1/4-Inch 5Mp System-On-A-Chip (SOC) CMOS Digital Image Sensor

MT9P111

For the latest data sheet, refer to Aptina Imaging's Web site: www.aptina.com

Features

- DigitalClarity® CMOS imaging technology
- Superior low-light performance
- Ultra-low-power, low-cost
- Anti-shake support
- One time programmable (OTP) memory for automatic positional gain adjustments and other uses
- Parallel data output and serial mobile industry processor interface (MIPI) data output
- Integrated real-time JPEG encoder
- Flexible support for external auto focus
- Internal master clock generated by on-chip phase-locked loop (PLL) oscillator with dithering
- Electronic rolling shutter (ERS), progressive scan
- Integrated image flow processor (IFP) for single-die camera module
- Automatic image correction and enhancement
- Selectable output data format: YCbCr, 565RGB, 555RGB, 444RGB, JPEG 4:2:2, processed Bayer, RAW8- and RAW10-bit
- Output FIFO for data rate equalization
- Programmable I/O slew rate
- Xenon and LED flash support with fast exposure adaptation
- Configurable gamma correction based on scene brightness
- Arbitrary image scaling with anti-aliasing
- Two-wire serial interface providing access to registers and microcontroller memory, additional serial interface under user control
- Includes internal VCM driver and access to internal A/D converter

Applications

- Cellular phones
- PC cameras
- PDAs

Table 1: Key Performance Parameters

Parameter		Value
Optical format		1/4-inch
Full resolution		2592 x 1944 pixels
Pixel size		1.4 µm x 1.4 µm
Dynamic range		60 dB
SNR MAX		34.7 dB
Responsivity		0.5 V/lux-sec (preliminary)
Chief ray angle		25.03° MAX at 80% image height
Color filter array		RGB Bayer pattern
Active pixel array area		3.62 mm x 2.72 mm
Shutter type		Electronic rolling shutter (ERS) and Global reset release (GRR)
Input clock frequency		8–54 MHz
Maximum frame rate		15 fps at full resolution (JPEG), 30 fps in preview mode
Maximum pixel data output		MIPI: 768 Mb/s MAX Parallel: 96Mp/s
Maximum pixel clock frequency		96 MHz
Supply voltage	Analog	2.5–3.1V
	Digital	1.7–1.95V
	I/O	1.7–1.9V or 2.8–3.1V
	PLL	2.5–3.1V
	MIPI	2.5–3.1V
ADC resolution		12-bit, on-die
Power consumption		550 mW at 30 fps, 1280 x 720 video mode
		300 mW at 30 fps, preview mode
Current consumption		10 µA, shutdown, at +70°C
Operating temperature (at junction)		–30°C to +70°C

Ordering Information

Table 2: Available Part Numbers

Part Number	Description
MT9P111D00STCK28AC1	Bare die

Table of Contents

List of Figures

List of Tables

MT9P111 Overview

The MT9P111 has a color image sensor with a Bayer color filter arrangement and a 5Mp active-pixel array with electronic rolling shutter (ERS). The sensor core readout is 12-bit, supports skipping and binning, and can be flipped and/or mirrored. The sensor core also supports separate analog and digital gain for all four color channels (R, Gr, Gb, B).

The MT9P111 also has an embedded phase-locked loop oscillator (PLL) that can generate the internal sensor clock from the common clock signals available in typical mobile phone systems. When in use, the PLL adjusts the incoming clock frequency up, allowing the MT9P111 to run at almost any desired resolution and frame rate within the sensor's capabilities. The PLL can be bypassed and powered down to reduce power consumption. The PLL also has a dithering feature that can be used to reduce overall radiated EMI.

The MT9P111 has numerous power-conserving features including a soft standby mode and a hard standby mode. In standby mode, the sensor can be configured to consume less power than normal operation, with the option of retaining the internal configuration settings. By default, entering standby disables the internal V_{DD} power rail. In addition, there is a SHUTDOWN mode that will disable the power supplies in order to achieve the lowest power consumption possible.

The MT9P111 can be used with either a serial MIPI interface or the parallel data output interface, which has a programmable I/O slew rate to minimize EMI and an output FIFO to eliminate output data bursts. JPEG format can be output in both the MIPI and the parallel data output interfaces. EXIF, MIPI data type support is also included, along with Scalado support.

The advanced image flow processor (IFP) and flexible programmability of the MT9P111 provide a variety of ways to enhance and optimize the image sensor performance. Built-in optimization algorithms enable the MT9P111 to operate at factory settings as a fully automatic, highly adaptable camera; however, most of its settings are user-programmable.

These algorithms include black level conditioning, shading correction, defect correction, noise reduction, color interpolation, color correction, aperture correction, and image formatting such as cropping and scaling.

The MT9P111 also includes a sequencer that coordinates all events triggered by the user. The sequencer manages auto focus, auto white balance, flicker detection, anti-shake, and auto exposure for the different operating modes, which include preview, still capture, video, and snapshot with flash.

All modes of operation are individually configurable and are organized as two contexts. A context is defined by sensor image size, frame rate, resolution, and other associated parameters. The user can switch between the two contexts by sending a command through the two-wire serial interface.

A two-wire serial register interface bus enables read/write access to control registers, variables, and special function registers within the MT9P111. The hardware registers include sensor core controls, color pipeline controls, and output controls.

The general purpose VGPIO can be configured to allow the user extended platform functionality or achieve a 10-bit parallel Bayer output.

Signal Description

Table 3 provides the signal descriptions for the MT9P111.

Table 3: Signal Descriptions

Name	Type	Description	Notes
STANDBY	Input	Controls sensor's standby mode, active HIGH.	
RESET_BAR	Input	Master reset signal, active LOW (can be left floating if not used).	
SHUTDOWN	Input	Complete shutdown function for lowest power state	
EXTCLK	Input	Input clock signal 6–54 MHz.	
VPP	Input	High voltage programming pin for one-time programmable (OTP) memory (must be left floating for normal operation).	
SCLK	Input	Slave two-wire serial interface clock from the host processor.	
SADDR	Input	Selects device address for the slave two-wire serial interface. The address is 0x78 when SADDR is tied LOW, 0x7A if tied HIGH.	
SDATA	I/O	Slave two-wire serial interface data to and from the host processor.	
S_SCL	Output	Master two-wire serial interface clock to peripheral devices like AF mechanics.	
S_SDA	I/O	Master two-wire serial interface data to peripheral devices like AF mechanics.	
VGPIO[7:0]	I/O	General purpose digital I/O, used for auto focus function (can be left floating if not used).	
DOUT[7:0]	Output	8-bit image data output or most significant bits (MSB) of 10-bit SOC bypass mode.	
PIXCLK	Output	Pixel clock. Used for sampling DOUT, FRAME_VALID, and LINE_VALID.	
LINE_VALID	Output	Identifies pixels in the active line.	
FRAME_VALID	Output	Identifies rows in the active image.	
DOUT_N	Output	Differential MIPI data (sub-LVDS, negative) (must be left floating if not used).	
DOUT_P	Output	Differential MIPI data (sub-LVDS, positive) (must be left floating if not used).	
CLK_N	Output	Differential MIPI clock (sub-LVDS, negative) (must be left floating if not used).	
CLK_P	Output	Differential MIPI clock (sub-LVDS, positive) (must be left floating if not used).	
VCM_OUT	I/O	VCM actuator driver pad	
VCM_GND	I/O	Ground pad for VCM_OUT	
ATEST0	I/O	Internal ADC access	
ATEST1	I/O	Internal ADC access	
VDD	Supply	Digital power (1.8V typical).	
VAA_PIX	Supply	Pixel array power (2.8V typical).	
VAA	Supply	Analog power (2.8V typical).	
VDD_PLL	Supply	PLL power (2.8V typical).	
VDD_IO	Supply	I/O power supply (1.8V or 2.8V typical).	
GND_IO	Supply	I/O ground.	
DGND	Supply	Digital ground.	1
AGND	Supply	Analog ground.	1
VDDIO_TX	Supply	I/O power supply for the MIPI output interface, 2.8V typical, (Can be disconnected if the interface is not used).	
GNDIO_TX	Supply	I/O ground supply for the MIPI output interface (Can be disconnected if the interface is not used).	
VDD_VGPIO	Supply	I/O power supply for VGPIO[7:0] signals (can be left unconnected if the interface is not used).	
GND_VGPIO	Supply	I/O ground for VGPIO[7:0].	

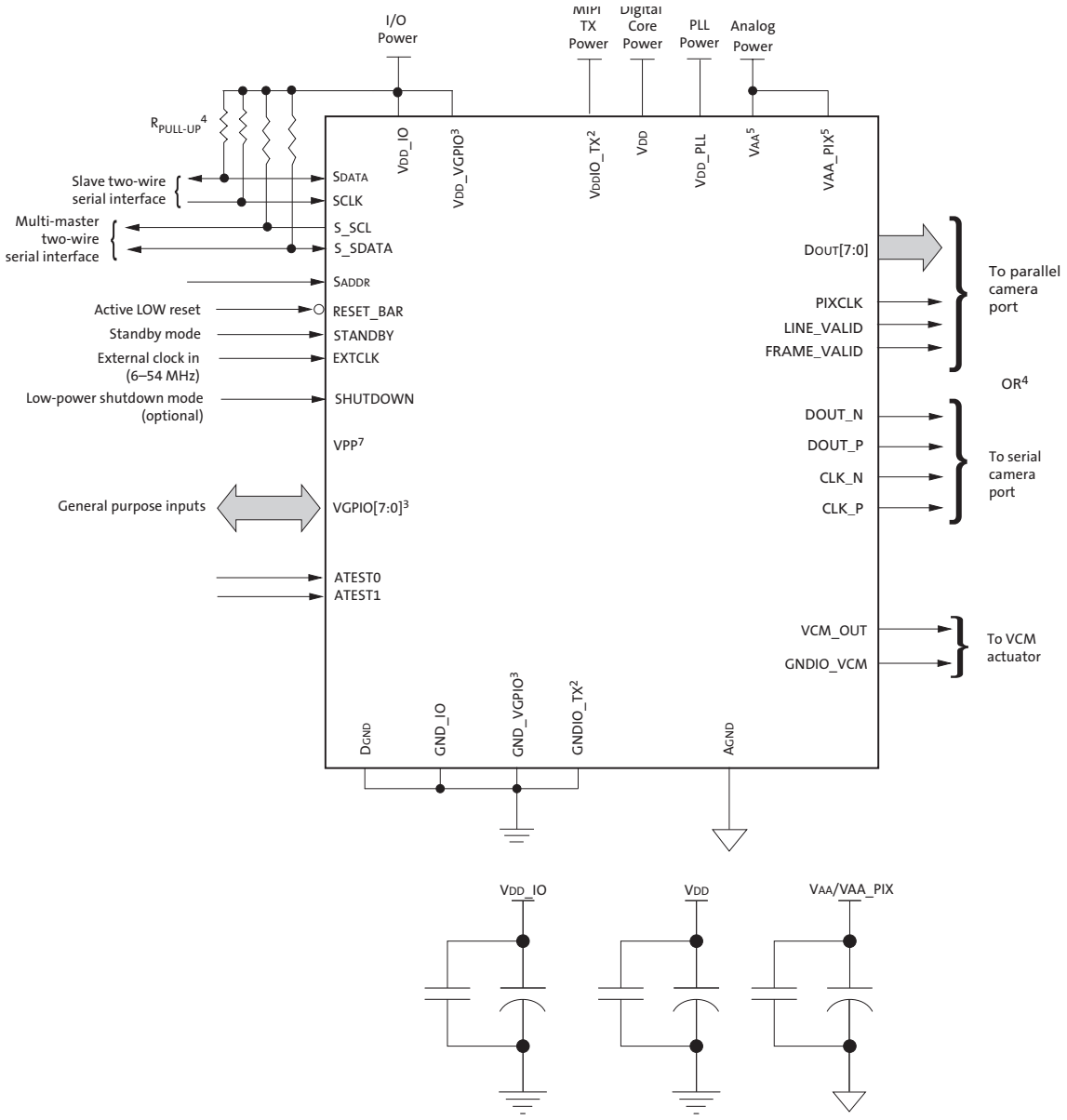
Notes: 1. AGND and DGND are not connected internally (inside the chip).

Typical Connections

Figure 1 shows typical MT9P111 device connections. For low-noise operation, the MT9P111 requires separate power supplies for analog and digital. Incoming digital and analog ground conductors can be tied together next to the die. Both power supply rails should be decoupled from ground using capacitors as close as possible to the die. Aptina does not recommend the use of inductance filters on the power supplies or output signals.

The MT9P111 supports different digital core (VDD/DGND), MIPI output (VDDIO_TX/GNDIO_TX), and I/O (VDD_IO/GND_IO) power domains that can be at different voltages. The PLL requires a clean power source (VDD_PLL).

Figure 1: Typical Configuration (connection)



Notes:
1. This typical configuration shows only one scenario out of multiple possible variations for this sensor. The minimum recommended decoupling configuration is 0.1µF per supply on module and 10µF off module.
1. If a MIPI Interface is not required, the following pads must be left floating: DOUT_P, DOUT_N, CLK_P, and CLK_N. VDDIO_TX and GNDIO_TX can also be left floating.
2. The VGPIO pads can serve multiple features that can be reconfigured. The function and direction will vary by applications. If VGPIO pads are not required, the VDD_VGPIO, GND_VGPIO and VGPIO[7:0] pads can be left floating.
3. Only one of the output modes (serial or parallel) can be used at any time.
4. Aptina recommends a resistor value of 1.5KΩ to VDD_IO for the two-wire serial interface RPULL-UP; however, greater values may be used for slower transmission speed.
5. VAA and VAA_PIX must be tied together.

6. VPP is the one-time programmable (OTP) memory programming voltage and should be left floating during normal operation.
7. VDDIO_TX can be connected to VDD_IO if VDD_IO=2.8 Volts. Otherwise, VDDIO_TX can be tied to the VAA supply if Aptina recommended decoupling capacitor is used. VDDIO_TX can be left floating if MIPI is not used.

Decoupling Capacitor Recommendations

It is important to provide clean, well regulated power to each power supply. The Aptina recommendation for capacitor placement and values are based on our internal demo camera design and verified in hardware.

Note: Since hardware design is influenced by many factors, such as layout, operating conditions, and component selection, the customer is ultimately responsible to ensure that clean power is provided for their own designs.

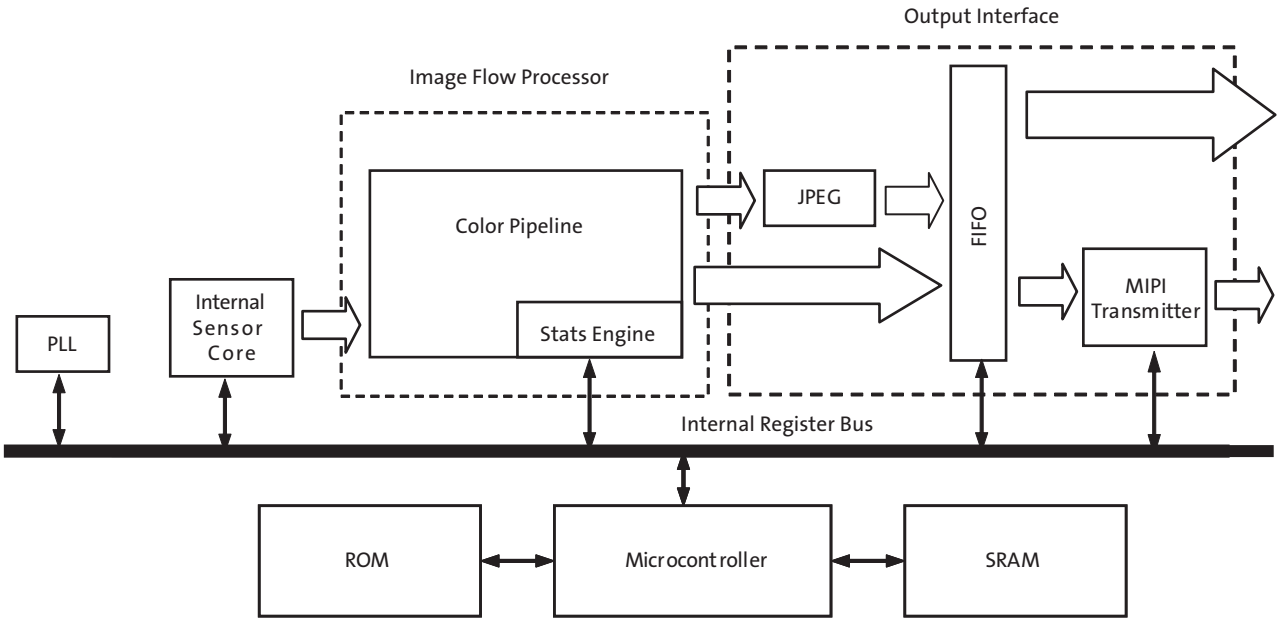
In order of preference, Aptina recommends:

1. Mount 0.1µF and 1µF decoupling capacitors for each power supply as close as possible to the pad and place a 10 µF capacitor nearby off-module.
2. If module limitations allow for only six decoupling capacitors for a three-regulator design (VDD_PLL tied to VDD), use a 0.1µF and 1µF capacitor for each of the three regulated supplies. Aptina also recommends placing a 10µF capacitor for each supply off-module, but close to each supply.
3. If module limitations allow for only three decoupling capacitors, a 1µF capacitor for each of the three regulated supplies is preferred. Aptina recommends placing a 10µF capacitor for each supply off-module but closed to each supply."
4. If module limitations allow for only three decoupling capacitors, a 0.1µF capacitor for each of the three regulated supplies is preferred. Aptina recommends placing a 10µF capacitor for each supply off-module but close to each supply.
5. Priority should be given to the VAA supply for additional decoupling capacitors.
6. Inductive filtering components are not recommended.
7. Follow best practices when performing physical layout. Refer to technical note TN-09-131.

Architecture Overview

The MT9P111 combines a 5Mp sensor core with an image flow processor (IFP) to form a stand-alone solution that includes both image acquisition and processing. Both the sensor core and the IFP have internal registers that can be controlled by the user. In normal operation though, an integrated microcontroller autonomously controls most aspects of operation. The processed image data is transmitted to the host system either through a parallel bus or a serial data interface through the output interface.

Figure 2: SOC Block Diagram



Sensor Core Description

The sensor core of the MT9P111 is a progressive-scan sensor that generates a stream of pixel data at a constant frame rate, qualified by LINE_VALID (LV) and FRAME_VALID (FV). The maximum pixel rate is 96 Mp/s, corresponding to a pixel clock rate of 96 MHz. Figure 3 on page 14 shows a block diagram of the sensor core. It includes a 5Mp active-pixel array. The timing and control circuitry sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and reading that row, the pixels in the row integrate incident light. The exposure is controlled by varying the time interval between reset and readout. Once a row has been read, the data from the columns is sequenced through an analog signal chain (providing offset correction and gain), and then through an ADC. The output from the ADC is a 12-bit value compressed to a 10-bit value for each pixel in the array.

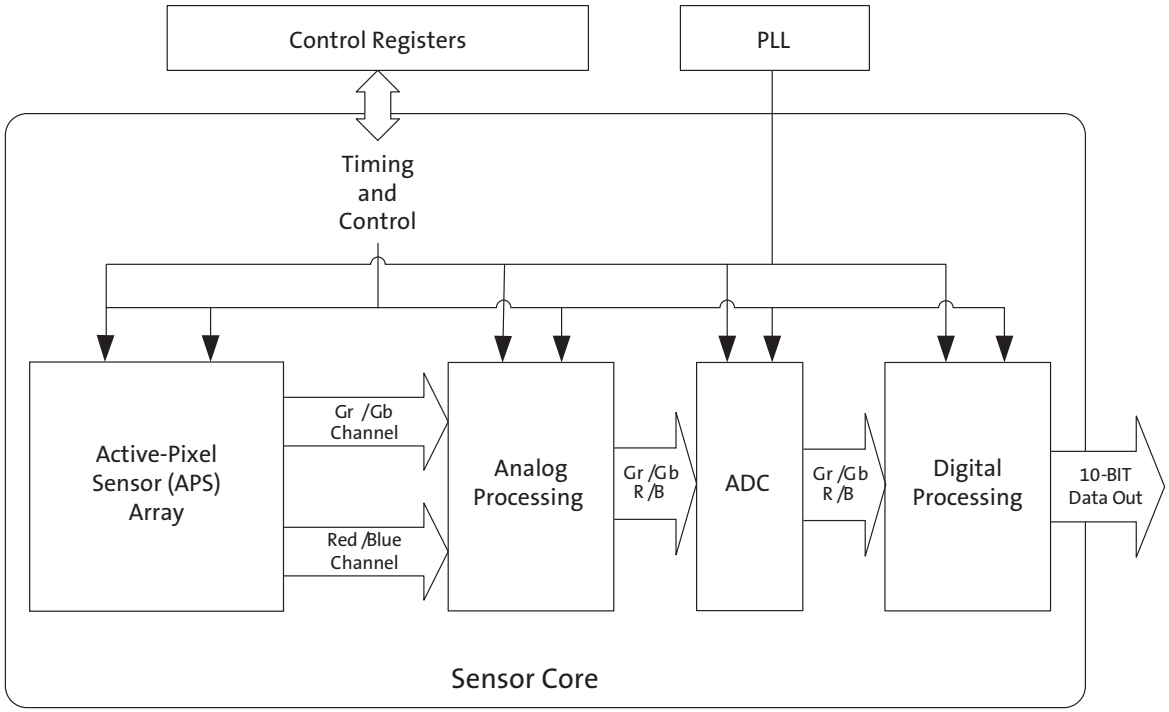
The pixel array contains optically active and light-shielded (dark) pixels. The dark pixels are used to provide data for the offset-correction algorithms (black level control).

The sensor core contains a set of control and status registers that can be used to control many aspects of the sensor behavior including the frame size, exposure, and gain setting. These registers are controlled by the SOC firmware and can be accessed through a two-wire serial interface. Register values written to the sensor core maybe overwritten by firmware.

The output from the core is a Bayer pattern; alternate rows are a sequence of either green and red pixels or blue and green pixels. The offset and gain stages of the analog signal chain provide per-color control of the pixel data.

A flash strobe output signal is provided to allow an external xenon or LED light source to synchronize with the sensor exposure time. Additional I/O signals support the provision of an external mechanical shutter.

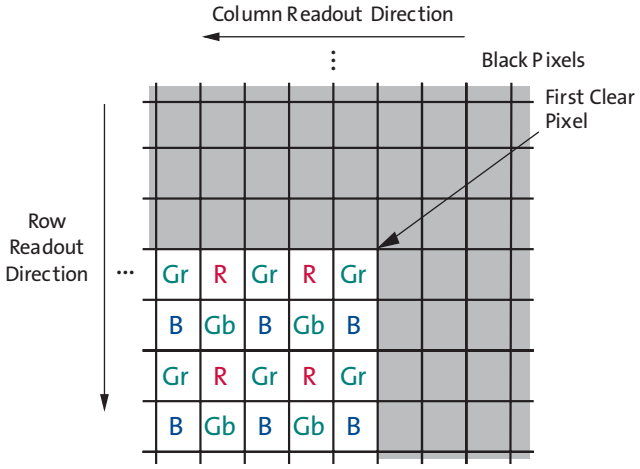
Figure 3: Sensor Core Block Diagram



Pixel Array

The sensor core uses a Bayer color pattern, as shown in Figure 4.

Figure 4: Pixel Color Pattern Detail (Top Right Corner)

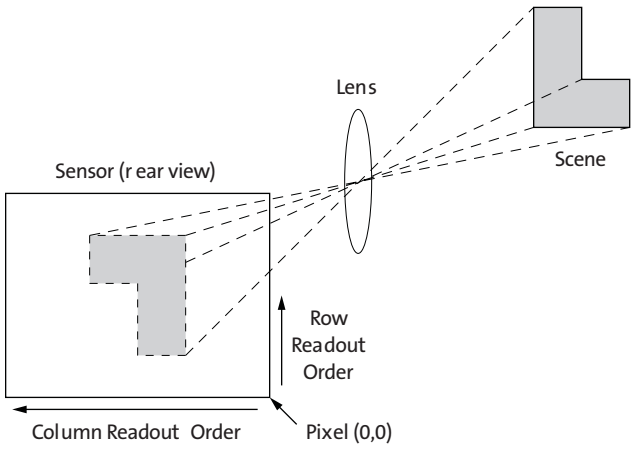


Default Readout Order

When the sensor is operating in a system, the active surface of the sensor faces the scene as shown in Figure 5.

When the image is read out of the sensor, it is read one row at a time, with the rows and columns sequenced. By convention, data from the sensor is shown with the first pixel read out in the case of the sensor core in the top left corner.

Figure 5: Imaging a Scene



PLL

PLL-Generated Clocks

The PLL can generate a pixel clock signal whose frequency is up to 96 MHz, using a EXTCLK input of 8 through 54 MHz. The PLL must be enabled for EXTCLK input below 8 MHz. The PLL also supports a dither function for reduced EMI.

PLL Setup

Because the input clock frequency is unknown, the sensor starts up with the PLL disabled. The PLL takes time to power up. The behavior of its output clock signal during lock phase is not guaranteed.

Digital Processing

Readout Options

The sensor core supports different readout options to modify the image before it is sent to the IFP. The readout can be limited to a specific window of the original pixel array.

For preview modes, the sensor core supports both skipping and binning in x and y directions.

By changing the readout direction the image can be flipped in the vertical and/or mirrored in the horizontal.

Window Size

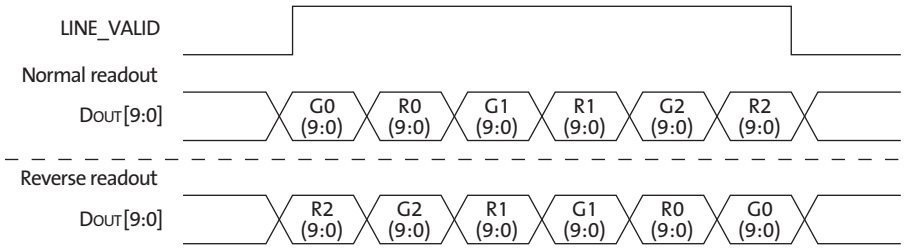
The image output size is set using firmware variables. The edge pixels in the array are present to avoid edge defects and should not be included in the visible window. Binning or skipping will change the image output size.

Readout Modes

Horizontal Mirror

When the sensor is configured to mirror the image horizontally, the order of pixel readout within a row is reversed. Figure 6 shows a sequence of 6 pixels being read out with normal readout and reverse readout.

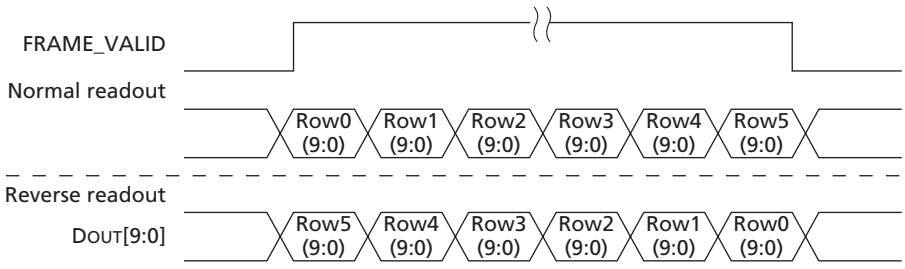
Figure 6: 6 Pixels in Normal and Column Mirror Readout Modes



Vertical Flip

When the sensor is configured to flip the image vertically, the order in which pixel rows are read out is reversed. Figure 7 shows a sequence of 6 rows being read out with normal readout and reverse readout.

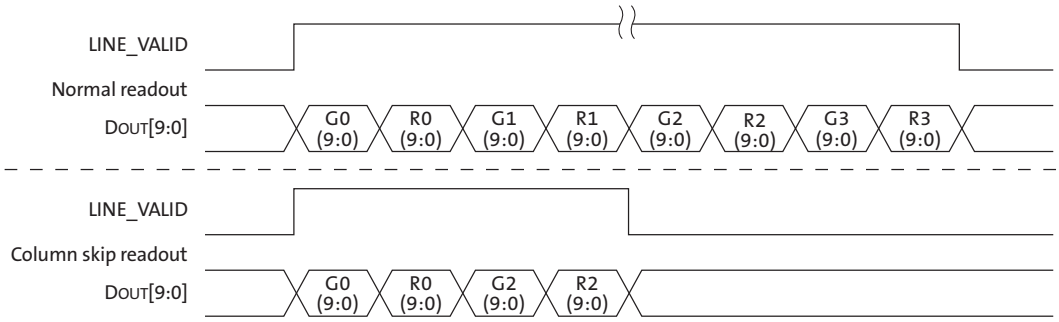
Figure 7: Six Rows in Normal and Row Mirror Readout Modes



Column and Row Skip

The sensor core supports subsampling. Subsampling reduces the amount of data processed by the analog signal chain in the sensor and thereby allows the frame rate to be increased. This reduces the amount of row and column data processed and is equivalent to the skip2 readout mode provided by earlier Aptina imaging sensors. When enabling subsampling, the proper image output and crop sizes must be updated beforehand.

Figure 8: 8 Pixels in Normal and Column Skip 2X Readout Modes



Pixel Readouts

The following diagrams show a sequence of data being read out with no skipping. The effect of the different subsampling on the pixel array readout is shown in Figures 9 through 13.

Figure 9: Pixel Readout (no skipping)

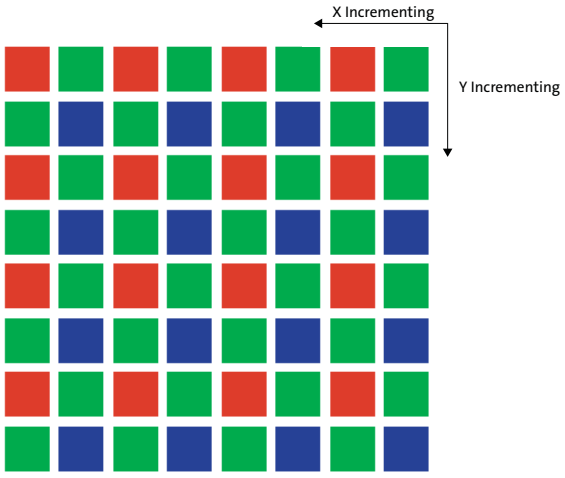


Figure 10: Pixel Readout (column skipping)

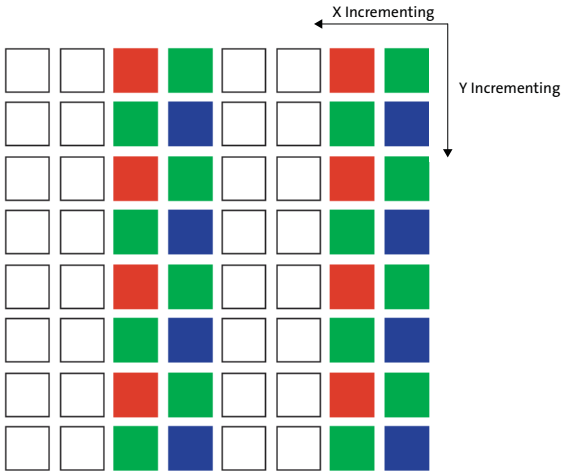


Figure 11: Pixel Readout (row skipping)

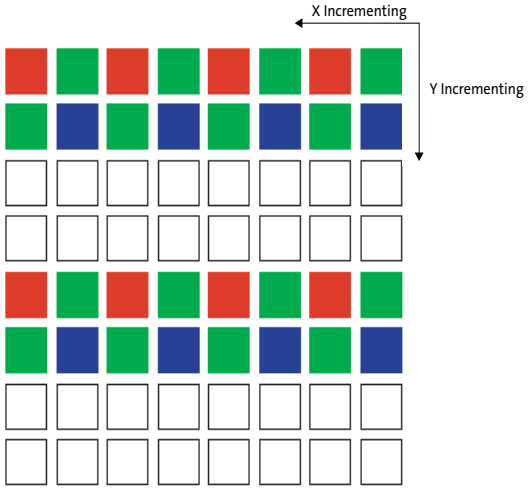


Figure 12: Pixel Readout (column and row skipping)

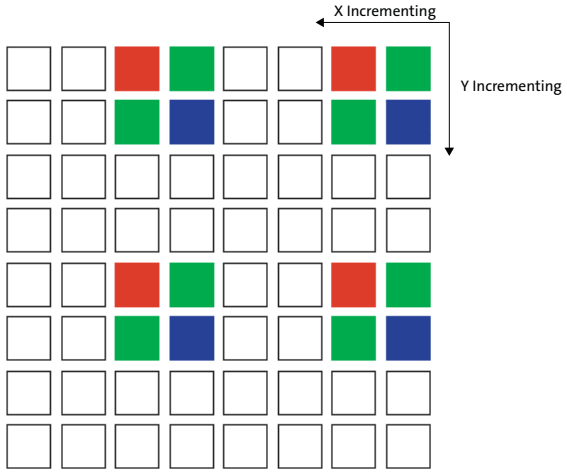


Table 4: Row Address Sequencing (Sampling)

Normal	Subsampled Sequence 1	Subsampled Sequence 2
0	0	No data
1	1	No data
2	No data	2
3	No data	3
4	4	No data
5	5	No data
6	No data	6
7	No data	7

Binning

The MT9P111 sensor core supports 2 x 1, 2 x 2, PMB3 and Bin2-Skip4 analog binning (column binning, also called x-binning and row/column binning, also called xy-binning). Binning has many of the same characteristics as subsampling but because it gathers image data from all pixels in the active window (rather than a subset of them), it achieves superior image quality and avoids the aliasing artifacts that can be a characteristic side effect of subsampling.

Binning is enabled by selecting the appropriate subsampling settings. Subsampling may require sensor window size adjustment when binning is enabled.

The effect of the different subsampling settings is shown in Figure 13 and Figure 14 on page 21.

Figure 13: Pixel Readout (column binning)

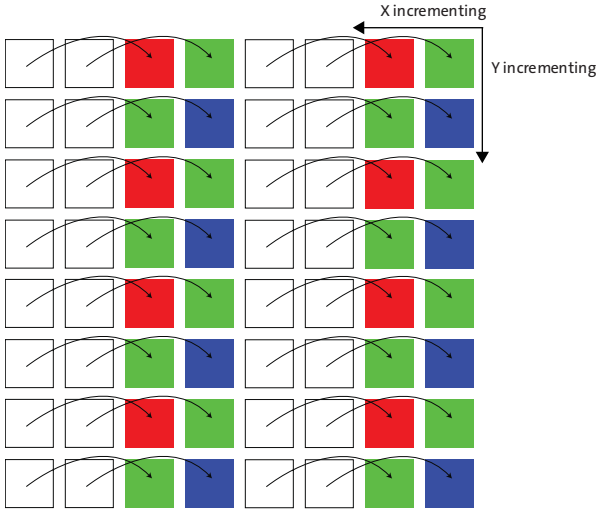
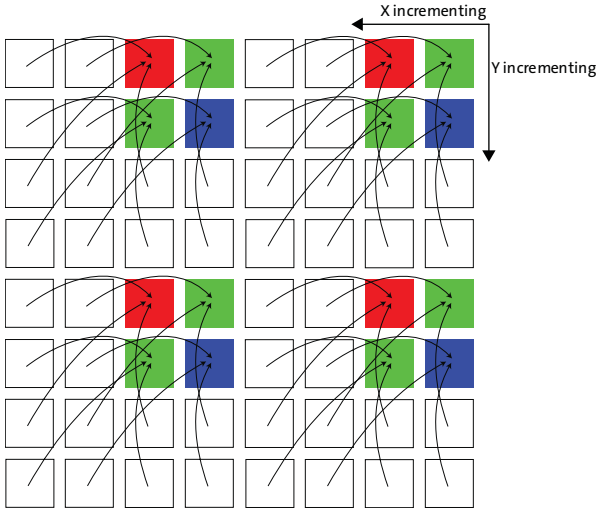


Figure 14: Pixel Readout (column and row binning)



Binning Limitations

The sensor must be taken out of streaming mode before switching between binned and non-binned operation. Binning requires different sequencing of the pixel array and imposes different timing limits on the operation of the sensor. In particular, xy-binning requires two read operations from the pixel array for each line of output data, which has the effect of increasing the minimum line blanking time.

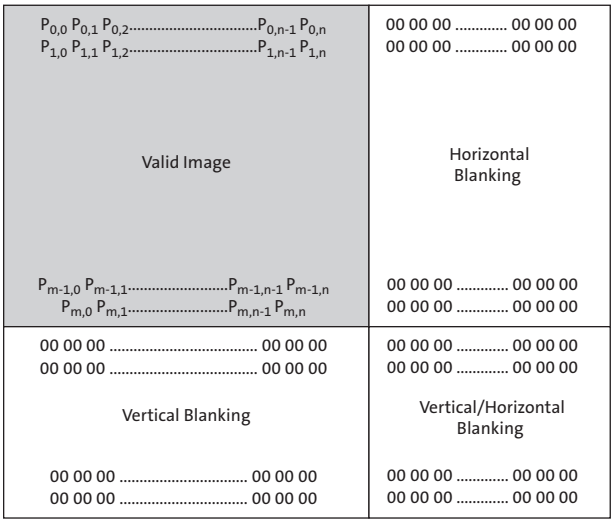
Table 5: Row Address Sequencing (Binning)

Normal	Binning Sequence 1	Binning Sequence 2
0	0,2	No data
1	1,3	No data
2	No data	2,4
3	No data	3,5
4	4,6	No data
5	5,7	No data
6	No data	6,8
7	No data	7,9

Raw Data Format

The sensor core image data is read out in a progressive scan. Valid image data is surrounded by horizontal blanking and vertical blanking, as shown in Figure 15 on page 22. The amount of horizontal blanking and vertical blanking is programmable. LV is HIGH during the shaded region of the figure.

Figure 15: Valid Image Data

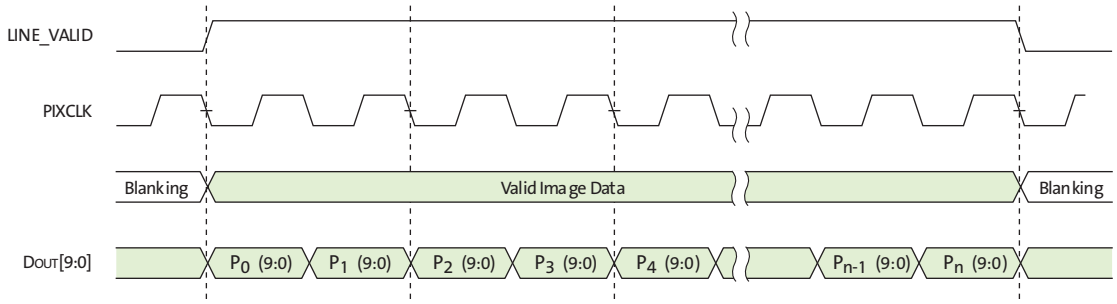


Raw Data Timing

DOUT[9:0]is synchronized with the PIXCLK output. When LV is HIGH, one pixel's data is output on the 10-bit DOUT output bus every PIXCLK period. By default, the PIXCLK signal runs at the same frequency as the master clock, and its rising edges occur one-half

of a master clock period after transitions on LV, FV, and DOUT (see Figure 16). This allows PIXCLK to be used as a clock to sample the data. PIXCLK is continuously enabled by default (but is configurable), even during the blanking period.

Figure 16: Pixel Data Timing Example

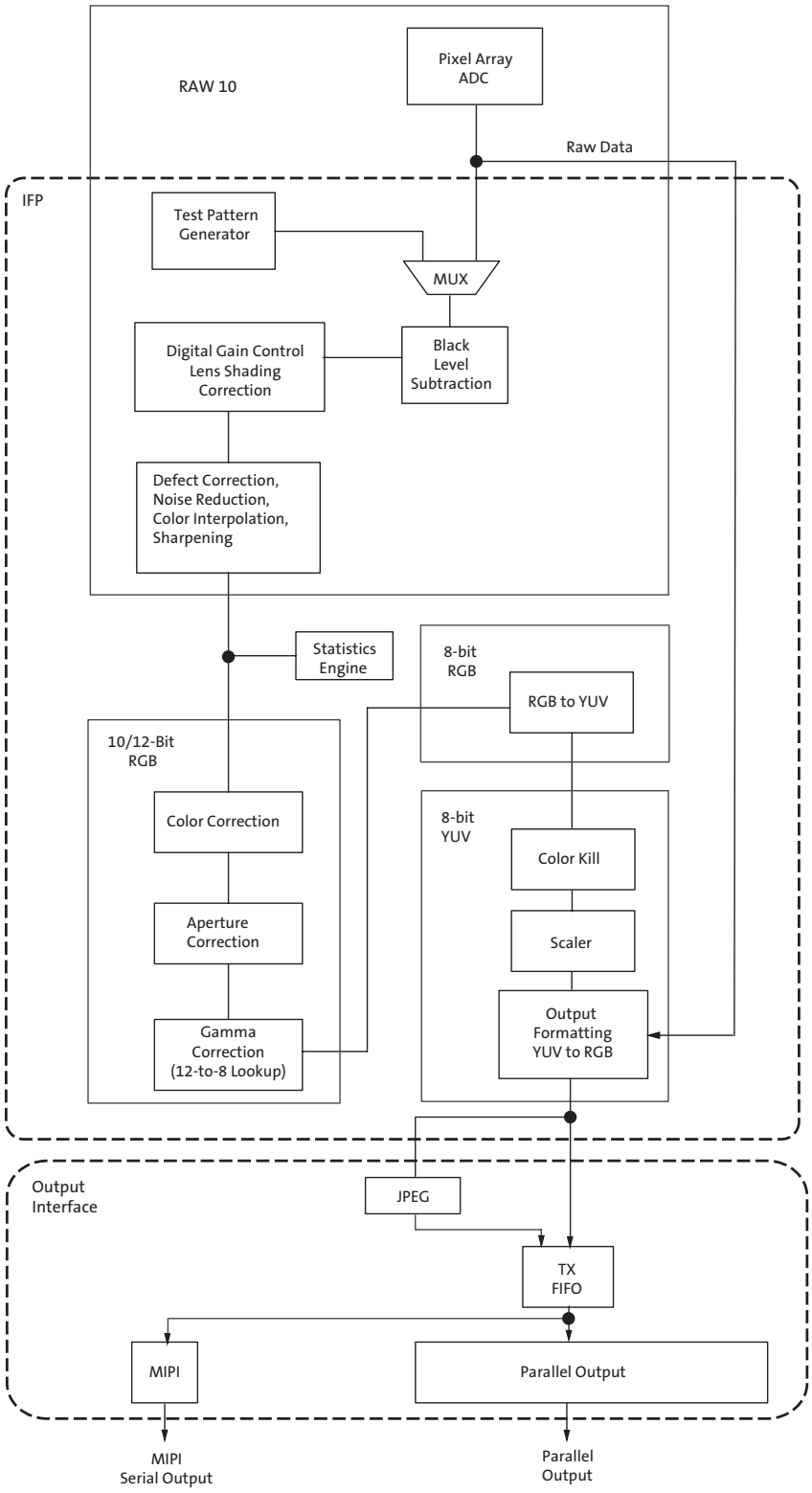


SOC Description

Image Flow Processor

Image and color processing in the MT9P111 are implemented as an image flow processor (IFP) coded in hardware logic. During normal operation, the embedded microcontroller will automatically adjust the operation parameters. The IFP is broken down into different sections, as outlined in Figure 17 on page 24.

Figure 17: Color Pipeline

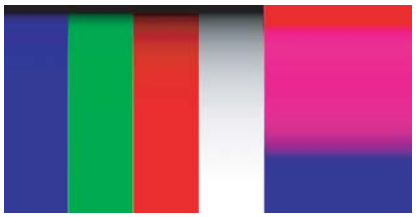


Test Patterns

During normal operation of the MT9P111, a stream of raw image data from the sensor core is continuously fed into the color pipeline. For test purposes, this stream can be replaced with a fixed image generated by a special test module in the pipeline. The module provides a selection of test patterns sufficient for basic testing of the pipeline.

Test patterns are accessible by programming a register and are shown in Figure 18. Disabling the MCU is recommended before enabling test patterns.

Figure 18: Color Bar Test Pattern

Test Pattern	Example
Flat Field	
Vertical Ramp	
Color Bar	
Vertical Stripes	
Pseudo-Random	

Black Level Subtraction and Digital Gain

Image stream processing starts with black level subtraction and multiplication of all pixel values by a programmable digital gain. Both operations can be independently set to separate values for each color channel (R, Gr, Gb, B). Independent color channel digital gain can be adjusted with registers. Independent color channel black level adjustments can also be made. If the black level subtraction produces a negative result for a particular pixel, the value of this pixel is set to "0."

Automatic Positional Gain Adjustments (APGA)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. There are also other factors causing fixed pattern signal gradients in images captured by image sensors. The cumulative result of all these factors is known as image shading. The MT9P111 has an embedded shading correction module that can be programmed to counter the shading effects on each individual R, Gb, Gr, and B color signal.

In some cases, different lighting conditions can introduce different color shading response. To compensate for the dependency of the lens shading to the illuminant that can result, different settings of lens shading correction (LC) coefficients can be used. The MT9P111 provides up to three settings to be stored. Each PGA setting should be optimized at a particular color temperature. In the MT9P111, color temperature is detected, stored in the firmware variable ccmPosition, and an appropriate PGA setting is applied.

The variable (ccmPosition) has a range from 0 through 255 and reflects the current color temperature, 0 corresponding to lowest color temperature, 255 the highest. The host specifies a range of ccmPosition values for a particular PGA setting. The ranges should overlap to provide hysteresis and prevent thrashing between PGA settings.

The Correction Function

For each illuminant, color-dependent solutions are calibrated using the sensor, lens system, and an image of an evenly illuminated, featureless gray calibration field. From the resulting image, the color correction functions can be derived.

The correction functions can then be applied to each pixel value to equalize the response across the image as follows:

$$P_{corrected}(row,col) = P_{sensor}(row,col) * f(row,col) \quad \text{(EQ 1)}$$

where P are the pixel values and f is the color dependent correction functions for each color channel.

One-Time Programmable Memory

The MT9P111 contains 10Kb of OTP memory, suitable for storing three separate lens shading correction settings, color calibration, external mechanisms, initialization settings, and module identification that can be programmed during the module manufacturing process. Programming the OTP memory requires the use of a high voltage at the VPP pin. During normal operation, the VPP pin should be left floating. The OTP memory can be accessed through the two-wire serial interface.

Defect Correction and Noise Reduction

The IFP performs continuous defect correction that can mask pixel array defects such as high dark-current (hot) pixels and pixels that are darker or brighter than their neighbors due to photoresponse nonuniformity. The module is edge-aware with exposure that is based on configurable thresholds. The thresholds are changed continuously based on the brightness of the current scene. Noise reduction can be enabled and disabled and thresholds can be set through register settings.

Color Interpolation

In the raw data stream fed by the sensor core to the IFP, each pixel is represented by a 10-bit integer number, which can be considered proportional to the pixel's response to a one-color light stimulus, red, green, or blue, depending on the pixel's position under the color filter array. Initial data processing steps, up to and including the defect correction, preserve the one-color-per-pixel nature of the data stream, but after the defect correction it must be converted to a three-colors-per-pixel stream appropriate for standard color processing. The conversion is done by an edge-sensitive color interpolation module. The module pads the incomplete color information available for each pixel with information extracted from an appropriate set of neighboring pixels. The algorithm used to select this set and extract the information seeks the best compromise between preserving edges and filtering out high frequency noise in flat field areas. The edge threshold can be set through register settings.

Color Correction and Aperture Correction

To achieve good color fidelity of the IFP output, interpolated RGB values of all pixels are subjected to color correction. The IFP multiplies each vector of three pixel colors by a 3 x 3 color correction matrix. The three components of the resulting color vector are all sums of three 10-bit numbers. Since such sums can have up to 12 significant bits, the bit width of the image data stream is widened to 12-bits per color (36 bits per pixel). The color correction matrix can be either programmed by the user or automatically selected by the auto white balance (AWB) algorithm implemented in the IFP. Color correction should ideally produce output colors that are corrected for the spectral sensitivity and color crosstalk characteristics of the image sensor. The optimal values of the color correction matrix elements depend on those sensor characteristics and on the spectrum of light incident on the sensor. The color correction variables can be adjusted through register settings.

To increase image sharpness, a programmable 2D aperture correction (sharpening filter) is applied to color-corrected image data. The gain and threshold for 2D correction can be defined through register settings.

Image Cropping

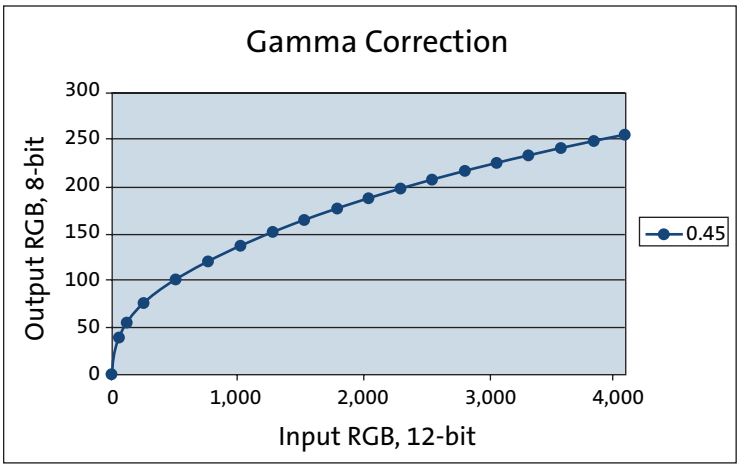
By configuring the cropped and output windows to various sizes, different zooming levels for example 4x, 2x, and 1x can be achieved. The location of the cropped window is also configurable so that panning is also supported. A separate cropped window is defined for context A and context B. In both contexts, the height and width definitions for the output window must be equal to or smaller than the cropped image.

Gamma Correction

The gamma correction curve (as shown in Figure 19 on page 28) is implemented as a piecewise linear function with 19 knee points, taking 12-bit arguments and mapping them to 8-bit output. The abscissas of the knee points are fixed at 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. The 8-bit ordinates are programmable through IFP registers.

The MT9P111 IFP includes a block for gamma correction that can adjust its shape based on brightness to enhance the performance under certain lighting conditions (see Figure 19 on page 28). Three custom gamma correction tables may be uploaded corresponding to a brighter lighting condition, a normal lighting condition, and a darker lighting condition. At power-up, the IFP loads the three tables with default values. The final gamma correction table used depends on the brightness of the scene and can take the form of either uploaded tables or an interpolated version of two of the three tables. A single (non-adjusting) table for all conditions can also be used.

Figure 19: Gamma Correction Curve



Special Effects

Special effects like negative image, sepia, or B/W can be applied to the data stream at this point. These effects can be enabled and selected by registers.

RGB to YUV Conversion

For further processing, the data is converted from RGB color space to YUV color space.

Color Kill

To remove high or low light color artifacts, a color kill circuit is included. It affects only pixels whose luminance exceeds a certain preprogrammed threshold. The U and V values of those pixels are attenuated proportionally to the difference between their luminance and the threshold.

YUV Color Filter

As an optional processing step, noise suppression by one-dimensional low-pass filtering of Y and/or UV signals is possible. A 3- or 5-tap filter can be selected for each signal.

Image Scaling

To ensure that the size of images output by the MT9P111 can be tailored to the needs of all users, the IFP includes a scaler module. When enabled, this module performs rescaling of incoming images—shrinks them to arbitrarily selected width and height without reducing the field of view and without discarding any pixel values.

The scaler performs pixel binning—divides each input image into rectangular bins corresponding to individual pixels of the desired output image, averages pixel values in these bins, and assembles the output image from the bin averages. Pixels lying on bin boundaries contribute to more than one bin average; their values are added to bin-wide sums of pixel values with fractional weights. The entire procedure preserves all image information that can be included in the downsized output image and filters out high frequency features that could cause aliasing.

The image cropping and scaler module can be used together to implement a digital zoom and pan. If the scaler is programmed to output images smaller than images coming from the sensor core, zoom effect can be produced by cropping the latter from their maximum size down to the size of the output images. The ratio of these two sizes determines the maximum attainable zoom factor. For example, a 2560 x 1920 image rendered on a 256 x 192 display can be zoomed up to ten times, since 2560/256 = 1920/192 = 10. Panning effect can be achieved by fixing the size of the cropping window and moving it around the pixel array.

If downscaling by 3:1 or more, 2D aperture correction may be applied to increase image sharpness lost due to pixel binning during image scaling.

YUV-to-RGB/YUV Conversion and Output Formatting

The YUV data stream emerging from the scaling module can either exit the color pipeline as-is or be converted before exit to an alternative YUV or RGB data format.

Output Interface (Parallel and MIPI Output)

The user can select to either use the serial MIPI output or the 8-bit parallel output to transmit the data. Only one of the output modes can be used at any time.

The parallel output is used with an output FIFO whose memory is shared with the MIPI output FIFO to retain a constant pixel output clock independent from the scaling factor.

The MIPI output transmitter implements a serial differential sub-LVDS transmitter capable of up to 768 Mb/s. It supports multiple formats, error checking, and custom short packets.

Table 6: Data Formats Supported by MIPI Interface

Data Format	Data Type
YUV 422 8-bit	0x1E
565RGB	0x22
555RGB	0x21
444RGB	0x20
RAW8	0x2A
RAW10	0x2B
User-defined byte-based data (including compressed data)	0x30 0x31 0x32 0x33

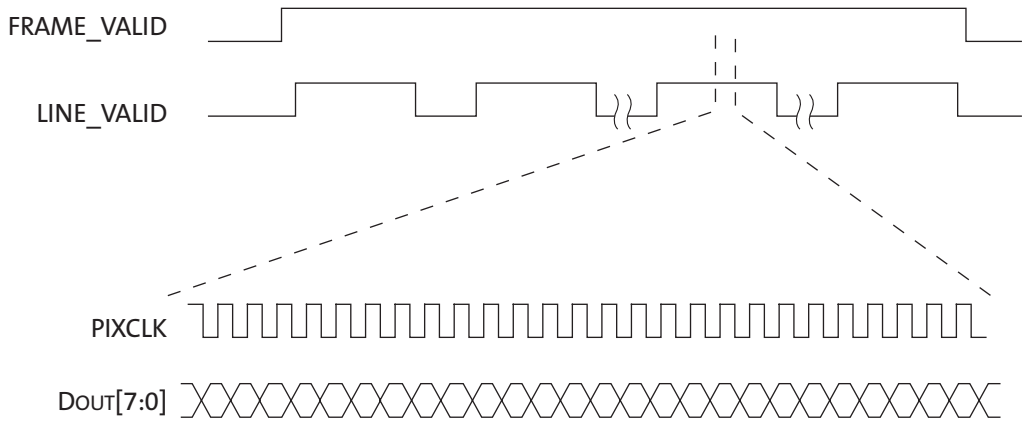
Notes: 1. Data will be packed as RAW8 if the data type specified does not match any of the above data types.

Output Format and Timing

YUV/RGB Output

Figure 20 depicts the output timing of YUV/RGB when a scaled data stream is equalized by buffering or when no scaling takes place. The pixel clock frequency remains constant during each LV high period.

Figure 20: Timing of Full Frame Data or Scaled Data Passing Through the FIFO



YUV/RGB Data Ordering

The MT9P111 supports swapping YCbCr mode, as illustrated in Table 7.

Table 7: YCbCr Output Data Ordering

Mode	Data Sequence			
Default (no swap)	Cb_i	Y_i	Cr_i	Y_{i+1}
Swapped CrCb	Cr_i	Y_i	Cb_i	Y_{i+1}
Swapped YC	Y_i	Cb_i	Y_{i+1}	Cr_i
Swapped CrCb, YC	Y_i	Cr_i	Y_{i+1}	Cb_i

The RGB output data ordering in default mode is shown in Table 8. The odd and even bytes are swapped when luma/chroma swap is enabled. R and B channels are bit-wise swapped when chroma swap is enabled.

Table 8: RGB Ordering in Default Mode

Mode (Swap Disabled)	Byte	$D_7D_6D_5D_4D_3D_2D_1D_0$
565RGB	Odd	$R_7R_6R_5R_4R_3G_7G_6G_5$
	Even	$G_4G_3G_2B_7B_6B_5B_4B_3$
555RGB	Odd	$0\ R_7R_6R_5R_4R_3G_7G_6$
	Even	$G_5G_4G_3B_7B_6B_5B_4B_3$
444xRGB	Odd	$R_7R_6R_5R_4G_7G_6G_5G_4$
	Even	$B_7B_6B_5B_4\ 0\ 0\ 0\ 0$
x444RGB	Odd	$0\ 0\ 0\ 0\ R_7R_6R_5R_4$
	Even	$G_7G_6G_5G_4B_7B_6B_5B_4$

Uncompressed 10-Bit Bypass Output

Raw 10-bit Bayer data from the sensor core can be output in bypass mode in two ways:

- Using 8 data output signals (DOUT[7:0]) and VGPIO[1:0]. The VGPIO signals are the least significant 2 bits of data.
- Using only 8 signals (DOUT[7:0]) and a special 8 + 2 data format, shown in Table 9.

