FPGA main component is clocked with. Usually this is the FPGA clock rate, however in case of downsampling of the FPGA component this parameter has to be adapted.

Input

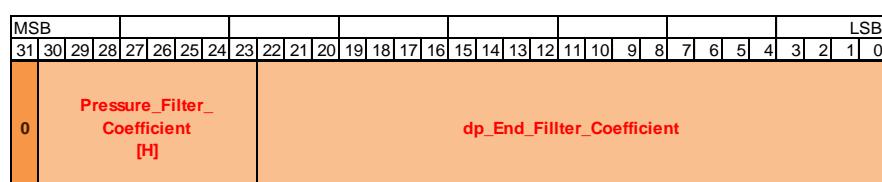
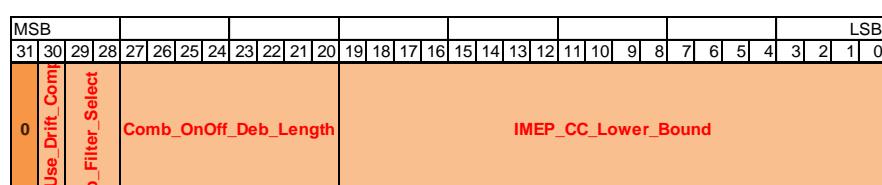
The CPI_4CYL_out block has the following inputs:

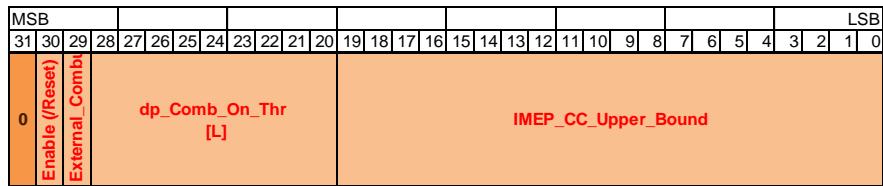
Name	Unit	Description	Range/Resolution
Pressure_Filter_CutOff_Freq	Hz	The in-cylinder pressures are filtered by an IIR-filter. The cut-off frequency of the filter can be set here.	Range: 0...1MHz Resolution: 1Hz
dp_Start_Filter_CutOff_Freq	Hz	IIR-filters are applied on the derivative pressure signal (Δp_{raw}) in order to filter out noise and measurement error (for automatic detection). This filter value applies for the combustion start detection. A suitable default value is 1kHz.	Range: 0...1MHz Resolution: 1Hz
dp_End_Filter_CutOff_Freq	Hz	IIR-filters are applied on the derivative pressure signal (Δp_{raw}) in order to filter out noise and measurement error (for automatic detection). This filter value applies for the combustion end detection. A suitable default value is 1kHz.	Range: 0...1MHz Resolution: 1Hz
dp_Comb_On_Thr	Pa / 0.1°	Combustion start threshold (in Pa/0.1°C) defines how steep the rise of pressure needs to be in order start of combustion shall be detected (for automatic detection).	Range: ±131071 Resolution: 1
dp_Comb_Off_Thr	Pa / 0.1°	Combustion end threshold (in Pa/0.1°C) defines how steep the fall of pressure needs to be in order end of combustion shall be detected (for automatic detection).	Range: ±131071 Resolution: 1
Comb_OnOff_De_Length	°CA	The combustion detect debounce filter value defines the minimum duration of a rise or a fall of pressure for a combustion begin or stop being detected (for automatic detection).	Range: 0...25.5° Resolution: 0.1°
External_Combust_Detect	-	Switches the combustion detection method. 0: automatic detection 1: manual window setting	0 1

Use_Drift_Compensation	-	Activates (1) or deactivates (0) drift compensation.	0 1
IMEP_CC_Lower_Bound	°CA	Start angle of the charge cycle.	Range: ±720° Resolution: 0.004°
IMEP_CC_Upper_Bound	°CA	End angle of the charge cycle.	Range: ±720° Resolution: 0.004°
IMEP_Calc_Start_Angle	°CA	For the calculation of IMEP a start bound of integration needs to be defined. The integral for IMEP is a closed curve integral (cyclic integral) over engine cycle. It can be any valid angle in the user defined crank cycle.	Range: ±720° Resolution: 0.004°

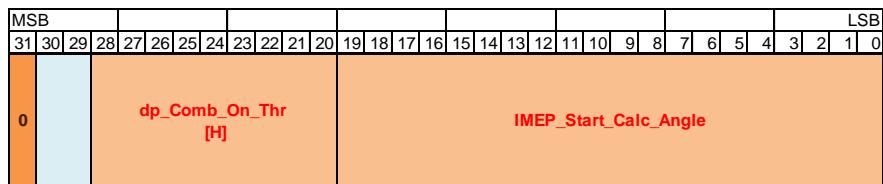
Output

The Processor Out block outputs up to 7217 register contents, mapped to two buffers. The first buffer has a depth of 6 and transmits cyclic data in every simulation step. The second buffer has a depth of 7211 and is used for initialization only and transmits calculated parameters for cylinder volume, drift compensation, etc.

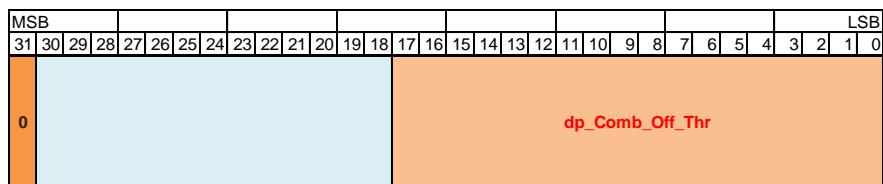
Buffer 1 – Register 1**Buffer 1 – Register 2****Buffer 1 – Register 3****Buffer 1 – Register 4**



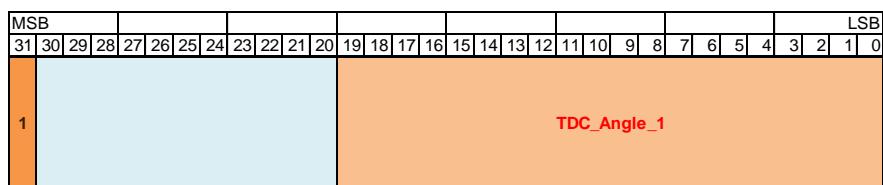
Buffer 1 – Register 5



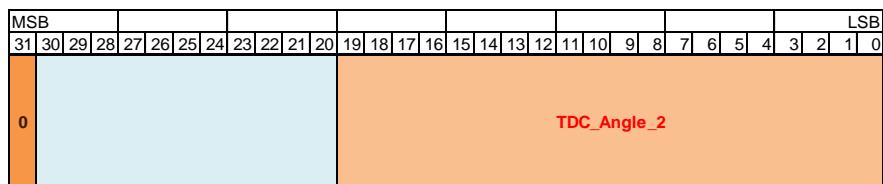
Buffer 1 – Register 6



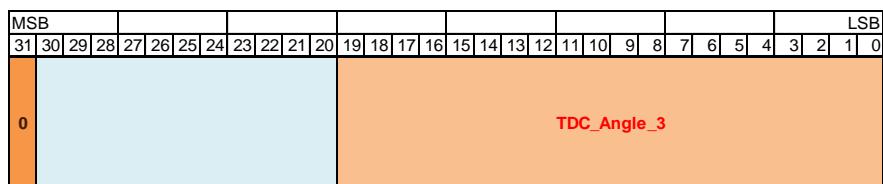
Buffer 2 – Register 1



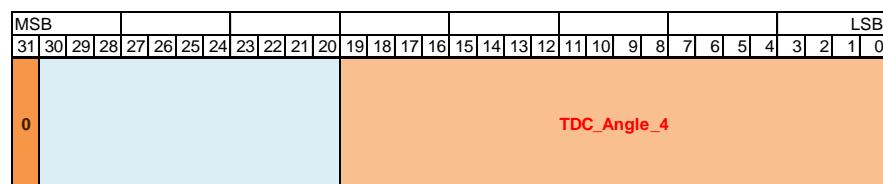
Buffer 2 – Register 2



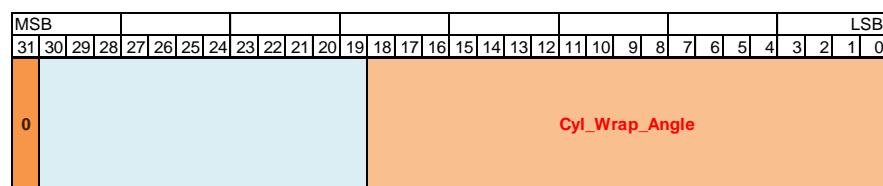
Buffer 2 – Register 3



Buffer 2 – Register 4



Buffer 2 – Register 5



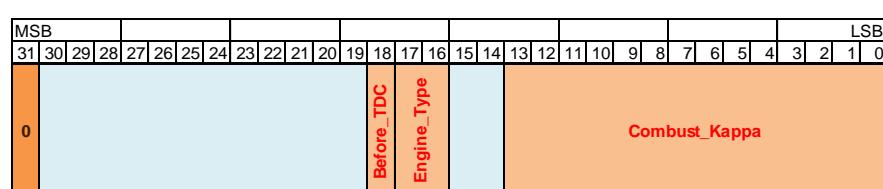
Buffer 2 – Register 6



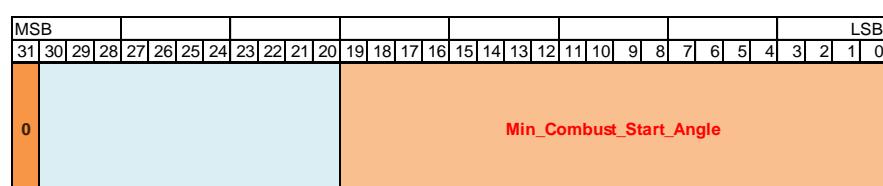
Buffer 2 – Register 7



Buffer 2 – Register 8



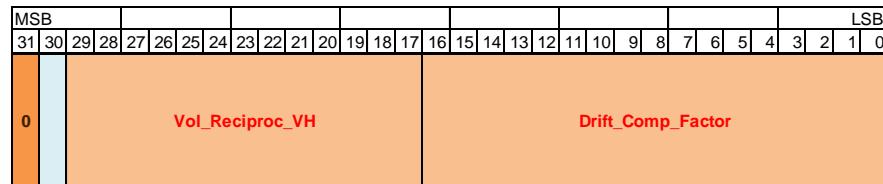
Buffer 2 – Register 9



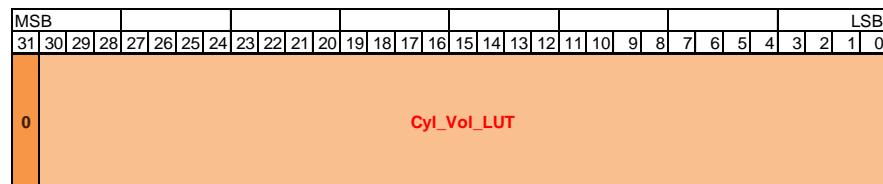
Buffer 2 – Register 10



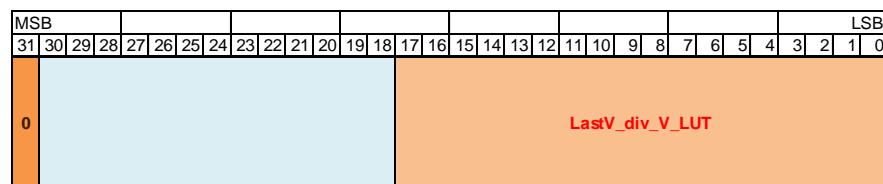
Buffer 2 – Register 11



Buffer 2 – Register 12, 14, 16 ... 7210



Buffer 2 – Register 13, 15, 17 ... 7211



MicroAutoBox III

For the MicroAutoBox III the initialization buffer is not implemented in Simulink. The initialization is done via the initial value of the FPGA function block in the ConfigurationDesk signal chain.

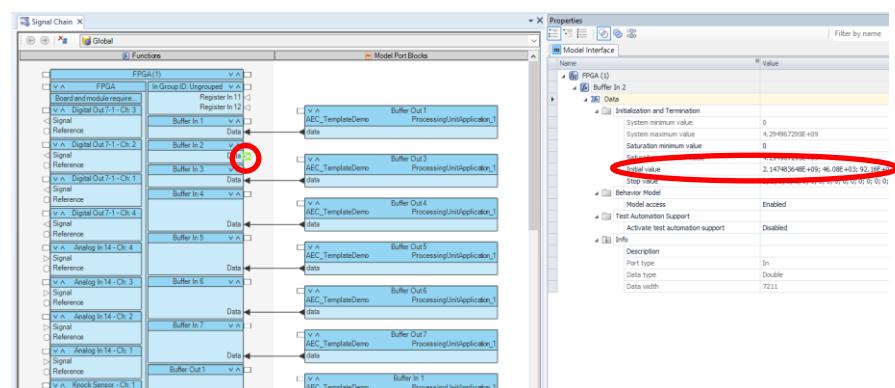


Figure 54: Initialization in ConfigurationDesk

FPGA Main Component

Block

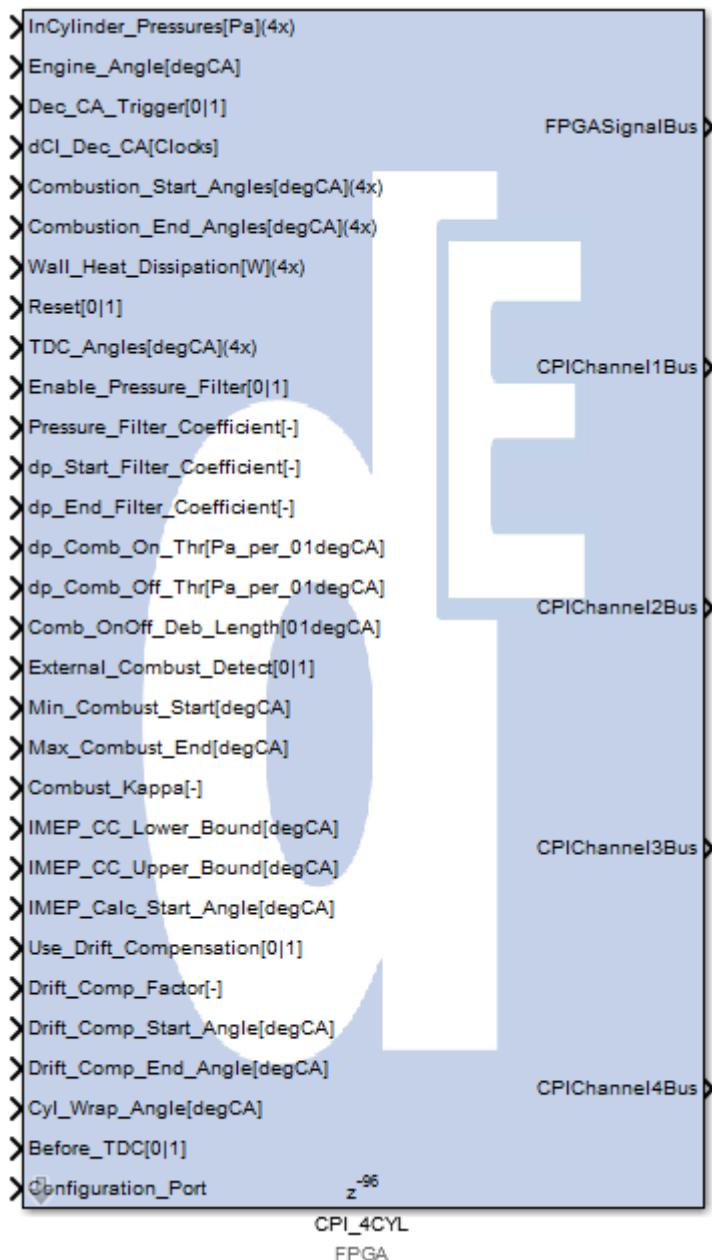


Figure 55: CPI_4CYL FPGA Main block

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.

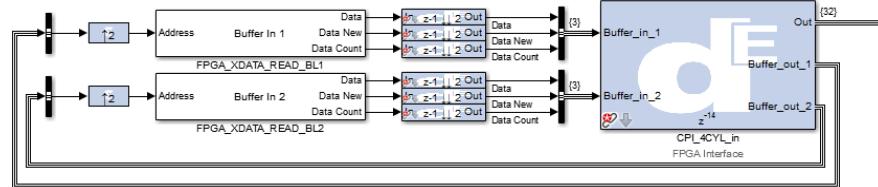
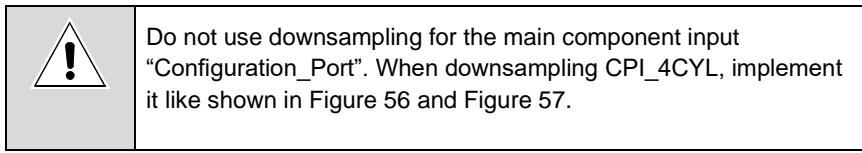


Figure 56: Correct input downsampling of CPI_4CYL

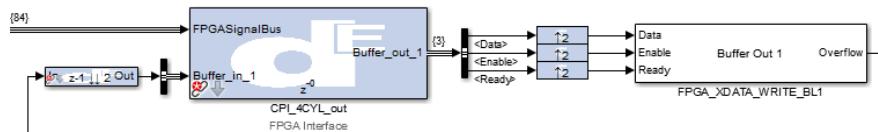
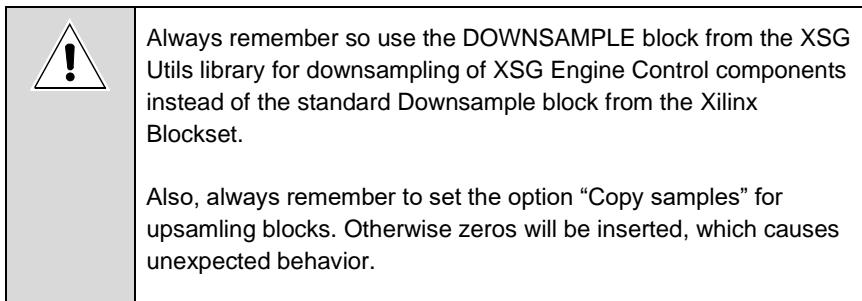


Figure 57: Correct output upsampling of CPI_4CYL



Input

The main block has the following inputs:

Name	Unit	Description	Format
InCylinder_Pressures	Pa	The in-cylinder pressures measured, scaled in Pa.	Fix_26_0 (4x)
Engine_Angle	°CA	The engine angle determined by the ACU or incremental encoder.	Fix_20_8
Dec_CA_Trigger	-	Impulse indicating that 0.1°CA have been passed (from ACU or encoder).	Bool
dCI_Dec_CA	Clocks	No of clocks cycles between the last two Dec_CA_Trigger impulses (from ACU or encoder).	UFix_24_0
Combustion_Start_Angles	°CA	If external combustion detection is selected, the start angles of combustion for each cylinder are provided by this input.	Fix_20_8 (4x)
Combustion_End_Angles	°CA	If external combustion detection is selected, the end angles of combustion for each cylinder are provided by this input.	Fix_20_8 (4x)
Wall_Heat_Dissipation	W	For a more correct dQB calculation the wall heat dissipation can be considered as well. It can be calculated by a user's algorithm and provided to this block input.	Fix_20_0 (4x)
Reset	-	Resets the cylinder pressure indexing component.	Bool
TDC_Angles	°CA	TDC angles of the 4 cylinders.	Fix_20_8
Enable_Pressure_Filter	-	Enables the IIR filter for in-cylinder pressure.	Bool
Pressure_Filter_Coefficient	-	Filter coefficient for the pressure input signal, calculated as $(1 + (\omega_c \cdot 4T_s)^{-1})^{-1}$	UFix_16_16
dp_Start_Filter_Coefficient	-	Filter coefficient of Δp for start of combustion detection, calculated as $(1 + (\omega_c \cdot 4T_s)^{-1})^{-1}$	UFix_23_23

dp_End_Filter_Coefficient	-	Filter coefficient of Δp for end of combustion detection, calculated as $(1+(\omega_c \cdot 4T_S)^{-1})^{-1}$	UFix_23_23
dp_Comb_On_Thr	Pa/0.1°	See Processor Output .	Fix_18_0
dp_Comb_Off_Thr	Pa/0.1°	See Processor Output .	Fix_18_0
Combust_OnOff_Deb_Length	0.1°CA	See Processor Output .	UFix_8_0
External_Combust_Detect	-	Switches between internal (0) and external (1) combustion detection. If set to 1, the combustion start and end angles are provided as a block input. If set to 0, these inputs are ignored.	Bool
Min_Combust_Start	°CA	The earliest possible combustion start angle.	Fix_20_8
Max_Combust_End	°CA	The latest possible combustion end angle.	Fix_20_8
Combust_Kappa	-	The polytropic exponent κ during the combustion phase	Fix_16_13
IMEP_CC_Lower_Bound	°CA	See Processor Output .	Fix_20_8
IMEP_CC_Upper_Bound	°CA	See Processor Output .	Fix_20_8
IMEP_Calc_Start_Angle	°CA	See Processor Output .	Fix_20_8
Use_Drift_Compensation	-	Activates (1) or deactivates(0) drift compensation	Bool
Drift_Comp_Factor	-	Drift compensation factor calculated as $((V_{p1}/V_{p2})^{\kappa}-1)^{-1}$, where V_{p1} is the cylinder volume at the drift compensation start angle, and V_{p2} the cylinder volume at the drift compensation end angle.	Fix_17_5
Drift_Comp_Start_Angle	°CA	The start angle for drift compensation.	Fix_20_8
Drift_Comp_End_Angle	°CA	The end angle for drift compensation.	Fix_20_8
Cyl_Wrap_Angle	-	The wrap angle for the cylinder angles.	UFix_19_8
Before_TCD	-	Determines the orientation of the cylinder angles. (0 = same direction as the engine angles, 1 = reverse direction than the engine angle).	Bool

Configuration_Port	-	Internal configuration port. Has to be connected to the CPI_4CYL_in component.	UFix_58_0
--------------------	---	--	-----------

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
CPIChannel1Bus	-	Contains the calculated physical parameters of the 1 st CPI channel/cylinder (see CPIChannel[x]Bus).	Bus
CPIChannel2Bus	-	Contains the calculated physical parameters of the 2 nd CPI channel/cylinder (see CPIChannel[x]Bus).	Bus
CPIChannel3Bus	-	Contains the calculated physical parameters of the 1 st CPI channel/cylinder (see CPIChannel[x]Bus).	Bus
CPIChannel4Bus	-	Contains the calculated physical parameters of the 2 nd CPI channel/cylinder (see CPIChannel[x]Bus).	Bus

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	17730	4.35%
LUTs	19014	9.33%
Block RAM	58	13.03%
DSP Slices	10	1.19%

Processor Input

Block Adapts the FPGA signals for the processor side.

