Bi8921

1/ 2.7”   2 MP (UXGA) Color Sensor w/JPEG

Version 3.2   September 1. 2006

Biomorphic VLSI

Description

Biomorphic’s Bi8921 is a highly affordable CMOS color image sensor which debuts new technology with an advanced 2 megapixel sensor, color processing pipeline, and JPEG compression into a single compact component. The Bi8921 offers UXGA resolution and is designed for cellular phone applications.

Our highly advanced sensor design provides single-chip integration of a full suite of image processing functions to directly support viewfinder, still image, and video capture capabilities in today’s latest mobile imaging platforms. Plus, with the addition of on board JPEG compression, full size images can be captured with high readout speeds, yet low data transfer rates. With an embedded thumbnail image in the jpeg stream, the host processor need not decompress the full image for review.

The Bi8921 provides superb image quality, ease of integration, compact size, and low power consumption for outstanding overall performance.

Features

□ **Imager Array**

- High image resolution with 2 million active image pixels.
- RGB Bayer color mosaic filters deliver high color fidelity
- Integrated Correlated Double Sampling, black level correction and 10-bit A/D
- Electronic rolling shutter
- Scan direction mirroring and flipping.
- 10 bit ADC resolution
- Supports full frame readout at >12 fps with 30 MHz imager clock and 20 MHz JPEG out

□ **Image Processing Unit**

- Picture enhancement features:
  - Channel equalization
  - Bad pixel correction
  - Lens shading correction (color dependent)
- Advanced interpolation of RGB data
- Programmable processing for:
  - Color correction
  - Color saturation
  - Gamma curves
  - Edge enhancement
  - Focus quality measure
- Programmable color space conversion for output in RGB, YCbCr, or YUV
- Sub sampling and windowing

□ **Automatic Exposure and White Balance**

- Internal statistic generation
- Internal loop closure for automatic operation
- Host application override possible

□ **Output Modes**

- 8 bit output
- RGB565 or RGB444 formats for convenient LCD preview
- YCbCr or YUV format for preview mode or video stream capture
- CCIR tags optional
- TI OMAP imager port compatible
- JPEG with programmable quality factor and embedded YUV thumbnail

□ **Power**

- Flexible supply range with power saving features for extended application battery life.

□ **Miscellaneous**

- Single clock input and embedded PLL for simplicity of system design.
- Fast-mode I^{2}C host interface
- Tristate-able outputs for multiple camera applications
- I^{2}C addressed at 0x45

Image format	1/ 2.7 inch 5.79 mm (H) x 4.35 mm (V)
Pixel size	3.6 µm x 3.6 µm
# of active pixels	1608 (H) x 1208 (V)
Die Size (before dicing)	6797 µm x 7774 µm
Optical black pixels -	24 cols left / 8 rows top
Green Sensitivity	3.0 V/lux-s
Power Supply	2.4 to 3.3 VDC
Package	8 inch wafer

Biomorphic VLSI, Inc
123 Hodencamp Road, Suite 204, Thousand Oaks, Ca, 91360
(805) 497-9055, fax (805)497-9725, info@biomorphic.com, http://www.biomorphic.com/

Functional Description

General Description

Figure 1 shows the Bi8921 internal architecture. There are four pipelined processing functions, the programmable clock generator PLL, the host interface, and imager block.

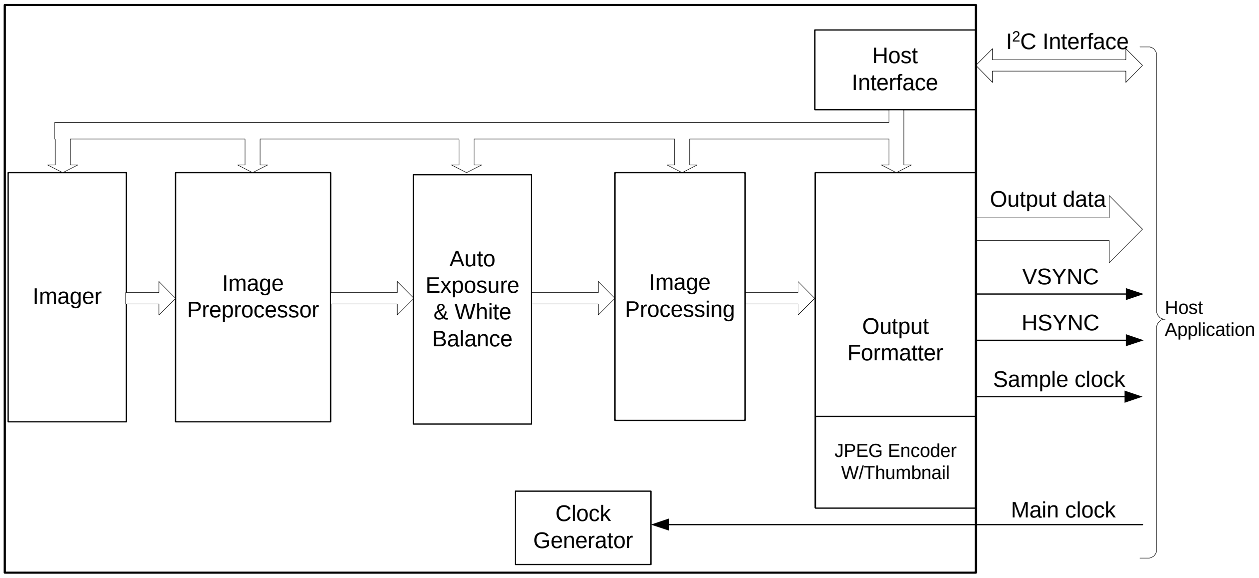


Figure 1. Bi8921 Sensor Internal Block Diagram

The host interface provides a common I^{2}C interface for the host application to access the internal read and write registers which control all other blocks.

The internal PLL allows the host to provide a single clock frequency to the Bi8921, yet still operate the imager at a wide range of other frequencies. The PLL can multiply the incoming clock by anywhere in a range of 8/16 to 255/16. With a front-end divider of up to 16 an additional divide-by-n final stage, a wide range can be achieved.

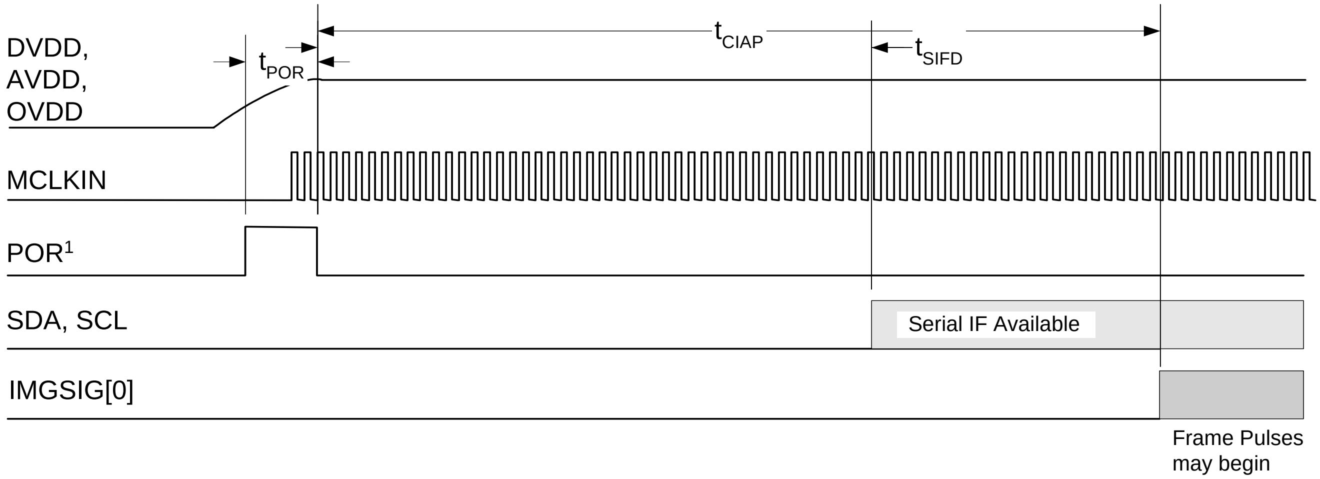
The four pipelined functions provide: the front end massaging and scaling of the input Bayer pattern data from the imager; the automatic exposure and white balance loop; the basic image processing to create the fully interpolated color image; the output format buffering, endian swapping, JPEG encoding and CCIR tag appending if required.

Power-Up Timing

CMOS devices are sensitive to the start up sequencing between the power supplies and clocks. All power supplies must be at operating level before the clock, in this case MCLKIN, becomes active. Do not present an active clock before the power is applied. The camera design must be sure to enforce this restriction.

The Bi8921 is equipped with an internal POR generation circuit in addition to being able to be reset externally. The internal POR is designed to create a pulse once the DVDD supply reaches the required level. The minimum duration is specified for an infinitely quick ramping supply. Supplies that ramp slowly will cause the pulse to be elongated. The internal POR is 'OR'ed with the external POR pin so that the longer duration of the two signals will control the reset circuit. There is no maximum for POR duration. In fact the POR may be generated with an RC network tied to OVDD or controlled from a host processor as well. POR should remain low for the remainder of the operating session. The serial interface is available at approximately 1 us after POR goes low. Figure 2 illustrates these details.

Symbol	Description	Min.	Typical	Max	Unit	Remark
t_{POR}	Power on reset active time	160			ns	
t_{SIFD}	Serial interface delay after POR	1			us	
t_{CIAP}	Capture inactive after POR	960000			MCLKINs	Should not cause readout before min t_{ADINIT} expires



[1] This is logical 'OR' of POR input pin, and internal reset circuit.

Figure 2. Sensor Power-Up Timing

Power Control

The Bi8921 has three levels of power consumption: Operating, Quiescent, and Standby. In Operating and Quiescent modes, the STANDBY pin must be held low, the device is powered up and all register values are retained. Individual functions are enabled or disabled according to the register settings. Operating mode means that images are being scanned out and processed. When not actually capturing images, the device automatically enters Quiescent mode to consume the minimum amount of power while maintaining the configuration registers.

The deepest level of power saving is called Standby mode. To active this mode the STANDBY pin should be brought to the high state. When this happens the device's internal regulators are disabled and all sections are completely disabled. Standby mode results in all register settings being lost.

Exiting the Standby mode is achieved by bringing the STANDBY pin back to a low state. When this occurs the internal automatic reset circuit will activate and reset the device. Registers will have to be re-initialized at that point.

Imager Subsection

The pixel array of the Bi8921 contains a total of 1632 x 1216 pixels, as shown in Figure 3. There are 24 columns of dark pixels on the left side and 8 rows of dark pixels along the top of the active pixel array. The dark pixels are the same physical cells used in the active array, except that they are shielded from light. They are scanned out just like the active pixels, and they provide a means to accurately determine the black level. Regardless of the any flip or mirror function, the dark lines are always scanned at the start of the image and dark columns are always at the start of a line. This leaves a 1608 x 1208 array of active pixels. The active imaging array is coated with a RGB primary color filter array and micro-lens in a Bayer pattern.

The Bi8921 supports a progressive scan pattern with the ability to mirror left-to-right and flip top-to-bottom. The imager data is read out row by row from the top of the image (bottom of the array) and from left to right within each row. Figure 3 depicts the row and column numbering with respect to the proper scene orientation for default, non-mirrored, non-flipped, progressive scan output. The first pixel of the scan is actually in the lower left of the die when viewed face-on in the normal orientation of up. This first pixel scanned out is labeled 0,0 in Figure 3.

Sub sampling is used to create a smaller readout image and thereby a faster readout time and higher possible frame rate. Sub sampling can be performed in both the horizontal and vertical directions and in combinations with each other. This provides 3 sub sampling modes that preserve aspect ratio. This permits fast (video rate) generation of images suitable for an LCD viewfinder or similar reduced resolution display.

A limited form of windowing is also supported to facilitate the implementation of digital zoom effects.

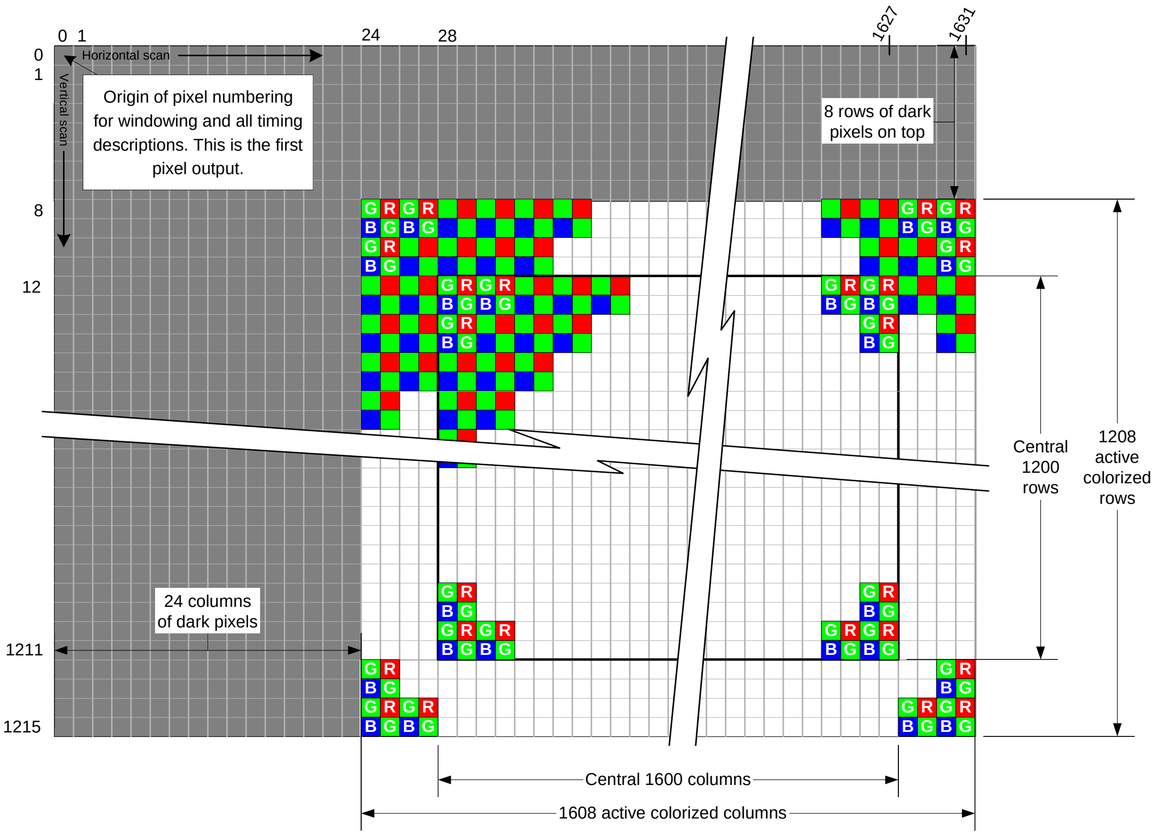

0 1
24
28
1627
1631
0
1
8
12
1211
1215
Horizontal scan
Vertical scan
Origin of pixel numbering for windowing and all timing descriptions. This is the first pixel output.
8 rows of dark pixels on top
G R G R
B G B G
Central 1200 rows
1208 active colorized rows
24 columns of dark pixels
Central 1600 columns
1608 active colorized columns

Figure 3. Color Filter Pattern and Pixel Addressing Convention

Electronic Shutter Modes

In electronic shutter mode there are two methods of exposing and reading out images. One method, called snapshot mode, consists of triggering a single exposure and readout. The other method, continuous mode, causes the imager to automatically repeat exposures and readouts. The methods are selected by the manner in which the FRMPLS bit of the IMGSIG register is controlled.

The FLSYNC signal is provided in both snapshot and continuous electronic shutter modes. It is however only truly useful for the snapshot condition, so detailed timing information is only provided for this condition, see figure 4. In order for the FLSYNC output to be generated, the exposure time must be programmed to take longer than the scan time of the sub sampled frame. In other words, the programmed exposure time in registers 0x2508, 0x2509 must represent more lines of imagery than the vertical frame size. This insures that at some point all the pixels in the frame are integrating simultaneously. The duration of this integration overlap is the difference in the frame size scan time and the exposure time. This is also the duration of the FLSYNC signal. Since the frame can be sub sampled in multiples of 2 lines and the exposure is in multiples of 1 lines, the minimum FLSYNC pulse will be 1 line long. If the exposure time is shorter than or equal to the vertical frame scan time, no FLSYNC pulse will be generated.

Snapshot Mode

In snapshot mode, a picture taking sequence begins with the FRMPLS bit low. Any chip parameters that require setting should be programmed at this point using the I^2C interface. At the conclusion of the programming sequence, the Update Required Command (URC) message should be sent. This will signal the imager to load all new register settings internally. The imager is now ready to expose and readout one image.

Generating a rising edge pulse on the FRMPLS bit triggers exposure. The FRMPLS bit is then left high long enough to insure the edge detection, but not too long. Immediate subsequent writes to the IMGSIG register are usually adequate. The imager will then begin the progressive expose and readout cycle. The exposure time programmed in register 0x2508/09 will be the time used, regardless of the duration of the FRMPLS bit. Leaving FRMPLS high past the t_{FPmax} duration will cause the imager to operate in continuous mode and an additional image will be generated.

After a period equal to approximately two line readout time plus the programmed exposure time has elapsed the readout sequence will begin. See Figure 4.

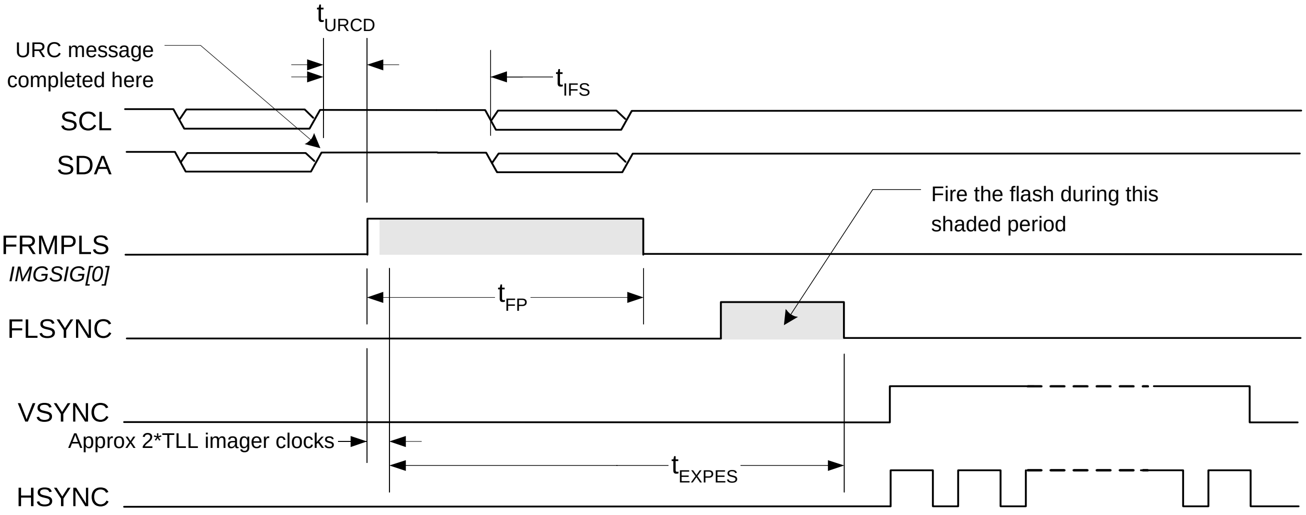

t_{URCD}
URC message completed here
t_{IFS}
SCL
SDA
Fire the flash during this shaded period
FRMPLS
IMGSIG[0]
t_{FP}
FLSYNC
VSYNC
Approx 2*TLL imager clocks
t_{EXPES}
HSYNC

Figure 4. Frame Timing for Electronic Shutter Mode – Snapshot Mode

Symbol	Description	Min.	Typical	Max	Unit	Remark
t_{URCD}	URC Message Delay to FRMPLS	8			clocks	
t_{IFS}	Inter-frame Serial Delay	3784			clocks	Time from URC to next parameter update
t_{EXPES}	Exposure Time	1	<= 1216	65,535	lines	Conversion to clocks depends on subsample and window mode and Hblank mode
t_{FP}	Frame Pulse Active	10		t_{EXPES}	clocks	Snap Shot
		>t_{EXPES}			clocks	Continuous

Continuous Mode

Continuous mode operation is intended for a video camera type application or LCD viewfinder support. Once the imager parameters are established, a continuous stream of image output can be initiated and sustained simply by setting the FRMPLS bit high and holding it there.

