LogiCORE IP Image Statistics v2.0

DS752 March 1, 2011 **Product Specification**

Introduction

The Xilinx Image Statistics LogiCORE™ IP implements the computationally intensive metering functionality common in digital cameras, camcorders and imaging devices. This core generates a set of statistics for color histograms, mean and variance values, edge and frequency content for 16 user-defined zones on a per frame basis. The statistical information collected may be used in the control algorithms for Auto-Focus, Auto-White Balance, and Auto-Exposure for image processing applications.

Features

- High-definition (1080p60) resolutions
- Up to 4096 total pixels and 4096 total rows
- Selectable processor interface: EDK pCore or General Purpose Processor
- 16 programmable zones
- 8, 10, or 12-bit input precision
- Outputs for all zones and color channels:
    - Minimum and maximum color values
    - Sum and sum of squares for each color value
    - Low and high frequency content
    - Horizontal, vertical and diagonal edge content
- Outputs for pre-selected zone(s):
    - Y channel histogram
    - R,G,B channel histograms
    - Two-dimensional Cr-Cb histogram

Applications

- Automatic Exposure (AE) control
- Automatic Sensor Gain (AG) control
- Auto Focus (AF) control of the lens assembly
- Digital contrast/brightness adjustment
- Global histogram equalization
- White Balance correction

LogiCORE IP Facts Table					
Core Specifics					
Supported Device Family[1]	Virtex®-6, Virtex-5, Spartan®-6, Spartan-3A DSP				
Supported User Interfaces	General Purpose Processor Interface, PLB pCore Interface				
	Resources[2]				**Frequency**
Configuration	**LUTs**	**FFs**	**DSP Slices**	**BRAMs[3]**	**Max. Freq.[4]**
Data Width=8	1736	2496	14	2(18)+1(36)	243.49
Data Width=10	1931	2777	14	1(18)+2(36)	249.81
Data Width=12	2143	3057	14	2(18)+2(36)	231.11
Provided with Core					
Documentation	Product Specification				
Design Files	Netlists, EDK pCore files, C drivers				
Example Design	Not Provided				
Test Bench	Provided on the product page				
Constraints File	Not Provided				
Simulation Model	VHDL or Verilog Structural model; C and MATLAB™ models provided on the product page				
Tested Design Tools					
Design Entry Tools	CORE Generator™, ISE® 13.1, Platform Studio (XPS)				
Simulation	ModelSim v6.6d, QuestaSim v6.6c, ISIM 13.1				
Synthesis Tools	XST 13.1				
Support					
Provided by Xilinx, Inc.					

1. For a complete listing of supported devices, see the release notes for this core.
2. Resources listed here are for Virtex-6 devices, selecting the General Purpose Processor interface, and setting the Maximum Number of Columns and Rows to 2200 in the CORE Generator GUI. For more complete device performance numbers, see Table 11.
3. Indicating the number of RAMB18E1 and RAMB36E1 primitives used.
4. Performance numbers listed are for Virtex-6 FPGAs. For more complete performance data, see Core Resource Utilization and Performance.

Overview

The statistics module supports multi-zone metering on rectangular regions. The Image Statistics core supports 16 software defined zones (Figure 1).

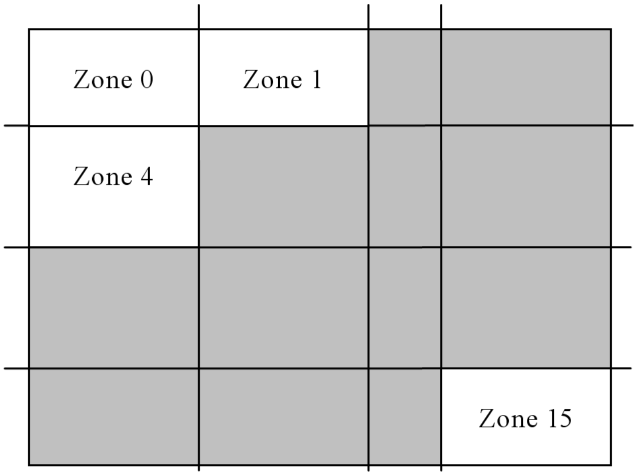


Figure 1: **16 Zone Metering**

Minimum, maximum, sum, and sum of squares values are calculated for all color channels for all 16 zones. Frequency and edge content is also calculated for all zones in luminance values provided by an RGB-to-YCrCb converter internal to the Image Statistics core. The RGB, Y, and two-dimensional Cr-Cb histograms are calculated over predefined sets of zones.

Minimum and Maximum Values

The minimum and maximum values can be useful for histogram stretching, Auto-Gain, Digital-Gain, Auto-Exposure, or simple White-Balance applications. These values are calculated for all zones and all R,G,B color channels simultaneously.

Sum of Color Values

The core provides the sum of color values for all zones (sum). The mean values for color channels can be calculated by dividing the sum value by the size of the zone (*N*):

$$\bar{x} = \frac{1}{N}\sum_{i=0}^{N-1} x_i = \frac{sum}{N}$$

Equation 1

Sum of Squares of Color Values

The core provides the power output equal to the sum of squared values for all color channels and zones (*pow*), from which the signal power or the variance can be calculated:

$$\sigma^2 = \frac{1}{N}\left[\sum_{i=0}^{N-1} x_i^2\right] - \bar{x}^2 = \frac{pow}{N} - \bar{x}^2$$

Equation 2

Frequency Content

The frequency content for each zone is calculated using the luminance channel. To calculate low-frequency content, luminance values are first low-pass filtered with a 7 tap FIR filter, with fixed coefficients [-1 0 9 16 9 0 -1]/32.

The Low Frequency power output (*LoFreq*) of the core provides the cumulative sum of the squared values of the FIR filter output for each zone:

$$LoFreq = \sum_{i=0}^{N-1} \left[\frac{FIR\left(x_i, \{-1,0,9,16,9,0,-1\}\right)}{32} \right]^2$$

Equation 3

In Equation 3, square brackets [] represent clipping at $max(x_i)=2^{DATA\_WIDTH-1}$ and clamping values at 0.

The high frequency power output (*HiFreq*) of the core provides the difference between the power of the original luminance values and the power of the low-pass filtered signal within each of these zones:

$$HiFreq = \lfloor pow - LoFreq \rfloor$$

Equation 4

In Equation 4, square brackets represent clamping values at 0.

Edge Content

The Image Statistics core filters the luminance values calculated for all zones using the Sobel operators:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

Figure 2: **Horizontal, Vertical and Diagonal (Left and Right) Sobel Operators**

The Sobel operators are implemented without multipliers to reduce size and increase performance. The edge content outputs (*Hsobel*, *Vsobel*, *Lsobel*, *Rsobel*) provide the cumulative sums of absolute values of filtered luminance values:

$$Lsobel = \sum_{i=0}^{N-1} ABS\left(\frac{1}{32} FIR2D\left(x_i, \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \right) \right)$$

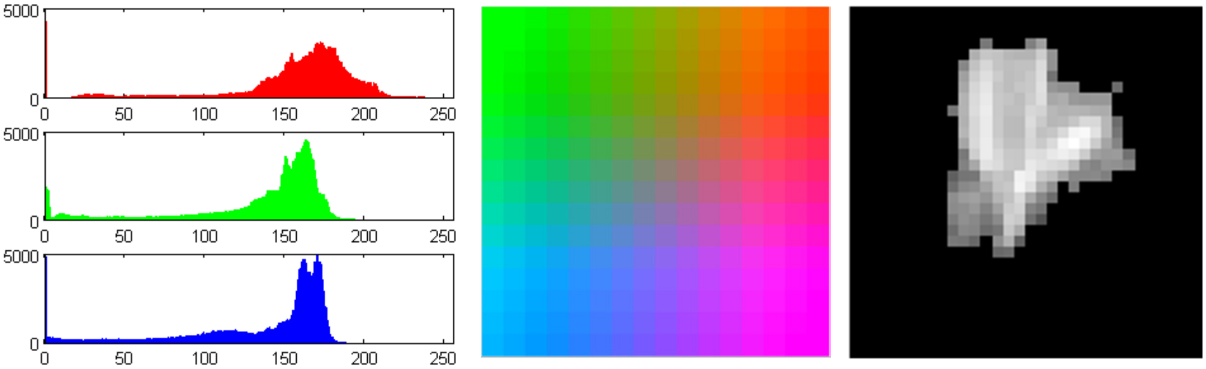
Equation 5

Equation 5 describes the calculation of the upper-left to lower-right diagonal frequency content, *Lsobel*. Output values for *Vsobel*, *Lsobel*, and *Rsobel* are calculated similarly by using the corresponding coefficient matrixes from Figure 2.

Histogram Data

For zones selected by a dynamically programmable register (`rgb_hist_zone_en`), the Image Statistics core bins R,G,B data and creates histograms as shown in Figure 3a. Similarly for zones selected by register `ycc_hist_zone_en`, Y and two-dimensional Cr-Cb histograms are calculated as shown in Figure 3c.

