****

**AXI XGS Remapper**

**User Guide**

|  |  |
| --- | --- |
| Authors | Raf Thys |
| Issue 1 | June 28 2019; Start doc |

Contents

[1 Disclaimer 3](#_Toc25058467)

[2 Description 4](#_Toc25058468)

[3 Main Blocks 5](#_Toc25058469)

[3.1 Register interface 5](#_Toc25058470)

[3.2 Scrambler 5](#_Toc25058471)

[3.3 Reorder 5](#_Toc25058472)

[3.4 Control BRAM 6](#_Toc25058473)

[3.5 Gearbox\_out 6](#_Toc25058474)

[4 Parameters 7](#_Toc25058475)

[5 Port Description 8](#_Toc25058476)

[6 Register map 9](#_Toc25058477)

[6.1 Overview 9](#_Toc25058478)

[6.2 Detailed description 10](#_Toc25058479)

[6.2.1 CONFIG: 0x0000 10](#_Toc25058480)

[6.2.2 FIFO\_STATUS: 0x0004 11](#_Toc25058481)

[6.2.3 BRAM\_STATUS: 0x0008 11](#_Toc25058482)

[7 Programming Sequence 12](#_Toc25058483)

# Disclaimer

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (c) 2019, ON Semiconductor Inc.

This intellectual property and/or documentation is provided by ON Semiconductor under limited terms and conditions.

The terms and conditions pertaining to the intellectual property and/or documentation are available at [www.onsemi.com](http://www.onsemi.com) ("ON Semiconductor Standard Terms and Conditions of Sale").

Do not use this intellectual property and/or documentation unless you have carefully read and you agree to the limited terms and conditions.

By using this intellectual property and/or documentation, you agree to the limited terms and conditions.

DEVELOPMENT PRODUCT(S), PROTOTYPE OR OTHER NON-PRODUCTION PRODUCT(S),SAMPLES OF PRODUCTION PRODUCT(S) SOFTWARE AND INTELLECTUAL PROPERTY ARE NOT WARRANTED AND ARE PROVIDED ON AN "AS IS" BASIS ONLY. IN NO EVENT SHALL ON SEMICONDUCTOR BE LIABLE FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING FROM INFRINGEMENT OR ALLEGED INFRINGEMENT OF PATENTS, COPYRIGHTS, OR OTHER INTELLECTUAL PROPERTY RIGHTS.

THE INTELLECTUAL PROPERTY PROVIDED IS PROVIDED "AS IS".

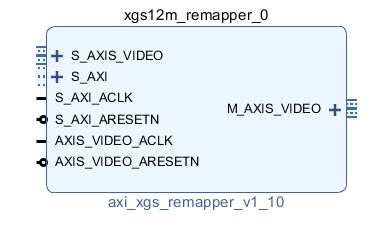
ON SEMICONDUCTOR EXPRESSLY DISCLAIMS ALL WARRANTIES WITH RESPECT TO THE INTELLECTUAL PROPERTY PROVIDED, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, WARRANTIES OF NONINFRINGMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTY OF CONTINUED OR UNINTERRUPTED OPERATION OF THE INTELLECTUAL PROPERTY PROVIDED HEREUNDER.

IN NO EVENT SHALL ON SEMICONDUCTOR BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY NATURE WHATSOEVER (INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS, LOSS OF USE AND LOSS OF GOODWILL), REGARDLESS OF WHETHER ON SEMICONDUCTOR HAS BEEN GIVEN NOTICE OF ANY SUCH ALLEGED DAMAGES, AND REGARDLESS OF WHETHER SUCH ALLEGED DAMAGES ARE SOUGHT UNDER CONTRACT, TORT OR OTHER THEORIES OF LAW.

