Helion
ENGINEERING VISION

Helion Standard HD ISP

Reference Design

User Manual
Lattice Embedded Vision Development Kit

July 27, 2017

Standard HD ISP

Reference Design for

Lattice Embedded Vision Development Kit

Description:

Helion Vision's IONOS Standard HD-ISP provides a turn-key solution for Lattice's Embedded Vision Development Kit and is one of five pre-configured plug and play Image Signal Processing Pipelines (ISP). It provides a full functional video preprocessing pipeline for basic video applications.

Function:

The standard HD-ISP image signal processing pipelines utilizing a basic element of library comprising more than 150 individual IP Cores. It provides a template for a fully customizable and configurable video processing projects and is sensor and resolution independent. In this reference design the 2k version is used.

Order Code ISP:
BE10029117(ISP-STHD-12)

Order Code Reference Design:
BE10029632(REF-EMVI-12)

ISP Features

- Full functional universal video pipeline containing:
  - Supports standard video timings like 720p60, 1080p60
  - Setup and input interface
  - Defect Pixel Correction
  - Debayer 5x5 High Quality (CFI)
  - Color Correction Matrix
  - RGB Gain
  - Color Saturation Matrix
  - Gamma Correction
  - Statistic modules
  - Auto Brightness
  - AWB Auto White Balancing
  - Configuration Interface
- 12 Bit per color processing
- Sensor independent
- Reference designs for
  - Lattice Embedded Vision Development Kit
  - Lattice HDR-60 Development Board

July 27, 2017

Contents

Chapter 1

Introduction

The Standard HD ISP intends to bring the Helion ISP pipeline closer to you. It provides a full functional video preprocessing pipeline for your image sensor.

It is equipped with a universal set of video preprocessing IP blocks to be used as a template for your own projects (i.e. HDR60 or EVDVK) or custom specific video processing pipelines. The Standard HD ISP is sensor and resolution independent.

The Standard HD ISP contains a easy to use and fully functional ISP solution for your designs. The implementation is very easy since you have no overloaded module header. It's resources optimized, while you still have full access to modify the ISP via a configuration interface. The IONOS ISP is sensor and resolution independent since all sensor dependent parts (sensor configuration and auto exposure) swapped out to a separate module. To work with the Standard HD ISP you just need a parallel video timing (e.g. 720p60 or 1080p60) as ISP input.

If you need a sensor port for your design get in touch with Helion and ask for generic sensor port.

1.1 Features of the IP used in the Standard HD ISP

- **Defect Pixel Correction**: **wbs_def_pix_correction_dpc**

  Defective hot and defective cold pixels are corrected with the defective pixel correction IP module. This IP module corrects the defective pixel with interpolated values based on neighbor pixels of the same color channel.
  Typical correction methods include detection of cold or hot pixels using median estimation on immediate pixel neighborhood.

  For further information and details see the related datasheet for wbs_def_pix_correction_dpc.

- **Debayer 5x5 High Quality (CFI)**: **wbs_debayer_5x5_cfi_5hq**

  This IP core generates full RGB output for each image pixel. The output of the **Debayer 5x5 High Quality (CFI)** is represented in a standard bayer format and is interpolated using a 5x5 sliding window to calculate the RGB results. The data stream width of input and output is configurable and also suitable for HDR application.

  The incoming image data is provided in raw bayer format. To calculate the missing two color informations all surrounding pixels will be used.
  The practical results of this calculation are low-pass filter behavior of the color channels. The results of

the debayer algorithms will be improved by chosing different mathematical strategies or larger windows for averaging.

For further information and details see the related datasheet for wbs_debayer_5x5_cfi_5hq.

- **Color Correction Matrix** : **wbs_color_correction_matrix_ccm**

  In order to provide high quality images, acquisition is most important in respect to all further processing steps. A whole variety of available image sensors possess incorrect color rendition due to crosscolor effects and different spectral characteristics compared to human perception. This effects lead to wrong color images (e.g. green with too much blue).

  Color response correction on each RGB channel by a 3x3 correction matrix multiplication.

$$\begin{aligned} red_{out} &= rr * red_{in} - rg * green_{in} - rb * blue_{in} \\ green_{out} &= -gr * red_{in} + gg * green_{in} - gb * blue_{in} \\ blue_{out} &= -br * red_{in} - bg * green_{in} + bb * blue_{in} \end{aligned}$$

  Important: There are two CCM modules used in Standard HD ISP. One for classic cross color correction and one for color saturation

  For further information and details see the related datasheet for wbs_color_correction_matrix_ccm.

- **RGB Gain** : **wbs_rgb_gain**

  Many sensors enable shutter control but only through coarse quantization steps. To achieve a smooth impression with automatic shutter control, a very fine gain adjustment is needed. Especially Auto Exposure(AE) and Auto White Balancing(AWB) functions require different and very fine gain-steps for each color channel. For every color channel a separate gain factor will be applied.

$$\begin{aligned} red_{out} &= gain_{red} * red_{in} \\ green_{out} &= gain_{green} * green_{in} \\ blue_{out} &= gain_{blue} * blue_{in} \end{aligned}$$

  For further information and details see the related datasheet for wbs_rgb_gain.

- **Gamma Correction** : **wbs_gamma_correction_gc**

  Pixels are illuminated in a linear way. To provide pixel data to common video systems a conversion to a non-linear encoding scheme may be needed.

  Gamma Correction provided conversion of a RGB-input signal from linear to non-linear value and vice versa. This function uses arbitrary **Gamma Correction** factors.

  For further information and details see the related datasheet for wbs_gamma_correction_gc.

- **RGB Statistics** : **wbs_rgb_hdr_stat**

  This IP core provides video statistics of the three color channels to be used with the auto-white-balance (AWB) routine. This routine require information like arithmetic mean values of the color channels etc. This IP core delivers the pixel sum and pixel numbers for each color channel within a freely configurable region of interest (ROI). This allows for closer examination of image areas. Inbounds are used to reject unwanted pixel values.

For further information and details see the related datasheet for wbs_rgb_hdr_stat.

Chapter 2

Functional Description

2.1 Block Diagram of the Standard HD ISP

Figure 2.1 illustrates the functional block diagram of the reference design. At the startup the ISP parameters get loaded from the *Default Memory* module. These parameters can be changed during operation by setting *UART* register commands (see chapter 4.2 for details). Also, a graphical user interface (GUI) will help to control the ISP parameters.

Figure 2.2 illustrates the functional block diagram of the video data stream processing. The top stage is the raw data input.

The first step is a defect pixel correction to eliminate hot or cold pixel in the stream. The defect pixel will be corrected with interpolated values of neighbor pixel of the same color channel.

A color filter interpolator (debayer) calculates raw RGB data from the incoming sensor data stream which will be adopted to a “human” color perception scheme by the color correction matrix operation.

To receive a high quality image with a correct color space the color correction matrix adjusts an incorrect color rendition due to cross color effects and different spectral characteristics compared to human perception.

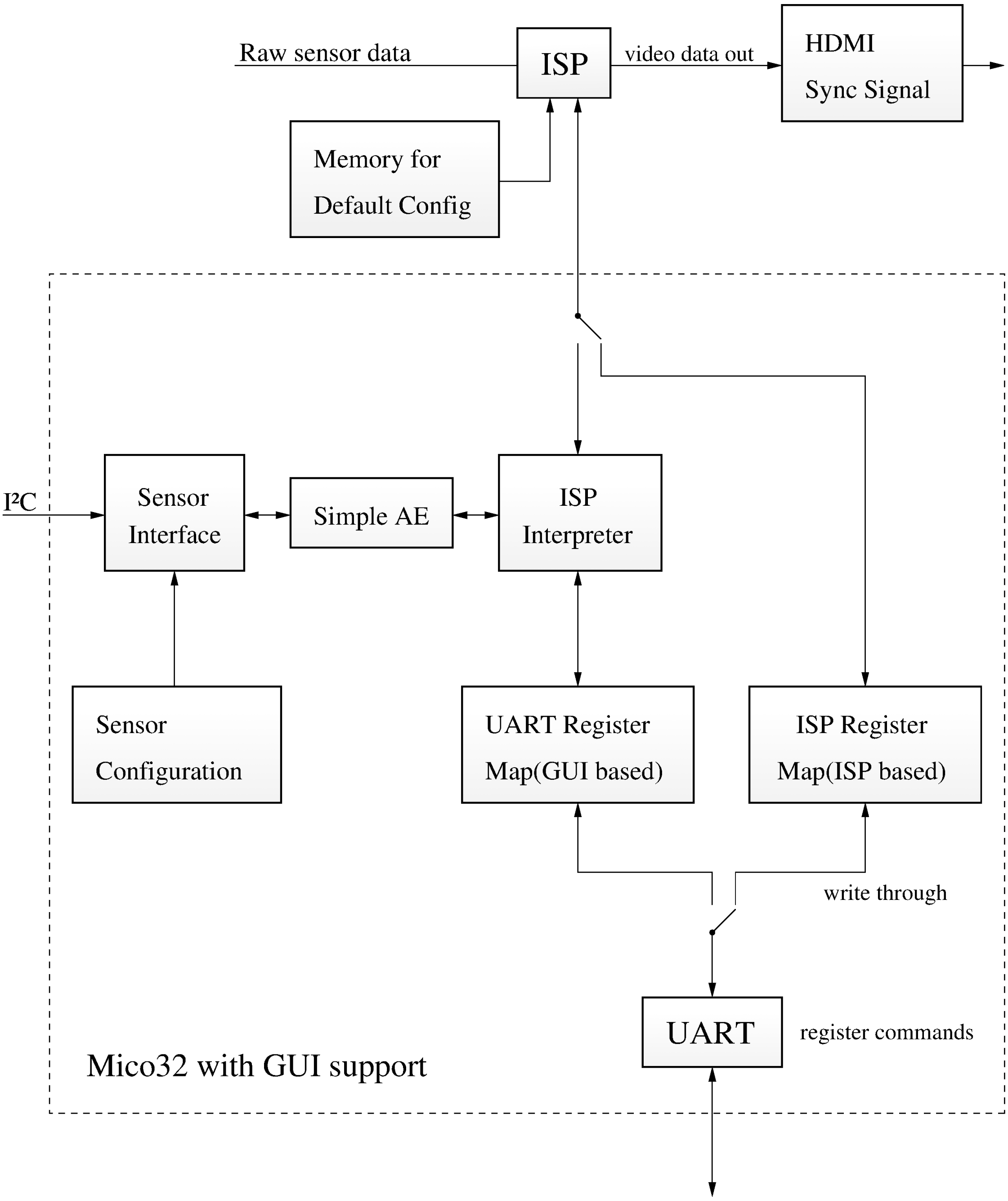
The RGB statistics modul derives statistic data per color channel to perform an auto white balance control operation within a feedback loop.