Continuous operation starts out the same as snapshot mode. The I^2C interface is first used to set any gain, sub sampling, or exposure time parameters. The sequence is finished with a URC message. After waiting a minimum of t_{URCD}, the FRMPLS signal can be brought high. Readout of the first image will then start at the same time as in the snapshot case. In the continuous mode case however, the FRMPLS will now be kept high past the expiration of the exposure time. This signals the timing generator that another image is needed and the exposure and readout cycle will be repeated. This will continue as long as FRMPLS continues to be high at the end of the exposure cycle. To end continuous mode at a specific time, i.e. get just one more image, the FRMPLS signal should be set low while VSYNC is low. This will produce one more readout. See Figure 5.

The imager parameters may be updated during continuous mode. This allows dynamically changing the gain or exposure settings without interrupting the train of readouts. To do this the new settings can be programmed over the I^2C interface while the VSYNC signal is high. When this constraint is met, the effect of the settings will appear in the second image (second VSYNC high) after the one in which the settings are changed.

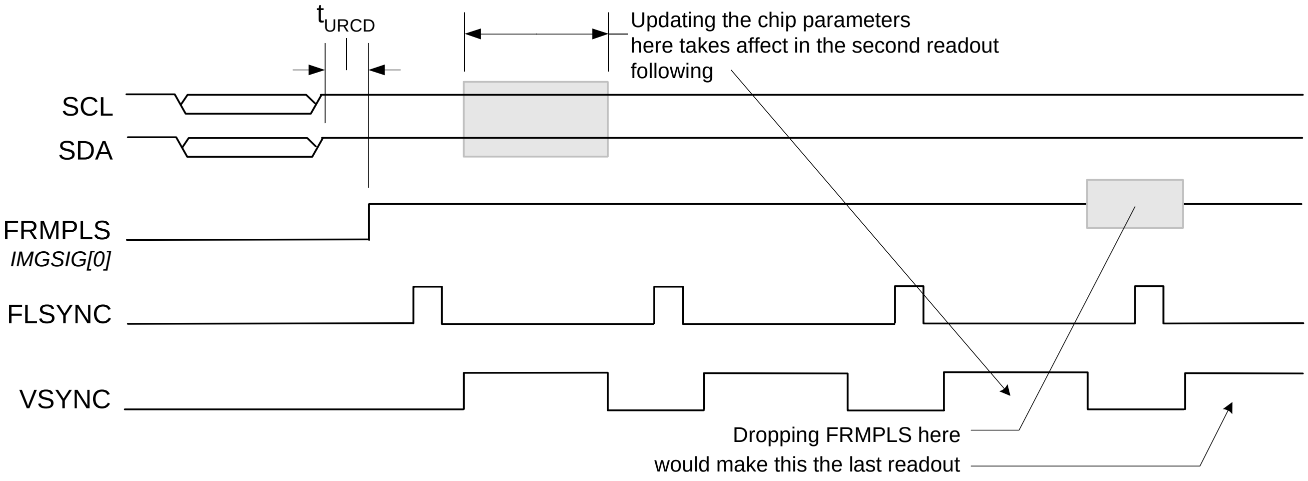

t_{URCD}
Updating the chip parameters here takes affect in the second readout following
SCL
SDA
FRMPLS
IMGSIG[0]
FLSYNC
VSYNC
Dropping FRMPLS here would make this the last readout

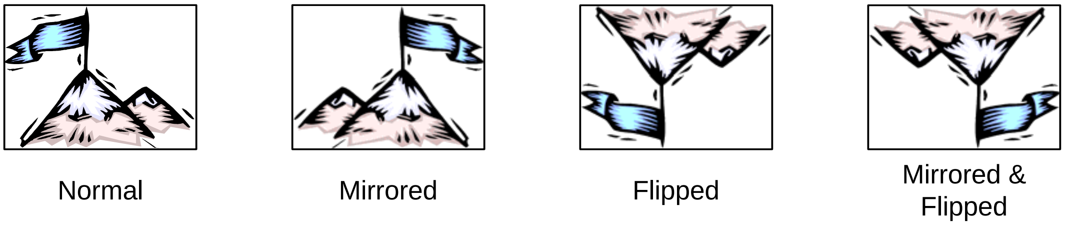
Figure 5. Frame Timing for Electronic Shutter Mode – Continuous Operation

Since the exposure time and vertical frame size can be programmed independently, they will control the maximum readout frequency and the duty cycle of the VSYNC signal at different times. When the exposure time is shorter than or equal to the readout time, the maximum readout frequency will be limited to the reciprocal of the Effective Vertical Size plus one line time. VSYNC will be low for only two line times. When the exposure time is greater than the readout time, the maximum readout frequency will be limited to the reciprocal of the exposure time. VSYNC will be low for the difference of exposure time less the readout time.

Imager Configuration

Mirroring and Flipping

The Bi8921 provides independent control of the image mirror and flip functions. The mirroring function will cause the right side of the image to be scanned out first. This swaps the image left for right, but does nothing to the vertical orientation of scene features. The flipping function will turn the image upside-down, but leave left and right as is. These functions may be selected individually or simultaneously, providing a total of four read out patterns. Figure 6 illustrates these.



Normal

Mirrored

Flipped

Mirrored & Flipped

Figure 6. Effects of Mirroring and Flipping

Exposure Time Control

Register addresses: 0x2508 and 0x2509

The exposure time is normally controlled by the Auto-exposure function, but it must be accessed directly in the case of using a flash strobe light. The explanation below is provided to aid the calculation of exposure time in seconds and the flicker avoidance interval.

The pixel exposure time is specified in terms of multiples of 1 row times (register EXPMSB, EPXLSB). The default on power-up is equal to one complete frame readout time. The available resolution will of course vary with the pixel clock provided.

The LSbit resolution can be calculated as: LSbit ms = (Total_Line_Length * 1000) / (ICLK Freq Hz)

Where: -Clock Freq is that which is provided to the Imager block

-Total_Line_Length includes blanking clocks and is a function of the sub sampling mode. windowing mode and horizontal blanking adjust. Without windowing, and with the default Horizontal blank, the line lengths are as follows.

In sub sample modes	0: Total_Line_Length = 1920
	1: Total_Line_Length = 1104
	2: Total_Line_Length = 696

In an environment with fluorescent lighting, which inherently modulates due to the alternating current power supply, the imager will most likely present a banding effect in the captured images. This is caused by the mismatch in timing between the imager and the light source. It is possible to select exposure times that avoid this problem. By using the proper exposure time, it is possible to insure that all rows of detectors see the same number of cycles of the light source modulation.

The modulation of the fluorescent lights is twice the power supply frequency (either 50 or 60 Hz). So the light modulates at 100 or 120 Hz. The proper exposure time to avoid flicker would be any time that results in exposing for an integer multiple of the light modulation period. Since Biomorphic image sensors define exposure time in terms of detector row scan times, the equations for selecting the flicker-free exposure time are a function of the imager pixel clock rate and the exposure time resolution. Since the Auto-exposure function controls exposure time to only use multiples of the flicker-free setting, the user need only program this step size into the FLICKT register

The general equation for flicker-free exposure time, T_{exp} (seconds) is given below:

$$T_{\exp} = N\left(\frac{1}{2 * F_L}\right)$$

Where: N is any positive, non-zero integer

F_L is the alternating current frequency of the light source in Hz.

The value programmed into the FLICKT register of the sensor can then be solved for by dividing the expression for 'LSbit ms' into the desired T_{exp} in milliseconds. Set N equal to 1 to get the increment size.

$$ExposureIncrement = \frac{\left(\frac{1}{2 * F_L}\right) * 1000}{\left[(TLL * 1000)/F_{PCLKI}\right]} = \frac{F_{PCLKI}}{\left[2 * F_L * TLL\right]}$$

Channel Independent Gain Control

Register addresses: 0x2501 and 0x2503

The first stage of analog gain control is applied in the imager block. These gains can be 1x, 2x, 4x, or 8x. There are two separate gains: one for the combined red and blue channels; and one for the two green channels. The control of these gains is normally performed through the Auto-exposure function and is always written as the same value. They are accessible directly as well and can be controlled independently if desired.

Dark Level Correction

Register addresses: 0x251A - 0x251c, 0x2523 - 0x2525, and 0x252A – 0x252F

The Dark Level Correction performed in the analog domain within the image block is divided into a pre-gain phase and a post-gain phase. The automatic process adjusts offsets, applied to the analog pixel value, to get the dark pixels at the top of the image to match a 'target' value specified by the application. These offsets are applied to all pixels within the frame. The algorithm used by this block will try to use only the post-gain correction to reach the target before using any pre-gain correction.

The offsets can be thought of as subtractive terms, so a sign bit of '1', results in a greater dark level value.

It is best to follow Biomorphic's recommendations in setting the registers of this function.

Sub Sampling Readout Control

Register addresses: 0x250A

Sub sampling allows a set of non-contiguous pixels that cover the full array to be scanned. Sub sampling in the imager block allows faster readouts than scanning a full image, thereby allowing the clock to be slower for a given framerate and reducing system processing and power loads in general. There are three modes of sub sampling, which can be controlled independently for the horizontal and vertical directions. This would create 9 possible combinations of which only 3 would preserve the image aspect ratio. Therefore only those three are listed below and referenced in this document. Their meanings are defined in the table below. The default setting is to perform no sub sampling so that all pixels are read out.

Sub sampling is performed by skipping lines and/or columns in the output. However, the 8 dark rows at the top of the frame are not sub sampled. The "Read 2 / Skip 2" designation means to output 2 lines (or columns) and then not output 2, effectively reducing 2:1. "Read 2 / Skip 6" means to output 2 and not output 6, making a 4:1 reduction.

Sub Sample Mode	Interpretation	Internal image size without dark rows, and without windowing (colsxrows)
0	No sub sampling, all pixels output in Bayer pattern (default)	1632 x 1208
1	Vertical Read 2 / Skip 2 and Horizontal Read 2 / Skip 2 (2:1)	816 x 604
2	Vertical Read 2 / Skip 6 and Horizontal Read 2 / Skip 6 (4:1)	408 x 302

Windowing Readout Control

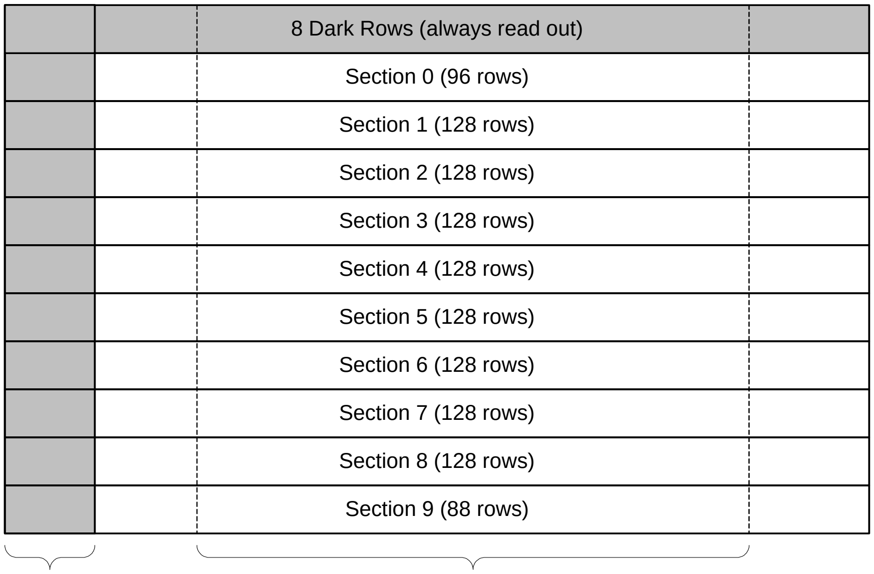
Register addresses: 0x250E

Limited windowing ability is provided in the image readout function. This is separate from the windowing available in the Trimmer function, which happens at a later stage. Unlike the trimmer, this windowing function will allow the internal readout time of the image to shorten and allow a faster possible frame rate. Both horizontal and vertical windowing is provided. The horizontal windowing will shorten the line time.

To achieve the vertical windowing, the image array is divided into 10 sections. The window is defined by programming a start and stop section. To get just section 1, you would select ‘start at section 1’ and ‘end at section 2’. The values used to program the start and end sections in register 0x250E, change as a function of the vertical flip setting. Normal operation is no windowing. When windowing is enabled, sections 0 and 9 are not available.

The horizontal window allows only two selections, full size, and the central 1288 columns plus the dark columns.

Sub sampling can be used in addition to windowing.



Horizontal window reads out dark columns and
central 1288 columns

Figure 7. Windowing Definition

Processor Configuration

Image Preprocessor

The Image Preprocessor stage, accepts digital pixel data from the imager and performs some pre-processing before passing it along to the Image Processing stage. All of the processing steps can be bypassed, or programmed to have no effect.

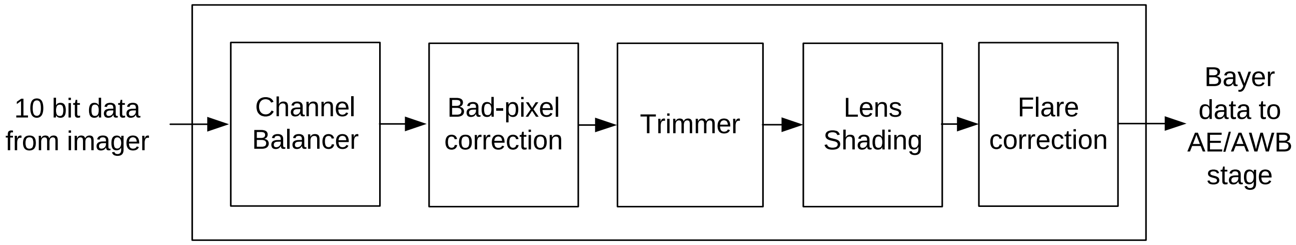


Figure 8. The Imager Interface Stage

Channel Balancer

Register addresses: 0x2110 thru 0x2117

The Channel Balancer provides an equalization of the base dark level of all pixels in either one of two automatic modes or a manual mode. The correction is to subtract an offset term from the intensity of each pixel. The terms may be input by the application host processor or may be generated in real time by one of two ways.

In method one, the top two dark rows are sampled and averaged to generate the terms that are applied to the entire picture. Eight separate terms are maintained. The averages may be generated by using 16, 32, or 64 total samples for each channel. Figure 9 shows the channel labeling arrangements relative to the Bayer color filter pattern.

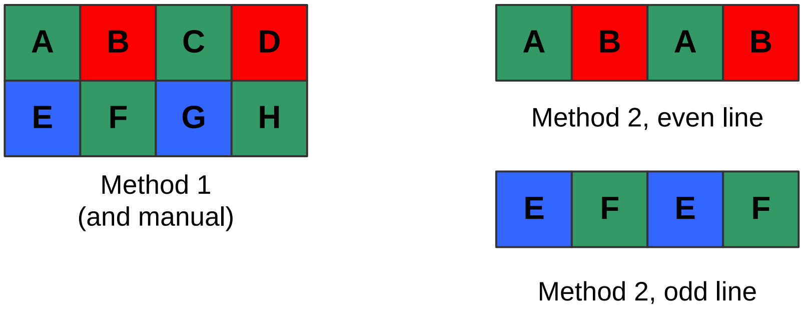


Figure 9. Channel to Color Mapping

In method two, the dark columns at the left of each row are sampled to generate terms for that row only. New terms are generated for each row. In this case, only two terms (one for each color) are generated for each row. Only 8, 16, or 32 samples may be used for the term average generation. This requires a dark column configuration at least as wide as the programmed sample number.

When manual loading of the terms is used, they are interpreted the same as for method 1. Manually loaded values are restricted to 0 to 127, but these are in the 10 bit domain of the input data.

Bad Pixel Correction

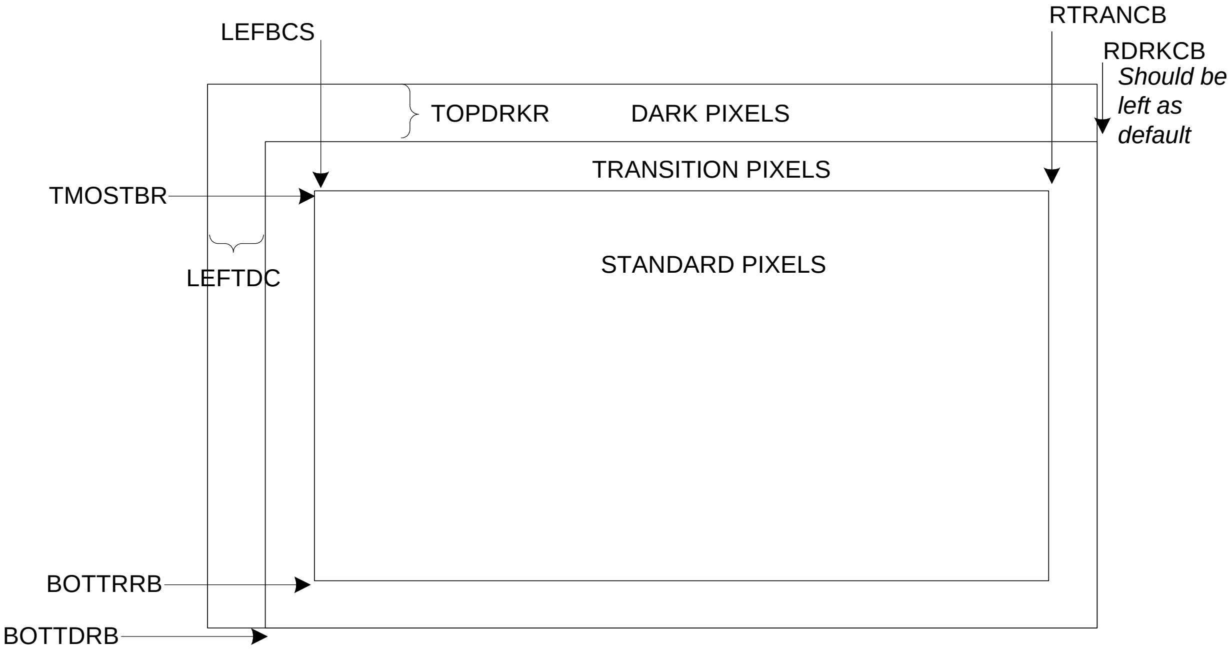
Register addresses: 0x2080 thru 0x208E

Bad Pixel Correction is the process by which lone pixels that are responding very differently than their neighboring pixels are replaced with values based on their neighbors. The three colors are processed independently each with its own threshold for what is determined as ‘responding very differently’. These thresholds are applied for both an ‘over responding’ (bright) and ‘under responding’ (dark) test. Both tests can be enabled/disabled independently. Thresholds are specified in the 10 bit input data range of 0 to 1023. Larger values will generally replace fewer pixels, and smaller thresholds will cause more pixels to be replaced. The number of pixels replaced may be read back, but it saturates at 1023.

Trimmer

Register addresses: 0x2010 thru 0x201F, 0x2038

The Trimmer allows the optional removal from the data stream of the dark pixels and the transition pixels. Dark pixels may be stripped alone, or both dark and transition may be stripped. By programming large values for the transition/standard pixel boundaries, windowing can be performed. The input image data can be thought of as consisting of three zones of pixels: dark, transition, and standard. Dark pixels are those that are electrically active but masked from light. Transition pixels are those between the dark pixels and the standard pixels or those that define the ‘standard’ imager array. This function must be programmed with the number of and starting row/column addresses of the dark pixels and standard pixels in the input stream. The rising or falling edge of the sample clock may be used.