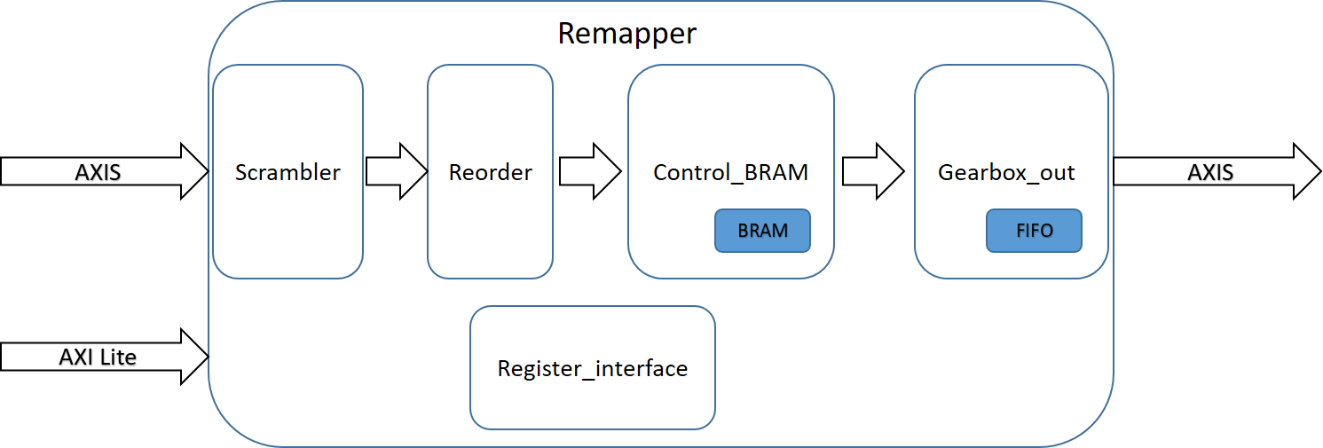
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Description

AXI XGS remapper is an IP block that takes data from the xgs decoder axi streaming bus and reorders the pixel data back into an axi video streaming bus.



# Main Blocks



## Register interface

The register interface connects through an AXI4\_LITE interface to the XGS\_REMAPPER IP. It is used to configure the IP block.

Specific registers and their function are described in the register map section.

## Scrambler

The pixel data stream enters here the remapper. It’s the task of the scrambler to reformat the input data stream to a fixed format with all the data channels in the correct sequence:

[ch#N-1, ch#N-2, ch#N-3, ch#N-4, …, ch#3, ch#2, ch#1, ch#0]

The expected data input format of the current implementation is as follows:

[all odd data channels + all even data channels], for N channels from 0 to N-1 this gives

[ch#N-1, ch#N-3 ,…, ch3, ch1, ch#N-2, ch#N-4,.. , ch#2, ch#0]

This input format can be different dependent on the hardware platform used. In this case, another scrambler function should be added in the code.

## Reorder

This block can swap the data lane sequence based on odd or even sensor line information.

## Control BRAM

In the control BRAM, all the pixel values are stored in RAM on a sensor line basis. There is a write and read mechanism inside to be able to reconstruct the pixel order of the line from the first to the last pixel.

There is also a bypass mode to directly send out the incoming data stream without any reshuffling of the pixel order.

## Gearbox\_out

The last block in the remapper takes care to output the data stream towards the next IP, which is typically a VDMA. The pixel output data is represented with16 bit per pixel, padding the non-significant bits with 0’s at the lsb’s. The output data stream coming from the FIFO is organized in blocks of 512 bits with the lowest pixel number at the lsb.

A Start Of Frame indicator is available in the M\_AXIS\_VIDEO\_TUSER(0) bit.

# Parameters

Following custom parameters control this block:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Default value | Description | Editable |
| C\_FAMILY | kintexu | Family generic, hidden in customization interface. Not editable. | n |
| C\_PLATFORM | 2 |  | n |
| C\_NROF\_DATACONN | 24 | Total number of incoming HiSpi lanes | y |
| C\_INPUT\_DATAWIDTH | 12 | Number of bits per pixel of incoming data | n |
| C\_OUTPUT\_DATAWIDTH | 16 | Number of bits per pixel of outgoing data | n |
| C\_M\_AXIS\_TDATA\_WIDTH | 512 | Master axi streaming bus data width | n |
| C\_M\_AXIS\_TUSER\_WIDTH | 3 | Master axi streaming number of tuser bits (used for SOF) | n |
| C\_S\_AXIS\_TDATA\_WIDTH | 24\*12 | Slave axi streaming bus data width | n |
| C\_S\_AXIS\_TUSER\_WIDTH | 3 | Slave axi streaming number of tuser bits (used for SOF) | n |
| C\_AXIS\_TID\_WIDTH | 1 | Number of axi streaming tid bits (not used for now) | n |
| C\_AXIS\_TDEST\_WIDTH | 1 | Number of axi streaming tdest bits (not used for now) | n |
| C\_S\_AXI\_DATA\_WIDTH | 32 | Axi4 lite data width | n |
| C\_S\_AXI\_ADDR\_WIDTH | 12 | Axi4 lite address width | n |

# Port Description

Signals grouped as per clock domain.

