GM8136

SCALER

User Guide
Rev.: 1.0
Issue Date: November 2013

REVISION HISTORY

GM8136 Scaler User Guide

Date	Rev.	From	To
Nov. 2013	1.0	-	Original

Printed in Taiwan 2013

Grain Media and the Grain Media Logo are trademarks of Grain Media, Inc. in Taiwan and/or other countries.
Other company, product and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in implantation or other life support application where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Grain Media's product specification or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Grain Media or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will Grain Media be liable for damages arising directly or indirectly from any use of the information contained in this document.

Grain Media, Inc.
5F, No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Grain Media's home page can be found at:
http://www.grain-media.com

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

This chapter contains the following sections:

1.1 Overview

Scaler300 in GM8136 is used to resize and enhance the video data. By using DMA, scaler300 can read the video data on AMBA AXI, and then write back the results. With the resizing ability, users can size-down or size-up an image to the necessary resolution or perform the aspect ratio conversion. The window clipping function can clip an interested region from an image before or after sizing down. Moreover, the luminance compensation, sharpness, and color saturation controls can make an image that is more pleasing to the human eyes.

1.2 Features

Scaler300 contains the following features:

- Supports maximum image size of 4096x4096
- Supports cropping function for front-end image source
- Supports cropping function for back-end image
- Supports scaling-down/up function with two output resolutions per scan line and bypass function with two output resolutions per scan line
- Supports input formats of YCbCr 4:2:2
- Supports 1-D false color suppression
- Supports 1-D denoise
- Supports smooth and sharpness
- Supports field scaling (Top/Bottom offsets)
- Supports frame-to-field extraction
- Supports up to eight mask windows in one frame with transparency degree control
- User programmable 8-type palette color for mask