The two-dimensional Cr-Cb histogram (Figure 3c) contains information about the color content of a frame. Different hues have distinct locations in the Cr-Cb color-space (Figure 3b). The center location and variance of the color gamut can be derived from its two-dimensional Cr-Cb histogram. The bounding shape of the color gamut, along with the center location and variance of the two-dimensional Cr-Cb histogram, can be used to drive higher level algorithms [Ref 4] for white-balance correction.



a. Example RGB Histograms

b. Colors in Cr-Cb Space

c. Example Color Gamut in Cr-Cb Space

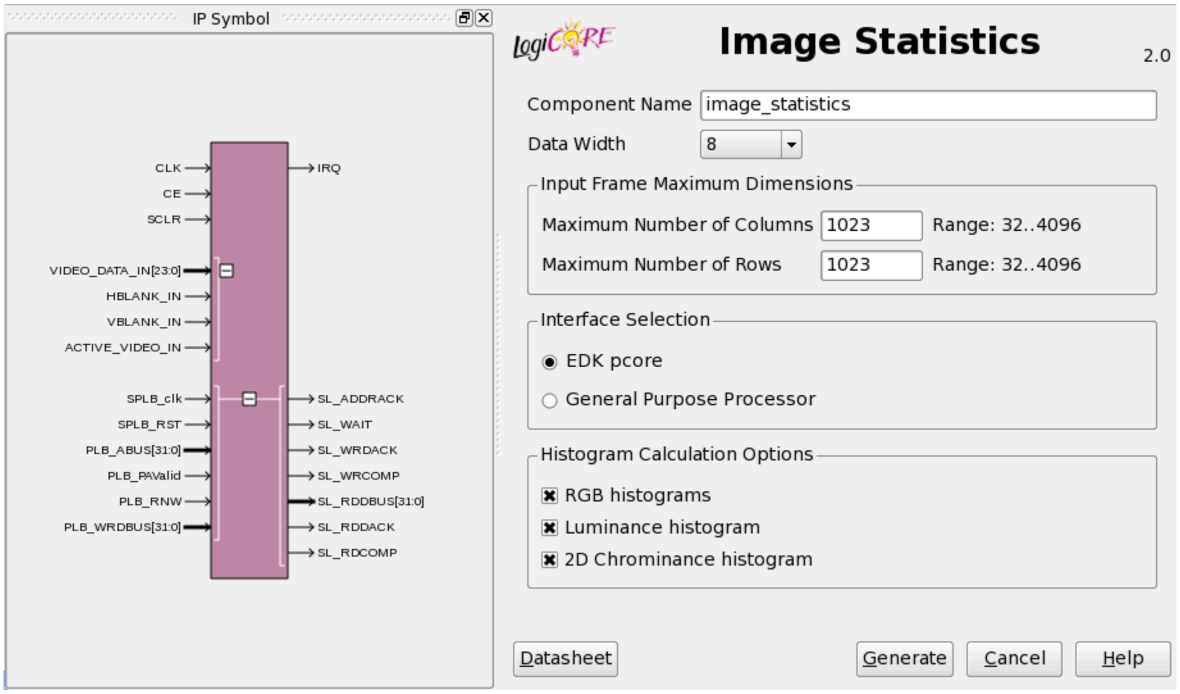
Figure 3: **Histogram Data**

For further details on histogram calculations, refer to Setting Up Histogram Calculations

The resolution of the R,G,B and Y histograms is the same as the resolution of the input data. The two-dimensional Cr-Cb histogram contains the same number of bins, but due to the two-dimensional configuration, the resolution is $2^{DATA\_WIDTH/2}$ along the Cr-Cb axes.

CORE Generator – Graphical User Interface

The Image Statistics core is easily configured to meet user-specific needs through the CORE Generator graphical user interface (GUI). This section provides a quick reference to the parameters that can be configured at generation time. Figure 4 shows the main Image Statistics screen.



Component Name: image_statistics

Data Width: 8

Input Frame Maximum Dimensions

Maximum Number of Columns: 1023 Range: 32..4096

Maximum Number of Rows: 1023 Range: 32..4096

Interface Selection

● EDK pcore

○ General Purpose Processor

Histogram Calculation Options

☒ RGB histograms

☒ Luminance histogram

☒ 2D Chrominance histogram

Figure 4: **Image Statistics CORE Generator GUI**

The GUI displays a representation of the IP symbol on the left side, and the parameter assignments on the right side, described as follows:

- **Component Name:** The component name is used as the base name of output files generated for the module. Names must begin with a letter and must be composed from characters a to z, 0 to 9 and "_".
- **Data Width** (*DATA_WIDTH*)**:** Specifies the bit width of input data. Permitted values are 8, 10, or 12.
- **Maximum Number of Columns** (*MAX_COLS*)**:** Specifies the number of total columns in a frame defined by the input timing signals. This value is used to determine the depth of line-buffers and the width of certain control paths in the core.
- **Maximum Number of Rows** (*MAX_ROWS*)**:** Specifies the number of total rows in a frame defined by the input timing signals. This value is used to determine the width of certain control paths in the core.
- **Interface Selection:** This option allows for the configuration of two different interfaces for the core.
    - **EDK pCore Interface:** CORE Generator software generates a pCore that can easily be imported into an EDK project as a hardware peripheral. Configuration parameters and statistical data can be accessed via registers. See the EDK pCore Interface section of Core Symbol and Port Descriptions.

- **General Purpose Processor Interface:** CORE Generator software generates a set of ports to be used to program the core and collect results. See the General Purpose Processor Interface section of Core Symbol and Port Descriptions.

- **Histogram Calculation Options:** Storing and calculating histograms utilize block RAM resources in the FPGA. By specifying which histograms calculations are needed, this option can be used to reduce FPGA resources required for the generated core instance.
  - **RGB Histograms:** The check box enables/disables instantiation of the R,G and B histogram calculating modules for zones pre-selected for RGB histogramming.
  - **Luminance Histogram:** The check box enables/disables instantiation of the luminance histogram calculating module for zones pre-selected for Y and CrCb histogramming.
  - **2D Chrominance Histogram:** The check box enables/disables instantiation of the two-dimensional chrominance (Cr-Cb) histogram calculating module for zones pre-selected for Y and CrCb histogramming

Core Symbol and Port Descriptions

Processor Interfaces

Processor interfaces provide the system designer with the ability to dynamically control the parameters within the core. The Image Statistics core supports two processor interface options:

- EDK pCore Interface
- General Purpose Processor Interface

The Xilinx Streaming Video Interface is a set of signals common to both interface options and to all video Image Processing (iPipe) cores. It is described in Table 1 along with more detailed descriptions of the ports following the table.

Table 1: **Port Descriptions for the Xilinx Streaming Video Interface**

Port Name	Port Width	Direction	Description
video_data_in	3*DATA_WIDTH	input	Data input bus
hblank_in	1	input	Horizontal blanking input
vblank_in	1	input	Vertical blanking input
active_video_in	1	input	Active video signal input

- **video_data_in:** This bus contains the video input in `DATA_WIDTH` bits wide unsigned integer representation.

Bits	3DATA_WIDTH-1:2DATA_WIDTH	2DATA_WIDTH-1:DATA_WIDTH	DATA_WIDTH-1:0
Video Data Signals	R	B	G

- **hblank_in:** The `hblank_in` signal conveys information about the blank/non-blank regions of video scan lines.
- **vblank_in:** The `vblank_in` signal conveys information about the blank/non-blank regions of video frames, and is used by the Image Statistics core to detect the end of a frame, when user registers can be copied to active registers to avoid visual tearing of the image.
- **active_video_in:** The `active_video_in` signal is high when valid data is presented at the input.

Xilinx Streaming Video Interface

The Xilinx Streaming Video Interface (XSVI) is a set of signals that is used to stream video data between video IP cores. XSVI is also defined as an Embedded Development Kit (EDK) bus type so that the tool can automatically create input and output connections to the core. This definition is embedded in the pCore interface provided with the IP, and it allows an easy way to cascade connections of Xilinx Video Cores. The Image Statistics IP core uses the following subset of the XSVI signals:

- `video_data`
- `vblank`
- `hblank`
- `active_video`

Other XSVI signals on the XSVI input bus, such as `video_clk`, `vsync`, `hsync`, `field_id`, and `active_chr` do not affect the function of this core.

Note: These signals are neither propagated, nor driven on the XSVI output of this core.

The following is an example EDK Microprocessor Peripheral Definition (.MPD) file definition.

Input Side:

```
BUS_INTERFACE BUS = XSVI_STATISTICS,  BUS_TYPE = TARGET, BUS_STD = XSVI

PORT hblank_i = hblank,            DIR=I,  BUS=XSVI_STATISTICS

PORT vblank_i = vblank,            DIR=I,  BUS=XSVI_STATISTICS

PORT active_video_i = active_video,DIR=I,  BUS=XSVI_STATISTICS

PORT video_data_i=video_data,DIR=I,VEC=[C_DATA_WIDTH1:0],BUS=XSVI_STATISTICS
```

The Image Statistics IP core is fully synchronous to the core clock, `clk`. Consequently, the input XSVI bus is expected to be synchronous to the input clock, `clk`. The `video_clk` signal of the input is not used.

EDK pCore Interface

Many imaging applications utilize an embedded processor to dynamically control parameters within IP cores. The EDK pCore Interface generates Processor Local Bus (PLB4.6) interface ports in addition to the Xilinx Streaming Video Interface, `clk`, `ce`, and `sclr` signals. For more information on the PLB4.6 signals, see the Processor Local Bus (PLB) v4.6 [Ref 1]. The PLB bus signals are automatically connected when the generated pCore is inserted into an EDK project. The Core Symbol for the EDK pCore Interface is shown in Figure 5.

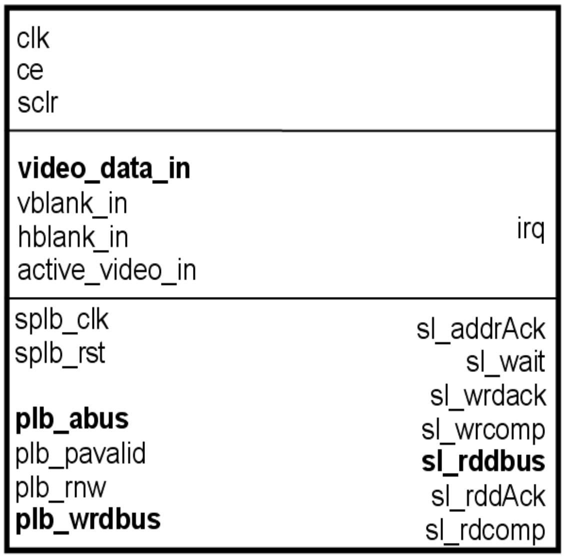


Figure 5: **Core Symbol**

Generating the Image Statistics core with an EDK pCore interface provides a memory-mapped interface for the programmable registers within the core, described in Table 2.

All of the registers are readable, enabling verification of written values or read back of current values.

