minerva
technology

UpScaler IP Core

Specification

Contents

1. Terms, Definitions, Abbreviations

Screen – surface of the display device. Represented by a set of pixels. Described by vertical resolution (number of pixels per line) and horizontal (row number).

Pixel – point of the image in RAM or on a screen. Defined by coordinates on the screen and color.

Picture – data array in the RAM, which is a series of color characteristics of the pixels with the addition of characteristics of "transparency" (alpha channel). It is characterized by costs of memory of one pixel (in bytes), length of a line (in pixels or bytes), quantity of lines.

Window – rectangular area on the screen on which the picture is carried out. It is characterized by an arrangement of the top left corner on the screen and the sizes (in pixels) across and verticals.

Stripe – set of two (or more) picture lines.

2. Introduction

2.1 Overview

UpScaler IP Core IP Core realizes scaling up (dimension increasing) function of the picture. The scaling factors are arbitrary. The IP Core is intended for use in video output path to the display device.

2.2 Features

- The image quality on the PSNR metric scaling to more than 2 times is better than the bilinear interpolation on 6-8 dB and on 0.5-1 dB better than – bicubic interpolation method.
- Supports RGB and YUV color spaces with an arbitrary bit color component (the bit is set the parameters of synthesis).
- Output resolution both horizontally and vertically is not limited.
- The maximum input horizontal resolution is determined by the volume of the internal buffer and set the parameters of the synthesis.
- The maximum input vertical resolution is not limited. It only will affected on the performance.
- Scaling coefficients arbitrary and can be changed dynamically.

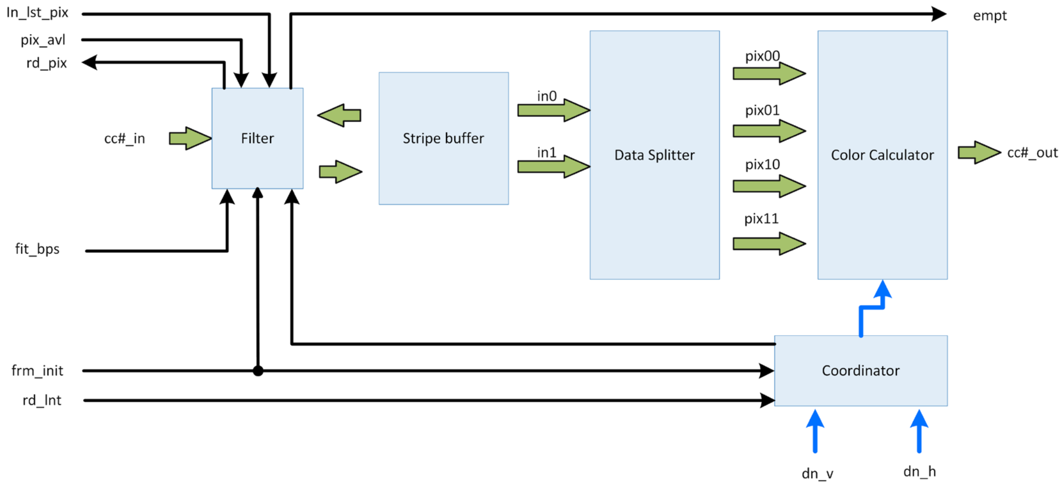
3. Functional Description

3.1 Block Diagram

A simplified block diagram of UpScaler IP Core is shown in Figure 1. UpScaler includes the following functional blocks:

- Correction filter (Filter);
- Stripe Buffer;
- Pixel data splitter (Data Splitter);
- Calculator of the synthesized pixel color (Color Calculator);
- Generator of the synthesized pixel coordinates (Coordinator).

Figure 1 UpScaler block diagram



The Stripe Buffer contains two lines of the input picture. Maximum length of a line of the input picture is limited by volume of the buffer and is set through synthesis parameter.

The Color Calculator consists of three untied identical blocks. One block for one color component.

Vertical and horizontal scaling coefficients are set through relative distances between pixels of the scalled picture (**rlv_dsc_h, rlv_dsc_v** respectively). Relative distances are presented in the form of fixed-point non-integer bits and defined by the formula:

rlv_dsc = (in_size-1)/(out_size-1)

where: **in_size** – the size of a row (column) of the input picture in pixels.
out_size – the size of a line (column) of the output picture in pixels.
The formula shows that the relative distance is a value close to 1 / K, where K - scaling coefficient.