1.3 Block Diagram

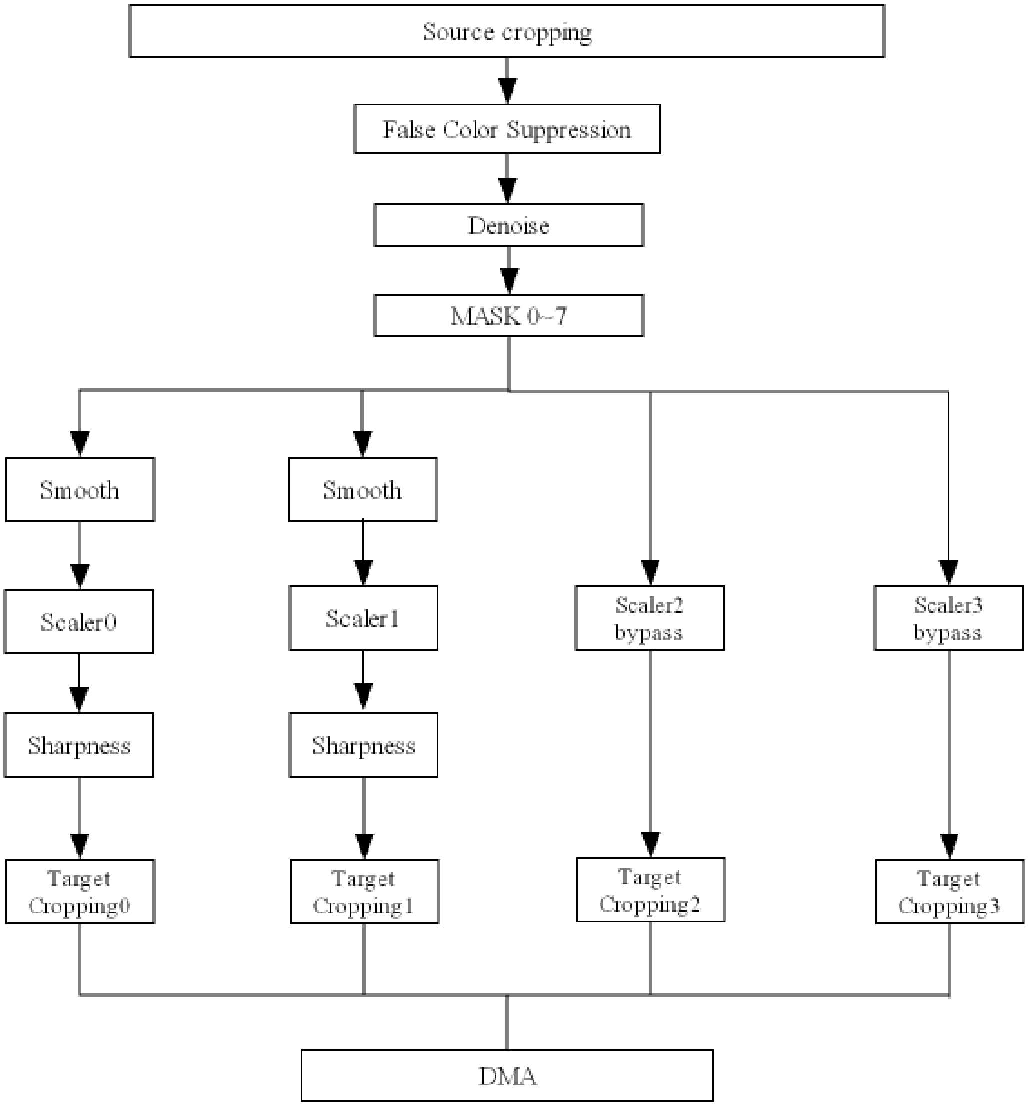


Figure 1.1 Block Diagram of Scaler300

Chapter 2

Scaler300 Driver Module

In the GM8136 SDK release package, users can find the source code and kernel module of scaler300 from "/module/scaler300".

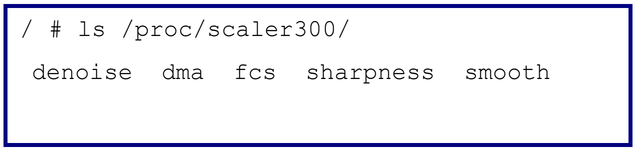
The scaler300 driver module contains the following part:

- fscaler300.ko
  This is the scaler300 core. It includes the scaler300 hardware control layer, middleware (GM_Graph), communication layer, and export library layer.

Chapter 3

Scaler300 Proc Nodes

The scaler300 module provides the proc nodes. Users can read the information of scaler300 or setup the configuration through these nodes. A sample of these nodes is listed below.



```
/ # ls /proc/scaler300/
 denoise  dma  fcs  sharpness  smooth
```

Figure 3-1. Proc Nodes of Scaler300 Module

Table 3-1 shows the proc node components of the scaler300 module.

Table 3-1. Proc Node Components of Scaler300 Module

/proc/scaler300	fcs	enable
		param
	denoise	enable
		param
	smooth	param
	sharpness	param
	dma	param

3.1 /proc/scaler300/fcs Proc Node

The scaler300 hardware supports the false-color suppression. Users can use these fcs proc nodes to adjust the fcs parameter for reducing the color noise.

3.1.1 Enable

Users can use the /proc/scaler300/fcs/enable node to control the fcs function.

Usage:

- Get the current value

  cat /proc/scaler300/fcs/enable

```
/ # cat /proc/scaler300/fcs/enable
FCS Enable : 0
```

- Set the value to enable/disable fcs

  echo [value] > /proc/scaler300/fcs/enable

  Value: 0 for disable, 1 for enable

3.1.2 param

Users can use the /proc/scaler300/fcs/param node to get and set the parameter for the fcs function.

Usage:

- Get the current value

  cat /proc/scaler300/fcs/param

```
/# cat /proc/scaler300/fcs/param


=== FCS Parameter ===
 [00]LV0_THRED : 0xfa
 [01]LV1_THRED : 0xc8
 [02]LV2_THRED : 0xc8
 [03]LV3_THRED : 0x32
 [04]LV4_THRED : 0x28
 [05]GREY_THRED: 0x384
```

- Set the param_id value to switch the parameter value for channel fcs

  echo [param_id] [value] > /proc/scaler300/fcs/param

  param_id:

  0: LV0_THRED, threshold for Y band width + C band width

  1: LV1_THRED, threshold for the first element of Y band width

  2: LV2_THRED, threshold for Y band width

  3: LV3_THRED, threshold for C band width

  4: LV4_THRED, threshold for C level

  5: GREY_THRED, threshold for the local grey area decision

3.2 /proc/scaler300/denoise Proc Nodes

The scaler300 hardware supports the 1-D de-noise function. Users can use these de-noise proc nodes to adjust the de-noise parameter for reducing the noise.

3.2.1 enable

Users can use the /proc/scaler300/denoise/enable node to control the de-noise function.