An additional RGB gain module enables a fine color channel adjustment. The following RGB statistics module delivers the required information for a feed-back controlled operation.

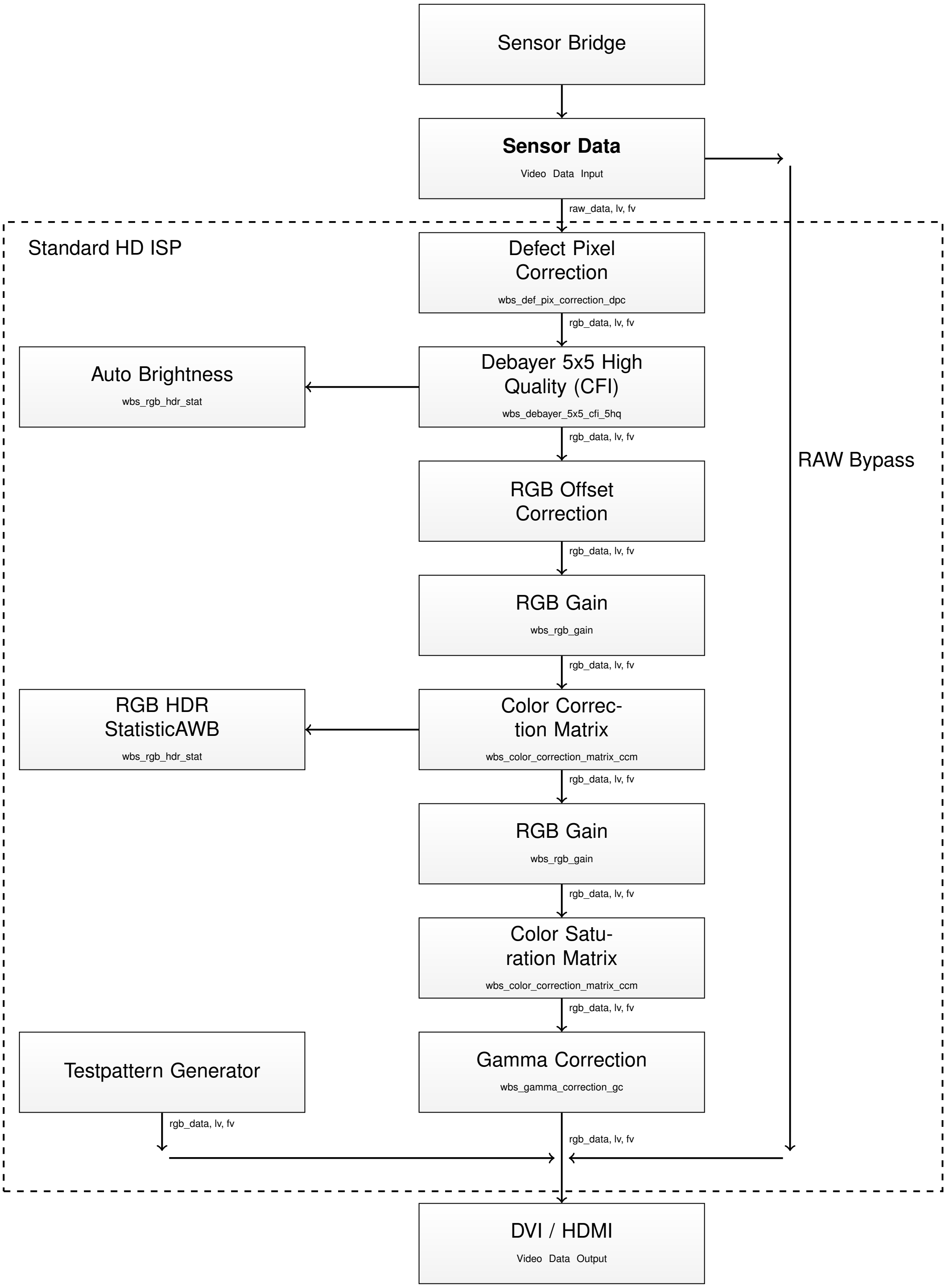
Furthermore, a look up table based on gamma transformation is imposed to adopt the luminance characteristics for a specific target video system.

In this reference design the video output will be realized by a DVI/HDMI compliant interface.

Each ISP can directly be modified by a configuration memory file.



Picture 2.1: Block diagram of the reference design.



Picture 2.2: Block diagram of IONOS Standard HD ISP

Chapter 3

Standard HD ISP Implementation and Parameter Description

3.1 Standard HD-ISP Implementation

To implement the HD-ISP in the design, it is necessary to connect a few ports like the video data bus to have the full funcionallity of the Helion IONOS ISP. In the following chapter the module header and the configuration file will be explained.

3.2 Modul Header of the Standard HD-ISP

Table 3.1 shows the required ports of the modul header. Helion will update this ISP regulary and add new functionallity. In one extended version I2C Master / Slave-, UART-, SPI-, memory interface and user IOs are planned to be included. Right now they are not connected inside the ISP, therefor connect 0'b0 to the inputs and leave the output unconnected. See Table 3.1 for more details.

Table 3.1: Portlist parameters

Parameter	Range	Description
main_clock_i		system clock, typical 27MHz or 54MHz
pixel_clock_i		image sensor clock, typical 74.24MHz or 148.5MHz
not_async_reset_i		asynchronous system reset, low active (connect to GSR Network)
raw_video_data_i	[11:0]	image sensor raw data input
raw_video_line_valid_i		image sensor line valid input
raw_video_frame_valid_i		image sensor frame valid input
spi_master_clock_o		internal not connected. Leave unconnected.
spi_master_not_select_o		internal not connected. Leave unconnected.
spi_master_mosi_o		internal not connected. Leave unconnected.
spi_master_miso_i		internal not connected. Connect 0'b0.
uart_txd_i		internal not connected. Connect 0'b0.
uart_rxd_o		internal not connected. Leave unconnected.
i2c_slave_clock_i		internal not connected. Connect 0'b0.
i2c_slave_not_select_o		internal not connected. Leave unconnected.
i2c_slave_sda_o		internal not connected. Leave unconnected.

continued on next page

continuation

Parameter	Range	Description
i2c_slave_sda_i		internal not connected. Connect 0'b0.
i2c_slave_scl_i		internal not connected. Connect 0'b0.
user_data_0_i	[31:0]	internal not connected. Connect 0'b0.
user_data_1_i	[31:0]	internal not connected. Connect 0'b0.
user_data_2_i	[31:0]	internal not connected. Connect 0'b0.
user_data_3_i	[31:0]	internal not connected. Connect 0'b0.
user_data_0_o	[31:0]	internal not connected. Leave unconnected.
user_data_1_o	[31:0]	internal not connected. Leave unconnected.
user_data_2_o	[31:0]	internal not connected. Leave unconnected.
user_data_3_o	[31:0]	internal not connected. Leave unconnected.
video_out_red_o	[7:0]	red video data output
video_out_green_o	[7:0]	green video data output
video_out_blue_o	[7:0]	blue video data output
video_out_line_valid_o		video line valid output
video_out_frame_valid_o		video frame valid output
config_memory_clock_i		default memory clock = system clock.
config_memory_address_i	[31:0]	input address for config memory register.
config_memory_we_i		write enable for config memory.
config_memory_data_i	[31:0]	data input for config memory.
config_memory_data_o	[31:0]	data output for config memory.
default_memory_clock_i		default memory clock = system clock.
default_memory_address_o	[31:0]	output address for default memory register.
default_memory_data_o	[31:0]	data output for default memory.