Table 2: **EDK pCore Interface Register Description**

Address Offset BASEADDR+	Register Name	Access Type	Default Value	Description
0x000	stats_reg_00_control	R/W	1	Control register (Table 3) Bit 0: SW_ENABLE Bit 1: REG_UPDATE Bit 2: READOUT Bit 3: CLR_STATUS
0x004	stats_reg_01_sw_reset	R/W	0	Bit 1: SW_RESET (1: reset, 0: not reset)
0x008	stats_reg_02_status	R	0	General status register (Table 4) Bit 0: VSYNC Bit 1: DONE Frame acquisition complete Bit 2: VBLANK_ERROR Bit 3: HBLANK_ERROR
0x00c	stats_reg_03_irq_control	R/W	258	Bit 0: VSYNC Interrupt enable Bit 1: DONE Interrupt enable Bit 2: VBLANK_ERROR Interrupt enable Bit 3: HBLANK_ERROR Interrupt enable Bit 8: General Interrupt Enable
0x010	stats_reg_04_hmax0	R/W	¼ MAX_COLS	Position of the first vertical zone delimiter
0x014	stats_reg_05_hmax1	R/W	½ MAX_COLS	Position of the second vertical zone delimiter
0x018	stats_reg_06_hmax2	R/W	¾ MAX_COLS	Position of the third vertical zone delimiter
0x01c	stats_reg_07_vmax0	R/W	¼ MAX_ROWS	Position of the first horizontal zone delimiter
0x020	stats_reg_08_vmax1	R/W	½ MAX_ROWS	Position of the second horizontal zone delimiter
0x024	stats_reg_09_vmax2	R/W	¾ MAX_ROWS	Position of the third horizontal zone delimiter
0x028	stats_reg_10_hist_zoom_factor	R/W	0	Bit 0-1 control CrCb histogram zooming: 00: No zoom, full Cb and Cr range 01: Zoom by 2 10: Zoom by 4 11: Zoom by 8
0x02c	stats_reg_11_rgb_hist_ zone_en	R/W	65535	Bits 0-15 correspond to zones 0-15, enabling RGB histogramming for the selected zones
0x030	stats_reg_12_ycc_hist_ zone_en	R/W	65535	Bits 0-15 correspond to zones 0-15, enabling Y and CrCb histogramming for the selected zones
0x034	stats_reg_13_zone_addr	R/W	0	Bits 0-3 select a zone for readout
0x038	stats_reg_14_color_addr	R/W	0	Bits 0-1 select a color channel for readout 00: Red 01: Green 1X: Blue

Table 2: **EDK pCore Interface Register Description *(Cont'd)***

Address Offset BASEADDR+	Register Name	Access Type	Default Value	Description
0x03c	stats_reg_15_hist_addr	R/W	0	Bits 0-[DATA_WIDTH-1] address histograms
0x040	stats_reg_16_addr_valid	R/W	0	Bit 0 qualifies zone_addr, color_addr and hist_addr
0x044	stats_reg_17_data_valid	R	0	Bit 0 qualifies valid data on core outputs corresponding to addr inputs
0x050	stats_reg_20_max	R	0	Maximum value measured for the currently selected zone and color channel
0x054	stats_reg_21_min	R	0	Minimum value measured for the currently selected zone and color channel
0x058	stats_reg_22_sum_lo	R	0	Higher and Lower 32 bits of the sum of values for the currently selected zone and color channel
0x005c	stats_reg_23_sum_hi	R	0	
0x060	stats_reg_24_pow_lo	R	0	Higher and Lower 32 bits of the sum of squared values for the currently selected zone and color channel
0x064	stats_reg_25_pow_hi	R	0	
0x068	stats_reg_26_hsobel_lo	R	0	Higher and lower 32 bits of the sum of absolute values of the Horizontal Sobel filter output, applied to luminance values of the currently selected zone
0x06c	stats_reg_27_hsobel_hi	R		
0x070	stats_reg_28_vsobel_lo	R	0	Higher and lower 32 bits of the sum of absolute values of the Vertical Sobel filter output, applied to luminance values of the currently selected zone
0x074	stats_reg_29_vsobel_hi	R	0	
0x078	stats_reg_30_lsobel_lo	R	0	Higher and lower 32 bits of the sum of absolute values of the Diagonal Sobel filter output, applied to luminance values of the currently selected zone
0x007c	stats_reg_31_lsobel_hi	R	0	
0x0080	stats_reg_32_rsobel_lo	R	0	Higher and lower 32 bits of the sum of absolute values of the anti-diagonal Sobel filter output, applied to luminance values of the currently selected zone
0x084	stats_reg_33_rsobel_hi	R	0	
0x088	stats_reg_34_hifreq_lo	R	0	Higher and lower 32 bits of the sum of absolute values of the High Frequency filter output, applied to luminance values of the currently selected zone
0x08c	stats_reg_35_hifreq_hi	R	0	
0x090	stats_reg_36_lofreq_lo	R	0	Higher and lower 32 bits of the sum of absolute values of the Low Frequency filter output, applied to luminance values of the currently selected zone
0x094	stats_reg_37_lofreq_hi	R	0	
0x098	stats_reg_38_rhist	R	0	Red histogram values calculated over the zones selected by rgb_hist_zone_en
0x09c	stats_reg_39_ghist	R	0	Green histogram values calculated over the zones selected by rgb_hist_zone_en
0x0a0	stats_reg_40_bhist	R	0	Blue histogram values calculated over the zones selected by rgb_hist_zone_en

Table 2: **EDK pCore Interface Register Description *(Cont'd)***

Address Offset BASEADDR+	Register Name	Access Type	Default Value	Description
0x0a4	stats_reg_41_yhist	R	0	Luminance histogram values calculated over the zones selected by ycc_hist_zone_en
0x0a8	stats_reg_42_cchist	R	0	Two-dimensional Cr-Cb chrominance histogram values calculated over the zones selected by ycc_hist_zone_en

The core can be effectively reset in-system by asserting `stats_reg01_reset` (bit 0), which returns all register values to their default values. Core outputs are forced to 0 instantaneously until the software reset bit is deasserted. However, block RAMs internal to the core are not initialized until `stats_reg01_sw_reset` is deasserted, and the core becomes ready for the next data-acquisition cycle. For more information on initialization, see Processing States

Additional information about programming user registers is provided in the API documentation available in XPS and located in the generated pCore directory under `doc/html/api/index.html` in the EDK pCore Interface section of Processor Interfaces

Control Register (stats_reg00_control)

The Software Enable bit of register `stats_reg00_control` allows the core to be dynamically enabled or disabled. Disabling the core reduces power consumption when statistical data collection is not needed. The default value of Software Enable is 1 (enabled). See Table 3.

Table 3: **Control Register**

Position	Name of Flag	Corresponding Event
Bit 0	SW_ENABLE	0: indicates the Image Statistics core is disabled 1: indicates the Image Statistics core is enabled
Bit 1	REG_UPDATE	Semaphore for PLB register update 0: indicates the Host processor is updating registers 1: indicates the Host processor is done updating registers See the Synchronization section for further information.
Bit 2	READOUT	0: directs the Image Statistics core to bypass readout mode. When in readout mode, writing 0 to this flag directs the Image Statistics core to exit readout mode. 1: directs the Image Statistics core to enter readout mode. See the Synchronization section for further information.
Bit 3	CLR_STATUS	Resets values of the status register (stats_reg50_status) to 0, thereby clearing any interrupt requests (irq pin) as well

Bits 1 (REG_UPDATE) and 2 (READOUT) of `stats_reg00_control` provide a frame synchronization mechanism between the EDK processor and the Image Statistics core. For more information on the use of this register, see Synchronization Bit 3 (CLR_STATUS) of `stats_reg00_control` provides a mechanism to clear the `Status` register (`stats_reg50_status`).

Status Register (stats_reg50_status)

The `status` register contains information about events, such as past timing errors, that the host processor must clear out to be able to detect new or recurring events. See Table 4.

Table 4: **Status Register**

Position	Name of Flag	Corresponding Event
Bit 0	VSYNC	Falling edge on vblank_in detected
Bit 1	DONE	Frame Data acquisition complete
Bit 2	VBLANK_ERROR	Measured number of total rows per frame is larger than MAX_ROWS parameter
Bit 3	HBLANK_ERROR	Measured number of total columns per frame is larger than MAX_COLS parameter
Bit 4	INIT_DONE	Timing parameters stabilized (goes high after the second frame is completed)
Bits 23-16	VERSION	Core Version number in 5 + 3 bits format. Default value 10h corresponding to version 2.0
Bits 31-29	HISTOGRAM_CONF	These 3 bits indicate whether the core was instantiated with RGB, CC, and Y histograms, respectively, enabled in CORE Generator.

Contents of the `status` register clears with `SCLR` or by asserting `CLR_STATUS` (bit 2 of the `stats_reg00_control` register).

IRQ Control Register (stats_reg02_irq_control)

Once the user application/interrupt handler routine is done servicing the Image Statistics core, the flag that triggered the interrupt should be cleared from software using the `CLR_STATUS` bit of `stats_reg50_status`, which in turn deasserts the `irq` output pin. See Table 5.

Table 5: **Interrupt Control Register**

Position	Name of Flag	Description
Bit 0	VSYNC_IRQ_EN	Falling edge on vblank_in (VSYNC) event interrupt enable
Bit 1	DONE_IRQ_EN	Frame Data acquisition complete (DONE) event interrupt enable
Bit 2	VBLANK_IRQ_EN	VBLANK_ERROR event interrupt enable
Bit 3	HBLANK_IRQ_EN	HBLANK_ERROR event interrupt enable
Bit 8	IRQ_EN	General Interrupt Enable

General Purpose Processor Interface

The General Purpose Processor Interface exposes statistical data outputs and all control registers as ports. This option can be used in a system with a user-defined bus interface (decoding logic and register banks) to an arbitrary processor.

The Core Symbol for the General Purpose Processor Interface is shown in Figure 6. The Xilinx Streaming Video Interface is described in Table 1, and additional ports are described in Table 6.