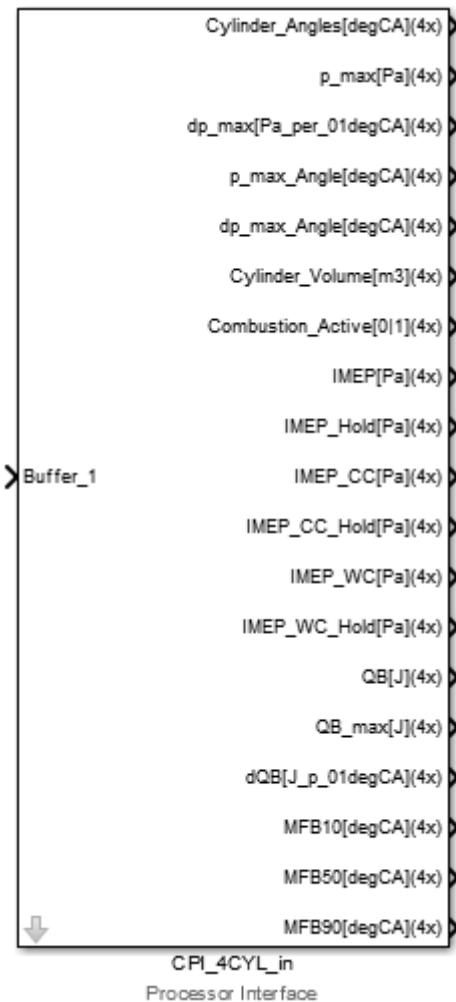


Figure 58: CPI_4CYL_in block

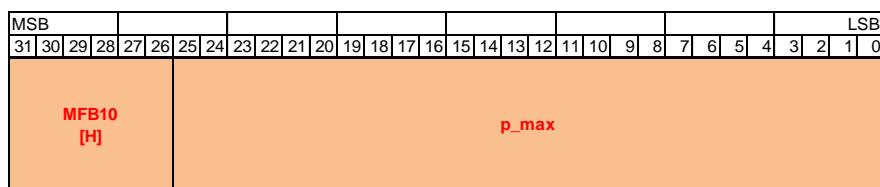


This block is only used to provide output signals to the processor application. If the signals are not required in the processor application, computation resources can be saved by not using this block.

Input

The processor input block has 1 input buffer containing 48 registers with a sectioning shown below.

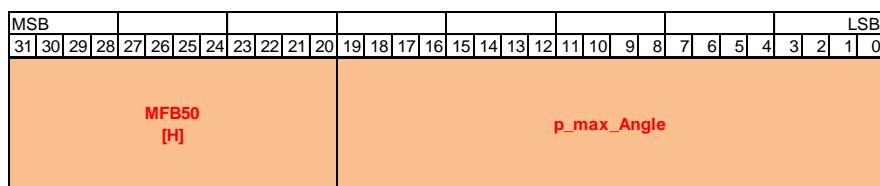
Register 1,13,25,37



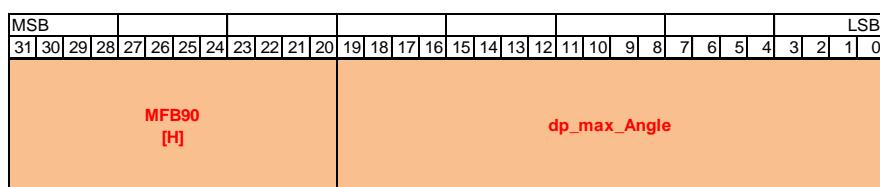
Register 2,14,26,38



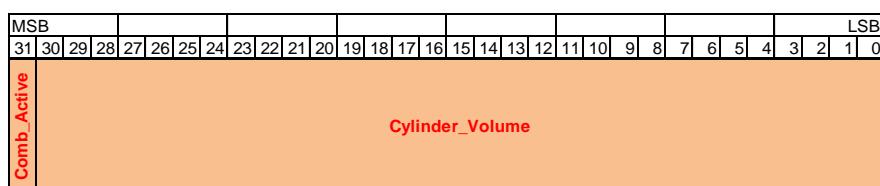
Register 3,15,27,39



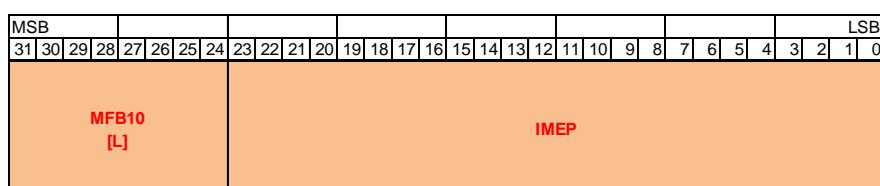
Register 4,16,28,40



Register 5,17,29,41



Register 6,18,30,42



Register 7,19,31,43

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
MFB50 [L]												IMEP_Hold																			

Register 8,20,32,44

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

Register 9,21,33,45

Register 10,22,34,46

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
QB_max [H]												QB [H]																			

Register 11,23,35,47

MSB | 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

dQB

Register 12,24,36,48

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
																Cylinder_Angle															

Output

The CPI_4CYL_in block has the following outputs:

Name	Unit	Description	Range/Resolution
Cylinder_Angles	°CA	The angles of the 4 cylinders according to the TDC and wrap settings	Range: ±720° Resolution: 0.004°
p_max	Pa	The measured maximum in-cylinder pressure of the current engine cycle.	Range: ±67MPa Resolution: 1Pa
dp_max	Pa / 0.1°	Maximum pressure slew rate of the current engine cycle.	Range: ±2·10 ⁷ Resolution: 1
p_max_Angle	°CA	The angle corresponding to the measured maximum pressure of the current engine cycle.	Range: ±720° Resolution: 0.004°
dp_max_Angle	°CA	The angle corresponding to the measured maximum pressure slew rate of the current engine cycle.	Range: ±720° Resolution: 0.004°
Cylinder_Volume	m ³	Current cylinder volume.	Range: 0...1m ³ Resolution: 0.5mm ³
Combustion_Active	-	Indicates if combustion is currently active.	0 1
IMEP	Pa	Current integration value of indicated mean effective pressure.	Range: ±16MPa Resolution: 1Pa
IMEP_Hold	Pa	Final value of the indicated mean effective pressure.	Range: ±16MPa Resolution: 1Pa
IMEP_CC	Pa	Current integration value of indicated mean effective pressure of the charge cycle.	Range: ±16MPa Resolution: 1Pa
IMEP_CC_Hold	Pa	Final value of integration value of indicated mean effective pressure of the charge cycle.	Range: ±16MPa Resolution: 1Pa
IMEP_WC	Pa	Current integration value of indicated mean effective pressure of the working cycle.	Range: ±33MPa Resolution: 1Pa
IMEP_WC_Hold	Pa	Final value of indicated mean effective pressure of the working cycle.	Range: ±33MPa Resolution: 1Pa
QB	J	The integrated released heat caused by combustion.	Range: ±32kJ Resolution: 125mJ
QB_max	J	The maximum released heat during the current engine cycle.	Range: ±32kJ Resolution: 125mJ
dQB	J / 0.1°	The incremental released heat caused by combustion.	Range: ±512J/0.1° Resolut.:238nJ/0.1°
MFB10	°CA	Angle where 10% of the mass fraction is burned.	Range: ±720° Resolution: 0.004°
MFB50	°CA	Angle where 50% of the mass fraction is burned.	Range: ±720° Resolution: 0.004°

MFB90	°CA	Angle where 90% of the mass fraction is burned.	Range: $\pm 720^\circ$ Resolution: 0.004°
-------	-----	---	---

	All CPI_4CYL_in outputs are vectors of 4 elements, one for each channel.
---	--

COMB_DETECT_4CYL

Description / Overview

The CPI component uses an integrated algorithm for combustion detection. However, user's also have the possibility to use own algorithms for determination of combustion start and end angles and provide those as inputs to the CPI component. The COMB_DETECT_4CYL is a simple example of such an external combustion detection algorithm, which uses the calibration outputs of the CPI component.



This block is not required for cylinder pressure indication, it just gives an example of a possible external combustion detection.

Figure 59 shows the principle of operation of this algorithm. According to the mass fraction burned estimated for calibration by the CPI unit, the start and end of combustion is extrapolated. According to the typical shape of the QB curve, non-linearity factors at the start and the end part of the curve can be defined by the user.

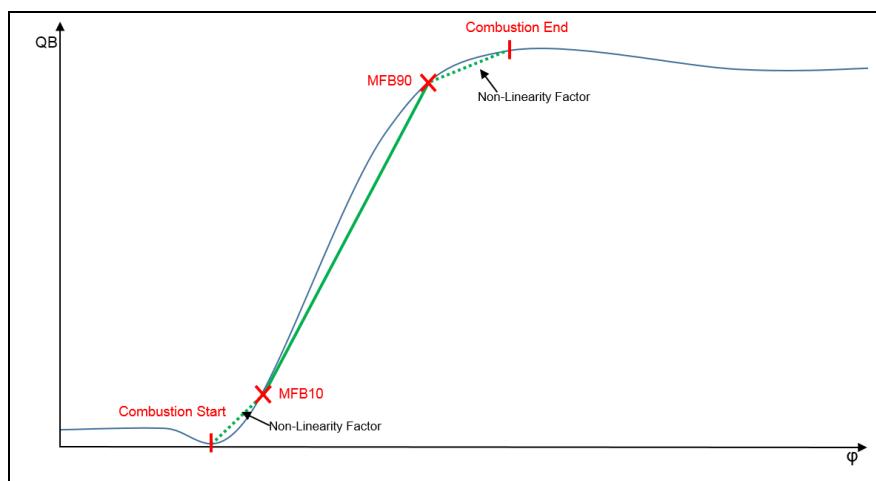
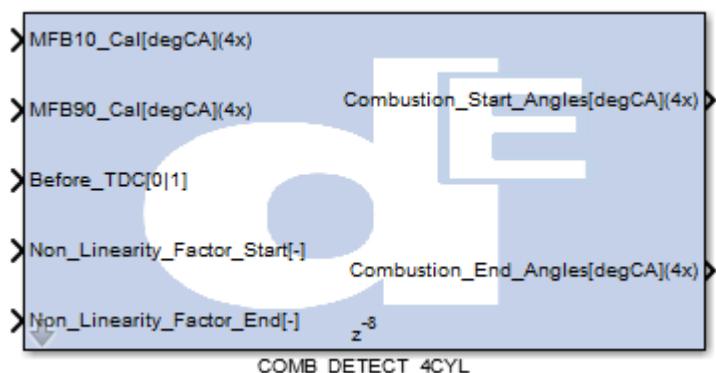
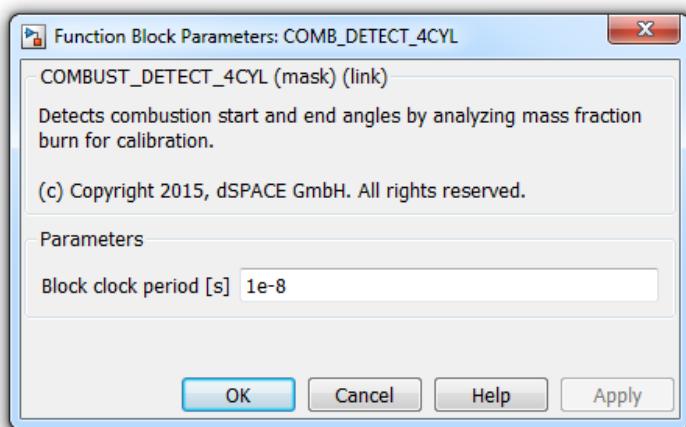


Figure 59: Example for external combustion detection

Block**Figure 60: COMB_DETECT_4CYL block****Block Dialog**

The processor output block provides the following dialog:

**Figure 61: COMB_DETECT_4CYL dialog**

The block dialog has the following parameters:

Name	Unit	Description	Range
Block clock period	-	The clock period the block is executed with. Generally this is the overall system rate, except downsampling or upsampling is applied.	0...inf

Input

The COMB_DETECT_4CYL block has the following inputs:

Name	Unit	Description	Format
MFB10_Cal	°CA	The MFB10 value for calibration, generated by the CPI_4CYL component.	Fix_20_8 (4x)
MFB90_Cal	°CA	The MFB90 value for calibration, generated by the CPI_4CYL component.	Fix_20_8 (4x)
Before_TCD	-	Determines the orientation of the cylinder angles. (0 = same direction as the engine angles, 1 = reverse direction than the engine angle).	Bool
Non_Linearity_Factor_Start	-	The combustion start is calculated as $MFB_{10} - (MFB_{90} - MFB_{10})/8 * x$ where x is the non-linearity factor for start of combustion.	UFix_16_12
Non_Linearity_Factor_End	-	The combustion end is calculated as $MFB_{90} + (MFB_{90} - MFB_{10})/8 * x$ where x is the non-linearity factor for end of combustion.	UFix_16_12

Output

The COMB_DETECT_4CYL block has the following outputs:

Name	Unit	Description	Format
Combustion_Start_Angles	°CA	The combustion start angles for 4 cylinders calculated by the component.	Fix_20_8 (4x)
Combustion_End_Angles	°CA	The combustion end angles for 4 cylinders calculated by the component.	Fix_20_8 (4x)

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	273	0.07%
LUTs	213	0.10%
Block RAM	0	0.00%
DSP Slices	2	0.24%

Knock Detection

Objective

This sub-library contains algorithms to determine the occurrence of knock in the engine. Each knock detection block processes one knock sensor input and calculates the three knock amplitudes with three user configurable FIR filters.



This blockset is only supported for the DS1554 Engine Control I/O Module as this module has dedicated inputs for knock sensor measurement!

Features

The *Knock Detection* components provide the following main features:

- Calculation of three knock amplitudes
- Configurable sensor gain factor
- Three FIR filters configurable by the user
- Definition of up to 16 measurement windows

Content

The library contains the following components:

- KNOCK_DETECTION

KNOCK_DETECTION

Objective	This component contains the calculation of the knock amplitude for one knock sensor.
------------------	--

Content	The blockset contains the following elements:
----------------	---

- Processor Interface: KNOCK_DETECTION_out (Processor Interface)
- FPGA Interface: KNOCK_DETECTION_in (FPGA Interface)
- FPGA: KNOCK_DETECTION (FPGA Main Component)
- FPGA Interface: KNOCK_DETECTION_out (FPGA Interface)
- Processor Interface: KNOCK_DETECTION_in (Processor Interface)

Principle of Operation

The KNOCK DETECTION block calculates the knock amplitude from an analog input signal coming from a piezoelectric knock sensor via customizable finite impulse response (FIR) filters. The following schematic shows the inner working of the block.

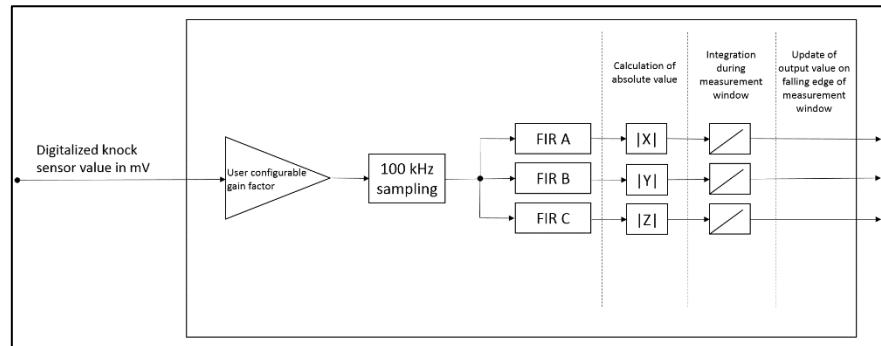


Figure 62: Calculation of knock sensor amplitude

The KNOCK DETECTION block expects a knock value in mV as input. This input then can be amplified by a user-specific gain factor. After that the input signal will be sampled by 100 kHz and is fed into three FIR filters simultaneously. Those filter outputs are then converted to absolute values and integrated during the active measurement window. The measurement windows can be specified by the user as well. At the falling edge of the measurement window, the values are then updated in the output registers.

FIR filter coefficients

The filter coefficients for each filter can be set manually or with the help of 10 presets. Here the user can choose a preset for each filter. The coefficients are set by the KNOCK_DETECTION_out processor block, which is described further below.

The following table shows the available presets. All presets are Kaiser Window band-pass filters with 50 coefficients each.

FIR preset	Center frequency (fce)	Lower cutoff frequency (fc1)	Upper cutoff frequency (fc2)
1	4 kHz	2.9 kHz	5.1 kHz
2	5 kHz	3.9 kHz	6.1 kHz
3	6 kHz	4.9 kHz	7.1 kHz
4	7 kHz	5.9 kHz	8.1 kHz
5	8 kHz	6.9 kHz	9.1 kHz
6	9 kHz	7.9 kHz	10.1 kHz
7	10 kHz	8.9 kHz	11.1 kHz
8	12 kHz	10.9 kHz	13.1 kHz
9	14 kHz	12.9 kHz	15.1 kHz
10	16 kHz	14.9 kHz	17.1 kHz

On top of using those presets, it's also possible to specify custom filters. The following limitations apply to custom filters:

- Has to be a FIR
- Exactly 50 coefficients are allowed

- The coefficients can be specified in double, but it has to be kept in mind that they are converted to Fix_24_24 by the application

Generating custom filters with the Matlab Filter Design and Analysis Tool

The Filter Design and Analysis Tool (FDATool) is a powerful user interface for designing and analyzing filters. FDATool enables to quickly design digital FIR filters by setting filter performance specifications, by importing filters from the MATLAB workspace, by directly specifying filter coefficients, or by adding, moving or deleting poles and zeros. FDATool also provides tools for analyzing filters, such as magnitude and phase response plots and pole-zero plots. FVTool, which can be launched from FDATool, provides a separate window for analyzing filters. FDATool can be used as a convenient alternative to the command line filter design functions.

Below is a brief introduction to how to design a bandpass FIR with FDATool, which should give a better understanding how designing filters can be done with it. For more information please refer to Matlab help.