3.3 External ISP Register Overview

Table 3.3 provides an overview of the IONOS ISP register set. A detailed register and parameter description will be given in the following chapters. The register offset address multiplied with 4 is the I2C register address. Each register is 32 bit wide. Ways to access the registers

- Default: memfile
- On the fly: Config Interface, I2C

Register Offset Address	Parameter	Default Value [31:0]	Description
0	blocked reserved address	0x0	blocked reserved address
1	blocked reserved address	0x0	blocked reserved address
2	blocked reserved address	0x0	blocked reserved address
3	blocked reserved address	0x0	blocked reserved address
4	Revision	0x1	read revision: 1
5	width	0x780	image video width for 1080p, for 720p: 0x500
6	height	0x438	image video height for 1080p, for 720p: 0x2D0
7	hblank	0x118	numbers of horizontal blanking cycles for 1080p, for 720p: 0x172
8	vblank	0x2D	numbers of vertical blanking lines for 1080p, for 720p: 0x1E

will be continued

continuation

Register Offset Address	Parameter	Default Value [31:0]	Description
9	bayer_phase_dpc	0x1	enable: 0x1, disable: 0x0 for defect pixel correction
10	bayer_phase_debayer	0xD	used bayer phase
11	ccm_rr	0x1000000	ccm[1] Value red → red
12	ccm_rg	0x0	ccm Value red → green
13	ccm_rb	0x0	ccm Value red → blue
14	ccm_gr	0x0	ccm Value green → red
15	ccm_gg	0x1000000	ccm Value green → green
16	ccm_gb	0x0	ccm Value green → blue
17	ccm_br	0x0	ccm Value blue → red
18	ccm_bg	0x0	ccm Value blue → green
19	ccm_bb	0x1000000	ccm Value blue → blue
20	rgb_gain_pre_ccm_r	0x100	gain for red channel before ccm
21	rgb_gain_pre_ccm_g	0x100	gain for green channel after before ccm
22	rgb_gain_pre_ccm_b	0x100	gain for blue channel after before ccm
23	rgb_gain_pre_ccm_enable	0x1	enable rgb gain after before ccm
24	rgb_gain_after_ccm_r	0x100	gain for red channel after ccm
25	rgb_gain_after_ccm_g	0x100	gain for green channel after ccm
26	rgb_gain_after_ccm_b	0x100	gain for blue channel after ccm
27	rgb_gain_after_ccm_enable	0x1	enable, disable rgb gain after ccm
28	gamma	0x64	gamma value, 0x64 (d100) is equal to 1
29	saturation	0x64	saturation value, 0x64 (d100) is equal to 1
30	awb_stat_roi_col_start	0xA	ROI for AWB, start column pixel,
31	awb_stat_roi_col_end	0x4F6	ROI for AWB, end column pixel,
32	awb_stat_roi_row_start	0x32	ROI for AWB, start row pixel,
33	awb_stat_roi_row_end	0x2BC	ROI for AWB, end row pixel,
34	awb_stat_upper_bound_r	0x2D000	Upper threshold on red channel
35	awb_stat_upper_bound_g	0x2D000	Upper threshold on green channel
36	awb_stat_upper_bound_b	0x2D000	Upper threshold on blue channel
37	awb_stat_upper_lower_r	0x1	Lower threshold on red channel
38	awb_stat_upper_lower_g	0x1	Lower threshold on green channel
39	awb_stat_upper_lower_b	0x1	Lower threshold on blue channel
40	awb_stat_command	0x3F	
41	awb_enable	0x1	enable (0x1) the automatic white balance
42	manual_wb_enable	0x0	enable the manual white balance with 0x1
43	awb_speed	0x5A	Sets the speed for automatic white balance
44	auto_brightness_enable	0x1	enable (0x1), disable (0x0) automatic brightness
45	auto_brightness_max_gain	0x3F	set max gain for automatic brightness
46	auto_brightness_lum_target	0x7D0	set the target luminance
47	video_mode	0x0	switch between image data (0x0), testpattern generator (0x1) and raw bypass (0x2)
48	auto_brightness_stat_roi_col_start	0xA	ROI for Auto Brightness, start column pixel
49	auto_brightness_stat_roi_col_end	0x4F6	ROI for Auto Brightness, end column pixel
50	auto_brightness_stat_roi_row_start	0x32	ROI for Auto Brightness, start row pixel
51	auto_brightness_stat_roi_row_end	0x2BC	ROI for Auto Brightness, end row pixel

will be continued

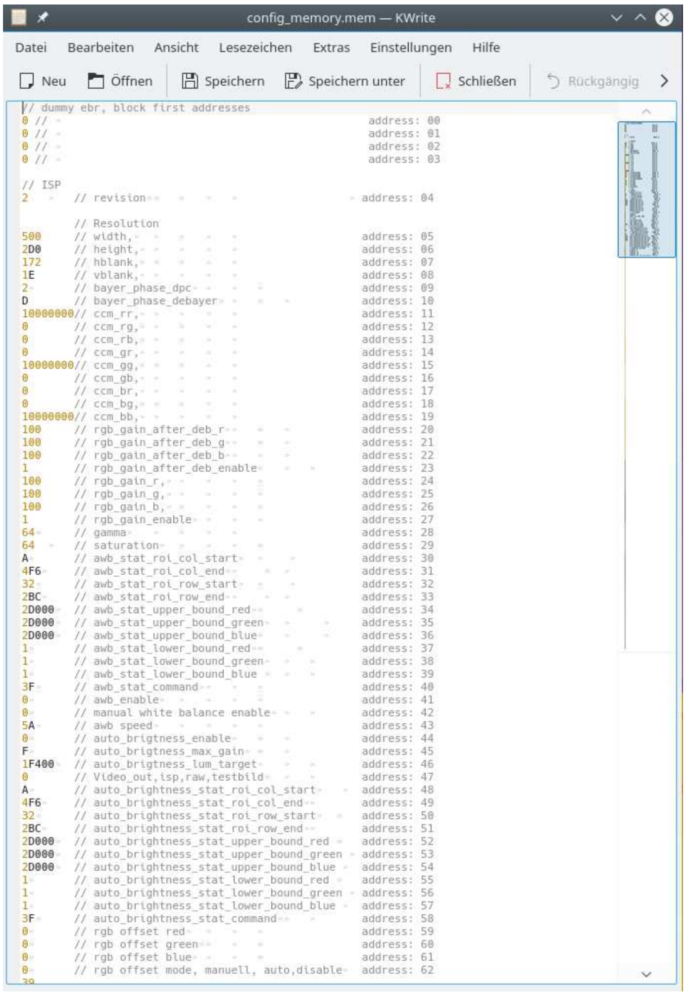
[1] ccm = color correction matrix

continuation

Register Offset Address	Parameter	Default Value [31:0]	Description
52	auto_brightness_stat_upper_bound_r	0x2D000	Upper threshold on red channel
53	auto_brightness_stat_upper_bound_g	0x2D000	Upper threshold on green channel
54	auto_brightness_stat_upper_bound_b	0x2D000	Upper threshold on blue channel
55	auto_brightness_stat_upper_lower_r	0x1	Lower threshold on red channel
56	auto_brightness_stat_upper_lower_g	0x1	Lower threshold on green channel
57	auto_brightness_stat_upper_lower_b	0x1	Lower threshold on blue channel
58	auto_brightness_stat_command	0x3F	
59	rgb_offset_red	0x0	value for the red channel offset
60	rgb_offset_green	0x0	value for the green channel offset
61	rgb_offset_blue	0x0	value for the blue channel offset
62	rgb_offset_mode	0x0	enable (0x1), disable (0x0)

3.4 Configuration Memory of the Standard HD-ISP

Picture 3.1: Configuration Memory File



```
// dummy ebr, block first addresses
0 //                                                address: 00
0 //                                                address: 01
0 //                                                address: 02
0 //                                                address: 03

// ISP
2       // revision                                address: 04

        // Resolution
500     // width,                                  address: 05
2D0     // height,                                 address: 06
172     // hblank,                                 address: 07
1E      // vblank,                                 address: 08
2       // bayer_phase_dpc                         address: 09
D       // bayer_phase_debayer                     address: 10
10000000// ccm_rr,                                 address: 11
0       // ccm_rg,                                 address: 12
0       // ccm_rb,                                 address: 13
0       // ccm_gr,                                 address: 14
10000000// ccm_gg,                                 address: 15
0       // ccm_gb,                                 address: 16
0       // ccm_br,                                 address: 17
0       // ccm_bg,                                 address: 18
10000000// ccm_bb,                                 address: 19
100     // rgb_gain_after_deb_r                    address: 20
100     // rgb_gain_after_deb_g                    address: 21
100     // rgb_gain_after_deb_b                    address: 22
1       // rgb_gain_after_deb_enable               address: 23
100     // rgb_gain_r,                             address: 24
100     // rgb_gain_g,                             address: 25
100     // rgb_gain_b,                             address: 26
1       // rgb_gain_enable                         address: 27
64      // gamma                                   address: 28
64      // saturation                              address: 29
A       // awb_stat_roi_col_start                  address: 30
4F6     // awb_stat_roi_col_end                    address: 31
32      // awb_stat_roi_row_start                  address: 32
2BC     // awb_stat_roi_row_end                    address: 33
2D000   // awb_stat_upper_bound_red                address: 34
2D000   // awb_stat_upper_bound_green              address: 35
2D000   // awb_stat_upper_bound_blue               address: 36
1       // awb_stat_lower_bound_red                address: 37
1       // awb_stat_lower_bound_green              address: 38
1       // awb_stat_lower_bound_blue               address: 39
3F      // awb_stat_command                        address: 40
0       // awb_enable                              address: 41
0       // manual white balance enable             address: 42
5A      // awb speed                               address: 43
0       // auto_brigtness_enable                   address: 44
F       // auto_brigtness_max_gain                 address: 45
1F400   // auto_brigtness_lum_target               address: 46
0       // Video_out,isp,raw,testbild              address: 47
A       // auto_brightness_stat_roi_col_start      address: 48
4F6     // auto_brightness_stat_roi_col_end        address: 49
32      // auto_brightness_stat_roi_row_start      address: 50
2BC     // auto_brightness_stat_roi_row_end        address: 51
2D000   // auto_brightness_stat_upper_bound_red    address: 52
2D000   // auto_brightness_stat_upper_bound_green  address: 53
2D000   // auto_brightness_stat_upper_bound_blue   address: 54
1       // auto_brightness_stat_lower_bound_red    address: 55
1       // auto_brightness_stat_lower_bound_green  address: 56
1       // auto_brightness_stat_lower_bound_blue   address: 57
3F      // auto_brightness_stat_command            address: 58
0       // rgb offset red                          address: 59
0       // rgb offset green                        address: 60
0       // rgb offset blue                         address: 61
0       // rgb offset mode, manuell, auto,disable  address: 62
39
```

The Standard HD ISP can be configurated by a fixed parameter set → *.mem- file.