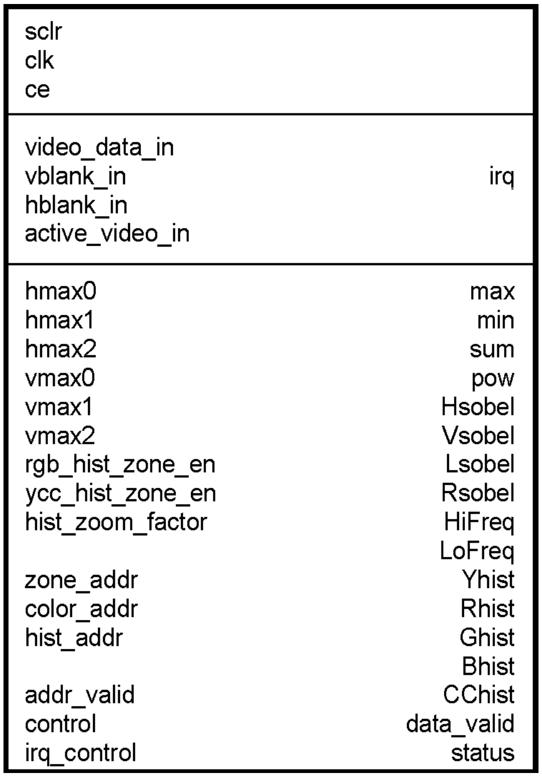


Figure 6: **Core I/O Diagram – General Purpose Processor Interface**

To specify the widths of statistical output ports, the following constants are defined:

$$COLS\_WIDTH= floor\ (\ log2\ (MAX\_COLS\ -1))\ \ +1,$$

$$ROWS\_WIDTH = floor\ (\ log2\ (MAX\_ROWS-1))\ +1,$$

$$ROWS\_WIDTH = \mathrm{floor}\ (\ log_2\ (MAX\_ROWS))\ +1,$$

$$SUM\_WIDTH= DATA\_WIDTH+COLS\_WIDTH+ROWS\_WIDTH,$$

$$SQR\_WIDTH = 2DATA\_WIDTH+COLS\_WIDTH+ROWS\_WIDTH,$$

$$HIST\_WIDTH = COLS\_WIDTH+ROWS\_WIDTH,$$

which are at the definitions of input port widths.

Table 6: **Ports for the General Purpose Processor Interface**

Signal	Width	Direction	Description
hmax0	COLS_WIDTH	IN	Horizontal coordinate of the first zone delineator
hmax1	COLS_WIDTH	IN	Horizontal coordinate of the second zone delineator
hmax2	COLS_WIDTH	IN	Horizontal coordinate of the third zone delineator
vmax0	ROWS_WIDTH	IN	Vertical coordinate of the first zone delineator
vmax1	ROWS_WIDTH	IN	Vertical coordinate of the second zone delineator
vmax2	ROWS_WIDTH	IN	Vertical coordinate of the third zone delineator
zone_addr	4	IN	During Readout, selects the zone for which max, min, sum, pow, Hsobel and Vsobel values are presented at corresponding outputs
color_addr	2	IN	Selects the color channel (0: Green, 1: Red, 2: Blue) for which max, min, sum, pow, Hsobel and Vsobel values are presented at corresponding outputs
hist_addr	DATA_WIDTH	IN	Address port for reading out histogram values through the Rhist, Ghist, Bhist, Yhist, and CChist outputs
rgb_hist_zone_en	16	IN	Bits 0..15 corresponding to the respective zones control whether the zone is included in RGB histograms
ycc_hist_zone_en	16	IN	Bits 0..15 corresponding to the respective zones control whether the zone is included in Y and 2D Cr-Cb histograms
hist_zoom_factor	2	IN	Values 0,1,2,3 refer to Two-dimensional YCC histogram zooming around the gray point by factors of 1,2,4,8.*
addr_valid	1	IN	Logic 1 indicates valid addresses on zone_addr, color_addr and hist_addr
control	4	IN	Bit 0: SW_ENABLE Bit 1: REG_UPDATE Bit 2: READOUT Bit 3: CLEAR_STATUS
irq_control	9	IN	Bit 0: Falling edge on vblank_in detected (VSYNC) interrupt enable Bit 1: Frame Acquisition done (DONE) interrupt enable Bit 2: Horizontal Framing Error Detected Bit 3: Vertical Framing Error Detected Bit 8: General Interrupt Enabled
max	*DATA_WIDTH*	OUT	Maximum value measured for the zone and color channel selected by zone_addr and color_addr
min	*DATA_WIDTH*	OUT	Minimum value measured for the zone and color channel selected by zone_addr and color_addr
sum	*SUM_WIDTH*	OUT	Sum of values measured for the zone and color channel selected by zone_addr and color_addr
pow	*POW_WIDTH*	OUT	Sum of squares of values measured for the zone and color channel selected by zone_addr and color_addr
hiFreq	*POW_WIDTH*	OUT	Sum of absolute values of the High Frequency filter output, applied to luminance values of the currently selected zone
loFreq	*POW_WIDTH*	OUT	Sum of absolute values of the Low Frequency filter output, applied to luminance values of the currently selected zone.

Table 6: **Ports for the General Purpose Processor Interface *(Cont'd)***

Signal	Width	Direction	Description
hsobel	*SUM_WIDTH*	OUT	Sum of luminosity values corresponding to the currently selected zone filtered by a horizontal Sobel Filter
vsobel	*SUM_WIDTH*	OUT	Sum of luminosity values corresponding to the currently selected zone filtered by a vertical Sobel Filter
lsobel	*SUM_WIDTH*	OUT	Sum of luminosity values corresponding to the currently selected zone filtered by a diagonal Sobel Filter
rsobel	*SUM_WIDTH*	OUT	Sum of luminosity values corresponding to the currently selected zone filtered by an anti-diagonal Sobel Filter
rhist	*HIST_WIDTH*	OUT	Red Histogram measurement result corresponding to the current hist_addr address value.
ghist	*HIST_WIDTH*	OUT	Green Histogram measurement result corresponding to the current hist_addr address value
bhist	*HIST_WIDTH*	OUT	Blue Histogram measurement result corresponding to the current hist_addr address value
yhist	*HIST_WIDTH*	OUT	Y (Luminance) Histogram measurement result corresponding to the current hist_addr address value.
cchist	*HIST_WIDTH*	OUT	Two-dimensional CrCb (Chrominance) Histogram measurement result corresponding to the current hist_addr address value
data_valid	1	OUT	Logic 1 indicates valid output on measurement output pins
status	5	OUT	Bit 0: VSYNC falling edge detected Bit 1: DONE: Frame Acquisition Completed Bit 2: VBLANK_error (total rows measured > MAX_ROWS) Bit 3: HBLANK_error (total columns measured > MAX_COLS) Bit 4: INIT_DONE: Timing parameter measurements stabilized
irq	1	OUT	Interrupt request pin
clk	1	input	Rising-edge clock
ce	1	input	Clock enable (active high)
sclr	1	input	Synchronous clear – reset (active high)

*Refer to Setting Up Histogram Calculations for more information.

- **clk - clock:** Master clock in the design, synchronous with, or identical to, the video clock.
- **ce - clock enable:** Pulling `CE` low suspends all operations within the core. Outputs are held, and no input signals are sampled, except for reset (`SCLR` takes precedence over `CE`).
- **sclr - synchronous clear:** Pulling `SCLR` high results in resetting all output pins to zero or their default values. Internal registers within the XtremeDSP™ slice and D-flip-flops are cleared.

Detailed Description

Programming Interface

Input Registers

Each of the following listed registers are double buffered to prevent the user from inadvertently changing register values while a frame of data is being processed, which could lead to inconsistent measurement results.

- stats_reg_04_hmax0
- stats_reg_05_hmax1
- stats_reg_05_hmax2
- stats_reg_07_vmax0
- stats_reg_08_vmax1
- stats_reg_09_vmax2
- stats_reg_10_hist_zoom_factor
- stats_reg_11_rgb_hist_zone_en
- stats_reg_12_ycc_hist_zone_en

The first set of registers is always available for the host processor to write, while the Image Statistics core is using values from the second set of registers. On frame boundaries (rising edge of vblank_in), values from the first set of registers are copied over to the second set of registers if and only if REG_UPDATE (bit 1 of the control register) is set. This mechanism ensures measurement parameters cannot change while data acquisition is in progress. To avoid using partially updated register values, the host processor should set REG_UPDATE=0 before modifying double-buffered input registers, program the registers as needed, then set REG_UPDATE=1 to commit changes.

Some controls, such as the control register itself, may be modified multiple times mid-frame, so the following registers are not double buffered. Writing into these registers elicits immediate response:

- stats_reg00_control
- stats_reg01_sw_reset
- stats_reg03_irq_control
- stats_reg13_zone_addr
- stats_reg14_color_addr
- stats_reg15_hist_addr
- stats_reg16_addr_valid

Processing States

The core distinguishes acquisition and readout periods to avoid modification of single-buffered measurement data while it is being read out. After readout, block RAMs and registers have to be re-initialized before the next acquisition cycle may commence.

After reset or power-up, the core cycles through the "Initialization," "Wait for VBLANK," and "Data Acquisition" states.

Figure 7 shows the top-level state diagram of the Image Statistics core.

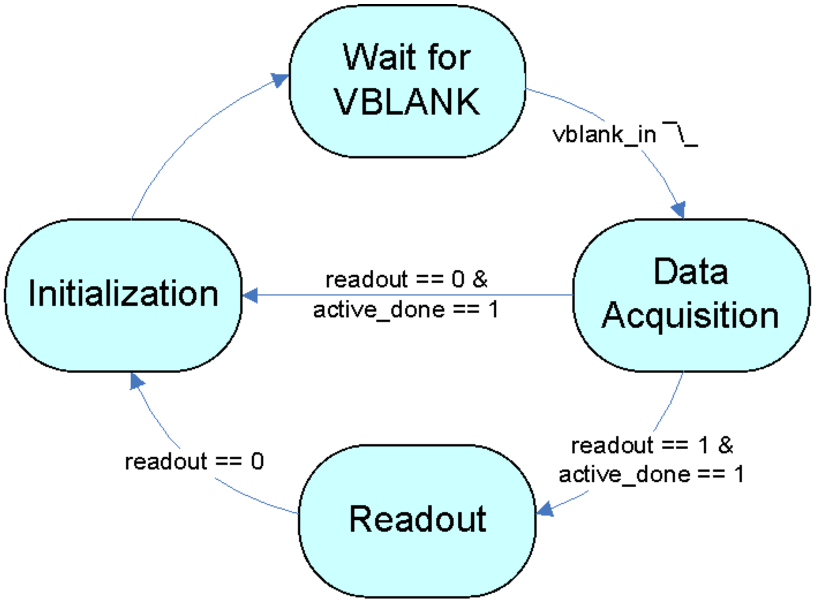


Figure 7: **Image Statistics IP Core State Diagram**

Initialization

The one- and two-dimensional histograms are stored in block RAMs, which should be cleared before the IP core can start data acquisition. Clearing of block RAMs may take 256, 1024 or 4096 CLK cycles corresponding to 8, 10, or 12-bit wide input data. Block RAM initialization may take several scan-lines, depending on the input resolution.