Table 9: 2-Byte RGB Format

Byte	Bits Used	Bit Sequence
Odd bytes	8 data bits	$D_9D_8D_7D_6D_5D_4D_3D_2$
Even bytes	2 data bits + 6 unused bits	$0\ 0\ 0\ 0\ 0\ 0\ D_1D_0$

JPEG Encoder

The JPEG compression engine in the MT9P111 is a highly integrated, high-performance solution that provides for low power consumption and full programmability of JPEG compression parameters for image quality control.

The JPEG encoding block is designed for continuous image flow and is ideal for low power applications. After initial configuration for a target application, it can be controlled easily for instantaneous stop or restart. A flexible configuration and control interface allows for full programmability of various JPEG-specific parameters and tables.

JPEG Encoding Highlights

- Sequential DCT (baseline) ISO/IEC 10918-1 JPEG-compliant
- YCbCr 4:2:2 and 4:2:0 format compression
- Support for two pairs of programmable quantization tables
- Quality/compression ratio control capability
- 15 fps JPEG capability at full resolution with or without JFIF- or EXIF-compliant header
- Support for interleaved RGB or YUV thumbnail up to 640 x 480
- Capture color pipe bypass stream (8- or 10-bit), JPEG bypass stream (16-bit), or JPEG encoded stream (8-bit), as programmed by host or microcontroller
- JPEG encoded stream can work in continuous mode or spoof mode

- JPEG encoded stream working in continuous mode can only transmit on the parallel output port
- Thumbnail can be enabled for the JPEG encoded stream in both continuous and spoof mode
- In spoof mode, data is output with programmed spoof frame sizes; dummy pixels may be padded as necessary
- Support for Scalado RAJPEG
- MIPI data types
- Spoof-frame height can be ignored in spoof mode
- Optional JFIF or EXIF header generation

JPEG Output Interface

JPEG Data

JPEG data can be output in both the parallel and the serial MIPI streams. In the parallel output interface, JPEG data is output on the 8-bit parallel bus DOUT[7:0], with FV, LV, and PIXCLK. JPEG output data is valid when both FV and LV are asserted. When the JPEG data output for the frame completes, or buffer overflow occurs, LV and FV are de-asserted.

The MT9P111 can transmit JPEG data using two different formats: JPEG continuous stream and JPEG spoof stream. In both formats, JPEG status segments containing information (resolution, file size, and status) about the image and the offsets of thumbnail data can be inserted into the output streams. The following sections describe the two streaming methods.

RGB or YCbCr Thumbnail

To support display of captured images without decoding a JPEG file, the MT9P111 can output a resized version of the captured JPEG data as an RGB or YCbCr thumbnail image embedded in the JPEG stream.

This thumbnail image is computed from the same image that is input to the JPEG compressor, and is scaled to a user-programmable size, from 160 x 120 to 640 x 480. The thumbnail size must be configured to be at least two times smaller than the JPEG image size.

This image can be separated by parsing the stream for tags surrounding the embedded image. Alternatively, the embedded image can be extracted without parsing by reading thumbnail data offsets from the thumbnail pointer table. This thumbnail pointer table is optionally output in the image status segment, and contains one entry for each line of thumbnail data.

JPEG Continuous Stream

JPEG continuous stream goes out only through the parallel output interface, and supports the following features:

- Adaptive clock switching
- Duplicate FV on LV
- Append JPEG status segment at the end of the data stream

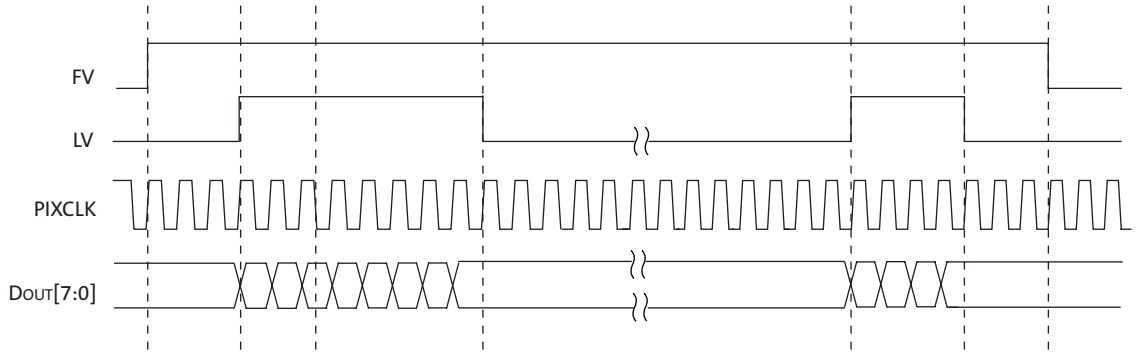
When enabled, the pixel clock output can be generated continuously during invalid data periods (between FV and between LV). In this streaming mode, the amount of valid data within each line (LV = 1) is variable. When adaptive clock mode is enabled, the pixel

clock is adjusted to lower clock rates, based on the fullness of the output FIFO. Figure 21 through Figure 24 on page 35 are examples of the JPEG stream through the parallel output interface.

Figure 21 illustrates data output when the pixel clock output is generated continuously during invalid data periods. LV is of variable length based on data output rate.

In default mode, data transitions on the falling of PIXCLK and the host must capture data on the rising edge of PIXCLK. The PIXCLK is also configurable and its polarity can be reversed through the use of register settings.

Figure 21: JPEG Continuous Data Output



Notes: 1. Under default conditions FV and LV are asserted on the falling edge of PIXCLK.
8. Data must be captured by the host on the rising edge of PIXCLK.

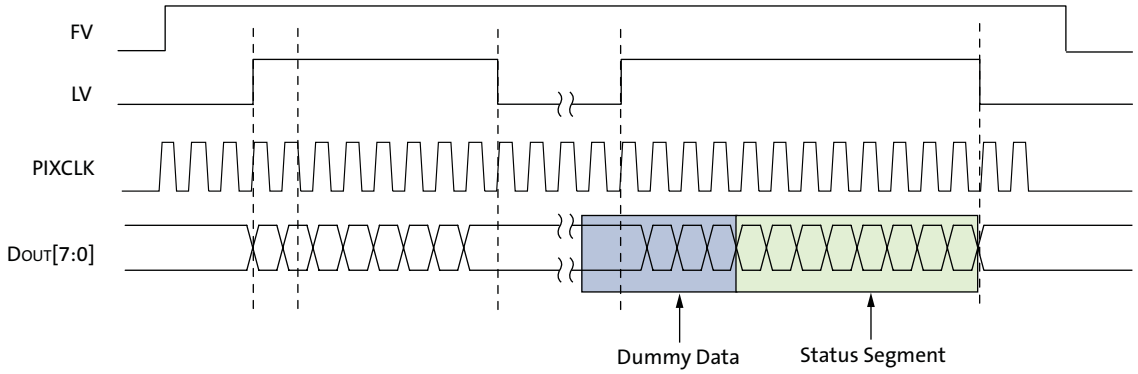
JPEG Spoof Stream

The JPEG compressed data can be output in spoof mode. The amount of expected pixel data is defined by the width and height registers in spoof mode. If the valid JPEG data is less than expected size defined, a register-programmable dummy data pattern with a default value of 0xFF will be padded.

When enabled, the pixel clock output can be generated continuously during invalid data periods (between FV and between LV). In this streaming mode, the amount of valid data within each line (LV = 1) is constant. When adaptive clock mode is enabled, the pixel clock is readjusted to lower clock rates, based on the fullness of the output FIFO. Below are some examples of the JPEG spoof stream.

Figure 22 illustrates the JPEG spoof output when pixel clock is generated continuously during invalid data periods between LV. The status segment is inserted at the end of the stream.

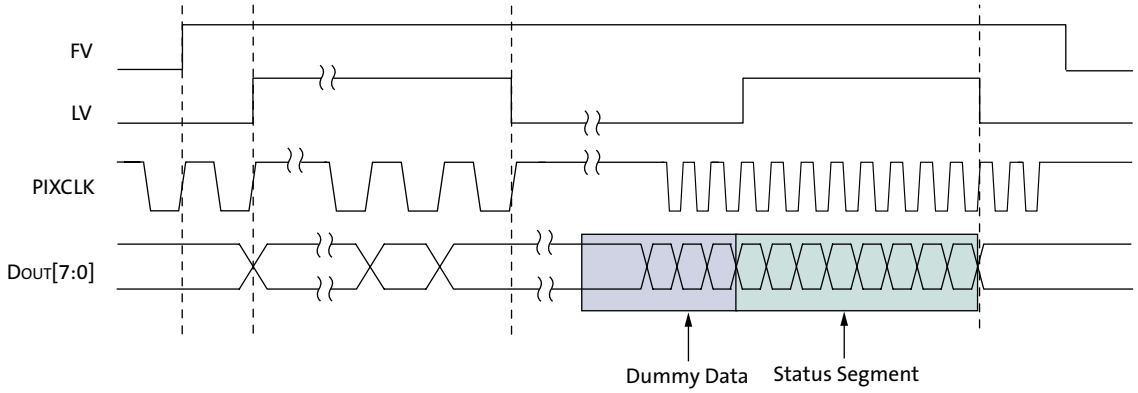
Figure 22: JPEG Spoof Mode Timing with Continuous Clock



Notes: 1. PIXCLK is reversed in this example with data output on the rising edge of PIXCLK and data captured by the host on the falling edge of PIXCLK.

Figure 23 illustrates the JPEG spoof output when the adaptive clock mode is enabled. With continuous PIXCLK, the switching of the PIXCLK frequency can happen at any time.

Figure 23: JPEG Spoof Mode Timing with Adaptive Clock



Notes: 1. PIXCLK is reversed in this example with data output on the rising edge of PIXCLK and data captured by the host on the falling edge of PIXCLK.

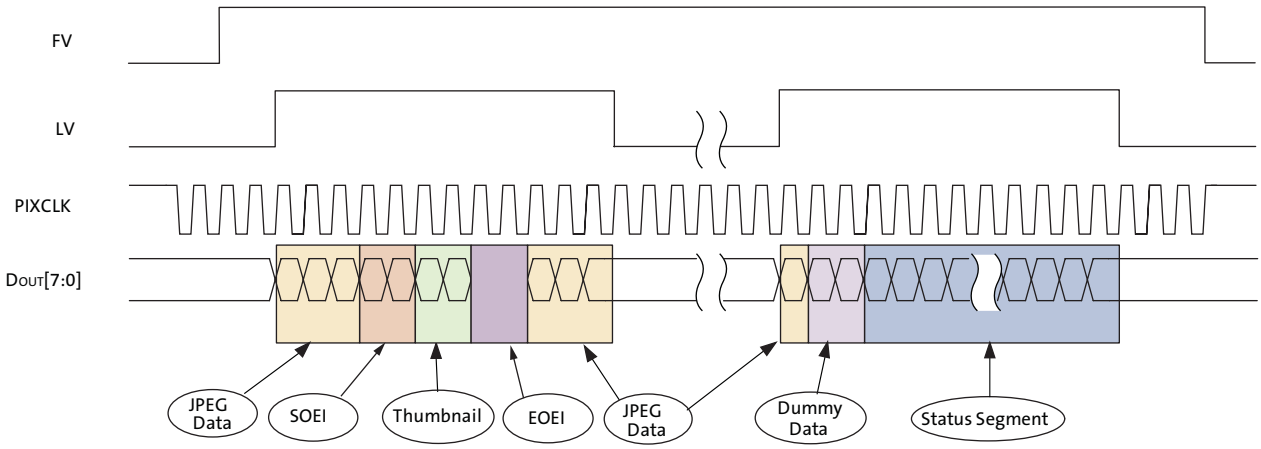
JPEG Spoof Stream in MIPI Output Mode

In MIPI output mode, only the JPEG spoof stream can be output. Similar to the parallel output interface, the amount of expected pixel data is defined by the width and height registers in spoof mode. If the valid JPEG data is less than expected size defined, register-specified dummy data will be padded.

JPEG Stream with Embedded Thumbnail Image

In JPEG mode, it is possible to embed a scaled uncompressed image to the compressed data stream. This image is interleaved within the data (as shown in Figure 24), and must be separated before saving the compressed image. The embedded image is separated from the main image by optional Start of Embedded Image (SOEI) and End of Embedded Image (EOEI) tags. These tags are register-programmable codes that enable a host to parse the thumbnail data from the compressed image stream.

Figure 24: JPEG Spoof Mode Timing with Thumbnail



Notes:
1. PIXCLK is inverted in this example.
9. Thumbnail start and end codes are programmable by register setting.
10. Status segment includes JPEG pointer table.

In addition, the output formatter can append a table of thumbnail data offsets to the status segment of the image. This thumbnail index pointer shall have one entry for each line of thumbnail data. Each entry is a 4-byte pointer containing the offset of the valid thumbnail data.

JPEG Status Segment

To provide the user quick knowledge of the status when the JPEG plus thumbnail is enabled, a JPEG status segment is appended at the end of frame. This segment is optional in continuous mode, while it is mandatory for spoof mode.The status segment is enclosed by SOSI/EOSI codes, as shown in Figure 25.

Figure 25: JPEG Status Segment Structure

SOSI on next line (optional)	Thumbnail Index Table Height * 4 bytes (optional)	Thumbnail Size (optional)	Original JPEG Size 4 bytes (optional)	Frame Length (4 bytes)	TXF Status (2 bytes)	TN 2 bytes (optional)	EOSI (0xFFBD)

The contents of the status segment are summarized as follows:

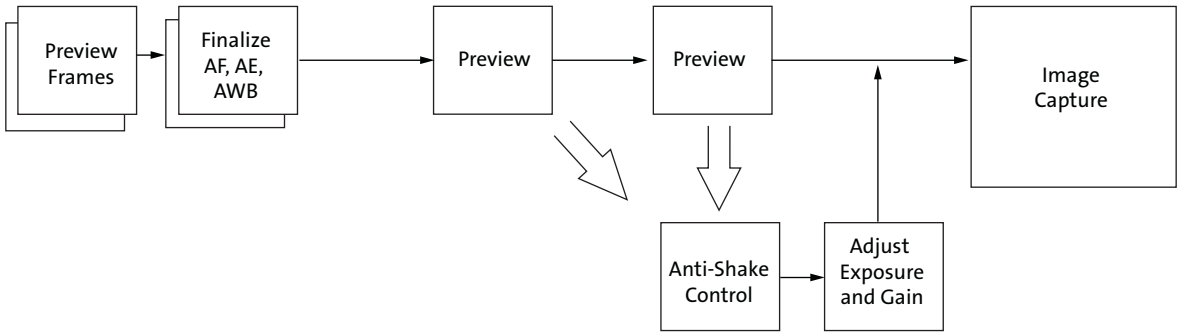
- SOSI, start of status information, which is coded as 0xFFBC
- Thumbnail index table (every entry has 4 bytes) is asserted and thumbnail is enabled
- The width of thumbnail in pixels (2 bytes)
- The height of thumbnail (2 bytes)
- The width of uncompressed full image
- The height of uncompressed full image
- 4-byte JPEG plus thumbnail length
- 2-byte status
- EOSI, end of status information, which is coded as 0xFFBD
- Options to use to match legacy parts

Either thumbnail data or JPEG data starts first, depending on the time of their availability.

Anti-Shake (AS)

As mobile devices become smaller, unavoidable handshaking make it difficult for a user to hold a slim mobile camera steady enough to get a flawless shot, especially when exposure time increases due to low light conditions. To reduce motion blur, the MT9P111 includes an anti-shake mode. When motion is detected, it will increase the sensor sensitivity and reduce the exposure time correspondingly. The anti-shake mode will reduce the motion blur caused by camera handshaking. Figure 26 shows the block diagram for the anti-shake algorithm.

Figure 26: Anti-Shake Algorithm



Camera Control

General Purpose I/Os

The eight general purpose I/Os of the MT9P111 can be configured in multiple ways. Each of the I/Os can be used for multiple purposes and can be programmed from the host. The VGPIOs are powered by their own power supply domain. The VGPIO configurations are shown in Table 10.

If the auto-focus mechanisms are controlled by the serial master, all eight VGPIOs will be available for advanced flash and mechanical shutter operations.

Table 10: VGPIO Configurations

VGPIO[7:0]	Standard Configuration w/ VGPIO as Inputs	Standard Configuration w/ VGPIO as Outputs	Optional Configuration w/ VGPIO as Inputs and Sensor Core Not Needed	Optional Configuration w/ VGPIO as Outputs and Sensor Core Not Needed	Default
VGPIO[0]	SHUTTER_SEL	Dout_LSB[0]	GPI	GPO_PWM	GPI
VGPIO[1]	FLASH_SEL	Dout_LSB[1]	GPI	GPO_PWM	GPI
VGPIO[2]	OE_BAR	SHUTTER	GPI	GPO_PWM	GPI
VGPIO[3]	GPI	FLASH	GPI	GPO_PWM	GPI
VGPIO[4]	GPI	GPO_PWM	GPI	GPO_PWM	GPI
VGPIO[5]	GPI	GPO_PWM	GPI	GPO_PWM	GPI
VGPIO[6]	GPI	GPO_PWM	GPI	GPO_PWM	GPI
VGPIO[7]	GPI	GPO_PWM	GPI	GPO_PWM	GPI

The general purpose inputs are enabled or disabled through register settings. The state of the general purpose inputs can be read from a register.

Output Enable Control

When the parallel pixel data interface is enabled, its signals can be switched asynchronously between the driven and High-Z under pin or register control.

Trigger Control

When the global reset feature is in use, the trigger for the sequence can be initiated either under pin or register control.

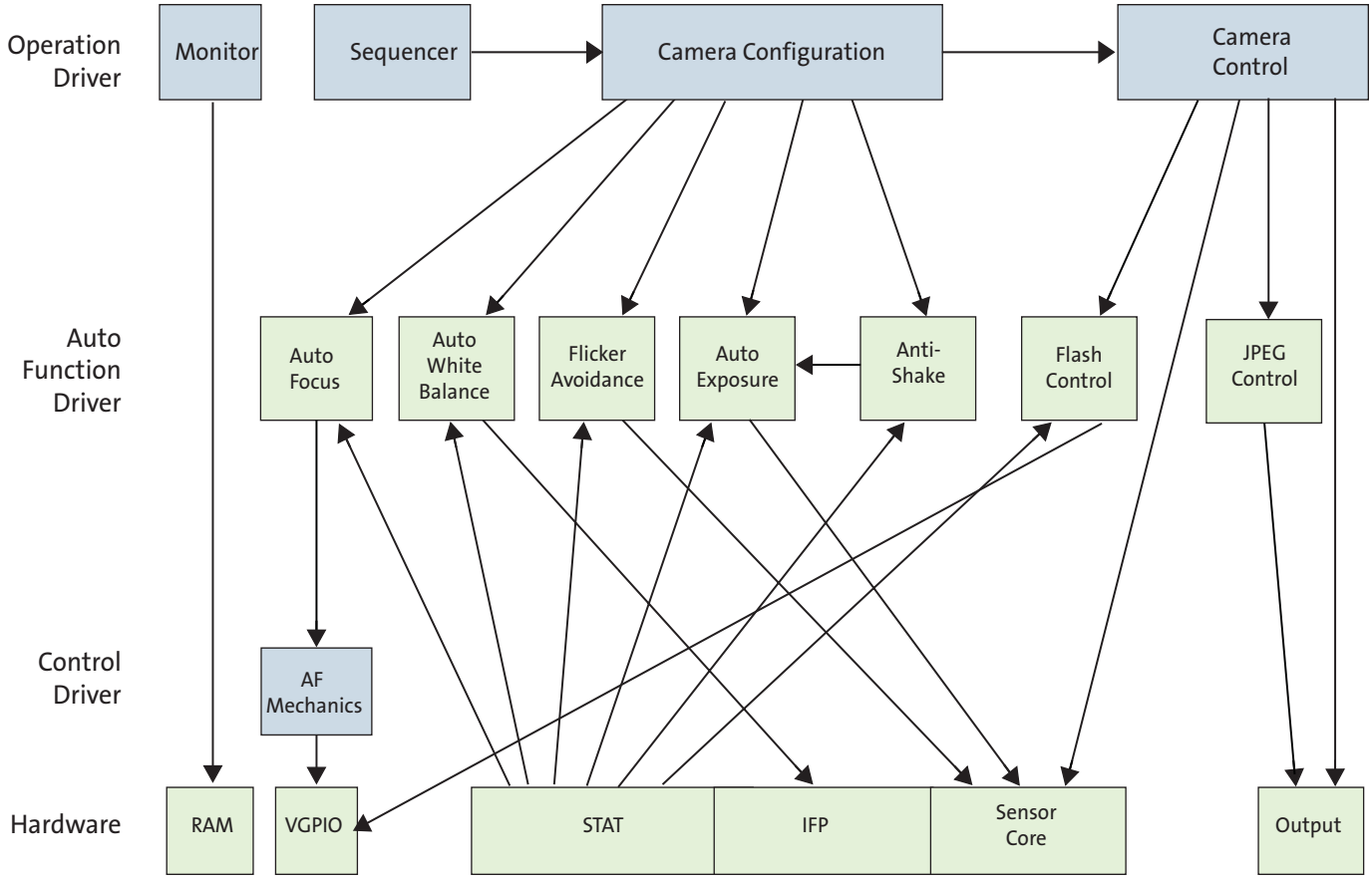
Table 11: Trigger Control

GPI Configured TRIGGER Pin	Global Trigger	Description
Disabled	0	Idle
Disabled	1	Trigger
0	0	Idle
X	1	Trigger
1	X	Trigger

Firmware Architecture

The firmware for the MT9P111 is implemented in multiple drivers that are responsible for different parts of operation.

Figure 27: Firmware Architecture Block Diagram



Sequencer

The sequencer is responsible for coordinating all events triggered by the user. It is implemented as a state machine. For example, sending a capture command to the sequencer will change the resolution from preview to full resolution, turn on or off an external LED, and switch back to preview after capturing the frame. The setup of the sensor can be defined by the user for preview and capture.

Context and Operational Modes

The MT9P111 can operate in several modes including preview, still capture (snapshot), and full resolution video. All modes of operation are individually configurable and are organized as two contexts—context A and context B. Context switching can be accomplished by sending a command through the two-wire serial interface.

Preview Mode

Context A is primarily intended for use in the preview mode. During preview, the sensor usually outputs low resolution images at a relatively high frame rate, and its power consumption is kept to a minimum. All automatic functions are enabled in this mode to adjust to the best image possible.

Still Capture and Video Modes

Context B can be configured for the full resolution still capture or video mode, as required by the user. For still capture configuration, the user typically specifies the desired output image size, if flash should be enabled, how many frames to capture, and so forth. For video, the user might select a different image size and a fixed frame rate.

Snapshot and Flash

To take a snapshot, the user must send a command that changes the context from A to B. A typical sequence of events after this command is:

1. The camera may turn on its LED flash, if it has one and is required to use it. With the flash on, the camera exposure and white balance are automatically adjusted to the changed illumination of the scene.
2. The camera captures one or more frames of desired size. A camera equipped with a xenon flash strobes while capturing images. When capturing images is completed, the camera automatically returns to context A and resumes running in preview mode.

Note: This sequence of events can take up to 10 frames.

Video

To start video capture, the user must change relevant context B settings, such as capture mode, image size and frame rate, and again send a context change command. Upon receiving it, the MT9P111 switches to the modified context B settings, while continuing to output YUV-encoded image data. AE adjusts automatically and provides a smooth continuous operation. To exit the video capture mode, the user must send another context change command, causing the sensor to switch back to context A.

Multi-Shot Image Capture Mode

This mode allows for a series of images with short intervals to be captured by continually storing full-scale images in a ring buffer (in the customer system) to allow the user to select the optimal image that occurred before, or after, the capture moment.

- 7.5 fps minimum capture frame rate
- 15 fps viewfinder frame rate
- selectable number of captured frames
- selectable delay intervals between shots can be set (0,1,2, 3 @ 15fps viewfinder)

Auto Exposure

The auto exposure algorithm performs automatic adjustments of the image brightness by controlling exposure time and analog gains of the sensor core as well as digital gains applied to the image.

Auto exposure is implemented by a firmware driver that analyzes image statistics collected by the exposure measurement engine, makes a decision, and programs the sensor core and color pipeline to achieve the desired exposure. The measurement engine subdivides the image into a 5 x 5 grid.

The AE uses a constant average tracking algorithm where a target brightness value is compared to a current brightness value, and the gain and integration time are adjusted accordingly to meet the target requirement.

AE Driver

The auto exposure mode is activated during preview. This mode can also be enabled during video capture mode. It relies on the statistics engine that tracks speed and amplitude of the change of the overall luminance in the selected windows of the image.

Backlight compensation is achieved by weighting the luminance in the center of the image higher than the luminance on the periphery. Other algorithm features include the rejection of fast fluctuations in illumination (time averaging), control of speed of response, and control of the sensitivity to the small changes. While the default settings are adequate in most situations, the user can program target brightness, measurement window, and other parameters described above.

The driver changes AE parameters (integration time, gains, and so on) to drive brightness to the programmable target. The value of the single step approach to the target value can be controlled.

To avoid unwanted reaction of AE on small fluctuations of scene brightness or momentary scene changes, the AE driver uses a temporal filter for luma and a threshold around the AE luma target. The driver changes AE parameters only if the buffered luma is larger than the AE target step and pushes the luma beyond the threshold.

Accelerated Settling During Overexposure

The AE speed is direction-dependent. Transitioning from oversaturation to target can take more time than transitioning from undersaturation. The AE driver has a mode that speeds up AE for overexposed scenes.

The AE driver counts the number of AE windows whose average brightness is equal to or greater than some value, 250 by default. For a scene having saturated regions, the average luma is underestimated due to signal clipping. The driver compensates underestimation by a factor that can be defined.

Exposure Control

To achieve the required amount of exposure, the AE driver adjusts the sensor integration time, gains, ADC reference, and IFP digital gains. In addition, a variable is available for the user to adjust the overall brightness of the scene. To reject flicker, integration time is typically adjusted in increments of steps. The incremental step specifies the duration in row times equal to one flicker period. Thus, flicker is rejected if integration time is kept a natural factor of the flicker period.

Auto White Balance

The MT9P111 has a built-in auto white balance algorithm designed to compensate for the effects of changing spectra of the scene illumination on the quality of the color rendition. The algorithm consists of two major parts: a measurement engine performing statistical analysis of the image and a driver performing the selection of the optimal color correction matrix, digital, and sensor core analog gains. While default settings of these algorithms are adequate in most situations, the user can reprogram base color correction matrices, place limits on color channel gains, and control the speed of both matrix and gain adjustments.

Flicker Detection

Flicker occurs when the integration time is not an integer multiple of the period of the light intensity. The automatic flicker detection block does not compensate for the flicker, but rather avoids it by detecting the flicker frequency and adjusting the integration time. For integration times below the light intensity period (10ms for 50Hz environment), flicker cannot be avoided. Flicker shows as horizontal bars rolling up or down.

Auto Focus

Overview

The auto focus (AF) algorithm implemented in the MT9P111 firmware seeks to maximize sharpness of vertical lines in images output by the sensor by guiding an external lens actuator to the position of best lens focus. The algorithm is actuator-independent; it provides guidance by means of an abstract one-dimensional position variable, leaving the translation of its changes into physical lens movements to a separate AF mechanics (AFM) driver. The AF algorithm relies on the AFM driver to generate digital output signals needed to move different lens actuators and to correctly indicate at all times if the lens is stationary or moving. The latter is required to prevent the AF algorithm from using line sharpness measurements distorted by concurrent lens motion.

For measuring line sharpness, the AF algorithm relies on the focus measurement engine in the color pipeline, which is a programmable vertical-edge-filtering module. In every interpolated image, statistics are collected in 16 equal-sized rectangular sub-blocks, referred to as AF windows or zones.

There are several motion sequences through which the MT9P111 AF algorithm can bring a lens to best focus position. All these sequences begin with a jump to a preselected start position, for example, the infinity focus position. This jump is referred to as the first flyback. It is followed by a unidirectional series of steps that puts the lens at up to 19 preselected positions different from the start position. This series of steps is called the first scan.

Before and during this scan, the AF algorithm stops the lens at each preselected position long enough to obtain valid sharpness scores. The first normalized score from each AF window is stored as both the worst (minimum) and best (maximum) score for that window. These two extreme scores are then updated as the lens moves from one position to the next and a new maximum position is memorized at every update of the maximum score. In effect, the preselected set of lens positions is scanned for maxima of the normalized sharpness scores, while at the same time information needed to validate each maximum is being collected.

Modes

There are two AF camera modes that the MT9P111 can fully support if it controls the position of the camera lens.

Snapshot mode

In this mode, a camera performs auto focusing upon a user command to do so. When the auto focusing is finished, a snapshot is normally taken and there is no further AF activity until the next appropriate user command. The MT9P111 can do the auto focusing using its built-in AF algorithm or a substitute algorithm loaded into RAM. It can then wait or automatically proceed with other operations required to take a snapshot.

Manual mode

In this mode there is no AF activity—focusing the camera is left to the user. The user typically can move the camera lens in steps, by manually issuing commands to the lens actuator, and observing the effect of his actions on a preview display. The MT9P111 can provide 30 fps image input for the display and simultaneously translate user commands received through the two-wire serial interface into digital waveforms driving the lens actuator.

Lens Actuator Interface

Actuators used to move lenses in AF cameras can be classified into several broad categories that differ significantly in their requirements for driving signals. These requirements also vary from one device to another within each category. To ensure its compatibility with many different actuators, the MT9P111 includes a general purpose input/output auto focus module.

The VGPIO is a programmable rectangular waveform generator, with eight individually controllable output signals (VGPIO0 through VGPIO7), a separate power supply pad (VDD_VGPIO), and a separate clock domain that can be disconnected from the master clock to save power when the VGPIO is not in use. The VGPIO can toggle its output signals as fast as half the master clock frequency.

An external host processor or the embedded microcontroller of the MT9P111 has two ways to control the voltages on the VGPIO output signals:

- Setting or clearing bits in a control register
  The state of the VGPIO signals is updated immediately after writing to the register. Because writing through the two-wire serial interface takes some time, this way does not give the host processor a very precise control over VGPIO output timing.
- Waveform programming
  The second way to obtain a desired output from the VGPIO is to program a set of periodic waveforms to the control registers and initialize their generation. The VGPIO then generates the programmed waveforms on its own, without waiting for any further input, and therefore with the best attainable timing precision. If necessary, the VGPIO can notify the MCU and the host processor about reaching certain points in the waveforms generation, for example, the end of a particular waveform.

The MT9P111 can be set up not only to output digital signals to a lens actuator and/or other similar devices, but also to receive their digital feedback. All VGPIO output signals are reconfigurable as high-impedance digital inputs. The logical state of each VGPIO pad is mirrored by the state of a bit in a dedicated register, which allows the MCU and host processor to sample digital input signals at intervals equal to their respective register read times.

In addition, the MT9P111 has an additional serial master available for use (S_CLK, S_DAT).

It may be implemented in such a way that if the auto-focus mechanisms are controlled by the serial master, all eight VGPIOs would be available for advanced flash and mechanical shutter operations.

Internal VCM Driver

The MT9P111 utilizes an internal Voice Coil Motor (VCM) driver. The VCM functions are register-controlled through the serial interface.

There are two output ports, VCM_OUT and GNDIO_VCM, which would connect directly to the AF actuator.

Take precautions in the design of the power supply routing to provide a low impedance path for the ground return. Appropriate filtering would also be required on the actuator supply. Typical values would be a 0.1µF and 10µF in parallel.

Figure 28: VCM Driver Typical Diagram

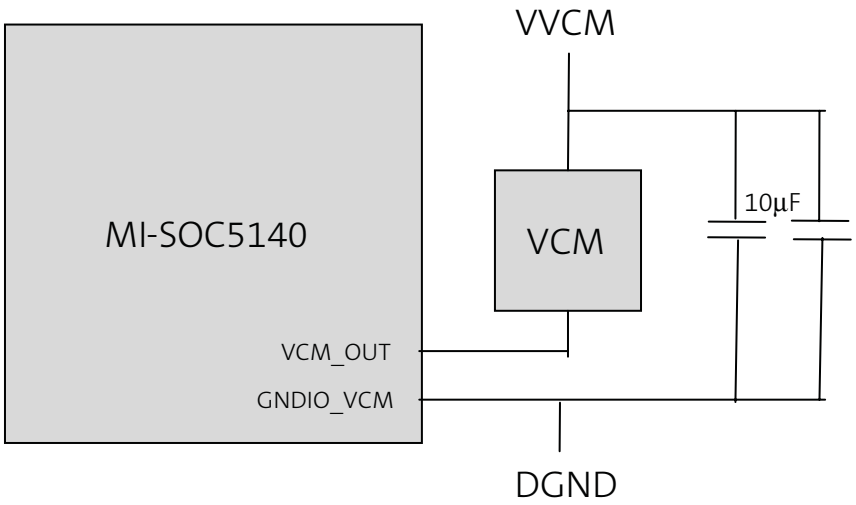


Table 12: VCM Driver Typical

Characteristic	Parameter	Min	Typ	Max	Units
VCM_OUT	Voltage at VCM current sink	2.5	2.8	3.3	V
WVCM	Voltage at VCM actuator	2.5	2.8	3.3	V
INL	Relative accuracy		±1.5	± 4	LSB
RES	Resolution		8		bits
DNL	Differential nonlinearity	–1		+1	LSB
IVCM	Output current	5		100	mA
	Slew rate		.3		mA/µs

User -Accessible Internal ADC

The MT9P111 provides access to an internal 12-bit ADC for customer use. The ADC provides sampling, correction, and filtering.

One application for the internal ADC is to use as an interface to AF drivers that require analog feedback support. The access to the internal ADC is through the ATEST0/1 pins and the data is accessible through the serial interface.

Multimaster Serial Interface

The MT9P111 provides, in addition to the standard serial interface (SDATA, SCLK), another two-wire serial interface for customer use. These could be used for any number of uses to control external components such as autofocus, mechanical shutter, and flash drivers.

These are the S_SCLK and S_DATA pins.

Master and Slave Two-Wire Serial Interface

The two-wire serial interface bus enables read/write access to control and status registers within the MT9P111. This interface is designed to be compatible with the MIPI Alliance Standard for Camera Serial Interface 2 (CSI-2) 1.0, which uses the electrical characteristics and transfer protocols of the two-wire serial interface specification.

The interface protocol uses a master/slave model in which a master controls one or more slave devices. The sensor acts as a slave device. The master generates a clock (SCLK) that is an input to the sensor and used to synchronize transfers.

Data is transferred between the master and the slave on a bidirectional signal (SDATA). SDATA is pulled up to VDD_IO off-chip by a 1.5KΩ resistor. Either the slave or master device can drive SDATA LOW—the interface protocol determines which device is allowed to drive SDATA at any given time.

The MT9P111 is a multi-master device. A separate serial master is provided for the control of external components. These are the S_CLK and S_DAT pins.

Protocol

Data transfers on the two-wire serial interface bus are performed by a sequence of low-level protocol elements, as follows:

- a start or restart condition
- a slave address/data direction byte
- a 16-bit register address
- an acknowledge or a no-acknowledge bit
- data bytes
- a stop condition

The bus is idle when both SCLK and SDATA are HIGH. Control of the bus is initiated with a start condition, and the bus is released with a stop condition. Only the master can generate the start and stop conditions.

Start Condition

A start condition is defined as a HIGH-to-LOW transition on SDATA while SCLK is HIGH. At the end of a transfer, the master can generate a start condition without previously generating a stop condition; this is known as a “repeated start” or “restart” condition.

Data Transfer

Data is transferred serially, 8 bits at a time, with the MSB transmitted first. Each byte of data is followed by an acknowledge bit or a no-acknowledge bit. This data transfer mechanism is used for the slave address/data direction byte and for message bytes.

One data bit is transferred during each SCLK clock period. SDATA can change when SCLK is low and must be stable while SCLK is HIGH.

Slave Address/Data Direction Byte

Bits [7:1] of this byte represent the device slave address and bit [0] indicates the data transfer direction. A “0” in bit [0] indicates a write, and a “1” indicates a read. The default slave addresses used by the MT9P111 are 0x78 (write address) and 0x79 (read address). Alternate slave addresses of 0x7A (write address) and 0x7B (read address) can be selected by asserting the SADDR input signal.

Message Byte

Message bytes are used for sending register addresses and register write data to the slave device and for retrieving register read data. The protocol used is outside the scope of the two-wire serial interface specification.

Acknowledge Bit

Each 8-bit data transfer is followed by an acknowledge bit or a no-acknowledge bit in the SCLK clock period following the data transfer. The transmitter (which is the master when writing, or the slave when reading) releases SDATA. The receiver indicates an acknowledge bit by driving SDATA LOW. As for data transfers, SDATA can change when SCLK is LOW and must be stable while SCLK is HIGH.

No-Acknowledge Bit

The no-acknowledge bit is generated when the receiver does not drive SDATA low during the SCLK clock period following a data transfer. A no-acknowledge bit is used to terminate a read sequence.

Stop Condition

A stop condition is defined as a LOW -to-HIGH transition on SDATA while SCLK is HIGH.

Typical Serial Transfer

A typical read or write sequence begins by the master generating a start condition on the bus. After the start condition, the master sends the 8-bit slave address/data direction byte. The last bit indicates whether the request is for a read or a write, where a “0” indicates a write and a “1” indicates a read. If the address matches the address of the slave device, the slave device acknowledges receipt of the address by generating an acknowledge bit on the bus.

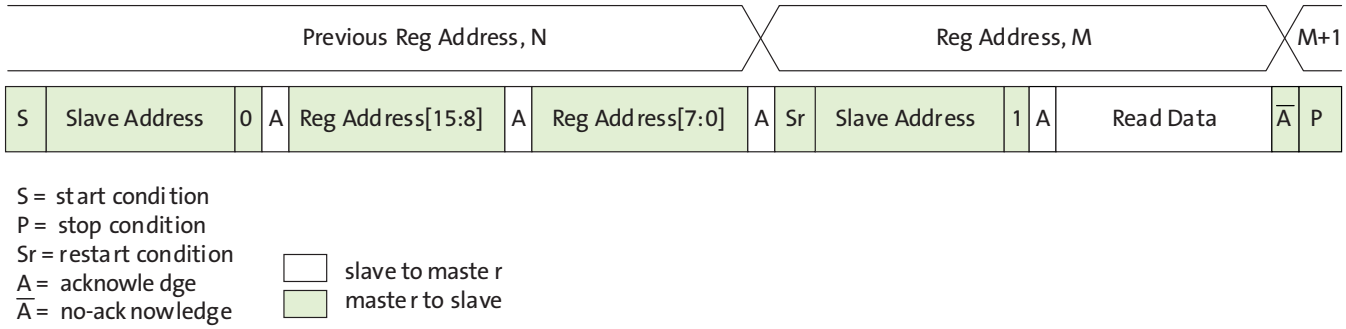
If the request was a write, the master then transfers the 16-bit register address to which a write should take place. This transfer takes place as two 8-bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master then transfers the data as an 8-bit sequence; the slave sends acknowledge bit at the end of the sequence. After 8 bits have been transferred, the slave’s internal register address is automatically incremented, so that the next 8 bits are written to the next register address. The master stops writing by generating a (re)start or stop condition.

If the request was a read, the master sends the 8-bit write slave address/data direction byte and 16-bit register address, just as in the write request. The master then generates a (re)start condition and the 8-bit read slave address/data direction byte, and clocks out the register data, 8 bits at a time. The master generates an acknowledge bit after each 8-bit transfer. The slave’s internal register address is automatically incremented after every 8 bits are transferred. The data transfer is stopped when the master sends a no-acknowledge bit.

Single Read from Random Location

This sequence (see Figure 29) starts with a dummy write to the 16-bit address that is to be used for the read. The master terminates the write by generating a restart condition. The master then sends the 8-bit read slave address/data direction byte and clocks out one byte of register data. The master terminates the read by generating a no-acknowledge bit followed by a stop condition. Figure 29 shows how the internal register address maintained by the MT9P111 is loaded and incremented as the sequence proceeds.

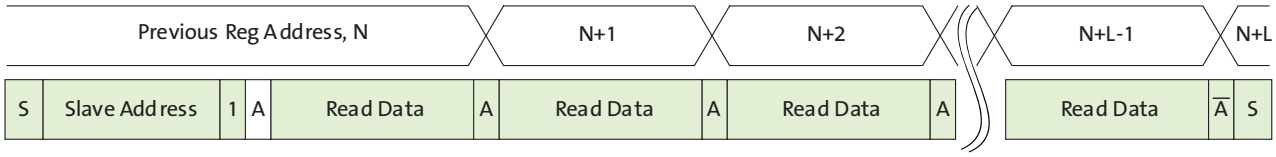
Figure 29: Single Read from Random Location



Single Read from Current Location

This sequence (Figure 30) performs a read using the current value of the MT9P111 internal register address. The master terminates the read by generating a no-acknowledge bit followed by a stop condition. The figure shows two independent read sequences.

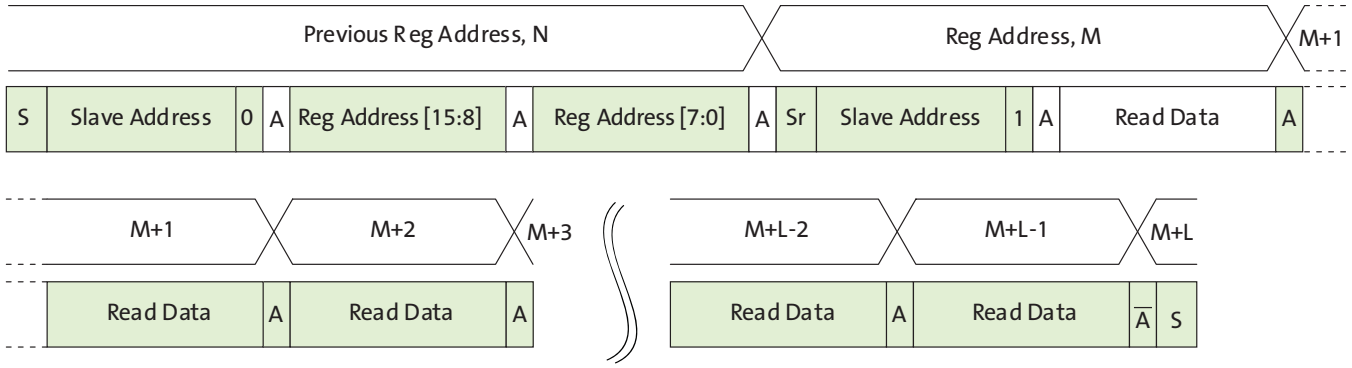
Figure 30: Single Read from Current Location



Sequential Read, Start from Random Location

This sequence (Figure 31) starts in the same way as the single read from random location (Figure 29). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit, and continues to perform byte reads until L bytes have been read.

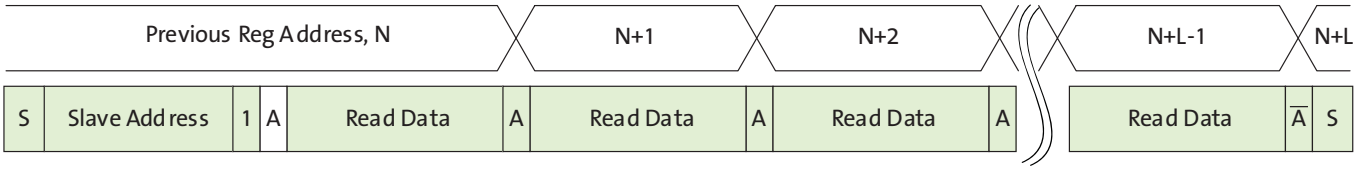
Figure 31: Sequential Read, Start from Random Location



Sequential Read, Start from Current Location

This sequence (Figure 32) starts in the same way as the single read from current location (Figure 30). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit, and continues to perform byte reads until L bytes have been read.

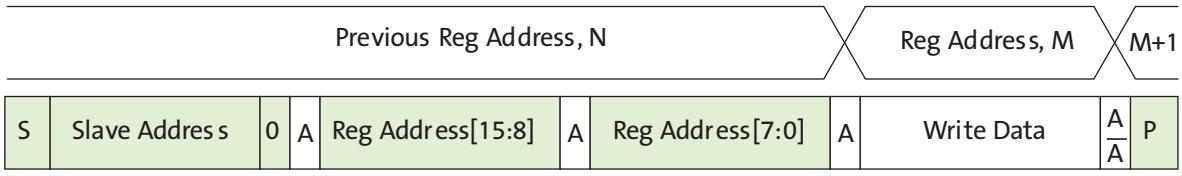
Figure 32: Sequential Read, Start from Current Location



Single Write to Random Location

This sequence (Figure 33) begins with the master generating a start condition. The slave address/data direction byte signals a write and is followed by the high then low bytes of the register address that is to be written. The master follows this with the byte of write data. The write is terminated by the master generating a stop condition.

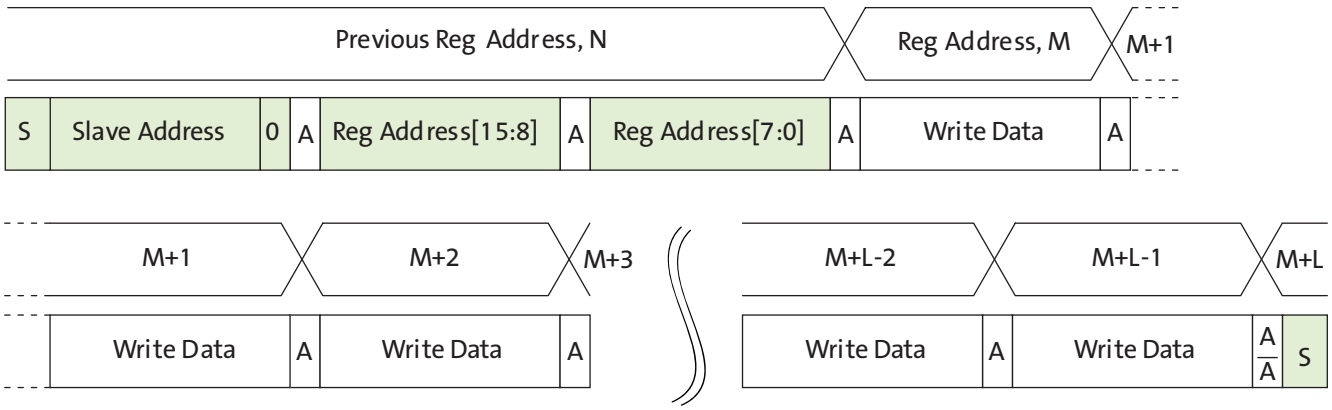
Figure 33: Single Write to Random Location



Sequential Write, Start at Random Location

This sequence (Figure 34) starts in the same way as the single write to random location (Figure 33). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit, and continues to perform byte writes until L bytes have been written. The write is terminated by the master generating a stop condition.

Figure 34: Sequential Write, Start at Random Location



Register Tables

Sorted By Number

Table 13: GPIO_SS Registers

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1536(R0x00000600)	second_snsr	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R1538(R0x00000602)	second_snsr_oe	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)
R1544(R0x00000608)	vgpio_int_ctl	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1546(R0x0000060A)	vgpio_int_status_clr	0000 0000 0000 0000 dddd dddd ???? ????	0 (0x00000000)
R1548(R0x0000060C)	vgpio_data_from_icb	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1550(R0x0000060E)	vgpio_dir	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1552(R0x00000610)	vgpio_data_status	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R1554(R0x00000612)	vgpio_data_alt_sel	0000 0000 0000 0000 d0dd 0000 dddd dddd	0 (0x00000000)
R1556(R0x00000614)	second_scl_sda_pd	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)
R1558(R0x00000616)	vgpio_data_rx_tx_sel	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R1568(R0x00000620)	vcm_control	0000 0000 0000 0000 d0dd dddd dd00 dddd	0 (0x00000000)
R1790(R0x000006FE)	wg_gr_trigger_sel	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)
R1792(R0x00000700)	ch0_cfg	0000 0000 0000 0000 000? ???d dddd 0ddd	0 (0x00000000)
R1794(R0x00000702)	wg0_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1796(R0x00000704)	wg0_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1798(R0x00000706)	wg0_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1800(R0x00000708)	wg0_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1802(R0x0000070A)	wg0_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1804(R0x0000070C)	wg0_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1806(R0x0000070E)	wg0_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1808(R0x00000710)	wg0_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 13: GPIO_SS Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1810(R0x00000712)	wg0_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1812(R0x00000714)	wg0_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1814(R0x00000716)	wg0_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1816(R0x00000718)	wg0_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1818(R0x0000071A)	wg0_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1820(R0x0000071C)	wg0_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1822(R0x0000071E)	wg0_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1824(R0x00000720)	ch1_cfg	0000 0000 0000 0000 000? ???d dddd 0ddd	0 (0x00000000)
R1826(R0x00000722)	wg1_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1828(R0x00000724)	wg1_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1830(R0x00000726)	wg1_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1832(R0x00000728)	wg1_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1834(R0x0000072A)	wg1_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1836(R0x0000072C)	wg1_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1838(R0x0000072E)	wg1_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1840(R0x00000730)	wg1_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1842(R0x00000732)	wg1_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1844(R0x00000734)	wg1_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1846(R0x00000736)	wg1_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1848(R0x00000738)	wg1_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1850(R0x0000073A)	wg1_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1852(R0x0000073C)	wg1_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1854(R0x0000073E)	wg1_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1856(R0x00000740)	ch2_cfg	0000 0000 0000 0000 000? ???d dddd 0ddd	0 (0x00000000)

Table 13: GPIO_SS Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1858(R0x00000742)	wg2_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1860(R0x00000744)	wg2_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1862(R0x00000746)	wg2_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1864(R0x00000748)	wg2_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1866(R0x0000074A)	wg2_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1868(R0x0000074C)	wg2_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1870(R0x0000074E)	wg2_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1872(R0x00000750)	wg2_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1874(R0x00000752)	wg2_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1876(R0x00000754)	wg2_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1878(R0x00000756)	wg2_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1880(R0x00000758)	wg2_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1882(R0x0000075A)	wg2_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1884(R0x0000075C)	wg2_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1886(R0x0000075E)	wg2_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1888(R0x00000760)	ch3_cfg	0000 0000 0000 0000 000? ???d dddd 0ddd	0 (0x00000000)
R1890(R0x00000762)	wg3_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1892(R0x00000764)	wg3_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1894(R0x00000766)	wg3_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1896(R0x00000768)	wg3_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1898(R0x0000076A)	wg3_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1900(R0x0000076C)	wg3_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1902(R0x0000076E)	wg3_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1904(R0x00000770)	wg3_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 13: GPIO_SS Registers (continued)

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R1906(R0x00000772)	wg3_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1908(R0x00000774)	wg3_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1910(R0x00000776)	wg3_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1912(R0x00000778)	wg3_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1914(R0x0000077A)	wg3_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1916(R0x0000077C)	wg3_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1918(R0x0000077E)	wg3_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1920(R0x00000780)	ch4_cfg	0000 0000 0000 0000 000? ???d dddd 0ddd	0 (0x00000000)
R1922(R0x00000782)	wg4_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1924(R0x00000784)	wg4_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1926(R0x00000786)	wg4_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1928(R0x00000788)	wg4_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1930(R0x0000078A)	wg4_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1932(R0x0000078C)	wg4_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1934(R0x0000078E)	wg4_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1936(R0x00000790)	wg4_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1938(R0x00000792)	wg4_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1940(R0x00000794)	wg4_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1942(R0x00000796)	wg4_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1944(R0x00000798)	wg4_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1946(R0x0000079A)	wg4_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1948(R0x0000079C)	wg4_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1950(R0x0000079E)	wg4_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1952(R0x000007A0)	ch5_cfg	0000 0000 0000 0000 000? ???d dddd 0ddd	0 (0x00000000)

Table 13: GPIO_SS Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1954(R0x000007A2)	wg5_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1956(R0x000007A4)	wg5_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1958(R0x000007A6)	wg5_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1960(R0x000007A8)	wg5_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1962(R0x000007AA)	wg5_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1964(R0x000007AC)	wg5_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1966(R0x000007AE)	wg5_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1968(R0x000007B0)	wg5_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1970(R0x000007B2)	wg5_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1972(R0x000007B4)	wg5_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1974(R0x000007B6)	wg5_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1976(R0x000007B8)	wg5_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1978(R0x000007BA)	wg5_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1980(R0x000007BC)	wg5_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1982(R0x000007BE)	wg5_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1984(R0x000007C0)	ch6_cfg	0000 0000 0000 0000 000? ???d dddd 0ddd	0 (0x00000000)
R1986(R0x000007C2)	wg6_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1988(R0x000007C4)	wg6_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1990(R0x000007C6)	wg6_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1992(R0x000007C8)	wg6_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1994(R0x000007CA)	wg6_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1996(R0x000007CC)	wg6_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1998(R0x000007CE)	wg6_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2000(R0x000007D0)	wg6_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 13: GPIO_SS Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R2002(R0x000007D2)	wg6_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2004(R0x000007D4)	wg6_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2006(R0x000007D6)	wg6_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2008(R0x000007D8)	wg6_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2010(R0x000007DA)	wg6_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R2012(R0x000007DC)	wg6_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2014(R0x000007DE)	wg6_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R2016(R0x000007E0)	ch7_cfg	0000 0000 0000 0000 000? ???d dddd 0ddd	0 (0x00000000)
R2018(R0x000007E2)	wg7_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R2020(R0x000007E4)	wg7_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R2022(R0x000007E6)	wg7_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2024(R0x000007E8)	wg7_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2026(R0x000007EA)	wg7_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2028(R0x000007EC)	wg7_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2030(R0x000007EE)	wg7_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2032(R0x000007F0)	wg7_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2034(R0x000007F2)	wg7_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2036(R0x000007F4)	wg7_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2038(R0x000007F6)	wg7_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2040(R0x000007F8)	wg7_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2042(R0x000007FA)	wg7_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R2044(R0x000007FC)	wg7_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2046(R0x000007FE)	wg7_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)

Table 14: 0: Sensor Core Registers

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12288(R0x00003000)	model_id_	0000 0000 0000 0000 dddd dddd dddd dddd	10368 (0x00002880)
R12290(R0x00003002)	y_addr_start_	0000 0000 0000 0000 0000 dddd dddd dddd	16 (0x00000010)
R12292(R0x00003004)	x_addr_start_	0000 0000 0000 0000 0000 dddd dddd dddd	28 (0x0000001C)
R12294(R0x00003006)	y_addr_end_	0000 0000 0000 0000 0000 dddd dddd dddd	1967 (0x000007AF)
R12296(R0x00003008)	x_addr_end_	0000 0000 0000 0000 0000 dddd dddd dddd	2627 (0x00000A43)
R12298(R0x0000300A)	frame_length_lines_	0000 0000 0000 0000 dddd dddd dddd dddd	2031 (0x000007EF)
R12300(R0x0000300C)	line_length_pck_	0000 0000 0000 0000 dddd dddd dddd ddd0	3878 (0x00000F26)
R12304(R0x00003010)	fine_correction	0000 0000 0000 0000 0ddd dddd dddd dddd	156 (0x0000009C)
R12306(R0x00003012)	coarse_integration_time_	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)
R12308(R0x00003014)	fine_integration_time_	0000 0000 0000 0000 dddd dddd dddd ddd0	842 (0x0000034A)
R12310(R0x00003016)	row_speed	0000 0000 0000 0000 0000 0ddd 0ddd 0ddd	273 (0x00000111)
R12312(R0x00003018)	extra_delay	0000 0000 0000 0000 dddd dddd dddd ddd0	0 (0x00000000)
R12314(R0x0000301A)	reset_register	0000 0000 0000 0000 dd0d 0ddd dddd dddd	4312 (0x000010D8)
R12316(R0x0000301C)	mode_select_	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R12317(R0x0000301D)	image_orientation_	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R12318(R0x0000301E)	data_pedestal_	0000 0000 0000 0000 0000 dddd dddd dddd	168 (0x000000A8)
R12321(R0x00003021)	software_reset_	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R12322(R0x00003022)	grouped_parameter_hold_	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R12323(R0x00003023)	mask_corrupted_frames_	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R12324(R0x00003024)	pixel_order_	0000 0000 0000 0000 0000 0000 0000 00??	0 (0x00000000)
R12326(R0x00003026)	gpi_status	0000 0000 0000 0000 dddd dddd dddd ????	65535 (0x0000FFFF)
R12328(R0x00003028)	analogue_gain_code_global_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)
R12330(R0x0000302A)	analogue_gain_code_greenr_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)
R12332(R0x0000302C)	analogue_gain_code_red_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)