3.5 Blocked Address

The first 4 addresses are reserved and cannot be used.

Parameter	Address[dec]	Range/Options	Default
blocked reserved adress	00	-	blocked reserved adress, don’t use
blocked reserved adress	01	-	blocked reserved adress, don’t use
blocked reserved adress	02	-	blocked reserved adress, don’t use
blocked reserved adress	03	-	blocked reserved adress, don’t use

3.6 ISP Version / Revision

Register address 4 contains the IONOS ISP version / revision.

Table 3.4: ISP revision

Parameter	Address[dec]	Range/Options	Default
Revision	04	2	ISP revision

3.7 Video image related parameters

These registers are to set up the correct timing parameter.

Table 3.6: Video image related configuration and parameters

Parameter	Address[dec]	Range/Options	Default
width	05	1...0x800	0x500 = 0d1280
height	06	1...0x800	0x2D0 = 0d720
hblanks	07	1...0x800	0x172 = 0d370
vblanks	08	1...0x255	0x2D = 0d45

3.7.1 width

This parameter adjusts the pixel width of the image. It is limited to a max value of 2047. The picture width is needed for internal counter settings and configurations.

3.7.2 height

This parameter adjusts the pixel height of the image. It is limited to a max value of 2047. The picture height is needed for internal counter settings and configurations.

3.7.3 hblanks

Based on the calculation of all pixels, there is the need for a line buffer in the core. The effect of this line buffer is, that the latency of the output is at least one line. Additionally, the last line which is generated by the core, does no longer have picture synchronization input signals. For the last output line, these synchronization signals have to be generated by the core itself. To have influence on the blankings and to adapt those to the horizontal blanking this parameter must be used to set the manual horizontal blanking.

3.7.4 vblanks

The vblanks defines the the time between the end of the final line of a frame or field and the beginning of the first line of the next frame.

3.8 Defetive Pixel Correction and Debayer (CFI)

Table 3.8 lists the different input parameters to configure the Debayer Core core. Furthermore the function needs a parameter to adjust the debayer phase.

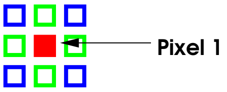
Table 3.8: Debayer configuration and parameters

Parameter	Address[dec]	Range/Options[dec]	Default
bayer_phase_dpc	09	0...0x1	0x1
bayer_first_phase	10	0...0x3	0x1

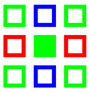
3.8.1 bayer_first_phase

This parameter adjusts the incoming bayer sequence. With four different cases, the first pixel of the first image line could be red, blue, green1 or green2. The 2-bit length bayer_first_phase parameter is used to encode the appropriate situation as follows:

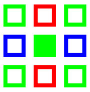
- “00”=First pixel is red



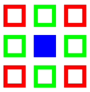
- “01”=First pixel is green in green-red line



- “10”=First pixel is green in green-blue line



- “11”=First pixel is blue



3.8.2 bayer_phase_dpc

This parameter enables the defect pixel correction IP.

3.9 Color correction matrix Parameter

Table 3.9 shows the different input parameters of the color correction matrix.

Table 3.10: CCM Configuration Parameters

Parameter	Address[dec]	Range/Options	Default
ccm_rr	11	0...0xFFFFFFFF	0x10000000
ccm_rg	12	0...0xFFFFFFFF	0
ccm_rb	13	0...0xFFFFFFFF	0
ccm_gr	14	0...0xFFFFFFFF	0
ccm_gg	15	0...0xFFFFFFFF	0x10000000
ccm_gb	16	0...0xFFFFFFFF	0
ccm_br	17	0...0xFFFFFFFF	0
ccm_bg	18	0...0xFFFFFFFF	0
ccm_bb	19	0...0xFFFFFFFF	0x10000000

3.9.1 ccm_rr_i, ... , ccm_bb_i

In order to provide high quality images, acquisition is most important in respect to all further processing steps. A whole variety of current image sensors possess incorrect color rendition due to crosscolor effects and different spectral characteristics compared to human perception. This effects leads to wrong color images (e.g. green with too much blue).

Color response correction on each RGB channel by a 3x3 correction matrix multiplication.

$$\begin{aligned} red_{out} &= rr * red_{in} - rg * green_{in} - rb * blue_{in} \\ green_{out} &= -gr * red_{in} + gg * green_{in} - gb * blue_{in} \\ blue_{out} &= -br * red_{in} - bg * green_{in} + bb * blue_{in} \end{aligned}$$

The gain factors were measured for this sensor in a foto laboratory, so they are sensor specific and do not need to be changed. If you want to deactivate the color correction matrix you have to set the parameters to the following values.

ccm_rr_i = 0x10000000
ccm_rg_i = 0x0
ccm_rb_i = 0x0
ccm_gr_i = 0x0
ccm_gg_i = 0x10000000
ccm_gb_i = 0x0
ccm_br_i = 0x0
ccm_bg_i = 0x0
ccm_bb_i = 0x10000000

3.10 Parameter RGB gain

In combination with the CCM, these registers are used for the internal Automatic White Balance (AWB) adjustment. The values are used as start values and in case of disabled AWB.

Table 3.12: RGB Gain configuration and parameters

Parameter	Address[dec]	Range/Options	Default
rgb_gain_pre_ccm_r	20	0...0x400	0x100
rgb_gain_pre_ccm_g	21	0...0x400	0x100
rgb_gain_pre_ccm_b	22	0...0x400	0x100
rgb_gain_pre_ccm_enable	23, Bit 0	0...0x1	0x1
rgb_gain_after_ccm_r	24	0...0x400	0x100
rgb_gain_after_ccm_g	25	0...0x400	0x100
rgb_gain_after_ccm_b	26	0...0x400	0x100
rgb_gain_after_ccm_enable	27	0...0x1	0x1

3.10.1 gain_red

The parameter gain_red is gaining the red channel with the gain_red factor:
$gain\_red\_factor = \frac{gained\_red}{0x100}$
Therefore the mathematical function is applied:
$data\_red\_o = gain\_red\_factor * data\_red\_i$

3.10.2 gain_green

The parameter gain_green is gaining the green channel with the gain_green factor:
$gain\_green\_factor = \frac{gained\_green}{0x100}$
Therefore the mathematical function is applied:
$data\_green\_o = gain\_green\_factor * data\_green\_i$

3.10.3 gain_blue

The parameter gain_blue is gaining the blue channel with the gain_blue factor:
$gain\_blue\_factor = \frac{gained\_blue}{0x100}$
Therefore the mathematical function is applied:
$data\_blue\_o = gain\_blue\_factor * data\_blue\_i$

3.11 Gamma

These registers set the gamma correction. The default value is 0x64 (d100), which sets the gamma curve to 1.00 (linear transmission). Example: To set the curve to 0.45 (normal TV operation), the value should be 0x2D (d45).

Table 3.14: Gamma configuration and parameters

Parameter	Address[dec]	Range/Options	Default
gamma	28	0...0x400	0x64

3.12 Saturation

This register sets the color intensity. The default value is 0x64 (d100), which sets the saturation to 1.00. For example: To set the saturation to 2.50, the value should be 0xFA (d250).

Table 3.16: Saturation configuration and parameters

Parameter	Address[dec]	Range/Options	Default
saturation	29	0...0x400	0x64

3.13 Automatic White Balance

These registers adjust the Automatic White Balance function.

Parameter	Address[dec]	Range/Options	Default
awb_stat_roi_col_start	30	0...0x500*	0xA
awb_stat_roi_col_end	31	0...0x500*	0x4F6
awb_stat_roi_row_start	32	0...0x780*	0x32
awb_stat_roi_row_end	33	0...0x780*	0x2BC
awb_stat_upper_bound_r	34	0...0xFFFFFF	0x2D000
awb_stat_upper_bound_g	35	0...0xFFFFFF	0x2D000
awb_stat_upper_bound_b	36	0...0xFFFFFF	0x2D000
awb_stat_upper_lower_r	37	0...0xFFFFFF	0xA000
awb_stat_upper_lower_g	38	0...0xFFFFFF	0xA000
awb_stat_upper_lower_b	39	0...0xFFFFFF	0xA000
awb_stat_command	40		0x3F
will be continued			

continuation

Parameter	Address[dec]	Range/Options	Default
awb_enable	41	0x00, 0x01	0x1
manual_wb_enable	42	0x00, 0x01	0x0
awb_speed	43	0x01...0x64	0x5A

*The ROI must fit into the image resolution.

3.13.1 awb_stat_roi_col_start

Start column of ROI.

3.13.2 awb_stat_roi_col_end

End column of ROI.

3.13.3 awb_stat_roi_row_start

Start row of ROI.

3.13.4 awb_stat_roi_row_end

End row of ROI.

3.13.5 awb_stat_upper_bound_r

Threshold for red pixel values. All pixels holding a value greater than the threshold are counted as saturated pixels.

3.13.6 awb_stat_upper_bound_g

Threshold for green pixel values. All pixels holding a value greater than the threshold are counted as saturated pixels.