3.2 Functioning

Signal the beginning of the picture (**frm_init,** corresponds to the vertical sync pulse) initializes the whole Core.
If the signal (**pix_avl**) is active, picture data will be written to the Stripe Buffer.
rd_pix signal confirms writing pixel data to the Strip Buffer. Writing latency is one clock cycle. **ln_lst_pix** signal set to active when last pixel of the line is written to Strip Buffer.

Interpolation of the line begins and ends according to the signal read line (**rd_lnt**). Interpolated data will output though 4 cycles delay. Interpolation is carried out line by line. It is not required second line data for the first line of picture interpolation. Further, the Stripe Buffer must contains full previous line and corresponded data of the current line. To compute the color of the synthesized pixel, four reference pixel passed through the filter – two of the current and two from the next line. The division lines on the pixels performs by Data Splitter. Calculator calculates color of synthesized pixel, in accordance with its coordinates relative to the four reference pixels.

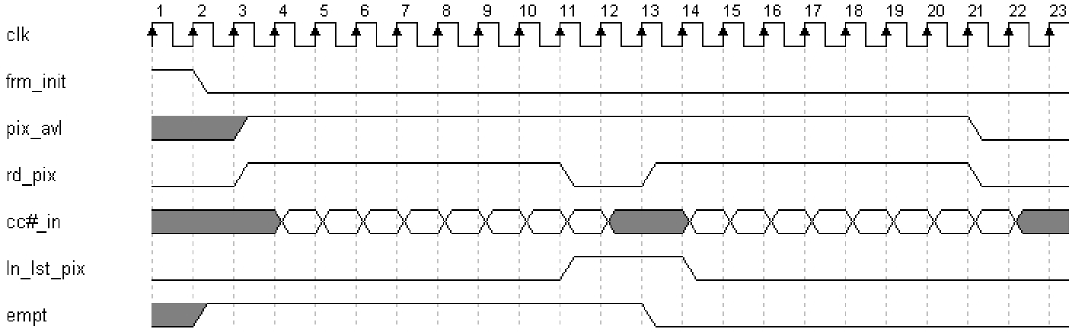
Coordinator calculates the synthesized pixel coordinates and reference pixels sampling. For calculation is required relative distance between the pixels of the resulting picture (**rlv_dss_h, rlv_dss_v**). Values of **rlv_dss_h**, **rlv_dss_v** must be valid within interpolation line and have constant values for all lines of a picture.

Signal **flt_bps** forces Core to functioning in linear interpolation mode.

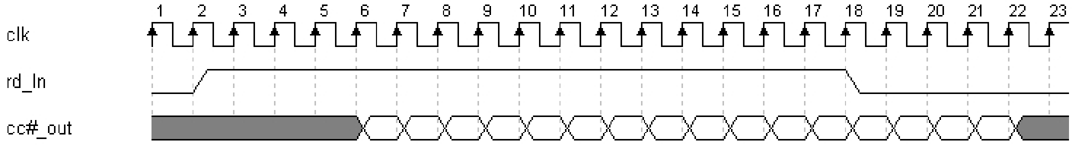
Signal **empt** informs that there is no data in the Strip Buffer.

3.3 Waveforms

Figure 2 shows waveform of two first line of picture loading (8 pixels length) to the Strip Buffer.



Figures 3 shows unloading process of synthesized line (scale factor 2, 16 pixel length).



4. Implementation

4.1 Synthesis Parameters

Synthesis parameters are shown on the Table 1.

Table 1 – Synthesis parameters

Parameter	Valid values	Description
SYNT_A_RST	true/false	Allow the synthesis of asynchronous reset signal [1,7]
CC1_WIDTH	> 0	The number of bits of the first color component [6]
CC2_WIDTH	> 0	The number of bits of the second color component [6]
CC3_WIDTH	> 0	The number of bits of the third color component [6]
CC_WIDTH_F	>= 0	The number of bits of the fractional part of floating point numbers [2,6]
COEF_WIDTH	> 0	The number of bits of scaling factors [3,6]
LN_WIDTH	> 0	The number of bits required to specify the address of the pixel in the line [4,5]

[1] Assigning an asynchronous reset signal of constant low-level lead to the inability to use the hardware units multiply-accumulate (MAC) for the logic of the Color Calculator

[2] The experiment showed that the accuracy of the calculations does not have a noticeable effect on the quality scale, so this option should be set to 0.