Table 14: 0: Sensor Core Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12334(R0x0000302E)	analogue_gain_code_blue_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)
R12336(R0x00003030)	analogue_gain_code_greenb_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)
R12338(R0x00003032)	digital_gain_greenr_	0000 0000 0000 0000 0000 0ddd 0000 0000	256 (0x00000100)
R12340(R0x00003034)	digital_gain_red_	0000 0000 0000 0000 0000 0ddd 0000 0000	256 (0x00000100)
R12342(R0x00003036)	digital_gain_blue_	0000 0000 0000 0000 0000 0ddd 0000 0000	256 (0x00000100)
R12344(R0x00003038)	digital_gain_greenb_	0000 0000 0000 0000 0000 0ddd 0000 0000	256 (0x00000100)
R12346(R0x0000303A)	smia_version_	0000 0000 0000 0000 0000 0000 ???? ????	10 (0x0000000A)
R12347(R0x0000303B)	frame_count_	0000 0000 0000 0000 0000 0000 ???? ????	255 (0x000000FF)
R12348(R0x0000303C)	frame_status	0000 0000 0000 0000 0000 0000 0000 00??	0 (0x00000000)
R12352(R0x00003040)	read_mode	0000 0000 0000 0000 dddd dddd dddd dddd	65 (0x00000041)
R12358(R0x00003046)	flash	0000 0000 0000 0000 ??dd dddd dddd 0000	1536 (0x00000600)
R12360(R0x00003048)	flash_count	0000 0000 0000 0000 dddd dddd dddd dddd	8 (0x00000008)
R12374(R0x00003056)	green1_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)
R12376(R0x00003058)	blue_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)
R12378(R0x0000305A)	red_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)
R12380(R0x0000305C)	green2_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)
R12382(R0x0000305E)	global_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)
R12400(R0x00003070)	test_pattern_mode_	0000 0000 0000 0000 0000 000d 0000 0ddd	0 (0x00000000)
R12402(R0x00003072)	test_data_red_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12404(R0x00003074)	test_data_greenr_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12406(R0x00003076)	test_data_blue_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12408(R0x00003078)	test_data_greenb_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12410(R0x0000307A)	test_raw_mode	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R12430(R0x0000308E)	ease_control	0000 0000 0000 0000 dddd dddd dddd dddd	52224 (0x0000CC00)

Table 14: 0: Sensor Core Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12432(R0x00003090)	ease_top_gain	0000 0000 0000 0000 0000 000d dddd dddd	32 (0x00000020)
R12434(R0x00003092)	ease_btm_gain	0000 0000 0000 0000 0000 000d dddd dddd	33 (0x00000021)
R12436(R0x00003094)	ease_top_calib	0000 0000 0000 0000 0000 000d dddd dddd	3 (0x00000003)
R12438(R0x00003096)	ease_btm_calib	0000 0000 0000 0000 0000 000d dddd dddd	5 (0x00000005)
R12440(R0x00003098)	ease_top_samp_mux	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12442(R0x0000309A)	ease_btm_samp_mux	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12444(R0x0000309C)	ease_top_ref_mux	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12446(R0x0000309E)	ease_btm_ref_mux	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12448(R0x000030A0)	x_even_inc_	0000 0000 0000 0000 0000 0000 0000 000?	1 (0x00000001)
R12450(R0x000030A2)	x_odd_inc_	0000 0000 0000 0000 0000 0000 0000 dddd	1 (0x00000001)
R12452(R0x000030A4)	y_even_inc_	0000 0000 0000 0000 0000 0000 0000 000?	1 (0x00000001)
R12454(R0x000030A6)	y_odd_inc_	0000 0000 0000 0000 0000 0000 00dd dddd	1 (0x00000001)
R12634(R0x0000315A)	global_flash_start	0000 0000 0000 0000 dddd dddd dddd dddd	154 (0x0000009A)
R12636(R0x0000315C)	global_bulb_trigger_count	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R12638(R0x0000315E)	global_seq_trigger	0000 0000 0000 0000 dddd 0d?? dddd 0ddd	0 (0x00000000)
R12640(R0x00003160)	global_rst_end	0000 0000 0000 0000 dddd dddd dddd dddd	152 (0x00000098)
R12642(R0x00003162)	global_shutter_start	0000 0000 0000 0000 dddd dddd dddd dddd	159 (0x0000009F)
R12644(R0x00003164)	global_shutter_start2	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R12646(R0x00003166)	global_read_start	0000 0000 0000 0000 dddd dddd dddd dddd	160 (0x000000A0)
R12648(R0x00003168)	global_read_start2	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R12776(R0x000031E8)	horizontal_cursor_position_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12778(R0x000031EA)	vertical_cursor_position_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12780(R0x000031EC)	horizontal_cursor_width_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12782(R0x000031EE)	vertical_cursor_width_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)

Table 14: 0: Sensor Core Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12786(R0x000031F2)	i2c_ids_mipi_default	0000 0000 0000 0000 dddd dddd dddd dddd	28268 (0x00006E6C)
R12796(R0x000031FC)	i2c_ids	0000 0000 0000 0000 dddd dddd dddd dddd	12320 (0x00003020)

Table 15: GPIO_SS Registers

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15104(R0x00003B00)	txbuffer_data_register_0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15106(R0x00003B02)	txbuffer_data_register_1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15108(R0x00003B04)	txbuffer_data_register_2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15110(R0x00003B06)	txbuffer_data_register_3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15112(R0x00003B08)	txbuffer_data_register_4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15114(R0x00003B0A)	txbuffer_data_register_5	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15116(R0x00003B0C)	txbuffer_data_register_6	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15118(R0x00003B0E)	txbuffer_data_register_7	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15120(R0x00003B10)	txbuffer_data_register_8	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15122(R0x00003B12)	txbuffer_data_register_9	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15124(R0x00003B14)	txbuffer_data_register_10	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15126(R0x00003B16)	txbuffer_data_register_11	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15128(R0x00003B18)	txbuffer_data_register_12	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15130(R0x00003B1A)	txbuffer_data_register_13	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15132(R0x00003B1C)	txbuffer_data_register_14	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15134(R0x00003B1E)	txbuffer_data_register_15	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15136(R0x00003B20)	txbuffer_data_register_16	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15138(R0x00003B22)	txbuffer_data_register_17	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15140(R0x00003B24)	txbuffer_data_register_18	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 15: GPIO_SS Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15142(R0x00003B26)	txbuffer_data_register_19	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15144(R0x00003B28)	txbuffer_data_register_20	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15146(R0x00003B2A)	txbuffer_data_register_21	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15148(R0x00003B2C)	txbuffer_data_register_22	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15150(R0x00003B2E)	txbuffer_data_register_23	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15152(R0x00003B30)	txbuffer_data_register_24	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15154(R0x00003B32)	txbuffer_data_register_25	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15156(R0x00003B34)	txbuffer_data_register_26	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15158(R0x00003B36)	txbuffer_data_register_27	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15160(R0x00003B38)	txbuffer_data_register_28	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15162(R0x00003B3A)	txbuffer_data_register_29	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15164(R0x00003B3C)	txbuffer_data_register_30	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15166(R0x00003B3E)	txbuffer_data_register_31	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15168(R0x00003B40)	rxbuffer_data_register_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15170(R0x00003B42)	rxbuffer_data_register_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15172(R0x00003B44)	rxbuffer_data_register_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15174(R0x00003B46)	rxbuffer_data_register_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15176(R0x00003B48)	rxbuffer_data_register_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15178(R0x00003B4A)	rxbuffer_data_register_5	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15180(R0x00003B4C)	rxbuffer_data_register_6	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15182(R0x00003B4E)	rxbuffer_data_register_7	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15184(R0x00003B50)	rxbuffer_data_register_8	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15186(R0x00003B52)	rxbuffer_data_register_9	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15188(R0x00003B54)	rxbuffer_data_register_10	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)

Table 15: GPIO_SS Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15190(R0x00003B56)	rxbuffer_data_register_11	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15192(R0x00003B58)	rxbuffer_data_register_12	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15194(R0x00003B5A)	rxbuffer_data_register_13	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15196(R0x00003B5C)	rxbuffer_data_register_14	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15198(R0x00003B5E)	rxbuffer_data_register_15	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15200(R0x00003B60)	rxbuffer_data_register_16	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15202(R0x00003B62)	rxbuffer_data_register_17	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15204(R0x00003B64)	rxbuffer_data_register_18	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15206(R0x00003B66)	rxbuffer_data_register_19	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15208(R0x00003B68)	rxbuffer_data_register_20	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15210(R0x00003B6A)	rxbuffer_data_register_21	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15212(R0x00003B6C)	rxbuffer_data_register_22	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15214(R0x00003B6E)	rxbuffer_data_register_23	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15216(R0x00003B70)	rxbuffer_data_register_24	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15218(R0x00003B72)	rxbuffer_data_register_25	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15220(R0x00003B74)	rxbuffer_data_register_26	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15222(R0x00003B76)	rxbuffer_data_register_27	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15224(R0x00003B78)	rxbuffer_data_register_28	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15226(R0x00003B7A)	rxbuffer_data_register_29	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15228(R0x00003B7C)	rxbuffer_data_register_30	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15230(R0x00003B7E)	rxbuffer_data_register_31	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15232(R0x00003B80)	i2c_master_status	0000 0000 0000 0000 0000 0000 000? ????	9 (0x00000009)
R15234(R0x00003B82)	i2c_master_control	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)
R15236(R0x00003B84)	i2c_master_frequency_divider	0000 0000 0000 0000 0000 00dd dddd dddd	99 (0x00000063)

Table 15: GPIO_SS Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R15238(R0x00003B86)	txbuffer_total_byte_count	0000 0000 0000 0000 0000 0000 00dd dddd	0 (0x00000000)

Table 16: 1: SOC1 Registers

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12816(R0x00003210)	color_pipeline_control	0000 0000 0000 0000 0ddd dd0d dddd d000	416 (0x000001A0)
R12834(R0x00003222)	zoom_window_x0	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12836(R0x00003224)	zoom_window_x1	0000 0000 0000 0000 0000 dddd dddd dddd	2591 (0x00000A1F)
R12838(R0x00003226)	zoom_window_y0	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12840(R0x00003228)	zoom_window_y1	0000 0000 0000 0000 0000 dddd dddd dddd	1943 (0x00000797)
R12844(R0x0000322C)	x_ratio	0000 0000 0000 0000 0000 dddd dddd dddd	2048 (0x00000800)
R12846(R0x0000322E)	y_ratio	0000 0000 0000 0000 0000 dddd dddd dddd	2048 (0x00000800)
R12852(R0x00003234)	awb_debug	0000 0000 0000 0000 0000 0000 000d dddd	X
R12854(R0x00003236)	awb_norm_sumr	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R12856(R0x00003238)	awb_norm_sumg	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R12858(R0x0000323A)	awb_norm_sumb	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R12860(R0x0000323C)	awb_x_shift	0000 0000 0000 0000 0000 0ddd dddd dddd	3 (0x00000003)
R12862(R0x0000323E)	awb_y_shift	0000 0000 0000 0000 0000 0ddd dddd dddd	3 (0x00000003)
R12864(R0x00003240)	awb_xy_scale	0000 0000 0000 0000 0000 0000 dddd dddd	51 (0x00000033)
R12866(R0x00003242)	awb_weight_r0	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12868(R0x00003244)	awb_weight_r1	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12870(R0x00003246)	awb_weight_r2	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12872(R0x00003248)	awb_weight_r3	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12874(R0x0000324A)	awb_weight_r4	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12876(R0x0000324C)	awb_weight_r5	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12878(R0x0000324E)	awb_weight_r6	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)

Table 16: 1: SOC1 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12880(R0x00003250)	awb_weight_r7	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12884(R0x00003254)	first_color	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R12886(R0x00003256)	last_row	0000 0000 0000 0000 0000 0ddd dddd dddd	1947 (0x0000079B)
R12888(R0x00003258)	awb_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12890(R0x0000325A)	awb_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R12892(R0x0000325C)	awb_win_width	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R12894(R0x0000325E)	awb_win_height	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R12896(R0x00003260)	awb_weight_cnt_hi	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R12898(R0x00003262)	awb_luma_th	0000 0000 0000 0000 dddd dddd dddd dddd	65288 (0x0000FF08)
R12900(R0x00003264)	awb_config	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R12902(R0x00003266)	awb_weight_th	0000 0000 0000 0000 0000 0000 dddd dddd	145 (0x00000091)
R12904(R0x00003268)	awb_weight_cnt_lo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R12906(R0x0000326A)	scale_sharp_control	0000 0000 0000 0000 00dd dddd dddd dddd	4616 (0x00001208)
R12910(R0x0000326E)	low_pass_yuv_filter	0000 0000 0000 0000 0000 0000 dddd dddd	128 (0x00000080)
R12912(R0x00003270)	threshold_for_y_filter_r_channel	0000 0000 0000 0000 0000 dddd dddd dddd	1962 (0x000007AA)
R12914(R0x00003272)	threshold_for_y_filter_g_channel	0000 0000 0000 0000 0000 dddd dddd dddd	2020 (0x000007E4)
R12916(R0x00003274)	threshold_for_y_filter_b_channel	0000 0000 0000 0000 0000 0000 0ddd dddd	42 (0x0000002A)
R12918(R0x00003276)	black_level_to_ccm	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R12922(R0x0000327A)	red_offset	0000 0000 0000 0000 0000 000d dddd dddd	168 (0x000000A8)
R12924(R0x0000327C)	green1_offset	0000 0000 0000 0000 0000 000d dddd dddd	168 (0x000000A8)
R12926(R0x0000327E)	green2_offset	0000 0000 0000 0000 0000 000d dddd dddd	168 (0x000000A8)
R12928(R0x00003280)	blue_offset	0000 0000 0000 0000 0000 000d dddd dddd	168 (0x000000A8)
R12930(R0x00003282)	aperture_knee_gain_parameters	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R12932(R0x00003284)	aperture_knee_values	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 16: 1: SOC1 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12934(R0x00003286)	aperture_gain_value	0000 0000 0000 0000 0000 0000 0ddd dddd	0 (0x00000000)
R12936(R0x00003288)	aperture_additive_clip_limit	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R12938(R0x0000328A)	sffb_noise_control_r	0000 0000 0000 0000 dddd dddd dddd dddd	33036 (0x0000810C)
R12940(R0x0000328C)	sffb_noise_control_g	0000 0000 0000 0000 dddd dddd dddd dddd	33036 (0x0000810C)
R12942(R0x0000328E)	sffb_noise_control_b	0000 0000 0000 0000 dddd dddd dddd dddd	33036 (0x0000810C)
R12944(R0x00003290)	sffb_weight_control	0000 0000 0000 0000 0ddd 0ddd dddd 0ddd	21508 (0x00005404)
R12946(R0x00003292)	sffb_sobel_flat	0000 0000 0000 0000 0000 0000 dddd dddd	31 (0x0000001F)
R12948(R0x00003294)	sffb_sobel_sharp	0000 0000 0000 0000 0000 0000 dddd dddd	255 (0x000000FF)
R12958(R0x0000329E)	preview_hunting_gain_enable	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R12960(R0x000032A0)	dkdelta_ccm_cc1	0000 0000 0000 0000 0000 000d dddd dddd	322 (0x00000142)
R12962(R0x000032A2)	dkdelta_ccm_cc2	0000 0000 0000 0000 0000 000d dddd dddd	217 (0x000000D9)
R12964(R0x000032A4)	dkdelta_ccm_cc3	0000 0000 0000 0000 0000 000d dddd dddd	485 (0x000001E5)
R12966(R0x000032A6)	dkdelta_ccm_cc4	0000 0000 0000 0000 0000 000d dddd dddd	43 (0x0000002B)
R12968(R0x000032A8)	dkdelta_ccm_cc5	0000 0000 0000 0000 0000 000d dddd dddd	327 (0x00000147)
R12970(R0x000032AA)	dkdelta_ccm_cc6	0000 0000 0000 0000 0000 000d dddd dddd	142 (0x0000008E)
R12972(R0x000032AC)	dkdelta_ccm_cc7	0000 0000 0000 0000 0000 000d dddd dddd	490 (0x000001EA)
R12974(R0x000032AE)	dkdelta_ccm_cc8	0000 0000 0000 0000 0000 000d dddd dddd	125 (0x0000007D)
R12976(R0x000032B0)	dkdelta_ccm_cc9	0000 0000 0000 0000 0000 000d dddd dddd	409 (0x00000199)
R12978(R0x000032B2)	dkdelta_ccm_ctl	0000 0000 0000 0000 00dd dddd dddd dddd	8980 (0x00002314)
R12980(R0x000032B4)	dkdelta_ccm_scale	0000 0000 0000 0000 0000 00dd dddd dddd	528 (0x00000210)
R12982(R0x000032B6)	dkdelta_ccm_exp1	0000 0000 0000 0000 0ddd dddd dddd dddd	18724 (0x00004924)
R12984(R0x000032B8)	dkdelta_ccm_exp2	0000 0000 0000 0000 0000 dddd dddd dddd	2340 (0x00000924)
R12986(R0x000032BA)	red_offset_to_ccm	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R12988(R0x000032BC)	green_offset_to_ccm	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)

Table 16: 1: SOC1 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12990(R0x000032BE)	blue_offset_to_ccm	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R12992(R0x000032C0)	ccm_exp_low_byte	0000 0000 0000 0000 0ddd dddd dddd dddd	14627 (0x00003923)
R12994(R0x000032C2)	ccm_exp_high_byte	0000 0000 0000 0000 0000 dddd dddd dddd	1828 (0x00000724)
R12996(R0x000032C4)	ccm_elements_1_and_2	0000 0000 0000 0000 dddd dddd dddd dddd	55790 (0x0000D9EE)
R12998(R0x000032C6)	ccm_elements_3_and_4	0000 0000 0000 0000 dddd dddd dddd dddd	11035 (0x00002B1B)
R13000(R0x000032C8)	ccm_elements_5_and_6	0000 0000 0000 0000 dddd dddd dddd dddd	35818 (0x00008BEA)
R13002(R0x000032CA)	ccm_elements_7_and_8	0000 0000 0000 0000 dddd dddd dddd dddd	32022 (0x00007D16)
R13004(R0x000032CC)	ccm_elements_9_and_signs	0000 0000 0000 0000 00dd dddd dddd dddd	11714 (0x00002DC2)
R13012(R0x000032D4)	red_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13014(R0x000032D6)	green1_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13016(R0x000032D8)	green2_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13018(R0x000032DA)	blue_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13020(R0x000032DC)	second_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13042(R0x000032F2)	fd_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13044(R0x000032F4)	fd_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13046(R0x000032F6)	fd_win_width	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R13048(R0x000032F8)	fd_win_height	0000 0000 0000 0000 0000 0ddd dddd dddd	5 (0x00000005)
R13050(R0x000032FA)	fd_sum	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13100(R0x0000332C)	fm_blank_frames	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R13102(R0x0000332E)	output_format_configuration	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13104(R0x00003330)	output_format_test	0000 0000 0000 0000 0000 d00d d0dd dddd	0 (0x00000000)
R13106(R0x00003332)	fm_line_count	0000 0000 0000 0000 0000 ???? ???? ????	0 (0x00000000)
R13108(R0x00003334)	fm_frame_count	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13128(R0x00003348)	special_effect_parameters	0000 0000 0000 0000 dddd dddd 0ddd dddd	25664 (0x00006440)

Table 16: 1: SOC1 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13130(R0x0000334A)	sepia_constants	0000 0000 0000 0000 dddd dddd dddd dddd	45091 (0x0000B023)
R13136(R0x00003350)	cdc15_hi_thr_comb	0000 0000 0000 0000 0000 00dd dddd dddd	48 (0x00000030)
R13138(R0x00003352)	cdc15_hi_thr_satur	0000 0000 0000 0000 0000 000d dddd dddd	240 (0x000000F0)
R13140(R0x00003354)	cdc15_lo_thr_comb	0000 0000 0000 0000 0000 00dd dddd dddd	288 (0x00000120)
R13142(R0x00003356)	cdc15_lo_thr_satur	0000 0000 0000 0000 0000 000d dddd dddd	368 (0x00000170)
R13144(R0x00003358)	pcr_color_kill_control	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13146(R0x0000335A)	pcr_color_gain1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13148(R0x0000335C)	pcr_color_gain2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13150(R0x0000335E)	pcr_color_gain3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13152(R0x00003360)	skin_tone_th1	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13154(R0x00003362)	skin_tone_th2	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13156(R0x00003364)	skin_tone_th3	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13158(R0x00003366)	skin_tone_th4	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13160(R0x00003368)	skin_tone_th5	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13162(R0x0000336A)	skin_tone_th6	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13164(R0x0000336C)	pcr_color_gain4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13166(R0x0000336E)	pcr_color_gain5	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13168(R0x00003370)	pcr_color_gain6	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13170(R0x00003372)	pcr_color_gain7	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13172(R0x00003374)	pcr_color_gain8	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13174(R0x00003376)	pcr_color_gain9	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13180(R0x0000337C)	yuv_ycbcr_control	0000 0000 0000 0000 0000 0000 0000 dddd	6 (0x00000006)
R13182(R0x0000337E)	y_rgb_offset	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13184(R0x00003380)	kernel_config	0000 0000 0000 0000 d00d dd0d dddd dddd	1231 (0x000004CF)

Table 16: 1: SOC1 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13186(R0x00003382)	cdc_hi_thr_comb	0000 0000 0000 0000 0000 00dd dddd dddd	48 (0x00000030)
R13188(R0x00003384)	cdc_hi_thr_satur	0000 0000 0000 0000 0000 000d dddd dddd	240 (0x000000F0)
R13190(R0x00003386)	cdc_lo_thr_comb	0000 0000 0000 0000 0000 00dd dddd dddd	288 (0x00000120)
R13192(R0x00003388)	cdc_lo_thr_satur	0000 0000 0000 0000 0000 000d dddd dddd	368 (0x00000170)
R13194(R0x0000338A)	bnr_minmax_thresh	0000 0000 0000 0000 0000 0000 dddd dddd	128 (0x00000080)
R13196(R0x0000338C)	bnr_thresh_low_red	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13200(R0x00003390)	bnr_thresh_gain_red	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13202(R0x00003392)	bnr_thresh_low_green	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13206(R0x00003396)	bnr_thresh_gain_green	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13208(R0x00003398)	bnr_thresh_low_blue	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13212(R0x0000339C)	bnr_thresh_gain_blue	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13214(R0x0000339E)	bnr_blend_strength	0000 0000 0000 0000 0ddd dddd dddd dddd	12684 (0x0000318C)
R13216(R0x000033A0)	grb_pos_thresholds	0000 0000 0000 0000 dddd dddd dddd dddd	4104 (0x00001008)
R13218(R0x000033A2)	grb_neg_thresholds	0000 0000 0000 0000 dddd dddd dddd dddd	4104 (0x00001008)
R13220(R0x000033A4)	grb_position_pos	0000 0000 0000 0000 dddd dddd dddd dddd	3855 (0x00000F0F)
R13222(R0x000033A6)	grb_position_neg	0000 0000 0000 0000 dddd dddd dddd dddd	3855 (0x00000F0F)
R13224(R0x000033A8)	grb_position_window_x0	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13226(R0x000033AA)	grb_position_window_y0	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13228(R0x000033AC)	grb_position_window_x1	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13230(R0x000033AE)	grb_position_window_y1	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13232(R0x000033B0)	ffnr_alpha_beta	0000 0000 0000 0000 0ddd dddd 0ddd dddd	5419 (0x0000152B)
R13234(R0x000033B2)	ffnr_mix_thresh_y	0000 0000 0000 0000 0000 0000 dddd dddd	10 (0x0000000A)
R13236(R0x000033B4)	ffnr_mix_thresh_y_gain	0000 0000 0000 0000 0000 0000 dddd dddd	8 (0x00000008)
R13238(R0x000033B6)	ffnr_mix_thresh_gain	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)

Table 16: 1: SOC1 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13242(R0x000033BA)	apedge_control	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R13246(R0x000033BE)	ua_knee_l	0000 0000 0000 0000 0000 00dd dddd dddd	8 (0x00000008)
R13250(R0x000033C2)	ua_weights	0000 0000 0000 0000 dddd dddd 0000 0000	4352 (0x00001100)

Table 17: 2: SOC2 Registers

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13412(R0x00003464)	clip3_config	0000 0000 0000 0000 0000 0000 0000 00dd	2 (0x00000002)
R13414(R0x00003466)	clip3_min	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13416(R0x00003468)	clip3_max	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13418(R0x0000346A)	clip3_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13420(R0x0000346C)	clip3_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13422(R0x0000346E)	clip3_win_x_end	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R13424(R0x00003470)	clip3_win_y_end	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R13426(R0x00003472)	clip3_cnt_lo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13428(R0x00003474)	clip3_cnt_hi	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R13430(R0x00003476)	af_zone_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	16 (0x00000010)
R13432(R0x00003478)	af_zone_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13434(R0x0000347A)	af_zone_width	0000 0000 0000 0000 0000 dddd dddd dddd	160 (0x000000A0)
R13436(R0x0000347C)	af_zone_height	0000 0000 0000 0000 0000 0ddd dddd dddd	120 (0x00000078)
R13438(R0x0000347E)	af_zones_oob_ctrl	0000 0000 0000 0000 0000 0000 00d? ????	63 (0x0000003F)
R13440(R0x00003480)	clip1_config	0000 0000 0000 0000 0000 0000 0000 0ddd	2 (0x00000002)
R13442(R0x00003482)	clip2_config	0000 0000 0000 0000 0000 0000 0000 00dd	2 (0x00000002)
R13446(R0x00003486)	af_lum_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13448(R0x00003488)	af_lum_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13450(R0x0000348A)	af_lum_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13452(R0x0000348C)	af_lum_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13454(R0x0000348E)	af_lum_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13458(R0x00003492)	af_config_filters	0000 0000 0000 0000 000d dddd dddd dddd	4111 (0x0000100F)
R13460(R0x00003494)	af_config_thresholds_1	0000 0000 0000 0000 dddd dddd dddd dddd	256 (0x00000100)
R13466(R0x0000349A)	af_config_coeffs_0	0000 0000 0000 0000 0ddd dddd dddd dddd	12 (0x0000000C)
R13468(R0x0000349C)	af_config_coeffs_1	0000 0000 0000 0000 0ddd dddd dddd dddd	0 (0x00000000)
R13470(R0x0000349E)	af_config_coeffs_2	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13472(R0x000034A0)	af_config_coeffs_3	0000 0000 0000 0000 0ddd dddd dddd dddd	2 (0x00000002)
R13474(R0x000034A2)	af_config_coeffs_4	0000 0000 0000 0000 0ddd dddd dddd dddd	2048 (0x00000800)
R13476(R0x000034A4)	af_config_coeffs_5	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13478(R0x000034A6)	af_filter1_sharp_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13480(R0x000034A8)	af_filter1_sharp_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13482(R0x000034AA)	af_filter1_sharp_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13484(R0x000034AC)	af_filter1_sharp_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13486(R0x000034AE)	af_filter1_sharp_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13488(R0x000034B0)	af_filter1_oflow_cnt	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13506(R0x000034C2)	af_filter2_sharp_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13508(R0x000034C4)	af_filter2_sharp_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13510(R0x000034C6)	af_filter2_sharp_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13512(R0x000034C8)	af_filter2_sharp_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13514(R0x000034CA)	af_filter2_sharp_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13516(R0x000034CC)	af_filter2_oflow_cnt	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13518(R0x000034CE)	daf_config_0	0000 0000 0000 0000 0ddd dddd dddd dddd	31745 (0x00007C01)
R13520(R0x000034D0)	daf_config_1	0000 0000 0000 0000 0000 dddd 0000 00dd	0 (0x00000000)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13522(R0x000034D2)	daf_filter_oflow_cnt	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13524(R0x000034D4)	daf_filter_sharp_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13526(R0x000034D6)	daf_filter_sharp_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13528(R0x000034D8)	daf_filter_sharp_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13530(R0x000034DA)	daf_filter_sharp_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13532(R0x000034DC)	daf_filter_sharp_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13534(R0x000034DE)	daf_config_thresholds_1	0000 0000 0000 0000 dddd dddd dddd dddd	256 (0x00000100)
R13632(R0x00003540)	enable_tonal_curve	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)
R13634(R0x00003542)	tonal_x0	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13636(R0x00003544)	tonal_x1	0000 0000 0000 0000 0000 00dd dddd dddd	256 (0x00000100)
R13638(R0x00003546)	tonal_x2	0000 0000 0000 0000 0000 00dd dddd dddd	384 (0x00000180)
R13640(R0x00003548)	tonal_x3	0000 0000 0000 0000 0000 00dd dddd dddd	512 (0x00000200)
R13642(R0x0000354A)	tonal_x4	0000 0000 0000 0000 0000 00dd dddd dddd	640 (0x00000280)
R13644(R0x0000354C)	tonal_x5	0000 0000 0000 0000 0000 00dd dddd dddd	768 (0x00000300)
R13646(R0x0000354E)	tonal_x6	0000 0000 0000 0000 0000 00dd dddd dddd	896 (0x00000380)
R13648(R0x00003550)	tonal_y0	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13650(R0x00003552)	tonal_y1	0000 0000 0000 0000 0000 00dd dddd dddd	256 (0x00000100)
R13652(R0x00003554)	tonal_y2	0000 0000 0000 0000 0000 00dd dddd dddd	384 (0x00000180)
R13654(R0x00003556)	tonal_y3	0000 0000 0000 0000 0000 00dd dddd dddd	512 (0x00000200)
R13656(R0x00003558)	tonal_y4	0000 0000 0000 0000 0000 00dd dddd dddd	640 (0x00000280)
R13658(R0x0000355A)	tonal_y5	0000 0000 0000 0000 0000 00dd dddd dddd	768 (0x00000300)
R13660(R0x0000355C)	tonal_y6	0000 0000 0000 0000 0000 00dd dddd dddd	896 (0x00000380)
R13664(R0x00003560)	reciprocal_of_x0_minus_zero	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13666(R0x00003562)	reciprocal_of_x1_minus_x0	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13668(R0x00003564)	reciprocal_of_x2_minus_x1	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13670(R0x00003566)	reciprocal_of_x3_minus_x2	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13672(R0x00003568)	reciprocal_of_x4_minus_x3	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13674(R0x0000356A)	reciprocal_of_x5_minus_x4	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13676(R0x0000356C)	reciprocal_of_x6_minus_x5	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13678(R0x0000356E)	reciprocal_of_400_minus_x6	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13686(R0x00003576)	ae_zone_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	X
R13688(R0x00003578)	ae_zone_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	X
R13692(R0x0000357C)	ae_zone_width	0000 0000 0000 0000 0000 dddd dddd dddd	X
R13694(R0x0000357E)	ae_zone_height	0000 0000 0000 0000 0000 0ddd dddd dddd	X
R13696(R0x00003580)	ae_zones_oob_ctrl	0000 0000 0000 0000 0000 0000 0d?? ????	127 (0x0000007F)
R13698(R0x00003582)	ae_sum_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13700(R0x00003584)	ae_sum_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13702(R0x00003586)	ae_sum_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13704(R0x00003588)	ae_sum_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13706(R0x0000358A)	ae_sum_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13708(R0x0000358C)	ae_sum_5	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13710(R0x0000358E)	hist_pga_data_select	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)
R13712(R0x00003590)	hist1b_bin_offset	0000 0000 0000 0000 0000 dddd dddd dddd	10 (0x0000000A)
R13714(R0x00003592)	hist1a_bin_offset	0000 0000 0000 0000 0000 dddd dddd dddd	10 (0x0000000A)
R13716(R0x00003594)	hist0_pre_divider	0000 0000 0000 0000 dddd dddd dddd dddd	778 (0x0000030A)
R13718(R0x00003596)	hist1a_pre_divider	0000 0000 0000 0000 dddd dddd dddd dddd	778 (0x0000030A)
R13720(R0x00003598)	hist1b_pre_divider	0000 0000 0000 0000 dddd dddd dddd dddd	778 (0x0000030A)
R13722(R0x0000359A)	hist0_bin_config	0000 0000 0000 0000 0000 0000 ddd0 dddd	13 (0x0000000D)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13724(R0x0000359C)	hist1a_bin_config	0000 0000 0000 0000 0000 0000 000d dddd	10 (0x0000000A)
R13726(R0x0000359E)	hist1b_bin_config	0000 0000 0000 0000 0000 0000 000d dddd	8 (0x00000008)
R13728(R0x000035A0)	hist0_bin_offset	0000 0000 0000 0000 0000 dddd dddd dddd	10 (0x0000000A)
R13730(R0x000035A2)	dark_color_kill_controls	0000 0000 0000 0000 0000 0000 dddd dddd	20 (0x00000014)
R13732(R0x000035A4)	bright_color_kill_controls	0000 0000 0000 0000 0000 0ddd dddd dddd	1428 (0x00000594)
R13736(R0x000035A8)	clip2_win_y_end	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R13738(R0x000035AA)	clip1_min	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13740(R0x000035AC)	clip1_max	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13742(R0x000035AE)	clip2_min	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13744(R0x000035B0)	clip2_max	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13746(R0x000035B2)	clip2_win_x_end	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R13748(R0x000035B4)	hist0_bin_stats1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13750(R0x000035B6)	hist0_bin_stats2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13752(R0x000035B8)	hist0_bin_stats3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13754(R0x000035BA)	hist0_bin_stats4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13756(R0x000035BC)	hist1a_bin_stats1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13758(R0x000035BE)	hist1a_bin_stats2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13760(R0x000035C0)	hist1a_bin_stats3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13762(R0x000035C2)	hist1a_bin_stats4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13764(R0x000035C4)	hist1b_bin_stats1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13766(R0x000035C6)	hist1b_bin_stats2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13768(R0x000035C8)	hist1b_bin_stats3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13770(R0x000035CA)	hist1b_bin_stats4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13772(R0x000035CC)	clip2_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13774(R0x000035CE)	clip2_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13776(R0x000035D0)	hist_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13778(R0x000035D2)	hist_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13780(R0x000035D4)	hist_win_x_end	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R13782(R0x000035D6)	hist_win_y_end	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R13784(R0x000035D8)	clip1_cnt_lo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13786(R0x000035DA)	clip1_cnt_hi	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R13792(R0x000035E0)	clip2_cnt_lo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13794(R0x000035E2)	clip2_cnt_hi	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R13796(R0x000035E4)	clip1_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13798(R0x000035E6)	clip1_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13800(R0x000035E8)	clip1_win_x_end	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R13802(R0x000035EA)	clip1_win_y_end	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R13828(R0x00003604)	r_gamma_curve_knees_0_1	0000 0000 0000 0000 dddd dddd dddd dddd	6912 (0x00001B00)
R13830(R0x00003606)	r_gamma_curve_knees_2_3	0000 0000 0000 0000 dddd dddd dddd dddd	19502 (0x00004C2E)
R13832(R0x00003608)	r_gamma_curve_knees_4_5	0000 0000 0000 0000 dddd dddd dddd dddd	39032 (0x00009878)
R13834(R0x0000360A)	r_gamma_curve_knees_6_7	0000 0000 0000 0000 dddd dddd dddd dddd	49584 (0x0000C1B0)
R13836(R0x0000360C)	r_gamma_curve_knees_8_9	0000 0000 0000 0000 dddd dddd dddd dddd	55759 (0x0000D9CF)
R13838(R0x0000360E)	r_gamma_curve_knees_10_11	0000 0000 0000 0000 dddd dddd dddd dddd	59617 (0x0000E8E1)
R13840(R0x00003610)	r_gamma_curve_knees_12_13	0000 0000 0000 0000 dddd dddd dddd dddd	62190 (0x0000F2EE)
R13842(R0x00003612)	r_gamma_curve_knees_14_15	0000 0000 0000 0000 dddd dddd dddd dddd	63990 (0x0000F9F6)
R13844(R0x00003614)	r_gamma_curve_knees_16_17	0000 0000 0000 0000 dddd dddd dddd dddd	65019 (0x0000FDFB)
R13846(R0x00003616)	r_gamma_curve_knee_18	0000 0000 0000 0000 0000 0000 dddd dddd	255 (0x000000FF)
R13848(R0x00003618)	g_gamma_curve_knees_0_1	0000 0000 0000 0000 dddd dddd dddd dddd	6912 (0x00001B00)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13850(R0x0000361A)	g_gamma_curve_knees_2_3	0000 0000 0000 0000 dddd dddd dddd dddd	19502 (0x00004C2E)
R13852(R0x0000361C)	g_gamma_curve_knees_4_5	0000 0000 0000 0000 dddd dddd dddd dddd	39032 (0x00009878)
R13854(R0x0000361E)	g_gamma_curve_knees_6_7	0000 0000 0000 0000 dddd dddd dddd dddd	49584 (0x0000C1B0)
R13856(R0x00003620)	g_gamma_curve_knees_8_9	0000 0000 0000 0000 dddd dddd dddd dddd	55759 (0x0000D9CF)
R13858(R0x00003622)	g_gamma_curve_knees_10_11	0000 0000 0000 0000 dddd dddd dddd dddd	59617 (0x0000E8E1)
R13860(R0x00003624)	g_gamma_curve_knees_12_13	0000 0000 0000 0000 dddd dddd dddd dddd	62190 (0x0000F2EE)
R13862(R0x00003626)	g_gamma_curve_knees_14_15	0000 0000 0000 0000 dddd dddd dddd dddd	63990 (0x0000F9F6)
R13864(R0x00003628)	g_gamma_curve_knees_16_17	0000 0000 0000 0000 dddd dddd dddd dddd	65019 (0x0000FDFB)
R13866(R0x0000362A)	g_gamma_curve_knee_18	0000 0000 0000 0000 0000 0000 dddd dddd	255 (0x000000FF)
R13868(R0x0000362C)	b_gamma_curve_knees_0_1	0000 0000 0000 0000 dddd dddd dddd dddd	6912 (0x00001B00)
R13870(R0x0000362E)	b_gamma_curve_knees_2_3	0000 0000 0000 0000 dddd dddd dddd dddd	19502 (0x00004C2E)
R13872(R0x00003630)	b_gamma_curve_knees_4_5	0000 0000 0000 0000 dddd dddd dddd dddd	39032 (0x00009878)
R13874(R0x00003632)	b_gamma_curve_knees_6_7	0000 0000 0000 0000 dddd dddd dddd dddd	49584 (0x0000C1B0)
R13876(R0x00003634)	b_gamma_curve_knees_8_9	0000 0000 0000 0000 dddd dddd dddd dddd	55759 (0x0000D9CF)
R13878(R0x00003636)	b_gamma_curve_knees_10_11	0000 0000 0000 0000 dddd dddd dddd dddd	59617 (0x0000E8E1)
R13880(R0x00003638)	b_gamma_curve_knees_12_13	0000 0000 0000 0000 dddd dddd dddd dddd	62190 (0x0000F2EE)
R13882(R0x0000363A)	b_gamma_curve_knees_14_15	0000 0000 0000 0000 dddd dddd dddd dddd	63990 (0x0000F9F6)
R13884(R0x0000363C)	b_gamma_curve_knees_16_17	0000 0000 0000 0000 dddd dddd dddd dddd	65019 (0x0000FDFB)
R13886(R0x0000363E)	b_gamma_curve_knee_18	0000 0000 0000 0000 0000 0000 dddd dddd	255 (0x000000FF)
R13888(R0x00003640)	p_g1_p0q0	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)
R13890(R0x00003642)	p_g1_p0q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13892(R0x00003644)	p_g1_p0q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13894(R0x00003646)	p_g1_p0q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13896(R0x00003648)	p_g1_p0q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13898(R0x0000364A)	p_r_p0q0	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)
R13900(R0x0000364C)	p_r_p0q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13902(R0x0000364E)	p_r_p0q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13904(R0x00003650)	p_r_p0q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13906(R0x00003652)	p_r_p0q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13908(R0x00003654)	p_b_p0q0	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)
R13910(R0x00003656)	p_b_p0q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13912(R0x00003658)	p_b_p0q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13914(R0x0000365A)	p_b_p0q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13916(R0x0000365C)	p_b_p0q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13918(R0x0000365E)	p_g2_p0q0	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)
R13920(R0x00003660)	p_g2_p0q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13922(R0x00003662)	p_g2_p0q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13924(R0x00003664)	p_g2_p0q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13926(R0x00003666)	p_g2_p0q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13952(R0x00003680)	p_g1_p1q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13954(R0x00003682)	p_g1_p1q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13956(R0x00003684)	p_g1_p1q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13958(R0x00003686)	p_g1_p1q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13960(R0x00003688)	p_g1_p1q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13962(R0x0000368A)	p_r_p1q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13964(R0x0000368C)	p_r_p1q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13966(R0x0000368E)	p_r_p1q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13968(R0x00003690)	p_r_p1q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13970(R0x00003692)	p_r_p1q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13972(R0x00003694)	p_b_p1q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13974(R0x00003696)	p_b_p1q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13976(R0x00003698)	p_b_p1q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13978(R0x0000369A)	p_b_p1q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13980(R0x0000369C)	p_b_p1q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13982(R0x0000369E)	p_g2_p1q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13984(R0x000036A0)	p_g2_p1q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13986(R0x000036A2)	p_g2_p1q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13988(R0x000036A4)	p_g2_p1q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13990(R0x000036A6)	p_g2_p1q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14016(R0x000036C0)	p_g1_p2q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14018(R0x000036C2)	p_g1_p2q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14020(R0x000036C4)	p_g1_p2q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14022(R0x000036C6)	p_g1_p2q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14024(R0x000036C8)	p_g1_p2q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14026(R0x000036CA)	p_r_p2q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14028(R0x000036CC)	p_r_p2q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14030(R0x000036CE)	p_r_p2q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14032(R0x000036D0)	p_r_p2q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14034(R0x000036D2)	p_r_p2q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14036(R0x000036D4)	p_b_p2q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14038(R0x000036D6)	p_b_p2q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14040(R0x000036D8)	p_b_p2q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R14042(R0x000036DA)	p_b_p2q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14044(R0x000036DC)	p_b_p2q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14046(R0x000036DE)	p_g2_p2q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14048(R0x000036E0)	p_g2_p2q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14050(R0x000036E2)	p_g2_p2q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14052(R0x000036E4)	p_g2_p2q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14054(R0x000036E6)	p_g2_p2q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14080(R0x00003700)	p_g1_p3q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14082(R0x00003702)	p_g1_p3q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14084(R0x00003704)	p_g1_p3q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14086(R0x00003706)	p_g1_p3q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14088(R0x00003708)	p_g1_p3q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14090(R0x0000370A)	p_r_p3q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14092(R0x0000370C)	p_r_p3q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14094(R0x0000370E)	p_r_p3q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14096(R0x00003710)	p_r_p3q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14098(R0x00003712)	p_r_p3q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14100(R0x00003714)	p_b_p3q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14102(R0x00003716)	p_b_p3q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14104(R0x00003718)	p_b_p3q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14106(R0x0000371A)	p_b_p3q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14108(R0x0000371C)	p_b_p3q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14110(R0x0000371E)	p_g2_p3q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14112(R0x00003720)	p_g2_p3q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 17: 2: SOC2 Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R14114(R0x00003722)	p_g2_p3q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14116(R0x00003724)	p_g2_p3q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14118(R0x00003726)	p_g2_p3q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14144(R0x00003740)	p_g1_p4q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14146(R0x00003742)	p_g1_p4q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14148(R0x00003744)	p_g1_p4q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14150(R0x00003746)	p_g1_p4q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14152(R0x00003748)	p_g1_p4q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14154(R0x0000374A)	p_r_p4q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14156(R0x0000374C)	p_r_p4q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14158(R0x0000374E)	p_r_p4q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14160(R0x00003750)	p_r_p4q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14162(R0x00003752)	p_r_p4q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14164(R0x00003754)	p_b_p4q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14166(R0x00003756)	p_b_p4q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14168(R0x00003758)	p_b_p4q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14170(R0x0000375A)	p_b_p4q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14172(R0x0000375C)	p_b_p4q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14174(R0x0000375E)	p_g2_p4q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14176(R0x00003760)	p_g2_p4q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14178(R0x00003762)	p_g2_p4q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14180(R0x00003764)	p_g2_p4q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14182(R0x00003766)	p_g2_p4q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 18: SYSCTL Registers

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R0(R0x00000000)	chip_id	0000 0000 0000 0000 ???? ???? ???? ????	10368 (0x00002880)
R6(R0x00000006)	i2c_control	0000 0000 0000 0000 dd00 ddd0 0000 0000	0 (0x00000000)
R16(R0x00000010)	pll_dividers	0000 0000 0000 0000 00dd dddd dddd dddd	520 (0x00000208)
R18(R0x00000012)	pll_p_dividers	0000 0000 0000 0000 00dd dddd dddd dddd	176 (0x000000B0)
R20(R0x00000014)	pll_control	0000 0000 0000 0000 ?0dd dddd dddd dddd	9253 (0x00002425)
R22(R0x00000016)	clocks_control	0000 0000 0000 0000 d0dd dddd 0ddd dddd	0 (0x00000000)
R24(R0x00000018)	standby_control_and_status	0000 0000 0000 0000 ?d?0 0000 0ddd dddd	24585 (0x00006009)
R26(R0x0000001A)	reset_and_misc_control	0000 0000 0000 0000 0000 0ddd 00dd dddd	24 (0x00000018)
R28(R0x0000001C)	mcu_boot_mode	0000 0000 0000 0000 ???? ???? 00d0 0ddd	0 (0x00000000)
R30(R0x0000001E)	pad_slew_pad_config	0000 0000 0000 0000 d000 0ddd 0ddd 0ddd	1024 (0x00000400)
R32(R0x00000020)	vdd_dis_soft	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)
R34(R0x00000022)	vdd_dis_counter	0000 0000 0000 0000 dddd dddd dddd dddd	1080 (0x00000438)
R38(R0x00000026)	mcu_rom_initialize_sensor	0000 0000 0000 0000 0000 0000 0000 0ddd	7 (0x00000007)
R40(R0x00000028)	en_vdd_dis_soft	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)
R42(R0x0000002A)	pll_p4_p5_p6_dividers	0000 0000 0000 0000 0ddd dddd dddd dddd	30682 (0x000077DA)
R44(R0x0000002C)	pll_p7_divider	0000 0000 0000 0000 000d 0000 0000 dddd	4103 (0x00001007)
R46(R0x0000002E)	sensor_clock_divider	0000 0000 0000 0000 0000 0d0d 0000 dddd	0 (0x00000000)
R48(R0x00000030)	clk_otpm_clock_divider_selection	0000 0000 0000 0000 0000 0000 0000 00dd	3 (0x00000003)
R50(R0x00000032)	s_clk_pad_slew_rate	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)
R52(R0x00000034)	reg_0034	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R54(R0x00000036)	reg_0036	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R64(R0x00000040)	user_defined_i2c_device_address_id	0000 0000 0000 0000 dddd ddd0 dddd ddd0	31352 (0x00007A78)
R66(R0x00000042)	burnin_passcode	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R68(R0x00000044)	gen_purp_0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 18: SYSCTL Registers (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R70(R0x00000046)	gen_purp_1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R72(R0x00000048)	gen_purp_2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R74(R0x0000004A)	gen_purp_3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R80(R0x00000050)	release_version	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)

Table 19: RX_SS Registers

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R256(R0x00000100)	test_pattern_control	0000 0000 0000 0000 dd?d dddd dddd dddd	8192 (0x00002000)
R258(R0x00000102)	test_pxl_red	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R260(R0x00000104)	test_pxl_green1	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R262(R0x00000106)	test_pxl_green2	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R264(R0x00000108)	test_pxl_blue	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R266(R0x0000010A)	row_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R268(R0x0000010C)	col_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R270(R0x0000010E)	row_end	0000 0000 0000 0000 0000 dddd dddd dddd	1543 (0x00000607)
R272(R0x00000110)	col_end	0000 0000 0000 0000 0000 dddd dddd dddd	2055 (0x00000807)
R274(R0x00000112)	rx_fifo_control	0000 0000 0000 0000 ??dd ??dd dddd dddd	20 (0x00000014)
R276(R0x00000114)	hblank_llead	0000 0000 0000 0000 000d dddd dddd dddd	100 (0x00000064)
R278(R0x00000116)	hblank_ltrail	0000 0000 0000 0000 000d dddd dddd dddd	10 (0x0000000A)
R280(R0x00000118)	lower_seed_value	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R282(R0x0000011A)	upper_seed_value	0000 0000 0000 0000 0000 0000 0000 dddd	0 (0x00000000)
R296(R0x00000128)	mipi_receiver_control	0000 0000 0000 0000 dddd d0dd dddd dd0d	4 (0x00000004)
R300(R0x0000012C)	mipi_and_ccp2_pixel_data_out_status_0	0000 0000 0000 0000 000? ???? ???? ????	2048 (0x00000800)
R302(R0x0000012E)	mipi_and_ccp2_pixel_data_out_status_1_	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R304(R0x00000130)	mipi_error_reporting_	0000 0000 0000 0000 0000 0??? ???? ????	0 (0x00000000)

Table 19: RX_SS Registers

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R306(R0x00000132)	packet_level_errors_0_	0000 0000 0000 0000 0000 0??? ???? ????	0 (0x00000000)
R308(R0x00000134)	packet_level_errors_1_and_protocol_layer_errors_	0000 0000 0000 0000 0000 00?? ???? ????	0 (0x00000000)
R310(R0x00000136)	data_visibility_0_	0000 0000 0000 0000 0000 0000 00?? ????	0 (0x00000000)
R312(R0x00000138)	data_visibility_1_	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R314(R0x0000013A)	data_visibility_2_	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R316(R0x0000013C)	phy_configuration_control	0000 0000 0000 0000 00dd dddd dddd dddd	6175 (0x0000181F)
R318(R0x0000013E)	testp_colorbar_width	0000 0000 0000 0000 0000 dddd dddd dddd	255 (0x000000FF)
R320(R0x00000140)	vblank_llead	0000 0000 0000 0000 0000 000d dddd dddd	16 (0x00000010)
R322(R0x00000142)	vblank_ltrail	0000 0000 0000 0000 0000 000d dddd dddd	6 (0x00000006)
R336(R0x00000150)	crc_lane_1_data_rx_ss	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R338(R0x00000152)	crc_lane_2_data_rx_ss	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R344(R0x00000158)	testp_vblank	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R346(R0x0000015A)	eofv_del	0000 0000 0000 0000 0000 0ddd dddd dddd	1151 (0x0000047F)

Table 20: XDMA Registers

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R2434(R0x00000982)	access_ctl_stat	0000 0000 0000 0000 0000 0000 dd0? ???d	0 (0x00000000)
R2442(R0x0000098A)	physical_address_access	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2446(R0x0000098E)	logical_address_access	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2448(R0x00000990)	mcu_variable_data0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2450(R0x00000992)	mcu_variable_data1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2452(R0x00000994)	mcu_variable_data2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2454(R0x00000996)	mcu_variable_data3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2456(R0x00000998)	mcu_variable_data4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2458(R0x0000099A)	mcu_variable_data5	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2460(R0x0000099C)	mcu_variable_data6	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2462(R0x0000099E)	mcu_variable_data7	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2560(R0x00000A00)	jtag_tms	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2562(R0x00000A02)	jtag_tdi	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2564(R0x00000A04)	jtag_ctl	0000 0000 0000 0000 0000 0000 0ddd dddd	0 (0x00000000)
R2566(R0x00000A06)	jtag_tdo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)

Table 21: TX_SS

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13312(R0x00003400)	mipi_control	0000 0000 0000 0000 dddd dddd dddd ?ddd	30766 (0x0000782E)
R13314(R0x00003402)	mipi_status	0000 0000 0000 0000 000d d??0 00?? 000?	17 (0x00000011)
R13316(R0x00003404)	custom_short_pkt	0000 0000 0000 0000 00?d dddd dd00 0000	0 (0x00000000)
R13318(R0x00003406)	txc_mipi_data_format	0000 0000 0000 0000 00dd dddd 00dd dddd	12336 (0x00003030)
R13320(R0x00003408)	txc_mipi_line_byte_cnt	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13322(R0x0000340A)	txss_mipi_control_addl	0000 0000 0000 0000 0000 0000 000d dddd	10 (0x0000000A)
R13324(R0x0000340C)	custom_short_pkt_wc	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 21: TX_SS (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13326(R0x0000340E)	mipi_test	0000 0000 0000 0000 000d 00?d 0ddd 0ddd	0 (0x00000000)
R13328(R0x00003410)	mipi_timing_t_hs_zero	0000 0000 0000 0000 0000 dddd 0000 0000	2048 (0x00000800)
R13330(R0x00003412)	mipi_timing_t_hs_exit_hs_trail	0000 0000 0000 0000 00dd dddd 0000 dddd	1797 (0x00000705)
R13332(R0x00003414)	mipi_timing_t_clk_post_clk_pre	0000 0000 0000 0000 00dd dddd 00dd dddd	2817 (0x00000B01)
R13334(R0x00003416)	mipi_t_clk_trail_clk_zero	0000 0000 0000 0000 0000 dddd 00dd dddd	1289 (0x00000509)
R13336(R0x00003418)	mipi_timing_t_lpx	0000 0000 0000 0000 0000 0000 00dd dddd	4 (0x00000004)
R13338(R0x0000341A)	mipi_init_timing	0000 0000 0000 0000 0ddd dddd 0ddd dddd	1544 (0x00000608)
R13340(R0x0000341C)	mipi_ccp__sel_class	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R13342(R0x0000341E)	mipi_test_checksum	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15360(R0x00003C00)	jpssmode	0000 0000 0000 0000 0??? 0??? 0??? 0ddd	13107 (0x00003333)
R15362(R0x00003C02)	jpss_ctrl	0000 0000 0000 0000 0ddd dddd dddd dddd	1552 (0x00000610)
R15364(R0x00003C04)	jpss_inwid	0000 0000 0000 0000 0000 dddd dddd dddd	2592 (0x00000A20)
R15366(R0x00003C06)	jpss_inhgt	0000 0000 0000 0000 0000 0ddd dddd dddd	1944 (0x00000798)
R15368(R0x00003C08)	jpss_spoof_width	0000 0000 0000 0000 0000 dddd dddd dddd	1280 (0x00000500)
R15370(R0x00003C0A)	jpss_spoof_height	0000 0000 0000 0000 0000 dddd dddd dddd	480 (0x000001E0)
R15372(R0x00003C0C)	jpss_jpeg_sosi_code	0000 0000 0000 0000 dddd dddd dddd dddd	65468 (0x0000FFBC)
R15374(R0x00003C0E)	jpss_jpeg_eosi_code	0000 0000 0000 0000 dddd dddd dddd dddd	65469 (0x0000FFBD)
R15376(R0x00003C10)	jpss_dummy_pattern	0000 0000 0000 0000 dddd dddd dddd dddd	65535 (0x0000FFFF)
R15378(R0x00003C12)	jpss_trigger_mark	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R15380(R0x00003C14)	jpss_exif_tn_ptr	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15382(R0x00003C16)	jpss_status	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15384(R0x00003C18)	jpss_frame_length0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15386(R0x00003C1A)	jpss_frame_length1	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R15388(R0x00003C1C)	jpss_rcvd_frame_cnt	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 21: TX_SS (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15390(R0x00003C1E)	jpss_rcvd_frame_line_cnt	0000 0000 0000 0000 0000 ???? ???? ????	0 (0x00000000)
R15394(R0x00003C22)	jpss_tx_frame_cnt	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15396(R0x00003C24)	crc_control	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R15398(R0x00003C26)	crc_data	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15400(R0x00003C28)	jpss_between_frame_status	0000 0000 0000 0000 0000 ???? ???? ????	4095 (0x00000FFF)
R15424(R0x00003C40)	jpeg_ctrl	0000 0000 0000 0000 0000 000d dddd dddd	14 (0x0000000E)
R15426(R0x00003C42)	jpeg_qtable_sel	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R15428(R0x00003C44)	jpeg_restart	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15430(R0x00003C46)	jpeg_jpop_limit	0000 0000 0000 0000 dddd dddd dddd dddd	65535 (0x0000FFFF)
R15432(R0x00003C48)	jpeg_exif_bytes	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R15434(R0x00003C4A)	jpeg_hdr_pad_bytes	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R15454(R0x00003C5E)	jpss_debug_sel	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R15456(R0x00003C60)	jpeg_hw_status	0000 0000 0000 0000 0000 000d ???? ????	0 (0x00000000)
R15488(R0x00003C80)	tn_ctrl	0000 0000 0000 0000 000d dddd dd0d dddd	533 (0x00000215)
R15490(R0x00003C82)	tn_scalar_x_ratio	0000 0000 0000 0000 0000 dddd dddd dddd	506 (0x000001FA)
R15492(R0x00003C84)	tn_scalar_y_ratio	0000 0000 0000 0000 0000 dddd dddd dddd	506 (0x000001FA)
R15494(R0x00003C86)	tn_crop_wid	0000 0000 0000 0000 0000 00dd dddd dddd	640 (0x00000280)
R15496(R0x00003C88)	tn_crop_hgt	0000 0000 0000 0000 0000 000d dddd dddd	480 (0x000001E0)
R15498(R0x00003C8A)	tn_crop_tl	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15500(R0x00003C8C)	tn_start_code	0000 0000 0000 0000 dddd dddd dddd dddd	65470 (0x0000FFBE)
R15502(R0x00003C8E)	tn_end_code	0000 0000 0000 0000 dddd dddd dddd dddd	65471 (0x0000FFBF)
R15504(R0x00003C90)	tn_scale_sharp_ctrl	0000 0000 0000 0000 00dd dddd dddd dddd	4616 (0x00001208)
R15520(R0x00003CA0)	txss_parameters	0000 0000 0000 0000 0000 0000 0ddd d00d	64 (0x00000040)
R15522(R0x00003CA2)	txc_parameters	0000 0000 0000 0000 0000 0000 dddd dddd	7 (0x00000007)