Once the core is finished with block RAM initialization, it progresses to the "Wait for VBLANK" state where it remains until a falling edge on `vblank_in` is detected, at which time it enters the "Data Acquisition" state.

Data Acquisition

In the "Data Acquisition" state, the core updates all internal measurement values with the pixel data presented on `video_data_in` when data is qualified with `active_video_in` = 1.

The core proceeds to the "Readout" state after the last active scan-line, which may occur several scan-lines before the rising edge on `vblank_in`. The core identifies the last active scan-line based on measurements of the previous frame.

Readout

In the readout state, the core does not collect any more statistical information, and the multiplexing/addressing mechanism on the output is activated. Once the user provides addresses that are qualified valid by asserting the `addr_valid` pin, the core fetches and displays information on its output ports pertaining to the input addresses. Valid output data is identified by the `data_valid` output pin.

Synchronization

A semaphore-based mechanism is used to synchronize external frame timing with host processor register writes, data readouts and data acquisition.

The semaphores involved in synchronizing host processor activity with the incoming video stream are:

- `DONE` flag (bit 1 of the `status` register)
- `REG_UPDATE` (bit 1 of the `control` register)
- `READOUT` (bit 2 of the `control` register)
- `CLR_STATUS` (bit 3 of the `control` register)

The software flow diagram for normal system operation is shown in Figure 8 and is described following the figure.

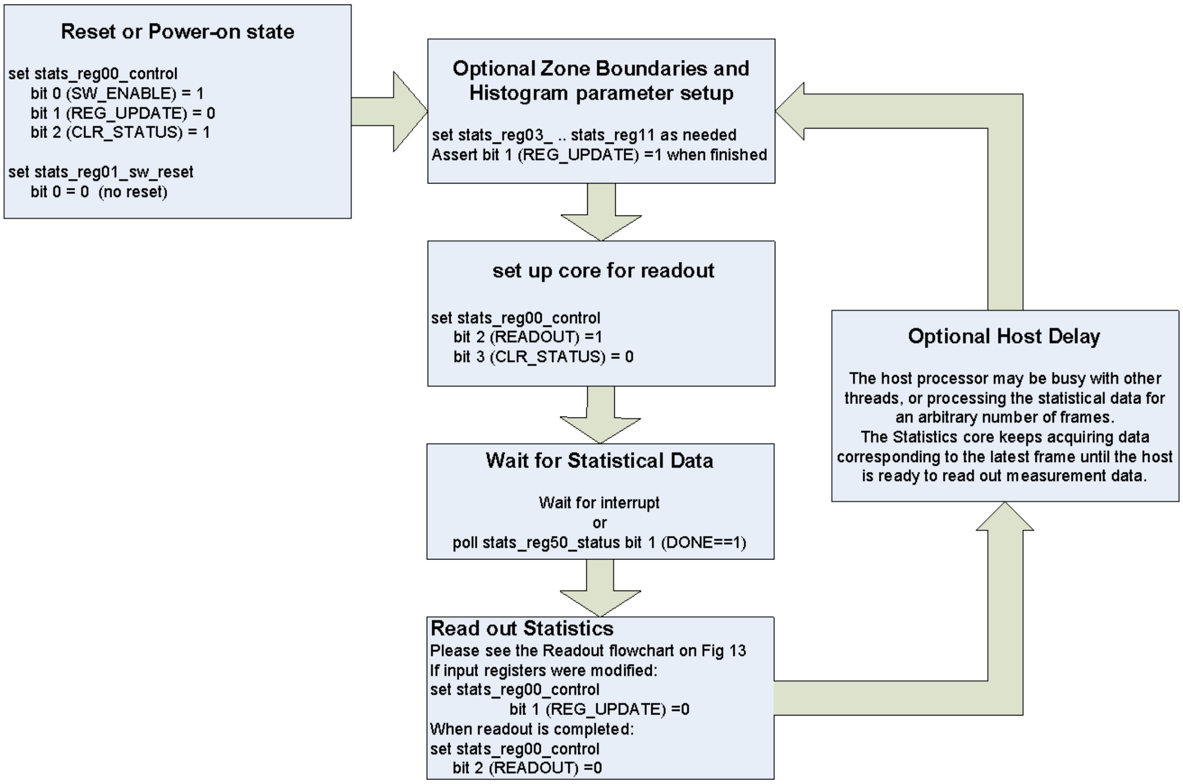

Reset or Power-on state
set stats_reg00_control
bit 0 (SW_ENABLE) = 1
bit 1 (REG_UPDATE) = 0
bit 2 (CLR_STATUS) = 1
set stats_reg01_sw_reset
bit 0 = 0 (no reset)
Optional Zone Boundaries and Histogram parameter setup
set stats_reg03_ .. stats_reg11 as needed
Assert bit 1 (REG_UPDATE) =1 when finished
set up core for readout
set stats_reg00_control
bit 2 (READOUT) =1
bit 3 (CLR_STATUS) = 0
Wait for Statistical Data
Wait for interrupt
or
poll stats_reg50_status bit 1 (DONE==1)
Read out Statistics
Please see the Readout flowchart on Fig 13
If input registers were modified:
set stats_reg00_control
bit 1 (REG_UPDATE) =0
When readout is completed:
set stats_reg00_control
bit 2 (READOUT) =0
Optional Host Delay
The host processor may be busy with other threads, or processing the statistical data for an arbitrary number of frames.
The Statistics core keeps acquiring data corresponding to the latest frame until the host is ready to read out measurement data.

Figure 8: **Software Flow Diagram for Normal System Operation**

When data acquisition is finished, the core asserts status flag DONE. If the corresponding Interrupt Enable (DONE_IRQ_EN) and the General Interrupt Enable (IRQ_EN) bits are set to 1, this event also triggers the interrupt request (IRQ) signal.

After the data acquisition state, depending on the state of the READOUT flag, the core either enters the readout state (READOUT = 1), or discards measurement data by re-initializing block RAMs and registers and preparing for acquiring data from the next frame (READOUT = 0).

This mechanism relieves the host processor from having to service the core when statistical data is not needed and allows the core to continuously process frames, so when the host processor is ready to poll statistical information, information from the last frame is available.

If the core enters the readout state, it remains there until the host processor signals being done with reading out measurement data, which may take a few lines, or several frames depending on the speed and the workload of the host processor. Therefore termination of the readout state is decoupled from the input video stream.

Once the host processor deasserts READOUT, the core immediately clears (re-initializes) block RAMs and registers and proceeds to acquire data from the next frame.

After deasserting READOUT, the host processor may assert it again immediately, enabling the core to enter the “Readout” state after acquisition of the current frame is complete.

NOTES:

1. READOUT has to be asserted before the statistics core is done acquiring the next frame, or the core discards the data and self-triggers to acquire the next frame.
2. For the core to process every subsequent frame, the vertical blanking period has to be at least as long as the number of scan-lines it takes to initialize the block RAMs. For example, in the pessimistic case of using SD sensor (720 pixels per line) with 12 bit data (4k deep block RAMs), the minimum vertical blanking period has to be 4096/720 = 5.68, or at least six lines.

Setting Up Zone Boundaries

The zone boundaries for the 16 zones can be set up by programming the positions of three vertical and three horizontal delimiters as shown in Figure 9. Complemented by the constraints that the top-left corner of Zone 0 is flush with the left-top corner of the active image, and the bottom-right corner of Zone 15 is flush with the bottom-right corner of the active image, these values uniquely define the corners of all zones.

Data is collected during the active (and non-blank) period of the frame, and all zones traversed by the current scan-line are updated with the input data in parallel. Zone boundaries should be set up before acquiring the first frame of data by programming the hmax and vmax registers.

***NOTE:* The minimum horizontal and vertical size of each zone must be at least 2**, along with the following geometric constraints:

- 0 < hmax0 < hmax1 < hmax2 < *MAX_COLS*
- 0 < vmax0 < vmax1 < vmax2 < *MAX_ROWS*

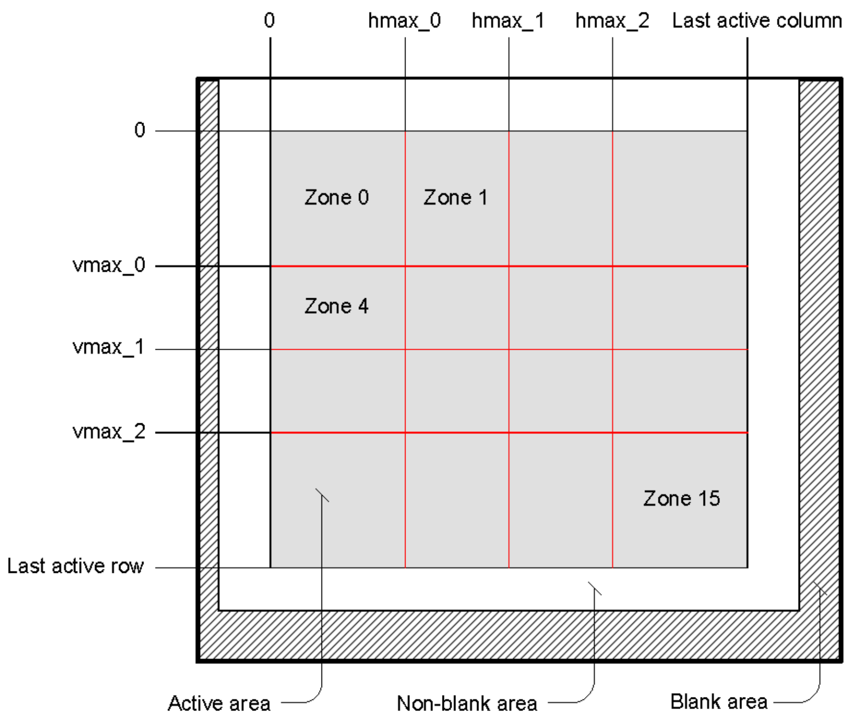


Figure 9: **Setting Up Zone Boundaries**

Setting Up Histogram Calculations

Histogram data is calculated and stored in block RAMs; hence calculating RGB and YCrCb histograms for all zones independently significantly increases the amount of FPGA resources required. Employing a mask to select which zones are involved in histogram calculation covers most typical applications:

- Calculating histogram values for one particular zone
- Calculating histogram values over an area of the image (such as the central zones, or zones in the corners)
- Calculating histogram values over the whole image

Figure 10 demonstrates how zones for RGB (red squares) and YCrCb (yellow circles) histogram calculations can be selected. For the example shown in Figure 10, zones 3, 4, 6, 7, 8 and 14 are selected for the Y and CrCb histograms. Correspondingly, bits 3,4,6,7,8 and 14 are set in `ycc_hist_zone_en`, resulting in a value of 0x000041D8.