Should be set equal to BOTTRRB if two pulse VSYNC method is desired.
Otherwise set to 1216.

Figure 10. Meanings of Trimmer Registers Relative to Image Format

If the two pulse VSYNC output style is needed, then the BOTTDRB register should be set to the same value as BOTTRRB and the BOTDR count should be greater than 2 (default is 3) even though there are no dark rows on the bottom. If the second pulse is not needed in the pulsed VSYNC method, then set BOTTDRB to a number equal or greater than the number of rows in the read out. There are also no right side dark columns, so the RDRKCB register can be left as the default value.

These registers define the frame relative to the imager block output. Therefore when sub sampling is enabled, these must be adjusted.

Lens Shading

Register addresses: 0x2300 thru 0x2391

Lens shading correction is based upon a piece-wise linear interpolation of gains. The table contains gains to be applied for every 64th pixel based upon that pixel's radial distance from the 'center pixel'. The pixel that coincides with the lens axis is called 'center pixel'. Ideally, this is the center of the imager array, but this may be programmed to optimize the performance of the correction and adjust for slight manufacturing skews.

All the gains are 12 bit values, with 1024 representing a gain of 1.0x. This allows a range of 1.0 to 3 $^{1023}/_{1024}$. There is a separate transfer function for each of the color channels. The first point in the transfer function is at 64 pixels from the center. The gain at the center is assumed to be 1.0x. Points are defined in steps of 64 pixels from there on out for a total of 17 intervals. The defined curve must be monotonically increasing. Distances between points in the function use adjacent gain entries and interpolate between them

The user defines the center pixel (LRCEN, LCCEN) of the array. These are specified in pixel units for the array's full readout size. The user also defines the initial square root (ISQRIN) and the square root error (ISRERR). These values are necessary for the correct computation of the radial distances of all the pixels.

If any sub-sampling is done in the imager, it needs to be specified in register ISSCON. As during the sub-sampling, the physical location of the pixel remains the same; sub-sampling factor needs to be taken into account while determining the radial distance for a pixel.

Flare Correction

Register addresses: 0x2290 thru 0x2298

The flare correction function subtracts a correction term to each channel's pixel data. Four separate terms can be specified, either positive or negative. Negative values are specified by setting the sign bit to 1. Because the terms are subtracted, a negative value will end up increasing the pixel's level. The four channels are assigned as:

1	2
3	4

Where the 4 pixel grid maps to a Bayer pattern quad. However, depending on how the Trimmer is programmed, the pattern will start in one of four ways (see figure 12). So if different values are being used for different colors, the channel assignments must match the actual color pattern.

Imager Control Interface

Register addresses: 0x2220, 0x2222, 0x2228, and 0x2229.

The Imager Control Interface provides two major functions: access to certain discrete imager control signals; and a feed through for read and write access to the imager 3-wire serial interface. However, accessing the Imager block through the 0x25xx register space is recommended.

One discrete output to the imager and one input are accessible in the IMGSIG register (0x2220). Bit 0 is used to control the image capture initiation (FMRPLS signal). Bit 6 is used to monitor the status of the FLSYNC signal which indicates the flash strobe firing window. Register 0x2228 can be used to encode the FLSYNC signal into the HSYNC.

Automatic Exposure and White Balance

The Automatic Exposure (AE) and White Balance (AWB) is nominally used as an autonomously controlled function. When operated in a video mode, image data is sampled, adjustments are calculated and the settings are updated in a continuous fashion to converge and maintain the exposure and white balance functions at the proper levels. These same components can however be used in a manual mode, by breaking the loop and having the host application control the settings.

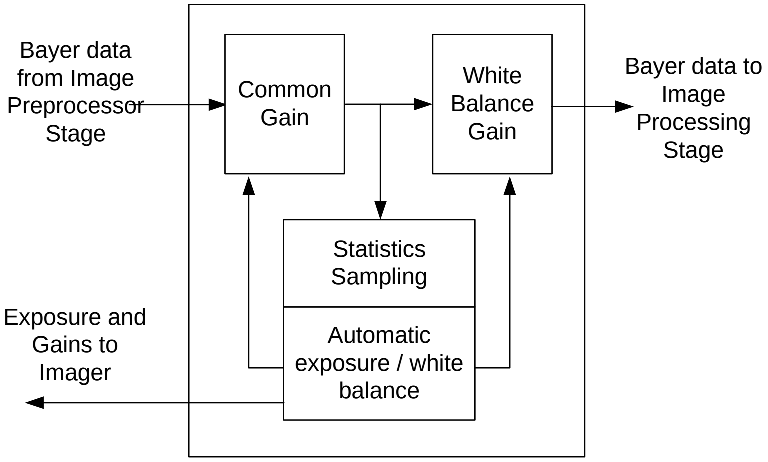

Bayer data from Image Preprocessor Stage
Common Gain
White Balance Gain
Bayer data to Image Processing Stage
Statistics Sampling
Automatic exposure / white balance
Exposure and Gains to Imager

Figure 11. The Automatic Exposure and Auto White Balance Stage

Statistics Sampling

Register addresses: 0x2041 thru 0x2049, 0x204D thru 0x2050, and 0x206E thru 0x207A

The Statistics Sampling block is responsible for collecting information on the image intensity statistics segregated by color. For each of the four, color channels, the average intensity is calculated. The separate color data is used in the automatic white balance (AWB) process, while a combination of the colors, as a derived Y intensity value, is used in the automatic exposure (AE) process. The Y value represents an average luminance and is calculated from the color averages as $Y = 0.299\ R + 0.587\ (G1+G2)/2 + 0.114\ B$. The AE process can optionally be programmed to use the highest average of the four color averages instead of the Y value.

Statistics are sampled on the upper 8 bits of the Bayer data, after the application of the Common Gain. Sampling is performed over one or both of two rectangular regions defined by the eight registers at 2041h thru 2048h and 206Eh thru 2075h. These registers define a starting row and column as counted from any remaining pixels and lines after the trimming process and a window size in number of rows and columns. The first row/column is number 0. All eight of these registers are 11 bits. The dimensions of the windows should both be multiples of 2 to insure that equal numbers of all colors are sampled. By default, only window 1 is sampled and used by the AE function. The second window must be explicitly enabled (AEWCON[3]) and its statistics are not used by the internal AE loop. They are however available to the AWB loop. W2SUBS[2] controls whether window 1 is used for both AE and AWB, or whether AWB is run on window 2 statistics.

Statistic sampling may be sub-sampled to reduce power consumption. In some cases, it may be adequate to only sample a portion of the image data for the intensity information. There are four selection for sub-sampling: none (all pixels); read 2 / skip 2; read 2 / skip 6; and read 2 / skip 14. This notation means to sample 2 and then skip some number of pixels in both dimensions. These result in sampling 1/4, 1/16, and 1/64 of the image area. The sub sampling is controlled separately for the two windows. Sub sampling must be utilized if the statistics window is greater than a total of 1280x1024 pixels.

AE / AWB Processing

Register addresses: 0x2040, 0x204A thru 0x204C, 0x2051 thru 0x2068, 0x206A thru 0x206D, 0x207B thru 0x207F

This block performs both the automatic exposure (AE) and automatic white balance (AWB) processing. It is intended to be operated during video mode processing when there is a continuous stream of images. This block may be bypassed, in which case the host application processor must determine and set all relevant imager registers for proper exposure. This block can also be used in a compute-only mode, where terms are generated but not applied.

AE operates by trying to keep the average luminance (Y), or maximum color average, at a user specified target value, within some tolerance band. The target value is set as an 8 bit value and is defaulted to mid-scale. The tolerance band is given by a single number, also in 8 bits and is symmetric about the Y target value. The imager exposure, imager gain, and common gain are all determined in unison. The imager exposure setting is then constrained by the maximum exposure number in MEXPT and the flicker cycle time in FLICKT. Both MEXPT and FLICKT are set by the host application processor and must be in the same units as the exposure time is specified in the imager being used. Note, that since imager exposure time is sensitive to the clock frequency and sub sampling modes, these must also be calculated the same way.

Once the constrained exposure value is determined, the gains are used to make up any difference between the ideal exposure and constrained exposure. Imager gains are used in steps of 1x, 2x, 4x, and 8x. So the total gain is truncated to one of these and programmed to the imager. Any truncated portion of the gain is accounted for in the Common Gain block.

Automatic white balance is performed in a similar manner, except that the imager registers are not used for the adjustment. White balance is simply a feed-forward process, where adjustments are made to the white balance gains based on the imager data. In this case, the maximum of the separate color averages is determined and is used as a reference. The color with a maximum value gets a gain of 1x. Ratios of the other colors are then computed to create the gain value needed to normalize all the colors to the reference color. If the maximum color average is less than the value in the WBMAVG register, then no white balance adjustment is made.

The rate at which the AE and AWB settings are updated may be controlled by the host processor. A number between 1 and 7 is set in the CFREQ register and represents the number of frames in one update cycle. If the default setting of 1 is used, then it is recommended that the AEWCON[2] bit also be kept at its default setting of 1. This prevents oscillations.

Another feature for auto exposure is setting the steps by which the exposure time will increase. If this value is set to 0, there is no restriction on the exposure time increase/decrease. If a value is set for that register (EXPSTPS) other than zero, combined exposure time (sensor exposure time * sensor gain * common gain) can only increase/decrease by this amount.

The AEWCON[4] can be set to force a fixed exposure time, thereby enforcing a fixed frame rate. In this case, image brightness is controlled only with gain. The AEWCON[6] bit will be set by the AE loop if the algorithm detects it is fighting a high fixed exposure time. The host might monitor this bit and take appropriate action.

All of the gains and exposure values computed in both AE and AWB, are available in read-only registers. This allows the host application processor to monitor or even modify the process. An example of that case would be when running video at one clock or resolution and taking a still picture at a different clock speed or resolution. Shifting clock speed and/or resolution requires the re-computation of the exposure register value and gain settings. But, the converged value arrived at in video mode can be taken advantage of.

Registers 0x2055 and 0x2056 are used to indicate the addresses in the imager register set for the sensor (analog) gain settings. These represent the offsets from 0x2500 for the red/blue gain and the green/green gain respectively. These default to the proper numbers and do not require programming. Registers 0x2059 thru 0x205C are used to indicate the value to program into sensor gain register to achieve gains of 1x, 2x, 4x, and 8x. The AE block will write these values to the addresses indicated by 0x2055/56 as needed. Registers 0x205F and 0x2060 represent the sensor block register offset from 0x2500 for the Exposure time setting from AE.

Common Gain

Register addresses: 0x2069

The Common Gain block is used to fill in gain steps, not achievable by the imager. When the auto-exposure process computes a gain other than 1, 2, 4, or 8, the imager is programmed to the next lowest setting out of 1, 2, 4, or 8 and the residual gain component is set into the Common Gain block. The gain is implemented as a digital multiplier. A single term is used on all pixels, regardless of color. The multiplier term is a 7 bit number with an LSB of 1/64th . This provides a range of 0 to 1.984 ($1^{63}/_{64}$).

White Balance Gain

Register addresses: 0x20F0 thru 0x20F4

The White Balance Gain block provides four digital gain multipliers. There is one for each color channel. The green pixels from 'red' rows and 'blue' rows are treated separately, though they frequently have the same programming. This block may be bypassed, programmed automatically by the AE/AWB processing, or set externally by the host application processor. All four gain channels accept an 8-bit multiplier with an LSB of 1/64th. This provides a gain range of 0 to 3.984 ($3^{63}/_{64}$).

Image Processing

The image processing function is responsible for making the viewable image from the pre-processor/corrected input image. The process includes: interpolation to create the full color image; color correction to adjust for color cross-talk; gamma correction and saturation to make the colors more pleasing to the eye; color space conversion; and finally edge enhancement for a crisp image. These blocks are all programmable by the host application through the I^{2}C interface. Each block can be bypassed individually if unneeded.

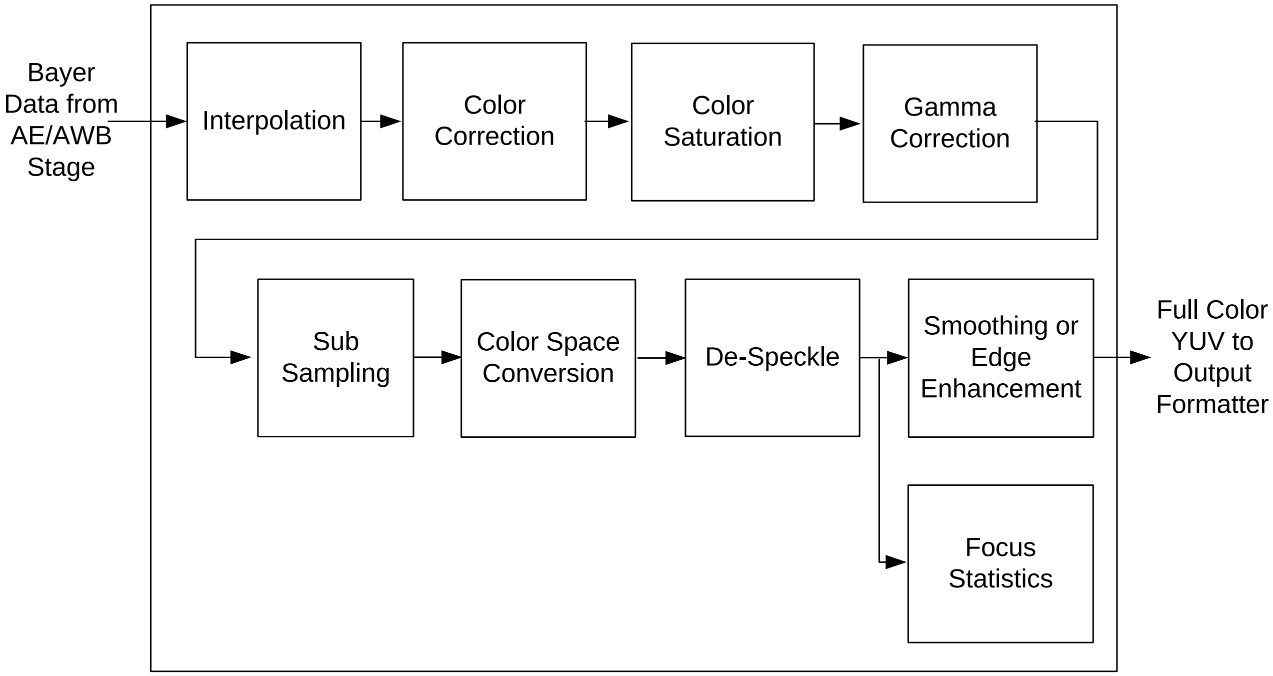
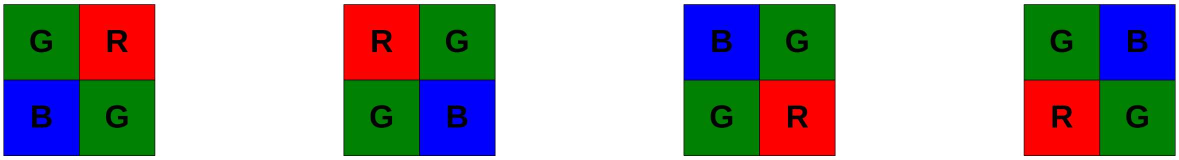


Figure 12. The Image Processing Stage

Interpolation

Register addresses: 0x2090

The Interpolation function processes the Bayer pattern data with one color per pixel into full RGB data with three colors per pixel. There is limited programmability of this function. The application host must specify how the bayer data begins in order for the process to get the sequence correct. Figure 13 shows the choices and values of the IPCON register bits 2 and 1. Since this process actually consumes a one-pixel border all the way around the image, it is advantageous to define the transition pixels in the Pixel Tagger block to pass an image two rows and two lines larger than the final desired result. In other words, to get a 1280x1024 final image, define the standard pixels to be 1282x1026, then set bit 3 of IPCON to trim the extra pixels after interpolation. If the trimming step is not performed, then the output will be the size of the input, but the one pixel border will be replications of the next pixel in. The entire process can be bypassed if raw Bayer data is the desired output.



IPCON[2..1] = 00 IPCON[2..1] = 10 IPCON[2..1] = 01 IPCON[2..1] = 11

Figure 13. Bayer Data Starting Patterns

Color Correction

Register addresses: 0x20A0 thru 0x20A9

Color correction is used to correct for overlaps in the spectral sensitivities of the image sensor pixels. Before color correction, the interpolated pixel data will produce inaccurate colors due to the overlap in the color spectrum of the three color filters.

Color correction relies on the selection of a "Color Correction Matrix." This matrix defines a transform that adjusts each color in an interpolated pixel with respect to the other colors. The entries in the 3x3 transform matrix are the ones that the application host processor must define. Each entry is 7 bits and is scaled by 32. The available range is therefore 0 to 3 $^{31}/_{32}$. While all entries are made as positive values, the terms on the diagonal (CCM11, CCM22, CCM33) are always applied as positive values and the off diagonal terms are always applied as negative values. the sum of the terms in any given row should be $^{32}/_{32nds}$. The power on default settings define a unity transform. This entire block may be bypassed as well.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} \times \begin{bmatrix} +CCM11 & -CCM12 & -CCM13 \\ -CCM21 & +CCM22 & -CCM23 \\ -CCM31 & -CCM32 & +CCM33 \end{bmatrix} = \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

Color Saturation

Register addresses: 0x20D0 thru 0x20D1

The Saturation function enhances the colors in the image by applying a gain to the saturation value of the pixel in HSV space. The gain is specified in steps of $^{1}/_{128}$ths in the 7-bit SATVAL register. The sign bit should be set to achieve desaturation (gains less than 1). For saturation values, the gain applied is 1 + (SATVAL[6:0]/128) for a range of 1 to 1.992. For desaturation terms, the applied gain is SATVAL[6:0]/128. The power on reset default setting is 0 for both sign and value, for a gain of 1.0. This function can be bypassed. The table below illustrates the mapping

	Desaturation (less color)		Saturation (more color)	
Applied Value	0.0	0.992	1.0	1.992
Register Value	0x80	0xFF	0x00	0x7F

Gamma Correction

Register addresses: 0x23B0 thru 0x23BF

The Gamma Correction block applies a gain transform to each of the three color components of the interpolated RGB data. It is implemented as a 17 point, piecewise linear interpolation. The coordinates of the first and last blocks are assumed to be 0,0 and 255,255, so only 15 points are actually specified. In order to minimize error relative to the ideal curves, the first two intervals are only 8 points apart and the last interval is 32 points long. A single transform is used for all three colors. Though this block is meant for applying the gamma correction, it can as well be used to implement any transfer function by loading the table as desired. The function may be bypassed completely. In all cases, the function must be monotonically increasing.

After a power on reset, the table is loaded with the function points corresponding to a gamma value of 1.9. The following chart shows the default-loaded curve.

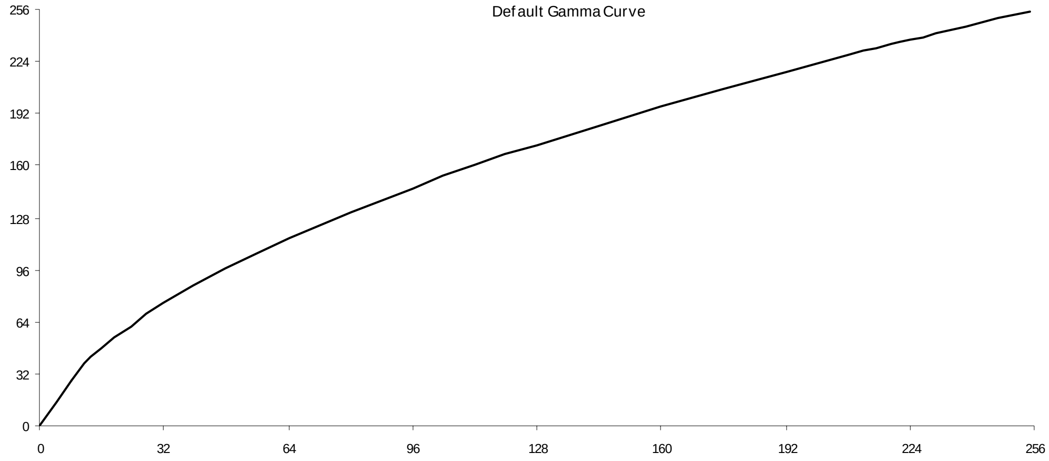


Figure 14. Default Gamma Curve After Reset

Sub-Sampling

Register addresses: 0x2230 thru 0x2233

The sub-sampling function provides a means to down sample the interpolated image in one of several pre-defined patterns. Whereas the imager sub-sampling is simply discarding samples, and is restricted to maintaining a valid Bayer pattern, this second sub-sampler sometimes performs horizontal averaging on fully reconstructed RGB data. This has the advantage of maintaining more scene detail and edge sharpness. A combination of vertical sub sampling and horizontal averaging is used to create the scaling ratios shown in the table below. There are actually two sub sampler stages that operate in sequence, each with the full set of 7 modes. This allows many combinations.