Table 21: TX_SS (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15524(R0x00003CA4)	txc_po_pclk1_config	0000 0000 0000 0000 0000 0000 ddd0 dddd	1 (0x00000001)
R15526(R0x00003CA6)	txc_po_pclk2_config	0000 0000 0000 0000 0000 0000 ddd0 dddd	1 (0x00000001)
R15528(R0x00003CA8)	txc_po_pclk3_config	0000 0000 0000 0000 0000 0000 ddd0 dddd	1 (0x00000001)
R15530(R0x00003CAA)	txc_timing	0000 0000 0000 0000 dddd dddd dddd dddd	1285 (0x00000505)
R15536(R0x00003CB0)	txc_line_ccir_sof01_code	0000 0000 0000 0000 dddd dddd dddd dddd	65280 (0x0000FF00)
R15538(R0x00003CB2)	txc_line_ccir_sof23_code	0000 0000 0000 0000 dddd dddd dddd dddd	171 (0x000000AB)
R15540(R0x00003CB4)	txc_line_ccir_eof01_code	0000 0000 0000 0000 dddd dddd dddd dddd	65280 (0x0000FF00)
R15542(R0x00003CB6)	txc_line_ccir_eof23_code	0000 0000 0000 0000 dddd dddd dddd dddd	182 (0x000000B6)
R15544(R0x00003CB8)	txc_line_ccir_sol01_code	0000 0000 0000 0000 dddd dddd dddd dddd	65280 (0x0000FF00)
R15546(R0x00003CBA)	txc_line_ccir_sol23_code	0000 0000 0000 0000 dddd dddd dddd dddd	128 (0x00000080)
R15548(R0x00003CBC)	txc_line_ccir_eol01_code	0000 0000 0000 0000 dddd dddd dddd dddd	65280 (0x0000FF00)
R15550(R0x00003CBE)	txc_line_ccir_eol23_code	0000 0000 0000 0000 dddd dddd dddd dddd	157 (0x0000009D)
R15552(R0x00003CC0)	txss_status	0000 0000 0000 0000 0000 0000 0000 0?0?	0 (0x00000000)
R15554(R0x00003CC2)	txss_clr_status	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R15584(R0x00003CE0)	indirect_ram_access_ctrl	0000 0000 0000 0000 0000 0000 0000 0?dd	1 (0x00000001)
R15586(R0x00003CE2)	indirect_address	0000 0000 0000 0000 dddd dddd dddd dddd	65535 (0x0000FFFF)
R15588(R0x00003CE4)	indirect_data	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 22: OTPM

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R14336(R0x00003800)	otpm_status	0000 0000 0000 0000 0000 000? ???? ????	0 (0x00000000)
R14338(R0x00003802)	otpm_control	0000 0000 0000 0000 0000 0000 0000 dddd	4 (0x00000004)
R14340(R0x00003804)	otpm_addr	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R14342(R0x00003806)	otpm_data_pgm_l	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14344(R0x00003808)	otpm_data_pgm_h	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14346(R0x0000380A)	otpm_data_pgm_extra	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R14348(R0x0000380C)	otpm_data_read_l	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R14350(R0x0000380E)	otpm_data_read_h	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R14352(R0x00003810)	otpm_data_read_extra	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R14354(R0x00003812)	otpm_cfg	0000 0000 0000 0000 0ddd dddd dddd dddd	8540 (0x0000215C)
R14356(R0x00003814)	otpm_tcfg_01	0000 0000 0000 0000 dddd dddd dddd dddd	1795 (0x00000703)
R14358(R0x00003816)	otpm_tcfg_23	0000 0000 0000 0000 dddd dddd dddd dddd	1799 (0x00000707)
R14360(R0x00003818)	otpm_tcfg_4b	0000 0000 0000 0000 dddd dddd dddd dddd	17415 (0x00004407)
R14362(R0x0000381A)	otpm_expr	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)
R14364(R0x0000381C)	otpm_cfg2	0000 0000 0000 0000 0000 00dd 00dd dddd	0 (0x00000000)

Sorted by Name

Table 23: 0: Sensor Core Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12334(R0x0000302E)	analogue_gain_code_blue_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)
R12328(R0x00003028)	analogue_gain_code_global_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)
R12336(R0x00003030)	analogue_gain_code_greenb_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)
R12330(R0x0000302A)	analogue_gain_code_greenr_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)
R12332(R0x0000302C)	analogue_gain_code_red_	0000 0000 0000 0000 0000 0000 0ddd dddd	13 (0x0000000D)
R12376(R0x00003058)	blue_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)

Table 24: GPIO_SS Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1792(R0x00000700)	ch0_cfg	0000 0000 0000 0000 000d dddd dddd 0ddd	0 (0x00000000)
R1824(R0x00000720)	ch1_cfg	0000 0000 0000 0000 000d dddd dddd 0ddd	0 (0x00000000)
R1856(R0x00000740)	ch2_cfg	0000 0000 0000 0000 000d dddd dddd 0ddd	0 (0x00000000)
R1888(R0x00000760)	ch3_cfg	0000 0000 0000 0000 000d dddd dddd 0ddd	0 (0x00000000)
R1920(R0x00000780)	ch4_cfg	0000 0000 0000 0000 000d dddd dddd 0ddd	0 (0x00000000)
R1952(R0x000007A0)	ch5_cfg	0000 0000 0000 0000 000d dddd dddd 0ddd	0 (0x00000000)
R1984(R0x000007C0)	ch6_cfg	0000 0000 0000 0000 000d dddd dddd 0ddd	0 (0x00000000)
R2016(R0x000007E0)	ch7_cfg	0000 0000 0000 0000 000d dddd dddd 0ddd	0 (0x00000000)

Table 25: 0: Sensor Core Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12306(R0x00003012)	coarse_integration_time_	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)
R12318(R0x0000301E)	data_pedestal_	0000 0000 0000 0000 0000 dddd dddd dddd	168 (0x000000A8)
R12342(R0x00003036)	digital_gain_blue_	0000 0000 0000 0000 0000 0ddd 0000 0000	256 (0x00000100)
R12344(R0x00003038)	digital_gain_greenb_	0000 0000 0000 0000 0000 0ddd 0000 0000	256 (0x00000100)
R12338(R0x00003032)	digital_gain_greenr_	0000 0000 0000 0000 0000 0ddd 0000 0000	256 (0x00000100)
R12340(R0x00003034)	digital_gain_red_	0000 0000 0000 0000 0000 0ddd 0000 0000	256 (0x00000100)
R12438(R0x00003096)	ease_btm_calib	0000 0000 0000 0000 0000 000d dddd dddd	5 (0x00000005)
R12434(R0x00003092)	ease_btm_gain	0000 0000 0000 0000 0000 000d dddd dddd	33 (0x00000021)
R12446(R0x0000309E)	ease_btm_ref_mux	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12442(R0x0000309A)	ease_btm_samp_mux	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12430(R0x0000308E)	ease_control	0000 0000 0000 0000 dddd dddd dddd dddd	52224 (0x0000CC00)
R12436(R0x00003094)	ease_top_calib	0000 0000 0000 0000 0000 000d dddd dddd	3 (0x00000003)
R12432(R0x00003090)	ease_top_gain	0000 0000 0000 0000 0000 000d dddd dddd	32 (0x00000020)
R12444(R0x0000309C)	ease_top_ref_mux	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12440(R0x00003098)	ease_top_samp_mux	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12312(R0x00003018)	extra_delay	0000 0000 0000 0000 dddd dddd dddd ddd0	0 (0x00000000)
R12304(R0x00003010)	fine_correction	0000 0000 0000 0000 0ddd dddd dddd dddd	156 (0x0000009C)
R12308(R0x00003014)	fine_integration_time_	0000 0000 0000 0000 dddd dddd dddd ddd0	842 (0x0000034A)
R12358(R0x00003046)	flash	0000 0000 0000 0000 dddd dddd dddd 0000	1536 (0x00000600)
R12360(R0x00003048)	flash_count	0000 0000 0000 0000 dddd dddd dddd dddd	8 (0x00000008)
R12347(R0x0000303B)	frame_count_	0000 0000 0000 0000 0000 0000 ???? ????	255 (0x000000FF)
R12298(R0x0000300A)	frame_length_lines_	0000 0000 0000 0000 dddd dddd dddd dddd	2031 (0x000007EF)
R12348(R0x0000303C)	frame_status	0000 0000 0000 0000 0000 0000 0000 00??	0 (0x00000000)
R12636(R0x0000315C)	global_bulb_trigger_count	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 25: 0: Sensor Core Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12634(R0x0000315A)	global_flash_start	0000 0000 0000 0000 dddd dddd dddd dddd	154 (0x0000009A)
R12382(R0x0000305E)	global_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)
R12646(R0x00003166)	global_read_start	0000 0000 0000 0000 dddd dddd dddd dddd	160 (0x000000A0)
R12648(R0x00003168)	global_read_start2	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R12640(R0x00003160)	global_rst_end	0000 0000 0000 0000 dddd dddd dddd dddd	152 (0x00000098)
R12638(R0x0000315E)	global_seq_trigger	0000 0000 0000 0000 dddd 0ddd dddd 0ddd	0 (0x00000000)
R12642(R0x00003162)	global_shutter_start	0000 0000 0000 0000 dddd dddd dddd dddd	159 (0x0000009F)
R12644(R0x00003164)	global_shutter_start2	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R12326(R0x00003026)	gpi_status	0000 0000 0000 0000 dddd dddd dddd dddd	65535 (0x0000FFFF)
R12374(R0x00003056)	green1_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)
R12380(R0x0000305C)	green2_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)
R12322(R0x00003022)	grouped_parameter_hold_	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R12776(R0x000031E8)	horizontal_cursor_position_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12780(R0x000031EC)	horizontal_cursor_width_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12796(R0x000031FC)	i2c_ids	0000 0000 0000 0000 dddd dddd dddd dddd	12320 (0x00003020)
R12786(R0x000031F2)	i2c_ids_mipi_default	0000 0000 0000 0000 dddd dddd dddd dddd	28268 (0x00006E6C)

Table 26: GPIO_SS Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15234(R0x00003B82)	i2c_master_control	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)
R15236(R0x00003B84)	i2c_master_frequency_divider	0000 0000 0000 0000 0000 00dd dddd dddd	99 (0x00000063)
R15232(R0x00003B80)	i2c_master_status	0000 0000 0000 0000 0000 0000 000? ????	9 (0x00000009)

Table 27: 0: Sensor Core Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12317(R0x0000301D)	image_orientation_	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R12300(R0x0000300C)	line_length_pck_	0000 0000 0000 0000 dddd dddd dddd ddd0	3878 (0x00000F26)
R12323(R0x00003023)	mask_corrupted_frames_	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R12288(R0x00003000)	model_id_	0000 0000 0000 0000 dddd dddd dddd dddd	10368 (0x00002880)
R12316(R0x0000301C)	mode_select_	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R12324(R0x00003024)	pixel_order_	0000 0000 0000 0000 0000 0000 0000 00??	0 (0x00000000)
R12352(R0x00003040)	read_mode	0000 0000 0000 0000 dddd dddd dddd dddd	65 (0x00000041)
R12378(R0x0000305A)	red_gain	0000 0000 0000 0000 0ddd 000d dddd dddd	4148 (0x00001034)
R12314(R0x0000301A)	reset_register	0000 0000 0000 0000 dd0d 0ddd dddd dddd	4312 (0x000010D8)
R12310(R0x00003016)	row_speed	0000 0000 0000 0000 0000 0ddd 0ddd 0ddd	273 (0x00000111)

Table 28: GPIO_SS Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15168(R0x00003B40)	rxbuffer_data_register_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15170(R0x00003B42)	rxbuffer_data_register_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15188(R0x00003B54)	rxbuffer_data_register_10	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15190(R0x00003B56)	rxbuffer_data_register_11	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15192(R0x00003B58)	rxbuffer_data_register_12	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15194(R0x00003B5A)	rxbuffer_data_register_13	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15196(R0x00003B5C)	rxbuffer_data_register_14	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15198(R0x00003B5E)	rxbuffer_data_register_15	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15200(R0x00003B60)	rxbuffer_data_register_16	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15202(R0x00003B62)	rxbuffer_data_register_17	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15204(R0x00003B64)	rxbuffer_data_register_18	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15206(R0x00003B66)	rxbuffer_data_register_19	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)

Table 28: GPIO_SS Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15172(R0x00003B44)	rxbuffer_data_register_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15208(R0x00003B68)	rxbuffer_data_register_20	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15210(R0x00003B6A)	rxbuffer_data_register_21	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15212(R0x00003B6C)	rxbuffer_data_register_22	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15214(R0x00003B6E)	rxbuffer_data_register_23	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15216(R0x00003B70)	rxbuffer_data_register_24	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15218(R0x00003B72)	rxbuffer_data_register_25	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15220(R0x00003B74)	rxbuffer_data_register_26	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15222(R0x00003B76)	rxbuffer_data_register_27	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15224(R0x00003B78)	rxbuffer_data_register_28	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15226(R0x00003B7A)	rxbuffer_data_register_29	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15174(R0x00003B46)	rxbuffer_data_register_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15228(R0x00003B7C)	rxbuffer_data_register_30	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15230(R0x00003B7E)	rxbuffer_data_register_31	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15176(R0x00003B48)	rxbuffer_data_register_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15178(R0x00003B4A)	rxbuffer_data_register_5	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15180(R0x00003B4C)	rxbuffer_data_register_6	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15182(R0x00003B4E)	rxbuffer_data_register_7	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15184(R0x00003B50)	rxbuffer_data_register_8	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15186(R0x00003B52)	rxbuffer_data_register_9	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R1556(R0x00000614)	second_scl_sda_pd	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)
R1536(R0x00000600)	second_snsr	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R1538(R0x00000602)	second_snsr_oe	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)

Table 29: 0: Sensor Core Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12346(R0x0000303A)	smia_version_	0000 0000 0000 0000 0000 0000 ???? ????	10 (0x0000000A)
R12321(R0x00003021)	software_reset_	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R12406(R0x00003076)	test_data_blue_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12408(R0x00003078)	test_data_greenb_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12404(R0x00003074)	test_data_greenr_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12402(R0x00003072)	test_data_red_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12400(R0x00003070)	test_pattern_mode_	0000 0000 0000 0000 0000 000d 0000 0ddd	0 (0x00000000)
R12410(R0x0000307A)	test_raw_mode	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)

Table 30: GPIO_SS Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15104(R0x00003B00)	txbuffer_data_register_0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15106(R0x00003B02)	txbuffer_data_register_1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15124(R0x00003B14)	txbuffer_data_register_10	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15126(R0x00003B16)	txbuffer_data_register_11	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15128(R0x00003B18)	txbuffer_data_register_12	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15130(R0x00003B1A)	txbuffer_data_register_13	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15132(R0x00003B1C)	txbuffer_data_register_14	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15134(R0x00003B1E)	txbuffer_data_register_15	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15136(R0x00003B20)	txbuffer_data_register_16	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15138(R0x00003B22)	txbuffer_data_register_17	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15140(R0x00003B24)	txbuffer_data_register_18	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15142(R0x00003B26)	txbuffer_data_register_19	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15108(R0x00003B04)	txbuffer_data_register_2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15144(R0x00003B28)	txbuffer_data_register_20	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15146(R0x00003B2A)	txbuffer_data_register_21	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15148(R0x00003B2C)	txbuffer_data_register_22	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15150(R0x00003B2E)	txbuffer_data_register_23	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15152(R0x00003B30)	txbuffer_data_register_24	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15154(R0x00003B32)	txbuffer_data_register_25	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15156(R0x00003B34)	txbuffer_data_register_26	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15158(R0x00003B36)	txbuffer_data_register_27	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15160(R0x00003B38)	txbuffer_data_register_28	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15162(R0x00003B3A)	txbuffer_data_register_29	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15110(R0x00003B06)	txbuffer_data_register_3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 30: GPIO_SS Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15164(R0x00003B3C)	txbuffer_data_register_30	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15166(R0x00003B3E)	txbuffer_data_register_31	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15112(R0x00003B08)	txbuffer_data_register_4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15114(R0x00003B0A)	txbuffer_data_register_5	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15116(R0x00003B0C)	txbuffer_data_register_6	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15118(R0x00003B0E)	txbuffer_data_register_7	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15120(R0x00003B10)	txbuffer_data_register_8	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15122(R0x00003B12)	txbuffer_data_register_9	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15238(R0x00003B86)	txbuffer_total_byte_count	0000 0000 0000 0000 0000 0000 00dd dddd	0 (0x00000000)
R1568(R0x00000620)	vcm_control	0000 0000 0000 0000 d0dd dddd dd00 dddd	0 (0x00000000)

Table 31: 0: Sensor Core Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12778(R0x000031EA)	vertical_cursor_position_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12782(R0x000031EE)	vertical_cursor_width_	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)

Table 32: GPIO_SS Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1554(R0x00000612)	vgpio_data_alt_sel	0000 0000 0000 0000 d0dd 0000 dddd dddd	0 (0x00000000)
R1548(R0x0000060C)	vgpio_data_from_icb	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1558(R0x00000616)	vgpio_data_rx_tx_sel	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R1552(R0x00000610)	vgpio_data_status	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R1550(R0x0000060E)	vgpio_dir	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1544(R0x00000608)	vgpio_int_ctl	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 32: GPIO_SS Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1546(R0x0000060A)	vgpio_int_status_clr	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1798(R0x00000706)	wg0_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1800(R0x00000708)	wg0_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1802(R0x0000070A)	wg0_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1804(R0x0000070C)	wg0_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1806(R0x0000070E)	wg0_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1796(R0x00000704)	wg0_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1820(R0x0000071C)	wg0_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1818(R0x0000071A)	wg0_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1794(R0x00000702)	wg0_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1808(R0x00000710)	wg0_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1810(R0x00000712)	wg0_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1812(R0x00000714)	wg0_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1814(R0x00000716)	wg0_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1816(R0x00000718)	wg0_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1822(R0x0000071E)	wg0_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1830(R0x00000726)	wg1_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1832(R0x00000728)	wg1_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1834(R0x0000072A)	wg1_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1836(R0x0000072C)	wg1_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1838(R0x0000072E)	wg1_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1828(R0x00000724)	wg1_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1852(R0x0000073C)	wg1_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1850(R0x0000073A)	wg1_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)

Table 32: GPIO_SS Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1826(R0x00000722)	wg1_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1840(R0x00000730)	wg1_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1842(R0x00000732)	wg1_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1844(R0x00000734)	wg1_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1846(R0x00000736)	wg1_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1848(R0x00000738)	wg1_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1854(R0x0000073E)	wg1_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1862(R0x00000746)	wg2_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1864(R0x00000748)	wg2_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1866(R0x0000074A)	wg2_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1868(R0x0000074C)	wg2_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1870(R0x0000074E)	wg2_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1860(R0x00000744)	wg2_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1884(R0x0000075C)	wg2_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1882(R0x0000075A)	wg2_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1858(R0x00000742)	wg2_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1872(R0x00000750)	wg2_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1874(R0x00000752)	wg2_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1876(R0x00000754)	wg2_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1878(R0x00000756)	wg2_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1880(R0x00000758)	wg2_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1886(R0x0000075E)	wg2_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1894(R0x00000766)	wg3_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1896(R0x00000768)	wg3_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)

Table 32: GPIO_SS Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1898(R0x0000076A)	wg3_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1900(R0x0000076C)	wg3_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1902(R0x0000076E)	wg3_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1892(R0x00000764)	wg3_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1916(R0x0000077C)	wg3_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1914(R0x0000077A)	wg3_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1890(R0x00000762)	wg3_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1904(R0x00000770)	wg3_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1906(R0x00000772)	wg3_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1908(R0x00000774)	wg3_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1910(R0x00000776)	wg3_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1912(R0x00000778)	wg3_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1918(R0x0000077E)	wg3_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1926(R0x00000786)	wg4_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1928(R0x00000788)	wg4_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1930(R0x0000078A)	wg4_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1932(R0x0000078C)	wg4_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1934(R0x0000078E)	wg4_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1924(R0x00000784)	wg4_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1948(R0x0000079C)	wg4_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1946(R0x0000079A)	wg4_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1922(R0x00000782)	wg4_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1936(R0x00000790)	wg4_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1938(R0x00000792)	wg4_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 32: GPIO_SS Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1940(R0x00000794)	wg4_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1942(R0x00000796)	wg4_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1944(R0x00000798)	wg4_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1950(R0x0000079E)	wg4_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1958(R0x000007A6)	wg5_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1960(R0x000007A8)	wg5_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1962(R0x000007AA)	wg5_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1964(R0x000007AC)	wg5_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1966(R0x000007AE)	wg5_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1956(R0x000007A4)	wg5_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R1980(R0x000007BC)	wg5_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1978(R0x000007BA)	wg5_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1954(R0x000007A2)	wg5_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1968(R0x000007B0)	wg5_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1970(R0x000007B2)	wg5_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1972(R0x000007B4)	wg5_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1974(R0x000007B6)	wg5_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1976(R0x000007B8)	wg5_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R1982(R0x000007BE)	wg5_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1990(R0x000007C6)	wg6_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1992(R0x000007C8)	wg6_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1994(R0x000007CA)	wg6_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1996(R0x000007CC)	wg6_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R1998(R0x000007CE)	wg6_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)

Table 32: GPIO_SS Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R1988(R0x000007C4)	wg6_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R2012(R0x000007DC)	wg6_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2010(R0x000007DA)	wg6_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R1986(R0x000007C2)	wg6_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R2000(R0x000007D0)	wg6_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2002(R0x000007D2)	wg6_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2004(R0x000007D4)	wg6_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2006(R0x000007D6)	wg6_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2008(R0x000007D8)	wg6_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2014(R0x000007DE)	wg6_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R2022(R0x000007E6)	wg7_cfg0	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2024(R0x000007E8)	wg7_cfg1	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2026(R0x000007EA)	wg7_cfg2	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2028(R0x000007EC)	wg7_cfg3	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2030(R0x000007EE)	wg7_cfg4	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R2020(R0x000007E4)	wg7_cfg_idle	0000 0000 0000 0000 0000 000d d000 0ddd	0 (0x00000000)
R2044(R0x000007FC)	wg7_n	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2042(R0x000007FA)	wg7_prescale	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R2018(R0x000007E2)	wg7_start	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R2032(R0x000007F0)	wg7_t0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2034(R0x000007F2)	wg7_t1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2036(R0x000007F4)	wg7_t2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2038(R0x000007F6)	wg7_t3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2040(R0x000007F8)	wg7_t4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 32: GPIO_SS Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R2046(R0x000007FE)	wg7_xor	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R1790(R0x000006FE)	wg_gr_trigger_sel	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)

Table 33: 0: Sensor Core Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12296(R0x00003008)	x_addr_end_	0000 0000 0000 0000 0000 dddd dddd dddd	2627 (0x00000A43)
R12292(R0x00003004)	x_addr_start_	0000 0000 0000 0000 0000 dddd dddd dddd	28 (0x0000001C)
R12448(R0x000030A0)	x_even_inc_	0000 0000 0000 0000 0000 0000 0000 000?	1 (0x00000001)
R12450(R0x000030A2)	x_odd_inc_	0000 0000 0000 0000 0000 0000 0000 dddd	1 (0x00000001)
R12294(R0x00003006)	y_addr_end_	0000 0000 0000 0000 0000 dddd dddd dddd	1967 (0x000007AF)
R12290(R0x00003002)	y_addr_start_	0000 0000 0000 0000 0000 dddd dddd dddd	16 (0x00000010)
R12452(R0x000030A4)	y_even_inc_	0000 0000 0000 0000 0000 0000 0000 000?	1 (0x00000001)
R12454(R0x000030A6)	y_odd_inc_	0000 0000 0000 0000 0000 0000 00dd dddd	1 (0x00000001)

Table 34: 1: SOC1 Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13242(R0x000033BA)	apedge_control	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R12936(R0x00003288)	aperture_additive_clip_limit	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R12934(R0x00003286)	aperture_gain_value	0000 0000 0000 0000 0000 0000 0ddd dddd	0 (0x00000000)
R12930(R0x00003282)	aperture_knee_gain_parameters	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R12932(R0x00003284)	aperture_knee_values	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R12900(R0x00003264)	awb_config	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R12852(R0x00003234)	awb_debug	0000 0000 0000 0000 0000 0000 000d dddd	1 (0x00000001)
R12898(R0x00003262)	awb_luma_th	0000 0000 0000 0000 dddd dddd dddd dddd	65288 (0x0000FF08)
R12858(R0x0000323A)	awb_norm_sumb	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R12856(R0x00003238)	awb_norm_sumg	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R12854(R0x00003236)	awb_norm_sumr	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R12896(R0x00003260)	awb_weight_cnt_hi	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R12904(R0x00003268)	awb_weight_cnt_lo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)

Table 34: 1: SOC1 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12866(R0x00003242)	awb_weight_r0	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12868(R0x00003244)	awb_weight_r1	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12870(R0x00003246)	awb_weight_r2	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12872(R0x00003248)	awb_weight_r3	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12874(R0x0000324A)	awb_weight_r4	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12876(R0x0000324C)	awb_weight_r5	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12878(R0x0000324E)	awb_weight_r6	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12880(R0x00003250)	awb_weight_r7	0000 0000 0000 0000 dddd dddd dddd dddd	20480 (0x00005000)
R12902(R0x00003266)	awb_weight_th	0000 0000 0000 0000 0000 0000 dddd dddd	145 (0x00000091)
R12894(R0x0000325E)	awb_win_height	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R12892(R0x0000325C)	awb_win_width	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R12888(R0x00003258)	awb_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12890(R0x0000325A)	awb_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R12864(R0x00003240)	awb_xy_scale	0000 0000 0000 0000 0000 0000 dddd dddd	51 (0x00000033)
R12860(R0x0000323C)	awb_x_shift	0000 0000 0000 0000 0000 0ddd dddd dddd	3 (0x00000003)
R12862(R0x0000323E)	awb_y_shift	0000 0000 0000 0000 0000 0ddd dddd dddd	3 (0x00000003)
R12918(R0x00003276)	black_level_to_ccm	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13018(R0x000032DA)	blue_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R12928(R0x00003280)	blue_offset	0000 0000 0000 0000 0000 000d dddd dddd	168 (0x000000A8)
R12990(R0x000032BE)	blue_offset_to_ccm	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13214(R0x0000339E)	bnr_blend_strength	0000 0000 0000 0000 0ddd dddd dddd dddd	12684 (0x0000318C)
R13194(R0x0000338A)	bnr_minmax_thresh	0000 0000 0000 0000 0000 0000 dddd dddd	128 (0x00000080)
R13212(R0x0000339C)	bnr_thresh_gain_blue	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13206(R0x00003396)	bnr_thresh_gain_green	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)

Table 34: 1: SOC1 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13200(R0x00003390)	bnr_thresh_gain_red	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13208(R0x00003398)	bnr_thresh_low_blue	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13202(R0x00003392)	bnr_thresh_low_green	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R13196(R0x0000338C)	bnr_thresh_low_red	0000 0000 0000 0000 0000 0000 dddd dddd	32 (0x00000020)
R12996(R0x000032C4)	ccm_elements_1_and_2	0000 0000 0000 0000 dddd dddd dddd dddd	55790 (0x0000D9EE)
R12998(R0x000032C6)	ccm_elements_3_and_4	0000 0000 0000 0000 dddd dddd dddd dddd	11035 (0x00002B1B)
R13000(R0x000032C8)	ccm_elements_5_and_6	0000 0000 0000 0000 dddd dddd dddd dddd	35818 (0x00008BEA)
R13002(R0x000032CA)	ccm_elements_7_and_8	0000 0000 0000 0000 dddd dddd dddd dddd	32022 (0x00007D16)
R13004(R0x000032CC)	ccm_elements_9_and_signs	0000 0000 0000 0000 00dd dddd dddd dddd	11714 (0x00002DC2)
R12994(R0x000032C2)	ccm_exp_high_byte	0000 0000 0000 0000 0000 dddd dddd dddd	1828 (0x00000724)
R12992(R0x000032C0)	ccm_exp_low_byte	0000 0000 0000 0000 0ddd dddd dddd dddd	14627 (0x00003923)
R13136(R0x00003350)	cdc15_hi_thr_comb	0000 0000 0000 0000 0000 00dd dddd dddd	48 (0x00000030)
R13138(R0x00003352)	cdc15_hi_thr_satur	0000 0000 0000 0000 0000 000d dddd dddd	240 (0x000000F0)
R13140(R0x00003354)	cdc15_lo_thr_comb	0000 0000 0000 0000 0000 00dd dddd dddd	288 (0x00000120)
R13142(R0x00003356)	cdc15_lo_thr_satur	0000 0000 0000 0000 0000 000d dddd dddd	368 (0x00000170)
R13186(R0x00003382)	cdc_hi_thr_comb	0000 0000 0000 0000 0000 00dd dddd dddd	48 (0x00000030)
R13188(R0x00003384)	cdc_hi_thr_satur	0000 0000 0000 0000 0000 000d dddd dddd	240 (0x000000F0)
R13190(R0x00003386)	cdc_lo_thr_comb	0000 0000 0000 0000 0000 00dd dddd dddd	288 (0x00000120)
R13192(R0x00003388)	cdc_lo_thr_satur	0000 0000 0000 0000 0000 000d dddd dddd	368 (0x00000170)
R12816(R0x00003210)	color_pipeline_control	0000 0000 0000 0000 0ddd dd0d dddd d000	416 (0x000001A0)
R12960(R0x000032A0)	dkdelta_ccm_cc1	0000 0000 0000 0000 0000 000d dddd dddd	322 (0x00000142)
R12962(R0x000032A2)	dkdelta_ccm_cc2	0000 0000 0000 0000 0000 000d dddd dddd	217 (0x000000D9)
R12964(R0x000032A4)	dkdelta_ccm_cc3	0000 0000 0000 0000 0000 000d dddd dddd	485 (0x000001E5)
R12966(R0x000032A6)	dkdelta_ccm_cc4	0000 0000 0000 0000 0000 000d dddd dddd	43 (0x0000002B)

Table 34: 1: SOC1 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12968(R0x000032A8)	dkdelta_ccm_cc5	0000 0000 0000 0000 0000 000d dddd dddd	327 (0x00000147)
R12970(R0x000032AA)	dkdelta_ccm_cc6	0000 0000 0000 0000 0000 000d dddd dddd	142 (0x0000008E)
R12972(R0x000032AC)	dkdelta_ccm_cc7	0000 0000 0000 0000 0000 000d dddd dddd	490 (0x000001EA)
R12974(R0x000032AE)	dkdelta_ccm_cc8	0000 0000 0000 0000 0000 000d dddd dddd	125 (0x0000007D)
R12976(R0x000032B0)	dkdelta_ccm_cc9	0000 0000 0000 0000 0000 000d dddd dddd	409 (0x00000199)
R12978(R0x000032B2)	dkdelta_ccm_ctl	0000 0000 0000 0000 00dd dddd dddd dddd	8980 (0x00002314)
R12982(R0x000032B6)	dkdelta_ccm_exp1	0000 0000 0000 0000 0ddd dddd dddd dddd	18724 (0x00004924)
R12984(R0x000032B8)	dkdelta_ccm_exp2	0000 0000 0000 0000 0000 dddd dddd dddd	2340 (0x00000924)
R12980(R0x000032B4)	dkdelta_ccm_scale	0000 0000 0000 0000 0000 00dd dddd dddd	528 (0x00000210)
R13050(R0x000032FA)	fd_sum	0000 0000 0000 0000 ???? ???? ???? ???? ????	0 (0x00000000)
R13048(R0x000032F8)	fd_win_height	0000 0000 0000 0000 0000 0ddd dddd dddd	5 (0x00000005)
R13046(R0x000032F6)	fd_win_width	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R13042(R0x000032F2)	fd_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13044(R0x000032F4)	fd_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13232(R0x000033B0)	ffnr_alpha_beta	0000 0000 0000 0000 0ddd dddd 0ddd dddd	5419 (0x0000152B)
R13238(R0x000033B6)	ffnr_mix_thresh_gain	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R13234(R0x000033B2)	ffnr_mix_thresh_y	0000 0000 0000 0000 0000 0000 dddd dddd	10 (0x0000000A)
R13236(R0x000033B4)	ffnr_mix_thresh_y_gain	0000 0000 0000 0000 0000 0000 dddd dddd	8 (0x00000008)
R12884(R0x00003254)	first_color	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R13100(R0x0000332C)	fm_blank_frames	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R13108(R0x00003334)	fm_frame_count	0000 0000 0000 0000 ???? ???? ???? ???? ????	0 (0x00000000)
R13106(R0x00003332)	fm_line_count	0000 0000 0000 0000 0000 ???? ???? ????	0 (0x00000000)
R13218(R0x000033A2)	grb_neg_thresholds	0000 0000 0000 0000 dddd dddd dddd dddd	4104 (0x00001008)
R13222(R0x000033A6)	grb_position_neg	0000 0000 0000 0000 dddd dddd dddd dddd	3855 (0x00000F0F)

Table 34: 1: SOC1 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13220(R0x000033A4)	grb_position_pos	0000 0000 0000 0000 dddd dddd dddd dddd	3855 (0x00000F0F)
R13224(R0x000033A8)	grb_position_window_x0	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13228(R0x000033AC)	grb_position_window_x1	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13226(R0x000033AA)	grb_position_window_y0	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13230(R0x000033AE)	grb_position_window_y1	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13216(R0x000033A0)	grb_pos_thresholds	0000 0000 0000 0000 dddd dddd dddd dddd	4104 (0x00001008)
R13014(R0x000032D6)	green1_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R12924(R0x0000327C)	green1_offset	0000 0000 0000 0000 0000 000d dddd dddd	168 (0x000000A8)
R13016(R0x000032D8)	green2_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R12926(R0x0000327E)	green2_offset	0000 0000 0000 0000 0000 000d dddd dddd	168 (0x000000A8)
R12988(R0x000032BC)	green_offset_to_ccm	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13184(R0x00003380)	kernel_config	0000 0000 0000 0000 d00d dd0d dddd dddd	1231 (0x000004CF)
R12886(R0x00003256)	last_row	0000 0000 0000 0000 0000 0ddd dddd dddd	1947 (0x0000079B)
R12910(R0x0000326E)	low_pass_yuv_filter	0000 0000 0000 0000 0000 0000 dddd dddd	128 (0x00000080)
R13102(R0x0000332E)	output_format_configuration	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13104(R0x00003330)	output_format_test	0000 0000 0000 0000 0000 d00d d0dd dddd	0 (0x00000000)
R13146(R0x0000335A)	pcr_color_gain1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13148(R0x0000335C)	pcr_color_gain2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13150(R0x0000335E)	pcr_color_gain3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13164(R0x0000336C)	pcr_color_gain4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13166(R0x0000336E)	pcr_color_gain5	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13168(R0x00003370)	pcr_color_gain6	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13170(R0x00003372)	pcr_color_gain7	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13172(R0x00003374)	pcr_color_gain8	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 34: 1: SOC1 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13174(R0x00003376)	pcr_color_gain9	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13144(R0x00003358)	pcr_color_kill_control	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R12958(R0x0000329E)	preview_hunting_gain_enable	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R13012(R0x000032D4)	red_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R12922(R0x0000327A)	red_offset	0000 0000 0000 0000 0000 000d dddd dddd	168 (0x000000A8)
R12986(R0x000032BA)	red_offset_to_ccm	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R12906(R0x0000326A)	scale_sharp_control	0000 0000 0000 0000 00dd dddd dddd dddd	4616 (0x00001208)
R13020(R0x000032DC)	second_digital_gain	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13130(R0x0000334A)	sepia_constants	0000 0000 0000 0000 dddd dddd dddd dddd	45091 (0x0000B023)
R12942(R0x0000328E)	sffb_noise_control_b	0000 0000 0000 0000 dddd dddd dddd dddd	33036 (0x0000810C)
R12940(R0x0000328C)	sffb_noise_control_g	0000 0000 0000 0000 dddd dddd dddd dddd	33036 (0x0000810C)
R12938(R0x0000328A)	sffb_noise_control_r	0000 0000 0000 0000 dddd dddd dddd dddd	33036 (0x0000810C)
R12946(R0x00003292)	sffb_sobel_flat	0000 0000 0000 0000 0000 0000 dddd dddd	31 (0x0000001F)
R12948(R0x00003294)	sffb_sobel_sharp	0000 0000 0000 0000 0000 0000 dddd dddd	255 (0x000000FF)
R12944(R0x00003290)	sffb_weight_control	0000 0000 0000 0000 0ddd 0ddd dddd 0ddd	21508 (0x00005404)
R13152(R0x00003360)	skin_tone_th1	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13154(R0x00003362)	skin_tone_th2	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13156(R0x00003364)	skin_tone_th3	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13158(R0x00003366)	skin_tone_th4	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13160(R0x00003368)	skin_tone_th5	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13162(R0x0000336A)	skin_tone_th6	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13128(R0x00003348)	special_effect_parameters	0000 0000 0000 0000 dddd dddd 0ddd dddd	25664 (0x00006440)
R12916(R0x00003274)	threshold_for_y_filter_b_channel	0000 0000 0000 0000 0000 0000 0ddd dddd	42 (0x0000002A)
R12914(R0x00003272)	threshold_for_y_filter_g_channel	0000 0000 0000 0000 0000 dddd dddd dddd	2020 (0x000007E4)

Table 34: 1: SOC1 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R12912(R0x00003270)	threshold_for_y_filter_r_channel	0000 0000 0000 0000 0000 dddd dddd dddd	1962 (0x000007AA)
R13246(R0x000033BE)	ua_knee_l	0000 0000 0000 0000 0000 00dd dddd dddd	8 (0x00000008)
R13250(R0x000033C2)	ua_weights	0000 0000 0000 0000 dddd dddd 0000 0000	4352 (0x00001100)
R12844(R0x0000322C)	x_ratio	0000 0000 0000 0000 0000 dddd dddd dddd	2048 (0x00000800)
R13180(R0x0000337C)	yuv_ycbcr_control	0000 0000 0000 0000 0000 0000 0000 dddd	6 (0x00000006)
R12846(R0x0000322E)	y_ratio	0000 0000 0000 0000 0000 dddd dddd dddd	2048 (0x00000800)
R13182(R0x0000337E)	y_rgb_offset	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R12834(R0x00003222)	zoom_window_x0	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12836(R0x00003224)	zoom_window_x1	0000 0000 0000 0000 0000 dddd dddd dddd	2591 (0x00000A1F)
R12838(R0x00003226)	zoom_window_y0	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R12840(R0x00003228)	zoom_window_y1	0000 0000 0000 0000 0000 dddd dddd dddd	1943 (0x00000797)

Table 35: 2: SOC2 Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13698(R0x00003582)	ae_sum_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13700(R0x00003584)	ae_sum_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13702(R0x00003586)	ae_sum_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13704(R0x00003588)	ae_sum_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13706(R0x0000358A)	ae_sum_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13708(R0x0000358C)	ae_sum_5	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13696(R0x00003580)	ae_zones_oob_ctrl	0000 0000 0000 0000 0000 0000 0ddd dddd	127 (0x0000007F)
R13694(R0x0000357E)	ae_zone_height	0000 0000 0000 0000 0000 0ddd dddd dddd	X
R13692(R0x0000357C)	ae_zone_width	0000 0000 0000 0000 0000 dddd dddd dddd	X
R13686(R0x00003576)	ae_zone_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	X

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13688(R0x00003578)	ae_zone_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	X
R13466(R0x0000349A)	af_config_coeffs_0	0000 0000 0000 0000 0ddd dddd dddd dddd	12 (0x0000000C)
R13468(R0x0000349C)	af_config_coeffs_1	0000 0000 0000 0000 0ddd dddd dddd dddd	0 (0x00000000)
R13470(R0x0000349E)	af_config_coeffs_2	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13472(R0x000034A0)	af_config_coeffs_3	0000 0000 0000 0000 0ddd dddd dddd dddd	2 (0x00000002)
R13474(R0x000034A2)	af_config_coeffs_4	0000 0000 0000 0000 0ddd dddd dddd dddd	2048 (0x00000800)
R13476(R0x000034A4)	af_config_coeffs_5	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R13458(R0x00003492)	af_config_filters	0000 0000 0000 0000 000d dddd dddd dddd	4111 (0x0000100F)
R13460(R0x00003494)	af_config_thresholds_1	0000 0000 0000 0000 dddd dddd dddd dddd	256 (0x00000100)
R13488(R0x000034B0)	af_filter1_oflow_cnt	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13478(R0x000034A6)	af_filter1_sharp_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13480(R0x000034A8)	af_filter1_sharp_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13482(R0x000034AA)	af_filter1_sharp_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13484(R0x000034AC)	af_filter1_sharp_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13486(R0x000034AE)	af_filter1_sharp_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13516(R0x000034CC)	af_filter2_oflow_cnt	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13506(R0x000034C2)	af_filter2_sharp_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13508(R0x000034C4)	af_filter2_sharp_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13510(R0x000034C6)	af_filter2_sharp_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13512(R0x000034C8)	af_filter2_sharp_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13514(R0x000034CA)	af_filter2_sharp_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13446(R0x00003486)	af_lum_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13448(R0x00003488)	af_lum_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13450(R0x0000348A)	af_lum_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13452(R0x0000348C)	af_lum_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13454(R0x0000348E)	af_lum_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13438(R0x0000347E)	af_zones_oob_ctrl	0000 0000 0000 0000 0000 0000 00dd dddd	63 (0x0000003F)
R13436(R0x0000347C)	af_zone_height	0000 0000 0000 0000 0000 0ddd dddd dddd	120 (0x00000078)
R13434(R0x0000347A)	af_zone_width	0000 0000 0000 0000 0000 dddd dddd dddd	160 (0x000000A0)
R13430(R0x00003476)	af_zone_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	16 (0x00000010)
R13432(R0x00003478)	af_zone_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13732(R0x000035A4)	bright_color_kill_controls	0000 0000 0000 0000 0000 0ddd dddd dddd	1428 (0x00000594)
R13868(R0x0000362C)	b_gamma_curve_knees_0_1	0000 0000 0000 0000 dddd dddd dddd dddd	6912 (0x00001B00)
R13878(R0x00003636)	b_gamma_curve_knees_10_11	0000 0000 0000 0000 dddd dddd dddd dddd	59617 (0x0000E8E1)
R13880(R0x00003638)	b_gamma_curve_knees_12_13	0000 0000 0000 0000 dddd dddd dddd dddd	62190 (0x0000F2EE)
R13882(R0x0000363A)	b_gamma_curve_knees_14_15	0000 0000 0000 0000 dddd dddd dddd dddd	63990 (0x0000F9F6)
R13884(R0x0000363C)	b_gamma_curve_knees_16_17	0000 0000 0000 0000 dddd dddd dddd dddd	65019 (0x0000FDFB)
R13870(R0x0000362E)	b_gamma_curve_knees_2_3	0000 0000 0000 0000 dddd dddd dddd dddd	19502 (0x00004C2E)
R13872(R0x00003630)	b_gamma_curve_knees_4_5	0000 0000 0000 0000 dddd dddd dddd dddd	39032 (0x00009878)
R13874(R0x00003632)	b_gamma_curve_knees_6_7	0000 0000 0000 0000 dddd dddd dddd dddd	49584 (0x0000C1B0)
R13876(R0x00003634)	b_gamma_curve_knees_8_9	0000 0000 0000 0000 dddd dddd dddd dddd	55759 (0x0000D9CF)
R13886(R0x0000363E)	b_gamma_curve_knee_18	0000 0000 0000 0000 0000 0000 dddd dddd	255 (0x000000FF)
R13786(R0x000035DA)	clip1_cnt_hi	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R13784(R0x000035D8)	clip1_cnt_lo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13440(R0x00003480)	clip1_config	0000 0000 0000 0000 0000 0000 0000 0ddd	2 (0x00000002)
R13740(R0x000035AC)	clip1_max	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13738(R0x000035AA)	clip1_min	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13800(R0x000035E8)	clip1_win_x_end	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13796(R0x000035E4)	clip1_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13802(R0x000035EA)	clip1_win_y_end	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R13798(R0x000035E6)	clip1_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13794(R0x000035E2)	clip2_cnt_hi	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R13792(R0x000035E0)	clip2_cnt_lo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13442(R0x00003482)	clip2_config	0000 0000 0000 0000 0000 0000 0000 00dd	2 (0x00000002)
R13744(R0x000035B0)	clip2_max	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13742(R0x000035AE)	clip2_min	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13746(R0x000035B2)	clip2_win_x_end	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R13772(R0x000035CC)	clip2_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13736(R0x000035A8)	clip2_win_y_end	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R13774(R0x000035CE)	clip2_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13428(R0x00003474)	clip3_cnt_hi	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R13426(R0x00003472)	clip3_cnt_lo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13412(R0x00003464)	clip3_config	0000 0000 0000 0000 0000 0000 0000 00dd	2 (0x00000002)
R13416(R0x00003468)	clip3_max	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R13414(R0x00003466)	clip3_min	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13422(R0x0000346E)	clip3_win_x_end	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R13418(R0x0000346A)	clip3_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13424(R0x00003470)	clip3_win_y_end	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R13420(R0x0000346C)	clip3_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13518(R0x000034CE)	daf_config_0	0000 0000 0000 0000 0ddd dddd dddd dddd	31745 (0x00007C01)
R13520(R0x000034D0)	daf_config_1	0000 0000 0000 0000 0000 dddd 0000 00dd	0 (0x00000000)
R13534(R0x000034DE)	daf_config_thresholds_1	0000 0000 0000 0000 dddd dddd dddd dddd	256 (0x00000100)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13522(R0x000034D2)	daf_filter_oflow_cnt	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13524(R0x000034D4)	daf_filter_sharp_0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13526(R0x000034D6)	daf_filter_sharp_1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13528(R0x000034D8)	daf_filter_sharp_2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13530(R0x000034DA)	daf_filter_sharp_3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13532(R0x000034DC)	daf_filter_sharp_4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13730(R0x000035A2)	dark_color_kill_controls	0000 0000 0000 0000 0000 0000 dddd dddd	20 (0x00000014)
R13632(R0x00003540)	enable_tonal_curve	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)
R13848(R0x00003618)	g_gamma_curve_knees_0_1	0000 0000 0000 0000 dddd dddd dddd dddd	6912 (0x00001B00)
R13858(R0x00003622)	g_gamma_curve_knees_10_11	0000 0000 0000 0000 dddd dddd dddd dddd	59617 (0x0000E8E1)
R13860(R0x00003624)	g_gamma_curve_knees_12_13	0000 0000 0000 0000 dddd dddd dddd dddd	62190 (0x0000F2EE)
R13862(R0x00003626)	g_gamma_curve_knees_14_15	0000 0000 0000 0000 dddd dddd dddd dddd	63990 (0x0000F9F6)
R13864(R0x00003628)	g_gamma_curve_knees_16_17	0000 0000 0000 0000 dddd dddd dddd dddd	65019 (0x0000FDFB)
R13850(R0x0000361A)	g_gamma_curve_knees_2_3	0000 0000 0000 0000 dddd dddd dddd dddd	19502 (0x00004C2E)
R13852(R0x0000361C)	g_gamma_curve_knees_4_5	0000 0000 0000 0000 dddd dddd dddd dddd	39032 (0x00009878)
R13854(R0x0000361E)	g_gamma_curve_knees_6_7	0000 0000 0000 0000 dddd dddd dddd dddd	49584 (0x0000C1B0)
R13856(R0x00003620)	g_gamma_curve_knees_8_9	0000 0000 0000 0000 dddd dddd dddd dddd	55759 (0x0000D9CF)
R13866(R0x0000362A)	g_gamma_curve_knee_18	0000 0000 0000 0000 0000 0000 dddd dddd	255 (0x000000FF)
R13722(R0x0000359A)	hist0_bin_config	0000 0000 0000 0000 0000 0000 ddd0 dddd	13 (0x0000000D)
R13728(R0x000035A0)	hist0_bin_offset	0000 0000 0000 0000 0000 dddd dddd dddd	10 (0x0000000A)
R13748(R0x000035B4)	hist0_bin_stats1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13750(R0x000035B6)	hist0_bin_stats2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13752(R0x000035B8)	hist0_bin_stats3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13754(R0x000035BA)	hist0_bin_stats4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13716(R0x00003594)	hist0_pre_divider	0000 0000 0000 0000 dddd dddd dddd dddd	778 (0x0000030A)
R13724(R0x0000359C)	hist1a_bin_config	0000 0000 0000 0000 0000 0000 000d dddd	10 (0x0000000A)
R13714(R0x00003592)	hist1a_bin_offset	0000 0000 0000 0000 0000 dddd dddd dddd	10 (0x0000000A)
R13756(R0x000035BC)	hist1a_bin_stats1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13758(R0x000035BE)	hist1a_bin_stats2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13760(R0x000035C0)	hist1a_bin_stats3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13762(R0x000035C2)	hist1a_bin_stats4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13718(R0x00003596)	hist1a_pre_divider	0000 0000 0000 0000 dddd dddd dddd dddd	778 (0x0000030A)
R13726(R0x0000359E)	hist1b_bin_config	0000 0000 0000 0000 0000 0000 000d dddd	8 (0x00000008)
R13712(R0x00003590)	hist1b_bin_offset	0000 0000 0000 0000 0000 dddd dddd dddd	10 (0x0000000A)
R13764(R0x000035C4)	hist1b_bin_stats1	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13766(R0x000035C6)	hist1b_bin_stats2	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13768(R0x000035C8)	hist1b_bin_stats3	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13770(R0x000035CA)	hist1b_bin_stats4	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13720(R0x00003598)	hist1b_pre_divider	0000 0000 0000 0000 dddd dddd dddd dddd	778 (0x0000030A)
R13710(R0x0000358E)	hist_pga_data_select	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)
R13780(R0x000035D4)	hist_win_x_end	0000 0000 0000 0000 0000 dddd dddd dddd	640 (0x00000280)
R13776(R0x000035D0)	hist_win_x_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R13782(R0x000035D6)	hist_win_y_end	0000 0000 0000 0000 0000 0ddd dddd dddd	480 (0x000001E0)
R13778(R0x000035D2)	hist_win_y_start	0000 0000 0000 0000 0000 0ddd dddd dddd	0 (0x00000000)
R13908(R0x00003654)	p_b_p0q0	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)
R13910(R0x00003656)	p_b_p0q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13912(R0x00003658)	p_b_p0q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13914(R0x0000365A)	p_b_p0q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13916(R0x0000365C)	p_b_p0q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13972(R0x00003694)	p_b_p1q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13974(R0x00003696)	p_b_p1q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13976(R0x00003698)	p_b_p1q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13978(R0x0000369A)	p_b_p1q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13980(R0x0000369C)	p_b_p1q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14036(R0x000036D4)	p_b_p2q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14038(R0x000036D6)	p_b_p2q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14040(R0x000036D8)	p_b_p2q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14042(R0x000036DA)	p_b_p2q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14044(R0x000036DC)	p_b_p2q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14100(R0x00003714)	p_b_p3q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14102(R0x00003716)	p_b_p3q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14104(R0x00003718)	p_b_p3q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14106(R0x0000371A)	p_b_p3q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14108(R0x0000371C)	p_b_p3q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14164(R0x00003754)	p_b_p4q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14166(R0x00003756)	p_b_p4q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14168(R0x00003758)	p_b_p4q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14170(R0x0000375A)	p_b_p4q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14172(R0x0000375C)	p_b_p4q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13888(R0x00003640)	p_g1_p0q0	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)
R13890(R0x00003642)	p_g1_p0q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13892(R0x00003644)	p_g1_p0q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13894(R0x00003646)	p_g1_p0q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13896(R0x00003648)	p_g1_p0q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13952(R0x00003680)	p_g1_p1q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13954(R0x00003682)	p_g1_p1q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13956(R0x00003684)	p_g1_p1q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13958(R0x00003686)	p_g1_p1q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13960(R0x00003688)	p_g1_p1q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14016(R0x000036C0)	p_g1_p2q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14018(R0x000036C2)	p_g1_p2q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14020(R0x000036C4)	p_g1_p2q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14022(R0x000036C6)	p_g1_p2q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14024(R0x000036C8)	p_g1_p2q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14080(R0x00003700)	p_g1_p3q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14082(R0x00003702)	p_g1_p3q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14084(R0x00003704)	p_g1_p3q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14086(R0x00003706)	p_g1_p3q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14088(R0x00003708)	p_g1_p3q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14144(R0x00003740)	p_g1_p4q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14146(R0x00003742)	p_g1_p4q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14148(R0x00003744)	p_g1_p4q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14150(R0x00003746)	p_g1_p4q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14152(R0x00003748)	p_g1_p4q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13918(R0x0000365E)	p_g2_p0q0	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)
R13920(R0x00003660)	p_g2_p0q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13922(R0x00003662)	p_g2_p0q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13924(R0x00003664)	p_g2_p0q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13926(R0x00003666)	p_g2_p0q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13982(R0x0000369E)	p_g2_p1q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13984(R0x000036A0)	p_g2_p1q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13986(R0x000036A2)	p_g2_p1q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13988(R0x000036A4)	p_g2_p1q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13990(R0x000036A6)	p_g2_p1q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14046(R0x000036DE)	p_g2_p2q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14048(R0x000036E0)	p_g2_p2q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14050(R0x000036E2)	p_g2_p2q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14052(R0x000036E4)	p_g2_p2q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14054(R0x000036E6)	p_g2_p2q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14110(R0x0000371E)	p_g2_p3q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14112(R0x00003720)	p_g2_p3q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14114(R0x00003722)	p_g2_p3q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14116(R0x00003724)	p_g2_p3q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14118(R0x00003726)	p_g2_p3q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14174(R0x0000375E)	p_g2_p4q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14176(R0x00003760)	p_g2_p4q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14178(R0x00003762)	p_g2_p4q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14180(R0x00003764)	p_g2_p4q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14182(R0x00003766)	p_g2_p4q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13898(R0x0000364A)	p_r_p0q0	0000 0000 0000 0000 dddd dddd dddd dddd	16 (0x00000010)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13900(R0x0000364C)	p_r_p0q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13902(R0x0000364E)	p_r_p0q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13904(R0x00003650)	p_r_p0q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13906(R0x00003652)	p_r_p0q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13962(R0x0000368A)	p_r_p1q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13964(R0x0000368C)	p_r_p1q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13966(R0x0000368E)	p_r_p1q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13968(R0x00003690)	p_r_p1q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13970(R0x00003692)	p_r_p1q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14026(R0x000036CA)	p_r_p2q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14028(R0x000036CC)	p_r_p2q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14030(R0x000036CE)	p_r_p2q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14032(R0x000036D0)	p_r_p2q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14034(R0x000036D2)	p_r_p2q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14090(R0x0000370A)	p_r_p3q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14092(R0x0000370C)	p_r_p3q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14094(R0x0000370E)	p_r_p3q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14096(R0x00003710)	p_r_p3q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14098(R0x00003712)	p_r_p3q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14154(R0x0000374A)	p_r_p4q0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14156(R0x0000374C)	p_r_p4q1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14158(R0x0000374E)	p_r_p4q2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14160(R0x00003750)	p_r_p4q3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14162(R0x00003752)	p_r_p4q4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13678(R0x0000356E)	reciprocal_of_400_minus_x6	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13664(R0x00003560)	reciprocal_of_x0_minus_zero	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13666(R0x00003562)	reciprocal_of_x1_minus_x0	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13668(R0x00003564)	reciprocal_of_x2_minus_x1	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13670(R0x00003566)	reciprocal_of_x3_minus_x2	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13672(R0x00003568)	reciprocal_of_x4_minus_x3	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13674(R0x0000356A)	reciprocal_of_x5_minus_x4	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13676(R0x0000356C)	reciprocal_of_x6_minus_x5	0000 0000 0000 0000 dddd 0ddd dddd dddd	61696 (0x0000F100)
R13828(R0x00003604)	r_gamma_curve_knees_0_1	0000 0000 0000 0000 dddd dddd dddd dddd	6912 (0x00001B00)
R13838(R0x0000360E)	r_gamma_curve_knees_10_11	0000 0000 0000 0000 dddd dddd dddd dddd	59617 (0x0000E8E1)
R13840(R0x00003610)	r_gamma_curve_knees_12_13	0000 0000 0000 0000 dddd dddd dddd dddd	62190 (0x0000F2EE)
R13842(R0x00003612)	r_gamma_curve_knees_14_15	0000 0000 0000 0000 dddd dddd dddd dddd	63990 (0x0000F9F6)
R13844(R0x00003614)	r_gamma_curve_knees_16_17	0000 0000 0000 0000 dddd dddd dddd dddd	65019 (0x0000FDFB)
R13830(R0x00003606)	r_gamma_curve_knees_2_3	0000 0000 0000 0000 dddd dddd dddd dddd	19502 (0x00004C2E)
R13832(R0x00003608)	r_gamma_curve_knees_4_5	0000 0000 0000 0000 dddd dddd dddd dddd	39032 (0x00009878)
R13834(R0x0000360A)	r_gamma_curve_knees_6_7	0000 0000 0000 0000 dddd dddd dddd dddd	49584 (0x0000C1B0)
R13836(R0x0000360C)	r_gamma_curve_knees_8_9	0000 0000 0000 0000 dddd dddd dddd dddd	55759 (0x0000D9CF)
R13846(R0x00003616)	r_gamma_curve_knee_18	0000 0000 0000 0000 0000 0000 dddd dddd	255 (0x000000FF)
R13634(R0x00003542)	tonal_x0	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13636(R0x00003544)	tonal_x1	0000 0000 0000 0000 0000 00dd dddd dddd	256 (0x00000100)
R13638(R0x00003546)	tonal_x2	0000 0000 0000 0000 0000 00dd dddd dddd	384 (0x00000180)
R13640(R0x00003548)	tonal_x3	0000 0000 0000 0000 0000 00dd dddd dddd	512 (0x00000200)
R13642(R0x0000354A)	tonal_x4	0000 0000 0000 0000 0000 00dd dddd dddd	640 (0x00000280)
R13644(R0x0000354C)	tonal_x5	0000 0000 0000 0000 0000 00dd dddd dddd	768 (0x00000300)