3.13.7 awb_stat_upper_bound_b

Threshold for blue pixel values. All pixels holding a value greater than the threshold are counted as saturated pixels.

3.13.8 awb_stat_upper_lower_r

Threshold for red pixel values. All pixels holding a value smaller than the threshold are counted as saturated pixels.

3.13.9 awb_stat_upper_lower_g

Threshold forgreen pixel values. All pixels holding a value smaller than the threshold are counted as saturated pixels.

3.13.10 awb_stat_upper_lower_b

Threshold for blue pixel values. All pixels holding a value smaller than the threshold are counted as saturated pixels.

3.13.11 awb_enable

This parameter enables (0x01) or disables (0x00) the automatic white balance.

3.13.12 manual_wb_enable

This parameter enables (0x01) or disables (0x00) the manual white balance.

3.13.13 awb_speed

Sets the control rate of the automatic white balance from very fast (0x1) to very slow (0x64).

3.14 Auto Brightness

These registers adjust the Auto Brightness. Since this module is sensor independet, it has no access to the exposure time of the image sensor (like the sensor dependent auto exposure). So the Auto Brightness adjust the brightness of the image. For a correct white point amplify the brightness of the incoming data.

Parameter	Address[dec]	Range/Options[dec]	Default
auto_brightness_enable	44	0x00, 0x01	0
auto_brightness_max_gain	45	0...0xF	0xF
auto_brightness_lum_target	46	0...0xFFFFF	0x1F400
auto_brightness_stat_roi_col_start	48	0...0x500*	0xA
auto_brightness_stat_roi_col_end	49	0...0x500*	0x4F6
auto_brightness_stat_roi_row_start	50	0...0x780*	0x32
auto_brightness_stat_roi_row_end	51	0...0x780*	0x2BC
auto_brightness_stat_upper_bound_r	52	0...0xFFFFF	0x2D000
auto_brightness_stat_upper_bound_g	53	0...0xFFFFF	0x2D000
auto_brightness_stat_upper_bound_b	54	0...0xFFFFF	0x2D000
auto_brightness_stat_upper_lower_r	55	0...0xFFFFF	0x1
auto_brightness_stat_upper_lower_g	56	0...0xFFFFF	0x1
auto_brightness_stat_upper_lower_b	57	0...0xFFFFF	0x1
auto_brightness_stat_command	58		0x3F

*The ROI must fit into the image resolution.

3.14.1 auto_brightness_enable

Enables (0x01) and disables (0x00) auto brightness.

3.14.2 auto_brightness_lum_target

Sets the target luminance of the brightness control. The luminance is measured in the Region of Interest (ROI) and compared with the target value.

3.14.3 auto_brightness_max_gain

The maximal applied gain value.

3.14.4 auto_brightness_stat_roi_col_start

Start column of ROI.

3.14.5 auto_brightness_stat_roi_col_end

End column of ROI.

3.14.6 auto_brightness_stat_roi_row_start

Start row of ROI.

3.14.7 auto_brightness_stat_roi_row_end

End row of ROI.

3.14.8 auto_brightness_stat_upper_bound_r

Threshold for red pixel values. All pixels holding a value greater than the threshold are counted as saturated pixels.

3.14.9 auto_brightness_stat_upper_bound_g

Threshold for green pixel values. All pixels holding a value greater than the threshold are counted as saturated pixels.

3.14.10 auto_brightness_stat_upper_bound_b

Threshold for blue pixel values. All pixels holding a value greater than the threshold are counted as saturated pixels.

3.14.11 auto_brightness_stat_upper_lower_r

Threshold for red pixel values. All pixels holding a value smaller than the threshold are counted as saturated pixels.

3.14.12 auto_brightness_stat_upper_lower_g

Threshold forgreen pixel values. All pixels holding a value smaller than the threshold are counted as saturated pixels.

3.14.13 auto_brightness_stat_upper_lower_b

Threshold for blue pixel values. All pixels holding a value smaller than the threshold are counted as saturated pixels.

3.15 Test Pattern Generator and RAW Bypass

Settings for enable / disable the test pattern generator and enable / disable the RAW Bypass. Set 0x0 for video mode, 0x1 for test pattern generator, 0x2 for raw bypass.

Table 3.20: Video mode, TPG and RAW Bypass configuration and parameters

Parameter	Address[dec]	Range/Options	Default
video_mode	47	0..0x02	0

3.16 RGB Offset

Settings for RGB Offset.

Table 3.22: RGB Offset configuration and parameters

Parameter	Address[dec]	Range/Options[dec]	Default
rgb_offset_red	59	0...0xFFF	0
rgb_offset_green	60	0...0xFFF	0
rgb_offset_blue	61	0...0xFFF	0
rgb_offset_mode	62	0...0x2	0 (0 = manual, 1 = auto, 2 = disable)

3.16.1 rgb_offset_red

Sets the offset for the red channel.

3.16.2 rgb_offset_green

Sets the offset for the green channel.

3.16.3 rgb_offset_blue

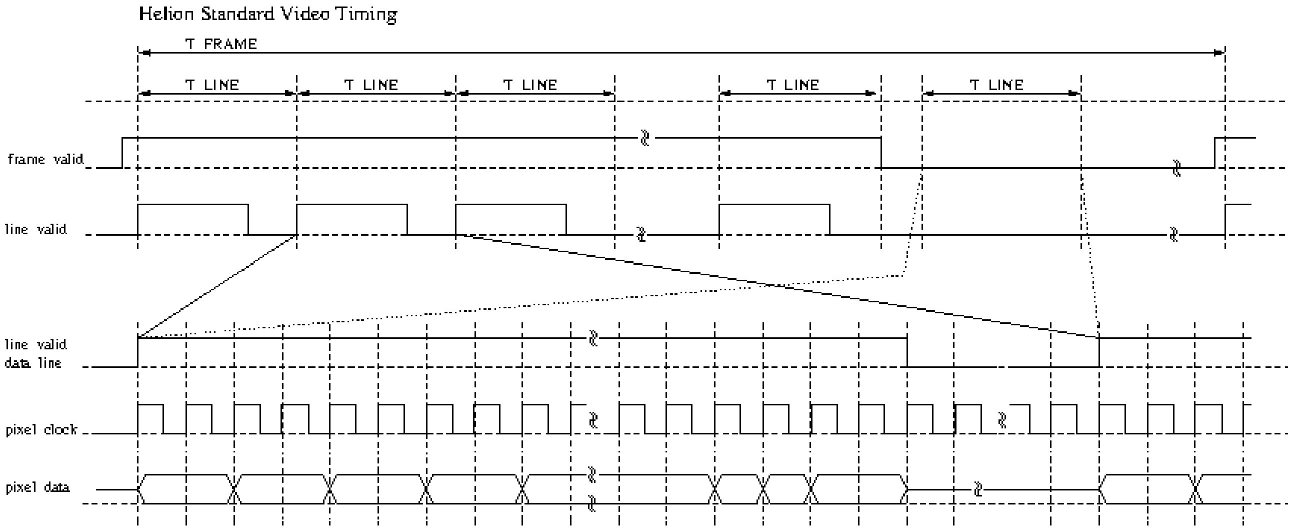
Sets the offset for the blue channel.

3.16.4 rgb_offset_mode

Switch between manual (0x0) or auto (0x1) mode. 0x2 for disabling this module.

3.17 Video Port Timing Specifications

The video input port and the video output port of the ISP module consists of the video data lines (pixel_data) and three sync signals: frame_valid and line_valid. Additionally, the pixel clock input (pixel_clock) provides the operating clock. Further a global asynchronous reset input is available to define the power-up state and initial configuration.



The signals frame_valid and line_valid are defining the active video frame. frame_valid is high during the whole active video frame, while line_valid is high during an active video line. The first pixel of a video frame is the first valid pixel after a low-high transition of frame_valid and line_valid. The first pixel of a video line is the first valid pixel after a low-high transition of line_valid. frame_valid and line_valid may change at the same time. A high-low transition of line_valid initiates a new line; a high-low transition of frame_valid initiates a new frame.

Please take care at any time to provide the same amount of valid pixels per line and the same count of lines per frame within a continuous video stream, while using non-continuous pixel-valid sequences.

Valid pixels are defined by frame_valid = line_valid = high.

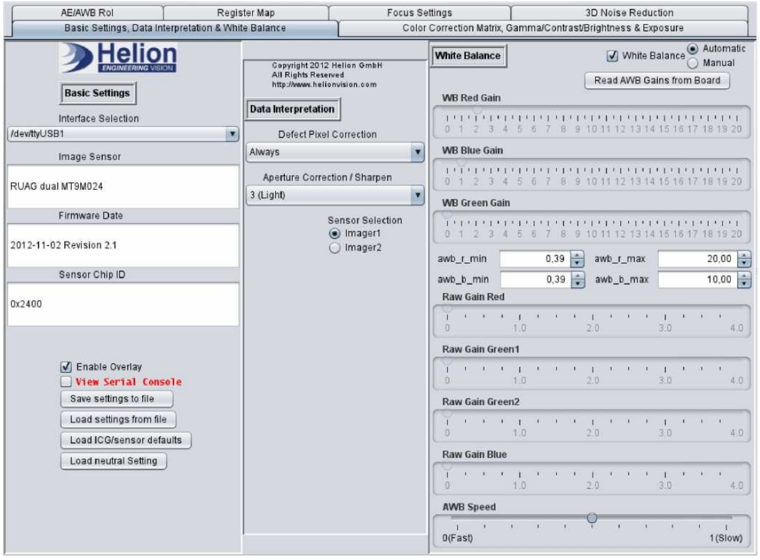
Chapter 4

Helion Standard HD ISP Usage

This chapter describes the different methods to set the ISP parameters.