[3] Reasonable value from 9 to 11.

[4] The number of pixels in one line of the buffer strip 2 ^ LN_WIDTH, the number of pixels in the buffer strip 2 * 2 ^ LN_WIDTH, the volume of buffer strips 2 * 2 ^ LN_WIDTH * (CC1_WIDTH + CC2_WIDTH + CC3_WIDTH) bits.

[5] For a reasonable resource consumption block memory, pay particular attention to the pixel bit depth and length of the line.

[6] To save the resources of hardware multiplier-accumulators (18x18 bits) must fulfill the conditions: COEF_WIDTH <= 17; CC # _WIDTH + CC_WIDTH_F <= 14.

[7] Collaborative synthesis of synchronous and asynchronous reset leads to the additional overhead of the FPGA resources. Preference should be given to synchronous restart of a less resource-intensive.

4.2 Interface Description

External interfaces UpScaler IP Core are shown in Figure 4 and described in Table 2.

Figure 4 – Interfaces

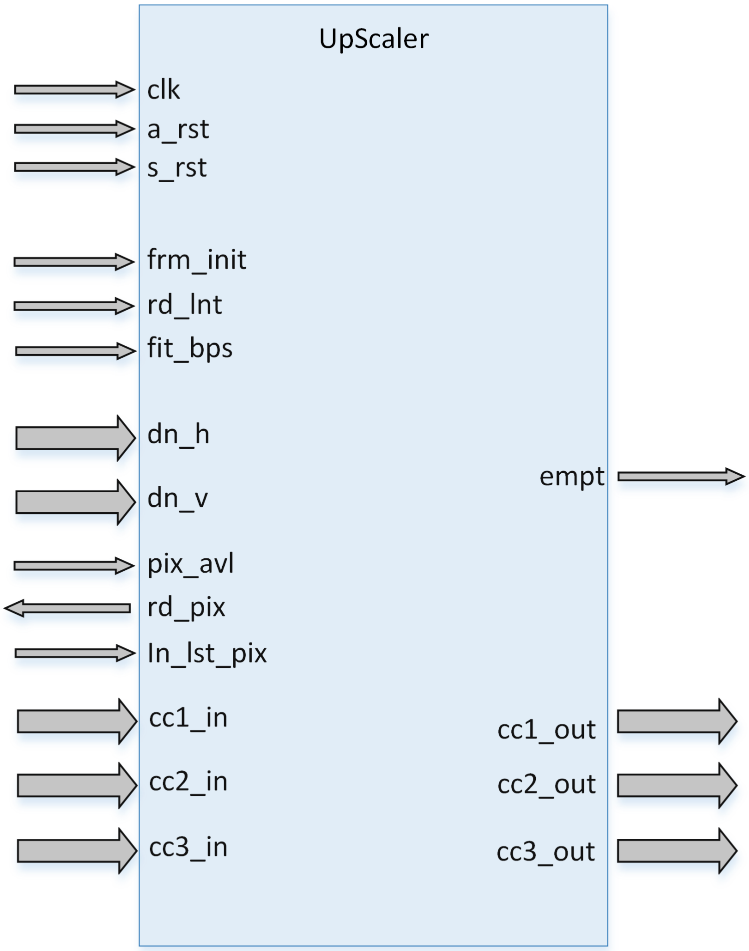


Table 2 – in/output signals description

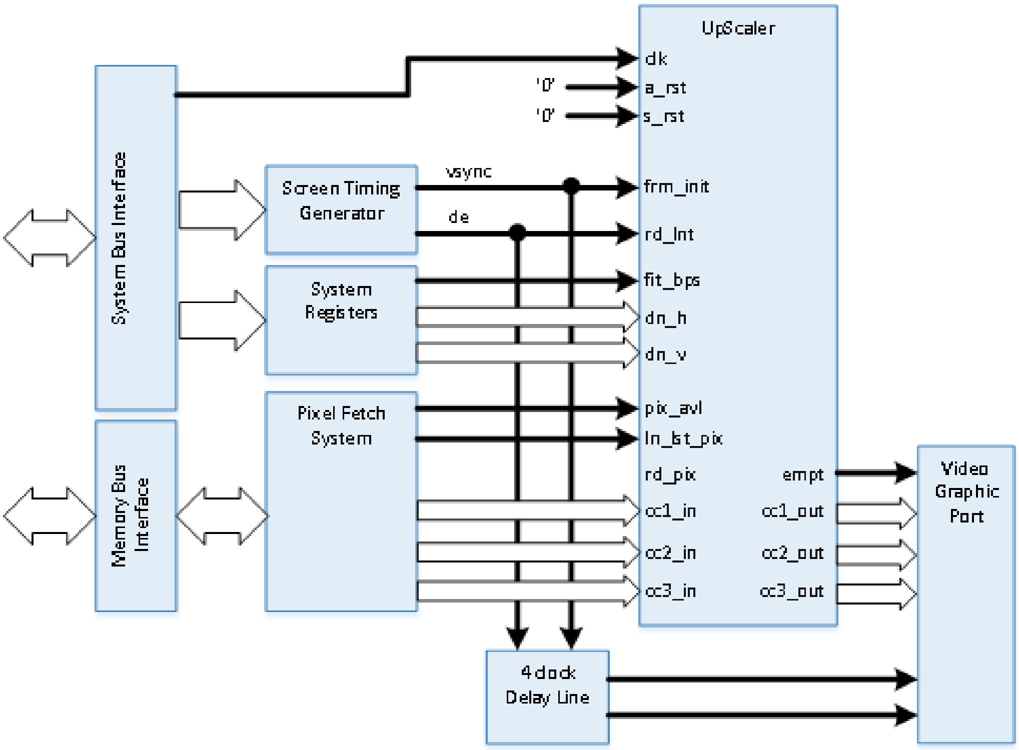
Signal	Direction	Description
a_rst	Input	Asynchronous reset signal
clk	Input	Clock signal
s_rst	Input	Synchronous reset signal
Control		
frm_init	Input	Start picture frame flag
rd_ln	Input	Pixel line reading
flt_bps	Input	Correcting filter disabling. Core works I linear interpolation mode.
dn_h[COEF_WIDTH-1:0]	Input	Horizontal scaling coefficient value
dn_v[COEF_WIDTH-1:0]	Input	Vertical scaling coefficient value
Input port		
pix_avl	Input	Input data are available
rd_pix	Output	Ready for pixel data reading
ln_lst_pix	Input	Last pixel of the line is coming
cc1_in[CC1_WIDTH-1:0]	Input	First color component data
cc2_in[CC2_WIDTH-1:0]	Input	Second color component data
cc3_in[CC3_WIDTH-1:0]	Input	Third color component data
Output port		
empt	Output	Buffer is empty
cc1_out[CC1_WIDTH-1:0]	Output	First color component data