Table 35: 2: SOC2 Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13646(R0x0000354E)	tonal_x6	0000 0000 0000 0000 0000 00dd dddd dddd	896 (0x00000380)
R13648(R0x00003550)	tonal_y0	0000 0000 0000 0000 0000 00dd dddd dddd	128 (0x00000080)
R13650(R0x00003552)	tonal_y1	0000 0000 0000 0000 0000 00dd dddd dddd	256 (0x00000100)
R13652(R0x00003554)	tonal_y2	0000 0000 0000 0000 0000 00dd dddd dddd	384 (0x00000180)
R13654(R0x00003556)	tonal_y3	0000 0000 0000 0000 0000 00dd dddd dddd	512 (0x00000200)
R13656(R0x00003558)	tonal_y4	0000 0000 0000 0000 0000 00dd dddd dddd	640 (0x00000280)
R13658(R0x0000355A)	tonal_y5	0000 0000 0000 0000 0000 00dd dddd dddd	768 (0x00000300)
R13660(R0x0000355C)	tonal_y6	0000 0000 0000 0000 0000 00dd dddd dddd	896 (0x00000380)

Table 36: SYSCTL Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R66(R0x00000042)	burnin_passcode	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R0(R0x00000000)	chip_id	0000 0000 0000 0000 ???? ???? ???? ????	10368 (0x00002880)
R48(R0x00000030)	clk_otpm_clock_divider_selection	0000 0000 0000 0000 0000 0000 0000 00dd	3 (0x00000003)
R22(R0x00000016)	clocks_control	0000 0000 0000 0000 d0dd dddd 0ddd dddd	0 (0x00000000)
R40(R0x00000028)	en_vdd_dis_soft	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)
R68(R0x00000044)	gen_purp_0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R70(R0x00000046)	gen_purp_1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R72(R0x00000048)	gen_purp_2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R74(R0x0000004A)	gen_purp_3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R6(R0x00000006)	i2c_control	0000 0000 0000 0000 dd00 ddd0 0000 0000	0 (0x00000000)
R28(R0x0000001C)	mcu_boot_mode	0000 0000 0000 0000 dddd dddd 00d0 0ddd	0 (0x00000000)
R38(R0x00000026)	mcu_rom_initialize_sensor	0000 0000 0000 0000 0000 0000 0000 0ddd	7 (0x00000007)
R30(R0x0000001E)	pad_slew_pad_config	0000 0000 0000 0000 d000 0ddd 0ddd 0ddd	1024 (0x00000400)
R20(R0x00000014)	pll_control	0000 0000 0000 0000 d0dd dddd dddd dddd	9253 (0x00002425)

Table 36: SYSCTL Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R16(R0x00000010)	pll_dividers	0000 0000 0000 0000 00dd dddd dddd dddd	520 (0x00000208)
R42(R0x0000002A)	pll_p4_p5_p6_dividers	0000 0000 0000 0000 0ddd dddd dddd dddd	30682 (0x000077DA)
R44(R0x0000002C)	pll_p7_divider	0000 0000 0000 0000 000d 0000 0000 dddd	4103 (0x00001007)
R18(R0x00000012)	pll_p_dividers	0000 0000 0000 0000 00dd dddd dddd dddd	176 (0x000000B0)
R52(R0x00000034)	reg_0034	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R54(R0x00000036)	reg_0036	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R80(R0x00000050)	release_version	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R26(R0x0000001A)	reset_and_misc_control	0000 0000 0000 0000 0000 0ddd 00dd dddd	24 (0x00000018)
R46(R0x0000002E)	sensor_clock_divider	0000 0000 0000 0000 0000 0d0d 0000 dddd	0 (0x00000000)
R24(R0x00000018)	standby_control_and_status	0000 0000 0000 0000 ddd0 0000 0ddd dddd	24585 (0x00006009)
R50(R0x00000032)	s_clk_pad_slew_rate	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)
R64(R0x00000040)	user_defined_i2c_device_address_id	0000 0000 0000 0000 dddd ddd0 dddd ddd0	31352 (0x00007A78)
R34(R0x00000022)	vdd_dis_counter	0000 0000 0000 0000 dddd dddd dddd dddd	1080 (0x00000438)
R32(R0x00000020)	vdd_dis_soft	0000 0000 0000 0000 0000 0000 0000 000d	1 (0x00000001)

Table 37: RX_SS Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R272(R0x00000110)	col_end	0000 0000 0000 0000 0000 dddd dddd dddd	2055 (0x00000807)
R268(R0x0000010C)	col_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R336(R0x00000150)	crc_lane_1_data_rx_ss	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R338(R0x00000152)	crc_lane_2_data_rx_ss	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R310(R0x00000136)	data_visibility_0_	0000 0000 0000 0000 0000 0000 00?? ????	0 (0x00000000)
R312(R0x00000138)	data_visibility_1_	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R314(R0x0000013A)	data_visibility_2_	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R346(R0x0000015A)	eofv_del	0000 0000 0000 0000 0000 0ddd dddd dddd	1151 (0x0000047F)
R276(R0x00000114)	hblank_llead	0000 0000 0000 0000 000d dddd dddd dddd	100 (0x00000064)
R278(R0x00000116)	hblank_ltrail	0000 0000 0000 0000 000d dddd dddd dddd	10 (0x0000000A)
R280(R0x00000118)	lower_seed_value	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R300(R0x0000012C)	mipi_and_ccp2_pixel_data_out_status_0	0000 0000 0000 0000 000? ???? ???? ????	2048 (0x00000800)
R302(R0x0000012E)	mipi_and_ccp2_pixel_data_out_status_1_	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R304(R0x00000130)	mipi_error_reporting_	0000 0000 0000 0000 0000 0??? ???? ????	0 (0x00000000)
R296(R0x00000128)	mipi_receiver_control	0000 0000 0000 0000 dddd d0dd dddd dd0d	4 (0x00000004)
R306(R0x00000132)	packet_level_errors_0_	0000 0000 0000 0000 0000 0??? ???? ????	0 (0x00000000)
R308(R0x00000134)	packet_level_errors_1_and_protocol_layer_errors_	0000 0000 0000 0000 0000 00?? ???? ????	0 (0x00000000)
R316(R0x0000013C)	phy_configuration_control	0000 0000 0000 0000 00dd dddd dddd dddd	6175 (0x0000181F)
R270(R0x0000010E)	row_end	0000 0000 0000 0000 0000 dddd dddd dddd	1543 (0x00000607)
R266(R0x0000010A)	row_start	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)
R274(R0x00000112)	rx_fifo_control	0000 0000 0000 0000 dddd dddd dddd dddd	20 (0x00000014)
R318(R0x0000013E)	testp_colorbar_width	0000 0000 0000 0000 0000 dddd dddd dddd	255 (0x000000FF)
R344(R0x00000158)	testp_vblank	0000 0000 0000 0000 0000 dddd dddd dddd	0 (0x00000000)

Table 37: RX_SS Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R256(R0x00000100)	test_pattern_control	0000 0000 0000 0000 dddd dddd dddd dddd	8192 (0x00002000)
R264(R0x00000108)	test_pxl_blue	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R260(R0x00000104)	test_pxl_green1	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R262(R0x00000106)	test_pxl_green2	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R258(R0x00000102)	test_pxl_red	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R282(R0x0000011A)	upper_seed_value	0000 0000 0000 0000 0000 0000 0000 dddd	0 (0x00000000)
R320(R0x00000140)	vblank_llead	0000 0000 0000 0000 0000 000d dddd dddd	16 (0x00000010)
R322(R0x00000142)	vblank_ltrail	0000 0000 0000 0000 0000 000d dddd dddd	6 (0x00000006)

Table 38: XDMA Registers (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R2434(R0x00000982)	access_ctl_stat	0000 0000 0000 0000 0000 0000 dd0d dddd	0 (0x00000000)
R2564(R0x00000A04)	jtag_ctl	0000 0000 0000 0000 0000 0000 0ddd dddd	0 (0x00000000)
R2562(R0x00000A02)	jtag_tdi	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2566(R0x00000A06)	jtag_tdo	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R2560(R0x00000A00)	jtag_tms	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2446(R0x0000098E)	logical_address_access	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2448(R0x00000990)	mcu_variable_data0	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2450(R0x00000992)	mcu_variable_data1	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2452(R0x00000994)	mcu_variable_data2	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2454(R0x00000996)	mcu_variable_data3	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2456(R0x00000998)	mcu_variable_data4	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2458(R0x0000099A)	mcu_variable_data5	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2460(R0x0000099C)	mcu_variable_data6	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 38: XDMA Registers (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R2462(R0x0000099E)	mcu_variable_data7	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R2442(R0x0000098A)	physical_address_access	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)

Table 39: TX_SS (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15396(R0x00003C24)	crc_control	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R15398(R0x00003C26)	crc_data	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13316(R0x00003404)	custom_short_pkt	0000 0000 0000 0000 00dd dddd dd00 0000	0 (0x00000000)
R13324(R0x0000340C)	custom_short_pkt_wc	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15586(R0x00003CE2)	indirect_address	0000 0000 0000 0000 dddd dddd dddd dddd	65535 (0x0000FFFF)
R15588(R0x00003CE4)	indirect_data	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15584(R0x00003CE0)	indirect_ram_access_ctrl	0000 0000 0000 0000 0000 0000 0000 0ddd	1 (0x00000001)
R15424(R0x00003C40)	jpeg_ctrl	0000 0000 0000 0000 0000 000d dddd dddd	14 (0x0000000E)
R15432(R0x00003C48)	jpeg_exif_bytes	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R15434(R0x00003C4A)	jpeg_hdr_pad_bytes	0000 0000 0000 0000 0000 00dd dddd dddd	0 (0x00000000)
R15456(R0x00003C60)	jpeg_hw_status	0000 0000 0000 0000 0000 000d dddd dddd	0 (0x00000000)
R15430(R0x00003C46)	jpeg_jpop_limit	0000 0000 0000 0000 dddd dddd dddd dddd	65535 (0x0000FFFF)
R15426(R0x00003C42)	jpeg_qtable_sel	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R15428(R0x00003C44)	jpeg_restart	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15360(R0x00003C00)	jpssmode	0000 0000 0000 0000 0ddd 0ddd 0ddd 0ddd	13107 (0x00003333)
R15400(R0x00003C28)	jpss_between_frame_status	0000 0000 0000 0000 0000 ???? ???? ????	4095 (0x00000FFF)
R15362(R0x00003C02)	jpss_ctrl	0000 0000 0000 0000 0ddd dddd dddd dddd	1552 (0x00000610)
R15454(R0x00003C5E)	jpss_debug_sel	0000 0000 0000 0000 0000 0000 000d dddd	0 (0x00000000)
R15376(R0x00003C10)	jpss_dummy_pattern	0000 0000 0000 0000 dddd dddd dddd dddd	65535 (0x0000FFFF)

Table 39: TX_SS (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15380(R0x00003C14)	jpss_exif_tn_ptr	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15384(R0x00003C18)	jpss_frame_length0	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R15386(R0x00003C1A)	jpss_frame_length1	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R15366(R0x00003C06)	jpss_inhgt	0000 0000 0000 0000 0000 0ddd dddd dddd	1944 (0x00000798)
R15364(R0x00003C04)	jpss_inwid	0000 0000 0000 0000 0000 dddd dddd dddd	2592 (0x00000A20)
R15374(R0x00003C0E)	jpss_jpeg_eosi_code	0000 0000 0000 0000 dddd dddd dddd dddd	65469 (0x0000FFBD)
R15372(R0x00003C0C)	jpss_jpeg_sosi_code	0000 0000 0000 0000 dddd dddd dddd dddd	65468 (0x0000FFBC)
R15388(R0x00003C1C)	jpss_rcvd_frame_cnt	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15390(R0x00003C1E)	jpss_rcvd_frame_line_cnt	0000 0000 0000 0000 0000 ???? ???? ????	0 (0x00000000)
R15370(R0x00003C0A)	jpss_spoof_height	0000 0000 0000 0000 0000 dddd dddd dddd	480 (0x000001E0)
R15368(R0x00003C08)	jpss_spoof_width	0000 0000 0000 0000 0000 dddd dddd dddd	1280 (0x00000500)
R15382(R0x00003C16)	jpss_status	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15378(R0x00003C12)	jpss_trigger_mark	0000 0000 0000 0000 0000 dddd dddd dddd	4095 (0x00000FFF)
R15394(R0x00003C22)	jpss_tx_frame_cnt	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R13340(R0x0000341C)	mipi_ccp__sel_class	0000 0000 0000 0000 0000 0000 0000 00dd	0 (0x00000000)
R13312(R0x00003400)	mipi_control	0000 0000 0000 0000 dddd dddd dddd dddd	30766 (0x0000782E)
R13338(R0x0000341A)	mipi_init_timing	0000 0000 0000 0000 0ddd dddd 0ddd dddd	1544 (0x00000608)
R13314(R0x00003402)	mipi_status	0000 0000 0000 0000 000d ddd0 00dd 000d	17 (0x00000011)
R13326(R0x0000340E)	mipi_test	0000 0000 0000 0000 000d 00dd 0ddd 0ddd	0 (0x00000000)
R13342(R0x0000341E)	mipi_test_checksum	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R13332(R0x00003414)	mipi_timing_t_clk_post_clk_pre	0000 0000 0000 0000 00dd dddd 00dd dddd	2817 (0x00000B01)
R13330(R0x00003412)	mipi_timing_t_hs_exit_hs_trail	0000 0000 0000 0000 00dd dddd 0000 dddd	1797 (0x00000705)
R13328(R0x00003410)	mipi_timing_t_hs_zero	0000 0000 0000 0000 0000 dddd 0000 0000	2048 (0x00000800)
R13336(R0x00003418)	mipi_timing_t_lpx	0000 0000 0000 0000 0000 0000 00dd dddd	4 (0x00000004)

Table 39: TX_SS (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R13334(R0x00003416)	mipi_t_clk_trail_clk_zero	0000 0000 0000 0000 0000 dddd 00dd dddd	1289 (0x00000509)
R15496(R0x00003C88)	tn_crop_hgt	0000 0000 0000 0000 0000 000d dddd dddd	480 (0x000001E0)
R15498(R0x00003C8A)	tn_crop_tl	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15494(R0x00003C86)	tn_crop_wid	0000 0000 0000 0000 0000 00dd dddd dddd	640 (0x00000280)
R15488(R0x00003C80)	tn_ctrl	0000 0000 0000 0000 000d dddd dd0d dddd	533 (0x00000215)
R15502(R0x00003C8E)	tn_end_code	0000 0000 0000 0000 dddd dddd dddd dddd	65471 (0x0000FFBF)
R15490(R0x00003C82)	tn_scalar_x_ratio	0000 0000 0000 0000 0000 dddd dddd dddd	506 (0x000001FA)
R15492(R0x00003C84)	tn_scalar_y_ratio	0000 0000 0000 0000 0000 dddd dddd dddd	506 (0x000001FA)
R15504(R0x00003C90)	tn_scale_sharp_ctrl	0000 0000 0000 0000 00dd dddd dddd dddd	4616 (0x00001208)
R15500(R0x00003C8C)	tn_start_code	0000 0000 0000 0000 dddd dddd dddd dddd	65470 (0x0000FFBE)
R15540(R0x00003CB4)	txc_line_ccir_eof01_code	0000 0000 0000 0000 dddd dddd dddd dddd	65280 (0x0000FF00)
R15542(R0x00003CB6)	txc_line_ccir_eof23_code	0000 0000 0000 0000 dddd dddd dddd dddd	182 (0x000000B6)
R15548(R0x00003CBC)	txc_line_ccir_eol01_code	0000 0000 0000 0000 dddd dddd dddd dddd	65280 (0x0000FF00)
R15550(R0x00003CBE)	txc_line_ccir_eol23_code	0000 0000 0000 0000 dddd dddd dddd dddd	157 (0x0000009D)
R15536(R0x00003CB0)	txc_line_ccir_sof01_code	0000 0000 0000 0000 dddd dddd dddd dddd	65280 (0x0000FF00)
R15538(R0x00003CB2)	txc_line_ccir_sof23_code	0000 0000 0000 0000 dddd dddd dddd dddd	171 (0x000000AB)
R15544(R0x00003CB8)	txc_line_ccir_sol01_code	0000 0000 0000 0000 dddd dddd dddd dddd	65280 (0x0000FF00)
R15546(R0x00003CBA)	txc_line_ccir_sol23_code	0000 0000 0000 0000 dddd dddd dddd dddd	128 (0x00000080)
R13318(R0x00003406)	txc_mipi_data_format	0000 0000 0000 0000 00dd dddd 00dd dddd	12336 (0x00003030)
R13320(R0x00003408)	txc_mipi_line_byte_cnt	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R15522(R0x00003CA2)	txc_parameters	0000 0000 0000 0000 0000 0000 dddd dddd	7 (0x00000007)
R15524(R0x00003CA4)	txc_po_pclk1_config	0000 0000 0000 0000 0000 0000 ddd0 dddd	1 (0x00000001)
R15526(R0x00003CA6)	txc_po_pclk2_config	0000 0000 0000 0000 0000 0000 ddd0 dddd	1 (0x00000001)
R15528(R0x00003CA8)	txc_po_pclk3_config	0000 0000 0000 0000 0000 0000 ddd0 dddd	1 (0x00000001)

Table 39: TX_SS (sorted by name) (continued)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R15530(R0x00003CAA)	txc_timing	0000 0000 0000 0000 dddd dddd dddd dddd	1285 (0x00000505)
R15554(R0x00003CC2)	txss_clr_status	0000 0000 0000 0000 0000 0000 0000 000d	0 (0x00000000)
R13322(R0x0000340A)	txss_mipi_control_addl	0000 0000 0000 0000 0000 0000 000d dddd	10 (0x0000000A)
R15520(R0x00003CA0)	txss_parameters	0000 0000 0000 0000 0000 0000 0ddd d00d	64 (0x00000040)
R15552(R0x00003CC0)	txss_status	0000 0000 0000 0000 0000 0000 0000 0?0?	0 (0x00000000)

Table 40: OTPM (sorted by name)

Register Dec (Hex)	Name	Data Format (Binary)	Default Value Dec (Hex)
R14340(R0x00003804)	otpm_addr	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R14354(R0x00003812)	otpm_cfg	0000 0000 0000 0000 0ddd dddd dddd dddd	8540 (0x0000215C)
R14364(R0x0000381C)	otpm_cfg2	0000 0000 0000 0000 0000 00dd 00dd dddd	0 (0x00000000)
R14338(R0x00003802)	otpm_control	0000 0000 0000 0000 0000 0000 0000 dddd	4 (0x00000004)
R14346(R0x0000380A)	otpm_data_pgm_extra	0000 0000 0000 0000 0000 0000 dddd dddd	0 (0x00000000)
R14344(R0x00003808)	otpm_data_pgm_h	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14342(R0x00003806)	otpm_data_pgm_l	0000 0000 0000 0000 dddd dddd dddd dddd	0 (0x00000000)
R14352(R0x00003810)	otpm_data_read_extra	0000 0000 0000 0000 0000 0000 ???? ????	0 (0x00000000)
R14350(R0x0000380E)	otpm_data_read_h	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R14348(R0x0000380C)	otpm_data_read_l	0000 0000 0000 0000 ???? ???? ???? ????	0 (0x00000000)
R14362(R0x0000381A)	otpm_expr	0000 0000 0000 0000 0000 0000 0000 0ddd	0 (0x00000000)
R14336(R0x00003800)	otpm_status	0000 0000 0000 0000 0000 000? ???? ????	0 (0x00000000)
R14356(R0x00003814)	otpm_tcfg_01	0000 0000 0000 0000 dddd dddd dddd dddd	1795 (0x00000703)
R14358(R0x00003816)	otpm_tcfg_23	0000 0000 0000 0000 dddd dddd dddd dddd	1799 (0x00000707)
R14360(R0x00003818)	otpm_tcfg_4b	0000 0000 0000 0000 dddd dddd dddd dddd	17415 (0x00004407)

Table 41: GPIO_SS Registers

Register Dec(Hex)	Bits	Default	Name
1536 R0x00000600	31:0	0x0000	second_snsr (R/W)
	31:2	X	Reserved
	1	0x0000	second_snsr_standby Secondary sensor standby 0: disable 1: enable
	0	0x0000	second_snsr_rst Secondary sensor rst 0: disable 1: enable
	To control the secondary sensor		
1538 R0x00000602	31:0	0x0000	second_snsr_oe (R/W)
	31:3	X	Reserved
	2	0x0000	second_snsr_clk_oe OE for second sensor clk from aopd_ss 0: input 1: output
	1	0x0000	second_snsr_standby_oe OE for second_snsr_ctl_standby 0: input 1: output
	0	0x0000	second_snsr_rst_oe OE for second_snsr_ctl_rst 0: input 1: output
	The OE for second sensor controls		
1544 R0x00000608	31:0	0x0000	vgpio_int_ctl (R/W)
	31:16	X	Reserved
	15:8	0x0000	vgpio_int_edge Interrupt edges for vgpio 0: falling edge 1: rising edge Legal values: [0, 255].
	7:0	0x0000	vgpio_int_en Enable interrupts for vgpio Legal values: [0, 255].
	Interrupt control for VGPIO		
1546 R0x0000060A	31:0	0x0000	vgpio_int_status_clr (R/W)
	31:16	X	Reserved
	15:8	0x0000	vgpio_int_clear Clears int if set, auto clearing Legal values: [0,255].
	7:0	RO	vgpio_int_status Interrupt status for VGPIO Read-only.
	Interrupt status for VGPIO		

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1548 R0x0000060C	31:0	0x0000	vgpio_data_from_icb (R/W)
	31:16	X	Reserved
	15:8	0x0000	vgpio_data_icb_sel To select ICB to drive vgpio output 0: Select other source to drive vgpio output 1: Select vgpio_data_icb[7:0] to drive vgpio output Legal values: [0, 255].
	7:0	0x0000	vgpio_data_icb ICB drives vgpio out Legal values: [0, 255].
	ICB drives vgpio out		
1550 R0x0000060E	31:0	0x0000	vgpio_dir (R/W)
	OE for vgpio 0: input 1: output Legal values: [0, 255].		
1552 R0x00000610	31:0	0x0000	vgpio_data_status (RO)
	Monitor vgpio data status both for IN and OUT Read-only. Legal values: [0, 255].		
1554 R0x00000612	31:0	0x0000	vgpio_data_alt_sel (R/W)
	31:16	X	Reserved
	15	0x0000	vgpio_data_jtag_sel 0: Select other source for vgpio_out[6:2] 1: Select vgpio[6:2] to be used by JTAG Legal values: [0,1].
	14	X	Reserved
	13	0x0000	vgpio_data_b1_sel_uart_txd 0: Select other source for vgpio_out[1] 1: Select vgpio[1] to be used by UART TXD
	12	0x0000	vgpio_data_b0_sel_uart_rxd 0: Select other source for vgpio_out[0] 1: Select vgpio[0] to be used by UART RXD
	11:8	X	Reserved
	7:0	0x0000	vgpio_data_wg_sel 0: Select other source for vgpio output 1: Select WG to drive vgpio Legal values: [0, 255].
	Selects data source for VGPIO pins		
1556 R0x00000614	31:0	0x0001	second_scl_sda_pd (R/W)
	Power down s_scl and s_sda 0: Power up 1: Power down		
1558 R0x00000616	31:0	0x0000	vgpio_data_rx_tx_sel (R/W)
	To select rx_mipi to drive vgpio output 0: Select other source to drive vgpio output 1: Select rx_mipi to drive vgpio output Legal values: [0, 4095].		

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1568 R0x00000620	31:0	0x0000	vcm_control (R/W)
	31:16	X	Reserved
	15	0x0000	vcm_enable Enable the VCM output
	14	X	Reserved
	13:6	0x0000	vcm_data Legal values: [0, 255].
	5:4	X	Reserved
	3:0	0x0000	vcm_slew Legal values: [0, 15].
	Voice Coil Module control		
1790 R0x000006FE	31:0	0x0000	wg_gr_trigger_sel (R/W)
	Selects which WG supplies wg_gr_trigger to sensor.		
1792 R0x00000700	31:0	0x0000	ch0_cfg (R/W)
	31:13	X	Reserved
	12:9	RO	wg0_irq_status Read only. If IRQ from this wg channel was triggered will show reason(s) for IRQ Possible reasons are numbered in description of bits 7:4 of this register Unused bits should be present as reserved at the moment Read-only.
	8	0x0000	wg0_irq_clear
	7:4	0x0000	wg0_irq_enable Mask for IRQ generation. 4 = wave generation finished, 5 = repetition counter is decremented, 6 = 0->1 transition of the output, 7 = 1->0 transition at the output.
	3	X	Reserved
	2	0x0000	wg0_suspend
	1	0x0000	wg0_stop Stops wg and reloads all counters
	0	0x0000	wg0_manual_start manual start signal

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1794 R0x00000702	31:0	0x0000	wg0_start (R/W)
	31:8	X	Reserved
	7:6	0x0000	wg0_start_type 0 = counting completed (WG only) 1 = single waveform completed (WG only) 2 = falling edge on selected signal 3 = rising edge on selected signal
	5:4	0x0000	wg0_input_type 0 = manual start condition 1 = WG channel number 0..7 specified in 3:0 2 = GPIO number 0..7 specified in 3:0 3 = special signal specified in 3:0
	3:0	0x0000	wg0_start_source GPIO number 0..7 or WG number 0..7 or special signal as specified below: 0 = sensor FV 1 = sensor STROBE 2 = sensor input TRIGGER signal 3 = sensor output FLASH signal 4 = IFP FV
1796 R0x00000704	31:0	0x0000	wg0_cfg_idle (R/W)
	31:9	X	Reserved
	8	0x0000	wg0_idle_output_control use value from bit 7 for the output if 1 else simply invert output upon entering idle state Idle state has # = 5;
	7	0x0000	wg0_idle_output state of the output when wg is in idle state
	6:3	X	Reserved
	2:0	0x0000	wg0_start_state State # where wg goes upon receiving start condition
1798 R0x00000706	31:0	0x0000	wg0_cfg0 (R/W)
	31:9	X	Reserved
	8	0x0000	wg0_cfg0_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg0_cfg0_output state assigned to the output when wg0 is in state 0, if bit 8 is set to 1.
	6	0x0000	wg0_cfg0_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg0_cfg0_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg0_cfg0_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1800 R0x00000708	31:0	0x0000	wg0_cfg1 (R/W)
	31:9	X	Reserved
	8	0x0000	wg0_cfg1_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg0_cfg1_output state assigned to the output when wg0 is in state 1, if bit 8 is set to 1.
	6	0x0000	wg0_cfg1_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg0_cfg1_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg0_cfg1_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1802 R0x0000070A	31:0	0x0000	wg0_cfg2 (R/W)
	31:9	X	Reserved
	8	0x0000	wg0_cfg2_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg0_cfg2_output state assigned to the output when wg0 is in state 2, if bit 8 is set to 1.
	6	0x0000	wg0_cfg2_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg0_cfg2_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg0_cfg2_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1804 R0x0000070C	31:0	0x0000	wg0_cfg3 (R/W)
	31:9	X	Reserved
	8	0x0000	wg0_cfg3_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg0_cfg3_output state assigned to the output when wg0 is in state 3, if bit 8 is set to 1.
	6	0x0000	wg0_cfg3_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg0_cfg3_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg0_cfg3_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1806 R0x0000070E	31:0	0x0000	wg0_cfg4 (R/W)
	31:9	X	Reserved
	8	0x0000	wg0_cfg4_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg0_cfg4_output state assigned to the output when wg0 is in state 4, if bit 8 is set to 1.
	6	0x0000	wg0_cfg4_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg0_cfg4_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg0_cfg4_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1808 R0x00000710	31:0	0x0000	wg0_t0 (R/W)
	Number of prescaled delay intervals for the 0th state		
1810 R0x00000712	31:0	0x0000	wg0_t1 (R/W)
	Number of prescaled delay intervals for the 1th state		
1812 R0x00000714	31:0	0x0000	wg0_t2 (R/W)
	Number of prescaled delay intervals for the 2th state		
1814 R0x00000716	31:0	0x0000	wg0_t3 (R/W)
	Number of prescaled delay intervals for the 3th state		
1816 R0x00000718	31:0	0x0000	wg0_t4 (R/W)
	Number of prescaled delay intervals for the 4th state		
1818 R0x0000071A	31:0	0x0000	wg0_prescale (R/W)
	Clock frequency for wg can be divided by 2^[3:0] to slow down wg operation.		
1820 R0x0000071C	31:0	0x0000	wg0_n (R/W)
	Number of repetitions of the waveform. When this counter is 0 upon delay completion wg goes into state specified in bits 5:3 of the configuration register rather that 2:0 (for the case when this counter is not 0). Decrement of this counter occurs upon leaving any state in which configuration bit 6 is set.		
1822 R0x0000071E	31:0	0x0000	wg0_xor (R/W)
	Selects 8 WG bits to WG0, they are combined by XOR.		

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1824 R0x00000720	31:0	0x0000	ch1_cfg (R/W)
	31:13	X	Reserved
	12:9	RO	wg1_irq_status Read only. If IRQ from this wg channel was triggered will show reason(s) for IRQ Possible reasons are numbered in description of bits 7:4 of this register Unused bits should be present as reserved at the moment Read-only.
	8	0x0000	wg1_irq_clear
	7:4	0x0000	wg1_irq_enable Mask for IRQ generation. 4 = wave generation finished, 5 = repetition counter is decremented, 6 = 0->1 transition of the output, 7 = 1->0 transition at the output.
	3	X	Reserved
	2	0x0000	wg1_suspend
	1	0x0000	wg1_stop Stops wg and reloads all counters
	0	0x0000	wg1_manual_start manual start signal
1826 R0x00000722	31:0	0x0000	wg1_start (R/W)
	31:8	X	Reserved
	7:6	0x0000	wg1_start_type 0=counting completed (WG only) 1=single waveform completed (WG only) 2=falling edge on selected signal 3=rising edge on selected signal
	5:4	0x0000	wg1_input_type 0=manual start condition 1=WG channel number 0..7 specified in 3:0 2=GPIO number 0..7 specified in 3:0 3=special signal specified in 3:0
	3:0	0x0000	wg1_start_source GPIO number 0..7 or WG number 0..7 or special signal as specified below: 0 sensor FV 1 sensor STROBE 2 sensor input TRIGGER signal 3 sensor output FLASH signal 4 IFP FV

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1828 R0x00000724	31:0	0x0000	wg1_cfg_idle (R/W)
	31:9	X	Reserved
	8	0x0000	wg1_idle_output_control use value from bit 7 for the output if 1 else simply invert output upon entering idle state Idle state has # = 5;
	7	0x0000	wg1_idle_output state of the output when wg is in idle state
	6:3	X	Reserved
	2:0	0x0000	wg1_start_state State # where wg goes upon receiving start condition
1830 R0x00000726	31:0	0x0000	wg1_cfg0 (R/W)
	31:9	X	Reserved
	8	0x0000	wg1_cfg0_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg1_cfg0_output state assigned to the output when wg1 is in state 0, if bit 8 is set to 1.
	6	0x0000	wg1_cfg0_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg1_cfg0_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg1_cfg0_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1832 R0x00000728	31:0	0x0000	wg1_cfg1 (R/W)
	31:9	X	Reserved
	8	0x0000	wg1_cfg1_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg1_cfg1_output state assigned to the output when wg1 is in state 1, if bit 8 is set to 1.
	6	0x0000	wg1_cfg1_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg1_cfg1_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg1_cfg1_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1834 R0x0000072A	31:0	0x0000	wg1_cfg2 (R/W)
	31:9	X	Reserved
	8	0x0000	wg1_cfg2_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg1_cfg2_output state assigned to the output when wg1 is in state 2, if bit 8 is set to 1.
	6	0x0000	wg1_cfg2_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg1_cfg2_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg1_cfg2_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1836 R0x0000072C	31:0	0x0000	wg1_cfg3 (R/W)
	31:9	X	Reserved
	8	0x0000	wg1_cfg3_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg1_cfg3_output state assigned to the output when wg1 is in state 3, if bit 8 is set to 1.
	6	0x0000	wg1_cfg3_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg1_cfg3_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg1_cfg3_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1838 R0x0000072E	31:0	0x0000	wg1_cfg4 (R/W)
	31:9	X	Reserved
	8	0x0000	wg1_cfg4_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg1_cfg4_output state assigned to the output when wg1 is in state 4, if bit 8 is set to 1.
	6	0x0000	wg1_cfg4_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg1_cfg4_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg1_cfg4_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1840 R0x00000730	31:0	0x0000	wg1_t0 (R/W)
	Number of prescaled delay intervals for the 0th state		
1842 R0x00000732	31:0	0x0000	wg1_t1 (R/W)
	Number of prescaled delay intervals for the 1th state		
1844 R0x00000734	31:0	0x0000	wg1_t2 (R/W)
	Number of prescaled delay intervals for the 2th state		
1846 R0x00000736	31:0	0x0000	wg1_t3 (R/W)
	Number of prescaled delay intervals for the 3th state		
1848 R0x00000738	31:0	0x0000	wg1_t4 (R/W)
	Number of prescaled delay intervals for the 4th state		
1850 R0x0000073A	31:0	0x0000	wg1_prescale (R/W)
	Clock frequency for wg can be divided by 2^[3:0] to slow down wg operation.		
1852 R0x0000073C	31:0	0x0000	wg1_n (R/W)
	Number of repetitions of the waveform. When this counter is 0 upon delay completion wg goes into state specified in bits 5:3 of the configuration register rather that 2:0 (for the case when this counter is not 0). Decrement of this counter occurs upon leaving any state in which configuration bit 6 is set.		
1854 R0x0000073E	31:0	0x0000	wg1_xor (R/W)
	Selects 8 WG bits to WG1, they are combined by XOR.		
1856 R0x00000740	31:0	0x0000	ch2_cfg (R/W)
	31:13	X	Reserved
	12:9	RO	wg2_irq_status Read only. If IRQ from this wg channel was triggered will show reason(s) for IRQ Possible reasons are numbered in description of bits 7:4 of this register Unused bits should be present as reserved at the moment Read-only.
	8	0x0000	wg2_irq_clear
	7:4	0x0000	wg2_irq_enable Mask for IRQ generation. 4 = wave generation finished, 5 = repetition counter is decremented, 6 = 0->1 transition of the output, 7 = 1->0 transition at the output.
	3	X	Reserved
	2	0x0000	wg2_suspend
	1	0x0000	wg2_stop Stops wg and reloads all counters
	0	0x0000	wg2_manual_start manual start signal

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1858 R0x00000742	31:0	0x0000	wg2_start (R/W)
	31:8	X	Reserved
	7:6	0x0000	wg2_start_type 0=counting completed (WG only) 1=single waveform completed (WG only) 2=falling edge on selected signal 3=rising edge on selected signal
	5:4	0x0000	wg2_input_type 0=manual start condition 1=WG channel number 0..7 specified in 3:0 2=GPIO number 0..7 specified in 3:0 3=special signal specified in 3:0
	3:0	0x0000	wg2_start_source GPIO number 0..7 or WG number 0..7 or special signal as specified below: 0 sensor FV 1 sensor STROBE 2 sensor input TRIGGER signal 3 sensor output FLASH signal 4 IFP FV
1860 R0x00000744	31:0	0x0000	wg2_cfg_idle (R/W)
	31:9	X	Reserved
	8	0x0000	wg2_idle_output_control use value from bit 7 for the output if 1 else simply invert output upon entering idle state Idle state has # = 5;
	7	0x0000	wg2_idle_output state of the output when wg is in idle state
	6:3	X	Reserved
	2:0	0x0000	wg2_start_state State # where wg goes upon receiving start condition
1862 R0x00000746	31:0	0x0000	wg2_cfg0 (R/W)
	31:9	X	Reserved
	8	0x0000	wg2_cfg0_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg2_cfg0_output state assigned to the output when wg2 is in state 0, if bit 8 is set to 1.
	6	0x0000	wg2_cfg0_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg2_cfg0_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg2_cfg0_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1864 R0x00000748	31:0	0x0000	wg2_cfg1 (R/W)
	31:9	X	Reserved
	8	0x0000	wg2_cfg1_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg2_cfg1_output state assigned to the output when wg2 is in state 1, if bit 8 is set to 1.
	6	0x0000	wg2_cfg1_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg2_cfg1_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg2_cfg1_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1866 R0x0000074A	31:0	0x0000	wg2_cfg2 (R/W)
	31:9	X	Reserved
	8	0x0000	wg2_cfg2_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg2_cfg2_output state assigned to the output when wg2 is in state 2, if bit 8 is set to 1.
	6	0x0000	wg2_cfg2_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg2_cfg2_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg2_cfg2_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1868 R0x0000074C	31:0	0x0000	wg2_cfg3 (R/W)
	31:9	X	Reserved
	8	0x0000	wg2_cfg3_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg2_cfg3_output state assigned to the output when wg2 is in state 3, if bit 8 is set to 1.
	6	0x0000	wg2_cfg3_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg2_cfg3_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg2_cfg3_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1870 R0x0000074E	31:0	0x0000	wg2_cfg4 (R/W)
	31:9	X	Reserved
	8	0x0000	wg2_cfg4_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg2_cfg4_output state assigned to the output when wg2 is in state 4, if bit 8 is set to 1.
	6	0x0000	wg2_cfg4_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg2_cfg4_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg2_cfg4_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1872 R0x00000750	31:0	0x0000	wg2_t0 (R/W)
	Number of prescaled delay intervals for the 0th state		
1874 R0x00000752	31:0	0x0000	wg2_t1 (R/W)
	Number of prescaled delay intervals for the 1th state		
1876 R0x00000754	31:0	0x0000	wg2_t2 (R/W)
	Number of prescaled delay intervals for the 2th state		
1878 R0x00000756	31:0	0x0000	wg2_t3 (R/W)
	Number of prescaled delay intervals for the 3th state		
1880 R0x00000758	31:0	0x0000	wg2_t4 (R/W)
	Number of prescaled delay intervals for the 4th state		
1882 R0x0000075A	31:0	0x0000	wg2_prescale (R/W)
	Clock frequency for wg can be divided by 2^[3:0] to slow down wg operation.		
1884 R0x0000075C	31:0	0x0000	wg2_n (R/W)
	Number of repetitions of the waveform. When this counter is 0 upon delay completion wg goes into state specified in bits 5:3 of the configuration register rather that 2:0 (for the case when this counter is not 0). Decrement of this counter occurs upon leaving any state in which configuration bit 6 is set.		
1886 R0x0000075E	31:0	0x0000	wg2_xor (R/W)
	Selects 8 WG bits to WG2, they are combined by XOR.		

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1888 R0x00000760	31:0	0x0000	ch3_cfg (R/W)
	31:13	X	Reserved
	12:9	RO	wg3_irq_status Read only. If IRQ from this wg channel was triggered will show reason(s) for IRQ Possible reasons are numbered in description of bits 7:4 of this register Unused bits should be present as reserved at the moment Read-only.
	8	0x0000	wg3_irq_clear
	7:4	0x0000	wg3_irq_enable Mask for IRQ generation. 4 = wave generation finished, 5 = repetition counter is decremented, 6 = 0->1 transition of the output, 7 = 1->0 transition at the output.
	3	X	Reserved
	2	0x0000	wg3_suspend
	1	0x0000	wg3_stop Stops wg and reloads all counters
	0	0x0000	wg3_manual_start manual start signal
1890 R0x00000762	31:0	0x0000	wg3_start (R/W)
	31:8	X	Reserved
	7:6	0x0000	wg3_start_type 0=counting completed (WG only) 1=single waveform completed (WG only) 2=falling edge on selected signal 3=rising edge on selected signal
	5:4	0x0000	wg3_input_type 0=manual start condition 1=WG channel number 0..7 specified in 3:0 2=GPIO number 0..7 specified in 3:0 3=special signal specified in 3:0
	3:0	0x0000	wg3_start_source GPIO number 0..7 or WG number 0..7 or special signal as specified below: 0 sensor FV 1 sensor STROBE 2 sensor input TRIGGER signal 3 sensor output FLASH signal 4 IFP FV

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1892 R0x00000764	31:0	0x0000	wg3_cfg_idle (R/W)
	31:9	X	Reserved
	8	0x0000	wg3_idle_output_control use value from bit 7 for the output if 1 else simply invert output upon entering idle state Idle state has # = 5;
	7	0x0000	wg3_idle_output state of the output when wg is in idle state
	6:3	X	Reserved
	2:0	0x0000	wg3_start_state State # where wg goes upon receiving start condition
1894 R0x00000766	31:0	0x0000	wg3_cfg0 (R/W)
	31:9	X	Reserved
	8	0x0000	wg3_cfg0_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg3_cfg0_output state assigned to the output when wg3 is in state 0, if bit 8 is set to 1.
	6	0x0000	wg3_cfg0_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg3_cfg0_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg3_cfg0_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1896 R0x00000768	31:0	0x0000	wg3_cfg1 (R/W)
	31:9	X	Reserved
	8	0x0000	wg3_cfg1_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg3_cfg1_output state assigned to the output when wg3 is in state 1, if bit 8 is set to 1.
	6	0x0000	wg3_cfg1_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg3_cfg1_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg3_cfg1_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1898 R0x0000076A	31:0	0x0000	wg3_cfg2 (R/W)
	31:9	X	Reserved
	8	0x0000	wg3_cfg2_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg3_cfg2_output state assigned to the output when wg3 is in state 2, if bit 8 is set to 1.
	6	0x0000	wg3_cfg2_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg3_cfg2_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg3_cfg2_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1900 R0x0000076C	31:0	0x0000	wg3_cfg3 (R/W)
	31:9	X	Reserved
	8	0x0000	wg3_cfg3_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg3_cfg3_output state assigned to the output when wg3 is in state 3, if bit 8 is set to 1.
	6	0x0000	wg3_cfg3_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg3_cfg3_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg3_cfg3_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1902 R0x0000076E	31:0	0x0000	wg3_cfg4 (R/W)
	31:9	X	Reserved
	8	0x0000	wg3_cfg4_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg3_cfg4_output state assigned to the output when wg3 is in state 4, if bit 8 is set to 1.
	6	0x0000	wg3_cfg4_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg3_cfg4_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg3_cfg4_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1904 R0x00000770	31:0	0x0000	wg3_t0 (R/W)
	Number of prescaled delay intervals for the 0th state		
1906 R0x00000772	31:0	0x0000	wg3_t1 (R/W)
	Number of prescaled delay intervals for the 1th state		
1908 R0x00000774	31:0	0x0000	wg3_t2 (R/W)
	Number of prescaled delay intervals for the 2th state		
1910 R0x00000776	31:0	0x0000	wg3_t3 (R/W)
	Number of prescaled delay intervals for the 3th state		
1912 R0x00000778	31:0	0x0000	wg3_t4 (R/W)
	Number of prescaled delay intervals for the 4th state		
1914 R0x0000077A	31:0	0x0000	wg3_prescale (R/W)
	Clock frequency for wg can be divided by 2^[3:0] to slow down wg operation.		
1916 R0x0000077C	31:0	0x0000	wg3_n (R/W)
	Number of repetitions of the waveform. When this counter is 0 upon delay completion wg goes into state specified in bits 5:3 of the configuration register rather that 2:0 (for the case when this counter is not 0). Decrement of this counter occurs upon leaving any state in which configuration bit 6 is set.		
1918 R0x0000077E	31:0	0x0000	wg3_xor (R/W)
	Selects 8 WG bits to WG3, they are combined by XOR.		
1920 R0x00000780	31:0	0x0000	ch4_cfg (R/W)
	31:13	X	Reserved
	12:9	RO	wg4_irq_status Read only. If IRQ from this wg channel was triggered will show reason(s) for IRQ Possible reasons are numbered in description of bits 7:4 of this register Unused bits should be present as reserved at the moment Read-only.
	8	0x0000	wg4_irq_clear
	7:4	0x0000	wg4_irq_enable Mask for IRQ generation. 4 = wave generation finished, 5 = repetition counter is decremented, 6 = 0->1 transition of the output, 7 = 1->0 transition at the output.
	3	X	Reserved
	2	0x0000	wg4_suspend
	1	0x0000	wg4_stop Stops wg and reloads all counters
	0	0x0000	wg4_manual_start manual start signal

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1922 R0x00000782	31:0	0x0000	wg4_start (R/W)
	31:8	X	Reserved
	7:6	0x0000	wg4_start_type 0=counting completed (WG only) 1=single waveform completed (WG only) 2=falling edge on selected signal 3=rising edge on selected signal
	5:4	0x0000	wg4_input_type 0=manual start condition 1=WG channel number 0..7 specified in 3:0 2=GPIO number 0..7 specified in 3:0 3=special signal specified in 3:0
	3:0	0x0000	wg4_start_source GPIO number 0..7 or WG number 0..7 or special signal as specified below: 0 sensor FV 1 sensor STROBE 2 sensor input TRIGGER signal 3 sensor output FLASH signal 4 IFP FV
1924 R0x00000784	31:0	0x0000	wg4_cfg_idle (R/W)
	31:9	X	Reserved
	8	0x0000	wg4_idle_output_control use value from bit 7 for the output if 1 else simply invert output upon entering idle state Idle state has # = 5;
	7	0x0000	wg4_idle_output state of the output when wg is in idle state
	6:3	X	Reserved
	2:0	0x0000	wg4_start_state State # where wg goes upon receiving start condition
1926 R0x00000786	31:0	0x0000	wg4_cfg0 (R/W)
	31:9	X	Reserved
	8	0x0000	wg4_cfg0_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg4_cfg0_output state assigned to the output when wg4 is in state 0, if bit 8 is set to 1.
	6	0x0000	wg4_cfg0_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg4_cfg0_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg4_cfg0_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1928 R0x00000788	31:0	0x0000	wg4_cfg1 (R/W)
	31:9	X	Reserved
	8	0x0000	wg4_cfg1_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg4_cfg1_output state assigned to the output when wg4 is in state 1, if bit 8 is set to 1.
	6	0x0000	wg4_cfg1_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg4_cfg1_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg4_cfg1_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1930 R0x0000078A	31:0	0x0000	wg4_cfg2 (R/W)
	31:9	X	Reserved
	8	0x0000	wg4_cfg2_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg4_cfg2_output state assigned to the output when wg4 is in state 2, if bit 8 is set to 1.
	6	0x0000	wg4_cfg2_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg4_cfg2_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg4_cfg2_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1932 R0x0000078C	31:0	0x0000	wg4_cfg3 (R/W)
	31:9	X	Reserved
	8	0x0000	wg4_cfg3_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg4_cfg3_output state assigned to the output when wg4 is in state 3, if bit 8 is set to 1.
	6	0x0000	wg4_cfg3_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg4_cfg3_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg4_cfg3_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1934 R0x0000078E	31:0	0x0000	wg4_cfg4 (R/W)
	31:9	X	Reserved
	8	0x0000	wg4_cfg4_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg4_cfg4_output state assigned to the output when wg4 is in state 4, if bit 8 is set to 1.
	6	0x0000	wg4_cfg4_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg4_cfg4_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg4_cfg4_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1936 R0x00000790	31:0	0x0000	wg4_t0 (R/W)
	Number of prescaled delay intervals for the 0th state		
1938 R0x00000792	31:0	0x0000	wg4_t1 (R/W)
	Number of prescaled delay intervals for the 1th state		
1940 R0x00000794	31:0	0x0000	wg4_t2 (R/W)
	Number of prescaled delay intervals for the 2th state		
1942 R0x00000796	31:0	0x0000	wg4_t3 (R/W)
	Number of prescaled delay intervals for the 3th state		
1944 R0x00000798	31:0	0x0000	wg4_t4 (R/W)
	Number of prescaled delay intervals for the 4th state		
1946 R0x0000079A	31:0	0x0000	wg4_prescale (R/W)
	Clock frequency for wg can be divided by 2^[3:0] to slow down wg operation.		
1948 R0x0000079C	31:0	0x0000	wg4_n (R/W)
	Number of repetitions of the waveform. When this counter is 0 upon delay completion wg goes into state specified in bits 5:3 of the configuration register rather that 2:0 (for the case when this counter is not 0). Decrement of this counter occurs upon leaving any state in which configuration bit 6 is set.		
1950 R0x0000079E	31:0	0x0000	wg4_xor (R/W)
	Selects 8 WG bits to WG4, they are combined by XOR.		

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1952 R0x000007A0	31:0	0x0000	ch5_cfg (R/W)
	31:13	X	Reserved
	12:9	RO	wg5_irq_status Read only. If IRQ from this wg channel was triggered will show reason(s) for IRQ Possible reasons are numbered in description of bits 7:4 of this register Unused bits should be present as reserved at the moment Read-only.
	8	0x0000	wg5_irq_clear
	7:4	0x0000	wg5_irq_enable Mask for IRQ generation. 4 = wave generation finished, 5 = repetition counter is decremented, 6 = 0->1 transition of the output, 7 = 1->0 transition at the output.
	3	X	Reserved
	2	0x0000	wg5_suspend
	1	0x0000	wg5_stop Stops wg and reloads all counters
	0	0x0000	wg5_manual_start manual start signal
1954 R0x000007A2	31:0	0x0000	wg5_start (R/W)
	31:8	X	Reserved
	7:6	0x0000	wg5_start_type 0=counting completed (WG only) 1=single waveform completed (WG only) 2=falling edge on selected signal 3=rising edge on selected signal
	5:4	0x0000	wg5_input_type 0=manual start condition 1=WG channel number 0..7 specified in 3:0 2=GPIO number 0..7 specified in 3:0 3=special signal specified in 3:0
	3:0	0x0000	wg5_start_source GPIO number 0..7 or WG number 0..7 or special signal as specified below: 0 sensor FV 1 sensor STROBE 2 sensor input TRIGGER signal 3 sensor output FLASH signal 4 IFP FV

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1956 R0x000007A4	31:0	0x0000	wg5_cfg_idle (R/W)
	31:9	X	Reserved
	8	0x0000	wg5_idle_output_control use value from bit 7 for the output if 1 else simply invert output upon entering idle state Idle state has # = 5;
	7	0x0000	wg5_idle_output state of the output when wg is in idle state
	6:3	X	Reserved
	2:0	0x0000	wg5_start_state State # where wg goes upon receiving start condition
1958 R0x000007A6	31:0	0x0000	wg5_cfg0 (R/W)
	31:9	X	Reserved
	8	0x0000	wg5_cfg0_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg5_cfg0_output state assigned to the output when wg5 is in state 0, if bit 8 is set to 1.
	6	0x0000	wg5_cfg0_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg5_cfg0_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg5_cfg0_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1960 R0x000007A8	31:0	0x0000	wg5_cfg1 (R/W)
	31:9	X	Reserved
	8	0x0000	wg5_cfg1_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg5_cfg1_output state assigned to the output when wg5 is in state 1, if bit 8 is set to 1.
	6	0x0000	wg5_cfg1_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg5_cfg1_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg5_cfg1_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1962 R0x000007AA	31:0	0x0000	wg5_cfg2 (R/W)
	31:9	X	Reserved
	8	0x0000	wg5_cfg2_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg5_cfg2_output state assigned to the output when wg5 is in state 2, if bit 8 is set to 1.
	6	0x0000	wg5_cfg2_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg5_cfg2_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg5_cfg2_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1964 R0x000007AC	31:0	0x0000	wg5_cfg3 (R/W)
	31:9	X	Reserved
	8	0x0000	wg5_cfg3_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg5_cfg3_output state assigned to the output when wg5 is in state 3, if bit 8 is set to 1.
	6	0x0000	wg5_cfg3_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg5_cfg3_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg5_cfg3_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1966 R0x000007AE	31:0	0x0000	wg5_cfg4 (R/W)
	31:9	X	Reserved
	8	0x0000	wg5_cfg4_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg5_cfg4_output state assigned to the output when wg5 is in state 4, if bit 8 is set to 1.
	6	0x0000	wg5_cfg4_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg5_cfg4_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg5_cfg4_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1968 R0x000007B0	31:0	0x0000	wg5_t0 (R/W)
	Number of prescaled delay intervals for the 0th state		
1970 R0x000007B2	31:0	0x0000	wg5_t1 (R/W)
	Number of prescaled delay intervals for the 1st state		
1972 R0x000007B4	31:0	0x0000	wg5_t2 (R/W)
	Number of prescaled delay intervals for the 2nd state		
1974 R0x000007B6	31:0	0x0000	wg5_t3 (R/W)
	Number of prescaled delay intervals for the 3rd state		
1976 R0x000007B8	31:0	0x0000	wg5_t4 (R/W)
	Number of prescaled delay intervals for the 4th state		
1978 R0x000007BA	31:0	0x0000	wg5_prescale (R/W)
	Clock frequency for wg can be divided by 2^[3:0] to slow down wg operation.		
1980 R0x000007BC	31:0	0x0000	wg5_n (R/W)
	Number of repetitions of the waveform. When this counter is 0 upon delay completion wg goes into state specified in bits 5:3 of the configuration register rather that 2:0 (for the case when this counter is not 0). Decrement of this counter occurs upon leaving any state in which configuration bit 6 is set.		
1982 R0x000007BE	31:0	0x0000	wg5_xor (R/W)
	Selects 8 WG bits to WG5, they are combined by XOR.		
1984 R0x000007C0	31:0	0x0000	ch6_cfg (R/W)
	31:13	X	Reserved
	12:9	RO	wg6_irq_status Read only. If IRQ from this wg channel was triggered will show reason(s) for IRQ Possible reasons are numbered in description of bits 7:4 of this register Unused bits should be present as reserved at the moment Read-only.
	8	0x0000	wg6_irq_clear
	7:4	0x0000	wg6_irq_enable Mask for IRQ generation. 4 = wave generation finished, 5 = repetition counter is decremented, 6 = 0->1 transition of the output, 7 = 1->0 transition at the output.
	3	X	Reserved
	2	0x0000	wg6_suspend
	1	0x0000	wg6_stop Stops wg and reloads all counters
	0	0x0000	wg6_manual_start manual start signal

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1986 R0x000007C2	31:0	0x0000	wg6_start (R/W)
	31:8	X	Reserved
	7:6	0x0000	wg6_start_type 0=counting completed (WG only) 1=single waveform completed (WG only) 2=falling edge on selected signal 3=rising edge on selected signal
	5:4	0x0000	wg6_input_type 0=manual start condition 1=WG channel number 0..7 specified in 3:0 2=GPIO number 0..7 specified in 3:0 3=special signal specified in 3:0
	3:0	0x0000	wg6_start_source GPIO number 0..7 or WG number 0..7 or special signal as specified below: 0 sensor FV 1 sensor STROBE 2 sensor input TRIGGER signal 3 sensor output FLASH signal 4 IFP FV
1988 R0x000007C4	31:0	0x0000	wg6_cfg_idle (R/W)
	31:9	X	Reserved
	8	0x0000	wg6_idle_output_control Use value from bit 7 for the output if 1; else simply invert output upon entering idle state Idle state has # = 5;
	7	0x0000	wg6_idle_output State of the output when wg is in idle state
	6:3	X	Reserved
	2:0	0x0000	wg6_start_state State # where wg goes upon receiving start condition
1990 R0x000007C6	31:0	0x0000	wg6_cfg0 (R/W)
	31:9	X	Reserved
	8	0x0000	wg6_cfg0_output_control Use value from bit 7 for the output if 1; else simply invert output upon leaving 0 state.
	7	0x0000	wg6_cfg0_output state assigned to the output when wg6 is in state 0, if bit 8 is set to 1.
	6	0x0000	wg6_cfg0_dec If 1 repetition counter is decremented when leaving 0 state
	5:3	0x0000	wg6_cfg0_rpt_next State # where wg goes from this state when delay counter = 0 and repetition counter =0
	2:0	0x0000	wg6_cfg0_next State # where wg goes from this state when delay counter = 0 and repetition counter != 0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1992 R0x000007C8	31:0	0x0000	wg6_cfg1 (R/W)
	31:9	X	Reserved
	8	0x0000	wg6_cfg1_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg6_cfg1_output state assigned to the output when wg6 is in state 1, if bit 8 is set to 1.
	6	0x0000	wg6_cfg1_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg6_cfg1_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg6_cfg1_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1994 R0x000007CA	31:0	0x0000	wg6_cfg2 (R/W)
	31:9	X	Reserved
	8	0x0000	wg6_cfg2_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg6_cfg2_output state assigned to the output when wg6 is in state 2, if bit 8 is set to 1.
	6	0x0000	wg6_cfg2_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg6_cfg2_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg6_cfg2_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
1996 R0x000007CC	31:0	0x0000	wg6_cfg3 (R/W)
	31:9	X	Reserved
	8	0x0000	wg6_cfg3_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg6_cfg3_output state assigned to the output when wg6 is in state 3, if bit 8 is set to 1.
	6	0x0000	wg6_cfg3_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg6_cfg3_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg6_cfg3_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
1998 R0x000007CE	31:0	0x0000	wg6_cfg4 (R/W)
	31:9	X	Reserved
	8	0x0000	wg6_cfg4_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg6_cfg4_output state assigned to the output when wg6 is in state 4, if bit 8 is set to 1.
	6	0x0000	wg6_cfg4_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg6_cfg4_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg6_cfg4_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
2000 R0x000007D0	31:0	0x0000	wg6_t0 (R/W)
	Number of prescaled delay intervals for the 0 state		
2002 R0x000007D2	31:0	0x0000	wg6_t1 (R/W)
	Number of prescaled delay intervals for the 1st state		
2004 R0x000007D4	31:0	0x0000	wg6_t2 (R/W)
	Number of prescaled delay intervals for the 2nd state		
2006 R0x000007D6	31:0	0x0000	wg6_t3 (R/W)
	Number of prescaled delay intervals for the 3rd state		
2008 R0x000007D8	31:0	0x0000	wg6_t4 (R/W)
	Number of prescaled delay intervals for the 4th state		
2010 R0x000007DA	31:0	0x0000	wg6_prescale (R/W)
	Clock frequency for wg can be divided by 2^[3:0] to slow down wg operation.		
2012 R0x000007DC	31:0	0x0000	wg6_n (R/W)
	Number of repetitions of the waveform. When this counter is 0 upon delay completion wg goes into state specified in bits 5:3 of the configuration register rather that 2:0 (for the case when this counter is not 0). Decrement of this counter occurs upon leaving any state in which configuration bit 6 is set.		
2014 R0x000007DE	31:0	0x0000	wg6_xor (R/W)
	Selects 8 WG bits to WG6, they are combined by XOR.		