The FDATool is opened with typing *fdatool* in the Matlab command window. Doing that will display the following panel:

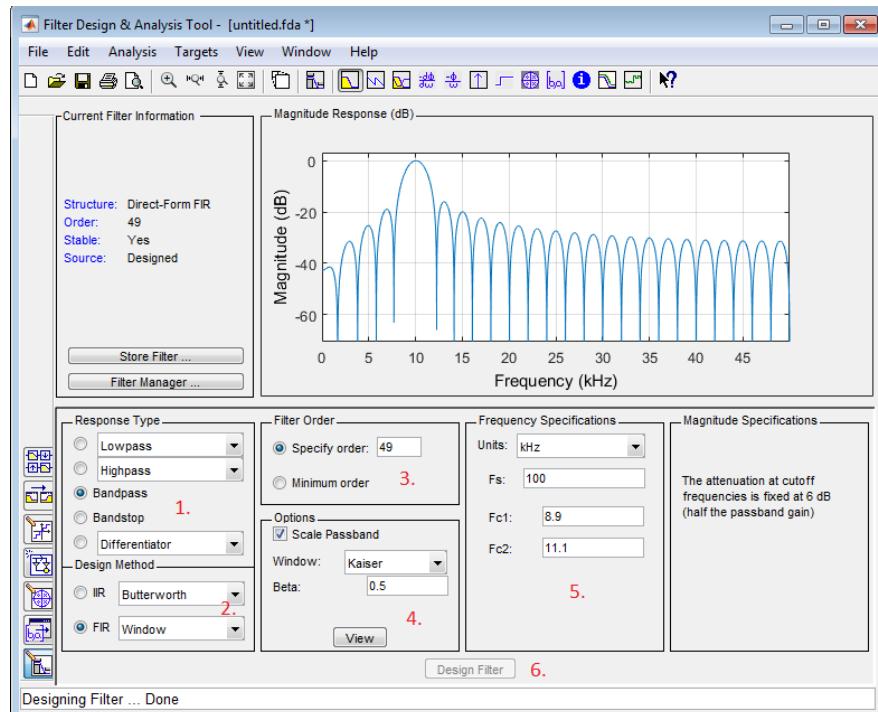


Figure 63: Designing filters with FDATool

In it the filter can be designed. The red numbers should give a better understanding on what to do.

Choose a response type (1):

The user can choose from the following response types. As this introduction shows how a bandpass has to be created, the radio button next to bandpass is selected.

Choose a filter design method (2):

Now a filter design method has to be chosen. The FDATool can create several Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filter algorithms. The KNOCK_DETECTION library block only allows FIR filters though, so FIR is what has to be chosen here. There are several methods to implement said FIR filter, including Equiripple,

Least-squares, Window and Interpolated. For the purpose of this tutorial Window is chosen.

Choose a filter order (3):

After that the filter order has to be specified. The user can choose between letting Matlab determine the minimum order or to choose the order themselves. As the bandpass has to conform to the implementation of the FIR filter in the KNOCK_DETECTION library block, order 49 has to be specified here. This will later generate exactly 50 filter coefficients.

Set filter options (4):

Depending on which design methods were chosen earlier, in that menu you can set further options. In this case the options for the Window FIR implementation are shown. Here the user is able to choose the exact Window filter implementation. Further options are displayed depending on the chosen implementation. In this tutorial a Kaiser window with a Beta value of 0.5 will be implemented.

Set frequency specifications (5):

Before the filter can be designed, the sampling frequency as well as the cutoff frequencies have to be chosen. The user has the choice to do that in normalized (0 ... 1) units, Hz, kHz, MHz or GHz. The Kaiser Window implementation needs two cutoff frequencies. The chosen sampling frequency has to be 100 kHz according to the KNOCK_DETECTION implementation.

Design the filter (6):

After all filter implementation details are specified, now it's time to press the *Design Filter* button and the filter will be designed by Matlab. Choosing *Analysis->Filter Coefficients* from the context menu will show all calculated coefficients, which the KNOCK_DETECTION application will need to build this filter.

Export the coefficients to the Matlab workspace:

Last but not least, the calculated filter coefficients have to be made available outside the context of FDAtool. This can be done with the export functionality under *File->Export...* in the context menu. With that an array variable for the workspace can be specified, in which the coefficients will be exported. This variable then can be used in the processor application to load this coefficients in the KNOCK_DETECTION FPGA implementation.

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

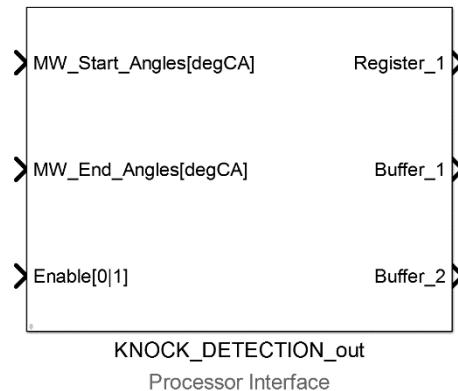


Figure 64: KNOCK_DETECTION_out block

Block Dialog

The processor output block provides the following dialogs:

- General Settings
- Filter Settings

General Settings

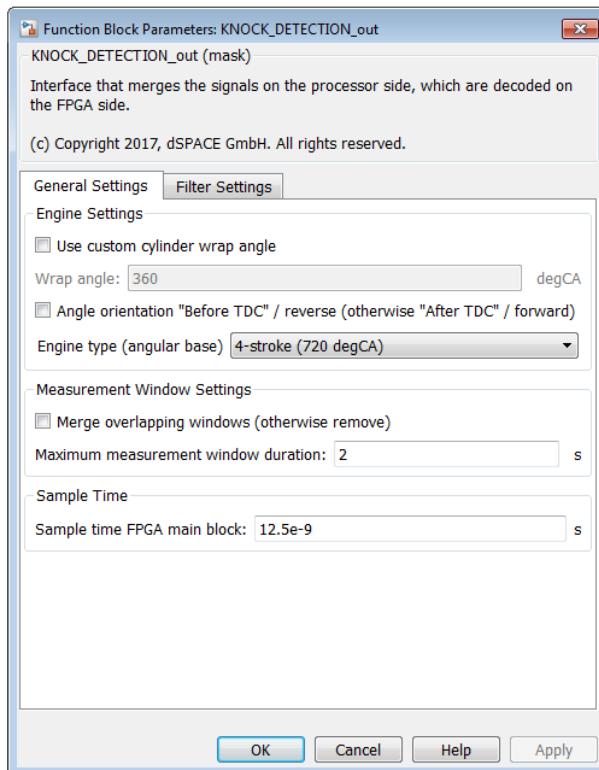


Figure 65: KNOCK_DETECTION_out dialog (General Settings)

The *General Settings* tab has the following parameters:

Name	Unit	Description	Range/Resolution
Use custom cylinder wrap angle	°CA	If selected, a custom wrap angle can be specified for the cylinder angles. Otherwise the default wrap angle (e.g. 360° for 4-stroke engines) is applied.	Range: 0°...1080° Resolution: 0.004°
Angle orientation "Before TDC"	-	If this option is checked the direction of the cylinder angles is reverse to the direction of the engine angle.	on / off
Engine type (angular base)	-	The engine type can be specified here. The angle ranges depend on the engine type (360°/720°/1080°)	2-stroke (360°CA) 4-stroke (720°CA) 6-stroke (1080°CA)
Merge overlapping windows	-	If this option is set, overlapping measurement windows will be merged. Otherwise overlapping measurement windows will be removed.	on / off
Maximum measurement window duration	s	Safety function to disable measurement windows after a specified maximum duration.	Range: 0...17.17s Resolution: 8ns (*)
Sample time FPGA main block	s	The sample time the FPGA main component is clocked with. Usually this is the FPGA clock rate, however in case of downsampling of the FPGA component this parameter has to be adapted.	-

(*) Assuming FPGA sample time of 8ns. For other sample times, the ranges and resolutions scale accordingly.

	The information about engine and cylinder wrap angles are required by several components. As the wrap angles are usually the same for all components, it is recommended to use a variable for these parameters.
---	---

Filter Settings

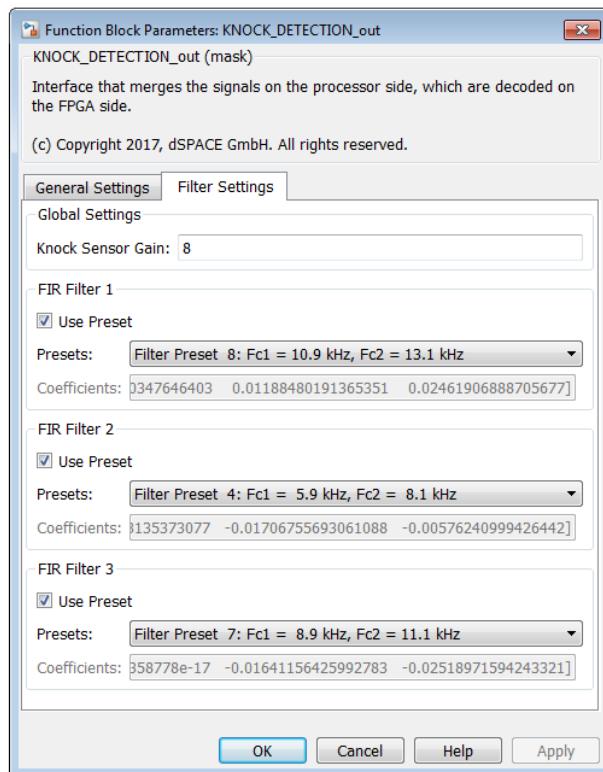


Figure 66: KNOCK_DETECTION_out dialog (Filter Settings)

The *Filter Settings* tab has the following parameters:

Name	Unit	Description	Range/Resolution
Knock Sensor Gain	-	Specifies the gain which is applied to the knock sensor input.	Range: 0...63 Resolution: 1
Use Preset (FIR Filter 1-3)	-	If set the user can choose one of 10 filter presets, whose coefficients are then transferred to the FIR implementation. If disabled the user can set those coefficients themselves in the coefficient field.	on/off
Presets (FIR Filter 1-3)	-	Drop down menu for the filter presets. Please refer to the paragraph <i>FIR filter coefficients</i> at the beginning of this document for further details.	Range: 10 presets
Coefficients (FIR Filter 1-3)	-	Defines all 50 FIR coefficients. The input expects an array of exactly 50 values. Please refer to the paragraph <i>FIR filter coefficients</i> at the beginning of the KNOCK_DETECTION chapter for further details.	Range: double

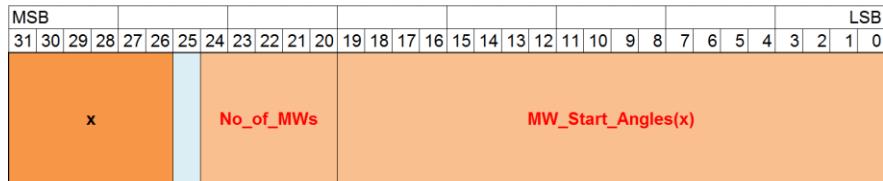
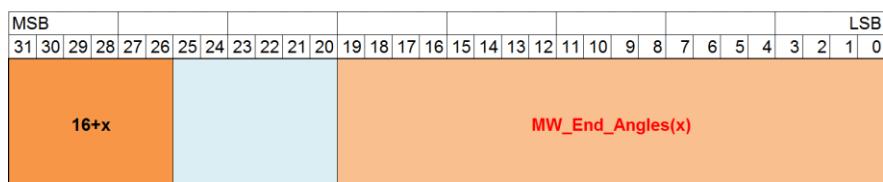
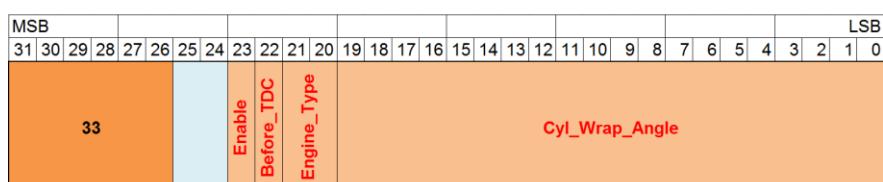
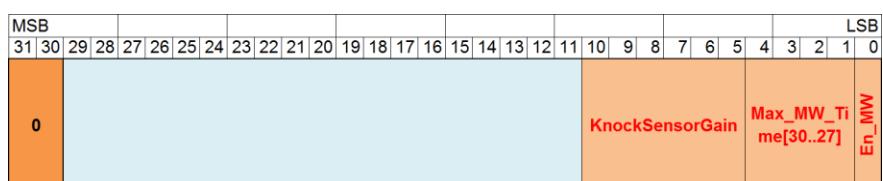
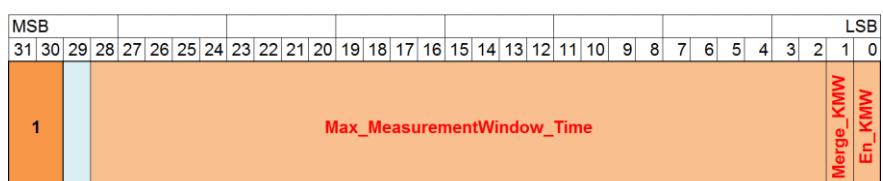
Input

The KNOCK_DETECTION_out block has the following inputs:

Name	Unit	Description	Range
MW_Start_Angles	°CA	Start angles for measurement windows (up to 16 elements)	Range: 0°...1080° Resolution: 0.004°
MW_End_Angles	°CA	End angles for measurement windows (up to 16 elements)	Range: 0°...1080° Resolution: 0.004°
Enable	-	Enables the knock amplitude calculation.	Range: 0 1

Output

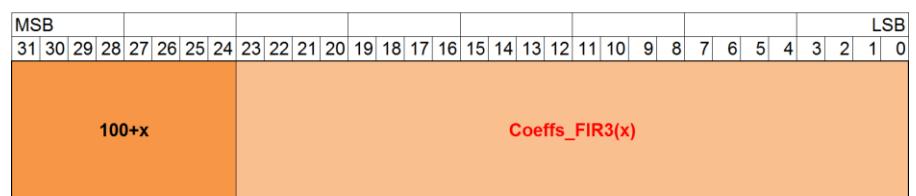
The Processor Out block outputs up to 33 register contents, mapped to 1 buffer. The actual amount of data is dependent on the settings and the length of the input vectors. All data is transferred each simulation step. Additionally another buffer with 150 register contents is output for the FIR coefficients. At last another 2 registers mapped to one register with further information are output as well.

Buffer1: Register 1...16**Buffer 1: Register 17...32****Buffer 1: Register 33****Register 1.1****Register 1.2****Buffer 2: Register 1...50**

Buffer 2: Register 51...100



Buffer 2: Register 101...150



FPGA Main Component

Block

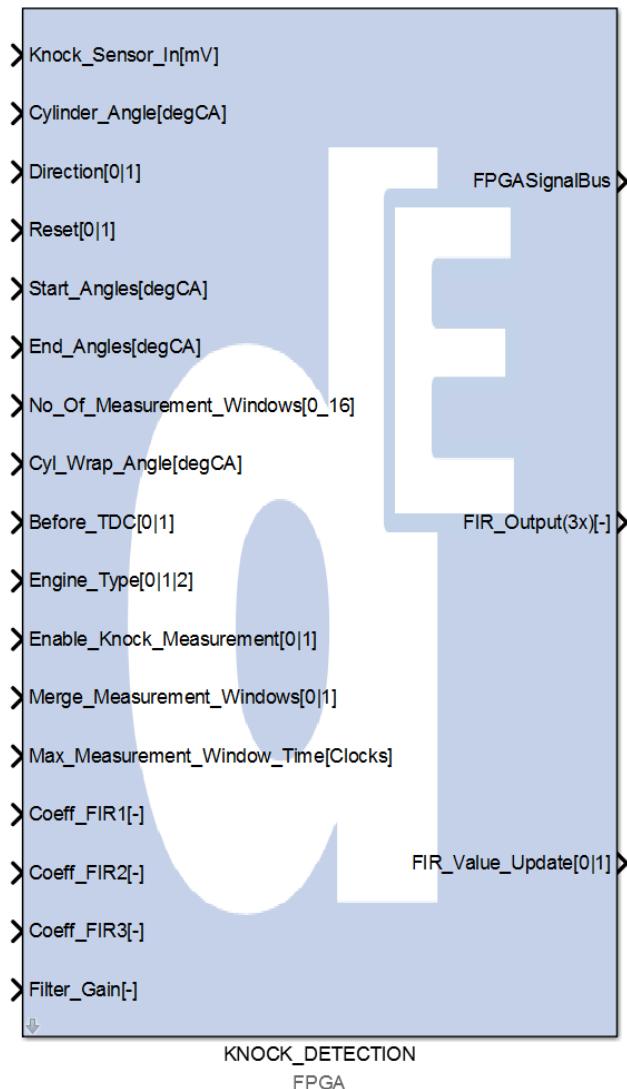


Figure 67: KNOCK_DETECTION FPGA Main block

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Knock_Sensor_In	mV	The input of the knock sensor.	Fix_25_8
Cylinder_Angle	°CA	The cylinder angle derived from the engine angle and the TDC.	Fix_20_8
Direction	-	The direction of the engine motion.	Bool
Reset	-	Resets the angle selection.	Bool
Start_Angles	°CA	Start angles for measurement windows (16 elements).	Fix_20_8 (16x)
End_Angles	°CA	End angles for measurement windows (16 elements).	Fix_20_8 (16x)
No_of_Measurement_Windows	-	Number of actually used measurement windows.	UFix_5_0
Cyl_Wrap_Angle	°CA	The wrap angle for the cylinder angle.	UFix_19_8
Before_TDC	-	If this signal is True, the direction of the cylinder angles is reverse to the direction of the engine angle.	Bool
Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0
Enable_KnockMeasurement	-	Enables the knock amplitude calculation.	Bool
MergeMeasurement_Windows	-	Activates the merging of overlapping measurement windows (otherwise overlapping measurement windows will be removed).	Bool
Max_Measurement_Window_Time	Clocks	The maximum valid pulse duration. Measurement windows will be stopped after exceeding this time.	UFix_31_0
Coeff_FIR1	-	Vector of 50 coefficients for FIR filter 1.	UFix_24_0 (50)
Coeff_FIR2	-	Vector of 50 coefficients for FIR filter 2.	UFix_24_0 (50)
Coeff_FIR3	-	Vector of 50 coefficients for FIR filter 3.	UFix_24_0 (50)
Filter_Gain	-	Filter gain value of the sensor input.	UFix_6_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
FIR_Output(3x)	-	Vector containing all three calculated knock amplitudes of the FIRs.	3x UFix_31_0
FIR_Value_Update	-	Is active on update of the output registers.	Bool

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	5083	1.25%
LUTs	3900	1.91%
Block RAM	4	0.45%
DSP Slices	10	1.19%

Processor Input

Block Adapts the FPGA signals for the processor side.

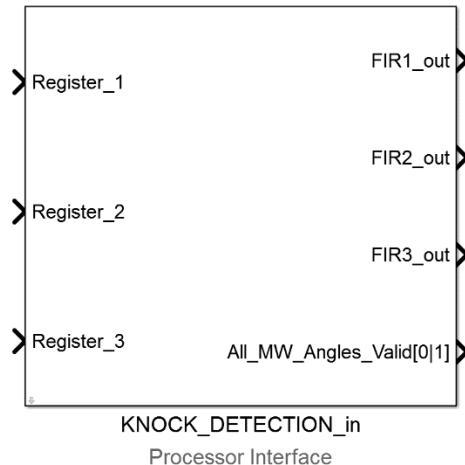
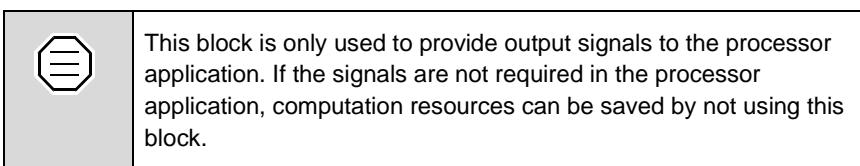


Figure 68: KNOCK_DETECTION_in block



Input The processor input block has 3 input register with a sectioning shown below.

Register 1



Register 2



Register 3

Output

The KNOCK_DETECTION_in block has the following outputs:

Name	Unit	Description	Range
FIR1_out	-	Latest knock amplitude of first FIR filter calculated by the FPGA block.	Range: 0...2 ³¹ -1 Resolution: 1
FIR2_out	-	Latest knock amplitude of second FIR filter calculated by the FPGA block.	Range: 0...2 ³¹ -1 Resolution: 1
FIR3_out		Latest knock amplitude of third FIR filter calculated by the FPGA block.	Range: 0...2 ³¹ -1 Resolution: 1
All_MW_Angles_Valid	-	True, if all angles of the measurement window are valid, false otherwise.	0 1

Auxiliary Components

Objective The sub-library *Auxiliary Components* contains multiple different functions which can be useful but are not necessarily required.

Content The section contains the following components:

- ENGINE_2_CYLINDER
- CYLINDER_2_ENGINE
- ANGLE_BASED_PULSE
- AAPB_SERIALIZER
- ANGLE_INTERRUPT
- THRESHOLD_INTERRUPT
- RAPID_PRO_INPUT

ENGINE_2_CYLINDER

Objective

This block transforms an engine input angle to cylinder angles, according to the cylinders' TDC. A wrap angle for engine angle and cylinder angles can optionally be specified.