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **Interface** | **Signal Type** | **Description** |
| **Clock Domain:** AXI\_MM\_CLOCK | | | |
| S\_AXI\_ACLK | clock | input | Axi clock |
| S\_AXI\_ARESETN | reset | input | Active low reset |
| S\_AXI | AXI LITE | Slave | Axi MM bus interface with 32-bit data register interface and 16 registers. |
| **Clock Domain:** AXI\_STREAM\_CLK | | | |
| AXIS\_VIDEO\_ACLK | clock | input | Axi streaming clock |
| AXIS\_VIDEO\_ARESETN | reset | input | Active low reset |
| S\_AXIS\_VIDEO | AXI STREAM | Sink | Axi Slave streaming bus interface |
| M\_AXIS\_VIDEO | AXI STREAM | Source | Axi Master streaming bus interface |

# Register map

## Overview

The following registers are accessible through the AXI4 Lite Slave (S\_AXI) interface.

|  |  |  |  |
| --- | --- | --- | --- |
| **Address Offset** | **Register Name** | **R/W** | **Description** |
| 0x0000 | CONFIG | R/W | Configuration bits. See detailed description for more info. |
| 0x0004 | FIFO\_STATUS | R | Output fifo status bits. See detailed description for more info. |
| 0x0008 | BRAM\_STATUS | R |  |
| 0x000C | RFU |  |  |
| 0x0010 | RFU |  |  |
| 0x0014 | RFU |  |  |
| 0x0018 | RFU |  |  |
| 0x001C | RFU |  |  |
| 0x0020 | RFU |  |  |
| 0x0024 | RFU |  |  |
| 0x0028 | RFU |  |  |
| 0x002C | RFU |  |  |
| 0x0030 | RFU |  |  |
| 0x0034 | RFU |  |  |
| 0x0038 | RFU |  |  |
| 0x003C | RFU |  |  |

## Detailed description

### CONFIG: 0x0000

|  |  |  |
| --- | --- | --- |
| **Bit Offset** | **Field Name** | **Description** |
| [31:12] | RESERVED | Not used |
| [11:8] | BRAM\_READOUT\_MODE | Bit 0 = Sensor readout mode  0 : Incremental readout  1 : Nested readout  Bit [3:1]: RFU |
| 7 | RESERVED | Not used |
| 6 | BRAM\_PIXEL\_0\_LANE | 0: Pixel 0 is located at HiSpi Lane 0 (even row)  1: Pixel 0 is located at HiSpi Lane 1 (even row) |
| 5 | BRAM\_RESET | 0: the bram control logic is in operational mode  1: the bram control logic is in reset mode |
| 4 | REMAP\_ENABLE | 0: no pixel shuffling, keep the sensor pixel output sequence  1: reorder the pixels from 0 to 4176 per line |
| [3:2] | RESERVED | Not used |
| 1 | FIFO\_RESET | 0: the fifo remapper is in operational mode  1: the fifo remapper is in reset mode |
| 0 | FIFO\_ENABLE | 0: disable the remapper output fifo: no data is processed  1: enable the remapper output fifo: pixel data is processed and will be send out on the axi streaming bus |

### FIFO\_STATUS: 0x0004

|  |  |  |
| --- | --- | --- |
| **Bit Offset** | **Field Name** | **Description** |
| [31:5] | RESERVED | Not used |
| 4 | FIFO\_AFULL | FIFO almost full flag |
| 3 | FIFO\_FULL | FIFO full flag |
| 2 | FIFO\_AEMPTY | FIFO almost empty flag |
| 1 | FIFO\_EMPTY | FIFO empty flag |
| 0 | FIFO\_ERROR | FIFO error flag |

### BRAM\_STATUS: 0x0008

|  |  |  |
| --- | --- | --- |
| **Bit Offset** | **Field Name** | **Description** |
| [31:15] | RESERVED | Not used |
| [14] | BRAM\_FULL | Full flag: all 4 line buffers are filled. |
| [13:12] | RESERVED | Not used |
| [11:8] | VALID | Flag per line buffer to indicate valid data is present. |
| [7:4] | CLEAR\_VALID\_ERROR | Flag per line buffer; error raised when clearing a non-valid buffer |
| [3:0] | SET\_VALID\_ERROR | Flag per line buffer; error raised when setting a (previously set) valid buffer |

# Programming Sequence

* Reset the FIFO by toggling bit 1 of the CONFIG register.
* Initialize the control BRAM block by toggling bit 5 of the CONFIG register.
* Set BRAM\_READOUT\_MODE, BRAM\_PIXEL\_0\_LANE and REMAP\_ENABLE according to your use-case.
* Enable the FIFO by putting bit 0 of the CONFIG register (0x00) to ‘1’.