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
2016 R0x000007E0	31:0	0x0000	ch7_cfg (R/W)
	31:13	X	Reserved
	12:9	RO	wg7_irq_status Read only. If IRQ from this wg channel was triggered will show reason(s) for IRQ Possible reasons are numbered in description of bits 7:4 of this register Unused bits should be present as reserved at the moment Read-only.
	8	0x0000	wg7_irq_clear
	7:4	0x0000	wg7_irq_enable Mask for IRQ generation. 4 = wave generation finished, 5 = repetition counter is decremented, 6 = 0->1 transition of the output, 7 = 1->0 transition at the output.
	3	X	Reserved
	2	0x0000	wg7_suspend
	1	0x0000	wg7_stop Stops wg and reloads all counters
	0	0x0000	wg7_manual_start manual start signal
2018 R0x000007E2	31:0	0x0000	wg7_start (R/W)
	31:8	X	Reserved
	7:6	0x0000	wg7_start_type 0=counting completed (WG only) 1=single waveform completed (WG only) 2=falling edge on selected signal 3=rising edge on selected signal
	5:4	0x0000	wg7_input_type 0=manual start condition 1=WG channel number 0..7 specified in 3:0 2=GPIO number 0..7 specified in 3:0 3=special signal specified in 3:0
	3:0	0x0000	wg7_start_source GPIO number 0..7 or WG number 0..7 or special signal as specified below: 0 sensor FV 1 sensor STROBE 2 sensor input TRIGGER signal 3 sensor output FLASH signal 4 IFP FV

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
2020 R0x000007E4	31:0	0x0000	wg7_cfg_idle (R/W)
	31:9	X	Reserved
	8	0x0000	wg7_idle_output_control use value from bit 7 for the output if 1 else simply invert output upon entering idle state Idle state has # = 5;
	7	0x0000	wg7_idle_output state of the output when wg is in idle state
	6:3	X	Reserved
	2:0	0x0000	wg7_start_state State # where wg goes upon receiving start condition
2022 R0x000007E6	31:0	0x0000	wg7_cfg0 (R/W)
	31:9	X	Reserved
	8	0x0000	wg7_cfg0_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg7_cfg0_output state assigned to the output when wg7 is in state 0, if bit 8 is set to 1.
	6	0x0000	wg7_cfg0_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg7_cfg0_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg7_cfg0_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
2024 R0x000007E8	31:0	0x0000	wg7_cfg1 (R/W)
	31:9	X	Reserved
	8	0x0000	wg7_cfg1_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg7_cfg1_output state assigned to the output when wg7 is in state 1, if bit 8 is set to 1.
	6	0x0000	wg7_cfg1_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg7_cfg1_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg7_cfg1_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
2026 R0x000007EA	31:0	0x0000	wg7_cfg2 (R/W)
	31:9	X	Reserved
	8	0x0000	wg7_cfg2_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg7_cfg2_output state assigned to the output when wg7 is in state 2, if bit 8 is set to 1.
	6	0x0000	wg7_cfg2_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg7_cfg2_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg7_cfg2_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
2028 R0x000007EC	31:0	0x0000	wg7_cfg3 (R/W)
	31:9	X	Reserved
	8	0x0000	wg7_cfg3_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg7_cfg3_output state assigned to the output when wg7 is in state 3, if bit 8 is set to 1.
	6	0x0000	wg7_cfg3_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg7_cfg3_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg7_cfg3_next State # where wg goes from this state when delay counter =0 and repetition counter !=0
2030 R0x000007EE	31:0	0x0000	wg7_cfg4 (R/W)
	31:9	X	Reserved
	8	0x0000	wg7_cfg4_output_control use value from bit 7 for the output if 1 else simply invert output upon leaving 0th state.
	7	0x0000	wg7_cfg4_output state assigned to the output when wg7 is in state 4, if bit 8 is set to 1.
	6	0x0000	wg7_cfg4_dec if 1 repetition counter is decremented when leaving 0th state
	5:3	0x0000	wg7_cfg4_rpt_next State # where wg goes from this state when delay counter =0 and repetition counter =0
	2:0	0x0000	wg7_cfg4_next State # where wg goes from this state when delay counter =0 and repetition counter !=0

Table 41: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
2032 R0x000007F0	31:0	0x0000	wg7_t0 (R/W)
	Number of prescaled delay intervals for the 0 state		
2034 R0x000007F2	31:0	0x0000	wg7_t1 (R/W)
	Number of prescaled delay intervals for the 1st state		
2036 R0x000007F4	31:0	0x0000	wg7_t2 (R/W)
	Number of prescaled delay intervals for the 2nd state		
2038 R0x000007F6	31:0	0x0000	wg7_t3 (R/W)
	Number of prescaled delay intervals for the 3rd state		
2040 R0x000007F8	31:0	0x0000	wg7_t4 (R/W)
	Number of prescaled delay intervals for the 4th state		
2042 R0x000007FA	31:0	0x0000	wg7_prescale (R/W)
	Clock frequency for wg can be divided by 2^[3:0] to slow down wg operation.		
2044 R0x000007FC	31:0	0x0000	wg7_n (R/W)
	Number of repetitions of the waveform. When this counter is 0 upon delay completion wg goes into state specified in bits 5:3 of the configuration register rather that 2:0 (for the case when this counter is not 0). Decrement of this counter occurs upon leaving any state in which configuration bit 6 is set.		
2046 R0x000007FE	31:0	0x0000	wg7_xor (R/W)
	Selects 8 WG bits to WG7, they are combined by XOR.		

Table 42: 0: Sensor Core Registers

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12288 R0x00003000	31:0	0x2880	model_id_ (R/W)	N	N
	Model ID. Read-only. Can be made read/write by clearing Reg0x301A-B[3].				
12290 R0x00003002	31:0	0x0010	y_addr_start_ (R/W)	Y	YM
	The first row of visible pixels to be read out (not counting any dark rows that may be read). To move the image window, set this register to the starting Y value.				
12292 R0x00003004	31:0	0x001C	x_addr_start_ (R/W)	Y	N
	The first column of visible pixels to be read out (not counting any dark columns that may be read). To move the image window, set this register to the starting X value.				
12294 R0x00003006	31:0	0x07AF	y_addr_end_ (R/W)	Y	YM
	The last row of visible pixels to be read out.				
12296 R0x00003008	31:0	0x0A43	x_addr_end_ (R/W)	Y	N
	The last column of visible pixels to be read out.				
12298 R0x0000300A	31:0	0x07EF	frame_length_lines_ (R/W)	Y	YM
	The number of complete lines (rows) in the output frame. This includes visible lines and vertical blanking lines.				
12300 R0x0000300C	31:0	0x0F26	line_length_pck_ (R/W)	Y	YM
	The number of pixel clock periods in one line (row) time. This includes visible pixels and horizontal blanking time.				

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12304 R0x00003010	31:0	0x009C	fine_correction (R/W)	N	Y
	Fine integration time correction factor. This is an offset that is applied to the programmed value of fine_integration_time such that the actual integration time matches the integration time equation. This register should not be modified under normal operation, but must be modified when binning is enabled or the internal pixel clock divider (pc_speed[2:0]) is used.				
12306 R0x00003012	31:0	0x0010	coarse_integration_time_ (R/W)	Y	N
	Integration time specified in multiples of line_length_pck_.				
12308 R0x00003014	31:0	0x034A	fine_integration_time_ (R/W)	Y	N
	Integration time specified as a number of pixel clocks.				
12310 R0x00003016	31:0	0x0111	row_speed (R/W)		
	31:11	X	Reserved		
	10:8	0x0001	row_speed_opclk_speed Slows down the output pixel clock frequency relative to the system clock frequency. A programmed value of N gives a output pixel clock period of N system clocks. Only values 1, 2 and 4 are supported. A value of 0 is illegal: it causes the clock to stop.	N	N
	7	X	Reserved		
	6:4	0x0001	row_speed_opclk_delay Reserved	N	N
	3	X	Reserved		
	2:0	0x0001	row_speed_pixclk_speed Slows down the internal pixel clock frequency relative to the system clock frequency. A programmed value of N gives a pixel clock period of N system clocks. Only values 1, 2 and 4 are supported. A value of 0 is illegal: it causes the clock to stop.	Y	YM
12312 R0x00003018	31:0	0x0000	extra_delay (R/W)	Y	N
	Extra blanking inserted between frames. A programmed value of N increases the vertical blanking time by N pixel clock periods. Can be used to get a more exact frame rate. May affect the integration times of parts of the image when the integration time is less than 1 frame.				
12314 R0x0000301A	31:0	0x10D8	reset_register (R/W)		
	31:16	X	Reserved		
	15	0x0000	reset_register_grouped_parameter_hold 0 = Update of many of the registers is synchronized to frame start. 1 = Inhibit register updates; register changes will remain pending until this bit is returned to 0. When this bit is returned to 0, all pending register updates will be made on the next frame start.	N	N
	14	0x0000	reset_register_gain_update When set, the gain values will always take effect the following frame independent of the integration time. When not set, gain will not update if an integration time change is in progress.	N	Y
	13	X	Reserved		
	12	0x0001	reset_register_smia_serialiser_dis This bit disables the SMIA high-speed serialiser and differential output buffers.	N	N

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
	11	X	Reserved		
	10	0x0000	reset_register_restart_bad 1 = a restart is forced any time a bad frame is detected. This can shorten the delay when waiting for a good frame, since the delay for masking out a bad frame will be the integration time rather than the full-frame time.	N	N
	9	0x0000	reset_register_mask_bad 0 = The sensor will produce bad (corrupted) frames as a result of some register changes. 1 = Bad (corrupted) frames are masked within the sensor by extending the vertical blanking time for the duration of the bad frame.	N	N
	8	0x0000	reset_register_gpi_en 0 = the primary input buffers associated with the GPI0, GPI1, GPI2, GPI3 inputs are powered down and the GPI cannot be used. 1 = the input buffers are enabled and can be read through Reg0x3026-7.	N	N
	7	0x0001	reset_register_parallel_en Reserved, do not change from default	N	N
	6	0x0001	reset_register_drive_pins Reserved, do not change from default	N	N
	5	0x0000	Reserved		
	4	0x0001	reset_register_stdby_eof 0 = Transition to standby is synchronized to the end of a sensor row readout (held-off until LINE_VALID has fallen). 1 = Transition to standby is synchronized to the end of a frame.	N	Y
	3	0x0001	reset_register_lock_reg Many SMIA registers that are specified as read-only are actually implemented as read/write registers. Clearing this bit allows such registers to be written.	N	N
	2	0x0000	reset_register_stream Setting this bit places the sensor in streaming mode. Clearing this bit places the sensor in a low power mode. The result of clearing this bit depends upon the operating mode of the sensor. Entry and exit from streaming mode can also be controlled from the signal interface.	Y	N
	1	0x0000	reset_register_restart This bit always reads as 0. Setting this bit causes the sensor to truncate the current frame at the end of the current row and start resetting (integrating) the first row. The delay before the first valid frame is read out is equal to the integration time.	N	Y

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
	0	0x0000	reset_register_reset This bit always reads as 0. Setting this bit initiates a reset sequence: the frame being generated will be truncated. The reset sequence is described in "Power-On Reset Sequence" in "Top Level Overview.doc".	N	Y
12316 R0x0000301C	31:0	0x0000	mode_select_ (R/W)	Y	N
	This bit is an alias of Reg0x301A-B[2].				
12317 R0x0000301D	31:0	0x0000	image_orientation_ (R/W)		
	31:2	X	Reserved		
	1	0x0000	image_orientation_vert_flip This bit is an alias of Reg0x3040[1].	Y	YM
	0	0x0000	image_orientation_horiz_mirror This bit is an alias of Reg0x3040[0].	Y	YM
12318 R0x0000301E	31:0	0x00A8	data_pedestal_ (R/W)	N	Y
	Constant offset that is added to the ADC output for all visible pixels in order to set the black level to a value greater than 0. Read-only. Can be made read/write by clearing Reg0x301A-B[3].				
12321 R0x00003021	31:0	0x0000	software_reset_ (R/W)	N	Y
	This bit is an alias of Reg0x301A-B[0].				
12322 R0x00003022	31:0	0x0000	grouped_parameter_hold_ (R/W)	N	N
	This bit is an alias of Reg0x301A-B[15].				
12323 R0x00003023	31:0	0x0000	mask_corrupted_frames_ (R/W)	N	N
	This bit is an alias of Reg0x301A-B[9].				
12324 R0x00003024	31:0	0x0000	pixel_order_ (RO)	N	N
	00 = First row is GreenR/Red, first pixel is GreenR 01 = First row is GreenR/Red, first pixel is Red 02 = First row is Blue/GreenB, first pixel is Blue 03 = First row is Blue/GreenB, first pixel is GreenB The value in this register changes as a function of Reg0x3040[1:0].				
12326 R0x00003026	31:0	0xFFFF	gpi_status (R/W)		
	31:16	X	Reserved		
	15:13	0x0007	gpi_status_standby_pin_select Associate the standby function with an active-high input pin 0 = associate with GPI0 1 = associate with GPI1 2 = associate with GPI2 3 = associate with GPI3 4-6 = RESERVED 7 = standby function cannot be controlled by any pin Must be set to 7 if reset[8]=0.	N	N
	12:10	0x0007	gpi_status_oe_n_pin_select Associate the output-enable function with an active-low input pin 0 = associate with GPI0 1 = associate with GPI1 2 = associate with GPI2 3 = associate with GPI3 4-6 = RESERVED 7 = output-enable function is not controlled by any pin Must be set to 7 if reset[8]=0.	N	N

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
	9:7	0x0007	gpi_status_trigger_pin_select Associate the trigger function with an active-high input pin 0 = associate with GPI0 1 = associate with GPI1 2 = associate with GPI2 3 = associate with GPI3 4-6 = RESERVED 7 = trigger function is not controlled by any pin Must be set to 7 if Reg0x301A-B[8]=0.	N	N
	6:4	0x0007	gpi_status_saddr_pin_select Associate the SADDR function with an active-high input pin 0 = associate with GPI0 1 = associate with GPI1 2 = associate with GPI2 3 = associate with GPI3 4-6 = RESERVED 7 = SADDR function is not controlled by any pin Must be set to 7 if Reg0x301A-B[8]=0.	N	N
	3	RO	gpi_status_gpi3 Read-only. Return the current state of the GPI3 input pin. Invalid if Reg0x301A-B[8]=0.	N	N
	2	RO	gpi_status_gpi2 Read-only. Return the current state of the GPI2 input pin. Invalid if Reg0x301A-B[8]=0.	N	N
	1	RO	gpi_status_gpi1 Read-only. Return the current state of the GPI1 input pin. Invalid if Reg0x301A-B[8]=0.	N	N
	0	RO	gpi_status_gpi0 Read-only. Return the current state of the GPI0 input pin. Invalid if Reg0x301A-B[8]=0.	N	N
12328 R0x00003028	31:0	0x000D	analogue_gain_code_global_ (R/W)	Y	N
	Writing a gain code to this register is equivalent to writing that code to each of the 4 color-specific gain code registers. Reading from this register returns the value most recently written to the analogue_gain_code_greenR register.				
12330 R0x0000302A	31:0	0x000D	analogue_gain_code_greenr_ (R/W)	Y	N
	The gain code written to this register sets the gain for green pixels on red/green rows of the pixel array.				
12332 R0x0000302C	31:0	0x000D	analogue_gain_code_red_ (R/W)	Y	N
	The gain code written to this register sets the gain for red pixels.				
12334 R0x0000302E	31:0	0x000D	analogue_gain_code_blue_ (R/W)	Y	N
	The gain code written to this register sets the gain for blue pixels.				
12336 R0x00003030	31:0	0x000D	analogue_gain_code_greenb_ (R/W)	Y	N
	The gain code written to this register sets the gain for green pixels on blue/green rows of the pixel array.				
12338 R0x00003032	31:0	0x0100	digital_gain_greenr_ (R/W)	Y	N
	Digital gain applied to green pixels on red/green rows of the pixel array. The value is an unsigned 8.8 fixed-point format. Bits [10:8] are significant and are an alias of Reg0x3056[11:9].				
12340 R0x00003034	31:0	0x0100	digital_gain_red_ (R/W)	Y	N
	Digital gain applied to red pixels of the pixel array. The value is an unsigned 8.8 fixed-point format. Bits [10:8] are significant and are an alias of Reg0x305A[11:9].				

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12342 R0x00003036	31:0	0x0100	digital_gain_blue_ (R/W)	Y	N
	Digital gain applied to blue pixels of the pixel array. The value is an unsigned 8.8 fixed-point format. Bits [10:8] are significant and are an alias of Reg0x3058[11:9].				
12344 R0x00003038	31:0	0x0100	digital_gain_greenb_ (R/W)	Y	N
	Digital gain applied to green pixels on blue/green rows of the pixel array. The value is an unsigned 8.8 fixed-point format. Bits [10:8] are significant and are an alias of Reg0x305C[11:9].				
12346 R0x0000303A	31:0	0x000A	smia_version_ (RO)	N	N
	Return the value 10 to indicate an implementation of revision 1.0 of the SMIA specification. Read-only.				
12347 R0x0000303B	31:0	0x00FF	frame_count_ (RO)	Y	N
	In the soft standby state this counter is set to 0xFF. In streaming state this counter increments by 1 (modulo 255) at the start of each frame. The counter is incremented for both good frames and bad (corrupted) frames - its behavior is not affected by the state of Reg0x301A-B[9] (mask_corrupted_frames). After entry to the streaming state, the first frame will show a frame count of 0x01 in its embedded data. Read-only.				
12348 R0x0000303C	31:0	0x0000	frame_status (RO)		
	31:2	X	Reserved		
	1	RO	frame_status_standby This bit tells you whether the sensor is in standby state. Can be polled after standby is entered to see when the real low-power state is entered; which can happen at the end of row or frame depending on bit 0x301A[4].	N	N
	0	RO	frame_status_framesync Set on register write and reset on frame synchronization. Acts as debug flag to verify that register writes completed before last frame synchronization.	N	N

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12352 R0x00003040	31:0	0x0041	read_mode (R/W)		
	31:16	X	Reserved		
	15	0x0000	read_mode_vert_flip 0 = Normal readout 1 = Readout is flipped (mirrored) vertically so that the row specified by y_addr_end_ is read out of the sensor first. Setting this bit will change the bayer pixel order (see Reg0x3024).	Y	YM
	14	0x0000	read_mode_horiz_mirror 0 = Normal readout 1 = Readout is mirrored horizontally so that the column specified by x_addr_end_ is read out of the sensor first. Setting this bit will change the bayer pixel order (see Reg0x3024).	Y	YM
	13	0x0000	read_mode_y_sum_en Enable summing mode for rows	Y	N
	12	0x0000	read_mode_reserved RESERVED Do not change from default	Y	N
	11	0x0000	read_mode_x_bin_en Enable analogue binning in X (column) direction. When set, x_odd_inc must be set to 3 and y_odd_inc must be set to 1, along with other register changes.	Y	N
	10	0x0000	read_mode_xy_bin_en Enable analogue binning in X and Y (column and row) directions. When set, x_odd_inc and y_odd_inc must be set to 3, along with other register changes.	Y	N
	9	0x0000	read_mode_low_power Enables low power mode. This will automatically half the pixel clock speed. Can not be used when pc_speed[2:0] = 4.	Y	YM
	8:6	0x0001	read_mode_x_odd_inc Increment applied to odd addresses in X (column) direction. 1= Normal readout 3 = Read out alternate pixel pairs to halve the amount of horizontal data in a frame. 7 = Read out 1 of 4 pixel pairs to reduce the amount of horizontal data in a frame by 4.	Y	YM
	5:0	0x0001	read_mode_y_odd_inc Increment applied to odd addresses in Y (row) direction. 1= Normal readout 3 = Read out alternate pixel pairs to halve the amount of vertical data in a frame. 7 = Read out 1 of 4 pixel pairs to reduce the amount of vertical data in a frame by 4. f = Reserved, Bit 3 should be set to 0.	Y	YM

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12358 R0x00003046	31:0	0x0600	flash (R/W)		
	31:16	X	Reserved		
	15	RO	flash_strobe Reflects the current state of the FLASH output signal. Read-only.	N	N
	14	RO	flash_triggered Indicates that the FLASH output signal was asserted for the current frame. Read-only.	N	N
	13	0x0000	flash_xenon_flash Enable xenon flash. When set, the FLASH output signal will assert for the programmed period (bits [7:0]) during vertical blanking. This is achieved by keeping the integration time equal to one frame, and the pulse width less than the vertical blanking time.	Y	N
	12:11	0x0000	flash_frame_delay Flash pulse delay measured in frames.	N	N
	10	0x0001	flash_end_of_reset 1 = In xenon mode, the flash is triggered after resetting a frame. 0 = In Xenon mode, the flash is triggered after a frame readout.	N	N
	9	0x0001	flash_every_frame 1 = Flash should be enabled every frame. 0 = Flash should be enabled for 1 frame only.	N	N
	8	0x0000	flash_led_flash Enable LED flash. When set, the FLASH output signal will assert prior to the start of the resetting of a frame and will remain asserted until the end of the frame readout.	Y	Y
	7	0x0000	flash_invert_flash Invert flash output signal. When set, the FLASH output signal will be active low.	N	N
	6:5	0x0000	flash_scale Scale Factor for Flash Count	N	N
	4	0x0000	flash_trigger_timed 1 = Enable Flash Count	N	N
	3:0	X	Reserved		
12360 R0x00003048	31:0	0x0008	flash_count (R/W)	N	N
	Length of flash pulse when xenon flash is enabled. The value specifies the length in units of 256 x system_clock. When the xenon count is set to its maximum value (0x3FF), the flash pulse will automatically be truncated prior to the readout of the first row, giving the longest pulse possible.				
12374 R0x00003056	31:0	0x1034	green1_gain (R/W)		
	31:15	X	Reserved		
	14:12	0x0001	green1_gain_digital_gain Digital Gain. Legal values 1-7.	Y	N
	11:9	X	Reserved		
	8:7	0x0000	green1_gain_analog_gain Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain.	Y	N
	6:0	0x0034	green1_gain_initial_gain Initial gain = bits [6:0] * 1/32.	Y	N

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12376 R0x00003058	31:0	0x1034	blue_gain (R/W)		
	31:15	X	Reserved		
	14:12	0x0001	blue_gain_digital_gain Digital Gain. Legal values 1-7.	Y	N
	11:9	X	Reserved		
	8:7	0x0000	blue_gain_analog_gain Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain.	Y	N
	6:0	0x0034	blue_gain_initial_gain Initial gain = bits [6:0] * 1/32.	Y	N
12378 R0x0000305A	31:0	0x1034	red_gain (R/W)		
	31:15	X	Reserved		
	14:12	0x0001	red_gain_digital_gain Digital Gain. Legal values 1-7.	Y	N
	11:9	X	Reserved		
	8:7	0x0000	red_gain_analog_gain Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain.	Y	N
	6:0	0x0034	red_gain_initial_gain Initial gain = bits [6:0] * 1/32.	Y	N
12380 R0x0000305C	31:0	0x1034	green2_gain (R/W)		
	31:15	X	Reserved		
	14:12	0x0001	green2_gain_digital_gain Digital Gain. Legal values 1-7.	Y	N
	11:9	X	Reserved		
	8:7	0x0000	green2_gain_analog_gain Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain.	Y	N
	6:0	0x0034	green2_gain_initial_gain Initial gain = bits [6:0] * 1/32.	Y	N
12382 R0x0000305E	31:0	0x1034	global_gain (R/W)	Y	N
	Writing a gain to this register is equivalent to writing that code to each of the 4 color-specific gain registers. Reading from this register returns the value most recently written to the green1_gain register.				
12400 R0x00003070	31:0	0x0000	test_pattern_mode_ (R/W)	N	Y
	0 = Normal operation: Generate output data from pixel array 1 = Solid color test pattern. 2 = 100% color bar test pattern 3 = Fade to grey color bar test pattern 4 = PN9 Link integrity test pattern 256 = Marching 1's test pattern (10 bit) 257 = Marching 1's test pattern (8 bit) other = Reserved.				
12402 R0x00003072	31:0	0x0000	test_data_red_ (R/W)	N	Y
	The value for red pixels in theBayer data used for the solid color test pattern and the test cursors.				
12404 R0x00003074	31:0	0x0000	test_data_greenr_ (R/W)	N	Y
	The value for green pixels in red/green rows of the Bayer data used for the solid color test pattern and the test cursors.				
12406 R0x00003076	31:0	0x0000	test_data_blue_ (R/W)	N	Y
	The value for blue pixels in the Bayer data used for the solid color test pattern and the test cursors.				

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12408 R0x00003078	31:0	0x0000	test_data_greenb_ (R/W)	N	Y
	The value for green pixels in blue/green rows of the Bayer data used for the solid color test pattern and the test cursors.				
12410 R0x0000307A	31:0	0x0000	test_raw_mode (R/W)		
	31:2	X	Reserved		
	1	0x0000	Reserved		
	0	0x0000	Reserved		
12430 R0x0000308E	31:0	0xCC00	ease_control (R/W)		
	31:16	X	Reserved		
	15	0x0001	ease_control_btm_filter_en Enables bottom (odd 1X sample column) filter	N	N
	14:12	0x0004	ease_control_btm_bcnt bottom filter set to 8*2^BCnt samples	N	N
	11	0x0001	ease_control_top_filter_en Enables top (odd 1X sample column) filter	N	N
	10:8	0x0004	ease_control_top_bcnt top filter set to 8*2^BCnt samples	N	N
	7:6	0x0000	ease_control_corr_mode select external analog correction modes	N	N
	5:4	0x0000	ease_control_show_mode select external analog show modes	N	N
	3:2	0x0000	ease_control_ease_read_mode select external analog read modes	N	N
	1	0x0000	ease_control_reserved1 Unused do not core bit	N	N
	0	0x0000	ease_control_ease_enable Enable this bit to enable external analog sampling	N	N
	External analog sampling				
12432 R0x00003090	31:0	0x0020	ease_top_gain (R/W)		
	31:9	X	Reserved		
	8:7	0x0000	ease_top_gain_analog_gain Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain.	Y	N
	6:0	0x0020	ease_top_gain_initial_gain Initial gain = bits [6:0] * 1/32.	Y	N
	Top gain during external analog sampling				
12434 R0x00003092	31:0	0x0021	ease_btm_gain (R/W)		
	31:9	X	Reserved		
	8:7	0x0000	ease_btm_gain_analog_gain Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain.	Y	N
	6:0	0x0021	ease_btm_gain_initial_gain Initial gain = bits [6:0] * 1/32.	Y	N
	BTM gain during external analog sampling				
12436 R0x00003094	31:0	0x0003	ease_top_calib (R/W)	N	Y
	DAC Calib for top external sample data				
12438 R0x00003096	31:0	0x0005	ease_btm_calib (R/W)	N	Y
	DAC Calib for bottom external sample data				

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12440 R0x00003098	31:0	0x0000	ease_top_samp_mux (R/W)	N	Y
	Top sample data based on EASE read and corr modes				
12442 R0x0000309A	31:0	0x0000	ease_btm_samp_mux (R/W)	N	Y
	Btm sample data based on EASE read and corr modes				
12444 R0x0000309C	31:0	0x0000	ease_top_ref_mux (R/W)	N	Y
	Top reference data based on EASE read modes				
12446 R0x0000309E	31:0	0x0000	ease_btm_ref_mux (R/W)	N	Y
	Btm reference data based on EASE read modes				
12448 R0x000030A0	31:0	0x0001	x_even_inc_ (RO)	N	N
	Read-only.				
12450 R0x000030A2	31:0	0x0001	x_odd_inc_ (R/W)	Y	YM
	This register field is an alias of Reg0x3040[7:5]				
12452 R0x000030A4	31:0	0x0001	y_even_inc_ (RO)	N	N
	Read-only.				
12454 R0x000030A6	31:0	0x0001	y_odd_inc_ (R/W)	Y	YM
	This register field is an alias of Reg0x3040[3:0]				
12634 R0x0000315A	31:0	0x009A	global_flash_start (R/W)	N	Y
12636 R0x0000315C	31:0	0x0000	global_bulb_trigger_count (R/W)	N	Y
12638 R0x0000315E	31:0	0x0000	global_seq_trigger (R/W)		
	31:16	X	Reserved		
	15:12	0x0000	global_seq_trigger_global_bulb_trigger_scale 0000 = 1x 0001 = 16x 0010 = 256x 0011 = 1024x	N	Y
	11	X	Reserved		
	10	0x0000	global_seq_trigger_global_trigger_timer	N	Y
	9	RO	global_seq_trigger_grst_rd Read-Only. Global reset read sequence indicator.	N	N
	8	RO	global_seq_trigger_grst_seq Read-only. Global reset sequence indicator.	N	N
	7	0x0000	global_seq_trigger_flash_sync	N	Y
	6	0x0000	global_seq_trigger_use_flash_start	N	Y
	5:4	0x0000	global_seq_trigger_global_scale	N	Y
	3	X	Reserved		
	2	0x0000	global_seq_trigger_global_flash 0 = When a Global Reset sequence is triggered, the FLASH output will remain negated. 1 = When a Global Reset sequence is triggered, the FLASH output will pulse during the integration phase.	N	Y

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
	1	0x0000	global_seq_trigger_global_bulb 0 = Shutter open is triggered from bit[0] and shutter close is timed from the trigger point. 1 = Shutter open and close are triggered from bit[0]. This corresponds to the shutter "B" setting on a traditional camera, where "B" originally stood for "Bulb" (the shutter setting used for synchronization with a magnesium foil flash bulb) and was later considered to stand for "Brief" (an exposure that was longer than the shutter could automatically accommodate).	N	Y
	0	0x0000	global_seq_trigger_global_trigger When bit[1]=0, a 0-to-1 transition of this bit initiates (triggers) a global reset sequence. When bit[1]=1, a 0-to-1 transition of this bit initiates a global reset sequence, and leaves the shutter open; a 1-to-0 transition of this bit closes the shutter. These operations can also be controlled from the signal interface by enabling one of the GPI[3:0] signals as a trigger input.	N	Y
12640 R0x00003160	31:0	0x0098	global_rst_end (R/W)	N	N
	Controls the duration of the global reset row reset phase. A value of N gives a duration of N * 512 / vt_pix_clk_freq_mhz.				
12642 R0x00003162	31:0	0x009F	global_shutter_start (R/W)	N	N
	Controls the delay before the assertion of the SHUTTER output during a global reset sequence. A value of N gives an assertion time of N * 512 / vt_pix_clk_freq_mhz timed from the end of row that was in progress when the global reset sequence was triggered.				
12644 R0x00003164	31:0	0x0000	global_shutter_start2 (R/W)	N	N
	Controls the delay before the assertion of the SHUTTER output during a global reset sequence. A value of N gives an assertion time of N * 512 / vt_pix_clk_freq_mhz timed from the end of row that was in progress when the global reset sequence was triggered.				
12646 R0x00003166	31:0	0x00A0	global_read_start (R/W)	N	N
	Controls the delay before the start of the global reset readout phase (equivalent to the end of global reset integration phase). A value of N gives a delay of N * 512 / vt_pix_clk_freq_mhz. The integration time is given by (global_read_start - global_rst_end) * 512 / vt_pix_clk_freq_mhz.				
12648 R0x00003168	31:0	0x0000	global_read_start2 (R/W)	N	N
	Controls the delay before the start of the global reset readout phase (equivalent to the end of global reset integration phase). A value of N gives a delay of N * 512 / vt_pix_clk_freq_mhz. The integration time is given by (global_read_start - global_rst_end) * 512 / vt_pix_clk_freq_mhz.				
12776 R0x000031E8	31:0	0x0000	horizontal_cursor_position_ (R/W)	N	N
	Specify the start row for the test cursor.				
12778 R0x000031EA	31:0	0x0000	vertical_cursor_position_ (R/W)	N	N
	Specify the start column for the test cursor.				
12780 R0x000031EC	31:0	0x0000	horizontal_cursor_width_ (R/W)	N	N
	Specify the width, in rows, of the horizontal test cursor. A width of 0 disables the cursor.				
12782 R0x000031EE	31:0	0x0000	vertical_cursor_width_ (R/W)	N	N
	Specify the width, in columns, of the vertical test cursor. A width of 0 disables the cursor.				
12786 R0x000031F2	31:0	0x6E6C	i2c_ids_mipi_default (R/W)	N	N
	Programmable two-wire serial interface slave addresses for MIPI operation.				

Table 42: 0: Sensor Core Registers (continued)

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12796 R0x000031FC	31:0	0x3020	i2c_ids (R/W)	N	N
	I2C addresses.				

Table 43: GPIO_SS Registers

Register Dec(Hex)	Bits	Default	Name
15104 R0x00003B00	31:0	0x0000	txbuffer_data_register_0 (R/W)
	Location 0 of TXBuffer Legal values: [0, 65535].		
15106 R0x00003B02	31:0	0x0000	txbuffer_data_register_1 (R/W)
	Location 1 of TXBuffer Legal values: [0, 65535].		
15108 R0x00003B04	31:0	0x0000	txbuffer_data_register_2 (R/W)
	Location 2 of TXBuffer Legal values: [0, 65535].		
15110 R0x00003B06	31:0	0x0000	txbuffer_data_register_3 (R/W)
	Location 3 of TXBuffer Legal values: [0, 65535].		
15112 R0x00003B08	31:0	0x0000	txbuffer_data_register_4 (R/W)
	Location 4 of TXBuffer Legal values: [0, 65535].		
15114 R0x00003B0A	31:0	0x0000	txbuffer_data_register_5 (R/W)
	Location 5 of TXBuffer Legal values: [0, 65535].		
15116 R0x00003B0C	31:0	0x0000	txbuffer_data_register_6 (R/W)
	Location 6 of TXBuffer Legal values: [0, 65535].		
15118 R0x00003B0E	31:0	0x0000	txbuffer_data_register_7 (R/W)
	Location 7 of TXBuffer Legal values: [0, 65535].		
15120 R0x00003B10	31:0	0x0000	txbuffer_data_register_8 (R/W)
	Location 8 of TXBuffer Legal values: [0, 65535].		
15122 R0x00003B12	31:0	0x0000	txbuffer_data_register_9 (R/W)
	Location 9 of TXBuffer Legal values: [0, 65535].		
15124 R0x00003B14	31:0	0x0000	txbuffer_data_register_10 (R/W)
	Location 10 of TXBuffer Legal values: [0, 65535].		
15126 R0x00003B16	31:0	0x0000	txbuffer_data_register_11 (R/W)
	Location 11 of TXBuffer Legal values: [0, 65535].		
15128 R0x00003B18	31:0	0x0000	txbuffer_data_register_12 (R/W)
	Location 12 of TXBuffer Legal values: [0, 65535].		

Table 43: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
15130 R0x00003B1A	31:0	0x0000	txbuffer_data_register_13 (R/W)
	Location 13 of TXBuffer Legal values: [0, 65535].		
15132 R0x00003B1C	31:0	0x0000	txbuffer_data_register_14 (R/W)
	Location 14 of TXBuffer Legal values: [0, 65535].		
15134 R0x00003B1E	31:0	0x0000	txbuffer_data_register_15 (R/W)
	Location 15 of TXBuffer Legal values: [0, 65535].		
15136 R0x00003B20	31:0	0x0000	txbuffer_data_register_16 (R/W)
	Location 16 of TXBuffer Legal values: [0, 65535].		
15138 R0x00003B22	31:0	0x0000	txbuffer_data_register_17 (R/W)
	Location 17 of TXBuffer Legal values: [0, 65535].		
15140 R0x00003B24	31:0	0x0000	txbuffer_data_register_18 (R/W)
	Location 18 of TXBuffer Legal values: [0, 65535].		
15142 R0x00003B26	31:0	0x0000	txbuffer_data_register_19 (R/W)
	Location 19 of TXBuffer Legal values: [0, 65535].		
15144 R0x00003B28	31:0	0x0000	txbuffer_data_register_20 (R/W)
	Location 20 of TXBuffer Legal values: [0, 65535].		
15146 R0x00003B2A	31:0	0x0000	txbuffer_data_register_21 (R/W)
	Location 21 of TXBuffer Legal values: [0, 65535].		
15148 R0x00003B2C	31:0	0x0000	txbuffer_data_register_22 (R/W)
	Location 22 of TXBuffer Legal values: [0, 65535].		
15150 R0x00003B2E	31:0	0x0000	txbuffer_data_register_23 (R/W)
	Location 23 of TXBuffer Legal values: [0, 65535].		
15152 R0x00003B30	31:0	0x0000	txbuffer_data_register_24 (R/W)
	Location 24 of TXBuffer Legal values: [0, 65535].		
15154 R0x00003B32	31:0	0x0000	txbuffer_data_register_25 (R/W)
	Location 25 of TXBuffer Legal values: [0, 65535].		
15156 R0x00003B34	31:0	0x0000	txbuffer_data_register_26 (R/W)
	Location 26 of TXBuffer Legal values: [0, 65535].		
15158 R0x00003B36	31:0	0x0000	txbuffer_data_register_27 (R/W)
	Location 27 of TXBuffer Legal values: [0, 65535].		
15160 R0x00003B38	31:0	0x0000	txbuffer_data_register_28 (R/W)
	Location 28 of TXBuffer Legal values: [0, 65535].		

Table 43: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
15162 R0x00003B3A	31:0	0x0000	txbuffer_data_register_29 (R/W)
	Location 29 of TXBuffer Legal values: [0, 65535].		
15164 R0x00003B3C	31:0	0x0000	txbuffer_data_register_30 (R/W)
	Location 30 of TXBuffer Legal values: [0, 65535].		
15166 R0x00003B3E	31:0	0x0000	txbuffer_data_register_31 (R/W)
	Location 31 of TXBuffer Legal values: [0, 65535].		
15168 R0x00003B40	31:0	0x0000	rxbuffer_data_register_0 (RO)
	Location 0 of RXBuffer Read-only. Legal values: [0, 65535].		
15170 R0x00003B42	31:0	0x0000	rxbuffer_data_register_1 (RO)
	Location 1 of RXBuffer Read-only. Legal values: [0, 65535].		
15172 R0x00003B44	31:0	0x0000	rxbuffer_data_register_2 (RO)
	Location 2 of RXBuffer Read-only. Legal values: [0, 65535].		
15174 R0x00003B46	31:0	0x0000	rxbuffer_data_register_3 (RO)
	Location 3 of RXBuffer Read-only. Legal values: [0, 65535].		
15176 R0x00003B48	31:0	0x0000	rxbuffer_data_register_4 (RO)
	Location 4 of RXBuffer Read-only. Legal values: [0, 65535].		
15178 R0x00003B4A	31:0	0x0000	rxbuffer_data_register_5 (RO)
	Location 5 of RXBuffer Read-only. Legal values: [0, 65535].		
15180 R0x00003B4C	31:0	0x0000	rxbuffer_data_register_6 (RO)
	Location 6 of RXBuffer Read-only. Legal values: [0, 65535].		
15182 R0x00003B4E	31:0	0x0000	rxbuffer_data_register_7 (RO)
	Location 7 of RXBuffer Read-only. Legal values: [0, 65535].		
15184 R0x00003B50	31:0	0x0000	rxbuffer_data_register_8 (RO)
	Location 8 of RXBuffer Read-only. Legal values: [0, 65535].		
15186 R0x00003B52	31:0	0x0000	rxbuffer_data_register_9 (RO)
	Location 9 of RXBuffer Read-only. Legal values: [0, 65535].		
15188 R0x00003B54	31:0	0x0000	rxbuffer_data_register_10 (RO)
	Location 10 of RXBuffer Read-only. Legal values: [0, 65535].		
15190 R0x00003B56	31:0	0x0000	rxbuffer_data_register_11 (RO)
	Location 11 of RXBuffer Read-only. Legal values: [0, 65535].		
15192 R0x00003B58	31:0	0x0000	rxbuffer_data_register_12 (RO)
	Location 12 of RXBuffer Read-only. Legal values: [0, 65535].		

Table 43: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
15194 R0x00003B5A	31:0	0x0000	rxbuffer_data_register_13 (RO)
	Location 13 of RXBuffer Read-only. Legal values: [0, 65535].		
15196 R0x00003B5C	31:0	0x0000	rxbuffer_data_register_14 (RO)
	Location 14 of RXBuffer Read-only. Legal values: [0, 65535].		
15198 R0x00003B5E	31:0	0x0000	rxbuffer_data_register_15 (RO)
	Location 15 of RXBuffer Read-only. Legal values: [0, 65535].		
15200 R0x00003B60	31:0	0x0000	rxbuffer_data_register_16 (RO)
	Location 16 of RXBuffer Read-only. Legal values: [0, 65535].		
15202 R0x00003B62	31:0	0x0000	rxbuffer_data_register_17 (RO)
	Location 17 of RXBuffer Read-only. Legal values: [0, 65535].		
15204 R0x00003B64	31:0	0x0000	rxbuffer_data_register_18 (RO)
	Location 18 of RXBuffer Read-only. Legal values: [0, 65535].		
15206 R0x00003B66	31:0	0x0000	rxbuffer_data_register_19 (RO)
	Location 19 of RXBuffer Read-only. Legal values: [0, 65535].		
15208 R0x00003B68	31:0	0x0000	rxbuffer_data_register_20 (RO)
	Location 20 of RXBuffer Read-only. Legal values: [0, 65535].		
15210 R0x00003B6A	31:0	0x0000	rxbuffer_data_register_21 (RO)
	Location 21 of RXBuffer Read-only. Legal values: [0, 65535].		
15212 R0x00003B6C	31:0	0x0000	rxbuffer_data_register_22 (RO)
	Location 22 of RXBuffer Read-only. Legal values: [0, 65535].		
15214 R0x00003B6E	31:0	0x0000	rxbuffer_data_register_23 (RO)
	Location 23 of RXBuffer Read-only. Legal values: [0, 65535].		
15216 R0x00003B70	31:0	0x0000	rxbuffer_data_register_24 (RO)
	Location 24 of RXBuffer Read-only. Legal values: [0, 65535].		
15218 R0x00003B72	31:0	0x0000	rxbuffer_data_register_25 (RO)
	Location 25 of RXBuffer Read-only. Legal values: [0, 65535].		
15220 R0x00003B74	31:0	0x0000	rxbuffer_data_register_26 (RO)
	Location 26 of RXBuffer Read-only. Legal values: [0, 65535].		
15222 R0x00003B76	31:0	0x0000	rxbuffer_data_register_27 (RO)
	Location 27 of RXBuffer Read-only. Legal values: [0, 65535].		
15224 R0x00003B78	31:0	0x0000	rxbuffer_data_register_28 (RO)
	Location 28 of RXBuffer Read-only. Legal values: [0, 65535].		

Table 43: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
15226 R0x00003B7A	31:0	0x0000	rxbuffer_data_register_29 (RO)
	Location 29 of RXBuffer Read-only. Legal values: [0, 65535].		
15228 R0x00003B7C	31:0	0x0000	rxbuffer_data_register_30 (RO)
	Location 30 of RXBuffer Read-only. Legal values: [0, 65535].		
15230 R0x00003B7E	31:0	0x0000	rxbuffer_data_register_31 (RO)
	Location 31 of RXBuffer Read-only. Legal values: [0, 65535].		
15232 R0x00003B80	31:0	0x0009	i2c_master_status (RO)
	31:5	X	Reserved
	4	RO	collision indication of I2C multi-master collision 0- no collision 1- collision, needs to set i2cm_go again to repeat the intended I2C transaction Read-only.
	3	RO	rxbuffer_empty I2C master RXBuffer empty 0- not empty 1- empty Read-only.
	2	RO	i2cm_ack_error I2C master receiving acknowledge error 0- no error 1- error Read-only.
	1	RO	rxbuffer_overflow I2C master RXBuffer overflow 0- not overflow 1- overflow Read-only.
	0	RO	i2cm_done I2C master operation completion 0- not complete 1- complete Read-only.
15234 R0x00003B82	31:0	0x0000	i2c_master_control (R/W)
	31:3	X	Reserved
	2	0x0000	disable_arb disable the capability of I2C multi-master arbitration 0- enable multi-master arbitration 1- disable multi-master arbitration
	1	0x0000	ignore_ack_err I2C master ignores the ack error condition and continue to finish the i2c transactions; this intends to support AD5821 for AF functions 0- I2C Master terminates i2c transcations if ack error occurs 1- I2C Master continues i2c transcations if ack error occurs

Table 43: GPIO_SS Registers (continued)

Register Dec(Hex)	Bits	Default	Name
	0	0x0000	i2cm_go start I2C master operation 0- no operation 1- start operation
15236 R0x00003B84	31:0	0x0063	i2c_master_frequency_divider (R/W)
	Number to divide down ICB frequency to generate I2CM clock Legal values: [0, 1023].		
15238 R0x00003B86	31:0	0x0000	txbuffer_total_byte_count (R/W)
	Number to indicate bytes programmed into TXBuffer Legal values: [0, 63].		

Table 44: 1: SOC1 Registers

Register Dec (Hex)	Bits	Default	Name
12816 R0x00003210	31:0	0x01A0	color_pipeline_control (R/W)
	31:15	X	Reserved
	14	0x0000	sffb_flatten_enable Enable Flat Region Noise Filter of the Sharpen-Flatten-Filter Block (SFFB).
	13	0x0000	sffb_sharp_enable Enable Sharpening function of SFFB
	12	0x0000	sffb_smooth_enable Enable Smooting function of SFFB
	11	0x0000	sffb_sigma_enable Enable Sigma Filter function of SFFB
	10	0x0000	scale_sharp_enable Enable Post-Scale Sharpening. This function should only be enabled when there is a 3:1 downsizing of full image width. 0 - Bypass sharpening 1 - Enable sharpening
	9	X	Reserved
	8	0x0001	scale_enable 1=Enable scale 0=Bypass scale
	7	0x0001	gamma_en Enable gamma correction. 1= enable gamma correction. 0=disable gamma_correction
	6	0x0000	pcr_enable PCR enable
	5	0x0001	en_ccm Enable color correction
	4	0x0000	en_ap enable 2D aperture correction
	3	0x0000	Reserved
	2:0	X	Reserved

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12834 R0x00003222	31:0	0x0000	zoom_window_x0 (R/W)
	Lower X Boundary for zoom/crop window. In preview mode be sure to load this register with the X coordinate divided by the X skip factor Writes are synchronized to frame boundaries. Legal values: [0, 4095].		
12836 R0x00003224	31:0	0x0A1F	zoom_window_x1 (R/W)
	Upper X Boundary for zoom/crop window. In preview mode be sure to load this register with the X coordinate divided by the X skip factor Writes are synchronized to frame boundaries. Legal values: [0, 4095].		
12838 R0x00003226	31:0	0x0000	zoom_window_y0 (R/W)
	Lower Y Boundary for zoom/crop window. In preview mode be sure to load this register with the Y coordinate divided by the Y skip factor Writes are synchronized to frame boundaries. Legal values: [0, 4095].		
12840 R0x00003228	31:0	0x0797	zoom_window_y1 (R/W)
	Upper Y Boundary for zoom/crop window. In preview mode be sure to load this register with the Y coordinate divided by the Y skip factor Writes are synchronized to frame boundaries. Legal values: [0, 4095].		
12844 R0x0000322C	31:0	0x0800	x_ratio (R/W)
	Horizontal Scaling Weight Horizontal scaling weight = int_roundup((Output_X_Size / ScaleInput_X_size) * 2048). (ScaleInput_X_Size = 2600 (Full res-mode)) = int_roundup(Output_X_Size * 0.787)) Weight = 2048 implies unity scale In high precision mode, Output_X_Size should be in the range [1, 1312]. In low precision mode, Output_X_Size should be in the range [2, 2600]. Legal values: [1, 2048].		
12846 R0x0000322E	31:0	0x0800	y_ratio (R/W)
	Vertical Scaling Weight Vertical Scaling weight = int_roundup((Output_Y_Size /ScaleInput_Y_Size) * 2048) ScaleInput_Y_Size = 1952 (Full res-mode) Assuming no cropping = int_roundup(Output_Y_Size * 1.05) (Full-res-mode) Weight = 2048 implies unity scale Output_Y_Size should be in the range [1, 1952]. Legal values: [1, 2048].		
12852 R0x00003234	31:0	X	awb_debug (R/W)
	31:5	X	Reserved
	4	0x0000	awb_debug_enable enable awb debug mode
	3:0	X	awb_debug_weight force constant weight in debug mode Volatile. Legal values: [1, 9].
12854 R0x00003236	31:0	0x0000	awb_norm_sumr (RO)
	normalized sum of weighted red Read-only. Volatile. Legal values: [0, 65535].		
12856 R0x00003238	31:0	0x0000	awb_norm_sumg (RO)
	normalized sum of weighted green Read-only. Volatile. Legal values: [0, 65535].		
12858 R0x0000323A	31:0	0x0000	awb_norm_sumb (RO)
	normalized sum of weighted blue Read-only. Volatile. Legal values: [0, 65535].		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12860 R0x0000323C	31:0	0x0003	awb_x_shift (R/W)
	Shift parameter in horizontal direction in probability table, $log2_rbgg = log2_red + log2_blue - {log2_green, 1'b0} + awb_x_shift$ awb_x_shift[10] is the sign bit. It has 5 fractional bits. Legal values: [-1023, 1023].		
12862 R0x0000323E	31:0	0x0003	awb_y_shift (R/W)
	Shift parameter in horizontal direction in probability table, $log2_bg = log2_red - log2_blue + awb_y_shift$ awb_y_shift[10] is the sign bit. It has 5 fractional bits. Legal values: [-1023, 1023].		
12864 R0x00003240	31:0	0x0033	awb_xy_scale (R/W)
	31:8	X	Reserved
	7:4	0x0003	awb_y_scale Scale factor in vertical direction in probability table, $y <= log2_b/g << awb_y_scale$ Legal values: [0, 15].
	3:0	0x0003	awb_x_scale Scale factor in horizontal direction in probability table, $log2_rb/g2 << awb_x_scale$ Legal values: [0, 15].
12866 R0x00003242	31:0	0x5000	awb_weight_r0 (R/W)
	Awb weight for y0 of log2(b/g) Legal values: [0, 65535].		
12868 R0x00003244	31:0	0x5000	awb_weight_r1 (R/W)
	Awb weight for y1 of log2(b/g) Legal values: [0, 65535].		
12870 R0x00003246	31:0	0x5000	awb_weight_r2 (R/W)
	Awb weight for y2 of log2(b/g) Legal values: [0, 65535].		
12872 R0x00003248	31:0	0x5000	awb_weight_r3 (R/W)
	Awb weight for y3 of log2(b/g) Legal values: [0, 65535].		
12874 R0x0000324A	31:0	0x5000	awb_weight_r4 (R/W)
	Awb weight for y4 of log2(b/g) Legal values: [0, 65535].		
12876 R0x0000324C	31:0	0x5000	awb_weight_r5 (R/W)
	Awb weight for y5 of log2(b/g) Legal values: [0, 65535].		
12878 R0x0000324E	31:0	0x5000	awb_weight_r6 (R/W)
	Awb weight for y6 of log2(b/g) Legal values: [0, 65535].		
12880 R0x00003250	31:0	0x5000	awb_weight_r7 (R/W)
	Awb weight for y7 of log2(b/g) Legal values: [0, 65535].		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12884 R0x00003254	31:0	0x0000	first_color (R/W)
	Color of first pixel into colorpipe. 0: G1 1: R 2: B 3: G2 Legal values: [0, 3].		
12886 R0x00003256	31:0	0x079B	last_row (R/W)
	Last active row from the sensor. The sensor output a certain amount of border rows for the kernel to use. The color pipe keeps a count of the number of rows from the sensor and when the count reaches this values it signals that this is the last active row. Legal values: [0, 2047].		
12888 R0x00003258	31:0	0x0000	awb_win_x_start (R/W)
	Starting column of AWB window. Legal values: [0, 4095].		
12890 R0x0000325A	31:0	0x0000	awb_win_y_start (R/W)
	Starting row of AWB window. Legal values: [0, 2047].		
12892 R0x0000325C	31:0	0x0280	awb_win_width (R/W)
	Width of AWB window. Legal values: [0, 4095].		
12894 R0x0000325E	31:0	0x01E0	awb_win_height (R/W)
	Height of AWB window. Legal values: [0, 2047].		
12896 R0x00003260	31:0	0x0000	awb_weight_cnt_hi (RO)
	High 16bit for awb highlighted (pass weight and luma check) pixel count Read-only. Volatile. Legal values: [0, 255].		
12898 R0x00003262	31:0	0xFF08	awb_luma_th (R/W)
	31:16	X	Reserved
	15:8	0x00FF	awb_luma_th_high Upper luma threshold for pixel used in awb. Can compare to luma or maxRGB, depending on awb_thresh_max_sel. Legal values: [0, 255].
	7:0	0x0008	awb_luma_th_low Lower luma threshold for pixel used in awb. Can compare to luma or minRGB, depending on awb_thresh_min_sel. Legal values: [0, 255].
12900 R0x00003264	31:0	0x0000	awb_config (R/W)
	31:2	X	Reserved
	1	0x0000	awb_thresh_max_sel Select type of luminance threshold for max limit. 0=maxRGB 1=luma
	0	0x0000	awb_thresh_min_sel Select type of luminance threshold for min limit. 0=minRGB 1=luma

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12902 R0x00003266	31:0	0x0091	awb_weight_th (R/W)
	31:8	X	Reserved
	7:4	0x0009	awb_weight_th_high high weight threshold for pixel used in awb Legal values: [0, 9].
	3:0	0x0001	awb_weight_th_low lower weight threshold for pixel used in awb Legal values: [0, 9].
12904 R0x00003268	31:0	0x0000	awb_weight_cnt_lo (RO)
	Low 16bit for awb highlighted (pass weight and luma check) pixel count Read-only. Volatile. Legal values: [0, 65535].		
12906 R0x0000326A	31:0	0x1208	scale_sharp_control (R/W)
	31:14	X	Reserved
	13:11	0x0002	scale_sharp_gain_exp Scale Sharp Exponent Legal values: [0,7].
	10:8	0x0002	scale_sharp_gain_mant Scale Sharp Gain Legal values: [0,7].
	7:0	0x0008	scale_sharp_knee Scale Sharp threshold Legal values: [0,255].
12910 R0x0000326E	31:0	0x0080	low_pass_yuv_filter (R/W)
	31:8	X	Reserved
	7	0x0001	en_dis B/W filter enable switch
	6	0x0000	th_mode switch for adaptive Y filter threshold
	5	0x0000	eny_i2c enable y permanently
	4:3	0x0000	y_mode y filter mode 0- no filter 1- median 3 2- median 5 Legal values: [0, 2].
	2:0	0x0000	uv_mode uv filter mode 0- no filter 1- use 11110 as filter weight 2- use 01100 as filter weight 3- use 01210 as filter weight 4- use 12221 as filter weight 5- use median 3 6- use median 5 Legal values: [0, 6].