Each transformation block is designed for 4 cylinders. If more than 4 cylinders are required, use the transformation block multiple times.

Content

The blockset contains the following elements:

- Processor Interface: ENGINE_2_CYLINDER_out (Processor Interface)
- FPGA Interface: ENGINE_2_CYLINDER_in (FPGA Interface)
- FPGA: ENGINE_2_CYLINDER (FPGA Main Component)
- FPGA Interface: ENGINE_2_CYLINDER_out (FPGA Interface)
- Processor Interface: ENGINE_2_CYLINDER_in (Processor Interface)

Coordinate Systems

There are two main angle value representations in use:

The engine-position based angle value is related to the (zero reference) global engine position. Its range covers a complete combustion cycle. The engine-position based angle value is same for all cylinders. According to this, it is independent of the actual cylinder number, total cylinder number and their mechanical arrangement. Its value is periodical in range of 0°...360° for a 2-stroke-engine, in range of 0°...720° for a 4-stroke engine and in range of 0°...1080° for a 6-stroke engine.

The cylinder-based angle value is given related to the Top Dead Center (TDC) of the actual cylinder. It is individual for each particular cylinder. The cylinder-based angle value depends hence on the actual cylinder number as well as on the total cylinder number and their mechanical arrangement (e.g. series or V). The cylinder-based angle value is also a periodical value and it is mostly represented in range -360°...360° with the same or sometimes opposite direction reference as the engine-position based angle value.

The figure below shows an example for a 4-stroke 4-cylinder engine. The circular depiction is chosen just to underline the periodic nature of the engine angle. A full circle represents a complete combustion period (720° in this case). The TDC of the first cylinder is here 90° (TDC1), and the cylinder-1-related coordinate system has its zero point at 90° in the engine-related coordinate system. The TDC of the second cylinder is set to 270°, where the cylinder-2-related coordinate system's origin is set.

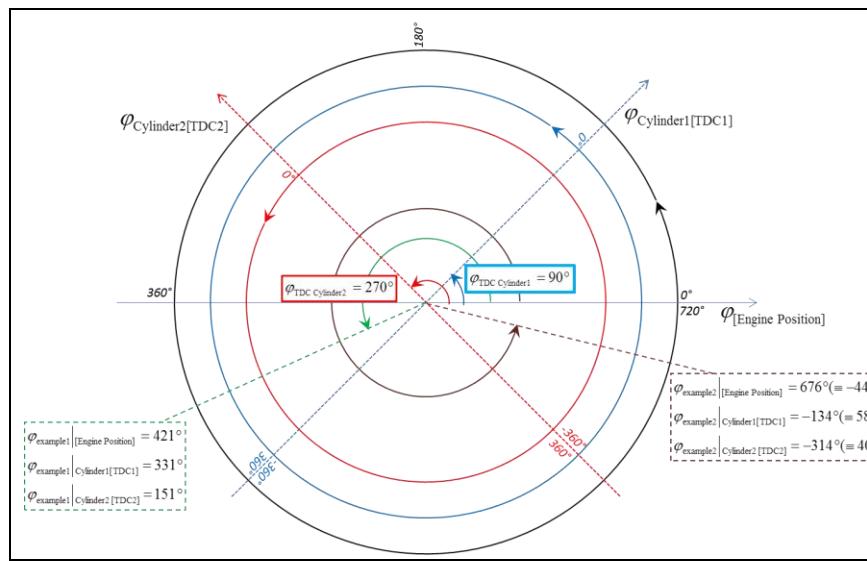


Figure 69: Example of cylinder angles in relation to the engine angle

Wrap Angles

Certain wrap angles can be defined for both engine and cylinder angles. As e.g. in case of a 4-stroke engine the angle range is $0^\circ \dots 720^\circ$ by default, the usual wrap angle is 720° . For a cylinder in a 4-stroke engine the default angle range is $-360^\circ \dots 360^\circ$, the usual wrap angle is 360° . However, if for example a wrap angle of 180° is defined, the angle range in a 4-stroke engine changes to $-540^\circ \dots 180^\circ$. This principle is illustrated in Figure 70.

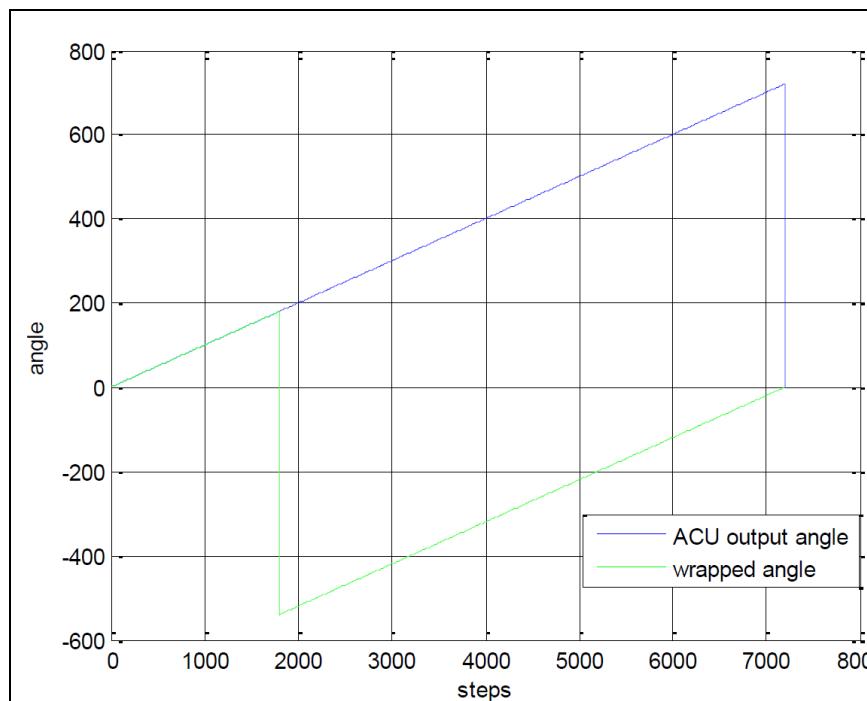


Figure 70: Example of a 4-stroke engine angle, wrapped at 180°

Processor Output

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



Figure 71: ENGINE_2_CYLINDER_out block

Block Dialog The processor output block provides the following dialog:

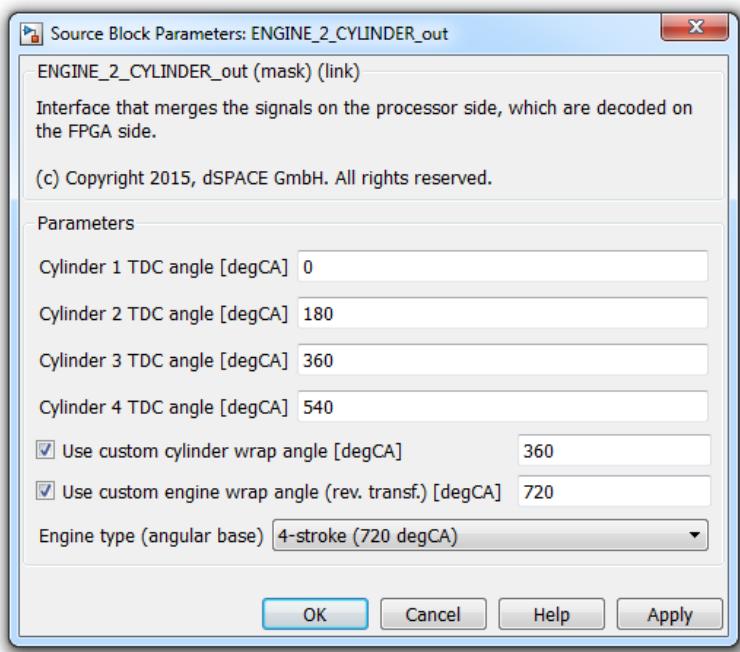


Figure 72: ENGINE_TO_CYLINDER_out dialog

The block dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Cylinder 1 TDC angle	°CA	TDC of the 1 st cylinder	Range: 0°...1080° Resolution: 0.004°
Cylinder 2 TDC angle	°CA	TDC of the 2 nd cylinder	Range: 0°...1080° Resolution: 0.004°
Cylinder 3 TDC angle	°CA	TDC of the 3 rd cylinder	Range: 0°...1080° Resolution: 0.004°
Cylinder 4 TDC angle	°CA	TDC of the 4 th cylinder	Range: 0°...1080° Resolution: 0.004°
Use custom cylinder wrap angle	°CA	If selected, a custom wrap angle can be specified for the cylinder angles. Otherwise the default wrap angle (e.g. 360° for 4-stroke engines) is applied.	Range: 0°...1080° Resolution: 0.004°
Use custom engine wrap angle	°CA	If selected, a custom wrap angle can be specified for the engine angle. Otherwise the default wrap angle (e.g. 720° for 4-stroke engines) is applied.	Range: 0°...1080° Resolution: 0.004°
Angle orientation "Before TDC"	-	If this option is checked the direction of the cylinder angles is reverse to the direction of the engine angle.	on / off
Engine type (angular base)	-	The engine type can be specified here. The angle ranges depend on the engine type (360°/720°/1080°).	2-stroke (360°CA) 4-stroke (720°CA) 6-stroke (1080°CA)

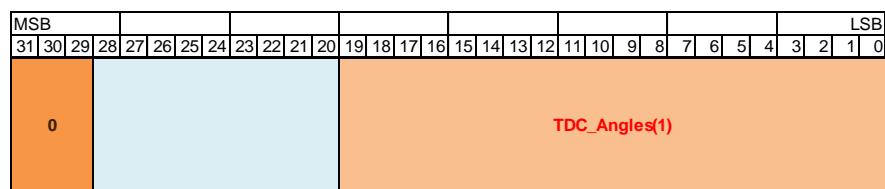


The information about engine and cylinder wrap angles are required by several components. As the wrap angles are usually the same for all components, it is recommended to use a variable for these parameters.

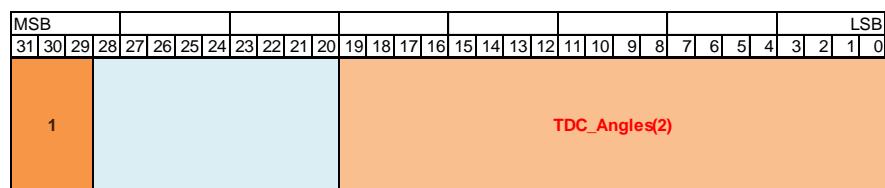
Output

The Processor Out block outputs 8 register contents, mapped to 1 register by time multiplexing. The sectioning is shown below:

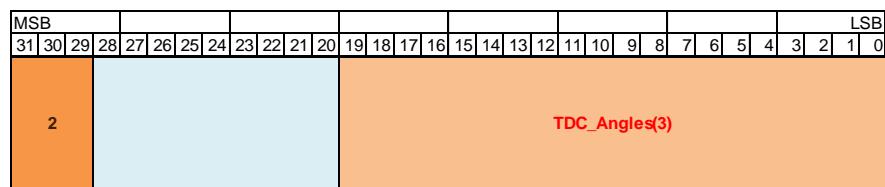
Register 1.1



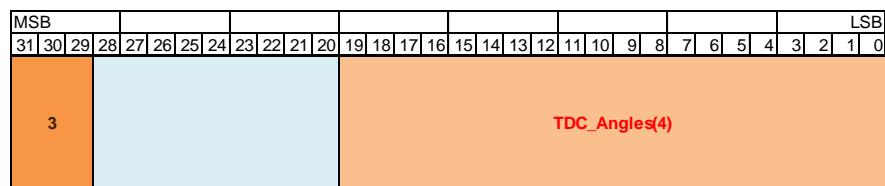
Register 1.2



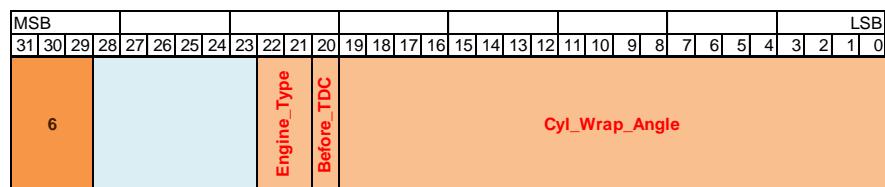
Register 1.3



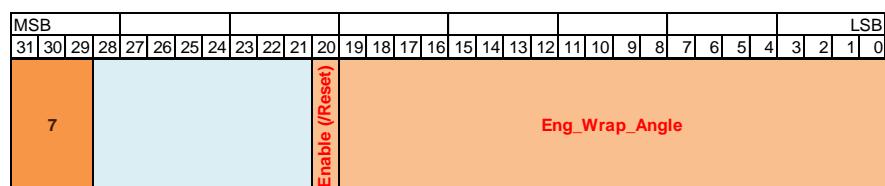
Register 1.4



Register 1.5



Register 1.6



FPGA Main Component

Block

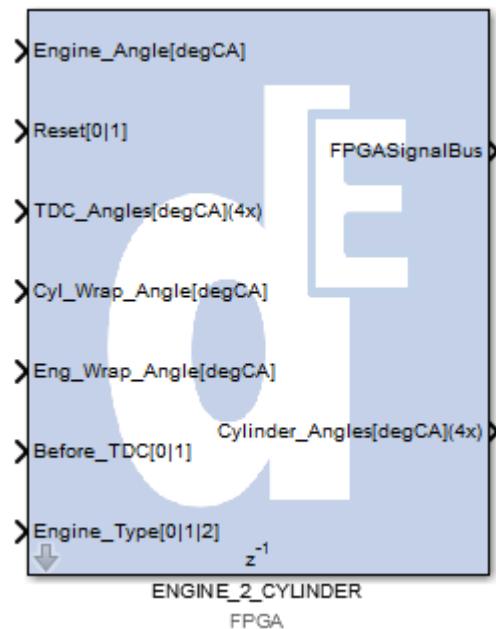


Figure 73: ENGINE_2_CYLINDER FPGA Main block

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Engine_Angle	°CA	The engine angle calculated by the ACU or encoder processing.	Fix_20_8
Reset	-	Resets the angle transformation	Bool
TDC_Angles	°CA	The TDC angles for each cylinder.	Fix_20_8 (4x)
Cyl_Wrap_Angle	-	The wrap angle for all cylinder angles	UFix_19_8
Eng_Wrap_Angle	-	The wrap angle for the engine angle.	UFix_19_8
Before_TCD	-	Determines the orientation of the cylinder angles. (0 = same direction as the engine angles, 1 = reverse direction than the engine angle).	Bool
Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0



The information about engine and cylinder wrap angles are required by several components. As the wrap angles are usually the same for all components, it is recommended to use a variable for these parameters.

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
Cylinder_Angles	°CA	The calculated cylinder angles according to TDC angles and wrap angle.	Fix_20_8 (4x)

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	319	0.08%
LUTs	824	0.40%
Block RAM	0	0.00%
DSP Slices	0	0.00%

Processor Input

Block Adapts the FPGA signals for the processor side.

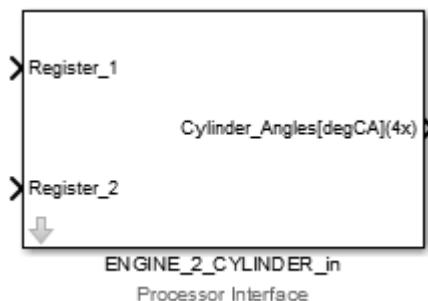
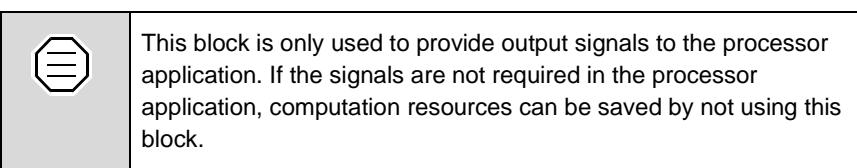
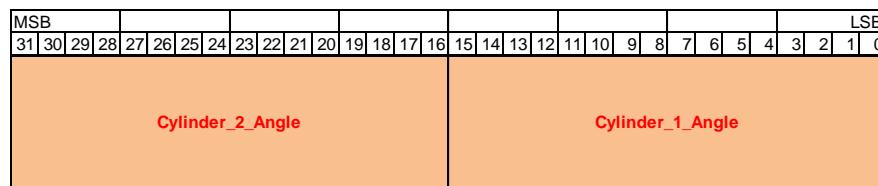


Figure 74: ENGINE_2_CYLINDER_in block

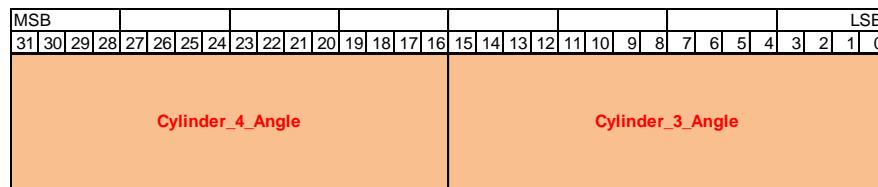


Input The processor input block has 2 input register with a sectioning shown below.

Register 1



Register 2



Output The ENGINE_2_CYLINDER_in block has the following outputs:

Name	Unit	Description	Range
Cylinder_Angles	°CA	The cylinder angles derived from the engine angle.	Range: ±1080° Resolution: 0.06°

CYLINDER_2_ENGINE

Objective

This block transforms cylinder input angle to engine angle, according to the cylinders' TDC. A wrap angle for engine angle and cylinder angles can optionally be specified.



The cylinder 2 engine component shares its input with the CYLINDER_2_ENGINE component. Therefore it can directly be connected to the ENGINE_2_CYLINDER_out FPGA block!



Each transformation block is designed for 4 cylinders. If more than 4 cylinders are required, use the transformation block multiple times.

Content

The blockset contains the following elements:

- FPGA: CYLINDER_2_ENGINE
(FPGA Main Component)
- FPGA Interface: CYLINDER_2_ENGINE_out
(FPGA Interface)
- Processor Interface: CYLINDER_2_ENGINE_in
(Processor Interface)

FPGA Main Component

Block

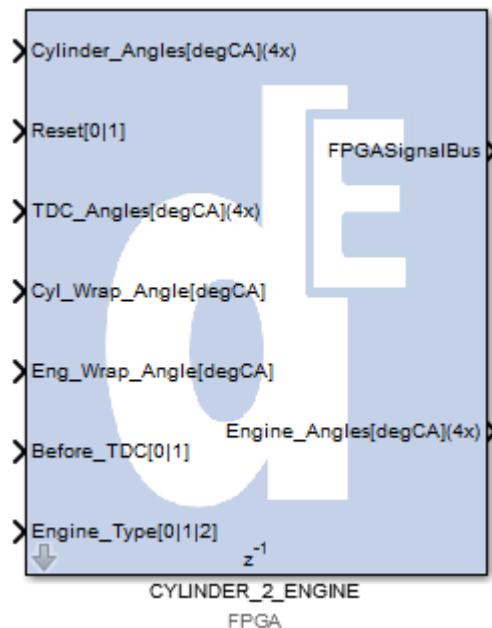


Figure 75: CYLINDER_2_ENGINE FPGA Main block

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Cylinder_Angles	°CA	The engine angle calculated by the ACU or encoder processing.	Fix_20_8 (4x)
Reset	-	Resets the angle transformation	Bool
TDC_Angles	°CA	The TDC angles for each cylinder.	Fix_20_8 (4x)
Cyl_Wrap_Angle	-	The wrap angle for all cylinder angles	UFix_19_8
Eng_Wrap_Angle	-	The wrap angle for the engine angle.	UFix_19_8
Before_TCD	-	Determines the orientation of the cylinder angles. (0 = same direction as the engine angles, 1 = reverse direction than the engine angle).	Bool
Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
Engine_Angles	°C	The back-transformed engine angles according to TDC angles and wrap angle.	Fix_20_8 (4x)

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	442	0.11%
LUTs	916	0.44%
Block RAM	0	0.00%
DSP Slices	0	0.00%

Processor Input

Block Adapts the FPGA signals for the processor side.