Mode (Reg.s 0x2231 & 0x2233)	Vertical Scale Factor	Horizontal Scale Factor
0	1/2	1/2
1	1/4	1/4
2	3/5	11/20
3	3/10	11/40
4	4/5	4/5
5	2/3	2/3
6	3/4	3/4

Color Space Conversion

Register addresses: 0x20C0 thru 0x20C2 and 0x2408 thru 0x241E

The Color Space Conversion function is actually divided, with some steps happening before the Edge Enhancement and some steps after. Edge Enhancement requires that the data be in either YUV or YCbCr space, so this happens before the sharpening. Since the output options include RGB, the conversion from to RGB takes place after the Edge Enhancement. The default mode of operation is to create YUV, full range data, since that is what is needed for JPEG compression. That is, output values will range from 0-255. YCbCr, limited range, format can also be generated from YUV, but should not be used for JPEG. If RGB is chosen, there is a choice of 565 or 444 formats as well. Conversion from YUV to YCbCr or to RGB assumes that the matrix values (regs 0x2408 to 0x241e) are programmed to make YUV.

All three conversions can be bypassed individually. In fact the YUV to RGB and YUV to YUV conversions should never both be selected.

Following formulae are used for format conversions.

Y = RTYYR*r + RTYYG*g + RTYYB*b + RTYYO
U = RTYCBR*r + RTYCBG*g + RTYCBB*b + RTYCBO
V = RTYCRR*r + RTYCRG*g + RTYCRB*b + RTYCRO

Where: r, g, and b are the pixel’s interpolated RGB values and the multipliers and offsets are as below:

For YUV output (full range 0-255)				
Y	RTYYR = 0x04D	RTYYG = 0x096	RTYYB = 0x01D	RTYYO = 0x00
U	RTYCBR = 0x12B	RTYCBG = 0x155	RTYCBB = 0x080	RTYCBO = 0x80
V	RTYCRR = 0x080	RTYCRG = 0x16B	RTYCRB = 0x115	RTYCRO = 0x80
For YCbCr output (limited range)				
Y (16-244)	RTYYR = 0x041	RTYYG = 0x081	RTYYB = 0x019	RTYYO = 0x16
Cb (16-235)	RTYCBR = 0x125	RTYCBG = 0x14A	RTYCBB = 0x070	RTYCBO = 0x80
Cr (16-235)	RTYCRR = 0x070	RTYCRG = 0x15D	RTYCRB = 0x112	RTYCRO = 0x80
For Negative Image effect in YUV				
Y	RTYYR = 0x14D	RTYYG = 0x196	RTYYB = 0x11D	RTYYO = 0xFF
U	RTYCBR = 0x02B	RTYCBG = 0x055	RTYCBB = 0x180	RTYCBO = 0x80
V	RTYCRR = 0x180	RTYCRG = 0x06B	RTYCRB = 0x015	RTYCRO = 0x80
For Black and White effect in YUV				
Y	RTYYR = 0x04D	RTYYG = 0x096	RTYYB = 0x01D	RTYYO = 0x00
U	RTYCBR = 0x000	RTYCBG = 0x000	RTYCBB = 0x000	RTYCBO = 0x80
V	RTYCRR = 0x000	RTYCRG = 0x000	RTYCRB = 0x000	RTYCRO = 0x80
For Sepia effect in YUV				
Y	RTYYR = 0x04D	RTYYG = 0x096	RTYYB = 0x01D	RTYYO = 0x00
U	RTYCBR = 0x2B	RTYCBG = 0x000	RTYCBB = 0x000	RTYCBO = 0x59
V	RTYCRR = 0x180	RTYCRG = 0x000	RTYCRB = 0x000	RTYCRO = 0x98

De-Speckle

Register addresses: 0x22A0

The De-speckle function provides a low pass filter operation on the three channels of Y, U, and V (or YCbCr). This provides a noise and false color reduction option in low light environments. This can be applied for Y and UV (CbCr) channels independently. Two neighborhood modes are available. The filter operator can take advantage of extra border pixels or not. If extra border pixels are to be used and then trimmed, the Trimmer block should be programmed accordingly to output a suitably larger (by 1 pixel all around). The De-speckle block can be bypassed entirely, and should be if the Color Correction phase is not outputting YUV (or YCbCr).

Edge Enhancement

Register addresses: 0x2240 thru 0x2244 and 0x226D thru 0x226F

The Edge Enhancement function provides a means to sharpen edges attenuated in other processing steps. At any given pixel location, the Y component of the YUV (YCbCr) data is evaluated for edge strength and a term is added to the Y component as a function of the edge magnitude. The additive term, or edge boost, is derived from a programmable transfer function. The edge magnitude at a given pixel is calculated from the neighboring pixels.

The enhanced pixel Y component value is then:

Y new = Y+ transfer function (edge magnitude)

The transfer function is assumed to have the shape as defined in figure 15. The transfer function is accessed by edge magnitude. The resulting edge boost is found by tracing edge magnitude up to the intersection of the bold transfer function line, and then across to where it intersects the vertical axis. The resulting boost value is then added to the Y component of the pixel in question. Operation on the negative side results in the boost value decreasing the Y component.

The exact break points are programmed by the host application processor. The function is symmetric about 0 edge magnitude, so only the positive side points need be defined. The lower end at which any edge enhancement begins is defined by point 1 (register ESHTLO). This also defines a dead band where edges are not modified. The maximum boost that may be used is set in the ESHPK register. The ramps leading up-to and down-from the peak boost level are defined by giving a slope value in register ESHSLP. The slope is specified with an LSB of ¼ and range of 0-15 (or 0 to 3.75). The ramps are symmetric about a point that must be defined in the ESHTHI register. The slope should be calculated as the peak boost divided by the difference of edge magnitudes at points 1 and 2.

At power on reset, the default settings define a straight line of slope 0 along the horizontal axis. This effectively performs no enhancement. The entire block may also be bypassed.

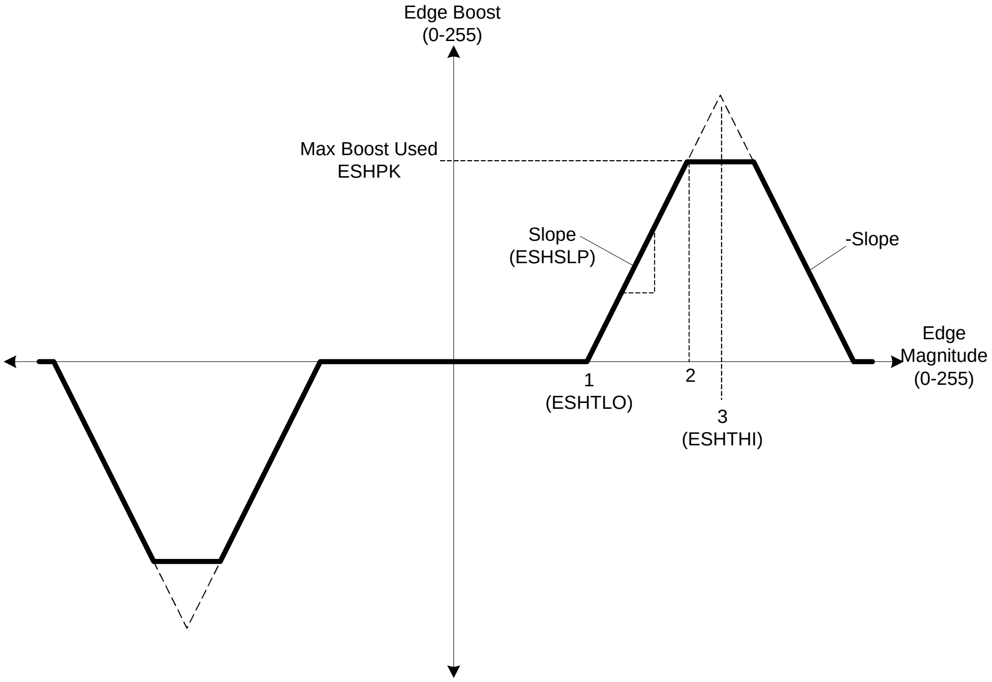


Figure 15. Edge Enhancement Transfer Function

The Edge Enhancement block can operate in several special modes besides the normal sharpening operation. These modes are listed below. These five additional modes are mutually exclusive of each other and the normal sharpen mode.

Emboss – replaces the Y channel with the edge map. Flat areas have low values, edges have higher values. This produce a white outline sketch on a grey background effect.

Supersharpen – doubles the slope of the normal sharpening function. This will give a kind of outlined effect.

Sketch – replaces the Y channel with the 255-edge map. Flat areas have high values, edges have lower values. This produces a black outline sketch on a grey background effect.

Smoothing filter - The smoothing filter is a 2 by 2 averaging filter applied to the 'Y' component. Recommended only for low lux conditions.

De-sharpening - transfer function is an inversion of the normal sharpening function

Focus Statistics

Register addresses: 0x2245 thru 0x226C

The Focus Quality function accumulates the sum of the edge magnitudes, as computed for edge enhancement, in four separate user defined windows. These maximum edge values can be used to feedback to a focus adjustment loop, either for real-time lens control or in-process setting of a fixed focus lens. The windows are defined by starting position and size. There is no restriction on placement. Window coordinates must be specified in the context of any sub sampling being performed.

Output Formatter

Register addresses: 0x2210

The Output Formatter is the final stage of pixel processing before transmission to the host. In this stage, the data, in whatever content form (RGB, YUV. YCbCr) is buffered and clocked out synchronously to the sample clock and horizontal and vertical synch signals. This block is also responsible for the generation of the synch signals. Options for the data output format are: big or little endian; with or without CCIR tags; and processed or raw data.

The output rate must generally be twice the imager pixel rate since the processing amplifies the amount of data per pixel. Use of the line buffer allows a speed mismatch between the imager and output data clocks. The processed data is in 16 bit format internally, 8 bit values are output by MSByte and LSByte alternately.

The 'endianness', or big endian versus little endian, controls whether the MSB or LSB comes out first.

CCIR tags can be appended to the stream as well. The separate synch signals are still generated when CCIR tags are used, but can be ignored if the host can process the embedded tags.

The Raw-10 data option outputs Bayer pattern data as 10 bit values within a 16 bit word. The data is LSB justified. with the upper 6 bits set to 0, output in the normal alternating byte method.

The Raw-8 data option truncates the internal 10 bit values down to 8 bit values, preserving the most significant bits. These are then output one per clock cycle. By utilizing the internal line buffer, or not, they can be doubled up and output in the double clocking mode like YUV data, or the output clock can be slowed to match the internal imager clock.

Also in this function is the ability to select either the 'gating' VSYNC or the pulsed VSYNC. See the Output Timing discussion.

In the output format block, there is also a line buffer that can be used if the output clock is asynchronous with the imager clock. The line buffer can hold maximum of 800x16 bits of data. For example, if the imager clock is 18MHz and clock multiplier is used to generate output clock of 27MHz, then the line buffer needs to be used to synchronize the data between the clocks. The use of this buffer must be explicitly enabled.

Clock Generator

Register addresses: 0x2200 thru 0x2205 *(NOTE: Default settings are not valid. Programming is REQUIRED)*

The internal clock generation function, figure 16, consists of a digital PLL, three dividers, a factor of two divider, and muxes for the imager clock selection and output clock selection. The PLL allows the input MCLK to be scaled by anywhere from $^{1}/_{16}$th to $^{255}/_{16}$ths. However, use of values below $^{8}/_{16}$ths is not recommended. The output of the PLL is used to provide the imager with its clock. The PLL is configured via I^{2}C registers and may be bypassed. All three dividers can be operated with whole number divisors of 1 thru 16.

When the PLL multiplier can be set as 1:1, some power can be saved by bypassing the PLL. However, note that the output of the PLL is stopped. So CLKCON[6] and CLKCON[0] must be selected to provide the CLKOUT and internal imager clock sources from the divider before the PLL.

The CLKOUT source should always be near, but not more than twice the internal imager clock frequency. The internal line buffer must be used if the frequency ratio is not exactly 2:1.

If the clock settings are changed, the PLL status should be monitored to verify that it has re-locked (MLOCKS[0]==1) before any other operations are performed. This should only take about 256 MCLKIN cycles.

Certain constraints are placed on the clocks at various stages as follows and must all be met at the same time:

Output of PLL <= 120 Mhz

Internal Imager clock (ICLK) <= 30 Mhz

Output at CLKOUT (8 bit) <= 100 Mhz

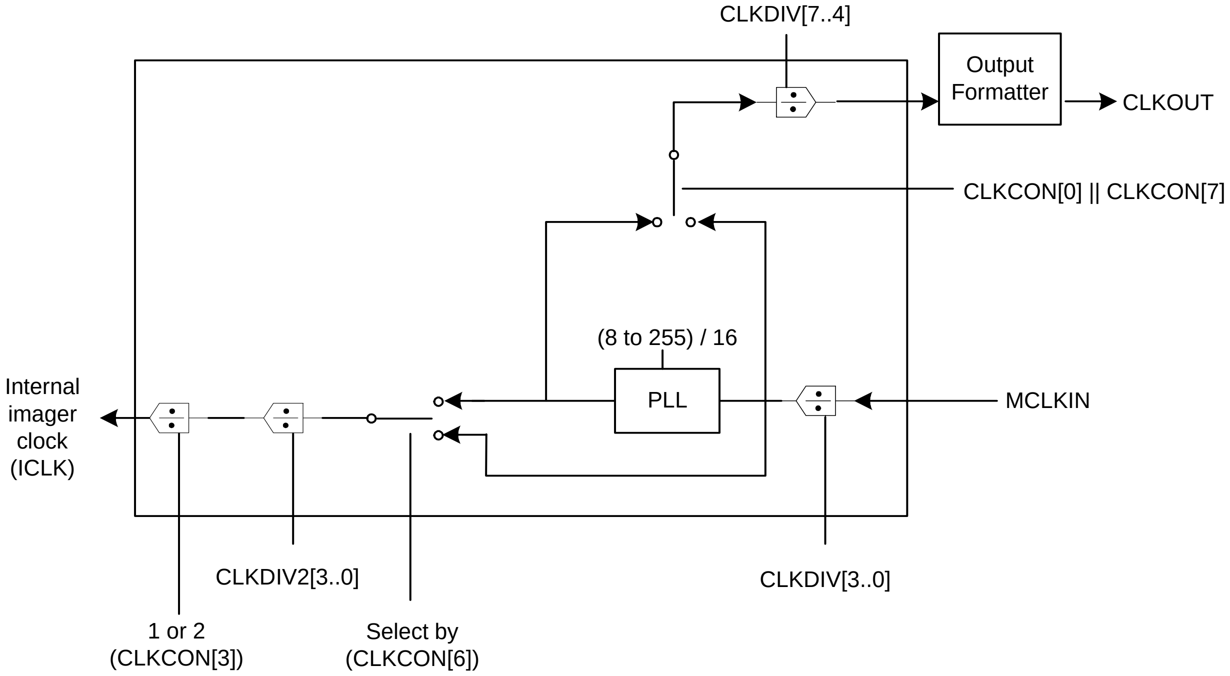


Figure 16. Clock Generator Function

General Output Timing

The readout process does not start until an internal signal called FRMPLS is activated. For video mode, continuous image output, the FRMPLS signal needs to be raised and held high. For still frame, single shot output, it should be just pulsed. The FRMPLS signal to the imager is controlled via bit 0 of the IMGSIG register.

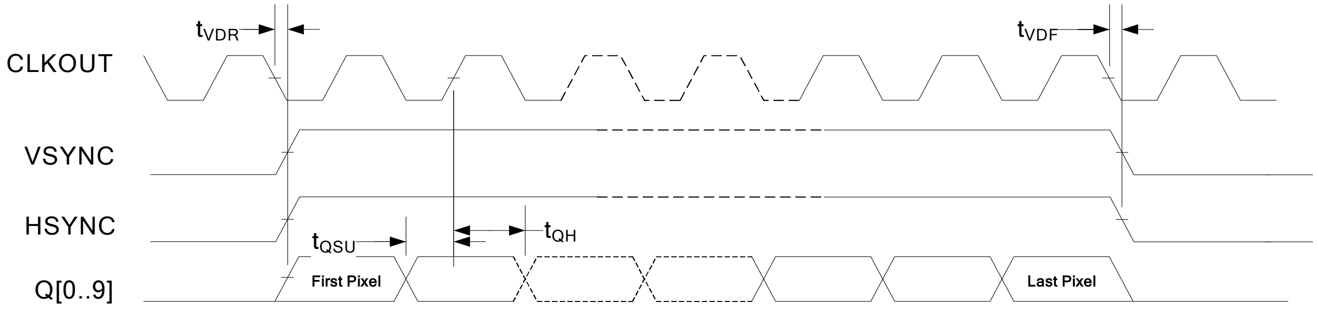
The VSYNC can be generated in two different ways. In one method, called the ‘gating VSYNC’, the VSYNC line goes active high and stays high from the first pixel output data to the last. In the other method, the VSYNC line is pulsed high for two line times, once at the beginning of the frame readout and once at the end. The second ending pulse is optional and may programmed off.

The VSYNC and HSYNC signals can be inverted independently. Due to a silicon error, the VSYNC inversion is not available when sync shaping is enabled.   The VSYNC and HSYNC pin assignments can also be swapped (reg 0x2507[7]).

In order for the FLSYNC output to be generated, the exposure time must be programmed to take longer than the scan time of the sub sampled frame. In other words the exposure time must represent more lines of imagery than the vertical frame size. This insures that at some point all the pixels in the frame are integrating simultaneously. The duration of this integration overlap is the difference in the frame size scan time and the exposure time. This is also the duration of the FLSYNC signal. If the exposure time is shorter than or equal to the vertical frame scan time, no FLSYNC pulse will be generated.

In applications where the number of I/Os is limited, the FLSYNC signal may be embedded in the HSYNC signal. This is controlled by the FLHSYN register. Since the FLSYNC should only ever be active in between VSYNCs, the FLSYNC and HSYNC can be combined by simply ORing the two together. The flash firing window can also be indicated by starting the VSYNC early, when the FLSYNC signal would normally be rising. VSYNC would then be held high until the normal starting point. This can also be selected in the FLHSYN register.

The tables and figures below show the various timing details of the sensor output signals. In some cases the non-visible. Internal timing signals are shown as a reference so that their relationship to clock and trimming settings is understood.