Usage:

- Get the current value

  cat /proc/scaler300/denoise/enable

```
/# cat /proc/scaler300/denoise/enable


DeNoise_Enable: 0
```

- Set the value to disable/enable the de-noise function

  echo [value] > /proc/scaler300/denoise/enable

  Value: 0 for disable, 1 for enable

3.2.2 param

Users can use the /proc/scaler300/denoise/param node to get and set the parameter of the de-noise function.

Usage:

- Get the current value

  cat /proc/scaler300/denoise/param

```
/# cat /proc/scaler300/denoise/param


=== DeNoise Parameter ===
 [00]DN_GEOMATRIC     : 0x0
 [01]DN_SIMILARITY    : 0x0
```

```
[02]DN_ADAPTIVE     : 0x1
[03]DN_ADAPTIVE_STEP: 0x10
```

- Set the param_id value to switch the parameter value of the de-noise function

  echo [param_id] [value] > /proc/scaler300/denoise/param

  param_id:

  0: DN_GEOMATRIC is 1D DN strength based on the distance. 0 means bypass.

  1: DN_SIMILARITY is 1D DN strength based on the difference. 0 means bypass.

  2: DN_ADAPTIVE disables/enables the adaptive denoise (0: Diable, 1: Enable)

  3: DN_ADAPTIVE_STEP is de-noise adaptive step size (Size: 4, 8, 16, 32, 64, 128, and 256)

3.3 /proc/scaler300/sharpness Proc Nodes

The scaler300 hardware supports sharpness for each path. Users can use these sharpness proc nodes to strength the edges of an image.

3.3.1 param

Users can use the /proc/scaler300/sharpness/param node to get and set the parameter of sharpness.

Usage:

- Get all current sharpness path values

  cat /proc/scaler300/sharpness/param

```
/# cat /proc/scaler300/sharpness/param
<Path#0>
 [00]SHARP_ADAPTIVE_ENABLE      : 0x1
 [01]SHARP_PARAM_RADIUS         : 0x0
 [02]SHARP_PARAM_AMOUNT         : 0x16
 [03]SHARP_PARAM_THRED          : 0x5
 [04]SHARP_PARAM_ADAPTIVE_START: 0x2
 [05]SHARP_PARAM_ADAPTIVE_STEP : 0x3
```

- Set the param_id value to change the parameter value of sharpness

  echo [path] [param_id] [value] > /proc/scaler300/sharpness/param

  path:

  0: Sharpness path 0

  1: Sharpness path 1

  param_id:

  0: SHARP_ADAPTIVE_ENABLE, disable/enable the adaptive sharpness

  (0: Disable, 1: Enable)

  1: SHARP_PARAM_RADIUS is sharpness radius. 0 means bypass.

  2: SHARP_PARAM_AMOUNT is sharpness amount.

  3: SHARP_PARAM_THRED is sharpness dn level.

  4: SHARP_PARAM_ADAPTIVE_START is sharpness adaptive starting strength.

  5: SHARP_PARAM_ADAPTIVE_STEP is sharpness adaptive step.

3.4 /proc/scaler300/smooth Proc Nodes

The scaler300 hardware supports the smooth function. Users can use these proc nodes to smooth an image.

3.4.1 param

Users can use the /proc/scaler300/smooth/param node to get and set the parameter of the smooth function.

Usage:

- Get all current smooth path values

  cat /proc/scaler300/smooth/param

```
/# cat /proc/scaler300/smooth/param
<Path#0>
 [00]SMOOTH_HORIZONTAL_Strength     : 0x0
 [01]SMOOTH_VERTICAL_Strength       : 0x0
```

Usage:

- Set the param_id value to change the parameter value of the smooth function

  echo [path] [param_id] [value] > /proc/scaler300/smooth/param

  path:

  0: Smooth path 0

  1: Smooth path 1

  param_id:

  0: SMOOTH_HORIZONTAL_Strength, smooth horizontal strength. 0 means bypass.

  1: SMOOTH_VERTICAL_Strength, smooth vertical strength. 0 means bypass.

3.5 /proc/scaler300/dma Proc Nodes

The scaler300 hardware supports the setting for the DMA read and write wait intervals. Users can use these proc nodes to set the DMA read and write wait intervals.

3.5.1 param

Users can use the /proc/scaler300/dma/param node to get and set the parameter for the DMA read and write wait intervals.