Similarly, for the R,G, and B histograms, zones 1, 2, 3, 7, 8, 10, 11 and 14 are selected. Correspondingly bits 1, 2, 3, 7, 8, 10, 11 and 14 are set in `rgb_hist_zone_en`, resulting in a value of 0x00004D8E.

For the two-dimensional Cr-Cb histogram, there is another control, `stats_reg10_hist_zoom_factor`, that helps tailor the Cr-Cb histogram calculation to the higher-level algorithm that consumes the 2D histogram results.

Consequently, Cr and Cb values have the same dynamic range as the input data. Cr and Cb are represented internally on `DATA_WIDTH` bits. A full precision Cr-Cb histogram would constitute a sparse 4k x 4k table that may be too large to implement within an FPGA. Therefore Cr and Cb are quantized to `DATA_WIDTH`/2 bits for histogramming. This quantization process inevitably involves loss of information. The use of the zoom factor (`stats_reg10_hist_zoom_factor`) enables focusing on certain aspects of the histogram to minimize the effects of the information loss.

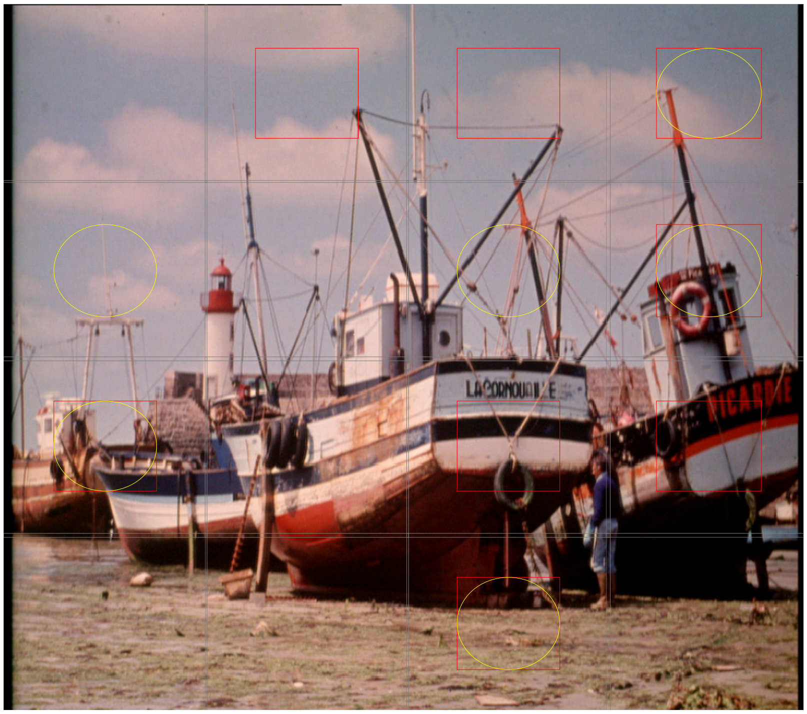


Figure 10: **Selecting Individual Zones for RGB and YCrCb Histograms**

Some higher level algorithms, such as gamut stretching [Ref 4], are concerned with the overall histogram, while other methods are concerned only with the central section, the area around the neutral point, to identify color casts. To support either type of algorithm, the histogram zoom factor allows the user to trade off resolution with range. The histogram zoom factor controls which bits of Cr and Cb values are selected for histogram binning. By setting the `hist_zoom_factor` to 0, the whole Cr-Cb histogram is represented at the output, as Cr and Cb values are simply quantized to `DATA_WIDTH`/2 bits. For example if `DATA_WIDTH` = 8, this quantization results in only the most significant four bits, bits 4, 5, 6 and 7, being used for histogram binning (see Table 7).

Table 7: **Histogram Zoom Factor Bit Selections for DATA_WIDTH = 8 Bit Input Data**

Histogram zoom Factor	Bits Used for Binning							
0	▨ 7	▨ 6	▨ 5	▨ 4	3	2	1	0
1	▨ 7	▨ 6	▨ 5	▨ 4	▨ 3	2	1	0
2	▨ 7	▨ 6	▨ 5	▨ 4	▨ 3	▨ 2	1	0
3	▨ 7	▨ 6	▨ 5	▨ 4	▨ 3	▨ 2	▨ 1	0

When `hist_zoom_factor` (or `stats_reg10_hist_zoom_factor`) is set to a value other than 0, the resulting two-dimensional histogram represents only the central portion of the Cr-Cb histogram; pixels with extreme Cr-Cb values may fall outside the range represented by `DATA_WIDTH`/2 bits.

To enable further reduction of core footprint, RGB, Y, and Cr-Cb histograms can be individually enabled/disabled during generation time via the CORE Generator graphical user interface. If a particular type of histogram is not needed by the higher level algorithms, the core footprint can be reduced by 1,2, or 4 block RAMs depending on the input data resolution (`DATA_WIDTH`) and the target family.

Control Signals and Timing

Reading Out Statistical Results

Figure 11 shows an example of data readout timing and use of the control, irq_control and status registers. In this example, an empty frame followed by a test frame (vblank_in, hblank_in, active_video_in, video_data_in) are processed by the core. The core cycles through the "Clear RAMs" and "Wait on VBLANK" stages, from which it transitions to the "Acquisition" state on the second falling edge of vblank_in.

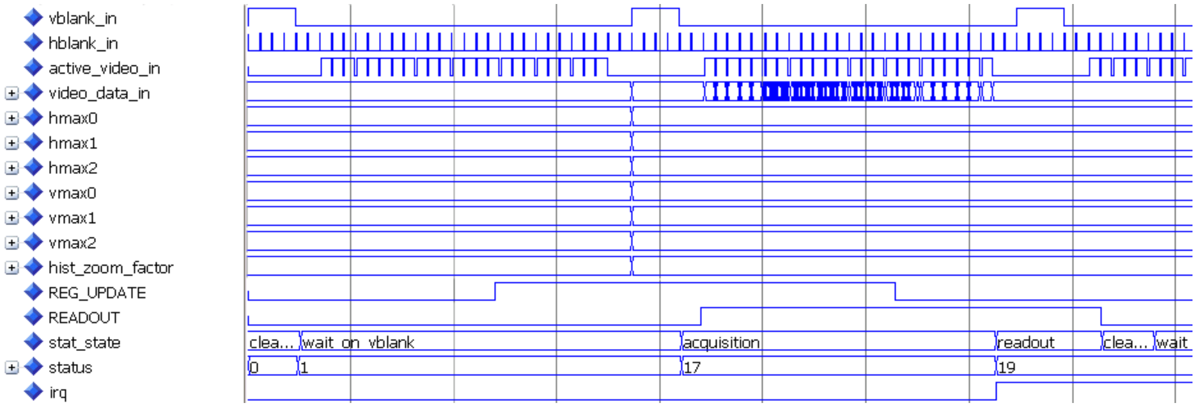


Figure 11: **Frame and Readout Timing**

As discussed in the Synchronization section, the Image Statistics core signals the end of data acquisition by asserting the DONE flag of the status register, which also triggers an interrupt if the irq_control register is set up to enable interrupts. In this example, bits 8 (IRQ_EN) and 1 (DONE_IRQ_EN) are set to 1, as indicated by the decimal value 258, resulting in the interrupt output (irq) transitioning high (event marked by the red cursor) at the same time the core signals the end of data acquisition (DONE flag of the status register).

During the frame, bit 2 (READOUT) was asserted, which at the end of the active portion of the frame instructs the core to enter the "Readout" state.

Bit READOUT of the control register has to be set before the end of the frame; otherwise the core does not enter the readout mode but clears measurement data and arms itself for capturing the next frame.

After the host processor is finished reading out relevant statistical data, it programs bit 2 (READOUT) of the control register to 0, which instructs the core to re-initialize by entering the "clear-RAMs" state. The example in Figure 11 also demonstrates the use of the REG_UPDATE flag. The user at any point could have modified values for input registers, such as hmax0, hmax1, hmax2, vmax0, vmax1, vmax2, rgb_hist_zone_en, ycc_hist_zone_en, or hist_zoom_factor. After setting all input registers to their desired value, REG_UPDATE was asserted, which resulted to all internal registers to latch in the user input at the rising edge of vblank_in. Signals hmax, vmax, rgb_hist_zone_en, ycc_hist_zone_en, and hist_zoom_factor values displayed in Figure 11 demonstrate how the internal register values change simultaneously on the rising edge of vblank_in if REG_UPDATE is set to 1.

Addressing

Due to the large number of statistical data collected by the core, presenting all data simultaneously on core outputs is not feasible.

Inputs `zone_addr` and `color_addr` for the General Purpose Processor Interface, or registers `stats_reg12_zone_addr` and `stats_reg13_color_addr` for the EDK pCore Interface, facilitate reading out the max, min, sum and power result for specific zones and color channels.

The input `hist_addr` for the General Purpose Processor Interface, or the `stats_reg14_hist_addr` register for the EDK pCore Interface, facilitate addressing of histogram values.

The Image Statistics core provides a simple handshaking interface for reading out data. After setting the address inputs (registers in case the EDK pCore interface is used) as needed, asserting the `addr_valid` pin signals to the core that valid addresses are present. In turn, the core fetches data corresponding to the addresses and marks valid data on the core outputs by asserting the `data_valid` output/register (Figure 12).