Symbol	Description	Min.	Typical	Max	Unit	Remark
t_{VDR}	Delay to FVLD/RVLD/Q[] Rising			3	ns	
T_{VDF}	Delay to FVLD/RVLD/Q[] Falling			2	ns	
T_{rise}	Output rise time			9	ns	
T_{fall}	Outputs fall time			8.4	ns	
t_{QSU}	Q[..] Outputs Setup time	9.1			ns	Assume output clock of 24 MHz, max delays and rise/fall times
T_{QH}	Q[..] Outputs Hold Time	15.1			ns	
Note:. Digital output timing figures assume a 40 pF load @ 2.8v OVDD						

Figure 17. Detailed Pixel and Sync Timing

Timing Characteristic	Symbol	Mode of Operation		
		Imager sub sampling		
		Full Frame	R2S2	R2S6
In units of CLKOUT's for YCbCr/YUV/RGB output		UXGA Output	VGA Output	QVGA Output
Effective Horizontal Size	EHSO	3200	1280	640
Effective Vertical Size*	EVSO	1200	600 (480[8])	300 (240[8])
ISP Sub Sample Mode[9]		Bypass / Bypass	4/5 / Bypass	4/5 / Bypass
Top Trim Size*[6]	TTS	4	2	0
Bottom Trim Size*	BTS	4	2	2
Left Trim Size[7]	LTS	$28* F_{CLKOUT} / F_{ICLK}$	$14* F_{CLKOUT} / F_{ICLK}$	$8* F_{CLKOUT} / F_{ICLK}$
Horizontal Blanking Period	HBPO	TLLO – EHSO		
Total Line Length	TLLO	$TLL * F_{CLKOUT} / F_{ICLK}$[5]		
Vertical Active Period	VAPO	EVSO * TLLO – HBPO		
Vertical Blanking Period	VBPO	$(TFL * F_{CLKOUT} / F_{ICLK}) – VAPO$		
Flash Sync Period	FSP	MAX(0, Texp*TLLO – EVS*TLLO)		
Lead In Blank [10]	LIBO	Texp*TLLO + (8+VPR)*TLLO + HBPO + TTS*TLLO + LTS		
In units of ICLK's		*Internal Readout size without trimming*		
Effective Horizontal Size	*EHS*	*1632*	*816*	*408*
Effective Vertical Size	*EVS*	*1208*	*604*	*302*
Horizontal Blanking Period	*HBP*	*184 or 288*[1]		
Total Line Length	*TLL*	*EHS + HBP*		
Vertical Active Period	*VAP*	*EVS*TLL – HBP*		
Total Frame Length	*TFL*	*MAX((EVS+8+VPR)*TLL, Texp*TLL)*[3]		
Vertical Blanking Period	*VBP*	*TFL – VAP*		
Lead In Blank[4]	*LIB*	*Texp*TLL + (VPR+8)*TLL + HBP*[2]		
Vertical Phantom Rows	*VPR*	*Either 2 or 8 as function of 0x250A[3]*		

** These are in units of lines as opposed to clocks*

[1] *Function of register 0x250A bit 7 and register 0x2510 bit 7*

[2] *The 8 dark rows must always be scanned out and can therefore be considered part of the 'blanking' time*

[3] *The Exposure time (Texp) is expressed in 'lines'*

[4] *Only for single shot or first frame of continuous*

[5] *The number of clock cycles per line scales by the ratio of the output and imager clock*

[6] *This is only the trimmed colorized pixels,. dark rows are accounted for in the blanking periods*

[7] *This trim number includes the dark columns*

[8] *This is actual number of HSYNCS generated within the EVSO period due to ISP sub sampling. Missing lines still occupy a line time, but there is no HSYNC active period during that line.*

[9] *Shown as Stage 1 Mode / Stage 2 Mode.*

[10] *Add an extra 8 clocks when any ISP Sub Sampling is performed.*

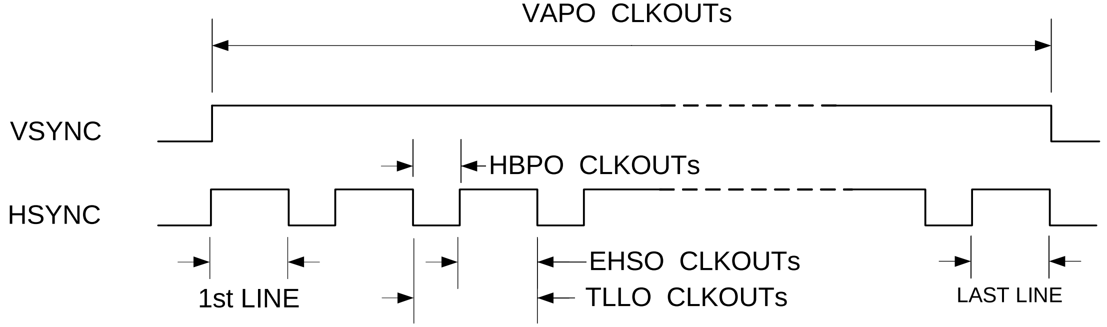


Figure 18. Details within Readout

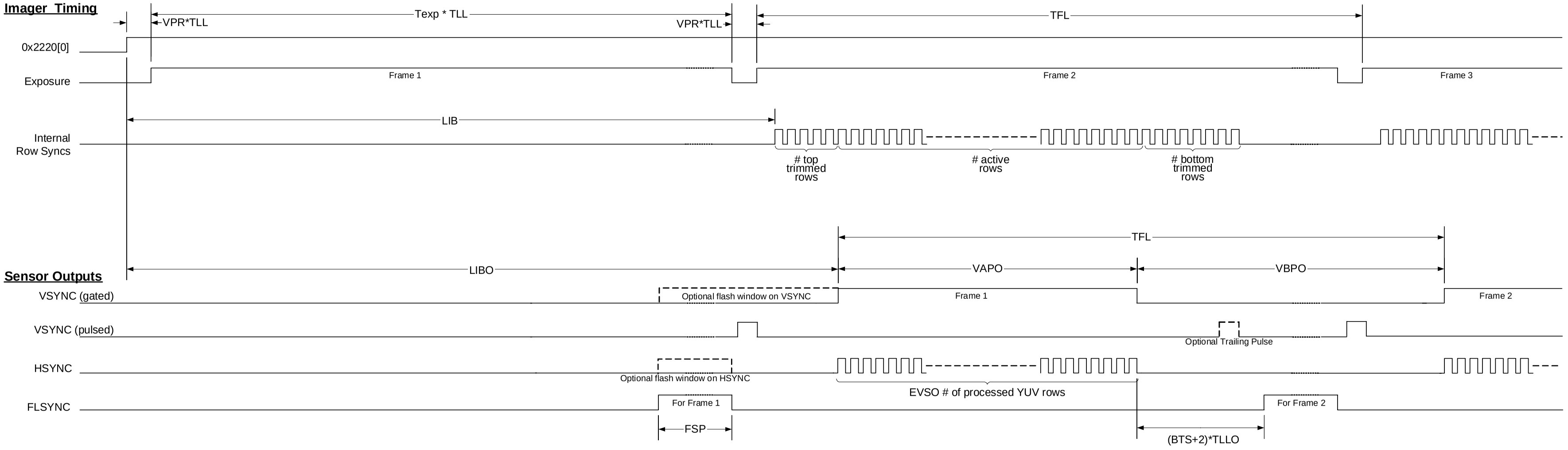


Figure 19. Frame Initiation and Readout Timing Details

Output Formats

There are 7 output data formats available for the non-JPEG image data. In all cases except one, the format will include 2 bytes of data per sampled sensor pixel. Every 'row' of readout will generate twice as many output samples as there are pixels. In all cases, the order of the two bytes may be swapped. Output type, endianess, tag presence are all selected via the I^{2}C interface. The available formats are"

RGB 565 – 16 bits per pixel: 5 bits red, 6 bits green, 5 bits blue

RGB 444 – 12 bits per pixel, 4bits each of red, green, and blue, can also be MSB or LSB justified.

YUV – 16 bits per pixel: 8 bits Y and 8 bits U or V on alternating clocks, full range 0-255.

YCbCr without CCIR tags – 16 pits per pixel: 8 Bits Y and 8 bits Cb or Cr on alternating clocks, limited range.

YCbCr with CCIR tags – 16 pits per pixel: 8 Bits Y and 8 bits Cb or Cr on alternating clocks, plus CCIR locating tags in blanking region.

RAW-10 Bayer data – 16 bits per pixel with upper 6 bits forced to '0'.

RAW-8 Bayer data – 8 bits per pixel outputting only the 8 MSbits of each (requires setting 0x2210[7]=1 when using the line buffer)..

Format	Endian	1st Clock	2nd Clock	3rd Clock	4th Clock
YUV full range	Big	Y_0	U_0	Y_1	V_1
	Little	U_0	Y_0	V_1	Y_1
YCbCr limited range	Big	Y_0	Cb_0	Y_1	Cr_1
	Little	Cb_0	Y_0	Cr_1	Y_1
RGB565	Big	R_0[4..0] G_0[5..3]	G_0[2..0] B_0[4..0]	R_1[4..0] G_1[5..3]	G_1[2..0] B_1[4..0]
	Little	G_0[2..0] B_0[4..0]	R_0[4..0] G_0[5..3]	G_1[2..0] B_1[4..0]	R_1[4..0] G_1[5..3]
RGB444 lsb 0's	Big	R_0[3..0] G_0[3..0]	B_0[3..0] 0's	R_1[3..0] G_1[3..0]	B_1[4..0] 0's
	Little	B_0[3..0] 0's	R_0[3..0] G_0[3..0]	B_1[4..0] 0's	R_1[3..0] G_1[3..0]
RGB444 msb 0's	Big	0's R_0[3..0]	G_0[3..0] B_0[3..0]	0's R_1[3..0]	G_1[3..0] B_1[3..0]
	Little	G_0[3..0] B_0[3..0]	0's R_0[3..0]	G_1[3..0] B_1[3..0]	0's R_1[3..0]
Raw-10	Big	0's Raw_0[9..8]	Raw_0[7..0]	0's Raw_1[9..8]	Raw_1[7..0]
	Little	Raw_0[7..0]	0's Raw_0[9..8]	Raw_1[7..0]	0's Raw_1[9..8]
Raw-8 0x2210[7]=0 w/linebuffer	Big	0's	Raw_0[9..2]	0's	Raw_1[9..2]
	Little	Raw_0[9..2]	0's	Raw_1[9..2]	0's
Raw-8 0x2210[7]=1 w/linebuffer	Big	Raw_0[9..2]	Raw_1[9..2]	Raw_2[9..2]	Raw_3[9..2]
	Little	Raw_1[9..2]	Raw_0[9..2]	Raw_3[9..2]	Raw_2[9..2]
Raw-8 w/o linebuffer		Raw_0[9..2]	Raw_1[9..2]	Raw_2[9..2]	Raw_3[9..2]
BT656	N/A	Cb_0	Y_0	Cr_1	Y_1

Output Pin Tri-state Control

Register addresses: 0x2217

The Bi8921 has the ability to tri-state the output pads for CLKOUT, VSYNC, HSYNC, and the 8 data lines Q[7.0]. This allows the sensor to be used in a system design with multiple imagers on a single bus without the need for extra bus multiplexer circuitry. The tri-state function can be implemented with either a hardware signal (on pad 'TE'), or an I^2C command.

The TE pad has an internal pull-up, so left floating it allows control via the I2C command. The I^2C bit in register 0x2217 is defaulted to '1, so this will tri-sate the outputs. The state of the outputs is also influenced by the STANDBY pad. The truth table for the various combination is shown below.

TE pad	I^2C register 0x2217[0]	STANDBY pad	Result
1	1	0	Pads are Tri-stated
1	0	0	Pads output.
0	1	0	Pads output.
0	0	0	Pads output.
1	1	1	Pads are Tri-stated
1	0	1	Pads are Tri-stated
0	1	1	Pads output.
0	0	1	Pads output.

JPEG Compression Function

Register addresses: 0x2600 thru 0x2622

Bi8921 has a built-in JPEG compression engine. The compression engine is bypassable so uncompressed data like YUV 4:2:2 or RGB565 can still be output. The compression engine has a built-in clock divider to adjust the output clock frequency. The JPEG clock divider will affect the output even if compression is bypassed, so be sure to set the divider appropriately even if not compressing.

JPEG Output

The JPEG output shares the same output lines of CLKOUT, VSYNC, HSYNC and Q[7:0] with the uncompressed output. However the JPEG data output rate is not uniform. The JPEG data rate for a portion of the image depends upon the complexity of the scene in that portion and also the quantization table used. Because of this the HSYNC is not inherently of uniform width but changes with the scene content.

Three kinds of JPEG output schemes will be available. In the first two schemes, the JPEG data comes out in bursts. The JPEG engine is continuously compressing the image data and output takes place whenever enough bytes of JPEG are generated. In the third scheme, the JPEG output has a uniform structure. There are a fixed number of HSYNCs in a JPEG frame and fixed number of bytes per HSYNC. In all cases the Q[7:0], VSYNC and HSYNC signals can be sampled with the rising edge of CLKOUT as in YCbCr mode.

Embedded Thumbnail

The user can choose to include or not to include a thumbnail image along with the JPEG output. Thumbnail output is in 4:2:2 sub-sampled YUV format. The output sequence is YUYV. The thumbnail size is controlled by providing a sub-sampling factor. Sub-sampled image size is generated by dividing both the row and column number of the output image by the sub-sampling factor. For example, if the original image is of size 1600x1200 and a sub-sampling factor of 5 is chosen, the sub-sampled image will be of size 320x240. How the thumbnail is presented in the output stream varies with the output scheme.

Scheme I:

In the first scheme, the CLK signal is free running and is active whether the output is present or not for a CLKOUT cycle. The output signals are shown below in Fig 20. The shaded area represents the time when a particular signal is transitioning. Clock is active for the entire period and is shown by keeping it shaded.

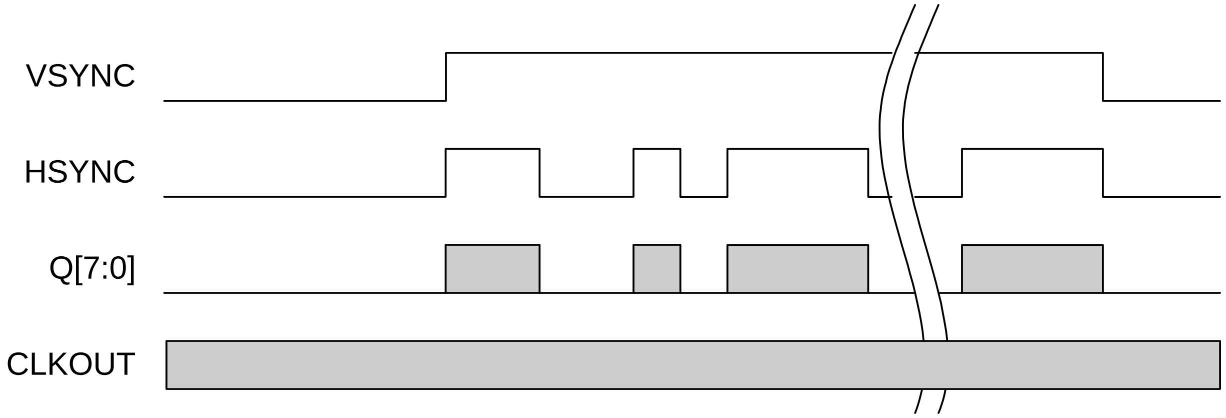


Figure 20. Scheme I - JPEG output signals when clock is NOT gated

Scheme II:

In the second scheme the clock is gated and it is active only at times when the VSYNC and HSYNC are high. The signaling scheme is shown in Figure 21.

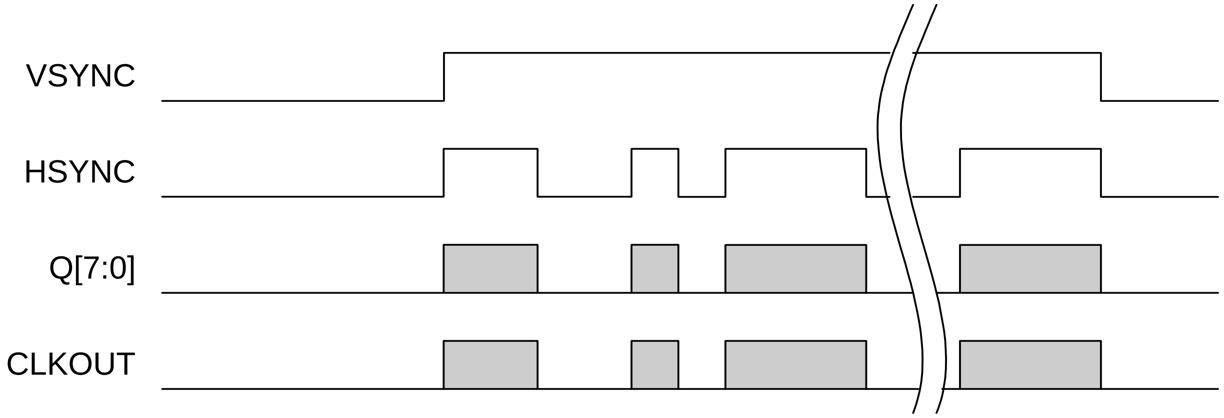


Figure 21. Scheme II - JPEG output signaling when clock is gated

In both scheme I and II, the thumbnail data, if enabled, is output whenever it is available and the JPEG is not being output. The thumbnail is marked relative to the other data with marker bytes. A chunk of thumbnail data will be bracketed with a two byte sequence of 0xFF-0xBE at the start, and 0xFF-0xBF at the end. The number of bytes of thumbnail data will always be a multiple of 4. So, if 0xFF-0xBE is detected, four bytes of thumbnail should be read. If the next two bytes are 0xFF-0xBF, the thumbnail chunk is finished. Otherwise read four more bytes of thumbnail and repeat the process.

Scheme III:

For some camera interface chips it is essential to provide data of uniform size and structure. In that case the number of bytes output per each HSYNC, number of clock periods in each blanking period between the HSYNCs, and the number of HSYNCs in a given image need to be constant and pre-determined.

As the JPEG output is bursty in nature, some filler bytes need to be added to satisfy these requirements. We choose the filler bytes to be a sequence of 5 bytes (0xFF 0xFE 0x00 0x03 0x00). This sequence corresponds to a JPEG comment with 3 bytes in the payload. All the occurrences of this sequence need to be removed while parsing the output to make a sensible JPEG image.

The user can program the number of JPEG + filler bytes per line, the number of thumbnail bytes per line and number of blank clock cycles per line. The sum of these three numbers should correspond in time to the uncompressed YUV line period itself. For example, consider a full resolution output mode. The number of uncompressed bytes output per line is 3200 and the blanking period is 640 clock cycles. So the total number of uncompressed data clock cycles per line is 3840. Now let us say dividing the uncompressed clock frequency by a factor of 3 generates the JPEG output clock. So, time equivalent of 3840 YUV clock cycles is 3840/3 = 1280 JPEG clock cycles. So, the number of JPEG + filler bytes per line, plus the number of thumbnail bytes per line and the number of blank clock cycles per line should be equal to 1280.

The number of lines in the JPEG output and the YUV output line on which the JPEG output begins can be programmed. Depending upon the sub-sampling settings for the thumbnail and the scaling setting of the full image itself, different values for these parameters need to be chosen. These values will be provided to the user by *Biomorphic VLSI, Inc.* The main aim is to reduce the possibility of JPEG errors.

The number of thumbnail bytes per line also should be calculated such that the total number of thumbnail bytes output per image corresponds to the thumbnail image size itself. For example, let us say a thumbnail image of size 320x240 is generated from a 1600x1200 image using a sub-sampling ratio of 5. The total number of thumbnail bytes per image is 320x240x2 = 153600. The number of thumbnail bytes per line should be set equal to 153600/1200 = 128.