Usage:

- Get all current smooth path values

  cat /proc/scaler300/dma/param

```
/# cat /proc/scaler300/dma/param
=== DMA Parameter ===
[00]WC_WAIT_VALUE : 0x200
[01]RC_WAIT_VALUE : 0x200
```

Usage:

- Set the param_id value to change the parameter value of DMA

  echo [path] [param_id] [value] > /proc/scaler300/dma/param

  param_id:

  0: WC_WAIT_VALUE is DMA write channel wait interval for each DMA write burst.

  1: RC_WAIT_VALUE is DMA read channel wait interval for each DMA read burst.

Chapter 4

Scaler300 Mask

This chapter contains the following sections:

Scaler300 supports eight independent mask windows. Users can use these mask windows to mask an image region within one frame. The Mask function will be performed before scaling; therefore, all output paths in one frame will display the same mask window on the same region of scaler300. Mask windows have priority. Mask window 0 has the highest priority, while mask window 7 has the lowest priority. In other words, if mask window 0 and mask window 1 have overlay on the same region, mask window 0 has higher priority, mask window 1 will not show, only mask window 0 will show the overlay region.

Scaler300 supports eight programmable palette colors. These palette colors are used on the mask window color. Users can customize the palette colors through middleware API. The format is YCbCr422.

4.1 Mask Features

Scaler300 contains the following features:

- Supports up to eight mask windows in one frame
- User-programmable eight palette colors for mask
- Supports eight window transparency types
- Mask window 0 has highest priority and mask window 7 has lowest priority.

4.2 Mask Parameters

The following items are the mask parameters:

- **X_pos/Y_pos :** Users can set the X/Y start positions of a mask window
- **X_end/Y_end :** Users can set the X/Y end positions of a mask window
- **Color:** Users can select 8 palette colors for a mask window.
- **Alpha:** Users can set the transparency level of a mask window.
  Selections of Alpha: 0:100%; 1: 87.5%; 2: 5%; 3: 62.5%; 4: 50%;
  5: 37.5%; 6: 25%; and 7: 0%

Figure 4-1 shows X_pos/Y_pos and X_end/Y_end that can be used on a mask window, while Figure 4-2 shows the color and alpha parameters on a mask window.

Figure 4-1 and Figure 4-2 show the following detailed mask window settings:

- Set the color of mask window 0 to green and alpha to 7 (0%), therefore, mask window 0 is green color.
- Set the color of mask window 1 to red and alpha as 1(87.5%), therefore, the mask window 1 is red color.
- Users can see that mask window 0 and mask window 1 have overlay region. Because mask window 0 has the highest priority, the overlay region only shows mask window 0.

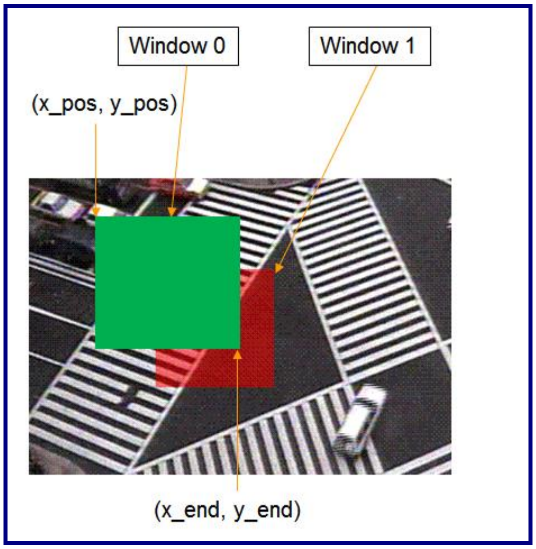


Figure 4-1. Mask Function 1 of Scaler300

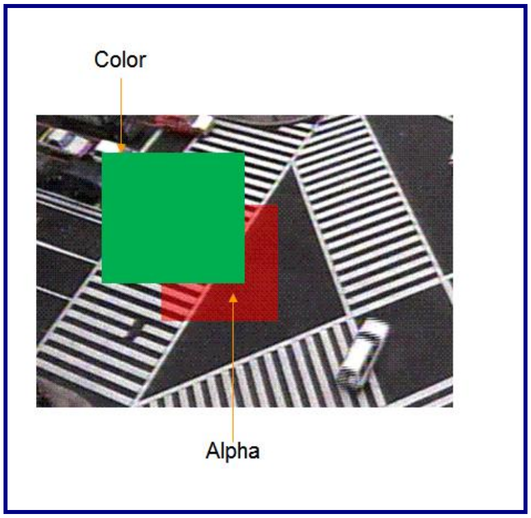


Figure 4-2. Mask Function 2 of Scaler300