All maximum, minimum, sum, sum of squares, Sobel and frequency contents can be read out by accessing zones 0-15 and color channels (coded 0,1,2) sequentially. If the host processor interface and the Image Statistics core are in the same CLK domain, addressing can be simplified, such that multiple addresses are supplied during the active portion of `addr_valid`. When a sequence of valid addresses is presented to the core, the sequence of corresponding valid data becomes available with a latency of five CLK cycles (Figure 13).

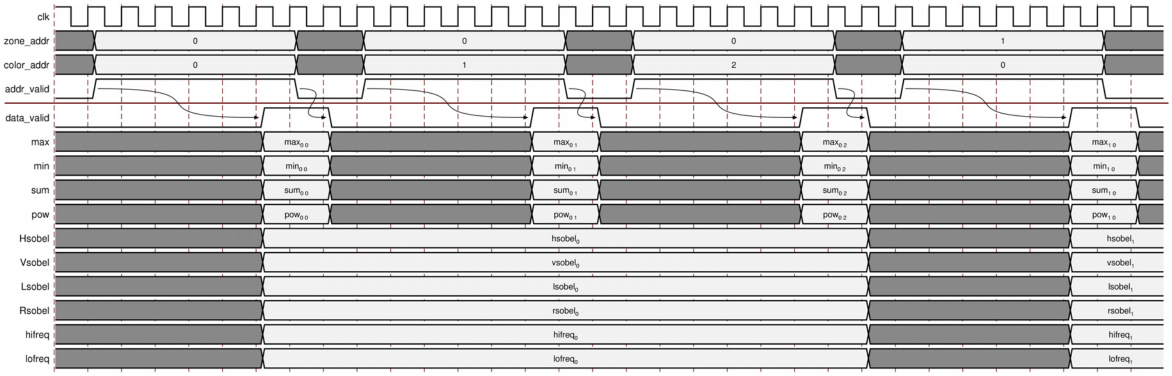


Figure 12: **Readout Addressing**

Figure 13 illustrates reading out histogram data. To shorten the readout period, histogram data can be read out in parallel with other statistical data.

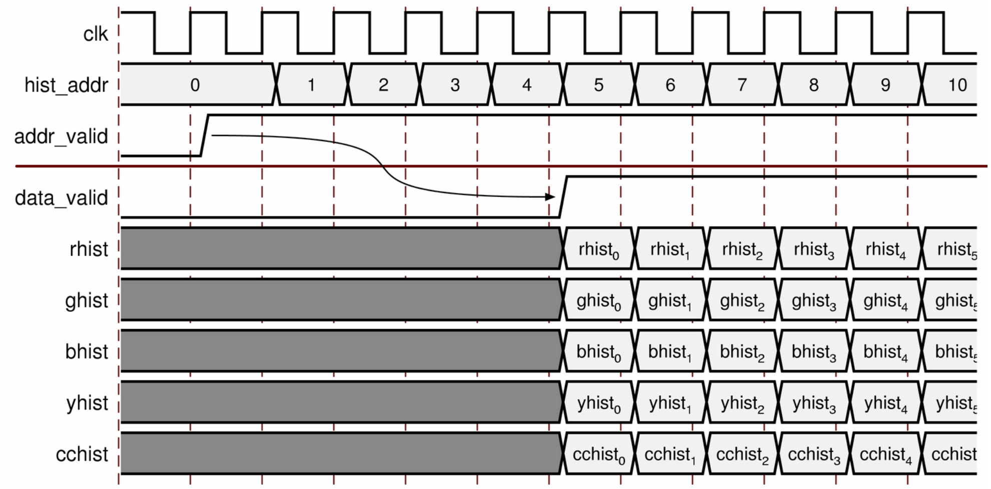

clk
hist_addr
0
1
2
3
4
5
6
7
8
9
10
addr_valid
data_valid
rhist
rhist$_0$ rhist$_1$ rhist$_2$ rhist$_3$ rhist$_4$ rhist$_5$
ghist
ghist$_0$ ghist$_1$ ghist$_2$ ghist$_3$ ghist$_4$ ghist$_5$
bhist
bhist$_0$ bhist$_1$ bhist$_2$ bhist$_3$ bhist$_4$ bhist$_5$
yhist
yhist$_0$ yhist$_1$ yhist$_2$ yhist$_3$ yhist$_4$ yhist$_5$
cchist
cchist$_0$ cchist$_1$ cchist$_2$ cchist$_3$ cchist$_4$ cchist

Figure 13: **Reading Out Histogram Data**

Programmer's Guide

EDK pCore API Functions

This section describes the functions included in the C driver (`stats.c` and `stats.h`) generated for the EDK pCore API.

The software API is provided to allow easy access to the pCore registers of the Image Statistics IP pCore defined in Table 2. To utilize the API functions provided, the following two header files must be included in the user C code:

```
#include "stats.h"

#include "xparameters.h"
```

The hardware settings of your system, including the base address of your Image Statistics core, are defined in the `xparameters.h` file. The `stats.h` file contains the macro function definitions for controlling the Image Statistics pCore.

The drivers subdirectory of the pCore contains a file, `example.c`, in the `stats_v2_00_a/example` subfolder. This file is a sample C program that demonstrates how to use the Image Statistics pCore API.

Each software register defined in Table 2 has a constant defined in `stats.h` that is set as the offset for that register. To write to a register, use the STATS_WriteReg() function using the base address of the Image Statistics pCore instance (from `xparameters.h`), the offset of the desired register, and the data to write.

The definition of this macro is:

`STATS_WriteReg(uint32 BaseAddress, uint32 RegOffset, uint32 Data)`

> This macro writes a given register.
>
> BaseAddress is the Xilinx EDK base address of the Image Statistics core (from `xparameters.h`).

RegOffset is the register offset of the register (defined in Table 2).

Data is the 32-bit value to write to the register.

Example:

```
STATS_WriteReg(XPAR_STATS_0_BASEADDR, STATS_REG_00_CONTROL, 1);
```

Similarly, reading a value from a register uses the base address and offset for the register:

`STATS_ReadReg(uint32 BaseAddress, uint32 RegOffset)`

This macro returns the 32-bit unsigned integer value of the register.

BaseAddress is the Xilinx EDK base address of the Image Statistics core (from xparameters.h).

RegOffset is the register offset of the register (defined in Table 2).

Example:

```
Xuint32 value = STATS_ReadReg(XPAR_STATS_0_BASEADDR, STATS_REG_01_STATUS);
```

Based on the register read and write primitives, the following macros are defined to control the operation of the Image Statistics IP pCore:

`STATS_Enable(uint32 BaseAddress)`

This macro enables an Image Statistics pCore instance.

BaseAddress is the Xilinx EDK base address of the Image Statistics core (from xparameters.h).

`STATS_Disable(uint32 BaseAddress)`

This macro disables an Image Statistics pCore instance.

BaseAddress is the Xilinx EDK base address of the Image Statistics core (from `xparameters.h`).

`STATS_Reset(uint32 BaseAddress);`

This macro resets an Image Statistics instance.

Reset affects the all core measurement outputs immediately, and forcing outputs to 0 until `STATS_ClearReset()` is called.

BaseAddress is the Xilinx EDK base address of the Image Statistics core (from `xparameters.h`).

`STATS_ClearReset(uint32 BaseAddress);`

This macro clears the reset flag of the core, which allows it to re-sync with the input video stream and return to normal operation.

BaseAddress is the Xilinx EDK base address of the Image Statistics core (from `xparameters.h`).

`STATS_RegUpdateEnable(uint32 BaseAddress);`

The zone boundary, histogram zone enablement, and histogram zoom factor registers are double-buffered inside the Image Processing core. The first set of registers is always available for the host processor to write, while the Image Statistics core is using values from the second set of registers. On frame boundaries, at the rising edge of `VBlank_in`, values from the first set of registers are copied over to the second set of registers only if `REG_UPDATE` (bit 1 of the control register) is set. This mechanism ensures that measurement parameters cannot change while data acquisition is in progress (for more information see, section Programming Interface).

After updating register values, calling RegUpdateEnable causes the Image Statistics pCore to start using the updated values on the next rising edge of `VBlank_in`. The user must manually disable the register update before register updates begin to make sure all updates will affect the same frame.

This function only works when the Image Statistics core is enabled.

BaseAddress is the Xilinx EDK base address of the Image Statistics core (from `xparameters.h`).

`STATS_RegUpdateDisable(uint32 BaseAddress);`

The zone boundary, histogram zone enablement, and histogram zoom factor registers are double-buffered inside the Image Processing core. The first set of registers is always available for the host processor to write, while the Image Statistics core is using values from the second set of registers (for more information, see Programming Interface). Disabling the Register Update prevents the Image Statistics pCore to use the freshly updated zone boundary, histogram zone enablement, or histogram zoom factor register values after rising `VBlank_in` edges.

Xilinx recommends that the Register Update be disabled while writing to the zone boundary, histogram zone enablement, and histogram zoom factor registers, until all register write operations are complete.

This function only works when the Image Statistics core is enabled.

BaseAddress is the Xilinx EDK base address of the Image Statistics core (from `xparameters.h`).

Core Resource Utilization and Performance

For an accurate measure of the usage of device resources for a particular instance, click **View Resource Utilization** in CORE Generator after generating the core.

Information presented in Table 8 – Table 11 are guidelines to the resource utilization of the Image Statistics core for Virtex-5, Virtex-6, Spartan-3A DSP, and Spartan-6 FPGA families. The design was tested using Xilinx ISE v13.1 tools with area constraints (see table footnotes) and default tool options.

Table 8 – Table 11 present the resource utilization and target clock frequencies of the Image Statistics core for all input width combinations with three typical values for Maximum Number of Rows and Maximum Number of Columns. These characterization tests were performed with the Maximum Number of Rows and Maximum Number of Columns parameters set to equal values, all histograms enabled and using the General Purpose Processor interface.