4.1 IONOS Configuration GUI

During operation, the ISP parameter can be changed by the IONOS Configuration GUI. Please see the related manual for more details.



Copyright 2012 Helion GmbH
All Rights Reserved
http://www.helionvision.com

Interface Selection: /dev/ttyUSB1

Image Sensor: RUAG dual MT9M024

Firmware Date: 2012-11-02 Revision 2.1

Sensor Chip ID: 0x2400

☑ Enable Overlay
□ View Serial Console

Defect Pixel Correction: Always

Aperture Correction / Sharpen: 3 (Light)

Sensor Selection
● Imager1
○ Imager2

☑ White Balance ● Automatic ○ Manual

awb_r_min 0,39 awb_r_max 20,00
awb_b_min 0,39 awb_b_max 10,00

Picture 4.1: IONOS Configuration GUI

The IONOS Configuration GUI will work by default. If not, a small routine is needed:

1. At UART address 0xD3 write 0x1 (set_init_mode)
2. At UART address 0xD4 write 0x0 (set_write_through)
3. At UART address 0xD3 write 0x0

4.2 Direct ISP access via UART

It is possible to read and write the ISP parameters via UART. This works on the fly. To get direct ISP access, a small routine is needed:

1. At UART address 0xD3 write 0x1 (set_init_mode)
2. At UART address 0xD4 write 0x1 (set_write_through)
3. At UART address 0xD3 write 0x0

This enables the write_through mode (see Picture 2.1).

4.3 Configuration Memory

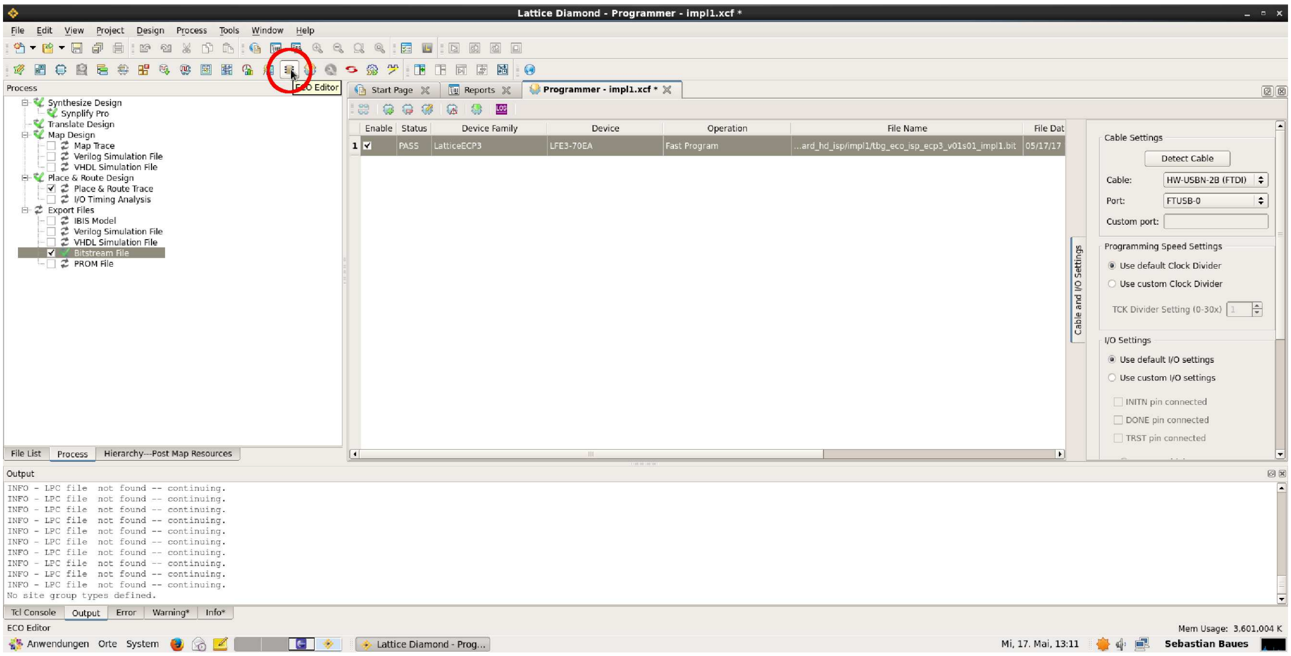
The IONOS ISP configuration is a stored in a *.mem file which includes all parameter settings. Usually, this file can be found inside the implementation folder of the project. Change the desired parameters with an editor and run a new synthesize. After programming the new bit file, the changes are visible.

4.4 Configuration Memory via ECO Editor

If the project is synthesized, a complete synthesis is not necessary. The next steps will show how to implement the configuration memory with ECO editor.

Please note: You need a synthesized project during the next steps. Will only work with Standard HD ISPV02 and higher.

Start the ECO Editor.



	Enable	Status	Device Family	Device	Operation	File Name	File Dat
1	☑	PASS	LatticeECP3	LFE3-70EA	Fast Program	...ard_hd_isp/impl1/tbg_eco_isp_ecp3_v01s01_impl1.bit	05/17/17

Cable Settings

Cable: HW-USBN-2B (FTDI)
Port: FTUSB-0
Custom port:

Programming Speed Settings

● Use default Clock Divider
○ Use custom Clock Divider
TCK Divider Setting (0-30x) 1

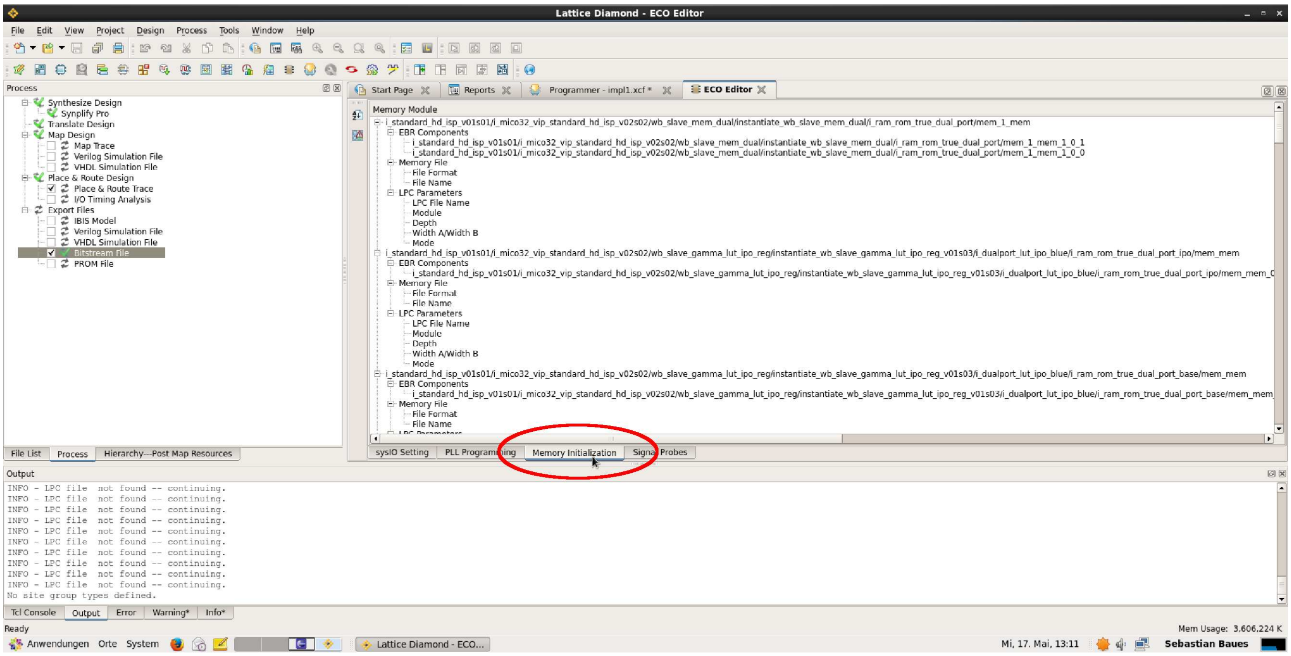
I/O Settings

● Use default I/O settings
○ Use custom I/O settings
□ INITN pin connected
□ DONE pin connected
□ TRST pin connected

```
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
No site group types defined.
```

4.4.1 Step 2

Click and Choose the Memory Initialization in the ECO Editor.