cc2_out[CC2_WIDTH-1:0]	Output	Second color component data
cc3_out[CC3_WIDTH-1:0]	Output	Third color component data

4.3 Application Diagram

Application Diagram UpScaler IP Core (example) is shown in Figure 5. Example shows how to use Core in application for rendering picture to the video display device. In order to run IP Core the following external blocks are required: Screen Timing Generator for sweep generation, registers for storing scaling parameters (System Registers), block of delivery of pixel data (Pixel Fetch System) from the frame buffer and physical layer of video port.

It is not necessary to use the reset signals to render the output image on the display device, because leaving the state of uncertainty will happen in one frame after switching on.

Figure 5. application diagram



4.4 Resource and Performance

Resources expenses and maximum operating clock frequency (XST synthesis) is shown on Table 3. Asynchronous reset signal is not used. Showed below statistic is correct for the following synthesis parameters:

SYNT_A_RST = false;
CC1_WIDTH = 8;
CC2_WIDTH = 8;
CC3_WIDTH = 8;
CC_WIDTH_F = 0;
COEF_WIDTH = 10;
LN_WIDTH = 11 (2048 pixels).

Table 3 – performance and resources

Family	Chip	Clock frequency, MHz	Flip Flops/ LUTs	Slice Registe	RAMB16	MULT18X18, DSP48, DSP48E	Occupied Slices LUT/ LUT Flip Flop pairs
Virtex-4TM	XC4VLX25-10FF668	158	184 540		8	10	294
Virtex-5TM	XC5VLX30-1FF676	162		184	4	10	535 91
Kintex-7TM	XC7K70T-1FBG676	162		184	4	10	535 91

5. Additional information

5.1 Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com

5.2 Technical Support

For technical support, contact the Minerva Technical Support Team:
tsup@minerva-tech.com

5.3 Ordering Information

Please contact Minerva Sales department for pricing and additional information about the product: sales@minerva-tech.com