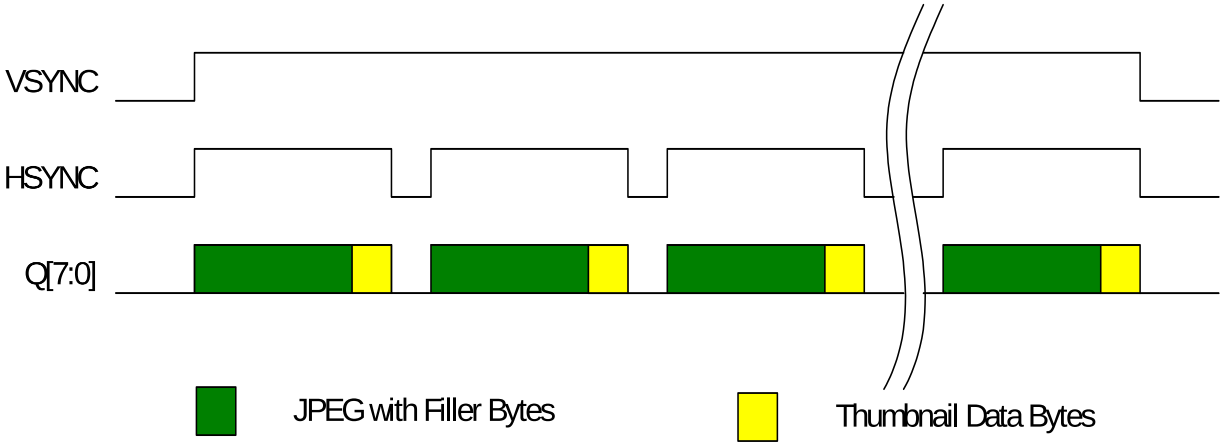


Figure 22. Scheme III - Uniform JPEG Output organization

Controlling the number of JPEG bytes

The number of bytes in the JPEG output stream depends upon the scene. Though it is not possible to specify the size of a given compressed image, it can be influenced using the quantization table. The Bi8921 provides for two ways of setting the quantization table. It can either be set as multiples of the default quantization table specified in the JPEG specification or the individual values in the quantization table (total of 128, eight bit values) can be set explicitly. In the former case, multipliers (QFACTOR) of 2, 1, ½, ¼ and 1/8 are allowed. Each value in the default table is multiplied by the QFACTOR and rounded off to the nearest integer. The higher the QFACTOR, the smaller the compressed image will be and the poorer the quality of the compressed images. All table values must be in the range 1 to 255. They are stored in the zig-zag sequence in which they are also normally found in the JPEG file header. This is shown below

The default luminance table and default chrominance table are shown below for reference.

While the specific size of the JPEG data cannot be set, a maximum allowable size may be programmed. This is available only for the burst mode output schemes, since scheme III is implicitly a fixed limited by the line count and line length parameters. If the actual data exceeds the specified limit, the output is truncated to the limit and an error is generated. The limit does not include any thumbnail bytes, however since the thumbnail size is known, except for any 0xFF-0xBF bytes, this can largely be accounted for when allocating storage.

Default luminance quantization table

(Offsets 0 thru 63)

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Default chrominance quantization table

(Offsets 63 thru 127)

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Offset within table for luminance and chrominance terms

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Other JPEG parameters

Only 422 sub-sampling (no sub-sampling of Y data and 2:1 sub-sampling of U and V data in columns and no sub-sampling in rows) is supported.

Only a Fixed Huffman table, as defined as the default in the JPEG specification, is supported.

The total number of bytes in the JPEG portion can be read back in registers 0x2620-0x2622.

Choosing the output rate (output clock frequency)

The main objective of selecting the JPEG output is to reduce the number of bytes output from the imager and thereby allow reduction of the output clock frequency while maintaining a short readout time. The Bi8921 provides a way of dividing the frequency of the clock used internally for synchronizing the uncompressed data (referred to as YUV_CLK hereafter) by ratios of 1, 2, 3 … 16 to generate the CLKOUT.

Indicating the error condition

It is possible that the internal buffer for synchronizing the JPEG data between YUV_CLK and CLKOUT can overflow. This is more probable when the division factor chosen to generate CLKOUT from YUV_CLK is higher. Two methods for indicating the error condition are supported.

Method 1:

If such an overflow occurs, it is indicated by bringing HSYNC high for a short duration of time after VSYNC goes low at the end of the frame. The number of CLKOUT cycles for which HSYNC is high after VSYNC goes low can be anywhere from 1 to 16 and can be chosen by register setting. The signaling is shown below.

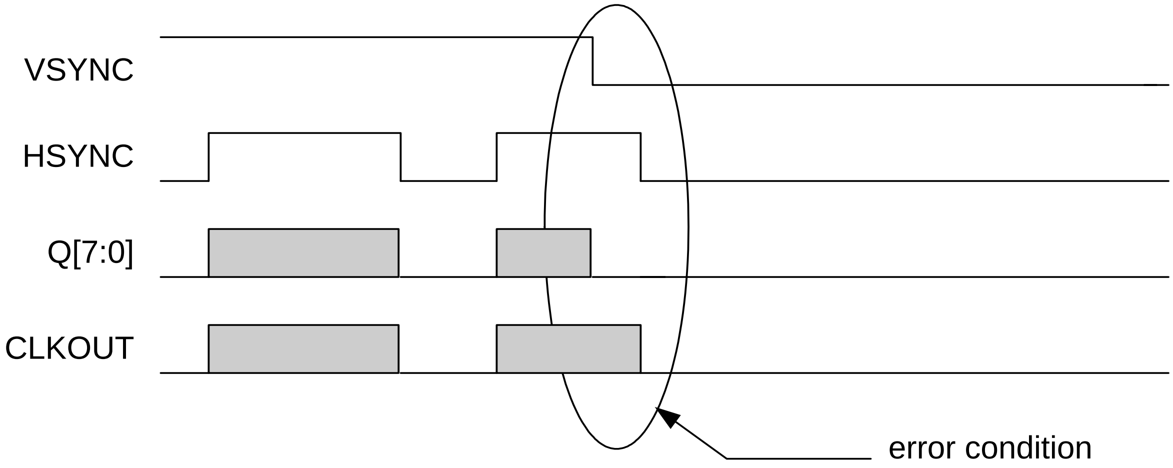


Figure 23. Signaling the error in JPEG compression

Method 2:

Instead of signaling the error condition on HSYNC, the Bi8921 can also use the data signals to indicate error condition. Normally, when there is no JPEG error, DATA lines are set to 0 after falling edge of VSYNC. In the case of an error condition, after the falling edge of VSYNC the DATA line can output a 4-byte “string” to indicate the error. The string can be sampled at 4 consecutive rising edges of CLKOUT after VSYNC falls. The string can be set using I^{2}C register setting. An example string could be 0xBADD.

Reshaping the VSYNC and HSYNC

It may be necessary in some applications to have predetermined relationships between the VSYNC, HSYNC and output clock. This means there needs to be a fixed number of clock cycles between the rising edge of VSYNC and the first rising edge of HSYNC, the width of HSYNC valid period and the width of the HSYNC blanking period. Moreover, these numbers may need to be restricted to even numbers.

To accommodate this, the Bi8921 includes a sync reshape block. VSYNC can be set to rise early. The delay between the rising edge of VSYNC and first rising edge of HSYNC can be set using register HSNDLY. The width of HSYNC and HSYNC blanking period are programmable through registers VLDCLK and BLKCLK respectively. The reshaping block can be enabled in both JPEG and YUV output cases.

Note that these numbers cannot be set arbitrarily but should be set very close to 'non-shaped' timing. The HSNDLY registers should be set to a value 2 or 3 higher than what the delay would be without having the reshaping block enabled. However, HSNDLY should not exceed the actual delay by more than 64.

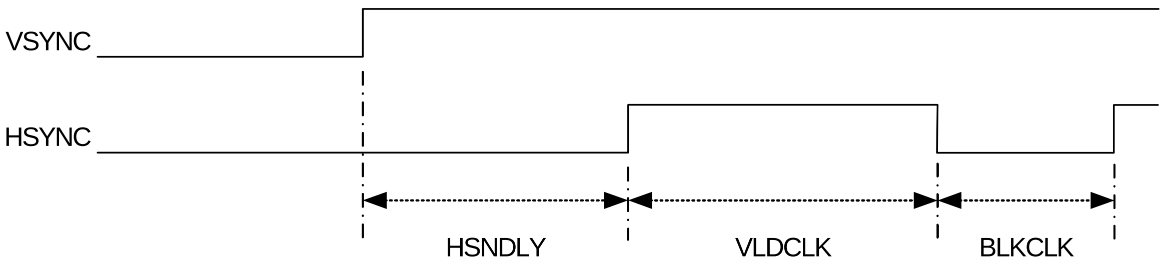


Figure 24. Relationship of registers to signal components

Serial Interface & Control Registers

I^{2}C Interface

The host application interface is implemented as a standard, Fast-mode (400kb/s), I^{2}C slave device. The Bi8921 responds to device address 45h (8Ah for write and 8Bh for read). It operates in 7 bit mode only. The General Call address is ignored, should it be received. The HighSpeed mode of I^{2}C is also supported by selective bonding of the I2CSEL pad. In HighSpeed mode, only the 8-bit address mode is supported.

The Bi8921 does not include termination resistors for the I^{2}C bus, these must be supplied by the host circuit design. The Bi8921 will never pull-down the SCL line to signal a not-ready condition. This condition can not occur.

The register address and data are coded as data to the I^{2}C transaction. These bytes follow the I^{2}C slave device address byte and acknowledge. The register addresses are 16 bits; so two bytes are required for the address. The address MSB is sent first. Any register data follows the address field. The data is always a single byte. Auto increment for multiple address operations is not supported.

The details of the I^{2}C bus operation are not described here. They may be found at numerous locations on the World Wide Web. The specific operation of the register write and read protocols implemented by the Bi8921 are however described below.

Write Operation

Write operations are only supported in the single byte mode. The complete operation requires addressing the Bi8921 and then sending 3 bytes of 'data'. The first two bytes are the address of the Bi8921 register to be written to. These are sent as high byte first, then low byte. The third byte is the data that will be written to the desired address. The complete sequence is shown below.

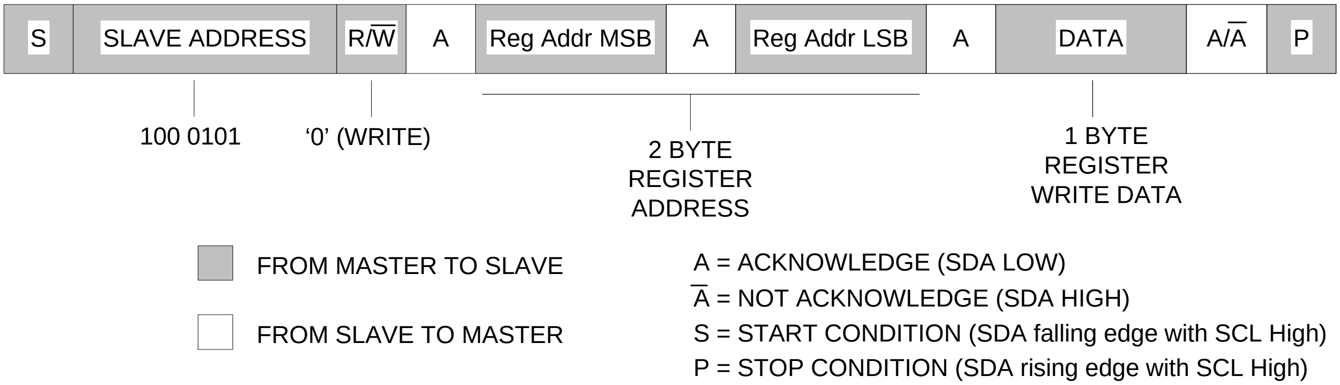

S
SLAVE ADDRESS
R/$\overline{W}$
A
Reg Addr MSB
A
Reg Addr LSB
A
DATA
A/$\overline{A}$
P
100 0101
'0' (WRITE)
2 BYTE REGISTER ADDRESS
1 BYTE REGISTER WRITE DATA
FROM MASTER TO SLAVE
FROM SLAVE TO MASTER
A = ACKNOWLEDGE (SDA LOW)
$\overline{A}$ = NOT ACKNOWLEDGE (SDA HIGH)
S = START CONDITION (SDA falling edge with SCL High)
P = STOP CONDITION (SDA rising edge with SCL High)

Figure 25. I^{2}C Register Write Transaction

Read Operation

Read operations can occur as single word transfers only. A read operation begins with a register write without data. The address of the register to be queried should be used as the register address in the write cycle. This readies the interface to respond with the data from that register on the next read request. The actual read request is then performed by generating a read operation. The Bi8921 will respond with the contents of the register address in the previous write cycle.

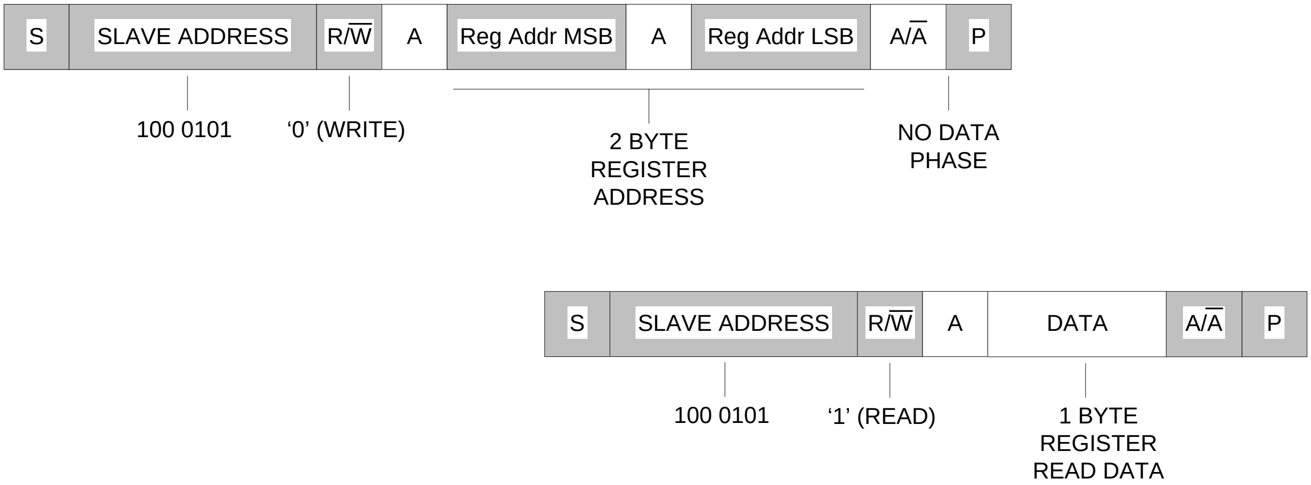


Figure 26. I²C Register Read Transaction

Control Registers

Note: In all the register definitions, if the register width is more than one byte, the lower address occupies the LSB byte and the higher addresses refer to the MSB bytes. Access is by single byte only, so multi-byte registers require two separate accesses, one to each address (LSB and MSB) to completely update the register. These accesses may be performed in any order.

Refer to application notes for recommended settings.

Address	Register	Bits	Type	Description	Default
				Trimmer Module registers	
0x2010	TOPDRKR	[3:0]	RW	Number of dark pixel rows on the top.	0x08
0x2011 0x2012	TMOSTBR	[10:0]	RW	Topmost row containing bright pixels	0x000B
0x2013	LEFTDC	[5:0]	RW	Dark columns on the left side of the imager	0x18
0x2014 0x2015	LEFBCS	[10:0]	RW	Leftmost column containing the bright pixels	0x001B
0x2016 0x2017	RDRKCB	[10:0]	RW	Right Dark column begin. The beginning column for right side dark pixels. In case there are no dark columns on the right put any number greater than or equal to the number of total columns	0x0660
0x2018 0x2019	RTRANCB	[10:0]	RW	Right transition column begin. Column number where the transition pixels begin on the right	0x065D
0x201A 0x201B	BOTTDRB	[10:0]	RW	Bottom dark row begin. The beginning row for bottom dark pixels.	0x04C0
0x201C 0x201D	BOTTRRB	[10:0]	RW	Bottom transition row begin. The beginning row for bottom transition pixels	0x04BD
0x201F	BOTDR	[7:0]	RW	Bottom dark rows. Used for creating pulsed VSYNC.	0x03
0x2038	TRCON	[1:0]	RW	Configuration input word for Trimmer module [0]: Dark pixel trim bit 1: Remove the dark pixels from the data stream. 0: Do not remove the dark pixels. Note that this value will be ignored if the transition pixel remove bit is set. This is done in order to eliminate the possibility that there could be a "hole" left inside the image [1]: Transition pixel trim bit 1: Remove the transition pixels 0: Do not remove the transition pixels. Note: All the values input to the block are latched at the beginning of the frame. So if a value is changed during the frame read out, it will take effect only in the next frame.	0x03

Auto Exposure and White Balance registers

Address	Register	Bits	Type	Description	Default
0x2040	AEWCON	[6:0]	RW	Auto exposure and white balance configuration. [0]: AE/AWB bypass 0 → process 1 → bypass [1]: AE/AWB compute only. 0 → normal, closed loop operation. 1 → open loop. Compute settings only, do not apply [2]: Flag to indicate that reduction in exposure time should result in AEWB not being done in the next frame [3]: Extra window statistics enable. Set this bit to 1 to compute extra statistics from window2. [4]: Fix the exposure time. Exposure time is set at the maximum exposure time and only the gain is varied to get the desired exposure brightness level. [5]: Use the maximum color average instead of the computed Y for auto exposure. [6]: Exposure time warning. If the applied digital common gain is less than 1, warning to reduce the maximum exposure time or change the fix exposure time bit.	0x04
0x2041 0x2042	AEWSR	[10:0]	RW	AEWB starting row. Defines the window over which AEWB statistics are calculated	0x0000
0x2043 0x2044	AEWSC	[10:0]	RW	AEWB starting column. Defines the window over which AEWB statistics are calculated.	0x0000
0x2045 0x2046	AEWRC	[10:0]	RW	Number of rows in AEWB window.	0x0402
0x2047 0x2048	AEWCC	[10:0]	RW	Number of columns in the AEWB window.	0x0502
0x2049	AEWSSC	[1:0]	RW	Sub-sampling to be done before collecting statistics. This reduces the number of pixels used in the AEWB window for statistics computation. 00 → No sub-sampling needed. 01 → R2S2 sub-sampling done on Bayer data for AEWB. 10 → R2S6 sub-sampling done on Bayer data for AEWB. 11→ R2S14 sub-sampling done on Bayer data for AEWB.	0x00
0x204A	YDES	[7:0]	RW	Desired value for Y. This is the target of automatic exposure algorithm. AE algorithm tries to set the values of exposure time and gain such that the average luminance of the image is close to YDES.	0x80

Address	Register	Bits	Type	Description	Default
				This value is on 8-bit scale.	
0x204B	YTOLR	[7:0]	RW	Tolerance in the value of Y. The exposure time, common gain values are not changed if the average luminance of the image is within the range of (YDES – YTOLR) to (YDES + YTOLR).	0x0C
0x204C	WBTOLR	[7:0]	RW	White balance tolerance. If the average value for a color lies within WBTOLR of the maximum color average, its gain is not changed.	0x08
0x204D	G1PAVG	[7:0]	RO	Computed Green1 pixel average.	NA
0x204E	RPAVG	[7:0]	RO	Computed Red pixel average.	NA
0x204F	BPAVG	[7:0]	RO	Computed Blue pixel average.	NA
0x2050	G2PAVG	[7:0]	RO	Computed Green2 pixel average.	NA
0x2051	G1WBG	[7:0]	RO	Computed White balance gain for Green1 pixels.	NA
0x2052	RWBG	[7:0]	RO	Computed White balance gain for Red pixels.	NA
0x2053	BWBG	[7:0]	RO	Computed White balance gain for Blue pixels.	NA
0x2054	G2WBG	[7:0]	RO	Computed White balance gain for Green2 pixels.	NA
0x2055	SRGREG	[7:0]	RW	Sensor register for setting the Red (and Blue) gain.	0x01
0x2056	SG1GREG	[7:0]	RW	Sensor register for setting the Green1 (and G2) gain.	0x03
0x2057	SG2GREG	[7:0]	RW	Not used	n/a
0x2058	SBGREG	[7:0]	RW	Not used	n/a
0x2059	SSFG1	[7:0]	RW	Sensor gain register setting to achieve a gain of 1x.	0x00
0x205A	SSFG2	[7:0]	RW	Sensor gain register setting to achieve a gain of 2x.	0x48 override to 0x02
0x205B	SSFG4	[7:0]	RW	Sensor gain register setting to achieve a gain of 4x.	0x90 override to 0x01
0x205C	SSFG8	[7:0]	RW	Sensor gain register setting to achieve a gain of 8x.	0xD7 override to 0x03
0x205D	CFREQ	[3:0]	RW	AEWB computation frequency. AEWB is computed only after every CFREQ frame/frames.	0x01
0x205E	EXPSTPS	[7:0]	RW	Exposure time change steps. The ideal exposure time (applied exposure time * gains applied) can vary only in steps of EXPSTPS at a time. Set this value to 0 to disable this feature.	0x00
0x205F	EXLSBR	[7:0]	RW	Sensor exposure time LSB register.	0x09
0x2060	EXMSBR	[7:0]	RW	Sensor exposure time MSB register.	0x08
0x2061	URCSR	[7:0]	RW	Sensor register for update required command.	0x7F
0x2062 0x2063	CEXPT	[15:0]	RO	Computed exposure time.	NA
0x2064	CSGAIN	[3:0]	RO	Computed sensor gain. 1, 2, 4 or 8.	NA
0x2065	CDGAIN	[6:0]	RO	Computed digital common gain. The value in this register is divided by 64 to get the multiplication factor. (minimum value is 0x20 = 0.5x)	NA
0x2066	ESEXPT	[15:0]	RW	Externally set exposure time.	0x0430