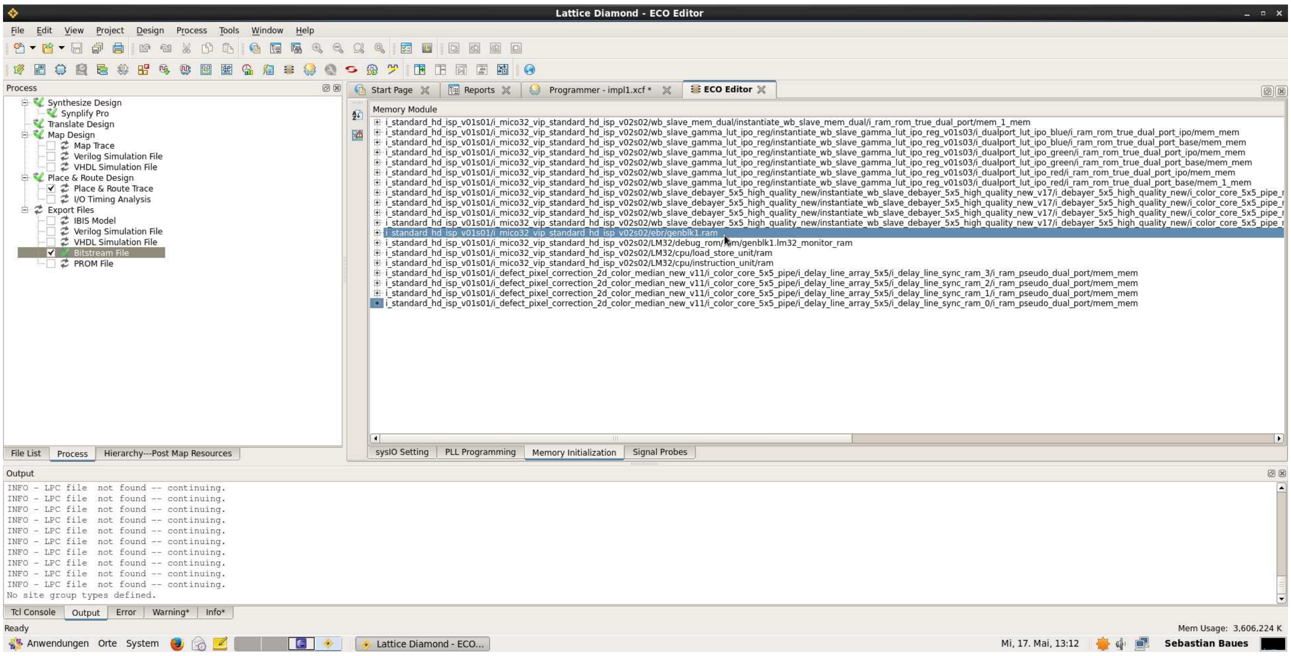


```
Memory Module
⊟ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_mem_dual/instantiate_wb_slave_mem_dual/i_ram_rom_true_dual_port/mem_1_mem
  ⊟ EBR Components
      i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_mem_dual/instantiate_wb_slave_mem_dual/i_ram_rom_true_dual_port/mem_1_mem_1_0_1
      i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_mem_dual/instantiate_wb_slave_mem_dual/i_ram_rom_true_dual_port/mem_1_mem_1_0_0
  ⊟ Memory File
      File Format
      File Name
  ⊟ LPC Parameters
      LPC File Name
      Module
      Depth
      Width A/Width B
      Mode
⊟ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_blue/i_ram_rom_true_dual_port_ipo/mem_mem
  ⊟ EBR Components
      i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_blue/i_ram_rom_true_dual_port_ipo/mem_mem_0
  ⊟ Memory File
      File Format
      File Name
  ⊟ LPC Parameters
      LPC File Name
      Module
      Depth
      Width A/Width B
      Mode
⊟ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_blue/i_ram_rom_true_dual_port_base/mem_mem
  ⊟ EBR Components
      i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_blue/i_ram_rom_true_dual_port_base/mem_mem
  ⊟ Memory File
      File Format
      File Name
```

```
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
No site group types defined.
```

4.4.2 Step 3

For a better view, close all Memory Modules. Click on every icon.
Click on ".../i_mico_vip_standard_hd_isp_v02s02/ebr/genblk1.ram".

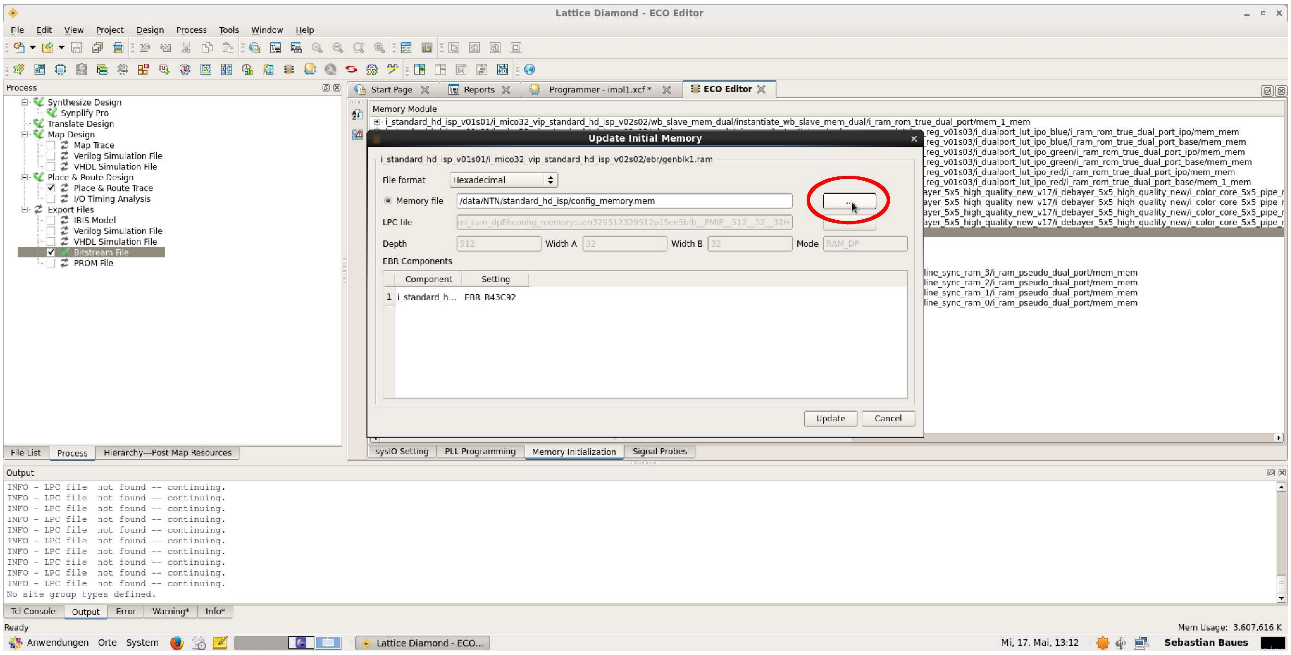


```
Memory Module
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_mem_dual/instantiate_wb_slave_mem_dual/i_ram_rom_true_dual_port/mem_1_mem
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_blue/i_ram_rom_true_dual_port_ipo/mem_mem
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_blue/i_ram_rom_true_dual_port_base/mem_mem
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_green/i_ram_rom_true_dual_port_ipo/mem_mem
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_green/i_ram_rom_true_dual_port_base/mem_mem
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_red/i_ram_rom_true_dual_port_ipo/mem_mem
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_gamma_lut_ipo_reg/instantiate_wb_slave_gamma_lut_ipo_reg_v01s03/i_dualport_lut_ipo_red/i_ram_rom_true_dual_port_base/mem_1_mem
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_debayer_5x5_high_quality_new/instantiate_wb_slave_debayer_5x5_high_quality_new_v17/i_debayer_5x5_high_quality_new/i_color_core_5x5_pipe_r
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_debayer_5x5_high_quality_new/instantiate_wb_slave_debayer_5x5_high_quality_new_v17/i_debayer_5x5_high_quality_new/i_color_core_5x5_pipe_r
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_debayer_5x5_high_quality_new/instantiate_wb_slave_debayer_5x5_high_quality_new_v17/i_debayer_5x5_high_quality_new/i_color_core_5x5_pipe_r
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_debayer_5x5_high_quality_new/instantiate_wb_slave_debayer_5x5_high_quality_new_v17/i_debayer_5x5_high_quality_new/i_color_core_5x5_pipe_r
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/ebr/genblk1.ram
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/LM32/debug_rom/ram/genblk1.lm32_monitor_ram
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/LM32/cpu/load_store_unit/ram
⊞ i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/LM32/cpu/instruction_unit/ram
⊞ i_standard_hd_isp_v01s01/i_defect_pixel_correction_2d_color_median_new_v11/i_color_core_5x5_pipe/i_delay_line_array_5x5/i_delay_line_sync_ram_3/i_ram_pseudo_dual_port/mem_mem
⊞ i_standard_hd_isp_v01s01/i_defect_pixel_correction_2d_color_median_new_v11/i_color_core_5x5_pipe/i_delay_line_array_5x5/i_delay_line_sync_ram_2/i_ram_pseudo_dual_port/mem_mem
⊞ i_standard_hd_isp_v01s01/i_defect_pixel_correction_2d_color_median_new_v11/i_color_core_5x5_pipe/i_delay_line_array_5x5/i_delay_line_sync_ram_1/i_ram_pseudo_dual_port/mem_mem
⊞ i_standard_hd_isp_v01s01/i_defect_pixel_correction_2d_color_median_new_v11/i_color_core_5x5_pipe/i_delay_line_array_5x5/i_delay_line_sync_ram_0/i_ram_pseudo_dual_port/mem_mem
```

```
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
No site group types defined.
```

4.4.3 Step 4

Load the configuration memory file in the momory file directionary by clicking "...".