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12912 R0x00003270	31:0	0x07AA	threshold_for_y_filter_r_channel (R/W)
	31:12	X	Reserved
	11:7	0x000F	u_th U threshold Legal values: [0, 31].
	6:0	0x002A	r_th Y Filter_R channel 4:0- threshold value for R channel 5- if set to 1, enable control signal for R channel 6- if set to 1, invert control signal for R channel Legal values: [0, 127].
12914 R0x00003272	31:0	0x07E4	threshold_for_y_filter_g_channel (R/W)
	31:12	X	Reserved
	11:7	0x000F	v_th V threshold Legal values: [0, 31].
	6:0	0x0064	g_th Y Filter_G channel 4:0- threshold value for G channel 5- if set to 1, enable control signal for G channel 6- if set to 1, invert control signal for G channel Legal values: [0, 127].
12916 R0x00003274	31:0	0x002A	threshold_for_y_filter_b_channel (R/W)
	31:7	X	Reserved
	6:0	0x002A	b_th Y Filter_B channel 4:0- threshold value for B channel 5- if set to 1, enable control signal for B channel 6- if set to 1, invert control signal for B channel Legal values: [0, 127].
12918 R0x00003276	31:0	0x0000	black_level_to_ccm (R/W)
	Second Black Level prior to CCM. Black Level value will be substracted from each pixel value. Legal values: [0, 1023].		
12922 R0x0000327A	31:0	0x00A8	red_offset (R/W)
	Offset subtracted from red pixels. This register contains the value for Red pixels subtracted by IFP from raw pixel values before applying lens shading correction and digital gains. If the subtraction produces a negative result for a particular pixel, the value of this pixel is set to 0. Typically, the subtracted value should be equal to the black level targeted by the sensor. This value is subtracted from all test patterns as well. Legal values: [0, 511].		
12924 R0x0000327C	31:0	0x00A8	green1_offset (R/W)
	Offset subtracted from green1 pixels. This register contains the value for Green1 pixels subtracted by IFP from raw pixel values before applying lens shading correction and digital gains. If the subtraction produces a negative result for a particular pixel, the value of this pixel is set to 0. Typically, the subtracted value should be equal to the black level targeted by the sensor. This value is subtracted from all test patterns as well. Legal values: [0, 511].		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12926 R0x0000327E	31:0	0x00A8	green2_offset (R/W)
	Offset subtracted from green2 pixels. This register contains the value for Green2 pixels subtracted by IFP from raw pixel values before applying lens shading correction and digital gains. If the subtraction produces a negative result for a particular pixel, the value of this pixel is set to 0. Typically, the subtracted value should be equal to the black level targeted by the sensor. This value is subtracted from all test patterns as well. Legal values: [0, 511].		
12928 R0x00003280	31:0	0x00A8	blue_offset (R/W)
	Offset subtracted from blue pixels. This register contains the value for Blue pixels subtracted by IFP from raw pixel values before applying lens shading correction and digital gains. If the subtraction produces a negative result for a particular pixel, the value of this pixel is set to 0. Typically, the subtracted value should be equal to the black level targeted by the sensor. This value is subtracted from all test patterns as well. Legal values: [0, 511].		
12930 R0x00003282	31:0	0x0000	aperture_knee_gain_parameters (R/W)
	31:10	X	Reserved
	9:5	0x0000	kn_gain Aperture negative knee gain. Legal values: [0, 31].
	4:0	0x0000	kp_gain Aperture positive knee gain. Legal values: [0, 31].
12932 R0x00003284	31:0	0x0000	aperture_knee_values (R/W)
	31:16	X	Reserved
	15:8	0x0000	a_knee_n Aperture knee value for negative aperture values. Legal values: [0, 255].
	7:0	0x0000	a_knee_p Aperture knee value for postive aperture values. Legal values: [0, 255].
12934 R0x00003286	31:0	0x0000	aperture_gain_value (R/W)
	31:7	X	Reserved
	6:4	0x0000	a_exp Aperture gain value. (Exponent). Legal values: [0, 7].
	3:0	0x0000	a_gain Aperture gain value. (Integer portion). Total Gain = a_gain*2^(a_exp-4) Legal values: [0, 15].
12936 R0x00003288	31:0	0x0000	aperture_additive_clip_limit (R/W)
	31:8	X	Reserved
	7:0	0x0000	a_clip_l Aperture additive clip limit. (Used for additive aperture application) Legal values: [0, 255].

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12938 R0x0000328A	31:0	0x810C	sffb_noise_control_r (R/W)
	31:16	X	Reserved
	15:10	0x0020	sffb_slope_r SFFB Slope for red pixels. Legal values: [1,63].
	9:5	0x0008	sffb_ramp_r SFFB Ramp for red pixels. Legal values: [0, 31].
	4:0	0x000C	sffb_maxthresh_r SFFB Max Threshhold for red pixels. Legal values: [0, 31].
	Controls red slope, ramp and threshhold for the SFFB block		
12940 R0x0000328C	31:0	0x810C	sffb_noise_control_g (R/W)
	31:16	X	Reserved
	15:10	0x0020	sffb_slope_g SFFB Slope for green pixels. Legal values: [1,63].
	9:5	0x0008	sffb_ramp_g SFFB Ramp for green pixels. Legal values: [0, 31].
	4:0	0x000C	sffb_maxthresh_g SFFB Max Threshhold for green pixels. Legal values: [0, 31].
	Controls rgreen slope, ramp and threshhold for the SFFB block		
12942 R0x0000328E	31:0	0x810C	sffb_noise_control_b (R/W)
	31:16	X	Reserved
	15:10	0x0020	sffb_slope_b SFFB Slope for blue pixels. Legal values: [1,63].
	9:5	0x0008	sffb_ramp_b SFFB Ramp for blue pixels. Legal values: [0, 31].
	4:0	0x000C	sffb_maxthresh_b SFFB Max Threshhold for blue pixels. Legal values: [0, 31].
	Controls rgreen slope, ramp and threshhold for the SFFB block		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12944 R0x00003290	31:0	0x5404	sffb_weight_control (R/W)
	31:15	X	Reserved
	14:12	0x0005	sffb_sharp_effect Sharpening effect. Increasing this value increases the sharpening effect. Legal values: [0, 7].
	11	X	Reserved
	10:4	0x0040	sffb_wt_flat Used to tune flatness as the brightness of image increases. Firmware should increase wt_flat with decreasing overall brightness of the image. Legal values: [32, 64].
	3	X	Reserved
	2:0	0x0004	sffb_wt_flat_incr Used to smooth out transition from a flat region to a filtered region. Increasing sffb_wt_flat_incr causes a less abrupt transition between flat and filtered regions. Legal values: [0, 4].
	SFFB Weighting Controls		
12946 R0x00003292	31:0	0x001F	sffb_sobel_flat (R/W)
	31:8	X	Reserved
	7:0	0x001F	sffb_sobel_flat Legal values: [0, 255].
	SFFB Sobel Flat		
12948 R0x00003294	31:0	0x00FF	sffb_sobel_sharp (R/W)
	31:8	X	Reserved
	7:0	0x00FF	sffb_sobel_sharp Legal values: [0, 255].
	SFFB Sobel Sharp		
12958 R0x0000329E	31:0	0x0000	preview_hunting_gain_enable (R/W)
	Enable x2 gain to R G B values for Preview Hunting Mode 0- disable 1- enable		
12960 R0x000032A0	31:0	0x0142	dkdelta_ccm_cc1 (R/W)
	Dark Delta CCM Mantissa of Coefficient 1 (C11) C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Legal values: [-256, 255].		
12962 R0x000032A2	31:0	0x00D9	dkdelta_ccm_cc2 (R/W)
	Dark Delta CCM Mantissa of Coefficient 2 (C12) C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Legal values: [-256, 255].		
12964 R0x000032A4	31:0	0x01E5	dkdelta_ccm_cc3 (R/W)
	Dark Delta CCM Mantissa of Coefficient 3 (C13) C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Legal values: [-256, 255].		
12966 R0x000032A6	31:0	0x002B	dkdelta_ccm_cc4 (R/W)
	Dark Delta CCM Mantissa of Coefficient 4 (C21) C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Legal values: [-256, 255].		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12968 R0x000032A8	31:0	0x0147	dkdelta_ccm_cc5 (R/W)
	Dark Delta CCM Mantissa of Coefficient 5 (C22) C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Legal values: [-256, 255].		
12970 R0x000032AA	31:0	0x008E	dkdelta_ccm_cc6 (R/W)
	Dark Delta CCM Mantissa of Coefficient 6 (C23) C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Legal values: [-256, 255].		
12972 R0x000032AC	31:0	0x01EA	dkdelta_ccm_cc7 (R/W)
	Dark Delta CCM Mantissa of Coefficient 7 (C31) C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Legal values: [-256, 255].		
12974 R0x000032AE	31:0	0x007D	dkdelta_ccm_cc8 (R/W)
	Dark Delta CCM Mantissa of Coefficient 8 (C32) C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Legal values: [-256, 255].		
12976 R0x000032B0	31:0	0x0199	dkdelta_ccm_cc9 (R/W)
	Dark Delta CCM Mantissa of Coefficient 9 (C33) C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Legal values: [-256, 255].		
12978 R0x000032B2	31:0	0x2314	dkdelta_ccm_ctl (R/W)
	31:14	X	Reserved
	13	0x0001	dkdelta_ccm_enable Enable
	12:8	0x0003	dkdelta_ccm_gain Threshold Gain Legal values: [0, 31].
	7:0	0x0014	dkdelta_ccm_threshold Lower Threshold Legal values: [0, 255].
	Dark Delta CCM Control		
12980 R0x000032B4	31:0	0x0210	dkdelta_ccm_scale (R/W)
	31:10	X	Reserved
	9:5	0x0010	dkdelta_ccm_scale_blue Blue Scale with 1-bit integer and 4-bit fractional Legal values: [0, 16].
	4:0	0x0010	dkdelta_ccm_scale_red Red Scale with 1-bit integer and 4-bit fractional Legal values: [0, 16].
	Dark Delta CCM Scale		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12982 R0x000032B6	31:0	0x4924	dkdelta_ccm_exp1 (R/W)
	31:15	X	Reserved
	14:12	0x0004	dkdelta_ccm_exp_c22 Dark Delta CCM Exponent for C22 Legal values: [0, 4].
	11:9	0x0004	dkdelta_ccm_exp_c21 Dark Delta CCM Exponent for C21 Legal values: [0, 4].
	8:6	0x0004	dkdelta_ccm_exp_c13 Dark Delta CCM Exponent for C13 Legal values: [0, 4].
	5:3	0x0004	dkdelta_ccm_exp_c12 Dark Delta CCM Exponent for C12 Legal values: [0, 4].
	2:0	0x0004	dkdelta_ccm_exp_c11 Dark Delta CCM Exponent for C11 Legal values: [0, 4].
	Dark Delta CCM Exponents 1		
12984 R0x000032B8	31:0	0x0924	dkdelta_ccm_exp2 (R/W)
	31:12	X	Reserved
	11:9	0x0004	dkdelta_ccm_exp_c33 Dark Delta CCM Exponent for C33 Legal values: [0, 4].
	8:6	0x0004	dkdelta_ccm_exp_c32 Dark Delta CCM Exponent for C32 Legal values: [0, 4].
	5:3	0x0004	dkdelta_ccm_exp_c31 Dark Delta CCM Exponent for C31 Legal values: [0, 4].
	2:0	0x0004	dkdelta_ccm_exp_c23 Dark Delta CCM Exponent for C23 Legal values: [0, 4].
	Dark Delta CCM Exponents 2		
12986 R0x000032BA	31:0	0x0000	red_offset_to_ccm (R/W)
	Red offset, Signed value with 8-bit integer and 2-bit fractional Legal values: [-512, 511].		
12988 R0x000032BC	31:0	0x0000	green_offset_to_ccm (R/W)
	Green offset, Signed value with 8-bit integer and 2-bit fractional Legal values: [-512, 511].		
12990 R0x000032BE	31:0	0x0000	blue_offset_to_ccm (R/W)
	Blue offset, Signed value with 8-bit integer and 2-bit fractional Legal values: [-512, 511].		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12992 R0x000032C0	31:0	0x3923	ccm_exp_low_byte (R/W)
	31:15	X	Reserved
	14:12	0x0003	ccm_cc5_exp CCM Exponent for C22 Legal values: [0, 4].
	11:9	0x0004	ccm_cc4_exp CCM Exponent for C21 Legal values: [0, 4].
	8:6	0x0004	ccm_cc3_exp CCM Exponent for C13 Legal values: [0, 4].
	5:3	0x0004	ccm_cc2_exp CCM Exponent for C12 Legal values: [0, 4].
	2:0	0x0003	ccm_cc1_exp CCM Exponent for C11 Legal values: [0, 4].
12994 R0x000032C2	31:0	0x0724	ccm_exp_high_byte (R/W)
	31:12	X	Reserved
	11:9	0x0003	ccm_cc9_exp CCM Exponent for C33 Legal values: [0, 4].
	8:6	0x0004	ccm_cc8_exp CCM Exponent for C32 Legal values: [0, 4].
	5:3	0x0004	ccm_cc7_exp CCM Exponent for C31 Legal values: [0, 4].
	2:0	0x0004	ccm_cc6_exp CCM Exponent for C23 Legal values: [0, 4].
12996 R0x000032C4	31:0	0xD9EE	ccm_elements_1_and_2 (R/W)
	31:16	X	Reserved
	15:8	0x00D9	ccm_cc2 CCM Mantissa of Coefficient 2 (C12) C(i,j) = Sign(i,j) * Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Row sum (C(i,1)+C(i,2)+C(i,3)) must be less than 128. Legal values: [0, 255].
	7:0	0x00EE	ccm_cc1 CCM Mantissa of Coefficient 1 (C11). Sign is positive. C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Row sum (C(i,1)+C(i,2)+C(i,3)) must be less than 128. Legal values: [0, 255].

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
12998 R0x000032C6	31:0	0x2B1B	ccm_elements_3_and_4 (R/W)
	31:16	X	Reserved
	15:8	0x002B	ccm_cc4 CCM Mantissa of Coefficient 4 (C21) C(i,j) = Sign(i,j) * Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Row sum (C(i,1)+C(i,2)+C(i,3)) must be less than 128. Legal values: [0, 255].
	7:0	0x001B	ccm_cc3 CCM Mantissa of Coefficient 3 (C13) C(i,j) = Sign(i,j) * Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Row sum (C(i,1)+C(i,2)+C(i,3)) must be less than 128. Legal values: [0, 255].
13000 R0x000032C8	31:0	0x8BEA	ccm_elements_5_and_6 (R/W)
	31:16	X	Reserved
	15:8	0x008B	ccm_cc6 CCM Mantissa of Coefficient 6 (C23) C(i,j) = Sign(i,j) * Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Row sum (C(i,1)+C(i,2)+C(i,3)) must be less than 128. Legal values: [0, 255].
	7:0	0x00EA	ccm_cc5 CCM Mantissa of Coefficient 5 (C22). Sign is positive. C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Row sum (C(i,1)+C(i,2)+C(i,3)) must be less than 128. Legal values: [0, 255].
13002 R0x000032CA	31:0	0x7D16	ccm_elements_7_and_8 (R/W)
	31:16	X	Reserved
	15:8	0x007D	ccm_cc8 CCM Mantissa of Coefficient 8 (C32) C(i,j) = Sign(i,j) * Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Row sum (C(i,1)+C(i,2)+C(i,3)) must be less than 128. Legal values: [0, 255].
	7:0	0x0016	ccm_cc7 CCM Mantissa of Coefficient 7 (C31) C(i,j) = Sign(i,j) * Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Row sum (C(i,1)+C(i,2)+C(i,3)) must be less than 128. Legal values: [0, 255].
13004 R0x000032CC	31:0	0x2DC2	ccm_elements_9_and_signs (R/W)
	31:14	X	Reserved
	13	0x0001	ccm_cc8_sign Sign for CCM C32, 1 - negative, 0 - positive
	12	0x0000	ccm_cc7_sign Sign for CCM C31, 1 - negative, 0 - positive
	11	0x0001	ccm_cc6_sign Sign for CCM C23, 1 - negative, 0 - positive
	10	0x0001	ccm_cc4_sign Sign for CCM C21, 1 - negative, 0 - positive
	9	0x0000	ccm_cc3_sign Sign for CCM C13, 1 - negative, 0 - positive
	8	0x0001	ccm_cc2_sign Sign for CCM C12, 1 - negative, 0 - positive

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	7:0	0x00C2	ccm_cc9 CCM Mantissa of Coefficient 9 (C33). Sign is positive. C(i,j) = Mantissa(i,j) * 2^ -(Exp(i,j) + 2) Row sum (C(i,1)+C(i,2)+C(i,3)) must be less than 128. Legal values: [0, 255].
13012 R0x000032D4	31:0	0x0080	red_digital_gain (R/W)
	Digital gain for Red channel * 128. Legal values: [0, 1023].		
13014 R0x000032D6	31:0	0x0080	green1_digital_gain (R/W)
	Digital gain for Green1 channel * 128. Legal values: [0, 1023].		
13016 R0x000032D8	31:0	0x0080	green2_digital_gain (R/W)
	Digital gain for Green2 channel * 128. Legal values: [0, 1023].		
13018 R0x000032DA	31:0	0x0080	blue_digital_gain (R/W)
	Digital gain for Blue channel * 128. Legal values: [0, 1023].		
13020 R0x000032DC	31:0	0x0080	second_digital_gain (R/W)
	Digital gain for all channels applied after pga. Legal values: [0, 1023].		
13042 R0x000032F2	31:0	0x0000	fd_win_x_start (R/W)
	Starting column of FD window. Legal values: [0, 4095].		
13044 R0x000032F4	31:0	0x0000	fd_win_y_start (R/W)
	Starting row of FD window. Legal values: [0, 2047].		
13046 R0x000032F6	31:0	0x0280	fd_win_width (R/W)
	Width of FD window. Legal values: [0, 4095].		
13048 R0x000032F8	31:0	0x0005	fd_win_height (R/W)
	Height of FD window. Legal values: [0, 2047].		
13050 R0x000032FA	31:0	0x0000	fd_sum (RO)
	Sum of pixel lumas in the flicker detect window, PMFP{3+13} Read-only. Volatile. Legal values: [0,65535].		
13100 R0x0000332C	31:0	0x0000	fm_blank_frames (R/W)
	Blank outgoing frames. The bit is synchronized with frame enable. Writes are synchronized to frame boundaries.		
13102 R0x0000332E	31:0	0x0000	output_format_configuration (R/W)
	31:11	X	Reserved
	10:9	0x0000	fm_proc_bayer_first_color Processed bayer output first color (0:Gr, 1:R, 2:B, 3:Gb) 00 = Green1 01 = Red 10 = Blue 11 = Green2 Legal values: [0, 3].

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	8:7	0x0000	fm_rgb_type RGB output format (0:565, 1:555, 2:444x, 3:x444) 00 = 16-bit RGB565 01 = 15-bit RGB555 10 = 12-bit RGB444x 11 = 12-bit RGBx444 Legal values: [0, 3].
	6:5	0x0000	fm_output_format Selects output format (0: YUV, 1: RGB, 2: Pbayer) 00 = YUV 01 = RGB 10 = Processed Bayer Legal values: [0, 2].
	4:3	0x0000	fm_yuv_sampling_mode Select sampling mode for YUV422 (0: even UV, 1: odd UV, 2: even U odd V) 00 = Use even columns for U and V 01 = Use odd columns for U and V 02 = Use current column for U and V Legal values: [0, 2].
	2	0x0000	fm_mono_enable Enable monochrome output. Causes formatt to only generate luma information.
	1	0x0000	fm_swap_bytes Swap output pixel hi byte with low byte. In YUV mode, swaps chroma with luma. In RGB mode, swaps odd and even bytes.
	0	0x0000	fm_swap_red_blue Swap R/B or Cr/Cb channels In YUV output mode, swaps Cb and Cr channels. In RGB mode, swaps R and B.
	Output Format Configuration		
13104 R0x00003330	31:0	0x0000	output_format_test (R/W)
	31:12	X	Reserved
	11	0x0000	fm_shift_output When enabled this left shifts the pixel data by 3 bits.
	10:9	X	Reserved
	8	0x0000	fm_pga_bypass_enable Enable lens correction bypass. Take output data from lens correction.
	7	0x0000	freeze Freeze update of R0x332E and SOC size registers 1=freeze update of R0x332E and SOC size registers
	6	X	Reserved
	5:3	0x0000	fm_test_ramp_type Test ramp output (0: Off, 1:Col, 2:Row, 3: Frame) 00 = Off 01 = Column 10 = Row 11 = Frame Legal values: [0, 3].

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	2	0x0000	fm_disable_cb Disable Cb/B channel 1=disable Cb channel (B in RGB mode)
	1	0x0000	fm_disable_y Disable Y/G channel 1=disable Y channel (G in RGB mode)
	0	0x0000	fm_disable_cr Disable Cr/R channel 1=disable Cr channel (R in RGB mode)
13106 R0x00003332	31:0	0x0000	fm_line_count (RO)
	Current line number. Read-only. Legal values: [0, 4095].		
13108 R0x00003334	31:0	0x0000	fm_frame_count (RO)
	Frame count since reset. Counter wraps around at 16-bit boundary. Read-only. Legal values: [0, 65535].		
13128 R0x00003348	31:0	0x6440	special_effect_parameters (R/W)
	31:16	X	Reserved
	15:8	0x0064	solarization_threshhold solarization threshold Legal values: [0, 255].
	7	X	Reserved
	6	0x0001	dither_luma dither luma only 0- dither in all channels 1- dither only in luma channel
	5:3	0x0000	dither_width bit width of dither valid dithering if set width of dither bit to 1, 2, 3 or 4 no dithering if set width of dither bit to 0, 5, 6 or 7 Legal values: [1, 4].
	2:0	0x0000	special_effect special effect selection bits 0- disabled 1- monochrome 2- sepia 3- negative 4- solarization with unmodified UV 5- solarization with -UV Legal values: [0, 5].
13130 R0x0000334A	31:0	0xB023	sepia_constants (R/W)
	31:16	X	Reserved
	15:8	0x00B0	sepia_cb sepia constants for Cb Legal values: [0, 255].
	7:0	0x0023	sepia_cr sepia constants for Cr Legal values: [0, 255].

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
13136 R0x00003350	31:0	0x0030	cdc15_hi_thr_comb (R/W)
	31:10	X	Reserved
	9:0	0x0030	cdc15_hi_thr_comb 1.5D Bright Clusters DC Combined Theshold. (CDC15_Hi_Thr_Satur/Slope). This register indirectly defines the CDC threshold slope for bright Clusters in darker regions. 10-bit unsigned integer. Legal values: [1,1023].
13138 R0x00003352	31:0	0x00F0	cdc15_hi_thr_satur (R/W)
	31:9	X	Reserved
	8:6	0x0003	cdc15_hi_thr_exp CDC Saturation Threshold Exponent. Legal values: [0,7].
	5:0	0x0030	cdc15_hi_thr_mantissa 1.5D Bright Clusters DC Saturation Threshold Mantissa. Saturation threshold contributes to the effective DC threshold for bright Clusters mostly in brighter regions. The effective threshold increases asymptotically toward this value as region brightness increases. Value = Mantissa * 2^Exponent. Legal values: [1,63].
13140 R0x00003354	31:0	0x0120	cdc15_lo_thr_comb (R/W)
	31:10	X	Reserved
	9:0	0x0120	cdc15_lo_thr_comb 1.5D Dark Clusters DC Combined Theshold. (CDC15_Lo_Thr_Satur/Slope). This register indirectly defines the CDC threshold slope for dark pixels in brighter regions. 10-bit unsigned integer. Legal values: [1,1023].
13142 R0x00003356	31:0	0x0170	cdc15_lo_thr_satur (R/W)
	31:9	X	Reserved
	8:6	0x0005	cdc15_lo_thr_exp CDC Saturation Threshold Exponent. Legal values: [0,7].
	5:0	0x0030	cdc15_lo_thr_mantissa 1.5D Bright Clusters DC Saturation Threshold Mantissa. Saturation threshold contributes to the effective DC threshold for dark Clusters mostly in darker regions. The effective threshold decreases asymptotically toward this value as region brightness increases. Value = Mantissa * 2^Exponent. Legal values: [1,63].
13144 R0x00003358	31:0	0x0000	pcr_color_kill_control (R/W)
	31:11	X	Reserved
	10:8	0x0000	pcr_ckgain pcr colorkill gain Legal values: [0, 7].
	7:0	0x0000	pcr_ckth pcr colorkill threshold Legal values: [0, 255].
13146 R0x0000335A	31:0	0x0000	pcr_color_gain1 (R/W)
	PCR saturation gain, value 0 Legal values: [1, 65535].		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
13148 R0x0000335C	31:0	0x0000	pcr_color_gain2 (R/W)
	PCR saturation gain, value 1 Legal values: [1, 65535].		
13150 R0x0000335E	31:0	0x0000	pcr_color_gain3 (R/W)
	PCR saturation gain, value 2 Legal values: [1, 65535].		
13152 R0x00003360	31:0	0x0000	skin_tone_th1 (R/W)
	Skin tone th for Cr/Cb<-0.1 Legal values: [0, 1023].		
13154 R0x00003362	31:0	0x0000	skin_tone_th2 (R/W)
	Skin tone th for Cr/Cb>-1.2 Legal values: [0, 1023].		
13156 R0x00003364	31:0	0x0000	skin_tone_th3 (R/W)
	Skin tone th for Y>1/8 Y max Legal values: [0, 1023].		
13158 R0x00003366	31:0	0x0000	skin_tone_th4 (R/W)
	Skin tone th for Y<7/8 Y max Legal values: [0, 1023].		
13160 R0x00003368	31:0	0x0000	skin_tone_th5 (R/W)
	Skin tone th for Cr>0.1*Crmin Legal values: [0, 1023].		
13162 R0x0000336A	31:0	0x0000	skin_tone_th6 (R/W)
	Skin tone th for Cb<0.1*Cbmin Legal values: [0, 1023].		
13164 R0x0000336C	31:0	0x0000	pcr_color_gain4 (R/W)
	PCR saturation gain, value 3 Legal values: [1, 65535].		
13166 R0x0000336E	31:0	0x0000	pcr_color_gain5 (R/W)
	PCR saturation gain, value 4 Legal values: [1, 65535].		
13168 R0x00003370	31:0	0x0000	pcr_color_gain6 (R/W)
	PCR saturation gain, value 5 Legal values: [1, 65535].		
13170 R0x00003372	31:0	0x0000	pcr_color_gain7 (R/W)
	PCR saturation gain, value 6 Legal values: [1, 65535].		
13172 R0x00003374	31:0	0x0000	pcr_color_gain8 (R/W)
	PCR saturation gain, value 7 Legal values: [1, 65535].		
13174 R0x00003376	31:0	0x0000	pcr_color_gain9 (R/W)
	PCR saturation gain, value 8 Legal values: [1, 65535].		
13180 R0x0000337C	31:0	0x0006	yuv_ycbcr_control (R/W)
	31:4	X	Reserved
	3	0x0000	fm_clip Clip Y in 16-235; U and V in 16-240
	2	0x0001	fm_auv_offset Add 128 to U and V

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	1	0x0001	select_601 coefficient control 0- YUV (BT-709) coefficients 1- YCbCr (BT-601) coefficients
	0	0x0000	fm_normalize Normalize Y in 16-235; U and V in 16-240
13182 R0x0000337E	31:0	0x0000	y_rgb_offset (R/W)
	31:16	X	Reserved
	15:8	0x0000	fm_y_offset Y offset Legal values: [0, 255].
	7:0	0x0000	fm_rgb_offset RGB offset Legal values: [0, 255].
13184 R0x00003380	31:0	0x04CF	kernel_config (R/W)
	31:16	X	Reserved
	15	0x0000	Reserved
	14:13	X	Reserved
	12	0x0000	grb_position_type GRB - Set PD mode. 1'b1 = sum, 1'b0 = max. When set, delta_mult = delta_x + delta_y, when clear delta_mult = max(delta_x,delta_y) Legal values: [0,1].
	11	0x0000	grb_position_enable GRB - Enable Position Dependent Parameters. When set, enables position dependent parameter control of green channel rebalance. Legal values: [0,1].
	10	0x0001	grb_enable GRB - Enable Green Channel Reblance When set, enables the GRB module. Legal values: [0,1].
	9	X	Reserved
	8	0x0000	dc_1d_cluster_enable DC1D - Enable cluster 1.5D defect correction. When set to 1'b1, the 1D defect correction will correct for two pixel clusters sacrificing pixel quality. Legal values: [0,1].
	7	0x0001	dc_1d_fuse_enable DC1D - Enable fused 1.5D defect correction. This enables the 1D algorithm to look for and remove fused defects on the bottom two rows of the kernel. This bit MUST be set the same as dc_fuse_enable. Legal values: [0,1].
	6	0x0001	dc_1d_enable DC1D - Enable single pixel 1.5D defect correction. This enables defect correction on the bottom two lines of the kernel. Legal values: [0,1].

Table 44: 1: SOC1 Registers (continued)

	5	0x0000	bnr_filter_all BNR - Bayer Noise Reduction filter all pixel override. When set to 1, the results of the max-min < nr_minmax_thresh compare is ignored and all pixels are filtered by noise reduction. Legal values: [0,1].
	4	0x0000	bnr_enable BNR - Bayer Noise Reduction enable. When set to 1, enable noise reduction algorythm. Noise reduction is always overridden by defect correction. If a pixel is determined to be a defect, it will not be noise reduced. Legal values: [0,1].
	3	0x0001	dc_matrix_rotate DC - Defect correction matrix rotate. When set to 0, forces defect detection / correction to always use the sqaure pattern, like pixels for defect correction. When set to 1, use square pattern for blue and red pixels but use diamond pattern for green pixels. Legal values: [0,1].
	2	0x0001	dc_cluster_enable DC - Cluster (2 pix) defect correction defect enable. When set to 1 this bit enables 2D two pixel cluster defect correction. This bit overrides DC enable. Legal values: [0,1].
	1	0x0001	dc_fuse_enable DC - Fused defect correction enable bit. When set to 1, enable fused defect correction. Legal values: [0,1].
	0	0x0001	dc_enable DC - Single pixel defect correction enable. When set to 1, enable 2D single pixel defect correction. Legal values: [0,1].
13186 R0x00003382	31:0	0x0030	cdc_hi_thr_comb (R/W)
	31:10	X	Reserved
	9:0	0x0030	cdc_hi_thr_comb Bright Clusters DC Combined Theshold. (CDC_Hi_Thr_Satur/Slope). This register indirectly defines the CDC threshold slope for bright Clusters in darker regions. Increasing value in this register while holding cdc_hi_thr_satur constant will decrease CDC threshold slope and effective DC threshold for darker regions. 10-bit unsigned integer. Legal values: [1,1023].
13188 R0x00003384	31:0	0x00F0	cdc_hi_thr_satur (R/W)
	31:9	X	Reserved
	8:6	0x0003	cdc_hi_thr_exp CDC Saturation Threshold Exponent. Legal values: [0,7].
	5:0	0x0030	cdc_hi_thr_mantissa Bright Clusters DC Saturation Threshold Mantissa. Saturation threshold contributes to the effective DC threshold for bright Clusters mostly in brighter regions. The effective threshold increases asymptotically toward this value as region brightness increases. Value = Mantissa * 2^Exponent. Legal values: [1,63].

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
13190 R0x00003386	31:0	0x0120	cdc_lo_thr_comb (R/W)
	31:10	X	Reserved
	9:0	0x0120	cdc_lo_thr_comb Dark Clusters DC Combined Theshold. (CDC_Lo_Thr_Satur/Slope). This register indirectly defines the CDC threshold slope for dark pixels in brighter regions. Increasing the value in this register while holding cdc_lo_thr_satur constant will decrease CDC threshold slope and the effective DC threshold for brighter regions. 10-bit unsigned integer. Legal values: [1,1023].
13192 R0x00003388	31:0	0x0170	cdc_lo_thr_satur (R/W)
	31:9	X	Reserved
	8:6	0x0005	cdc_lo_thr_exp CDC Saturation Threshold Exponent. Legal values: [0,7].
	5:0	0x0030	cdc_lo_thr_mantissa Dark Clusters DC Saturation Threshold Mantissa. Saturation threshold contributes to the effective DC threshold for dark Clusters mostly in darker regions. The effective threshold decreases asymptotically toward this value as region brightness increases. Value = Mantissa * 2^Exponent. Legal values: [1,63].
13194 R0x0000338A	31:0	0x0080	bnr_minmax_thresh (R/W)
	BNR - 8msb of max-min threshold to filter a pixel. If the max surround - min surround is less than this value * 4, then BNR filters the current pixel. Legal values: [0,255].		
13196 R0x0000338C	31:0	0x0020	bnr_thresh_low_red (R/W)
	BNR - Pedestal for red pixel threshold calculation. This value is added to the threshold. It represents the lowest value the threshold can be for red pixels. It is added to the output of the gain multiplication. Legal values: [0,255].		
13200 R0x00003390	31:0	0x0020	bnr_thresh_gain_red (R/W)
	BNR - Red pixel threshold gain. This value multiplied by the red pixels value minus the red barrier pixel intensity determines the red pixel threshold. Legal values: [0,255].		
13202 R0x00003392	31:0	0x0020	bnr_thresh_low_green (R/W)
	BNR - Pedestal for green pixel threshold calculation. This value is added to the threshold. It represents the lowest value the threshold can be for green pixels. It is added to the output of the gain multiplication. Legal values: [0,255].		
13206 R0x00003396	31:0	0x0020	bnr_thresh_gain_green (R/W)
	BNR - Green pixel threshold gain. This value multiplied by the green pixels value minus the green barrier pixel intensity determines the green pixel threshold. Legal values: [0,255].		
13208 R0x00003398	31:0	0x0020	bnr_thresh_low_blue (R/W)
	BNR - Pedestal for blue pixel threshold calculation. This value is added to the threshold. It represents the lowest value the threshold can be for blue pixels. It is added to the output of the gain multiplication. Legal values: [0,255].		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
13212 R0x0000339C	31:0	0x0020	bnr_thresh_gain_blue (R/W)
	BNR - Blue pixel threshold gain. This value multiplied by the blue pixels value minus the blue barrier pixel intensity determines the blue pixel threshold. Legal values: [0,255].		
13214 R0x0000339E	31:0	0x318C	bnr_blend_strength (R/W)
	31:15	X	Reserved
	14:10	0x000C	bnr_r_strength BNR Red Blend Strength. Determines how much BNR pixels are blended with non-BNR pixels at the output of dcnr. 16 corresponds to full BNR, 0 corresponds to no BNR. Legal values: [0,16].
	9:5	0x000C	bnr_g_strength BNR Green Blend Strength. Determines how much BNR pixels are blended with non-BNR pixels at the output of dcnr. 16 corresponds to full BNR, 0 corresponds to no BNR. Legal values: [0,16].
	4:0	0x000C	bnr_b_strength BNR Blue Blend Strength. Determines how much BNR pixels are blended with non-BNR pixels at the output of dcnr. 16 corresponds to full BNR, 0 corresponds to no BNR. Legal values: [0,16].
13216 R0x000033A0	31:0	0x1008	grb_pos_thresholds (R/W)
	31:16	X	Reserved
	15:8	0x0010	grb_apos GRB - maximum positive delta_g slope. This is the slope of the line denoting the maximum positive delta_g. This number is multiplied by the median green. In position dependent mode, this is a0pos. Legal values: [0,255].
	7:0	0x0008	grb_bpos GRB - maximum positive delta_g offset. This is the offset of the line denoting the maximum positive delta_g. This number is added to the scaled center green pixel. In position dependent mode, this is b0pos. Legal values: [0,255].
13218 R0x000033A2	31:0	0x1008	grb_neg_thresholds (R/W)
	31:16	X	Reserved
	15:8	0x0010	grb_aneg GRB - maximum negative delta_g slope. This is the slope of the line denoting the maximum negative delta_g. This number is multiplied by the median green. In position dependent mode, this is a0pos. Legal values: [0,255].
	7:0	0x0008	grb_bneg GRB - maximum negative delta_g offset. This is the offset of the line denoting the maximum negative delta_g. This number is added to the scaled center green pixel. In position dependent mode, this is b0pos. Legal values: [0,255].

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
13220 R0x000033A4	31:0	0x0F0F	grb_position_pos (R/W)
	31:16	X	Reserved
	15:8	0x000F	grb_position_agpos GRB - apos positional dependent gain. This number divided by 64 is multiplied times the position delta to determine apos limit for PD GRB. Legal values: [0,255].
	7:0	0x000F	grb_position_bgpos GRB - bpos positional dependent gain. This number divided by 64 is multiplied times the position delta to determine bpos limit for PD GRB. Legal values: [0,255].
13222 R0x000033A6	31:0	0x0F0F	grb_position_neg (R/W)
	31:16	X	Reserved
	15:8	0x000F	grb_position_agneg GRB - aneg positional dependent gain. This number divided by 64 is multiplied times the position delta to determine bneg limit for PD GRB. Legal values: [0,255].
	7:0	0x000F	grb_position_bgneg GRB - bneg positional dependent gain. This number divided by 64 multiplied times the postion delta to determine aneg limit for PD GRB. Legal values: [0,255].
13224 R0x000033A8	31:0	0x0000	grb_position_window_x0 (R/W)
	GRB - X location of PD window upper left corner. Pixels to the left of this location will be balanced more agressively by position dependent GRB. GRB balancing to the right of this column will be limited. Legal values: [0,4095].		
13226 R0x000033AA	31:0	0x0000	grb_position_window_y0 (R/W)
	GRB - Y location of PD window upper left corner. Pixels above this row will be balanced more agressively by position dependent GRB. GRB balancing below this line will be limited. Legal values: [0,4095].		
13228 R0x000033AC	31:0	0x0FFF	grb_position_window_x1 (R/W)
	GRB - X location of PD window lower right corner. Pixels to the right of this column will be balanced more agressively by position dependent GRB. GRB balancing to the left of this column will be limited. Legal values: [0,4095].		
13230 R0x000033AE	31:0	0x0FFF	grb_position_window_y1 (R/W)
	GRB - Y location of PD window lower right corner. Pixels below of this row will be balanced more agressively by position dependent GRB. GRB balancing above of this row will be limited. Legal values: [0,4095].		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
13232 R0x000033B0	31:0	0x152B	ffnr_alpha_beta (R/W)
	31:15	X	Reserved
	14:8	0x0015	ffnr_alpha FFNR - Alpha Parameter These 7 bits determine how center weighted the demosaic of red and blue pixels are on red and blue bayer locations. The notation is 1.6 fixed point. 64 is the largest allowable number which represents 1.0. When added to FFNR beta must equal 1.0. Legal values: [0,64].
	7	X	Reserved
	6:0	0x002B	ffnr_beta FFNR - Beta Parameter These seven bits determine how center weighted the demosaic of red and blue pixels are on green bayer locations. The notation is 1.6 fixed point. 64 is the largest allowable number which represents 1.0. When added to FFNR alpha must equal 1.0. Legal values: [0,64].
13234 R0x000033B2	31:0	0x000A	ffnr_mix_thresh_y (R/W)
	FFNR - Threshold of Luma for mix to select HF. Pedestal for the threshold which feeds smooth transition demosaic. The luma must be bigger than this number plus the luma gain times luma to use hf_demosaic pixels. Legal values: [0,255].		
13236 R0x000033B4	31:0	0x0008	ffnr_mix_thresh_y_gain (R/W)
	FFNR - Gain applied to luma edge calculation. This number is multiplied times luma to determine the luma threshold for hf_demosaic pixels to be used in FFNR MIX's output. Legal values: [0,255].		
13238 R0x000033B6	31:0	0x0000	ffnr_mix_thresh_gain (R/W)
	FFNR - Gain applied to RGB deltas. Set this value to 0x00 to select only flat field demosaic. Set this value to 0xFF to select only high frequency demosaic. Other values mix the two outputs. Legal values: [0,255].		
13242 R0x000033BA	31:0	0x0000	apedge_control (R/W)
	31:8	X	Reserved
	7:4	0x0000	apedge_adj_ratio Adj_ratio which controls the virtual kernel size for "New Legacy Aperture" Legal values: [0,8].
	3:0	0x0000	apedge_kg KG mixing coefficient for "New Legacy Aperture" Legal values: [0,8].
	Control fields for new X40 APEDGE algorithm.		
13246 R0x000033BE	31:0	0x0008	ua_knee_l (R/W)
	UA - Legacy aperture knee point This value is subtracted from the legacy aperture value when calculationg the unified aperture value. Legal values: [0,1023].		

Table 44: 1: SOC1 Registers (continued)

Register Dec (Hex)	Bits	Default	Name
13250 R0x000033C2	31:0	0x1100	ua_weights (R/W)
	31:16	X	Reserved
	15:12	0x0001	ua_kp1 UA - legacy positive aperture slope This value sets the slope of the line determining the output of unified aperture for positive legacy values. A 1.3 fixed point notation. Must not be set greater than 1.0 (8) Legal values: [0,8].
	11:8	0x0001	ua_kn1 UA - legacy negative aperture slope This value sets the slope of the line determining the output of unified aperture for negative legacy values. A 1.3 fixed point notation. Must not be set greater than 1.0 (8) Legal values: [0,8].
	7:0	X	Reserved

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13412 R0x00003464	31:0	0x0002	clip3_config (R/W)
	31:2	X	Reserved
	1:0	0x0002	clip3_select 0: luma before gamma 1: minRGB before gamma 2:maxRGB before gamma Legal values: [0, 2].
13414 R0x00003466	31:0	0x0000	clip3_min (R/W)
	Min threshold for clip3 Legal values: [0, 4095].		
13416 R0x00003468	31:0	0x0FFF	clip3_max (R/W)
	Max threshold for clip3 Legal values: [0, 4095].		
13418 R0x0000346A	31:0	0x0000	clip3_win_x_start (R/W)
	Starting column of clip3 window. Legal values: [0, 4095].		
13420 R0x0000346C	31:0	0x0000	clip3_win_y_start (R/W)
	Starting row of clip3 window. Legal values: [0, 2047].		
13422 R0x0000346E	31:0	0x0280	clip3_win_x_end (R/W)
	Ending column of clip3 window. Legal values: [0, 4095].		
13424 R0x00003470	31:0	0x01E0	clip3_win_y_end (R/W)
	Ending row of clip3 window. Legal values: [0,2047].		
13426 R0x00003472	31:0	0x0000	clip3_cnt_lo (RO)
	Low 16bit of clip3 counter Read-only. Volatile. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13428 R0x00003474	31:0	0x0000	clip3_cnt_hi (RO)
	High 8bit of clip3 counter Read-only. Volatile. Legal values: [0, 255].		
13430 R0x00003476	31:0	0x0010	af_zone_x_start (R/W)
	Starting column of AF window. Must be 16 or greater. Legal values: [16, 4095].		
13432 R0x00003478	31:0	0x0000	af_zone_y_start (R/W)
	Starting row of AF window. Legal values: [0, 2047].		
13434 R0x0000347A	31:0	0x00A0	af_zone_width (R/W)
	Width of one AF zone, of the five zones within the window. Legal values: [0, 4095].		
13436 R0x0000347C	31:0	0x0078	af_zone_height (R/W)
	Height of AF zone/window. Legal values: [0, 2047].		
13438 R0x0000347E	31:0	0x003F	af_zones_oob_ctrl (R/W)
	31:6	X	Reserved
	5	0x0001	af_zones_oob_mask_en Enable masking of out-of-bounds zones for AF/DAF.
	4:0	RO	af_zones_oob_status Out-of-bounds status bits for AF/DAF zones. [4] = zone 4 exited [3] = zone 3 exited [2] = zone 2 exited [1] = zone 1 exited [0] = zone 0 exited If a zone is exited, then we know that the whole zone is in bounds. Otherwise, we know that the zone is partially or completely out of bounds. There is no way for a zone to be exited that has not been entered because the initial point cannot be less than (0, 0). Read-only. Volatile. Legal values: [0, 31].
13440 R0x00003480	31:0	0x0002	clip1_config (R/W)
	31:3	X	Reserved
	2:0	0x0002	clip1_select 0: green after PGA 1: all colors after PGA 2: luma before gamma 3: minRGB before gamma 4: maxRGB before gamma Legal values: [0, 4].
13442 R0x00003482	31:0	0x0002	clip2_config (R/W)
	31:2	X	Reserved
	1:0	0x0002	clip2_select 0: luma before gamma 1: minRGB before gamma 2:maxRGB before gamma Legal values: [0, 2].
13446 R0x00003486	31:0	0x0000	af_lum_0 (RO)
	Sum of luminance in AF window, zone 0 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13448 R0x00003488	31:0	0x0000	af_lum_1 (RO)
	Sum of luminance in AF window, zone 1 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13450 R0x0000348A	31:0	0x0000	af_lum_2 (RO)
	Sum of luminance in AF window, zone 2 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13452 R0x0000348C	31:0	0x0000	af_lum_3 (RO)
	Sum of luminance in AF window, zone 3 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13454 R0x0000348E	31:0	0x0000	af_lum_4 (RO)
	Sum of luminance in AF window, zone 4 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13458 R0x00003492	31:0	0x100F	af_config_filters (R/W)
	31:13	X	Reserved
	12	0x0001	af_mm_res AF min/max resolution When calculating local contrast: 0 = use inner five fifo subtotals 1 = use all seven fifo subtotals
	11:8	0x0000	af_filter2_scale How much to scale the resulting sharpness values by. When greater than 10, scale by pow(2,af_filter2_scale-10), otherwise scale by pow(2,-af_filter2_scale). Legal values: [0, 15].
	7:4	0x0000	af_filter1_scale How much to scale the resulting sharpness values by. When greater than 10, scale by pow(2,af_filter1_scale-10), otherwise scale by pow(2,-af_filter1_scale). Legal values: [0, 15].
	3	0x0001	af_filter2_symmetric When set, the second AF filter is symmetrical
	2	0x0001	af_filter2_odd When set, the second AF filter is odd
	1	0x0001	af_filter1_symmetric When set, the first AF filter is symmetrical
	0	0x0001	af_filter1_odd When set, the first AF filter is odd
13460 R0x00003494	31:0	0x0100	af_config_thresholds_1 (R/W)
	31:16	X	Reserved
	15:8	0x0001	af_thresh_scale Sharpness values less than this threshold are not used to calculate the zone sharpness af_threshold = (af_thresh_scale * avg_luma)/256 + af_thresh_offset Legal values: [0, 255].
	7:0	0x0000	af_thresh_offset Sharpness values less than this threshold are not used to calculate the zone sharpness af_threshold = (af_thresh_scale * avg_luma)/256 + af_thresh_offset Legal values: [0, 255].

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13466 R0x0000349A	31:0	0x000C	af_config_coeffs_0 (R/W)
	31:15	X	Reserved
	14:10	0x0000	af_filter1_c2 Coefficient for first AF filter Legal values: [-15, 15].
	9:5	0x0000	af_filter1_c1 Coefficient for first AF filter Legal values: [-15, 15].
	4:0	0x000C	af_filter1_c0 Coefficient for first AF filter Legal values: [-15, 15].
13468 R0x0000349C	31:0	0x0000	af_config_coeffs_1 (R/W)
	31:15	X	Reserved
	14:10	0x0000	af_filter1_c5 Coefficient for first AF filter Legal values: [-15, 15].
	9:5	0x0000	af_filter1_c4 Coefficient for first AF filter Legal values: [-15, 15].
	4:0	0x0000	af_filter1_c3 Coefficient for first AF filter Legal values: [-15, 15].
13470 R0x0000349E	31:0	0x0000	af_config_coeffs_2 (R/W)
	31:10	X	Reserved
	9:5	0x0000	af_filter1_c7 Coefficient for first AF filter Legal values: [-15, 15].
	4:0	0x0000	af_filter1_c6 Coefficient for first AF filter Legal values: [-15, 15].
13472 R0x000034A0	31:0	0x0002	af_config_coeffs_3 (R/W)
	31:15	X	Reserved
	14:10	0x0000	af_filter2_c2 Coefficient for second AF filter Legal values: [-15, 15].
	9:5	0x0000	af_filter2_c1 Coefficient for second AF filter Legal values: [-15, 15].
	4:0	0x0002	af_filter2_c0 Coefficient for second AF filter Legal values: [-15, 15].

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13474 R0x000034A2	31:0	0x0800	af_config_coeffs_4 (R/W)
	31:15	X	Reserved
	14:10	0x0002	af_filter2_c5 Coefficient for second AF filter Legal values: [-15, 15].
	9:5	0x0000	af_filter2_c4 Coefficient for second AF filter Legal values: [-15, 15].
	4:0	0x0000	af_filter2_c3 Coefficient for second AF filter Legal values: [-15, 15].
13476 R0x000034A4	31:0	0x0000	af_config_coeffs_5 (R/W)
	31:10	X	Reserved
	9:5	0x0000	af_filter2_c7 Coefficient for second AF filter Legal values: [-15, 15].
	4:0	0x0000	af_filter2_c6 Coefficient for second AF filter Legal values: [-15, 15].
13478 R0x000034A6	31:0	0x0000	af_filter1_sharp_0 (RO)
	Sum of sharpness values for first AF filter, AF window, zone 0 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13480 R0x000034A8	31:0	0x0000	af_filter1_sharp_1 (RO)
	Sum of sharpness values for first AF filter, AF window, zone 1 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13482 R0x000034AA	31:0	0x0000	af_filter1_sharp_2 (RO)
	Sum of sharpness values for first AF filter, AF window, zone 2 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13484 R0x000034AC	31:0	0x0000	af_filter1_sharp_3 (RO)
	Sum of sharpness values for first AF filter, AF window, zone 3 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13486 R0x000034AE	31:0	0x0000	af_filter1_sharp_4 (RO)
	Sum of sharpness values for first AF filter, AF window, zone 4 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13488 R0x000034B0	31:0	0x0000	af_filter1_oflow_cnt (RO)
	Number of pixels within AF window causing overflow of AF filter1 Read-only. Volatile. Legal values: [0, 65535].		
13506 R0x000034C2	31:0	0x0000	af_filter2_sharp_0 (RO)
	Sum of sharpness values for second AF filter, AF window, zone 0 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13508 R0x000034C4	31:0	0x0000	af_filter2_sharp_1 (RO)
	Sum of sharpness values for second AF filter, AF window, zone 1 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13510 R0x000034C6	31:0	0x0000	af_filter2_sharp_2 (RO)
	Sum of sharpness values for second AF filter, AF window, zone 2 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13512 R0x000034C8	31:0	0x0000	af_filter2_sharp_3 (RO)
	Sum of sharpness values for second AF filter, AF window, zone 3 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13514 R0x000034CA	31:0	0x0000	af_filter2_sharp_4 (RO)
	Sum of sharpness values for second AF filter, AF window, zone 4 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13516 R0x000034CC	31:0	0x0000	af_filter2_oflow_cnt (RO)
	Number of pixels within AF window causing overflow of AF filter2 Read-only. Volatile. Legal values: [0, 65535].		
13518 R0x000034CE	31:0	0x7C01	daf_config_0 (R/W)
	31:15	X	Reserved
	14:10	0x001F	daf_coeff_a3 Diagonal AF coefficient a3 Legal values: [-16, 15].
	9:5	0x0000	daf_coeff_a2 Diagonal AF coefficient a2 Legal values: [-16, 15].
	4:0	0x0001	daf_coeff_a1 Diagonal AF coefficient a1 Legal values: [-16, 15].
13520 R0x000034D0	31:0	0x0000	daf_config_1 (R/W)
	31:12	X	Reserved
	11:8	0x0000	daf_scale Diagonal AF filter sharpness scaler setting. Legal values: [0, 15].
	7:2	X	Reserved
	1	0x0000	daf_odd_en Configure diagonal AF filter as odd.
	0	0x0000	daf_asym_en Configure diagonal AF filter as asymmetric.
13522 R0x000034D2	31:0	0x0000	daf_filter_oflow_cnt (RO)
	Diagonal AF sharpness overflow counter. Read-only. Volatile. Legal values: [0, 65535].		
13524 R0x000034D4	31:0	0x0000	daf_filter_sharp_0 (RO)
	Sum of sharpness values for DAF filter, DAF window, zone 0 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13526 R0x000034D6	31:0	0x0000	daf_filter_sharp_1 (RO)
	Sum of sharpness values for DAF filter, DAF window, zone 1 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13528 R0x000034D8	31:0	0x0000	daf_filter_sharp_2 (RO)
	Sum of sharpness values for DAF filter, DAF window, zone 2 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13530 R0x000034DA	31:0	0x0000	daf_filter_sharp_3 (RO)
	Sum of sharpness values for DAF filter, DAF window, zone 3 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		
13532 R0x000034DC	31:0	0x0000	daf_filter_sharp_4 (RO)
	Sum of sharpness values for DAF filter, DAF window, zone 4 (PMFP {3+13}) Read-only. Volatile. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13534 R0x000034DE	31:0	0x0100	daf_config_thresholds_1 (R/W)
	31:16	X	Reserved
	15:8	0x0001	daf_thresh_scale DAF threshold scale parameter daf_threshold = (daf_thresh_scale * avg_green)/256 + daf_thresh_offset Legal values: [0, 255].
	7:0	0x0000	daf_thresh_offset DAF threshold offset parameter daf_threshold = (daf_thresh_scale * avg_green)/256 + daf_thresh_offset Legal values: [0, 255].
13632 R0x00003540	31:0	0x0001	enable_tonal_curve (R/W)
	0- Disable Tonal Curve 1- Enable Tonal Curve		
13634 R0x00003542	31:0	0x0080	tonal_x0 (R/W)
	Tonal Curve Control Point X0 Position Legal values: [1, 1017].		
13636 R0x00003544	31:0	0x0100	tonal_x1 (R/W)
	Tonal Curve Control Point X1 Position Legal values: [2, 1018].		
13638 R0x00003546	31:0	0x0180	tonal_x2 (R/W)
	Tonal Curve Control Point X2 Position Legal values: [3, 1019].		
13640 R0x00003548	31:0	0x0200	tonal_x3 (R/W)
	Tonal Curve Control Point X3 Position Legal values: [4, 1020].		
13642 R0x0000354A	31:0	0x0280	tonal_x4 (R/W)
	Tonal Curve Control Point X4 Position Legal values: [5, 1021].		
13644 R0x0000354C	31:0	0x0300	tonal_x5 (R/W)
	Tonal Curve Control Point X5 Position Legal values: [6, 1022].		
13646 R0x0000354E	31:0	0x0380	tonal_x6 (R/W)
	Tonal Curve Control Point X6 Position Legal values: [7, 1023].		
13648 R0x00003550	31:0	0x0080	tonal_y0 (R/W)
	Tonal Curve Control Point Y0 Position Legal values: [0, 1023].		
13650 R0x00003552	31:0	0x0100	tonal_y1 (R/W)
	Tonal Curve Control Point Y1 Position Legal values: [0, 1023].		
13652 R0x00003554	31:0	0x0180	tonal_y2 (R/W)
	Tonal Curve Control Point Y2 Position Legal values: [0, 1023].		
13654 R0x00003556	31:0	0x0200	tonal_y3 (R/W)
	Tonal Curve Control Point Y3 Position Legal values: [0, 1023].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13656 R0x00003558	31:0	0x0280	tonal_y4 (R/W)
	Tonal Curve Control Point Y4 Position Legal values: [0, 1023].		
13658 R0x0000355A	31:0	0x0300	tonal_y5 (R/W)
	Tonal Curve Control Point Y5 Position Legal values: [0, 1023].		
13660 R0x0000355C	31:0	0x0380	tonal_y6 (R/W)
	Tonal Curve Control Point Y6 Position Legal values: [0, 1023].		
13664 R0x00003560	31:0	0xF100	reciprocal_of_x0_minus_zero (R/W)
	31:16	X	Reserved
	15:12	0x000F	reciprocal_of_x0_minus_zero_exponent Exponent of Reciprocal of (Control Point X0 minus Zero) Legal values: [0,15].
	11	X	Reserved
	10:0	0x0100	reciprocal_of_x0_minus_zero_mantissa Mantissa of Reciprocal of (Control Point X0 minus Zero) * 2^exponent Legal values: [0,2047].
13666 R0x00003562	31:0	0xF100	reciprocal_of_x1_minus_x0 (R/W)
	31:16	X	Reserved
	15:12	0x000F	reciprocal_of_x1_minus_x0_exponent Exponent of Reciprocal of (Control Point X1 minus X0) Legal values: [0,15].
	11	X	Reserved
	10:0	0x0100	reciprocal_of_x1_minus_x0_mantissa Mantissa of Reciprocal of (Control Point X1 minus X0) * 2^exponent Legal values: [0,2047].
13668 R0x00003564	31:0	0xF100	reciprocal_of_x2_minus_x1 (R/W)
	31:16	X	Reserved
	15:12	0x000F	reciprocal_of_x2_minus_x1_exponent Exponent of Reciprocal of (Control Point X2 minus X1) Legal values: [0,15].
	11	X	Reserved
	10:0	0x0100	reciprocal_of_x2_minus_x1_mantissa Mantissa of Reciprocal of (Control Point X2 minus X1) * 2^exponent Legal values: [0,2047].
13670 R0x00003566	31:0	0xF100	reciprocal_of_x3_minus_x2 (R/W)
	31:16	X	Reserved
	15:12	0x000F	reciprocal_of_x3_minus_x2_exponent Exponent of Reciprocal of (Control Point X3 minus X2) Legal values: [0,15].
	11	X	Reserved
	10:0	0x0100	reciprocal_of_x3_minus_x2_mantissa Mantissa of Reciprocal of (Control Point X3 minus X2) * 2^exponent Legal values: [0,2047].