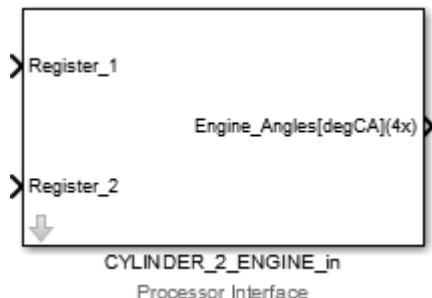


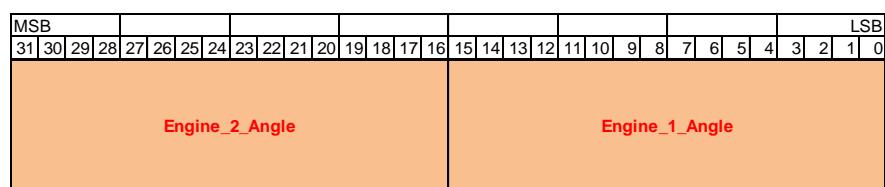
Figure 76: CYLINDER_2_ENGINE_in block



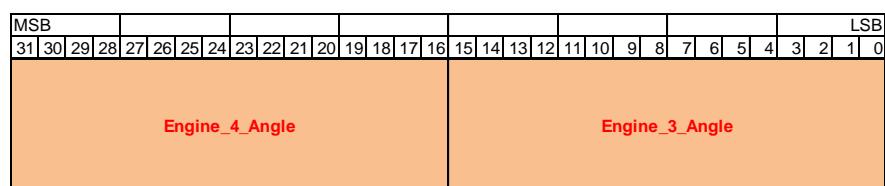
This block is only used to provide output signals to the processor application. If the signals are not required in the processor application, computation resources can be saved by not using this block.

Input The processor input block has 3 input register with a sectioning shown below.

Register 1



Register 2



Output The CYLINDER_2_ENGINE_in block has the following outputs:

Name	Unit	Description	Range
Engine_Angles	°CA	The engine angles derived via back transformation from the cylinder angles.	Range: ±1080° Resolution: 0.06°

ANGLE_BASED_PULSE

Objective

The angle-angle-based pulse generates additional pulses according to the calculated next start and end angles. If pulses overlap, the pulses can be merged or removed, according to the user's choice.



For detection of start or end of injection pulses, it is required that the next pulse angles cross the cylinder angle for at least one FPGA clock period (see [INJECTION IGNITION CORE](#)).
If the angle serializer is used for the next angle inputs, all angle inputs (including cylinder angle) can directly be connected without any delays to be inserted.

Content

The blockset contains the following elements:

- Processor Interface: ANGLE_BASED_PULSE_out
(Processor Interface)
- FPGA Interface: ANGLE_BASED_PULSE_in
(FPGA Interface)
- FPGA: ANGLE_BASED_PULSE
(FPGA Main Component)

Processor Output

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

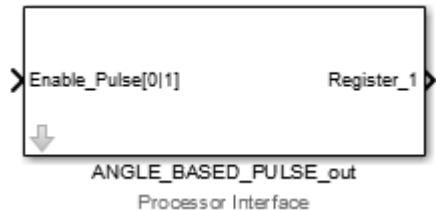


Figure 77: ANGLE_BASED_PULSE_out block

Block Dialog

The processor output block provides the following dialog:

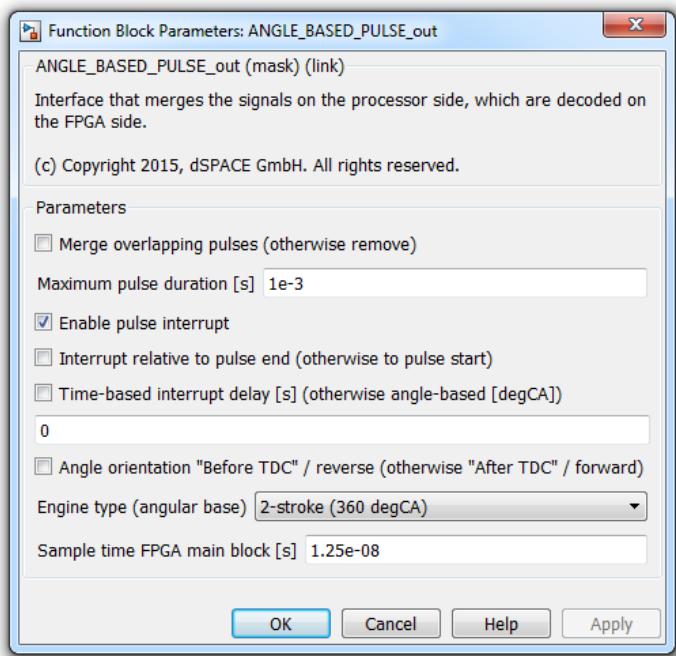


Figure 78: ANGLE_BASED_PULSE_out dialog

The block dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Merge overlapping pulses	-	If this option is set, overlapping pulses will be merged. Otherwise overlapping pulses will be removed.	on / off
Maximum pulse duration	s	Safety function to disable pulses after a specified maximum duration.	Range: 0...17.17s Resolution: 8ns (*)
Enable interrupt		If this option is set, an interrupt will be triggered upon generated pulses.	on / off
Interrupt relative to pulse end	-	If this option is set, the interrupt is triggered after the end of the pulse (and a specified delay). Otherwise it is triggered after the start of the pulse.	on / off
Time based interrupt delay	-	If this option is set, an interrupt time-based delay relative to the pulse start can be specified. Otherwise, an angle-based delay can be specified.	on / off
Interrupt delay	s °CA	The value of the interrupt delay (either angle or time) can be specified here.	Range: 0°...1080° 0...0.13s(*) Resolution: 0.004° 8ns(*)
Angle orientation "Before TDC"	-	If this option is checked the direction of the cylinder angles is reverse to the direction of the engine angle.	on / off
Engine type (angular base)	-	The engine type can be specified here. The angle ranges depend on the engine type (360°/720°/1080°).	2-stroke (360°CA) 4-stroke (720°CA) 6-stroke (1080°CA)
Sample time FPGA main block	s	The sample time the FPGA main component is clocked with. Usually this is the FPGA clock rate, however in case of downsampling of the FPGA component this parameter has to be adapted.	-

(*) Assuming FPGA sample time of 8ns. For other sample times, the ranges and resolutions scale accordingly.

Input

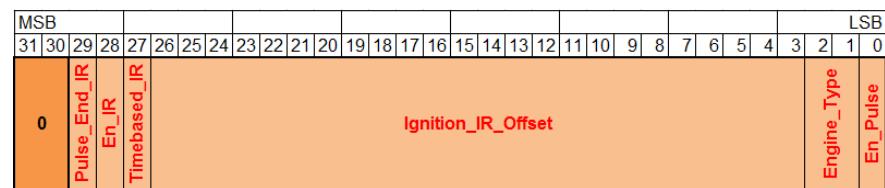
The ANGLE_BASED_PULSE_out block has the following inputs:

Name	Unit	Description	Range
Enable_Pulse	-	The generation of pulses can be activated or deactivated during runtime using this input	0 1

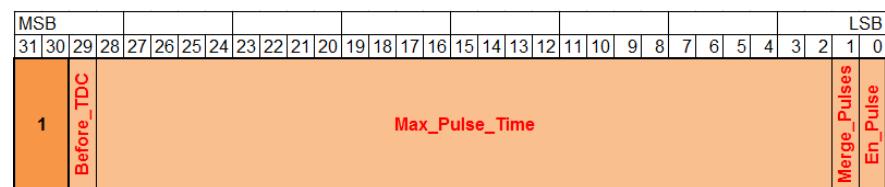
Output

The Processor Out block outputs to 2 register contents mapped to 1 register. The mapping is shown below.

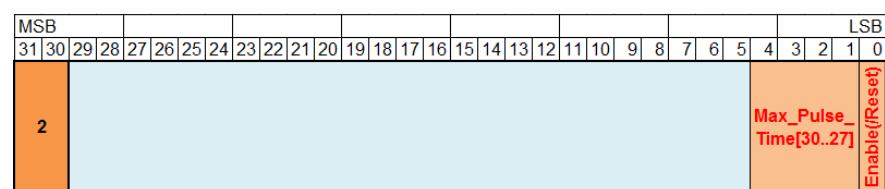
Register 1.1



Register 1.2



Register 1.3



FPGA Main Component

Block

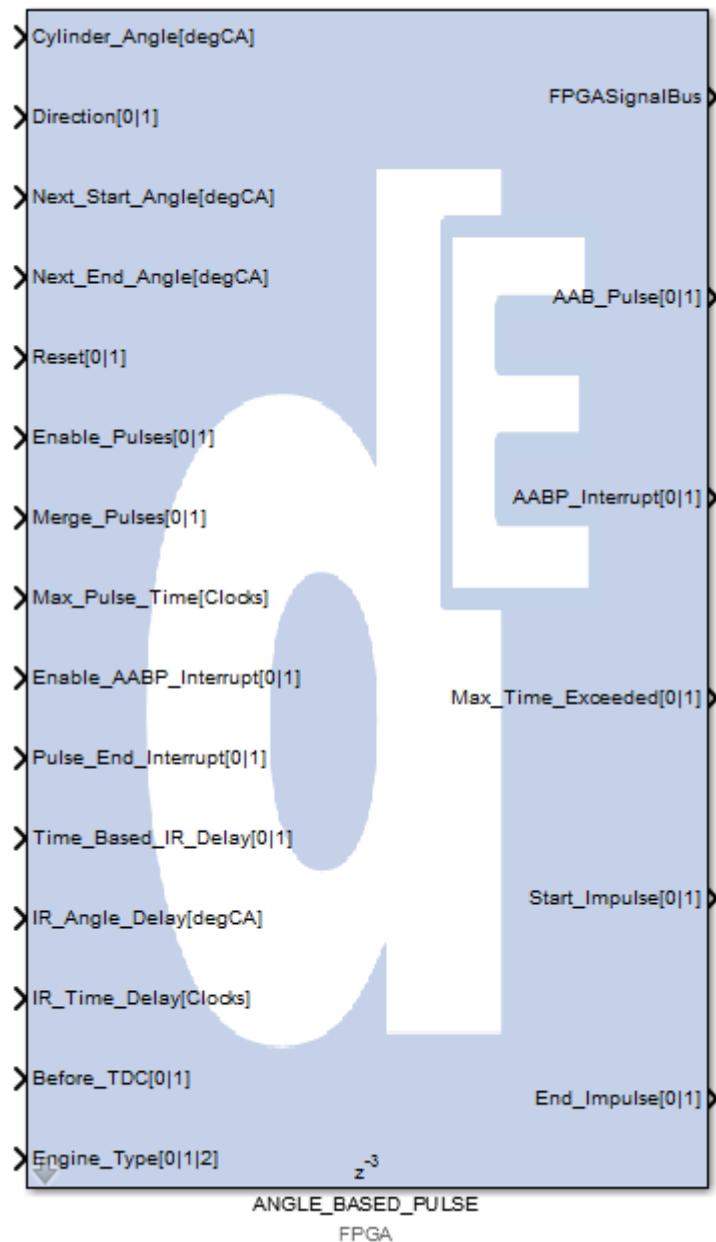


Figure 79: ANGLE_BASED_PULSE FPGA Main block

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Cylinder_Angle	°CA	The cylinder angle derived from the engine angle and the TDC.	Fix_20_8
Direction	-	The direction of the engine motion.	Bool
Next_Start_Angle	°CA	The start angle of the next pulse.	Fix_20_8
Next_End_Angle	°CA	The end angle of the next pulse.	Fix_20_8
Reset	-	Resets the pulse generation.	Bool
Enable_Pulses	-	Enables or disables the generation of pulses during runtime.	Bool
Merge_Pulses	-	Activates the merging of overlapping pulses (otherwise overlapping pulses will be removed).	Bool
Max_Pulse_Time	Clocks	The maximum valid pulse duration. Pulses will be stopped after exceeding this time.	UFix_31_0
Enable_AABP_Interrupt	-	Enables the generation of interrupts on the rising edge of the pulse.	Bool
Pulse_End_Interrupt	-	Interrupt is relative to pulse end (otherwise pulse start)	Bool
Time_Based_Ign_IR_Delay	-	The delay of the interrupt is set as a time in s.	Bool
Ign_IR_Angle_Delay	°CA	The time delay for the interrupt generation.	Fix_20_8
Ign_IR_Time_Delay	s	The angle delay for the interrupt generation.	UFix_20_20
Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
AAB_Pulse	-	The generated angle-based pulses.	Bool
AABP_Interrupt	-	The generated interrupt impulse.	Bool
Max_Time_Exceeded	-	Error flag indicating that the last pulse had to be stopped after exceeding the maximum duration. Will be reset after an ignition pulse ended without exceeding the maximum duration.	Bool
Start_Impulse	-	Impulse (1 clock cycle) indicating that the start of a pulse has been detected.	Bool
End_Impulse	-	Impulse (1 clock cycle) indicating that the end of a pulse has been detected.	Bool

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	274	0.07%
LUTs	559	0.27%
Block RAM	0	0.00%
DSP Slices	0	0.00%

AABP_SERIALIZER

Objective

This component selects the next angle-angle-based pulse start and end angles, according to the start and end angles defined by the user and the current cylinder angle.

	The angle input vectors for injection or ignition must be of ascending order. However the starting point does not matter. For example [-100°, 0°, 100°] as well as [0°, 100°, -100°] are valid angle inputs, while [0°, -100°, 100°] or [100°, 0°, -100°] are invalid inputs!
	If a pulse completely covers another pulse, this input is invalid as well (e.g. first pulse 0°-100°, second pulse 20°-80°).
	The minimum distance between two start or end angles is 0.1° CA.

Content

The blockset contains the following elements:

- Processor Interface: AABP_SERIALIZER_out (Processor Interface)
- FPGA Interface: AABP_SERIALIZER_in (FPGA Interface)
- FPGA: AABP_SERIALIZER (FPGA Main Component)
- FPGA Interface: AABP_SERIALIZER_out (FPGA Interface)
- Processor Interface: AABP_SERIALIZER_in (Processor Interface)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

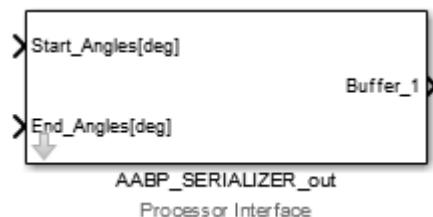


Figure 80: AABP_SERIALIZER_out block

Block Dialog

The processor output block provides the following dialog:

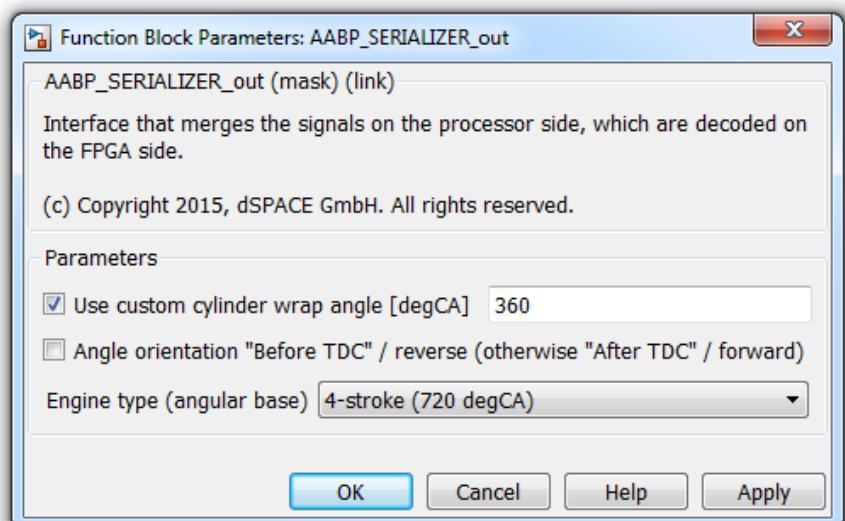


Figure 81: AABP_SERIALIZER_out dialog

The blockset dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Use custom cylinder wrap angle	°CA	If selected, a custom wrap angle can be specified for the cylinder angles. Otherwise the default wrap angle (e.g. 360° for 4-stroke engines) is applied.	Range: 0°...1080° Resolution: 0.004°
Angle orientation "Before TDC"	-	If this option is checked the direction of the cylinder angles is reverse to the direction of the engine angle.	on / off
Engine type (angular base)	-	The engine type can be specified here. The angle ranges depend on the engine type (360°/720°/1080°).	2-stroke (360°CA) 4-stroke (720°CA) 6-stroke (1080°CA)



The information about engine and cylinder wrap angles are required by several components. As the wrap angles are usually the same for all components, it is recommended to use a variable for these parameters.

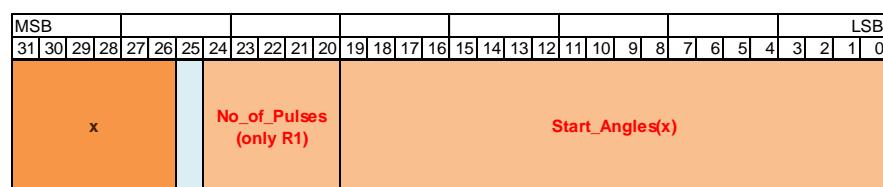
Input

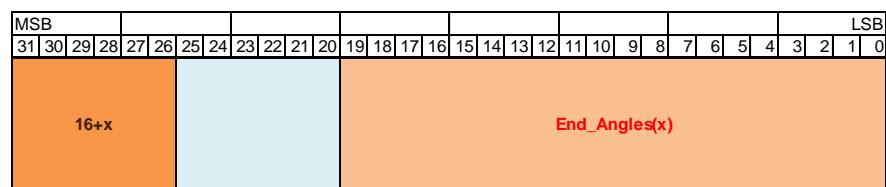
The AABP_SERIALZER_out block has the following inputs:

Name	Unit	Description	Range
Start_Angles	°CA	Start angles for pulses (up to 16 elements)	Range: 0°...1080° Resolution: 0.004°
End_Angles	°CA	End angles for pulses (up to 16 elements)	Range: 0°...1080° Resolution: 0.004°

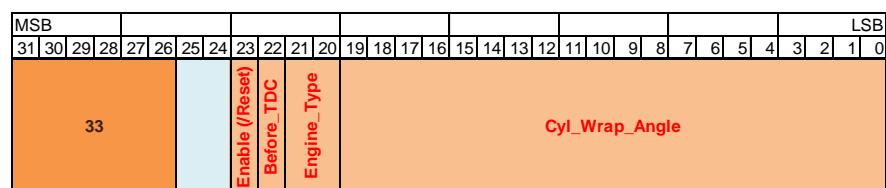
Output

The Processor Out block outputs up to 33 register contents, mapped to 1 buffer. The actual amount of data is dependent on the settings and the length of the input vectors. All data is transferred each simulation step.