Table 8: **Resource Utilization and Target Speed for Virtex-5 (xc5vlx330t-1ff1760)**[1]

Data Width	Max Rows Max Cols	Histogram			FFs	LUTs	Slices	DSP48s	RAMB		Fmax
		Y	RGB	CC					18X2	36EXP	
8	1023	Yes	Yes	Yes	2366	1723	1003	14	1	0	237.53
8	2200	Yes	Yes	Yes	2584	1878	1109	14	1	1	193.42
10	1023	Yes	Yes	Yes	2647	2012	1070	14	0	1	230.31
10	2200	Yes	Yes	Yes	2865	2169	1198	14	1	2	226.09
12	1023	Yes	Yes	Yes	2931	2240	1179	14	0	1	262.19
12	2200	Yes	Yes	Yes	3149	2144	1238	14	2	2	212.77

1. Speedfile version: PRODUCTION 1.72 2011-01-11

Table 9: **Resource Utilization and Target Speed for Spartan-3A DSP (xc3sd3400a-4fg676)**[1]

Data Width	Max Rows Max Cols	Histogram			FFs	LUTs	Slices	DSP48s	BRAMs	Fmax
		Y	RGB	CC						
8	1023	Yes	Yes	Yes	2309	2378	2152	14	1	135.2813853
8	2200	Yes	Yes	Yes	2527	2587	2260	14	4	136.1285053
10	1023	Yes	Yes	Yes	2595	2658	2316	14	2	136.1285053
10	2200	Yes	Yes	Yes	2812	2866	2414	14	5	138.1406272
12	1023	Yes	Yes	Yes	2876	2947	2545	14	2	134.7527287
12	2200	Yes	Yes	Yes	3091	3155	2549	14	6	135.3546291

1. Speedfile version: PRODUCTION 1.33 2011-01-11

Table 10: **Resource Utilization and Target Speed for Spartan-6 (xc6slx150fgg676-2)**[1]

Data Width	Max Rows Max Cols	Histogram			FFs	LUTs	Slices	DSP48s	RAMB		Fmax
		Y	RGB	CC					8BWER	16BWER	
8	1023	Yes	Yes	Yes	2286	1495	782	14	0	1	158.33
8	2200	Yes	Yes	Yes	2521	1622	905	14	0	4	147.69
10	1023	Yes	Yes	Yes	2585	1651	869	14	1	1	157.70
10	2200	Yes	Yes	Yes	2804	1827	892	14	0	5	157.88
12	1023	Yes	Yes	Yes	2870	1840	886	14	1	1	159.44
12	2200	Yes	Yes	Yes	3088	1996	985	14	0	6	157.73

1. Speedfile version: PRODUCTION 1.15 2011-01-12

Table 11: **Resource Utilization and Target Speed for Virtex-6 (xc6vsx315t-2ff1759)**[1]

Data Width	Max Rows Max Cols	Histogram			FFs	LUTs	Slices	DSP48s	RAMB		Fmax
		Y	RGB	CC					18E1	36E1	
8	1023	Yes	Yes	Yes	2278	1565	642	14	1	0	241.43
8	2200	Yes	Yes	Yes	2496	1736	737	14	2	1	243.49
10	1023	Yes	Yes	Yes	2558	1688	811	14	0	1	235.40
10	2200	Yes	Yes	Yes	2777	1931	855	14	1	2	249.81
12	1023	Yes	Yes	Yes	2838	1920	868	14	0	1	225.33
12	2200	Yes	Yes	Yes	3057	2143	913	14	2	2	231.11

1. Speedfile version: PRODUCTION 1.13a 2011-01-11

Optional Histogram Calculation

Table 12 – Table 15 present resource utilization numbers for all possible combinations of optional histogram settings, for all supported device families, using 8-bit input data and the maximum numbers of rows and columns set to 2047.

Table 12: **Effects of Optional Histogram Calculation on Resource Utilization for Virtex-5 (xc5vlx330t-1ff1760)**

Data Width	Max Rows Max Cols	Histogram			FFs	LUTs	Slices	DSP48s	RAMB		Fmax
		Y	RGB	CC					18X2	36EXP	
8	2200	No	No	No	2584	1879	1102	14	1	1	227.6867031
8	2200	No	No	Yes	2770	2006	1183	16	2	1	226.3980077
8	2200	No	Yes	No	2942	2057	1194	14	4	1	227.6348737
8	2200	No	Yes	Yes	3102	2191	1232	16	5	1	226.5518804
8	2200	Yes	No	No	2727	1940	1093	14	2	1	227.5830678
8	2200	Yes	No	Yes	2865	2044	1181	16	4	1	226.9117313
8	2200	Yes	Yes	No	3059	2123	1225	14	5	1	222.4694105
8	2200	Yes	Yes	Yes	3198	2247	1396	16	6	1	226.4492754

Table 13: **Effects of Optional Histogram Calculation on Resource Utilization for Spartan-3 (xc3sd3400a-4fg676-5)**

Data Width	Max Rows Max Cols	Histogram			FFs	LUTs	Slices	DSP48s	Block RAMs	Fmax
		Y	RGB	CC						
8	2200	No	No	No	2527	2587	2260	14	4	136.1285053
8	2200	No	No	Yes	2781	2846	2411	16	5	141.4027149
8	2200	No	Yes	No	2896	2831	2733	14	7	119.2037192
8	2200	No	Yes	Yes	3116	3055	2602	16	8	140.3508772
8	2200	Yes	No	No	2674	2691	2358	14	5	136.761488
8	2200	Yes	No	Yes	2879	2894	2531	16	6	144.1545337
8	2200	Yes	Yes	No	3013	2916	2596	14	8	138.064338
8	2200	Yes	Yes	Yes	3215	3100	3120	16	9	120.7437817

Table 14: **Effects of Optional Histogram Calculation on Resource Utilization for Spartan-6 (xc6slx150fgg676)**

Data Width	Max Rows Max Cols	Histogram			FFs	LUTs	Slices	DSP48s	RAMB		Fmax
		Y	RGB	CC					8BWER	16BWER	
8	2200	No	No	No	2521	1622	905	14	1	1	147.69
8	2200	No	No	Yes	2754	1785	905	16	2	1	157.85
8	2200	No	Yes	No	2878	1860	939	14	4	1	153.26
8	2200	No	Yes	Yes	3088	2073	988	16	5	1	153.78
8	2200	Yes	No	No	2664	1809	775	14	2	1	157.83
8	2200	Yes	No	Yes	2846	1899	863	16	4	1	158.86
8	2200	Yes	Yes	No	2998	1936	993	14	5	1	138.73
8	2200	Yes	Yes	Yes	3185	2045	1125	16	6	1	157.60

Table 15: **Effects of Optional Histogram Calculation on Resource Utilization for Virtex-6 (xc6vsx315t-2ff1759)**

Data Width	Max Rows Max Cols	Histogram			FFs	LUTs	Slices	DSP48s	RAMB		Fmax
		Y	RGB	CC					18E1	36E1	
8	2200	No	No	No	2496	1614	847	14	2	1	237.53
8	2200	No	No	Yes	2731	1794	948	16	3	1	193.42
8	2200	No	Yes	No	2857	1987	838	14	5	1	230.31
8	2200	No	Yes	Yes	3064	2064	977	16	6	1	226.09
8	2200	Yes	No	No	2641	1852	729	14	3	1	262.19
8	2200	Yes	No	Yes	2827	1872	913	16	4	1	212.77
8	2200	Yes	Yes	No	2974	2064	874	14	6	1	227.32
8	2200	Yes	Yes	Yes	3160	1966	1241	16	7	1	194.93

Known Issues

For for the latest Known Issues see XTP025.

References

1. Processor Local Bus (PLB) v4.6
2. Vicent Caselles, Jose-Luis Lisani, Jean-Michel Morel, Guillermo Sapiro: *Shape Preserving Local Histogram Modification*
3. Joung-Youn Kim, Lee-Sup Kim, and Seung-Ho Hwang: *An Advanced Contrast Enhancement Using Partially Overlapped Sub-Block Histogram Equalization*
4. Simone Bianco, Francesca Gasparini and Raimondo Schettini: *Combining Strategies for White Balance*
5. G. Finlayson, M. Drew, and B. Funt, “Diagonal Transform Suffice for Color Constancy” in *Proc. IEEE International Conference on Computer Vision*, Berlin, pp. 164-171, 1993
6. Keith Jack: Video Demystified, 4th Edition, *ISBN 0-7506-7822-4*, pp 15-19

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY.*

License Options

The Image Statistics core provides the following three licensing options:

- Simulation Only
- Full System Hardware Evaluation
- Full

After installing the required Xilinx ISE software and IP Service Packs, choose a license option.

Simulation Only

The Simulation Only Evaluation license key is provided with the Xilinx CORE Generator tool. This key lets you assess core functionality with either the example design provided with the Image Statistics core, or alongside your own design and demonstrates the various interfaces to the core in simulation. (Functional simulation is supported by a dynamically generated HDL structural model.)

No action is required to obtain the Simulation Only Evaluation license key; it is provided by default with the Xilinx CORE Generator software.

Full System Hardware Evaluation

The Full System Hardware Evaluation license is available at no cost and lets you fully integrate the core into an FPGA design, place-and-route the design, evaluate timing, and perform functional simulation of the Image Statistics core using the example design and demonstration test bench provided with the core.

In addition, the license key lets you generate a bitstream from the placed and routed design, which can then be downloaded to a supported device and tested in hardware. The core can be tested in the target device for a limited time before timing out (ceasing to function), at which time it can be reactivated by reconfiguring the device.

To obtain a Full System Hardware Evaluation license, do the following:

1. Navigate to the product page for this core.
2. Click Evaluate.
3. Follow the instructions to install the required Xilinx ISE software and IP Service Packs.

Full

The Full license key is available when you purchase the core and provides full access to all core functionality both in simulation and in hardware, including:

- Functional simulation support
- Full implementation support including place and route and bitstream generation
- Full functionality in the programmed device with no time outs

To obtain a Full license key, you must purchase a license for the core. Click on the "Order" link on the Xilinx.com IP core product page for information on purchasing a license for this core. After doing so, click the "How do I generate a license key to activate this core?" link on the Xilinx.com IP core product page for further instructions.

Installing Your License File

The Simulation Only Evaluation license key is provided with the ISE CORE Generator system and does not require installation of an additional license file. For the Full System Hardware Evaluation license and the Full license, an email will be sent to you containing instructions for installing your license file. Additional details about IP license key installation can be found in the ISE Design Suite Installation, Licensing and Release Notes document.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
12/02/09	1.0	Initial Xilinx release.
07/23/10	1.1	Fixed CR 54061 by adding Xilinx Streaming Video Interface (XSVI) information.
03/1/11	2.0	Updated for core version 2.0.

Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter "Information," to you "AS IS" with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.