Address	Register	Bits	Type	Description	Default
0x2067					
0x2068	ESSGAIN	[3:0]	RW	Externally set sensor gain.	0x01
0x2069	ESDGAIN	[6:0]	RW	Externally set digital gain. This value is divided by 64 to get the multiplier for digital gain.	0x40
0x206A 0x206B	MEXPT	[15:0]	RW	Maximum exposure time..	0xFFFF
0x206C 0x206D	FLICKT	[9:0]	RW	Flicker time in terms of number of rows of exposure. AE/AWB block sets the exposure time either to be a multiple of flicker period of less than that.	0x0001
0x206E 0x206F	W2SROW	[10:0]	RW	Statistics computation window2 starting row.	0x0000
0x2070 0x2071	W2SCOL	[10:0]	RW	Statistics computation window2 starting column.	0x0000
0x2072 0x2073	W2ROWS	[10:0]	RW	Number of rows in Statistics computation window2.	0x0402
0x2074 0x2075	W2COLS	[10:0]	RW	Number of columns in Statistics computation window2.	0x0502
0x2076	W2SUBS	[2:0]	RW	Window 2 control [1:0]: Sub-sampling 00 → No sub-sampling needed. 01 → R2S2 sub-sampling done on Bayer data for AEWB. 10 → R2S6 sub-sampling done on Bayer data for AEWB. 11 → R2S14 sub-sampling done on Bayer data for AEWB. [2]: Window selection 0 → Window 1 stats for AE & AWB 1 → Window 2 for AWB & Window 1 for AE	0x00
0x2077	W2G1AV	[7:0]	RO	Window 2 green1 pixel average.	--
0x2078	W2RAVG	[7:0]	RO	Window 2 red pixel average.	--
0x2079	W2BAVG	[7:0]	RO	Window 2 blue pixel average.	--
0x207A	W2G2AV	[7:0]	RO	Window 2 green2 pixel average.	--
0x207B	WBMAVG	[7:0]	RW	Minimum average for WB computation. If the maximum of color averages is less than this value, no WB adjustment is made. This prevents the WB oscillations in low light conditions.	0x00
0x207C	WBMAXA	[7:0]	RW	Maximum average for WB computation. If the maximum of the color averages is higher than this number, WB adjustments will not be made	0xFF
0x207D	AERTOL	[7:0]	RW	Automatic exposure readjust tolerance. Once the AE has converged near the desired value of Y, it is readjusted only if it deviates from the desired Y by this amount.	0x0C

Address	Register	Bits	Type	Description	Default
0x207E	WBGCAP	[7:0]	RW	White balance gain cap. Maximum value of white balance gain allowed.	0xFF
0x207F	MSAPIX	[7:0]	RW	Maximum saturated pixels allowed. If the number of saturated pixels in a frame is greater than 256 * MSAPIX then no WB gains are computed on that frame.	0xFF

White Balance Gain registers

Address	Register	Bits	Type	Description	Default
0x20F0	GCNFG	[1:0]	RW	[0]: Gain Multiplier Bypass. 0 → process 1 → bypass [1]: 0 → Use the values externally set for white balance. Registers 20F1h-20F4h 1 → Use the white balance gains calculated by the AE/AWB block.	0x02
0x20F1	RWBG	[7:0]	RW	Red pixel gain for external control. In 1/64ths per LSB	0x40
0x20F2	G1WBG	[7:0]	RW	Green1 pixel gain for external control. In 1/64ths per LSB	0x40
0x20F3	G2WBG	[7:0]	RW	Green2 pixel gain for external control. In 1/64ths per LSB	0x40
0x20F4	BWBG	[7:0]	RW	Blue pixel gain for external control. In 1/64ths per LSB	0x40

Bad Pixel Correction registers

Address	Register	Bits	Type	Description	Default
0x2080	BPCCON	[2:0]	RW	[0]: Bad Pixel Correction bypass. 0 → the data is corrected for bad pixels before the YCbCr conversion takes place. 1 → the data does not go through the bad pixel concealer and directly goes from trimmer block to the YCbCr conversion block. [1]: Brighter Pixel Replace Enable 0 → do not replace 1 → replace [2]: Darker Pixel Replace Enable 0 → do not replace 1 → replace	0x06
0x2081 0x2082	BPBGTH	[9:0]	RW	Threshold for identifying *bright outlier* green pixels	0x03FF
0x2083 0x2084	BPBRTH	[9:0]	RW	Threshold for identifying *bright outlier* red pixels	0x03FF
0x2085 0x2086	BPBBTH	[9:0]	RW	Threshold for identifying *bright outlier* blue pixels	0x03FF
0x2087	BCCNT	[9:0]	RO	Number of bad pixels per image. Maximum value 1023	--

Address	Register	Bits	Type	Description	Default
0x2088					
0x2089 0x208A	BPDGTH	[9:0]	RW	Threshold for identifying *dark outlier* green pixels	0x03FF
0x208B 0x208C	BPDRTH	[9:0]	RW	Threshold for identifying *dark outlier* red pixels	0x03FF
0x208D 0x208E	BPDBTH	[9:0]	RW	Threshold for identifying *dark outlier* blue pixels	0x03FF

Interpolation Configuration registers

0x2090	IPCON	[2:0]	RW	[0]: Interpolation Bypass 0 → process 1 → bypass [1]: Starting row 0 → Green & Red row 1→ Green & Blue row [2]: Starting column 0 → Green & Blue column 1→ Green & Red column [3]: If set to 1, one row and one column at each edge is trimmed.	0x0E

Color Correction configuration registers

0x20A0	CCBYP	[0]	RW	[0]: Color Correction Bypass. 0 → process 1 → bypass	0x00
0x20A1	CCM11	[6:0]	RW	Color correction matrix entry (1,1)	0x20
0x20A2	CCM12	[6:0]	RW	Color correction matrix entry (1,2)	0x00
0x20A3	CCM13	[6:0]	RW	Color correction matrix entry (1,3)	0x00
0x20A4	CCM21	[6:0]	RW	Color correction matrix entry (2,1)	0x00
0x20A5	CCM22	[6:0]	RW	Color correction matrix entry (2,2)	0x20
0x20A6	CCM23	[6:0]	RW	Color correction matrix entry (2,3)	0x00
0x20A7	CCM31	[6:0]	RW	Color correction matrix entry (3,1)	0x00
0x20A8	CCM32	[6:0]	RW	Color correction matrix entry (3,2)	0x00
0x20A9	CCM33	[6:0]	RW	Color correction matrix entry (3,3)	0x20

Color Saturation configuration registers

0x20D0	SATBYP	[0]	RW	[0]: Color saturation Bypass. 0 → process 1 → bypass	0x00
0x20D1	SATVAL	[7:0]	RW	Color Saturation Gain.	0x00

Address	Register	Bits	Type	Description	Default
				[7]: Sign Bit 0 → saturation increased (>= 1.0) 1 → desaturation (<1.0) [6:0]: Saturation amount	

Channel Balancer registers

Address	Register	Bits	Type	Description	Default
0x2110	CBCON	[2:0]	RW	Channel Balancer configuration register [0]: Bypass. 0 → process 1 → bypass [2:1]: Mode 00→ calculate offsets using top dark rows (method 1) 01→ calculate offsets using side dark columns (method 2) 10→ use manually loaded values	0x00
0x2111	CBCNT	[1:0]	RW	Total number of pixels to consider when calculating offsets. 00→ 8 (mode 01 only) 01→ 16 10→ 32 11→ 64 (mode 00 only)	0x02
0x2112	CBALOAD	[6:0]	RW	Channel A manual load	0x00
0x2113	CBBLOAD	[6:0]	RW	Channel B manual load	0x00
0x2114	CBCLOAD	[6:0]	RW	Channel C manual load	0x00
0x2115	CBDLOAD	[6:0]	RW	Channel D manual load	0x00
0x2116	CBELOAD	[6:0]	RW	Channel E manual load	0x00
0x2117	CBFLOAD	[6:0]	RW	Channel F manual load	0x00
0x2118	CBGLOAD	[6:0]	RW	Channel G manual load	0x00
0x2119	CBHLOAD	[6:0]	RW	Channel H manual load	0x00

Clock Generator registers

Address	Register	Bits	Type	Description	Default
0x2200	CLKCON	[7:0]	RW	Clock generator configuration. [0]: Clock generator bypass. When bypassed, bit 6 must be set to 1 so that the output stage has a source. 0 → process 1 → bypass (saves power if no multiplier needed) [1]: Freeze the delay setting. If set to 1 the delay setting will be frozen and the output from the clock generator block will be of constant frequency. [2]: Freeze the delay when locked. 0 → do not freeze 1 → freeze the delay when the PLL reports the	0x45 *(set to 0x11 while changing registers 2201H, 2204H, and 2203H, then set value as needed as final step)*

Address	Register	Bits	Type	Description	Default
				clock output locked to the specified frequency [3]: Divide the internal pixel clock frequency by 2. Use to adjust the pixel clock and output clock (PCLKO) rates depending upon the setting for 8-bit or 16-bit parallel output. 0 → no divide 1 → divide by 2 [4]: External reset for clock generator. [5]: Turn off the internal pixel clock. To save power. [6]: Pixel clock select. Internal clock is selected from 0 → output of clock multiplier circuit 1 → output of clock divider circuit. (select this if bit 0 is set) [7]: Output clock select. Internal clock is selected from 0 → output of clock multiplier circuit 1 → output of clock divider circuit.	
0x2201	MULTSET	[7:0]	RW	Clock multiplier setting. This value is divided by 16 to get the actual multiplication factor. ***Follow the PLL lock/unlock procedure CLKCON=0x11 when changing this register.***	0x18
0x2202	MLOCKS	[0]	RO	Clock multiplier locked state. (1 = locked)	NA
0x2203	CLKDIV	[7:0]	RW	Clock division factors. 0000 = divide by 16 0001 thru 1111 = divide by 1 thru 15 [3:0]: Divide the input clock (MCLK) frequency by this factor to generate the input to the clock multiplier. ***Follow the PLL lock/unlock procedure CLKCON=0x11 when changing these bits.***	0x13
0x2204	CLKDIV2	[3:0]	RW	[3:0]: Divide the output to imager path by this factor ***.Follow the PLL lock/unlock procedure CLKCON=0x11 when changing these bits.*** 0000 = divide by 16 0001 thru 1111 = divide by 1 thru 15	0x01
0x2205	CLKDEL	[2:0]	RW	Clock delay. Delays the internal pixel clock used be the image processing blocks. One LSB delays the clock by about 2.5 ns in the typical case.	0x00

Output Formatter registers

0x2210	OUTFMT	[7:0]	RW	[0]: Parallel output format. 0 → Set this bit to ‘0’ [1]: CCIR tag format. Use only with 8-bit output format. 0 → no CCIR tags 1 → CCIR tags enabled (must be set if either 0x2230 or 0x2232 are cleared)	0x20

Address	Register	Bits	Type	Description	Default
				[2]: FVLD format 0 → Gating VSYNC output 1 → Pulsed VSYNC output. [3]: Endian format 0 → Big-endian output Y U Y V… 1 → Little-endian output. U Y V Y … [4]: Output data select. 0 → Interpolated data 1 → Bayer data. [5]: Line Buffer usage. 0 → Do not use the line buffer 1 → Use the line buffer. [6]: Bayer Truncation 0 → Bayer data at 10 bit resolution 1 → Truncate lowest 2 bit to make Bayer in 8 bits [7]: Strip extra byte in Bayer 8-bit when using line buffer 0 → Do not strip extra byte. Even 8-bit Bayer data will take two clocks to output one pixel 1 → Strip empty byte and pack two pixels in line buffer so 8-bit data takes only one clock per pixel to output.	
0x2217	TRICTL	[0]	RW	[0]: Output Tri-state control. 0 → outputs enabled 1 → outputs tri-stated	0x01
				Imager Control Interface registers	
0x2220	IMGSIG	[6:0]	RW	Imager Signals [0]: FRMPLS [6]: FLSYNC (read only)	0x00
0x2222	PLACON	0]	RW	Pixel clock latch configuration **0 = Latch the pixel data on the rising edge of output clock**. 1 = Latch the pixel data on the falling edge of output clock.	0x00
0x2228	FLHSYN	[1:0]	RW	[0]: Flash window on HSYNC configuration. **0 = none** 1 = Flash window is denoted by setting HSYNC high when VSYNC is low. [1]: Flash window on VSYNC configuration. **0 = none** 1 = Flash window is denoted by setting VSYNC high when flash window begins.	0x00
0x2229	INVSYN	[1:0]	RW	[0]: Invert VSYNC. (***not available if VSYNC shaping is being used***)	0x00

Address	Register	Bits	Type	Description	Default
				0 = disable 1 = enable [1]: Invert HSYNC. **0 = disable** 1 = enable	
0x222F	CONFIGID	[7:0]	RO	Silicon configuration ID	0x0A
				Sub-Sampling configuration registers	
0x2230	SUBBYP1	[0]	RW	Stage 1 Sub Sampling bypass. 0 = process **1 = bypass**	0x01
0x2231	SUBMOD1	[2:0]	RW	Stage 1 Sub sample mode 0 → (1/2) 1 → (1/4) 2 → (3/5:11/20) (for CIF or QCIF) 3 → (3/10:11/40) (for CIF or QCIF) 4 → (4/5) 5 → (2/3 6 → (3/4)	0x00
0x2232	SUBBYP2	[0]	RW	Stage 2 Sub Sampling bypass. 0 → process 1 → bypass	0x01
0x2233	SUBMOD2	[2:0]	RW	Stage 2 Sub sample mode Settings are the same as Stage 1	0x00
				Edge Enhancement configuration registers	
0x2240	ESHBYP	[4:0]	RW	[0]: Edge sharpening bypass. 0 → process 1 → bypass [2:1]: Edge sharpening mode. 00 → Sharpen (normal) 01 → Emboss 10 → Super Sharpen 11 → Sketch [3]: Smoothing filter [4]: De-sharpening	0x00
0x2241	ESHTLO	[7:0]	RW	Edge sharpening transfer function low threshold. – Point 1	0x00
0x2242	ESHTHI	[7:0]	RW	Edge sharpening transfer function high threshold. – Point 3	0xFF
0x2243	ESHSLP	[3:0]	RW	Edge sharpening transfer function slope.	0x00
0x2244	ESHPK	[7:0]	RW	Edge sharpening transfer function peak boost value.	0xFF
0x226D	ESMINY	[7:0]	RW	Minimum value for Y	0x00

Address	Register	Bits	Type	Description	Default
0x226E	ESMAXY	[7:0]	RW	Maximum value for Y	0xFF
0x226F	ESSUBY	[7:0]	RW	Sketch mode background intensity	0xFF

Focus Statistic configuration registers

0x2245 0x2246	AFW1SR	[10:0]	RW	Focus window starting row – window 1	0x0020
0x2247 0x2248	AFW1SC	[10:0]	RW	Focus window starting column – window 1	0x0140
0x2249 0x224A	AFW1RC	[10:0]	RW	Focus window row count – window 1	0x0040
0x224B 0x224C	AFW1CC	[10:0]	RW	Focus window column count – window 1	0x0040
0x224D 0x224E	AFW2SR	[10:0]	RW	Focus window starting row – window 2	0x0020
0x224F 0x2250	AFW2SC	[10:0]	RW	Focus window starting column – window 2	0x0200
0x2251 0x2252	AFW2RC	[10:0]	RW	Focus window row count – window 2	0x0040
0x2253 0x2254	AFW2CC	[10:0]	RW	Focus window column count – window 2	0x0040
0x2255 0x2256	AFW3SR	[10:0]	RW	Focus window starting row – window 3	0x0020
0x2257 0x2258	AFW3SC	[10:0]	RW	Focus window starting column – window 3	0x02C0
0x2259 0x225A	AFW3RC	[10:0]	RW	Focus window row count – window 3	0x0040
0x225B 0x225C	AFW3CC	[10:0]	RW	Focus window column count – window 3	0x0040
0x225D 0x225E	AFW4SR	[10:0]	RW	Focus window starting row – window 4	0x0020
0x225F 0x2260	AFW4SC	[10:0]	RW	Focus window starting column – window 4	0x0380
0x2261 0x2262	AFW4RC	[10:0]	RW	Focus window row count – window 4	0x0040
0x2263 0x2264	AFW4CC	[10:0]	RW	Focus window column count – window 4	0x0040
0x2265 0x2266	AFW1E	[15:0]	RO	Focus statistic value – window 1	--
0x2267 0x2268	AFW2E	[15:0]	RO	Focus statistic value – window 2	--
0x2269 0x226A	AFW3E	[15:0]	RO	Focus statistic value – window 3	--
0x226B 0x226C	AFW4E	[15:0]	RO	Focus statistic value – window 4	--

Address	Register	Bits	Type	Description	Default
				Flare Correction configuration registers	
0x2290	FLBYP	[0]	RW	[0]: Flare Correction Bypass. 0 → process 1 → bypass	0x01
0x2291 0x2292	FLCH1	[9:0]	RW	Channel 1 Flare term. [9]: sign bit [8:0]: value	0x0000
0x2293 0x2294	FLCH2	[9:0]	RW	Channel 2 Flare term.	0x0000
0x2295 0x2296	FLCH3	[9:0]	RW	Channel 3 Flare term.	0x0000
0x2297 0x2298	FLCH4	[9:0]	RW	Channel 4 Flare term.	0x0000
				De-Speckle configuration registers	
0x22A0	DSCON	[3:0]	RW	De-speckle configuration – *CAN NOT BE USED IF ISP SUBSAMPLING IS USED* [0]: De-speckle bypass. 0 → process 1 → bypass – *must be bypassed if color space conversion is not generating YCbCr or YUV* [1]: Luminance channel filter enable. 0 → off 1 → on [2]: Chrominance channels filter enable. 0 → off 1 → on [3]: Trim border after filtering 0 → off 1 → on	0x01
				Lens Correction configuration registers	
0x2300	LCCON	[1:0]	RW	Lens correction configuration [0]: Lens correction bypass. 0 → process 1 → bypass [1]: Lens table read enable	0x00
0x2301 0x2302	LRCEN	[9:0]	RW	Lens row center. The row number of the pixel coinciding with the lens axis.	0x0200
0x2303 0x2304	LCCEN	[9:0]	RW	Lens column center. The column number of the pixel coinciding with the lens axis.	0x0280