Register 1...16**Register 17...32**



Register 33



FPGA Main Component

Block

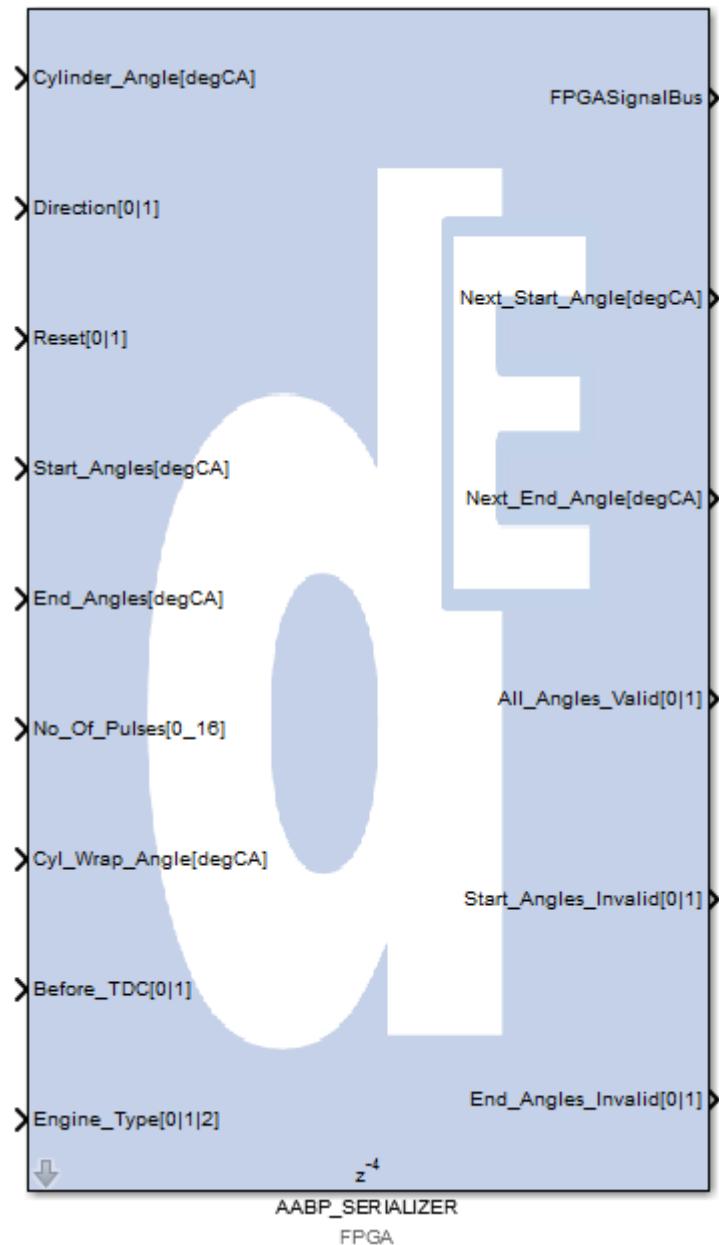


Figure 82: AABP_SERIALIZER FPGA Main block

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Cylinder_Angle	°CA	The cylinder angle derived from the engine angle and the TDC.	Fix_20_8
Direction	-	The direction of the engine motion.	Bool
Reset	-	Resets the angle selection.	Bool
Start_Angles	°CA	Start angles for pulses (16 elements).	Fix_20_8 (16x)
End_Angles	°CA	End angles for pulses (16 elements).	Fix_20_8 (16x)
No_of_Pulses	-	Number of actually configured ignition pulses.	UFix_5_0
Cyl_Wrap_Angle	°CA	The wrap angle for the cylinder angle.	UFix_19_8
Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
Next_Start_Angle	°CA	The next start angle, selected among all input angles, according to the current cylinder angle.	Fix_20_8
Next_End_Angle	°CA	The next ignition end angle, selected among all input angles, according to the current cylinder angle.	Fix_20_8
All_Angles_Valid	-	This signal indicates that all start and end input angles are valid. It is recommended to enable the pulse only if this signal is 1.	Bool
Start_Angles_Invalid	-	Vector of flags indicating if the corresponding pulse start angle is invalid (according to the rules mentioned at the beginning of the chapter).	Bool (16x)
End_Angles_Invalid	-	Vector of flags indicating if the corresponding pulse end angle is invalid (according to the rules mentioned at the beginning of the chapter).	Bool (16x)

FPGA Resources

The following FPGA (XC7K325T) resources are occupied by the component (including interface):

Type	Absolute Number	Percentage
Registers	3191	0.78%
LUTs	2984	1.46%
Block RAM	2	0.44%
DSP Slices	0	0.00%

Processor Input

Block	Adapts the FPGA signals for the processor side.
--------------	---

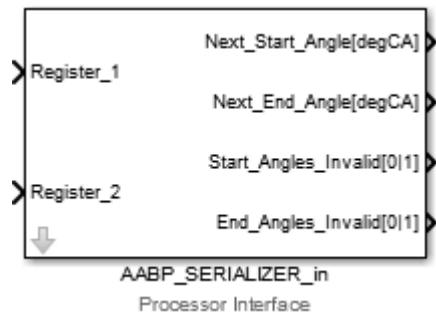


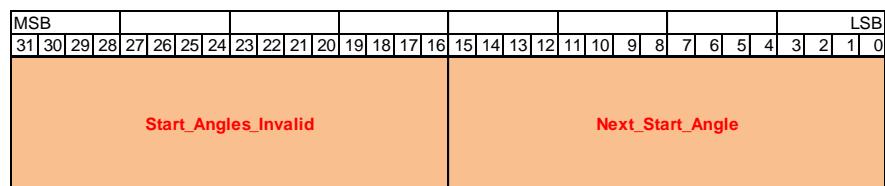
Figure 83: AABP_SERIALIZER_in block



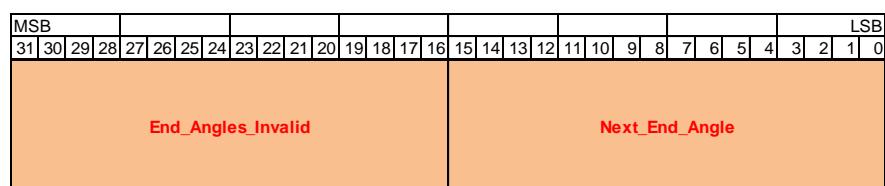
This block is only used to provide output signals to the processor application. If the signals are not required in the processor application, computation resources can be saved by not using this block.

Input	The processor input block has 2 input register with a sectioning shown below.
--------------	---

Register 1



Register 2



Output

The AABP_SERIALIZER_in block has the following outputs:

Name	Unit	Description	Range
Next_Start_Angle	°CA	See FPGA main component .	Range: ±1080° Resolution: 0.06°
Next_End_Angle	°CA	See FPGA main component .	Range: ±1080° Resolution: 0.06°
Start_Angles_Invalid	-	See FPGA main component .	0 1
End_Angles_Invalid	-	See FPGA main component .	0 1

ANGLE_INTERRUPT

Objective

The angle interrupt generates interrupts to the processor platform if a user-specified angle is passed.

Content

The blockset contains the following elements:

- Processor Interface: ANGLE_INTERRUPT_out
(Processor Interface)
- FPGA Interface: ANGLE_INTERRUPT_in
(FPGA Interface)
- FPGA: ANGLE_INTERRUPT
(FPGA Main Component)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

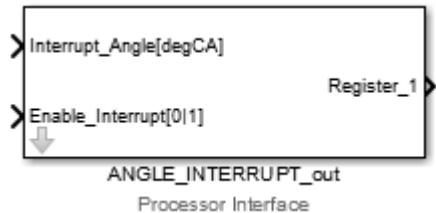


Figure 84: ANGLE_INTERRUPT_out block

Block Dialog

The processor output block provides the following dialog:

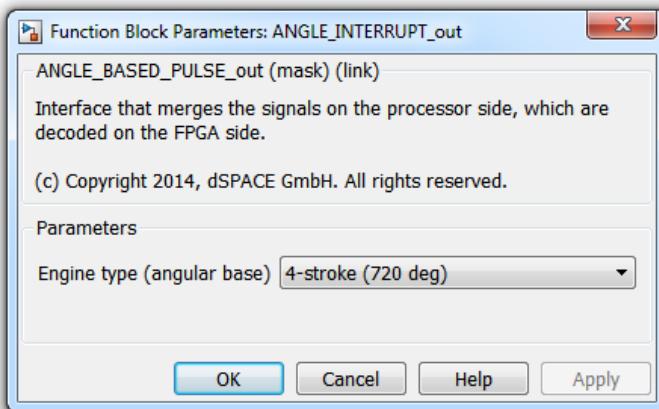


Figure 85: ANGLE_INTERRUPT_out dialog

The block dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Angle orientation “Before TDC”	-	If this option is checked the input angle decreases while in positive engine motion direction.	on / off
Engine type (angular base)	-	The engine type can be specified here. The angle ranges depend on the engine type (360°/720°/1080°).	2-stroke (360°CA) 4-stroke (720°CA) 6-stroke (1080°CA)

Input

The ANGLE_INTERRUPT_out block has the following inputs:

Name	Unit	Description	Range/Resolution
Interrupt_Angle	°CA		Range: ±1080° Resolution: 0.004°
Enable_Interrupt	-	The generation of interrupts can be activated or deactivated during runtime using this input	0 1

Output

The Processor Out block outputs 1 register. The mapping is shown below.

Register 1

FPGA Main Component

Block

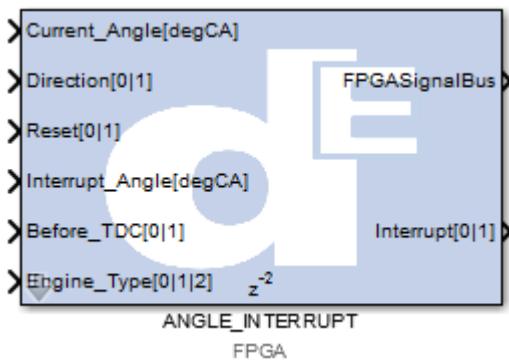


Figure 86: ANGLE_INTERRUPT FPGA Main block

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
Current_Angle	°CA	The angle to which the interrupt shall be generated.	Fix_20_8
Direction	-	The direction of the engine motion.	Bool
Reset	-	Resets the interrupt generation.	Bool
Interrupt_Angle	°CA	The angle at which an interrupt shall be generated.	Fix_20_8
Engine_Type	-	Select between 2-stroke (0), 4-stroke (1) or 6-stroke (2) engine.	UFix_2_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
Interrupt	-	The generated interrupt impulse.	Bool

FPGA Resources

The FPGA (XC7K325T) resources occupied by the component are negligible.

THRESHOLD_INTERRUPT

Objective The threshold interrupt generates interrupts to the processor platform if a user-specified value of any signal is crossed.

Content The blockset contains the following elements:

- Processor Interface: THRESHOLD_INTERRUPT_out
(Processor Interface)
- FPGA Interface: THRESHOLD_INTERRUPT_in
(FPGA Interface)
- FPGA: THRESHOLD_INTERRUPT
(FPGA Main Component)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.



Figure 87: THRESHOLD_INTERRUPT_out block

Block Dialog

The processor output block provides the following dialog:

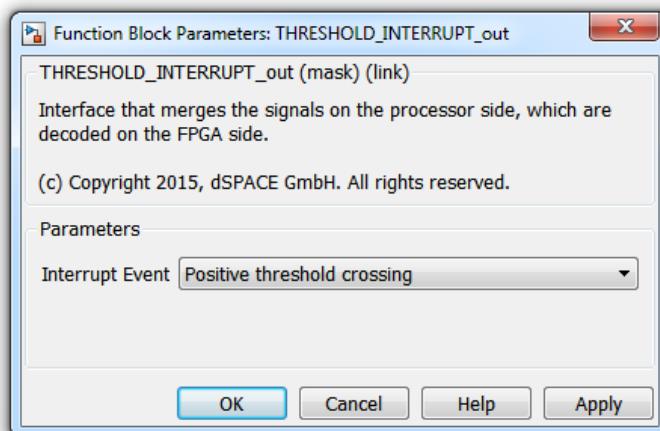


Figure 88: THRESHOLD_INTERRUPT_out dialog

The block dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Interrupt event	-	The event to which the interrupt has to be triggered.	Pos. thresh. cross. Neg. thresh. cross. Any thresh. cross.

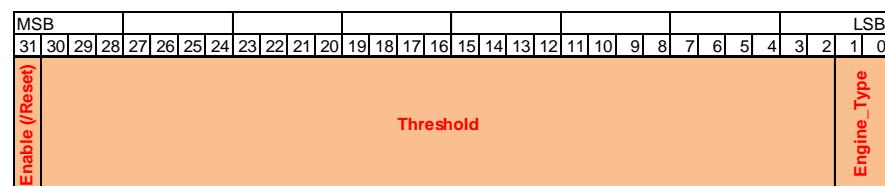
Input

The THRESHOLD_INTERRUPT_out block has the following inputs:

Name	Unit	Description	Range/Resolution
Threshold	-	The threshold at which the interrupt shall be generated.	Range: ±65535 Resolution: 0.0002
Enable Interrupt	-	The generation of interrupts can be activated or deactivated during runtime using this input	0 1

Output

The Processor Out block outputs 1 register. The mapping is shown below.

Register 1

FPGA Main Component

Block

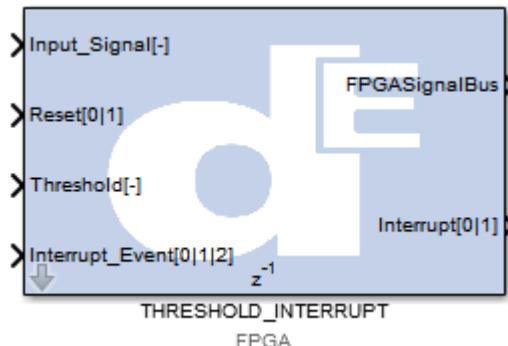


Figure 89: THRESHOLD_INTERRUPT FPGA Main block

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
Input_Signal	deg	The signal to which the interrupt shall be generated.	Fix_20_8
Reset	-	Resets the interrupt generation.	Bool
Threshold	deg	The angle at which an interrupt shall be generated.	Fix_29_12
Interrupt_Event	-	Select positive edge (0), negative edge (1) or any edge (2).	UFix_2_0

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
Interrupt	-	The generated interrupt impulse.	Bool

FPGA Resources

The FPGA (XC7K325T) resources occupied by the component are negligible.

RAPID_PRO_INPUT

Objective	The RapidPro input component transfers sensor data from a RapidPro Control Unit to the FPGA and provides them as normed bus output.
------------------	---

Content	The blockset contains the following elements:
----------------	---

- Processor Interface: RAPID_PRO_INPUT_out
(Processor Interface)
- FPGA Interface: RAPID_PRO_INPUT_in
(FPGA Interface)
- FPGA: RAPID_PRO_INPUT
(FPGA Main Component)

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

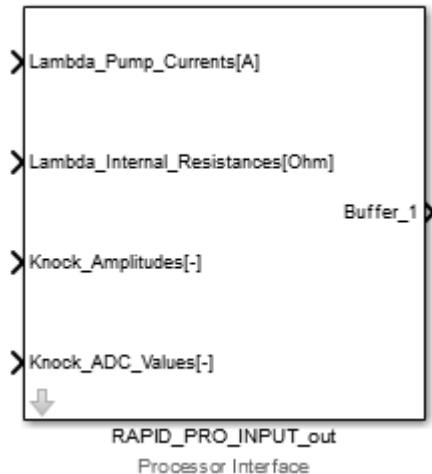


Figure 90: RAPID_PRO_INPUT_out block

Block Dialog

The processor output block provides the following dialog

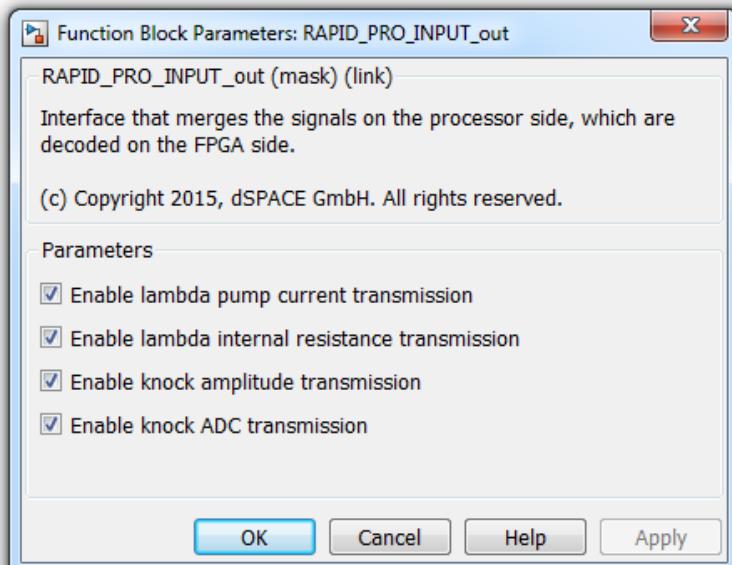


Figure 91: RAPID_RPO_INPUT_out dialog

The block dialog has the following parameters:

Name	Unit	Description	Range/Resolution
Enable lambda pump current transmission	-	Enables the transmission of lambda pump current measurements. Those are typically measured before and after the catalyzer. Up to 2 catalyzers are possible, which makes up a total of up to 4 measurements.	on / off
Enable lambda internal resistance transmission	-	Enables the transmission of lambda internal resistance measurements. Those are typically measured before and after the catalyzer. Up to 2 catalyzers are possible, which makes up a total of up to 4 measurements.	on / off
Enable knock amplitude transmission	-	Enables the transmission of knock detection amplitude. Typically there is 1 knock amplitude measurement per cylinder, which makes up a total of up to 6 amplitude measurements for a 6-cylinder engine.	on / off
Enable knock ADC transmission	-	Enables the transmission of knock ADCs for acceleration sensors with analog output to detect the exact position of knocking.	on / off

Input

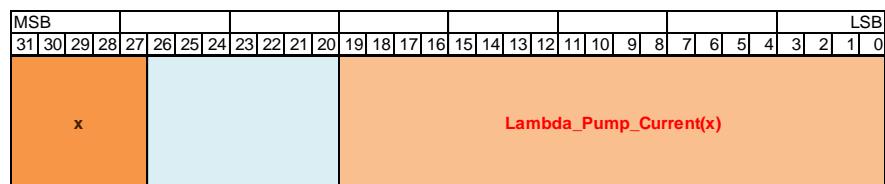
The RAPID_PRO_INPUT_out block has the following inputs:

Name	Unit	Description	Range
Lambda_Pump_Currents	A	Vector with measured pump currents (1...4 elements).	Range: -4A...4A Resolution: 7µA
Lambda_Internal_Resistances	Ω	Vector with measured internal resistances (1...4 elements)	Range: 0...32kΩ Resolution: 1Ω
Knock_Amplitudes	-	Vector with measured knock detection amplitudes (1...6 elements)	Range: 0... 2^{18} -1 Resolution: 1
Knock_ADC_Values	-	Vector with measured knock ADCs for acceleration sensors (1...4 elements)	Range: $\pm 2^{15}$ Resolution: 1

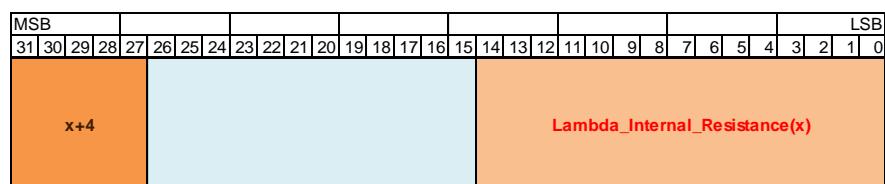
Output

The Processor Out block outputs up to 18 register contents, mapped to 1 buffer. The actual amount of data is dependent on the settings and the length of the input vectors. All data is transferred each simulation step.

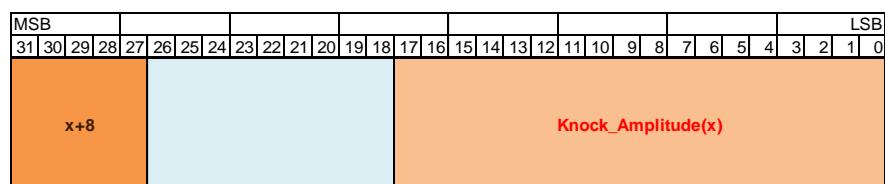
Register 1...4

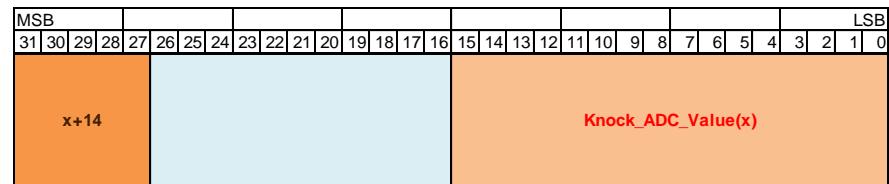


Register 5...8



Register 9...14



Register 15...18

FPGA Main Component

Block

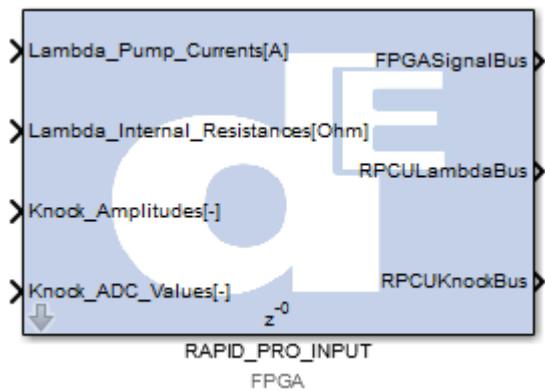


Figure 92: RAPID_PRO_INPUT FPGA Main block

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.

Input

The main block has the following inputs:

Name	Unit	Description	Format
Lambda_Pump_Currents	A	Vector with measured pump currents (4 elements).	Fix_20_17 (4x)
Lambda_Internal_Resistances	Ω	Vector with measured internal resistances (4 elements).	UFix_15_0 (4x)
Knock_Amplitudes	-	Vector with measured knock detection amplitudes (6 elements).	UFix_18_0 (6x)
Knock_ADC_Values	-	Vector with measured knock ADCs for acceleration sensors (4 elements).	Fix_16_0 (4x)

Output

The main block has the following outputs:

Name	Unit	Description	Format
FPGASignalBus	-	Contains all output signals generated by the FPGA main component.	Bus
RPCULambdaBus	-	Contains all lambda measurements of the RapidPro Control Unit (see RPCULambdaBus).	Bus
RPCULambdaBus	-	Contains all knock measurements of the RapidPro Control Unit (see RPCUKnockBus).	Bus

FPGA Resources

The FPGA (XC7K325T) resources are occupied by the component are negligible.

Bus Definitions

Objective

To ease the handling of data from different components, standardized busses are introduced. All busses with the specified name contain the signals and scaling defined in this chapter.