Lattice Diamond - ECO Editor

Memory Module

i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_mem_dual/instantiate_wb_slave_mem_dual/i_ram_rom_true_dual_port/mem_1_mem

Update Initial Memory

i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/ebr/genblk1.ram

File format: Hexadecimal

● Memory file: /data/NTN/standard_hd_isp/config_memory.mem

LPC file: mi_ram_dpEhconfig_memoryssen329512329512p15ce5bfb__PMIP__512__32__32H

Depth: 512 Width A: 32 Width B: 32 Mode: RAM_DP

EBR Components

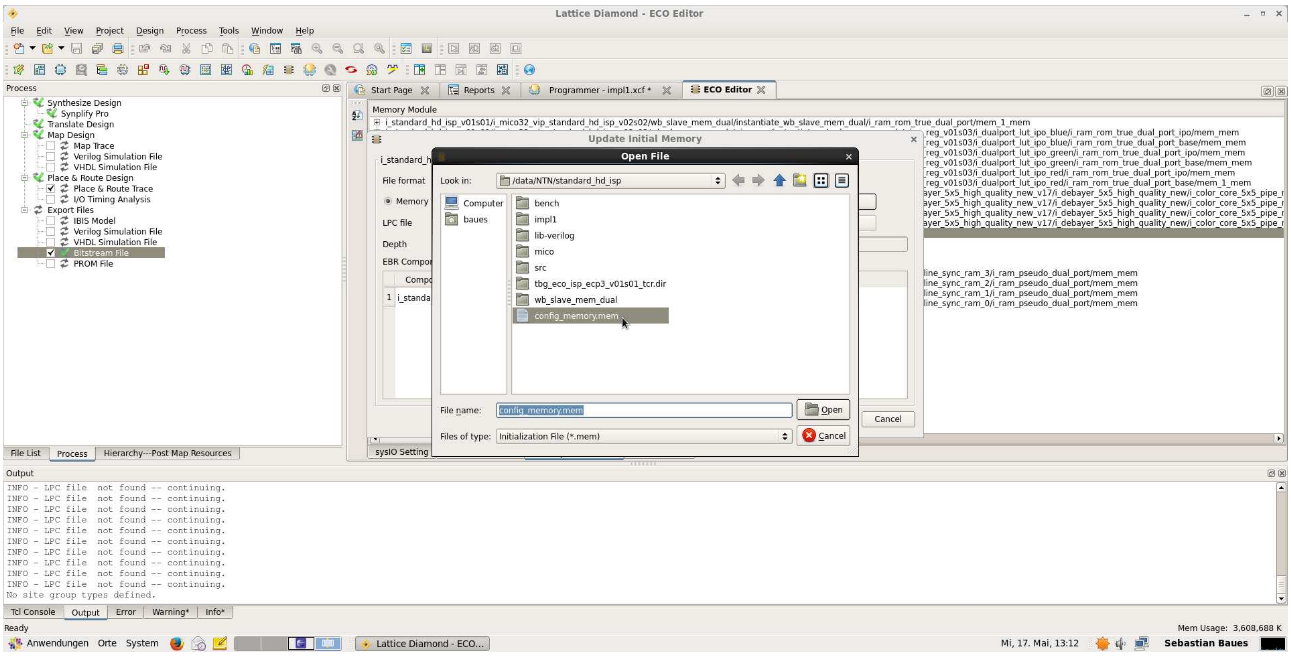
	Component	Setting
1	i_standard_h...	EBR_R43C92

```
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
No site group types defined.
```

Ready Mem Usage: 3,607,616 K

4.4.4 Step 5

... and choose the configuration memory file ...



Lattice Diamond - ECO Editor

Open File

Look in: /data/NTN/standard_hd_isp

Computer
baues

bench
impl1
lib-verilog
mico
src
tbg_eco_isp_ecp3_v01s01_tcr.dir
wb_slave_mem_dual
config_memory.mem

File name: config_memory.mem

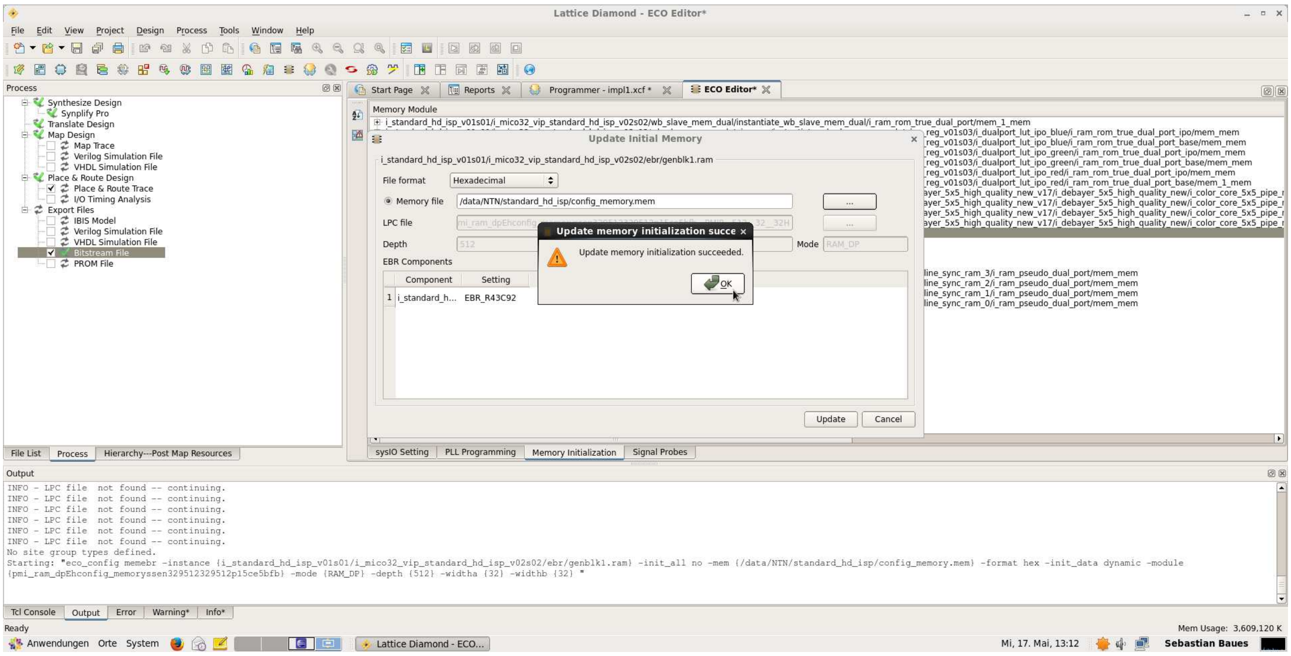
Files of type: Initialization File (*.mem)

```
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
No site group types defined.
```

Ready Mem Usage: 3,608,688 K

4.4.5 Step 6

... and click on ok.



Memory Module

i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/wb_slave_mem_dual/instantiate_wb_slave_mem_dual/i_ram_rom_true_dual_port/mem_1_mem

Update Initial Memory

i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/ebr/genblk1.ram

File format: Hexadecimal

● Memory file: /data/NTN/standard_hd_isp/config_memory.mem

LPC file

Depth: 512

Mode: RAM_DP

EBR Components

	Component	Setting
1	i_standard_h...	EBR_R43C92

Update | Cancel

Update memory initialization succe

Update memory initialization succeeded.

OK

```
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
INFO - LPC file  not found -- continuing.
No site group types defined.
Starting: "eco_config memebr -instance {i_standard_hd_isp_v01s01/i_mico32_vip_standard_hd_isp_v02s02/ebr/genblk1.ram} -init_all no -mem {/data/NTN/standard_hd_isp/config_memory.mem} -format hex -init_data dynamic -module
{pmi_ram_dpEhconfig_memoryssen329512329512p15ce5bfb} -mode {RAM_DP} -depth {512} -widtha {32} -widthb {32} "
```

And save the changes.

4.4.6 Step 7

A new bit file can be created through a double click on "Bitstream File".

Chapter 5

Resource Utilization and Performance

This chapter gives resource utilization information for Lattice FPGAs using the Standard HD ISP. Table 5.1 lists the parameter settings that were used to derive the performance[1] and utilization data shown in Table 5.2.[2]

Table 5.1: Resource Utilization

Function	**Standard HD ISP**	
Supported FPGA Families	Lattice ECP3 / ECP5	
Performance	Data Path Width	12 bit
	Fmax	150 MHz
	Target Application	720p60 / 1080p60
Resource Utilization	LUTs	~14000
	sysMEM EBRs	~67
	Registers	~14000
	Mult-18	~64
Design Tool Support	Lattice Implementation	Diamond 3.9.1.119
	Sythesis	Synopsys Synplify Pro for Lattice L-2016.09L-1
	Simulation	TBD

Table 5.2: **Performance**

Standard HD ISP				
Version	**Bit Width**	**Frame Size**	**Frame Rate**	**max. Pixel Rate**
v03	12 bit	720p / 1080p*	60 fps	150 MHz

*can be switched by changing the related configuration parameter (see chapter 3.8 for more details).

[1] Fmax determined with speedgrade 8

[2] Performance and utilization data are generated to target at a specific FPGA type using Lattice Diamond 3.9 and Synplify Pro H-2016.09L software. Performance may vary when using a different software version or targeting a different device density or speed grade within the Lattice family.

Chapter 6

Ordering Part Number

The IP cores can be obtained from Lattice under license for implementation and evaluation (run time limited). For obtaining evaluation and production licenses, please contact your local Lattice Sales Representative.

Table 6.1: **ISP Ordering Information**

Standard HD ISP				
Version	**Shortcut**	**Bit Width**	**Target FPGA**	**Ordering Code**
v03	ISP-STHD-12	12 bit	ECP3 / ECP5 Series	BE10029117

Table 6.2: **Reference Design Ordering Information**

Reference Design for Embedded Vision Development Kit				
Version	**Shortcut**	**Bit Width**	**Target FPGA**	**Ordering Code**
v03	REF-EMVI-12	12 bit	ECP3 / ECP5 Series	BE10029632

Chapter 7

Revision History

Table 7.1: Revision History Standard HD ISP (ISP-STHD-12)

Date	Version	Section	Change Summary
2017-05-18	v01s01	-	Initial Release
2017-07-21	v02s01	-	Improvements
2017-07-27	v03s01	-	Improvements