Address	Register	Bits	Type	Description	Default
0x2305 0x2306	ISQRIN	[10:0] only 10 bits on read	RW	Initial square root input. This value is equal to Integer part of SQRT(pow(LRCEN, 2.0) + pow(LCCEN, 2.0))	0x0333
0x2307 0x2308H	ISRERR	[10:0] only 10 bits on read	RW	Initial square root error. This value is equal to pow(LRCEN, 2.0) + pow(LCCEN, 2.0) – pow(ISQRIN, 2.0)	0x03D7
0x2309	ISSCON	[3:0]	RW	Imager sub-sampling configuration. [1:0]: Horizontal sub-sampling 00 or 11 → No sub-sampling in imager. 01 → read 2 skip 2 sub-sampling in imager. 10 → Read 2 skip 6 sub-sampling in imager. [3:2]: Vertical sub-sampling 00 or 11 → No sub-sampling in imager. 01 → read 2 skip 2 sub-sampling in imager. 10 → Read 2 skip 6 sub-sampling in imager.	0x00
0x2310 0x2311	RPG040	[11:0]	RW	Gain for red pixels at a distance of 0x040 pixels from the center pixels.	0x0400
0x2312 0x2313	RPG080	[11:0]	RW	Gain for red pixels at a distance of 0x080 pixels from the center pixels.	0x0400
0x2314 0x2315	RPG0C0	[11:0]	RW	Gain for red pixels at a distance of 0x0C0 pixels from the center pixels.	0x0400
0x2316 0x2317	RPG100	[11:0]	RW	Gain for red pixels at a distance of 0x100 pixels from the center pixels.	0x0400
0x2318 0x2319	RPG140	[11:0]	RW	Gain for red pixels at a distance of 0x140 pixels from the center pixels.	0x0400
0x231A 0x231B	RPG180	[11:0]	RW	Gain for red pixels at a distance of 0x180 pixels from the center pixels.	0x0400
0x231C 0x231D	RPG1C0	[11:0]	RW	Gain for red pixels at a distance of 0x1C0 pixels from the center pixels.	0x0400
0x231E 0x231F	RPG200	[11:0]	RW	Gain for red pixels at a distance of 0x200 pixels from the center pixels.	0x0400
0x2320 0x2321	RPG240	[11:0]	RW	Gain for red pixels at a distance of 0x240 pixels from the center pixels.	0x0400
0x2322 0x2323	RPG280	[11:0]	RW	Gain for red pixels at a distance of 0x280 pixels from the center pixels.	0x0400
0x2324 0x2325	RPG2C0	[11:0]	RW	Gain for red pixels at a distance of 0x2C0 pixels from the center pixels.	0x0400
0x2326 0x2327	RPG300	[11:0]	RW	Gain for red pixels at a distance of 0x300 pixels from the center pixels.	0x0400
0x2328	RPG340	[11:0]	RW	Gain for red pixels at a distance of 0x340 pixels from	0x0400

Address	Register	Bits	Type	Description	Default
0x2329				the center pixels.	
0x232A 0x232B	RPG380	[11:0]	RW	Gain for red pixels at a distance of 0x380 pixels from the center pixels.	0x0400
0x232C 0x232D	RPG3C0	[11:0]	RW	Gain for red pixels at a distance of 0x3C0 pixels from the center pixels.	0x0400
0x232E 0x232F	RPG400	[11:0]	RW	Gain for red pixels at a distance of 0x400 pixels from the center pixels.	0x0400
0x2330 0x2331	RPG440	[11:0]	RW	Gain for red pixels at a distance of 0x440 pixels from the center pixels.	0x0400
0x2340/41 thru 0x2360/61	GPG040 thru GPG440	[11:0]	RW	Gain for green pixels at a distance of 0x040 pixels from the center pixels thru 0x440 pixels from the center	0x0400
0x2370/71 thru 0x2390/91	BPG040 thru BPG440	[11:0]	RW	Gain for blue pixels at a distance of 0x040 pixels from the center pixels thru 0x440 pixels from the center	0x0400

Gamma Correction configuration registers

0x23B0	GCBYP	[0]	RW	[0]: Gamma correction bypass.	0x00
0x23B1	GTV08	[8:0]	RW	Gamma table value for index 0x08. This is the output value if input value is 0x08.	0x1C
0x23B2	GTV10	[7:0]	RW	Gamma table value for index 0x10.	0x30
0x23B3	GTV20	[7:0]	RW	Gamma table value for index 0x20.	0x4B
0x23B4	GTV30	[7:0]	RW	Gamma table value for index 0x30.	0x61
0x23B5	GTV40	[7:0]	RW	Gamma table value for index 0x40.	0x73
0x23B6	GTV50	[7:0]	RW	Gamma table value for index 0x50.	0x83
0x23B7	GTV60	[7:0]	RW	Gamma table value for index 0x60.	0x92
0x23B8	GTV70	[7:0]	RW	Gamma table value for index 0x70.	0xA0
0x23B9	GTV80	[7:0]	RW	Gamma table value for index 0x80.	0xAD
0x23BA	GTV90	[7:0]	RW	Gamma table value for index 0x90.	0xB9
0x23BB	GTVA0	[7:0]	RW	Gamma table value for index 0xA0.	0xC4
0x23BC	GTVB0	[7:0]	RW	Gamma table value for index 0xB0.	0xCF
0x23BD	GTVC0	[7:0]	RW	Gamma table value for index 0xC0.	0xD9
0x23BE	GTVD0	[7:0]	RW	Gamma table value for index 0xD0.	0xE4
0x23BF	GTVE0	[7:0]	RW	Gamma table value for index 0xE0.	0xED

Format conversion registers

0x2400 duplicated at 0x20C0	RTYCON	[1:0]	RW	[0]: RGB to YCbCr conversion bypass. If this bit is set to 1, then the conversion is bypassed. If any non-Bayer output is desired, this bit needs to be enabled. [1]: Set this bit to 1 if sub sampling is enabled	0x00

Address	Register	Bits	Type	Description	Default
				(SUBBYP=0), otherwise clear it.	
0x20C1	YTYCON	[0]	RW	[0]: YCbCr to YUV conversion bypass. If this bit is set to 1, then the conversion is bypassed. This conversion should be enabled only if YUV output is desired and it was not directly generated by the conversion from RGB already. *THIS FUNCTION IS NOT NORMALLY USED*	0x01
0x20C2	Y2RCON	[3:0]	RW	[0]: YCbCr to RGB conversion bypass. 0 → process 1 → bypass [2:1]: conversion configuration 00 → Output in RGB565 format. 01 → Output RGB444 followed by four bits of 0s. 10 → Output four bits of zeros followed by RGB444. [3]: Set this bit to 1 if sub sampling is enabled (SUBBYP=0), otherwise clear it. *GENERATING RGB FORMATS WITH THIS FUNCTION ASSUMES THAT THE MATRIX OPERATION PRODUCED YCBCR*	0x01
0x2408 0x2409	RTYYR	[8:0]	W	[7:0]: Red multiplier for Y calculation. Matrix entry Y_r [8]: Sign bit. 1 = negative	0x004D
0x240A 0x240B	RTYYG	[8:0]	W	[7:0]: Green multiplier for Y calculation. Matrix entry Y_g [8]: Sign bit	0x0096
0x240C 0x240D	RTYYB	[8:0]	W	[7:0]: Blue multiplier for Y calculation. Matrix entry Y_b [8]: Sign bit	0x001D
0x240E	RTYYO	[7:0]	W	Constant offset for Y calculation	0x00
0x2410 0x2411	RTYCBR	[8:0]	W	[7:0]: Red multiplier for Cb calculation. Matrix entry Cb_r [8]: Sign bit	0x012B
0x2412 0x2413	RTYCBG	[8:0]	W	[7:0]: Green multiplier for Cb calculation. Matrix entry Cb_g [8]: Sign bit	0x0155
0x2414 0x2415	RTYCBB	[8:0]	W	[7:0]: Blue multiplier for Cb calculation. Matrix entry Cb_b [8]: Sign bit	0x0080
0x2416	RTYCBO	[7:0]	W	Constant offset for Cb calculation	0x80
0x2418 0x2419	RTYCRR	[8:0]	W	[7:0]: Red multiplier for Cr calculation. Matrix entry Cr_r [8]: Sign bit	0x0080
0x241A 0x241B	RTYCRG	[8:0]	W	[7:0]: Green multiplier for Cr calculation. Matrix entry Cr_g	0x016B

Address	Register	Bits	Type	Description	Default
				[8]: Sign bit	
0x241C 0x241D	RTYCRB	[8:0]	W	[7:0]: Blue multiplier for Cr calculation. Matrix entry Cr_b [8]: Sign bit	0x0115
0x241E	RTYCRO	[7:0]	W	Constant offset for Cr calculation	0x80

Imager registers

(in all registers in this group, unused bits will read back as '1')

Address	Register	Bits	Type	Description	Default
0x2501	GNRB	[1:0]	RW	Red and Blue Channel gain. 00 = 1x 01 = 4x 10 = 2x 11 = 8x	0x00
0x2503	GNG12	[1:0]	RW	Green1 and Green2 Channel gain.	0x00
0x2505	reserved	[7:0]	RW		0x08
0x2506	reserved	[2:0]	RW		0x04
0x2507	ATESTMODE	[7:0]	RW	[6:0]: Reserved [7]: VSYNC/HSYNC Pad assignments: 0 = Normal 1 = Swapped (VSYNC <-> HSYNC)	0x00
0x2508	EXPMSB	[7:0]	RW	Exposure time MSB. Units are 1 line time of readout timing. So actual exposure time in seconds scales with clock NOTE this register and 0x2509 are the exception to "LSB in lower address rule."	0x04
0x2509	EXPLSB	[7:0]	RW	Exposure time LSB	0xC0
0x250A	SCANMODE	[7:0]	RW	Scan direction and size control. [0]: Horizontal mirror: 0 = no mirror 1 = mirror [2:1]: Horizontal sub sample: 00 = no sub sample 01 = R2S2 10 = R2S6 11 = do not use [3]: Vertical phantom rows 0 = 2 rows 1 = 8 rows [4]: Vertical flip: 0 = no flip 1 = flip [6:5]: Vertical sub sample: 00 = no sub sample	0x00

Address	Register	Bits	Type	Description	Default
				01 = R2S2 10 = R2S6 11 = do not use [7]: CDS length 0 = 208 clocks 1 = 104 clocks	
0x250B	SSCTL	[2:0]	RW	SuperSaturation Control [0]: Replacement Enable: 0 = Disable 1 = Enable (detection must be enabled as well) [2:1]: Replacement scheme 00 = Replace detected pixel only 01 = test only 10 = test only 11 = Replace detected +/- 1 pixel	0x00
0x250D	reserved	[7:0]	RW		0x00
0x250E	WNDWMODE	[7:0]	RW	Window Mode Control [2:0]: Start at section: Vertical Scan FWD REV section 1 = 000 = 000 section 2 = 001 = 001 section 3 = 010 = 010 section 4 = 011 = 011 section 5 = 100 = 100 section 6 = 101 = 101 section 7 = 110 = 110 section 8 = 111 = 111 [5:3]: End at section: Vertical Scan FWD REV section 0 = n/a = 111 section 1 = n/a = 000 section 2 = 001 = 001 section 3 = 010 = 010 section 4 = 011 = 011 section 5 = 100 = 100 section 6 = 101 = 101 section 7 = 110 = 110 section 8 = 111 = 111 section 9 = 000 = n/a [6]: Vertical window enable: 0 = disable - full readout 1 = enable (see bits [5:0]) [7]: Horizontal window enable 0 = disable - full readout 1 = enable - 1312 columns (24+1288)	0x00
0x2510	STALLMODE	[7:0]	RW	Stall Mode and HBlank Length	0x00

Address	Register	Bits	Type	Description	Default
				[2:0] JPEG Engine Stall Mode: 000 = No stall 001 = Prerow stall 010 = SHS stall 011 = SHR stall 100 = SHS-SHR stall 101 = Reset stall 110 = Reset extend stall 111 = No stall [7] HBlank Length: 0 = CDS Length plus 80 clocks (see 0x250A[7]) 1 = Fixed at 208+80 clocks	
0x251A	PREDLCG12	[3:0]	RW	Pre Gain Dark Level Control for channels Green 1 and Green 2	0x00
0x251B	PREDLCRB	[3:0]	RW	Pre Gain Dark Level Control for channels Red and Blue	0x00
0x251C	PREDLCM	[3:0]	RW	Pre Gain Dark Level Control MODE [0]: PREDLCG12 sign bit: 0 = positive 1 = negative [1]: PREDLCRB sign bit: 0 = positive 1 = negative [2]: Auto vs Manual operation: 0 = Auto 1 = Manual [3]: Enable vs Bypass 0 = Enable 1 = Bypass	0x00
0x251E	reserved	[7:0]	RW		0x77
0x2521	SSTHRESH	[7:0]	RW	SuperSaturation Threshold [1:0]: Green1/2 threshold 00 lowest 10 (bit 1is lsb) 01 (bit 0 is msb) 11 highest [4:3]: Red/Blue threshold [7:6]: Detection Enable: 00 = Disabled 10 = Enabled	0x00
0x2523	PSTDLCG12	[7:0]	RW	Post Gain Dark Level correction for channels Green 1 and Green 2	0x00
0x2524	PSTDLCRB	[7:0]	RW	Post Gain Dark Level correction for channels Red and	0x00

Address	Register	Bits	Type	Description	Default
				Blue	
0x2525	PSTDLCM	[3:0]	RW	Post Gain Dark Level Control MODE [0]: PSTDLCG12 sign bit: 0 = positive 1 = negative [1]: PSTDLCRB sign bit: 0 = positive 1 = negative [2]: Auto vs Manual operation: 0 = Auto 1 = Manual [3]: Enable vs Bypass 0 = Enable 1 = Bypass	0x00
0x2527	reserved	[6:0]	RW		0x09
0x2528	reserved	[3:1]	RW		0x00
0x2529	reserved	[2:0]	RW		0x00
0x252A	DLCTRGT	[7:0]	RW	Dark level correction target value	0x00
0x252B	DLCMODE	[4:0]	RW	Dark level initial mode control [2:0]: Starting dark row for sampling: [3]: max on zero average: 0 = Normal 1 = Jump to max [4]: DLC Freeze: 0 = Normal 1 = Freeze	0x00
0x252C	DLCBASE1	[4:0]	RW	Converged range	0x05
0x252D	DLCBASE2	[4:0]	RW	Recomputed range	0x0A
0x252E	PSTDLCMAX	[7:0]	RW	Maximum DAC setting for post gain DLC	0xFF
0x252F	DLCSTEP	[6:0]	RW	Pre and post gain DLC step sizes [4:0]: Post gain DLC step [6:5]: Pre gain DLC step	0x00
0x255F	reserved	[3:0]	RO		NA
0x2560	CHIPID	[7:0]	RO	Bi8921 = 0x18, Bi8921T = 0x1D, Final Production spec=0x27	0x27
0x257F	URC	[7:0]	W	Causes imager block internal registers to be updated synchronously with newest settings. No data byte is required but may be supplied. It is ignored if supplied.	NA

JPEG Compression Registers

Address	Register	Bits	Type	Description	Default
0x2600	JPGCON	[7:0]	RW	[0]: JPEG convert bypass 0 = process 1 = bypass [1]: Quantization table read access: 0 = disable 1 = enable read access [2]: Clock gating enable: 0 = disable – always allow PCLKO output 1 = Output PCLKO only when JPEG data is valid [4:3]: JPEG error indication: 00 = HSYNC not used to indicate error 01 = HSYNC reflects JPEG error state. 10 = error indicated by extending HSYNC 11 = error indicated by adding 4 bytes at end [5]: Dummy byte stuff enable: 0 = disable 1 = enable padding so HSYNC is fixed length [6]: Embedded thumbnail enable: 0 = disable 1 = enable thumbnail insertion. [7]: JPEG Compression Status (Read Only): 0 = No error 1 = Error	0x00
0x2601 0x2602	IMGROW	[10:0]	RW	Number of rows in the image to be compressed.	0x4B0
0x2603 0x2604	IMGCOL	[10:0]	RW	Number of columns in the image to be compressed.	0x640
0x2605	QUNCON	[3:0]	RW	Quantization table configuration [0]: Quantization table select. 1 = the quantization table is loaded from the quality factor set. (bit [3:1]) 0 = Values in quantization table are used. [3:1]: Quantization table quality factor. 000 = Use default table * 2 001 = Use default quantization table 010 = Use default table / 2 011 = Use default table / 4 100 = Use default table / 8	0x09
0x2606	QTABLA	[6:0]	RW	Quantization table load address. This address is automatically incremented every time there is a write to QTABLD. This is the table index at which to start loading terms.	0x00

Address	Register	Bits	Type	Description	Default
0x2607	QTABLD	[7:0]	RW	Quantization table load data.	0x00
0x2609	JCLKDIV	[3:0]	RW	Clock division factor for JPEG data. The YUV clock frequency is divided by this factor to generate the JPEG output clock. If set to 0, the clock frequency division factor is 16. Otherwise it represents a value between 1 and 15. Set this to 1 if the JPEG block is bypassed.	0x01
0x260A	HSNEXT	[3:0]	RW	Number of cycles to extend the HSYNC in case of error. Has effect only if JPGCON [4:3] = 2.	0x08
0x260B 0x260C 0x260D 0x260E	JEBYTE	[23:0]	RW	Bytes to be output to indicate error. Has effect only if JPGCON [4:3] = 3.	0x0000
0x260F 0x2610	VLDCLK	[11:0]	RW	JPEG output clock cycles per HSYNC period. This has effect only if JPGCON[5] = 1.	0x00
0x2611 0x2612	BLKCLK	[11:0]	RW	Blank clock cycles per row. This has effect only if JPGCON[5] = 1.	0x00
0x2613	SMPDIS	[3:0]	RW	Sampling distance for the thumbnail generation. This determines the size of the output thumbnail also. Ex. For an 1600x1200 image, if the sampling distance is chosen to be 5, the generated thumbnail will be of size 320x240. If the sampling distance is less than 7, horizontal and vertical averaging is used to generate thumbnail, otherwise only horizontal averaging is used.	0x0A
0x2614 0x2615	SMPROW	[8:0]	RW	Samples per row. Number of pixels of thumbnail data output per HSYNC. Note that the number of thumbnail bytes output per HSYNC will be double this number.	0x10
0x2616	HEDBEG	[2:0]	RW	Header begin row. The internal YUV rows are counted 0, 1, 2 … and when the count reaches this number the JPEG output starts.	0x07
0x2617 0x2618	HSNCNT	[10:0]	RW	Number of HSYNC’s in the compressed image output.	0x4B0
0x2619	SRSHPC	[2:0]	RW	HSYNC and VSYNC reshaping configuration. [0] Start VSYNC early 0 = disable 1 = enable [1] Pass through VSYNC (keep the VSYNC of the uncompressed image itself). Used for Pulsed VSYNC 0 = disable 1 = enable [2] Reshape VSYNC and HSYNC. Is used for setting a constant amount of clock cycle delay between the rising edge of VSYNC and the rising edge of HSYNC. 0 = disable 1 = enable	0x00

Address	Register	Bits	Type	Description	Default
0x261A 0x261B 0x261C	HSNDLY	[23:0]	RW	Number of output clock cycle delay from the rising edge of VSYNC to rising edge of HSYNC. Enable both bits 0 and 2 of 0x2619 when using this.	0x00
0x261D 0x261E 0x261F	OUTLIM	[23:0]	RW	Maximum number of output bytes that the chip should output. This number must be a multiple of 8. If the output byte count exceeds this number, an error is generated and the output is truncated to the specified number of bytes. Set this to 0x000000 to disable this feature.	0x00
0x2620 0x2621 0x2622	BYTCNT	[23:0]	RO	Number of output bytes contained in the most recently output frame.	--

Document Applicability:

Data Sheet Version	Parts Covered	Comments
1.0	Bi8921	First Release
2.0	Bi8921	Major revision for miscellaneous corrections, see change bars.
3.0	Bi8921T	Revised for production spec. Added De-speckle filter, tri-state option, and modified some JPEG functions.
3.1	Bi8921T	Swapped terminology of YUV and YCbCr to align with standards.
3.2	Bi8921T	Revised for final production spec: signal timing details, JPEG marker error, registers 0x250A, 0x2510, and 0x2560

In Taiwan (Headquarters):

Biomorphic Microsystems Corporation
6F, No. 6, Technology Road 5
Hsinchu Science Park
Hsinchu, Taiwan ROC 302
Voice: +886 (3) 668-6999, fax +886 (3) 668-6069

In United States:

Biomorphic VLSI, Inc
123 Hodencamp Road
Suite 204
Thousand Oaks, Ca 91360
Voice: 1 805.497.9055, fax 1 805.497.9725
info@biomorphic.com http://www.biomorphic.com/

In Korea:

Biomorphic VLSI
Room No.813, Hanseo office,
11-11, Yoido-Dong,
Youngdeungpo-Ku
Seoul, Korea 319
Tel : +82(2) 784-1867
Fax : +82(2) 784-1868