Injection & Ignition

IICAngleBus

Description / Overview	The IICAngleBus (Injection & Ignition Control Angle Bus) contains the calculated angles and pulse times of the ANGLE_SERIALIZER , which are provided to the INJECTION_IGNITION_CORE .
-------------------------------	---

Signals	The IICAngleBus contains the following signals:
----------------	---

Name	Unit	Description	Format
Next_Ignition_Start_Angle	°CA	The next ignition start angle, selected among all input angles, according to the current cylinder angle.	Fix_20_8
Next_Ignition_End_Angle	°CA	The next ignition end angle, selected among all input angles, according to the current cylinder angle.	Fix_20_8
Next_InjectionA_Start_Angle	°CA	The next injection A start angle, selected among all input angles, according to the current cylinder angle.	Fix_20_8
Next_InjectionA_End_Angle	°CA	The next injection A end angle, selected among all input angles, according to the current cylinder angle.	Fix_20_8
Next_InjectionB_Start_Angle	°CA	The next injection B start angle, selected among all input angles, according to the current cylinder angle.	Fix_20_8
Next_InjectionB_End_Angle	°CA	The next injection B end angle, selected among all input angles, according to the current cylinder angle.	Fix_20_8
Next_Ignition_Pulse_Length	Clocks	The pulse length of the next ignition pulse.	UFix_32_0
Next_InjectionA_Pulse_Length	Clocks	The pulse length of the next injection A pulse.	UFix_32_0
Next_InjectionB_Pulse_Length	Clocks	The pulse length of the next injection B pulse.	UFix_32_0
Next_PeakA_Pulse_Length	Clocks	The pulse length of the next PeakA pulse in double control signal injection mode.	UFix_32_0
Next_FullPulse_Length	Clocks	The pulse length of the next double control signal (time from start of pulse A to end of pulse B) in double control signal injection mode.	UFix_32_0

IICFaultBus

Description / Overview The IICFaultBus contains validity checks of the pulse start and end angles provided to or calculated by the [ANGLE_SERIALIZER](#).

Signals The IICFaultBus contains the following signals:

Name	Sub	Unit	Description	Format
All_Ign_Angles_Valid	-	-	This signal indicates that all ignition input angles of the serializer are valid. It is recommended to enable ignition only if this signal is 1.	Bool
All_Ign_Angles_Valid	-	-	This signal indicates that all injection A input angles are valid. It is recommended to enable ignition only if this signal is 1.	Bool
All_Ign_Angles_Valid	-	-	This signal indicates that all injection B input angles are valid. It is recommended to enable ignition only if this signal is 1.	Bool
Ign_Start_Angles_Invalid	1 . . 16	-	Flags indicating if the corresponding ignition start angle is invalid (according to the angle input rules).	Bool
Ign_End_Angles_Invalid	1 . . 16	-	Flags indicating if the corresponding ignition end angle is invalid (according to the angle input rules).	Bool
InjA_Start_Angles_Invalid	1 . . 16	-	Flags indicating if the corresponding injection A start angle is invalid (according to the angle input rules).	Bool
InjA_End_Angles_Invalid	1 . . 16	-	Flags indicating if the corresponding injection A end angle is invalid (according to the angle input rules).	Bool
InjB_Start_Angles_Invalid	1 . . 16	-	Flags indicating if the corresponding injection B start angle is invalid (according to the angle input rules).	Bool
InjB_End_Angles_Invalid	1 . . 16	-	Flags indicating if the corresponding injection Bend angle is invalid (according to the angle input rules).	Bool

IICDiagnosisBus

Description / Overview The IICDiagnosisBus contains error signals as well as internal flags of the [INJECTION IGNITION CORE](#).

Signals The IICAngleBus contains the following signals:

Name	Unit	Description	Format
Max_Ign_Time_Exceeded	-	Error flag indicating that the last ignition pulse had to be stopped after exceeding the maximum duration. Will be reset after an ignition pulse ended without exceeding the maximum duration.	Bool
Max_Inj(A)_Time_Exceeded	-	Error flag indicating that the last injection A pulse had to be stopped after exceeding the maximum duration (or total injection pulse in case of DS1664 double control signal mode). Will be reset after an ignition pulse ended without exceeding the maximum duration.	Bool
Max_InjB_Time_Exceeded	-	Error flag indicating that the last injection B pulse had to be stopped after exceeding the maximum duration. Will be reset after an ignition pulse ended without exceeding the maximum duration. In case of DS1664 double control signal mode this port is unused.	Bool
Ignition_Start_Impulse	-	Impulse (1 clock cycle) indicating that the start of an ignition pulse has been detected.	Bool
Ignition_End_Impulse	-	Impulse (1 clock cycle) indicating that the end of an ignition pulse has been detected.	Bool
InjectionA_Start_Impulse	-	Impulse (1 clock cycle) indicating that the start of an injection pulse A has been detected.	Bool
InjectionA_End_Impulse	-	Impulse (1 clock cycle) indicating that the end of an injection pulse A has been detected.	Bool
InjectionB_Start_Impulse	-	Impulse (1 clock cycle) indicating that the start of an injection pulse B has been detected.	Bool
InjectionB_End_Impulse	-	Impulse (1 clock cycle) indicating that the end of an injection pulse B has been detected.	Bool

Angle Measurement

ACUCommonBus

Description / Overview The ACUCommonBus contains the signals which all angle processing components (crank/cam-based ACU and incremental encoder) provide.

Signals The ACUCommonBus contains the following signals:

Name	Unit	Description	Format
Engine_Angle	°CA	The calculated (wrapped) engine angle. The possible range is -1080°...1080°.	Fix_20_8
Sync_Ok	-	Flag indicating that the index was passed and the angle output is therefore valid.	Bool
Speed	rpm	The engine speed scaled in rpm.	Fix_20_4
Speed_Available	-	Signals with a single impulse of 1 clock cycle that the output speed has been updated with a new value.	Bool
Direction	-	The direction of the engine rotation (0 = forward, 1 = reverse).	Bool
Dec_CA_Trigger	-	Impulse which occurs every 0.1° of engine rotation. The pulse sequence will be activated as soon as the synchronization is established (angle is referenced). The minimum distance between two adjacent 0.1° pulses is internally limited to 1.0 s.	Bool
CI_Dec_CA	Clocks	Number of FPGA clock periods since last Dec_CA_Trigger impulse.	UFix_24_0
dCI_Dec_CA	Clocks	Number of FPGA clock periods between last two Dec_CA_Trigger pulses	UFix_24_0

ACUExtendedBus

Description / Overview The ACUExtendedBus contains the signals which are only available for the crank/cam-based ACU.

Signals The ACUExtendedBus contains the following signals:

Name	Unit	Description	Format
Speed_Avg	rpm	Recursive averaged engine speed scaled in rpm.	Fix_20_4
Speed_Avg_Available	-	Signals with a single impulse of 1 clock cycle that the average speed has been updated with a new value.	Bool
Raw_Angle_Output	°CA/10	The raw angle output scaled in 0.1° increments.	Fix_15_0
Angle_Available	-	Signals with a single impulse of 1 clock cycle that the output angle has been updated with a new value.	Bool
Angle_Event	-	Output of the programmable angle event unit. If the unit is enabled this port is pulsed periodically. The following parameters can be programmed via the ACU register set: 1. Start angle (with a resolution of 0.1° CA). 2. Period (distance between consecutive pulses, in steps of 0.1° CA). 3. Pulse length (in clock cycles).	Bool
Cam_Phi_Available(4x)		Signals with a single impulse of 1 clock cycle that the corresponding camshaft (1 to 4) phase shift value has been updated.	Vector of 4 Bools
Cam_Phi(4x)	-	Outputs the calculated phase shift value of each camshaft. Is provided as the raw FPGA value and has to be multiplied by factor 0.1°CA to convert to physical units.	Bus of 4 Fix_12_0
Cam_Phi_Limits		Vector with bits signaling limit violations of all cam phase shift measurements. Each camshaft phase shift measurement has an <i>AboveMax</i> and <i>BelowMin</i> bit. The values are: Bit 7: Cam4_Phi_AboveMax Bit 6: Cam4_Phi_BelowMin Bit 5: Cam3_Phi_AboveMax Bit 4: Cam3_Phi_BelowMin Bit 3: Cam2_Phi_AboveMax Bit 2: Cam2_Phi_BelowMin Bit 1: Cam1_Phi_AboveMax Bit 0: Cam1_Phi_BelowMin	UFix_7_0

ACUDiagnosisBus

Description / Overview	The ACUDiagnosisBus contains additional debugging signals for the crank/cam-based ACU.
Signals	The ACUDiagnosisBus contains the following signals:

Name	Unit	Description	Format
ACU_Init_Done	-	Flag indicating that the initialization process of the ACU is complete. The ACU is ready for operation.	Bool
ACU_Cam_Init_Error	-	Flag indicating that an error was detected during the cam data verification which takes place during the initialization process. The ACU is not able to operate if an error was detected.	Bool
ACU_Error_Event	-	Pulse with a duration of 2µs in case of the occurrence of an error during runtime.	Bool
ACU_Error_Monitor	-	Information about the error which was signaled via the output ACU_Error_Event. Bit 31..16: Crank angle position of the error. Bit 15..0: Error flags (same as Error-Flag Register bit 15..0 of the ACU register set). ACU_Error_Monitor keeps its value until a new error will be detected.	UFix_32_0
SW_Test_Monitor	-	This port should pulse at least once during the initialization process for 2µs. Otherwise the initialization function might be erroneous.	Bool
Crank_Monitor	-	Digital spike-filtered crank input. In case of ACU malfunction, this signal should be mapped to a digital output and observed.	Bool
Crank_Active_Edge_Monitor	-	A pulse with the duration of 2µs, which is generated each time an active crank edge has been detected by the ACU. Please note that either the rising or the falling crank edge may be set as active crank.	Bool
Crank_Event0	-	Changes to '1' after recognition of that crank event which was defined as "Event 0". Stays '1' in case of consecutive detected crank events of this event type.	Bool
Crank_Event1	-	Changes to '1' after recognition of that crank event which was defined as "Event 1". Stays '1' in case of consecutive detected crank events of this event type.	Bool

Crank_Event2	-	Changes to '1' after recognition of that crank event which was defined as "Event 2". Stays '1' in case of consecutive detected crank events of this event type.	Bool
Cam1_Monitor	-	Digital spike-filtered cam 1 input. In case of ACU malfunction, this signal should be mapped to a digital output and observed.	Bool
Cam2_Monitor	-	Digital spike-filtered cam 2 input. In case of ACU malfunction, this signal should be mapped to a digital output and observed.	Bool
Cam3_Monitor	-	Digital spike-filtered cam 3 input. In case of ACU malfunction, this signal should be mapped to a digital output and observed.	Bool
Cam4_Monitor	-	Digital spike-filtered cam 4 input. In case of ACU malfunction, this signal should be mapped to a digital output and observed.	Bool

Pressure Indication

CPIChannel[x]Bus

Description / Overview The CPIChannel[x]Bus contains all results generated by cylinder pressure indication for a single channels.

Signals The CPIChannel[x]Bus contains the following signals:

Sub-Bus	Name	Unit	Description	Format
-	Cylinder_Angle	°CA	The current cylinder angle wrapped according to the user settings.	Fix_20_8
pMaxBus	p_max	Pa	The measured maximum in-cylinder pressure of the current engine cycle.	Fix_26_0
	p_max_Angle	°CA	The angle corresponding to the maximum in-cylinder pressure of the current engine cycle.	Fix_20_8
	dp_max	Pa 0.1°	The maximum pressure slew rate of the current engine cycle.	Fix_32_0
	dp_max_Angle	°CA	The angle corresponding to the measured maximum pressure slew rate of the current engine cycle.	Fix_20_8
VolumeBus	Cylinder_Volume	m³	Current cylinder volume.	UFix_31_31
IMEPBus	IMEP	Pa	Current integration value of indicated mean effective pressure.	Fix_24_0
	IMEP_Hold	Pa	Final value of the indicated mean effective pressure.	Fix_24_0

	IMEP_CC	Pa	Current integration value of indicated mean effective pressure of the charge cycle.	Fix_24_0
	IMEP_CC_Hold	Pa	Final value of integration value of indicated mean effective pressure of the charge cycle.	Fix_24_0
	IMEP_WC	Pa	Current integration value of indicated mean effective pressure of the working cycle.	Fix_25_0
	IMEP_WC_Hold	Pa	Final value of indicated mean effective pressure of the working cycle.	Fix_25_0
QBBus	QB	J	The integrated released heat caused by combustion.	Fix_16_0
	QB_max	J	The maximum released heat during the current engine cycle.	Fix_16_0
	dQB	$\frac{J}{0.1^\circ}$	The incremental released heat caused by combustion.	Fix_32_22
CombustionBus	Combustion_Active	-	Indicates if combustion is currently active.	Bool
MFBBus	MFB10	°CA	Angle where 10% of the mass fraction is burned.	Fix_20_8
	MFB50	°CA	Angle where 50% of the mass fraction is burned.	Fix_20_8

	MFB90	°CA	Angle where 90% of the mass fraction is burned.	Fix_20_8
ExtendedBus	InCylinder_Pressure_Comp	Pa	Filtered and drift-compensated in-cylinder pressures	Fix_26_0
	dp	Pa 0.1°	The current pressure slew rate.	Fix_26_0
	p_max_Hold	Pa	The maximum pressure, latched by the end of the combustion cycle.	Fix_26_0
	p_max_Angle_Hold	°CA	The angle of the maximum pressure, latched by the end of the combustion cycle.	Fix_20_8
	dp_max_Hold	Pa	The maximum pressure slew rate, latched by the end of the combustion cycle.	Fix_26_0
	dp_max_Angle_Hold	Pa 0.1°	The angle of the maximum pressure slew rate, latched by the end of the combustion cycle.	Fix_20_8
	Delta_p_Unfiltered	Pa 0.1°	The pressure delta for combustion detection.	Fix_26_0
	Delta_p_Start	Pa 0.1°	The pressure delta for combustion detection, filtered for combustion start detection.	Fix_26_0

	Delta_p_End	$\frac{\text{Pa}}{0.1^\circ}$	The pressure delta for combustion detection, filtered for combustion end detection.	Fix_26_0
	Combustion_Offset	0.1°	The signals on the Combustion and QB bus are delayed due to filtering. This output shows the angular delay of these signals.	UFix_8_0
	Combustion_Active_Est	-	Shows that the cylinder angle is in the possible combustion window.	Bool
	QB_Cal	J	QB calculated over the range of the whole combustion estimation window (defined by min. start and max. end angle). It is meant for calibration purpose.	Fix_16_0
	QB_max_Cal	J	QB_{\max} calculated over the range of the whole combustion estimation window.	Fix_16_0
	dQB	$\frac{J}{0.1^\circ}$	dQB calculated over the range of the whole combustion estimation window.	Fix_32_22

	MFB10_Cal	°CA	MFB10 calculated over the range of the whole combustion estimation window.	Fix_20_8
	MFB50_Cal	°CA	MFB50 calculated over the range of the whole combustion estimation window.	Fix_20_8
	MFB90_Cal	°CA	MFB90 calculated over the range of the whole combustion estimation window.	Fix_20_8
	New_Cycle	-	Impulse of one clock cycle, indicating that a new combustion cycle has started.	Bool
	Cycle_Count	-	Number of the current combustion cycle.	UFix_32_0

	The signals on CombustionBus and QBBus are delayed due to filtering. The output signal Combustion_Offset shows the angular delay. All other signals (including ExtendedBus) are not delayed.
---	---

RapidPro Interface

RPLambdaBus

Description / Overview	The RPLambdaBus contains all lambda measurement signals from the RapidPro Control Unit.
------------------------	---

Signals	The RPLambdaBus contains the following signals:
---------	---

Name	Sub	Unit	Description	Format
Pump_Currents	1	A	Measured pump current.	Fix_20_17
	2			
	3			
	4			
Internal_Resistances	1	Ω	Measured internal resistance.	UFix_15_0
	2			
	3			
	4			
	2			
	3			
	4			

RPKnockBus

Description / Overview	The RPKnockBus contains all knock measurement signals from the RapidPro Control Unit.
------------------------	---

Signals	The RPKnockBus contains the following signals:
---------	--

Name	Sub	Unit	Description	Format
Sensor_Type		-	0 = RapidPro 1 = FPGA	Bool
Amplitudes	1	-	Measured knock detection amplitude.	UFix_18_0
	2			
	3			
	4			
	5			
	6			
ADC_Values	1	-	Measured knock ADCs for acceleration sensor.	Fix_16_0
	2			
	3			
	4			

Demos

Overview

The XSG Advanced Engine Control library provides one template demo model.

This model is available for the platforms DS1005, DS1006, DS1007 configured for the DS5203 with onboard I/O and the DS1401 configured with the DS1552B1 I/O Module. For other hardware configurations the interfaces have to be adapted.



The XSG Advanced Engine Control Demos requires the XSG Advanced Engine Control FPGA library.

Starting Demos

To start a demo model, click on the “Demos” link in the XSG Advanced Engine Control library as shown below:

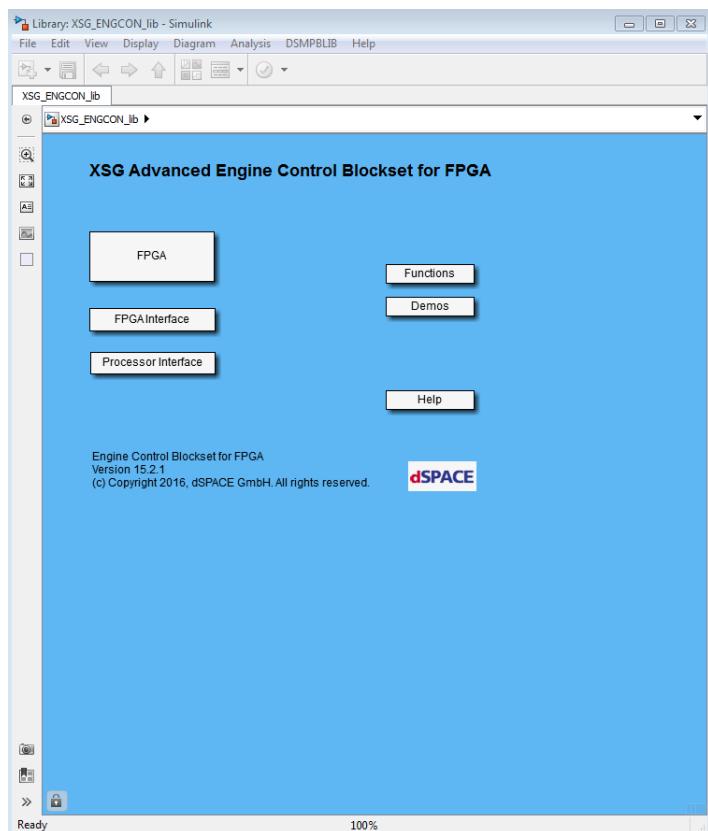


Figure 93: Demos in the XSG Advanced Engine Control library

When opening the “Demos” link the first time, the following dialog will appear:



Figure 94: Dialog when opening a demo for the first time

Choose “Install & Open” to copy the demo models to your personal folders. After the demos have been copied a second dialog will appear:

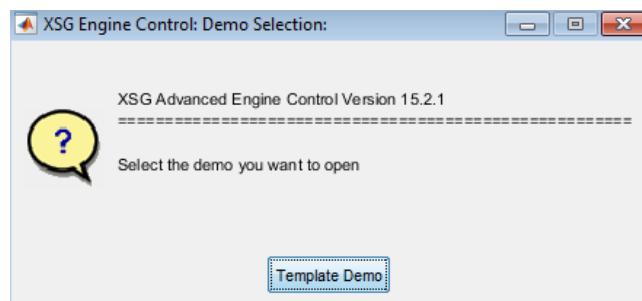


Figure 95: Dialog for selecting a demo model

For the time being only the template demo is available. After it has been chosen MATLAB will jump to the corresponding folder and open the Simulink model.

Template Demo

Overview

The template demo should give an idea how the different parts of the Advanced Engine Control Solution interact with each other. It is not supposed to control a real motor as no specific parametrization and control algorithms are included in the model. This demo can be connected to either a crankshaft signal with up to 4 camshafts or a TTL signals of an engine. With that information it outputs definable ignition and injection signals for cylinder 1. The model also includes the cylinder pressure indication for up to 4 cylinders.

Structure

The top-level contains the following elements:

- The processor setup block
- The FPGA subsystem
- An Interrupt subsystem containing four interrupt routines triggered by the FPGA application
- An input subsystem
- An output subsystem

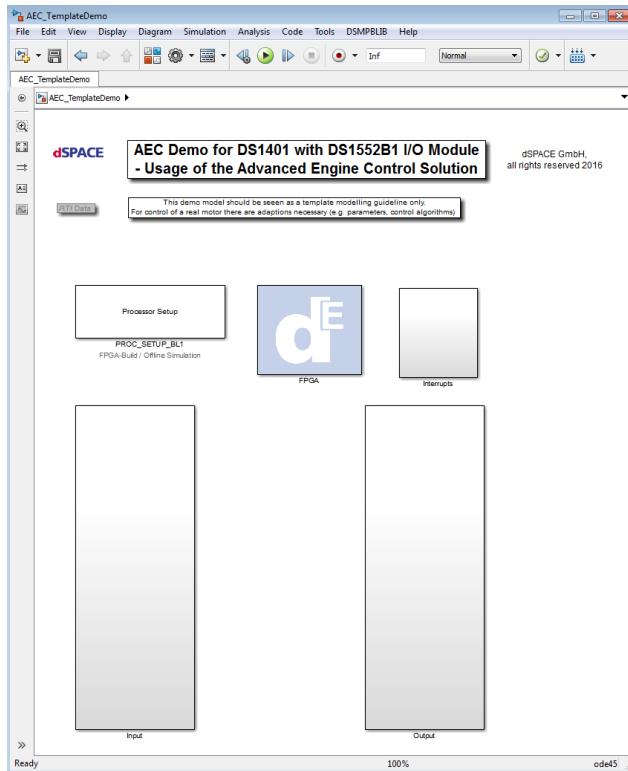


Figure 96: Top-level of the template demo model

FPGA subsystem

The FPGA subsystem contains the following elements:

- A subsystem for the encoder
- A subsystem for the ACU
- A subsystem for the encoder/ACU switch
- A subsystem for the angle and threshold interrupts
- A subsystem for the pressure ADCs
- A subsystem for one knock sensor ADC (*DS1554 only*)
- A subsystem for the Cylinder Pressure Indication (CPI)
- A subsystem for the Transformation blocks
- A subsystem for Injection/Ignition for one cylinder
- A subsystem for angle based pulse generation for one cylinder

Input

The input subsystem is used for configuration and read values of the several FPGA subsystems. It includes:

- An interrupt configuration subsystem to configure the angle and threshold interrupts.

- An ENGINE_2_CYLINDER subsystem to define the TDCs of the cylinders.
- A P_SENSOR_SCALING subsystem to scale the ADCs used for pressure indication and knock detection.
- A KNOCK_DETECTION subsystem to configure the measurement windows and the FIR filters and read the filter outputs (*DS1554 only*).
- An ENGINE_ENCODER subsystem to configure the engine encoder and read its outputs.
- An ACU subsystem to define Crankshafts and Camshaft signals for the ACU and read its outputs.
- A Source Select subsystem which configures the engine source and routes the corresponding signals of the ENGINE_ENCODER or ACU subsystems to its outputs.
- A CYLINDER_2_ENGINE subsystem.
- A CPI subsystem to configure the Cylinder Pressure Indication and read its outputs for up to four cylinders.

Output

The output subsystem is used for defining and enabling the ignition and injection signals as well as a pulse. Those signals are all based on the angle of cylinder 1.

I/O Mapping

DS1401 with DS1552B1

The following table shows the I/O assignment of the Template Demo model and the ZIF connector of the DS1552B1 I/O module.

Function	Name	ZIF Pin
Cylinder 1 pressure (+)	AnalogIn+ ch1	X3
Cylinder 1 pressure (-)	AnalogIn- ch1	X4
Cylinder 2 pressure (+)	AnalogIn+ ch2	W3
Cylinder 2 pressure (-)	AnalogIn- ch2	W4
Cylinder 3 pressure (+)	AnalogIn+ ch3	V3
Cylinder 3 pressure (-)	AnalogIn- ch3	V4
Cylinder 4 pressure (+)	AnalogIn+ ch4	U3
Cylinder 4 pressure (-)	AnalogIn- ch4	U4
Digital crank (+)	CrankCam+ ch1	R3
Digital crank (-)	CrankCam- ch1	R4
Digital cam 1 (+)	CrankCam+ ch2	B3
Digital cam 1 (-)	CrankCam- ch2	B4
Digital cam 2 (+)	CrankCam+ ch3	A3
Digital cam 2 (-)	CrankCam- ch3	A4
Digital cam 3 (+) *	DigIn ch8	T6
Digital cam 3 (-) *	GND	A1
Digital cam 4 (+) *	DigIn ch9	S2
Digital cam 4 (-) *	GND	A2
Ignition Pulse	DigIO ch1	N2
Injection A Pulse	DigIO ch2	N3
Injection B Pulse	DigIO ch3	N4
Angle Based Pulse	DigOut ch4	D2

**DS1401 with DS1554
Engine Control Module**

The following table shows the I/O assignment of the Template Demo model and the ZIF connector and SUB-D connector of the DS1554 Engine Control module.

Function	Name	ZIF Pin	SUB-D Pin
Cylinder 1 pressure (+)	AnalogIn+ ch1	W2	-
Cylinder 1 pressure (-)	AnalogIn- ch1	V2	-
Cylinder 2 pressure (+)	AnalogIn+ ch2	Y2	-
Cylinder 2 pressure (-)	AnalogIn- ch2	X2	-
Cylinder 3 pressure (+)	AnalogIn+ ch3	S2	-
Cylinder 3 pressure (-)	AnalogIn- ch3	R2	-
Cylinder 4 pressure (+)	AnalogIn+ ch4	T2	-
Cylinder 4 pressure (-)	AnalogIn- ch4	U2	-
Knock sensor 1 (+)	KnockIn+ ch1		16
Knock sensor 1 (-)	KnockIn- ch1		34
Digital crank	CrankCam ch1	-	13
Digital cam 1	CrankCam ch2	-	32
Digital cam 2	CrankCam ch3	-	14
Digital cam 3	CrankCam ch4	-	33
Digital cam 4	CrankCam ch5	-	12
CrankCam GND	CrankCam GND		11
Ignition Pulse	DigOut ch1	L5	-
Injection A Pulse	DigOut ch2	N2	
Injection B Pulse	DigOut ch3	D3	
Angle Based Pulse	DigOut ch4	N5	

DS1005/1006/1007 with DS5203

The following table shows the I/O assignment of the Template Demo model and the Sub-D of the DS5203 FPGA board.

Function	Name	Sub-D Pin
Cylinder 1 pressure (+)	ADC1	34
Cylinder 1 pressure (-)	/ADC1	1
Cylinder 2 pressure (+)	ADC2	2
Cylinder 2 pressure (-)	/ADC2	18
Cylinder 3 pressure (+)	ADC3	19
Cylinder 3 pressure (-)	/ADC3	35
Cylinder 4 pressure (+)	ADC4	36
Cylinder 4 pressure (-)	/ADC4	3
Digital crank (+)	DIG_IO1	22
Digital crank (-)	GND	39
Digital cam 1 (+)	DIG_IO2	6
Digital cam 1 (-)	GND	40
Digital cam 2 (+)	DIG_IO3	23
Digital cam 2 (-)	GND	41
Digital cam 3 (+) *	DIG_IO4	7
Digital cam 3 (-) *	GND	42
Digital cam 4 (+) *	DIG_IO5	24
Digital cam 4 (-) *	GND	43
Ignition Pulse	DIG_IO6	8
Injection A Pulse	DIG_IO7	25
Injection B Pulse	DIG_IO8	9
Angle Based Pulse	DIG_IO9	26

MicroAutoBox III**ConfigurationDesk**

For the MicroAutoBox III the demo folder also contains a ConfigurationDesk project. The FPGA build is imported to this project as a Custom Function. In the processor model the interface to the FPGA is implemented with Model Port blocks. For further information about ConfigurationDesk and FPGA implementations for SCALEXIO platforms please refer to the appropriate documentation of the dSPACE Help.

MABXII CPI Solution

Objective

In addition to the XSG Advanced Engine Control solution, a dedicated solution for cylinder pressure indication on the DS1401/DS151x/DS1514/DS1552 MicroAutoBox II is provided. This solution comes with a fixed FPGA application for indication of up to 8 cylinders and an ACU for angle and speed processing, as well as a template model for integrating ACU and CPI in the user Simulink model. There is no RTI FPGA or Xilinx Vivado license required for using the MABXII CPI solution.



For information about the ACU and CPI components please refer to the chapters [ACU](#) and [CPI_4CYL](#).

Scaling

Overview

The scaling component includes block sets to configure the scaling of the pressure sensors and read the scaled pressure values back to the processor.

Processor Output

Block

Merges the processor signals and writes them to the FPGA.

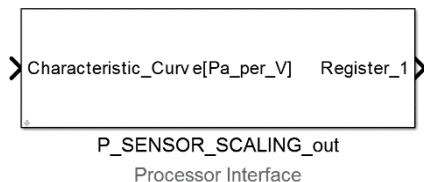


Figure 97: P_SENSOR_SCALING_out block

Block Dialog

The processor output block provides the following dialog:

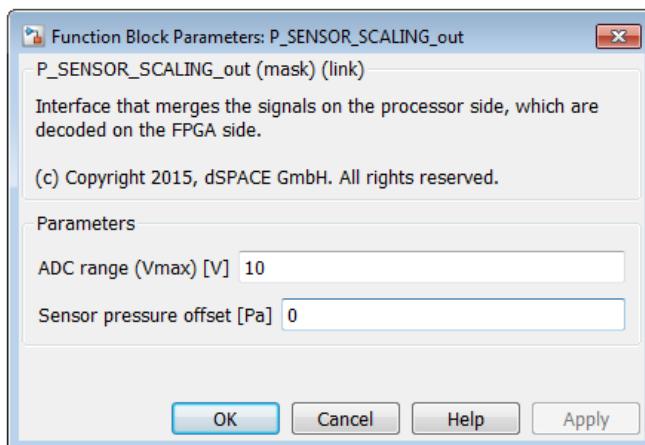


Figure 98: P_SENSOR_SCALING_out dialog

The block dialog has the following parameters:

Name	Unit	Description	Range/Resolution
ADC range (Vmax)	V	Specifies the voltage range of the ADC. The maximum ADC value has to be set here.	Range: 1...30 V Resolution 1 V
Sensor pressure offset	Pa	Specifies the sensor pressure offset.	Range: -33554432...33554431 Pa Resolution: 1 Pa

Input

The P_SENSOR_SCALING_out block has the following inputs:

Name	Unit	Description	Range/Resolution
Characteristic Curve	Pa/V	Define the conversion factor of the ADC in Pa per V.	Range: -16777216... 16777215 Resolution: 1

Processor Input

Block Adapts the FPGA signals for the processor side.

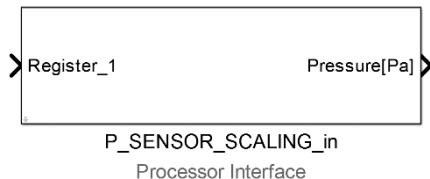


Figure 99: P_SENSOR_SCALING_in block

	<p>This block is only used to provide output signals to the processor application. If the signals are not required in the processor application, computation resources can be saved by not using this block.</p>
---	--

Output The P_SENSOR_SCALING_in block has the following outputs:

Name	Unit	Description	Range
Pressure	Pa	Outputs the converted ADC value in Pa according to the parameters defined in the P_SENSOR_SCALE_out block.	Range: -33554432...33554431 Pa Resolution: 1 Pa

Template Model

Description / Overview

The model contains the instantiations for all blocks required for the angular computation unit and cylinder pressure indication. The template can be opened by clicking the Template button in the MABXII Cylinder Pressure Indication library.

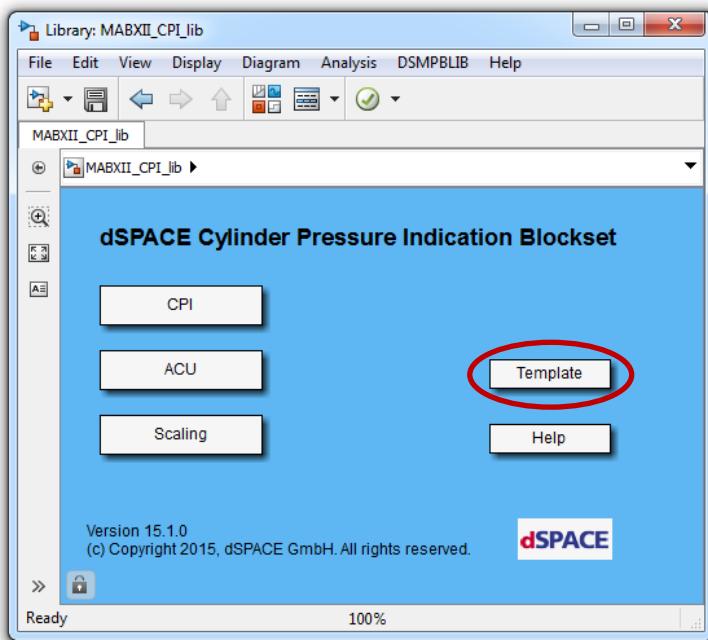


Figure 100: Template location in the MABXII CPI library

Figure 101 shows the structure of the template model. If four or less cylinders are used, only one of the CPI blocks have to be copied to the user model. The Processor Setup block will be required in the user model as well.

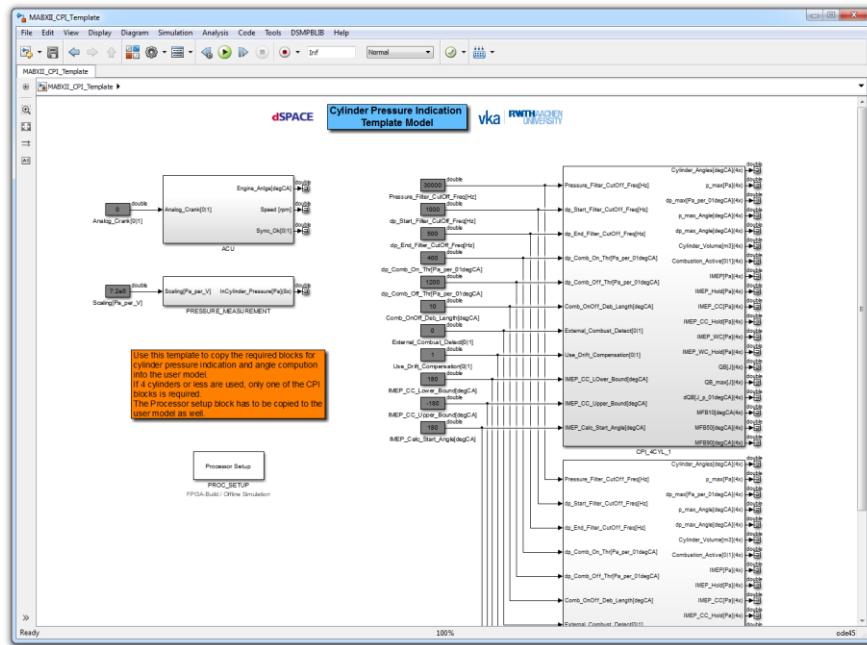


Figure 101: MABXII CPI template model



When copying the CPI subsystems, the reference to the second buffer of the CPI_4CYL_out component will be lost. This is indicated by the second buffer output of the component. Uncheck the option “Initialization Only” of the second buffer and connect it to the Buffer_2 output of CPI_4CYL_out to restore the reference.



Interrupts for the end of each cylinder cycle are available. Just drop a PROC_INT block from the RTI FPGA Programming Blockset Processor Interface into your model. The interrupt numbers 1-4 correspond to the four cylinders of CPI_4CYL_1, the interrupt numbers 5-8 correspond to the four cylinders of CPI_4CYL_2.

The processor setup block is needed to choose the FPGA bit stream for the right platform. A bit stream is available for the DS1552B1 (*mabx2_cpi_7k325_1552.ini*) and the DS1554 (*mabx2_cpi_7k325_1554.ini*) piggybacks. The figure below shows how the bit stream can be changed in the processor setup dialog:

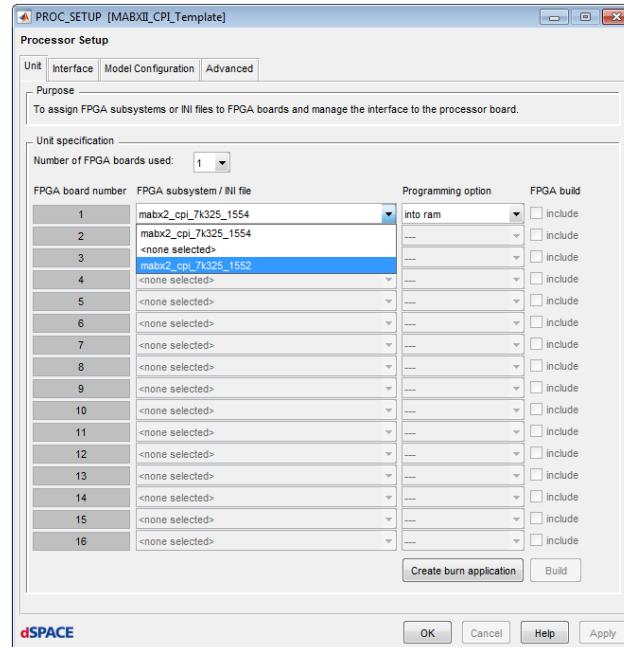


Figure 102: MABXII CPI change INI file

I/O Mapping

Description / Overview

The following table shows the I/O assignment of the MABXII CPI Solution and the ZIF connector of the DS1552B1 I/O module:

Function	Name	ZIF Pin
Cylinder 1 pressure (+)	AnalogIn+ ch1	X3
Cylinder 1 pressure (-)	AnalogIn- ch1	X4
Cylinder 2 pressure (+)	AnalogIn+ ch2	W3
Cylinder 2 pressure (-)	AnalogIn- ch2	W4
Cylinder 3 pressure (+)	AnalogIn+ ch3	V3
Cylinder 3 pressure (-)	AnalogIn- ch3	V4
Cylinder 4 pressure (+)	AnalogIn+ ch4	U3
Cylinder 4 pressure (-)	AnalogIn- ch4	U4
Cylinder 5 pressure (+)	AnalogIn+ ch5	H3
Cylinder 5 pressure (-)	AnalogIn- ch5	H4
Cylinder 6 pressure (+)	AnalogIn+ ch6	G3
Cylinder 6 pressure (-)	AnalogIn- ch6	G4
Cylinder 7 pressure (+)	AnalogIn+ ch7	F3
Cylinder 7 pressure (-)	AnalogIn- ch7	F4
Cylinder 8 pressure (+)	AnalogIn+ ch8	E3
Cylinder 8 pressure (-)	AnalogIn- ch8	E4
Inductive (analog) crank (+)	ZeroVoltageDetector+	P3
Inductive (analog) crank (-)	ZeroVoltageDetector-	P4
Digital crank (+)	CrankCam+ ch1	R3
Digital crank (-)	CrankCam- ch1	R4
Digital cam 1 (+)	CrankCam+ ch2	B3
Digital cam 1 (-)	CrankCam- ch2	B4
Digital cam 2 (+)	CrankCam+ ch3	A3
Digital cam 2 (-)	CrankCam- ch3	A4
Digital cam 3 (+)	DigIO ch1	N2
Digital cam 3 (-)	GND	A1
Digital cam 4 (+)	DigIO ch2	N3
Digital cam 4 (-)	GND	A2
0.1° output pulse (5V, 200ns)	DigIO ch3	N4

The following table shows the I/O assignment of the MABXII CPI Solution and the ZIF connector and SUB-D connector of the DS1554 Engine Control module:

Function	Name	ZIF Pin	SUB-D Pin
Cylinder 1 pressure (+)	AnalogIn+ ch1	W2	-
Cylinder 1 pressure (-)	AnalogIn- ch1	V2	-
Cylinder 2 pressure (+)	AnalogIn+ ch2	Y2	-
Cylinder 2 pressure (-)	AnalogIn- ch2	X2	-
Cylinder 3 pressure (+)	AnalogIn+ ch3	S2	-
Cylinder 3 pressure (-)	AnalogIn- ch3	R2	-
Cylinder 4 pressure (+)	AnalogIn+ ch4	T2	-
Cylinder 4 pressure (-)	AnalogIn- ch4	U2	-
Cylinder 5 pressure (+)	AnalogIn+ ch5	V5	-
Cylinder 5 pressure (-)	AnalogIn- ch5	W6	-
Cylinder 6 pressure (+)	AnalogIn+ ch6	W3	-
Cylinder 6 pressure (-)	AnalogIn- ch6	V3	-
Cylinder 7 pressure (+)	AnalogIn+ ch7	T3	-
Cylinder 7 pressure (-)	AnalogIn- ch7	U3	-
Cylinder 8 pressure (+)	AnalogIn+ ch8	U5	-
Cylinder 8 pressure (-)	AnalogIn- ch8	V6	-
Inductive (analog) crank (+)	ZeroVoltageDetector+	-	10
Inductive (analog) crank (-)	ZeroVoltageDetector-	-	29
Digital crank	CrankCam ch1	-	13
Digital cam 1	CrankCam ch2	-	32
Digital cam 2	CrankCam ch3	-	14
Digital cam 3	CrankCam ch4	-	33
Digital cam 4	CrankCam ch5	-	12
CrankCam GND	CrankCam GND		11
0.1° output pulse (5V, 200ns)	DigIO ch3	b2	-
GND	GND	b1	