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13672 R0x00003568	31:0	0xF100	reciprocal_of_x4_minus_x3 (R/W)
	31:16	X	Reserved
	15:12	0x000F	reciprocal_of_x4_minus_x3_exponent Exponent of Reciprocal of (Control Point X4 minus X3) Legal values: [0,15].
	11	X	Reserved
	10:0	0x0100	reciprocal_of_x4_minus_x3_mantissa Mantissa of Reciprocal of (Control Point X4 minus X3) * 2^exponent Legal values: [0,2047].
13674 R0x0000356A	31:0	0xF100	reciprocal_of_x5_minus_x4 (R/W)
	31:16	X	Reserved
	15:12	0x000F	reciprocal_of_x5_minus_x4_exponent Exponent of Reciprocal of (Control Point X5 minus X4) Legal values: [0,15].
	11	X	Reserved
	10:0	0x0100	reciprocal_of_x5_minus_x4_mantissa Mantissa of Reciprocal of (Control Point X5 minus X4) * 2^exponent Legal values: [0,2047].
13676 R0x0000356C	31:0	0xF100	reciprocal_of_x6_minus_x5 (R/W)
	31:16	X	Reserved
	15:12	0x000F	reciprocal_of_x6_minus_x5_exponent Exponent of Reciprocal of (Control Point X6 minus X5) Legal values: [0,15].
	11	X	Reserved
	10:0	0x0100	reciprocal_of_x6_minus_x5_mantissa Mantissa of Reciprocal of (Control Point X6 minus X5) * 2^exponent Legal values: [0,2047].
13678 R0x0000356E	31:0	0xF100	reciprocal_of_400_minus_x6 (R/W)
	31:16	X	Reserved
	15:12	0x000F	reciprocal_of_400_minus_x6_exponent Exponent of Reciprocal of (0x400 minus Control Point X6) Legal values: [0,15].
	11	X	Reserved
	10:0	0x0100	reciprocal_of_400_minus_x6_mantissa Mantissa of Reciprocal of (0x400 minus Control Point X6) * 2^exponent Legal values: [0,2047].
13686 R0x00003576	31:0	0x0000	ae_zone_x_start (R/W)
	First AE zone start in horizontal direction Volatile. Legal values: [0, 4095].		
13688 R0x00003578	31:0	0x0000	ae_zone_y_start (R/W)
	First AE zone start in vertical direction Volatile. Legal values: [0, 2047].		
13692 R0x0000357C	31:0	0x0080	ae_zone_width (R/W)
	Width of one AE zone, of the six zones within the window. Volatile. Legal values: [0, 4095].		
13694 R0x0000357E	31:0	0x0060	ae_zone_height (R/W)
	Height of AE zone/window. Volatile. Legal values: [0, 2047].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13696 R0x00003580	31:0	0x007F	ae_zones_oob_ctrl (R/W)
	31:7	X	Reserved
	6	0x0001	ae_zones_oob_mask_en Enable masking of out-of-bounds zones for AE.
	5:0	RO	ae_zones_oob_status Out-of-bounds status bits for AE zones. [5] = zone 5 exited [4] = zone 4 exited [3] = zone 3 exited [2] = zone 2 exited [1] = zone 1 exited [0] = zone 0 exited If a zone is exited, then we know that the whole zone is in bounds. Otherwise, we know that the zone is partially or completely out of bounds. There is no way for a zone to be exited that has not been entered because the initial point cannot be less than (0, 0). Read-only. Volatile. Legal values: [0, 63].
13698 R0x00003582	31:0	0x0000	ae_sum_0 (RO)
	Sum of luminance in the AE window for zone 0, PMFP{3+13}. Read-only. Volatile. Legal values: [0, 65535].		
13700 R0x00003584	31:0	0x0000	ae_sum_1 (RO)
	Sum of luminance in the AE window for zone 1, PMFP{3+13}. Read-only. Volatile. Legal values: [0, 65535].		
13702 R0x00003586	31:0	0x0000	ae_sum_2 (RO)
	Sum of luminance in the AE window for zone 2, PMFP{3+13}. Read-only. Volatile. Legal values: [0, 65535].		
13704 R0x00003588	31:0	0x0000	ae_sum_3 (RO)
	Sum of luminance in the AE window for zone 3, PMFP{3+13}. Read-only. Volatile. Legal values: [0, 65535].		
13706 R0x0000358A	31:0	0x0000	ae_sum_4 (RO)
	Sum of luminance in the AE window for zone 4, PMFP{3+13}. Read-only. Volatile. Legal values: [0, 65535].		
13708 R0x0000358C	31:0	0x0000	ae_sum_5 (RO)
	Sum of luminance in the AE window for zone 5, PMFP{3+13}. Read-only. Volatile. Legal values: [0, 65535].		
13710 R0x0000358E	31:0	0x0001	hist_pga_data_select (R/W)
	This bit selects the input for Histo_1A, Histo_1B, and Clip_1. 0: select lower 12 of 14 bits from PGA output 1: select upper 12 of 14 bits from PGA output		
13712 R0x00003590	31:0	0x000A	hist1b_bin_offset (R/W)
	Starting value for third bin Legal values: [0, 4095].		
13714 R0x00003592	31:0	0x000A	hist1a_bin_offset (R/W)
	Starting value for second bin Legal values: [0, 4095].		
13716 R0x00003594	31:0	0x030A	hist0_pre_divider (R/W)
	Pre-divider for hist0 bins Legal values: [1, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13718 R0x00003596	31:0	0x030A	hist1a_pre_divider (R/W)
	Pre-divider for hist1a bins Legal values: [1, 65535].		
13720 R0x00003598	31:0	0x030A	hist1b_pre_divider (R/W)
	Pre-divider for hist1b bins Legal values: [1, 65535].		
13722 R0x0000359A	31:0	0x000D	hist0_bin_config (R/W)
	31:8	X	Reserved
	7:5	0x0000	hist0_bin_select 0: green before demosaic 1: all colors before demosaic 2: luma afer demosaic 3: minRGB after demosaic 4: maxRGB after demosaic Legal values: [0, 4].
	4	X	Reserved
	3:0	0x000D	hist0_bin_width Bin width is pow(2,hist0_bin_width) Legal values: [0,15].
13724 R0x0000359C	31:0	0x000A	hist1a_bin_config (R/W)
	31:5	X	Reserved
	4	0x0000	hist1a_bin_select 0: green 1: all colors
	3:0	0x000A	hist1a_bin_width Bin width is pow(2,hist1a_bin_width) Legal values: [0, 15].
13726 R0x0000359E	31:0	0x0008	hist1b_bin_config (R/W)
	31:5	X	Reserved
	4	0x0000	hist1b_bin_select 0: green 1: all colors
	3:0	0x0008	hist1b_bin_width Bin width is pow(2,hist1b_bin_width) Legal values: [0, 15].
13728 R0x000035A0	31:0	0x000A	hist0_bin_offset (R/W)
	Starting value for first bin Legal values: [0, 4095].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13730 R0x000035A2	31:0	0x0014	dark_color_kill_controls (R/W)
	31:8	X	Reserved
	7	0x0000	en_dark_kl if set to 1, enable dark color kill
	6	0x0000	sel_yd if set to 1, use luma as min/max value
	5:3	0x0002	cktrh_d dark color kill threshold Legal values: [0, 7].
	2:0	0x0004	ckgain_d dark color kill gain Legal values: [0, 7].
13732 R0x000035A4	31:0	0x0594	bright_color_kill_controls (R/W)
	31:11	X	Reserved
	10	0x0001	en_kl if set to 1, enable bright color kill
	9	0x0000	sel_y if set to 1, use luma as min/max value
	8:6	0x0006	cktrh bright color kill threshold Legal values: [0, 7].
	5:3	0x0002	ckgain bright color kill gain Legal values: [0, 7].
	2:0	0x0004	cksat color kill saturation point Legal values: [0, 7].
13736 R0x000035A8	31:0	0x01E0	clip2_win_y_end (R/W)
	Ending row of clip2 window. Legal values: [0,2047].		
13738 R0x000035AA	31:0	0x0000	clip1_min (R/W)
	Min threshold for clip1 Legal values: [0,4095].		
13740 R0x000035AC	31:0	0x0FFF	clip1_max (R/W)
	Max threshold for clip1 Legal values: [0, 4095].		
13742 R0x000035AE	31:0	0x0000	clip2_min (R/W)
	Min threshold for clip2 Legal values: [0, 4095].		
13744 R0x000035B0	31:0	0x0FFF	clip2_max (R/W)
	Max threshold for clip2 Legal values: [0, 4095].		
13746 R0x000035B2	31:0	0x0280	clip2_win_x_end (R/W)
	Ending column of clip2 window. Legal values: [0, 4095].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13748 R0x000035B4	31:0	0x0000	hist0_bin_stats1 (RO)
	31:16	X	Reserved
	15:8	RO	hist0_bin_avg_1 Number of pixels in bin 1 of first set of bin stats divided by hist0_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist0_bin_avg_0 Number of pixels in bin 0 of first set of bin stats divided by hist0_pre_divider Read-only. Volatile. Legal values: [0, 255].
13750 R0x000035B6	31:0	0x0000	hist0_bin_stats2 (RO)
	31:16	X	Reserved
	15:8	RO	hist0_bin_avg_3 Number of pixels in bin 3 of first set of bin stats divided by hist0_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist0_bin_avg_2 Number of pixels in bin 2 of first set of bin stats divided by hist0_pre_divider Read-only. Volatile. Legal values: [0, 255].
13752 R0x000035B8	31:0	0x0000	hist0_bin_stats3 (RO)
	31:16	X	Reserved
	15:8	RO	hist0_bin_avg_5 Number of pixels in bin 5 of first set of bin stats divided by hist0_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist0_bin_avg_4 Number of pixels in bin 4 of first set of bin stats divided by hist0_pre_divider Read-only. Volatile. Legal values: [0, 255].
13754 R0x000035BA	31:0	0x0000	hist0_bin_stats4 (RO)
	31:16	X	Reserved
	15:8	RO	hist0_bin_avg_7 Number of pixels in bin 7 of first set of bin stats divided by hist0_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist0_bin_avg_6 Number of pixels in bin 6 of first set of bin stats divided by hist0_pre_divider Read-only. Volatile. Legal values: [0, 255].
13756 R0x000035BC	31:0	0x0000	hist1a_bin_stats1 (RO)
	31:16	X	Reserved
	15:8	RO	hist1a_bin_avg_1 Number of pixels in bin 1 of first set of bin stats divided by hist1a_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist1a_bin_avg_0 Number of pixels in bin 0 of first set of bin stats divided by hist1a_pre_divider Read-only. Volatile. Legal values: [0, 255].
13758 R0x000035BE	31:0	0x0000	hist1a_bin_stats2 (RO)
	31:16	X	Reserved
	15:8	RO	hist1a_bin_avg_3 Number of pixels in bin 3 of first set of bin stats divided by hist1a_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist1a_bin_avg_2 Number of pixels in bin 2 of first set of bin stats divided by hist1a_pre_divider Read-only. Volatile. Legal values: [0, 255].

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13760 R0x000035C0	31:0	0x0000	hist1a_bin_stats3 (RO)
	31:16	X	Reserved
	15:8	RO	hist1a_bin_avg_5 Number of pixels in bin 5 of first set of bin stats divided by hist1a_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist1a_bin_avg_4 Number of pixels in bin 4 of first set of bin stats divided by hist1a_pre_divider Read-only. Volatile. Legal values: [0, 255].
13762 R0x000035C2	31:0	0x0000	hist1a_bin_stats4 (RO)
	31:16	X	Reserved
	15:8	RO	hist1a_bin_avg_7 Number of pixels in bin 7 of first set of bin stats divided by hist1a_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist1a_bin_avg_6 Number of pixels in bin 6 of first set of bin stats divided by hist1a_pre_divider Read-only. Volatile. Legal values: [0, 255].
13764 R0x000035C4	31:0	0x0000	hist1b_bin_stats1 (RO)
	31:16	X	Reserved
	15:8	RO	hist1b_bin_avg_1 Number of pixels in bin 1 of first set of bin stats divided by hist1b_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist1b_bin_avg_0 Number of pixels in bin 0 of first set of bin stats divided by hist1b_pre_divider Read-only. Volatile. Legal values: [0, 255].
13766 R0x000035C6	31:0	0x0000	hist1b_bin_stats2 (RO)
	31:16	X	Reserved
	15:8	RO	hist1b_bin_avg_3 Number of pixels in bin 3 of first set of bin stats divided by hist1b_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist1b_bin_avg_2 Number of pixels in bin 2 of first set of bin stats divided by hist1b_pre_divider Read-only. Volatile. Legal values: [0, 255].
13768 R0x000035C8	31:0	0x0000	hist1b_bin_stats3 (RO)
	31:16	X	Reserved
	15:8	RO	hist1b_bin_avg_5 Number of pixels in bin 5 of first set of bin stats divided by hist1b_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist1b_bin_avg_4 Number of pixels in bin 4 of first set of bin stats divided by hist1b_pre_divider Read-only. Volatile. Legal values: [0, 255].
13770 R0x000035CA	31:0	0x0000	hist1b_bin_stats4 (RO)
	31:16	X	Reserved
	15:8	RO	hist1b_bin_avg_7 Number of pixels in bin 7 of first set of bin stats divided by hist1b_pre_divider Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	hist1b_bin_avg_6 Number of pixels in bin 6 of first set of bin stats divided by hist1b_pre_divider Read-only. Volatile. Legal values: [0, 255].

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13772 R0x000035CC	31:0	0x0000	clip2_win_x_start (R/W)
	Starting column of clip2 window. Legal values: [0, 4095].		
13774 R0x000035CE	31:0	0x0000	clip2_win_y_start (R/W)
	Starting row of clip2 window. Legal values: [0, 2047].		
13776 R0x000035D0	31:0	0x0000	hist_win_x_start (R/W)
	Starting column of histogram window. Legal values: [0, 4095].		
13778 R0x000035D2	31:0	0x0000	hist_win_y_start (R/W)
	Starting row of histogram window. Legal values: [0, 2047].		
13780 R0x000035D4	31:0	0x0280	hist_win_x_end (R/W)
	Ending column of histogram window. Legal values: [0, 4095].		
13782 R0x000035D6	31:0	0x01E0	hist_win_y_end (R/W)
	Ending row of histogram window. Legal values: [0, 2047].		
13784 R0x000035D8	31:0	0x0000	clip1_cnt_lo (RO)
	Low 16bit of clip1 counter Read-only. Volatile. Legal values: [0, 65535].		
13786 R0x000035DA	31:0	0x0000	clip1_cnt_hi (RO)
	High 8bit of clip1 counter Read-only. Volatile. Legal values: [0, 255].		
13792 R0x000035E0	31:0	0x0000	clip2_cnt_lo (RO)
	Low 16bit of clip2 counter Read-only. Volatile. Legal values: [0, 65535].		
13794 R0x000035E2	31:0	0x0000	clip2_cnt_hi (RO)
	High 8bit of clip2 counter Read-only. Volatile. Legal values: [0, 255].		
13796 R0x000035E4	31:0	0x0000	clip1_win_x_start (R/W)
	Starting column of clip1 window. Legal values: [0, 4095].		
13798 R0x000035E6	31:0	0x0000	clip1_win_y_start (R/W)
	Starting row of clip1 window. Legal values: [0, 2047].		
13800 R0x000035E8	31:0	0x0280	clip1_win_x_end (R/W)
	Ending column of clip1 window. Legal values: [0, 4095].		
13802 R0x000035EA	31:0	0x01E0	clip1_win_y_end (R/W)
	Ending row of clip1 window. Legal values: [0, 2047].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13828 R0x00003604	31:0	0x1B00	r_gamma_curve_knees_0_1 (R/W)
	31:16	X	Reserved
	15:8	0x001B	r_gamma_curve_knees_1 Red Gamma Curve Knees 1 Legal values: [0, 255].
	7:0	0x0000	r_gamma_curve_knees_0 Red Gamma Curve Knees 0 Legal values: [0, 255].
	Red Gamma Curve Knees 0 and 1 Note: If the slop of this line segment is negative the entire curve is interpreted as negative slope. There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13830 R0x00003606	31:0	0x4C2E	r_gamma_curve_knees_2_3 (R/W)
	31:16	X	Reserved
	15:8	0x004C	r_gamma_curve_knees_3 Red Gamma Curve Knees 3 Legal values: [0, 255].
	7:0	0x002E	r_gamma_curve_knees_2 Red Gamma Curve Knees 2 Legal values: [0, 255].
	Red Gamma Curve Knees 2 and 3 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13832 R0x00003608	31:0	0x9878	r_gamma_curve_knees_4_5 (R/W)
	31:16	X	Reserved
	15:8	0x0098	r_gamma_curve_knees_5 Red Gamma Curve Knees 5 Legal values: [0, 255].
	7:0	0x0078	r_gamma_curve_knees_4 Red Gamma Curve Knees 4 Legal values: [0, 255].
	Red Gamma Curve Knees 4 and 5 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13834 R0x0000360A	31:0	0xC1B0	r_gamma_curve_knees_6_7 (R/W)
	31:16	X	Reserved
	15:8	0x00C1	r_gamma_curve_knees_7 Red Gamma Curve Knees 7 Legal values: [0, 255].
	7:0	0x00B0	r_gamma_curve_knees_6 Red Gamma Curve Knees 6 Legal values: [0, 255].
	Red Gamma Curve Knees 6 and 7 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13836 R0x0000360C	31:0	0xD9CF	r_gamma_curve_knees_8_9 (R/W)
	31:16	X	Reserved
	15:8	0x00D9	r_gamma_curve_knees_9 Red Gamma Curve Knees 9 Legal values: [0, 255].
	7:0	0x00CF	r_gamma_curve_knees_8 Red Gamma Curve Knees 8 Legal values: [0, 255].
	Red Gamma Curve Knees 8 and 9 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13838 R0x0000360E	31:0	0xE8E1	r_gamma_curve_knees_10_11 (R/W)
	31:16	X	Reserved
	15:8	0x00E8	r_gamma_curve_knees_11 Red Gamma Curve Knees 11 Legal values: [0, 255].
	7:0	0x00E1	r_gamma_curve_knees_10 Red Gamma Curve Knees 10 Legal values: [0, 255].
	Red Gamma Curve Knees 10 and 11 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13840 R0x00003610	31:0	0xF2EE	r_gamma_curve_knees_12_13 (R/W)
	31:16	X	Reserved
	15:8	0x00F2	r_gamma_curve_knees_13 Red Gamma Curve Knees 13 Legal values: [0, 255].
	7:0	0x00EE	r_gamma_curve_knees_12 Red Gamma Curve Knees 12 Legal values: [0, 255].
	Red Gamma Curve Knees 12 and 13 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13842 R0x00003612	31:0	0xF9F6	r_gamma_curve_knees_14_15 (R/W)
	31:16	X	Reserved
	15:8	0x00F9	r_gamma_curve_knees_15 Red Gamma Curve Knees 15 Legal values: [0, 255].
	7:0	0x00F6	r_gamma_curve_knees_14 Red Gamma Curve Knees 14 Legal values: [0, 255].
	Red Gamma Curve Knees 14 and 15 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13844 R0x00003614	31:0	0xFDFB	r_gamma_curve_knees_16_17 (R/W)
	31:16	X	Reserved
	15:8	0x00FD	r_gamma_curve_knees_17 Red Gamma Curve Knees 17 Legal values: [0, 255].
	7:0	0x00FB	r_gamma_curve_knees_16 Red Gamma Curve Knees 16 Legal values: [0, 255].
	Red Gamma Curve Knees 16 and 17 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13846 R0x00003616	31:0	0x00FF	r_gamma_curve_knee_18 (R/W)
	Red Gamma Curve Knees 18 Legal values: [0, 255].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13848 R0x00003618	31:0	0x1B00	g_gamma_curve_knees_0_1 (R/W)
	31:16	X	Reserved
	15:8	0x001B	g_gamma_curve_knees_1 Green Gamma Curve Knees 1 Legal values: [0, 255].
	7:0	0x0000	g_gamma_curve_knees_0 Green Gamma Curve Knees 0 Legal values: [0, 255].
	Green Gamma Curve Knees 0 and 1 Note: If the slop of this line segment is negative the entire curve is interpreted as negative slope. There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13850 R0x0000361A	31:0	0x4C2E	g_gamma_curve_knees_2_3 (R/W)
	31:16	X	Reserved
	15:8	0x004C	g_gamma_curve_knees_3 Green Gamma Curve Knees 3 Legal values: [0, 255].
	7:0	0x002E	g_gamma_curve_knees_2 Green Gamma Curve Knees 2 Legal values: [0, 255].
	Green Gamma Curve Knees 2 and 3 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13852 R0x0000361C	31:0	0x9878	g_gamma_curve_knees_4_5 (R/W)
	31:16	X	Reserved
	15:8	0x0098	g_gamma_curve_knees_5 Green Gamma Curve Knees 5 Legal values: [0, 255].
	7:0	0x0078	g_gamma_curve_knees_4 Green Gamma Curve Knees 4 Legal values: [0, 255].
	Green Gamma Curve Knees 4 and 5 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13854 R0x0000361E	31:0	0xC1B0	g_gamma_curve_knees_6_7 (R/W)
	31:16	X	Reserved
	15:8	0x00C1	g_gamma_curve_knees_7 Green Gamma Curve Knees 7 Legal values: [0, 255].
	7:0	0x00B0	g_gamma_curve_knees_6 Green Gamma Curve Knees 6 Legal values: [0, 255].
	Green Gamma Curve Knees 6 and 7 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13856 R0x00003620	31:0	0xD9CF	g_gamma_curve_knees_8_9 (R/W)
	31:16	X	Reserved
	15:8	0x00D9	g_gamma_curve_knees_9 Green Gamma Curve Knees 9 Legal values: [0, 255].
	7:0	0x00CF	g_gamma_curve_knees_8 Green Gamma Curve Knees 8 Legal values: [0, 255].
	Green Gamma Curve Knees 8 and 9 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13858 R0x00003622	31:0	0xE8E1	g_gamma_curve_knees_10_11 (R/W)
	31:16	X	Reserved
	15:8	0x00E8	g_gamma_curve_knees_11 Green Gamma Curve Knees 11 Legal values: [0, 255].
	7:0	0x00E1	g_gamma_curve_knees_10 Green Gamma Curve Knees 10 Legal values: [0, 255].
	Green Gamma Curve Knees 10 and 11 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13860 R0x00003624	31:0	0xF2EE	g_gamma_curve_knees_12_13 (R/W)
	31:16	X	Reserved
	15:8	0x00F2	g_gamma_curve_knees_13 Green Gamma Curve Knees 13 Legal values: [0, 255].
	7:0	0x00EE	g_gamma_curve_knees_12 Green Gamma Curve Knees 12 Legal values: [0, 255].
	Green Gamma Curve Knees 12 and 13 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13862 R0x00003626	31:0	0xF9F6	g_gamma_curve_knees_14_15 (R/W)
	31:16	X	Reserved
	15:8	0x00F9	g_gamma_curve_knees_15 Green Gamma Curve Knees 15 Legal values: [0, 255].
	7:0	0x00F6	g_gamma_curve_knees_14 Green Gamma Curve Knees 14 Legal values: [0, 255].
	Green Gamma Curve Knees 14 and 15 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13864 R0x00003628	31:0	0xFDFB	g_gamma_curve_knees_16_17 (R/W)
	31:16	X	Reserved
	15:8	0x00FD	g_gamma_curve_knees_17 Green Gamma Curve Knees 17 Legal values: [0, 255].
	7:0	0x00FB	g_gamma_curve_knees_16 Green Gamma Curve Knees 16 Legal values: [0, 255].
	Green Gamma Curve Knees 16 and 17 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. Legal values: [0, 65535].		
13866 R0x0000362A	31:0	0x00FF	g_gamma_curve_knee_18 (R/W)
	Green Gamma Curve Knees 18 Legal values: [0, 255].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13868 R0x0000362C	31:0	0x1B00	b_gamma_curve_knees_0_1 (R/W)
	31:16	X	Reserved
	15:8	0x001B	b_gamma_curve_knees_1 Blue Gamma Curve Knees 1 Legal values: [0, 255].
	7:0	0x0000	b_gamma_curve_knees_0 Blue Gamma Curve Knees 0 Legal values: [0, 255].
	Blue Gamma Curve Knees 0 and 1 Note: If the slop of this line segment is negative the entire curve is interpreted as negative slope. There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096.		
13870 R0x0000362E	31:0	0x4C2E	b_gamma_curve_knees_2_3 (R/W)
	31:16	X	Reserved
	15:8	0x004C	b_gamma_curve_knees_3 Blue Gamma Curve Knees 3 Legal values: [0, 255].
	7:0	0x002E	b_gamma_curve_knees_2 Blue Gamma Curve Knees 2 Legal values: [0, 255].
	Blue Gamma Curve Knees 2 and 3 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096.		
13872 R0x00003630	31:0	0x9878	b_gamma_curve_knees_4_5 (R/W)
	31:16	X	Reserved
	15:8	0x0098	b_gamma_curve_knees_5 Blue Gamma Curve Knees 5 Legal values: [0, 255].
	7:0	0x0078	b_gamma_curve_knees_4 Blue Gamma Curve Knees 4 Legal values: [0, 255].
	Blue Gamma Curve Knees 4 and 5 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096.		
13874 R0x00003632	31:0	0xC1B0	b_gamma_curve_knees_6_7 (R/W)
	31:16	X	Reserved
	15:8	0x00C1	b_gamma_curve_knees_7 Blue Gamma Curve Knees 7 Legal values: [0, 255].
	7:0	0x00B0	b_gamma_curve_knees_6 Blue Gamma Curve Knees 6 Legal values: [0, 255].
	Blue Gamma Curve Knees 6 and 7 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096.		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13876 R0x00003634	31:0	0xD9CF	b_gamma_curve_knees_8_9 (R/W)
	31:16	X	Reserved
	15:8	0x00D9	b_gamma_curve_knees_9 Blue Gamma Curve Knees 9 Legal values: [0, 255].
	7:0	0x00CF	b_gamma_curve_knees_8 Blue Gamma Curve Knees 8 Legal values: [0, 255].
	Blue Gamma Curve Knees 8 and 9 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096.		
13878 R0x00003636	31:0	0xE8E1	b_gamma_curve_knees_10_11 (R/W)
	31:16	X	Reserved
	15:8	0x00E8	b_gamma_curve_knees_11 Blue Gamma Curve Knees 11 Legal values: [0, 255].
	7:0	0x00E1	b_gamma_curve_knees_10 Blue Gamma Curve Knees 10 Legal values: [0, 255].
	Blue Gamma Curve Knees 10 and 11 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096.		
13880 R0x00003638	31:0	0xF2EE	b_gamma_curve_knees_12_13 (R/W)
	31:16	X	Reserved
	15:8	0x00F2	b_gamma_curve_knees_13 Blue Gamma Curve Knees 13 Legal values: [0, 255].
	7:0	0x00EE	b_gamma_curve_knees_12 Blue Gamma Curve Knees 12 Legal values: [0, 255].
	Blue Gamma Curve Knees 12 and 13 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096.		
13882 R0x0000363A	31:0	0xF9F6	b_gamma_curve_knees_14_15 (R/W)
	31:16	X	Reserved
	15:8	0x00F9	b_gamma_curve_knees_15 Blue Gamma Curve Knees 15 Legal values: [0, 255].
	7:0	0x00F6	b_gamma_curve_knees_14 Blue Gamma Curve Knees 14 Legal values: [0, 255].
	Blue Gamma Curve Knees 14 and 15 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096.		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13884 R0x0000363C	31:0	0xFDFB	b_gamma_curve_knees_16_17 (R/W)
	31:16	X	Reserved
	15:8	0x00FD	b_gamma_curve_knees_17 Blue Gamma Curve Knees 17 Legal values: [0, 255].
	7:0	0x00FB	b_gamma_curve_knees_16 Blue Gamma Curve Knees 16 Legal values: [0, 255].
	Blue Gamma Curve Knees 16 and 17 There are 19 knee points on the Gamma curve. The knee points are at input data values - 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096.		
13886 R0x0000363E	31:0	0x00FF	b_gamma_curve_knee_18 (R/W)
	Blue Gamma Curve Knees 18 Legal values: [0, 255].		
13888 R0x00003640	31:0	0x0010	p_g1_p0q0 (R/W)
	P0 coefficients for Green1. Legal values: [0, 65535].		
13890 R0x00003642	31:0	0x0000	p_g1_p0q1 (R/W)
	P0 coefficients for Green1. Legal values: [0, 65535].		
13892 R0x00003644	31:0	0x0000	p_g1_p0q2 (R/W)
	P0 coefficients for Green1. Legal values: [0, 65535].		
13894 R0x00003646	31:0	0x0000	p_g1_p0q3 (R/W)
	P0 coefficients for Green1. Legal values: [0, 65535].		
13896 R0x00003648	31:0	0x0000	p_g1_p0q4 (R/W)
	P0 coefficients for Green1. Legal values: [0, 65535].		
13898 R0x0000364A	31:0	0x0010	p_r_p0q0 (R/W)
	P0 coefficients for Red. Legal values: [0, 65535].		
13900 R0x0000364C	31:0	0x0000	p_r_p0q1 (R/W)
	P0 coefficients for Red. Legal values: [0, 65535].		
13902 R0x0000364E	31:0	0x0000	p_r_p0q2 (R/W)
	P0 coefficients for Red. Legal values: [0, 65535].		
13904 R0x00003650	31:0	0x0000	p_r_p0q3 (R/W)
	P0 coefficients for Red. Legal values: [0, 65535].		
13906 R0x00003652	31:0	0x0000	p_r_p0q4 (R/W)
	P0 coefficients for Red. Legal values: [0, 65535].		
13908 R0x00003654	31:0	0x0010	p_b_p0q0 (R/W)
	P0 coefficients for Blue. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13910 R0x00003656	31:0	0x0000	p_b_p0q1 (R/W)
	P0 coefficients for Blue. Legal values: [0, 65535].		
13912 R0x00003658	31:0	0x0000	p_b_p0q2 (R/W)
	P0 coefficients for Blue. Legal values: [0, 65535].		
13914 R0x0000365A	31:0	0x0000	p_b_p0q3 (R/W)
	P0 coefficients for Blue. Legal values: [0, 65535].		
13916 R0x0000365C	31:0	0x0000	p_b_p0q4 (R/W)
	P0 coefficients for Blue. Legal values: [0, 65535].		
13918 R0x0000365E	31:0	0x0010	p_g2_p0q0 (R/W)
	P0 coefficients for Green2. Legal values: [0, 65535].		
13920 R0x00003660	31:0	0x0000	p_g2_p0q1 (R/W)
	P0 coefficients for Green2. Legal values: [0, 65535].		
13922 R0x00003662	31:0	0x0000	p_g2_p0q2 (R/W)
	P0 coefficients for Green2. Legal values: [0, 65535].		
13924 R0x00003664	31:0	0x0000	p_g2_p0q3 (R/W)
	P0 coefficients for Green2. Legal values: [0, 65535].		
13926 R0x00003666	31:0	0x0000	p_g2_p0q4 (R/W)
	P0 coefficients for Green2. Legal values: [0, 65535].		
13952 R0x00003680	31:0	0x0000	p_g1_p1q0 (R/W)
	P1 coefficients for Green1. Legal values: [0, 65535].		
13954 R0x00003682	31:0	0x0000	p_g1_p1q1 (R/W)
	P1 coefficients for Green1. Legal values: [0, 65535].		
13956 R0x00003684	31:0	0x0000	p_g1_p1q2 (R/W)
	P1 coefficients for Green1. Legal values: [0, 65535].		
13958 R0x00003686	31:0	0x0000	p_g1_p1q3 (R/W)
	P1 coefficients for Green1. Legal values: [0, 65535].		
13960 R0x00003688	31:0	0x0000	p_g1_p1q4 (R/W)
	P1 coefficients for Green1. Legal values: [0, 65535].		
13962 R0x0000368A	31:0	0x0000	p_r_p1q0 (R/W)
	P1 coefficients for Red. Legal values: [0, 65535].		
13964 R0x0000368C	31:0	0x0000	p_r_p1q1 (R/W)
	P1 coefficients for Red. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
13966 R0x0000368E	31:0	0x0000	p_r_p1q2 (R/W)
	P1 coefficients for Red. Legal values: [0, 65535].		
13968 R0x00003690	31:0	0x0000	p_r_p1q3 (R/W)
	P1 coefficients for Red. Legal values: [0, 65535].		
13970 R0x00003692	31:0	0x0000	p_r_p1q4 (R/W)
	P1 coefficients for Red. Legal values: [0, 65535].		
13972 R0x00003694	31:0	0x0000	p_b_p1q0 (R/W)
	P1 coefficients for Blue. Legal values: [0, 65535].		
13974 R0x00003696	31:0	0x0000	p_b_p1q1 (R/W)
	P1 coefficients for Blue. Legal values: [0, 65535].		
13976 R0x00003698	31:0	0x0000	p_b_p1q2 (R/W)
	P1 coefficients for Blue. Legal values: [0, 65535].		
13978 R0x0000369A	31:0	0x0000	p_b_p1q3 (R/W)
	P1 coefficients for Blue. Legal values: [0, 65535].		
13980 R0x0000369C	31:0	0x0000	p_b_p1q4 (R/W)
	P1 coefficients for Blue. Legal values: [0, 65535].		
13982 R0x0000369E	31:0	0x0000	p_g2_p1q0 (R/W)
	P1 coefficients for Green2. Legal values: [0, 65535].		
13984 R0x000036A0	31:0	0x0000	p_g2_p1q1 (R/W)
	P1 coefficients for Green2. Legal values: [0, 65535].		
13986 R0x000036A2	31:0	0x0000	p_g2_p1q2 (R/W)
	P1 coefficients for Green2. Legal values: [0, 65535].		
13988 R0x000036A4	31:0	0x0000	p_g2_p1q3 (R/W)
	P1 coefficients for Green2. Legal values: [0, 65535].		
13990 R0x000036A6	31:0	0x0000	p_g2_p1q4 (R/W)
	P1 coefficients for Green2. Legal values: [0, 65535].		
14016 R0x000036C0	31:0	0x0000	p_g1_p2q0 (R/W)
	P2 coefficients for Green1. Legal values: [0, 65535].		
14018 R0x000036C2	31:0	0x0000	p_g1_p2q1 (R/W)
	P2 coefficients for Green1. Legal values: [0, 65535].		
14020 R0x000036C4	31:0	0x0000	p_g1_p2q2 (R/W)
	P2 coefficients for Green1. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
14022 R0x000036C6	31:0	0x0000	p_g1_p2q3 (R/W)
	P2 coefficients for Green1. Legal values: [0, 65535].		
14024 R0x000036C8	31:0	0x0000	p_g1_p2q4 (R/W)
	P2 coefficients for Green1. Legal values: [0, 65535].		
14026 R0x000036CA	31:0	0x0000	p_r_p2q0 (R/W)
	P2 coefficients for Red. Legal values: [0, 65535].		
14028 R0x000036CC	31:0	0x0000	p_r_p2q1 (R/W)
	P2 coefficients for Red. Legal values: [0, 65535].		
14030 R0x000036CE	31:0	0x0000	p_r_p2q2 (R/W)
	P2 coefficients for Red. Legal values: [0, 65535].		
14032 R0x000036D0	31:0	0x0000	p_r_p2q3 (R/W)
	P2 coefficients for Red. Legal values: [0, 65535].		
14034 R0x000036D2	31:0	0x0000	p_r_p2q4 (R/W)
	P2 coefficients for Red. Legal values: [0, 65535].		
14036 R0x000036D4	31:0	0x0000	p_b_p2q0 (R/W)
	P2 coefficients for Blue. Legal values: [0, 65535].		
14038 R0x000036D6	31:0	0x0000	p_b_p2q1 (R/W)
	P2 coefficients for Blue. Legal values: [0, 65535].		
14040 R0x000036D8	31:0	0x0000	p_b_p2q2 (R/W)
	P2 coefficients for Blue. Legal values: [0, 65535].		
14042 R0x000036DA	31:0	0x0000	p_b_p2q3 (R/W)
	P2 coefficients for Blue. Legal values: [0, 65535].		
14044 R0x000036DC	31:0	0x0000	p_b_p2q4 (R/W)
	P2 coefficients for Blue. Legal values: [0, 65535].		
14046 R0x000036DE	31:0	0x0000	p_g2_p2q0 (R/W)
	P2 coefficients for Green2. Legal values: [0, 65535].		
14048 R0x000036E0	31:0	0x0000	p_g2_p2q1 (R/W)
	P2 coefficients for Green2. Legal values: [0, 65535].		
14050 R0x000036E2	31:0	0x0000	p_g2_p2q2 (R/W)
	P2 coefficients for Green2. Legal values: [0, 65535].		
14052 R0x000036E4	31:0	0x0000	p_g2_p2q3 (R/W)
	P2 coefficients for Green2. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
14054 R0x000036E6	31:0	0x0000	p_g2_p2q4 (R/W)
	P2 coefficients for Green2. Legal values: [0, 65535].		
14080 R0x00003700	31:0	0x0000	p_g1_p3q0 (R/W)
	P3 coefficients for Green1. Legal values: [0, 65535].		
14082 R0x00003702	31:0	0x0000	p_g1_p3q1 (R/W)
	P3 coefficients for Green1. Legal values: [0, 65535].		
14084 R0x00003704	31:0	0x0000	p_g1_p3q2 (R/W)
	P3 coefficients for Green1. Legal values: [0, 65535].		
14086 R0x00003706	31:0	0x0000	p_g1_p3q3 (R/W)
	P3 coefficients for Green1. Legal values: [0, 65535].		
14088 R0x00003708	31:0	0x0000	p_g1_p3q4 (R/W)
	P3 coefficients for Green1. Legal values: [0, 65535].		
14090 R0x0000370A	31:0	0x0000	p_r_p3q0 (R/W)
	P3 coefficients for Red. Legal values: [0, 65535].		
14092 R0x0000370C	31:0	0x0000	p_r_p3q1 (R/W)
	P3 coefficients for Red. Legal values: [0, 65535].		
14094 R0x0000370E	31:0	0x0000	p_r_p3q2 (R/W)
	P3 coefficients for Red. Legal values: [0, 65535].		
14096 R0x00003710	31:0	0x0000	p_r_p3q3 (R/W)
	P3 coefficients for Red. Legal values: [0, 65535].		
14098 R0x00003712	31:0	0x0000	p_r_p3q4 (R/W)
	P3 coefficients for Red. Legal values: [0, 65535].		
14100 R0x00003714	31:0	0x0000	p_b_p3q0 (R/W)
	P3 coefficients for Blue. Legal values: [0, 65535].		
14102 R0x00003716	31:0	0x0000	p_b_p3q1 (R/W)
	P3 coefficients for Blue. Legal values: [0, 65535].		
14104 R0x00003718	31:0	0x0000	p_b_p3q2 (R/W)
	P3 coefficients for Blue. Legal values: [0, 65535].		
14106 R0x0000371A	31:0	0x0000	p_b_p3q3 (R/W)
	P3 coefficients for Blue. Legal values: [0, 65535].		
14108 R0x0000371C	31:0	0x0000	p_b_p3q4 (R/W)
	P3 coefficients for Blue. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
14110 R0x0000371E	31:0	0x0000	p_g2_p3q0 (R/W)
	P3 coefficients for Green2. Legal values: [0, 65535].		
14112 R0x00003720	31:0	0x0000	p_g2_p3q1 (R/W)
	P3 coefficients for Green2. Legal values: [0, 65535].		
14114 R0x00003722	31:0	0x0000	p_g2_p3q2 (R/W)
	P3 coefficients for Green2. Legal values: [0, 65535].		
14116 R0x00003724	31:0	0x0000	p_g2_p3q3 (R/W)
	P3 coefficients for Green2. Legal values: [0, 65535].		
14118 R0x00003726	31:0	0x0000	p_g2_p3q4 (R/W)
	P3 coefficients for Green2. Legal values: [0, 65535].		
14144 R0x00003740	31:0	0x0000	p_g1_p4q0 (R/W)
	P4 coefficients for Green1. Legal values: [0, 65535].		
14146 R0x00003742	31:0	0x0000	p_g1_p4q1 (R/W)
	P4 coefficients for Green1. Legal values: [0, 65535].		
14148 R0x00003744	31:0	0x0000	p_g1_p4q2 (R/W)
	P4 coefficients for Green1. Legal values: [0, 65535].		
14150 R0x00003746	31:0	0x0000	p_g1_p4q3 (R/W)
	P4 coefficients for Green1. Legal values: [0, 65535].		
14152 R0x00003748	31:0	0x0000	p_g1_p4q4 (R/W)
	P4 coefficients for Green1. Legal values: [0, 65535].		
14154 R0x0000374A	31:0	0x0000	p_r_p4q0 (R/W)
	P4 coefficients for Red. Legal values: [0, 65535].		
14156 R0x0000374C	31:0	0x0000	p_r_p4q1 (R/W)
	P4 coefficients for Red. Legal values: [0, 65535].		
14158 R0x0000374E	31:0	0x0000	p_r_p4q2 (R/W)
	P4 coefficients for Red. Legal values: [0, 65535].		
14160 R0x00003750	31:0	0x0000	p_r_p4q3 (R/W)
	P4 coefficients for Red. Legal values: [0, 65535].		
14162 R0x00003752	31:0	0x0000	p_r_p4q4 (R/W)
	P4 coefficients for Red. Legal values: [0, 65535].		
14164 R0x00003754	31:0	0x0000	p_b_p4q0 (R/W)
	P4 coefficients for Blue. Legal values: [0, 65535].		

Table 45: 2: SOC2 Registers

Register Dec (Hex)	Bits	Default	Name
14166 R0x00003756	31:0	0x0000	p_b_p4q1 (R/W)
	P4 coefficients for Blue. Legal values: [0, 65535].		
14168 R0x00003758	31:0	0x0000	p_b_p4q2 (R/W)
	P4 coefficients for Blue. Legal values: [0, 65535].		
14170 R0x0000375A	31:0	0x0000	p_b_p4q3 (R/W)
	P4 coefficients for Blue. Legal values: [0, 65535].		
14172 R0x0000375C	31:0	0x0000	p_b_p4q4 (R/W)
	P4 coefficients for Blue. Legal values: [0, 65535].		
14174 R0x0000375E	31:0	0x0000	p_g2_p4q0 (R/W)
	P4 coefficients for Green2. Legal values: [0, 65535].		
14176 R0x00003760	31:0	0x0000	p_g2_p4q1 (R/W)
	P4 coefficients for Green2. Legal values: [0, 65535].		
14178 R0x00003762	31:0	0x0000	p_g2_p4q2 (R/W)
	P4 coefficients for Green2. Legal values: [0, 65535].		
14180 R0x00003764	31:0	0x0000	p_g2_p4q3 (R/W)
	P4 coefficients for Green2. Legal values: [0, 65535].		
14182 R0x00003766	31:0	0x0000	p_g2_p4q4 (R/W)
	P4 coefficients for Green2. Legal values: [0, 65535].		

Table 46: SYSCTL Registers

Register Dec (Hex)	Bits	Default	Name
0 R0x00000000	31:0	0x2880	chip_id (RO)
	Read-only.		
6 R0x00000006	31:0	0x0000	i2c_control (R/W)
	31:16	X	Reserved
	15	0x0000	i2c_disabled If it's 1, i2c is disabled.
	14	0x0000	i2c_noack If it's 1, enable i2c noack.
	13:12	X	Reserved
	11	0x0000	i2c_force_dreg_enabled
	10	0x0000	i2c_force_dreg_locked
	9	0x0000	i2c_mb_dma_locked
	8:0	X	Reserved
16 R0x00000010	31:0	0x0208	pll_dividers (R/W)
	31:14	X	Reserved
	13:8	0x0002	n PLL N divider VCO freq controller. VCO_freq = Clkin_freq * 2 * M / (N + 1) Clkin_freq / (N + 1) must be >= 2 MHz. 384 MHz <= VCO_freq <= 768 MHz. Legal values: [0, 63].
	7:0	0x0008	m PLL M divider VCO freq controller. VCO_freq = Clkin_freq * 2 * M / (N + 1) 384 MHz <= VCO_freq <= 768 MHz. Legal values: [16, 192].
	This register specifies the divider for the PLL.		
18 R0x00000012	31:0	0x00B0	pll_p_dividers (R/W)
	31:14	X	Reserved
	13:12	0x0000	word_clock_divider PLL word clock divider
	11:8	0x0000	p3 PLL p3 divider
	7:4	0x000B	p2 PLL p2 divider
	3:0	0x0000	p1 PLL p1 divider
	This register specifies the divider for the PLL.		

Table 46: SYSCTL Registers (continued)

Register Dec (Hex)	Bits	Default	Name
20 R0x00000014	31:0	0x2425	pll_control (R/W)
	31:16	X	Reserved
	15	RO	pll_lock PLL lock status Read-only.
	14	X	Reserved
	13	0x0001	clockin_hyst_en Enable hysteresis on clock input pin
	12	0x0000	sel_lock_det Select pll bypass or lock_detect: 0: pll bypass; 1: lock detect enable
	11	0x0000	test_reset Reset test counter
	10	0x0001	test_bypass Feeds tclk to P &M dividers instead of VCO output
	9	0x0000	clk_mux Selection of system clock source
	8	0x0000	reset_cntr Asynchronous/synchronous reset for N,M and P and divide by 8/10 counters
	7:4	0x0002	pfd Pfd tuning control
	3:2	0x0001	lock_mode PLL lock detector mode selection
	1	0x0000	pll_enable PLL enable
	0	0x0001	pll_bypass PLL bypass
	This register initializes and controls the PLL.		
22 R0x00000016	31:0	0x0000	clocks_control (R/W)
	31:16	X	Reserved
	15	0x0000	en_dbl_buf TESTP module of rx_ss will use double buffer version of sel_vsrc and sel_testp signals when en_dbl_buf is '1'.
	14	X	Reserved
	13	0x0000	aopd_sel_testp aopd_sel_testp
	12:11	0x0000	sel_vsrc SENSIF clock source selection: 00: RXPP1 (on chip sensor) 10: RX1MC2 clk_rx1 (mipi rx) Others: reserved
	10	0x0000	clk_jpeg_en Enable jpeg clock
	9	0x0000	clk_clkin_en Enable external clock pad
	8	0x0000	force_clk_icb_en Enable clock clk_icb

Table 46: SYSCTL Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	7	X	Reserved
	6	0x0000	clk_mp_cont_en Enable clock clk_mp_cont
	5	0x0000	nosleep Set no sleep to MCU
	4	0x0000	clk_i2c_master_en enable clock clk_i2c_master
	3	0x0000	clk_otpm_en enable clock clk_otpm
	2	0x0000	clk_pads_en Enable clock clk_pads
	1	0x0000	clk_sensif_en Enable clock clk_sensif
	0	0x0000	clk_pix_soc_en Enable clock clk_pix_soc
24 R0x00000018	31:0	0x6009	standby_control_and_status (R/W)
	31:16	X	Reserved
	15	RO	standby_in Hard standby status Read-only.
	14	0x0001	standby_done In Standby state
	13	RO	core_ready Core is ready, VDD on, reset cleared Read-only.
	12:7	X	Reserved
	6	0x0000	otpm_access_indicator 0: FW will not access OTPM in power-on/reset routine. 1: FW will access OTPM access in power-on/reset routine. (this bit is controlled by HOST only)
	5	0x0000	scas_reserved5 Unused bit.
	4	0x0000	sb_f_sync sync standby at the end of frame
	3	0x0001	en_irq Enable Interrupt request
	2	0x0000	powerup_stop_mcu Stop mcu when powerup
	1	0x0000	do_not_init_mcu_var Don't initialize MCU variable
	0	0x0001	standby_i2c Soft standby bit

Table 46: SYSCTL Registers (continued)

Register Dec (Hex)	Bits	Default	Name
26 R0x0000001A	31:0	0x0018	reset_and_misc_control (R/W)
	31:11	X	Reserved
	10	0x0000	reset_jpeg_soft Jpeg soft reset
	9	0x0000	parallel_enable 1=parallel output enable (this bit works with bit 8)
	8	0x0000	oe_gp_enable Enables GPIO function as output enable: 0 = disable, 1 = enable
	7:6	X	Reserved
	5	0x0000	clkin_ip_pd_en CLK pad input pd
	4	0x0001	ip_pd_en GPIO pad input pd
	3	0x0001	vgp_ip_pd_en VGPIO pad input pd
	2	0x0000	mipi_tx_en mipi_tx_en 0=mipi transmitter disable 1=mipi transmitter enable
	1	0x0000	mipi_tx_reset Reset MIPI TX
	0	0x0000	reset_soc_i2c Chip Soft reset bit
28 R0x0000001C	31:0	0x0000	mcu_boot_mode (R/W)
	31:16	X	Reserved
	15:8	RO	Reserved
	7:6	X	Reserved
	5	0x0000	mcu_bist_bypass_ram Bypass RAM built in self test.
	4:3	X	Reserved
	2	0x0000	mcu_bist_bypass_rom Bypass generation of the ROM CRC value during built in self test.
	1	0x0000	mcu_bist_enable If 1 MCU performs built in self test at reset. If 0 MCU boots normally at reset.
	0	0x0000	mcu_soft_rst_i2c Reset MCU

Table 46: SYSCTL Registers (continued)

Register Dec (Hex)	Bits	Default	Name
30 R0x0000001E	31:0	0x0400	pad_slew_pad_config (R/W)
	31:16	X	Reserved
	15	0x0000	phy_test Enable the phy word clock for observation.
	14:11	X	Reserved
	10:8	0x0004	slew_pxlclk PIXCLK pad slew rate
	7	X	Reserved
	6:4	0x0000	slew_gpio GPIO pads slew rate
	3	X	Reserved
	2:0	0x0000	slew_io DOUT* pads slew rate
32 R0x00000020	31:0	0x0001	vdd_dis_soft (R/W)
	Disable core Vdd power supply		
34 R0x00000022	31:0	0x0438	vdd_dis_counter (R/W)
	Control the delay of vdd_dis counter, by default, it is 20us with CLKIN of 27Mhz.		
38 R0x00000026	31:0	0x0007	mcu_rom_initialize_sensor (R/W)
	31:3	X	Reserved
	2	0x0001	reset_core_pads Resets core pads. This is needed AFTER fsio_poweroff is cleared!
	1	0x0001	fsio_poweroff fsio_std cell will enter fail- safe mode when asserted 0 = off; 1 = on
	0	0x0001	initialize_sensor This bit forces reset to the whole imaginf core (sensor, cpipe, ...). MCU clear this bit to '0' after loading sensor registers default and fuse rom.
40 R0x00000028	31:0	0x0001	en_vdd_dis_soft (R/W)
	Enable/ disable register 0x0020[0] (vdd_dis_soft) bit function. 1 = enable, 0 = disable		
42 R0x0000002A	31:0	0x77DA	pll_p4_p5_p6_dividers (R/W)
	31:15	X	Reserved
	14	0x0001	p6_en Enable p6_clk
	13	0x0001	p5_en Enable p5_clk
	12	0x0001	p4_en Enable p4_clk
	11:8	0x0007	p6 PLL clock divider ratio for p6_clk (clk_mp)
	7:4	0x000D	p5 PLL clock divider ratio for p5_clk (clk_pix_soc)
	3:0	0x000A	p4 PLL clock divider ratio for p4_clk (clk_pixel, ect)

Table 46: SYSCTL Registers (continued)

Register Dec (Hex)	Bits	Default	Name
44 R0x0000002C	31:0	0x1007	pll_p7_divider (R/W)
	31:13	X	Reserved
	12	0x0001	p7_en Enable p7_clk
	11:4	X	Reserved
	3:0	0x0007	p7
46 R0x0000002E	31:0	0x0000	sensor_clock_divider (R/W)
	31:11	X	Reserved
	10	0x0000	clk_sensor_pll_bypass Clk_sensor1 bypass the pll but all other clocks use pll in functional mode
	9	X	Reserved
	8	0x0000	clk_sensor1_en Enable clk_sensor1
	7:4	X	Reserved
	3:0	0x0000	clk_sensor1_divider Clk_sensor1 logic divider (/e) , even numbers from 1~32. e is the divider number divided by 2.
48 R0x00000030	31:0	0x0003	clk_otpm_clock_divider_selection (R/W)
	31:2	X	Reserved
	1:0	0x0003	clk_otpm_divider_sel 00: div 1 01: div 2 10: div 4 11: div 8 Note: clk_otpm maximum speed is 12mhz
50 R0x00000032	31:0	0x0000	s_clk_pad_slew_rate (R/W)
	External sensor (I2C) master clock pad slew rate 0x7 is maximum drive, 0 is minimum drive. See fsio pad documents for precise values.		
52 R0x00000034	31:0	0x0000	reg_0034 (R/W)
	This register has no connections in K28A		
54 R0x00000036	31:0	0x0000	reg_0036 (R/W)
64 R0x00000040	31:0	0x7A78	user_defined_i2c_device_address_id (R/W)
	31:16	X	Reserved
	15:9	0x003D	host_i2c_dev_id1 I2C device address when SADDR = 1.
	8	X	Reserved
	7:1	0x003C	host_i2c_dev_id0 I2C device address when SADDR = 0.
	0	X	Reserved
66 R0x00000042	31:0	0x0000	burnin_passcode (R/W)
	This is only operative during burnin test mode. If the contents of this register matches the value lateched from the VGPIO lines at the end of reset, then the I2C output (SDATA) will be enabled.		
68 R0x00000044	31:0	0x0000	gen_purp_0 (R/W)
	General purpose AOPD memory		

Table 46: SYSCTL Registers (continued)

Register Dec (Hex)	Bits	Default	Name
70 R0x00000046	31:0	0x0000	gen_purp_1 (R/W)
	General purpose AOPD memory		
72 R0x00000048	31:0	0x0000	gen_purp_2 (R/W)
	General purpose AOPD memory		
74 R0x0000004A	31:0	0x0000	gen_purp_3 (R/W)
	General purpose AOPD memory		
80 R0x00000050	31:0	0x0000	release_version (RO)
	31:16	X	Reserved
	15:8	RO	proto_release_version Contains proto release version. Read-only.
	7:0	RO	rtl_release_version Contains full chip release version. Read-only.

Table 47: RX_SS Registers

Register Dec(Hex)	Bits	Default	Name
256 R0x00000100	31:0	0x2000	test_pattern_control (R/W)
	31:16	X	Reserved
	15	0x0000	sel_testp Selects testp module as source of FV, LV and pixel. 0: input image either from parallel or serial interface. 1: input image from internal pattern generator.
	14	0x0000	xfenlen_en Selects external Frame and Line Valid. 0: rx_ss uses Frame and Line Valid from testp. 1: rx_ss uses external Frame and Line valid.
	13	RO	walking1_pass Indicates the status of the walking 1 test pattern check: 0: test failed 1: test passed Read-only.
	12:11	0x0000	walking1_pattern Indicates if the incoming data is a walking1 pattern: 0: no walking 1 test pattern 1: 8-bit pattern 2: 10-bit pattern 3: 12-bit pattern Legal values: [0, 3].
	10:8	0x0000	y_skip Indicates the row skip values: 0: no skip 1: 2x skip 2: 4x skip 3: 8x skip 4 - 7: 16x skip Legal values: [0, 7].
	7:5	0x0000	x_skip Indicates the column skip values: 0: no skip 1: 2x skip 2: 4x skip 3: 8x skip 4 - 7: 16x skip Legal values: [0, 7].
	4:3	0x0000	testp_first_color Indicates the color of the first pixel 0: Green1 1: Red 2: Blue 3: Green2 Legal values: [0, 3].

Table 47: RX_SS Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	2:0	0x0000	tst_mode Available test modes. 1: flat field values for RGB from R test pattern, G1 test pattern and B test pattern 2: vertical ramp 3: regular color bar 4: vertical strips, intensities are from R test pattern, G1 test pattern and B test pattern 5: pseudo random pattern (linear feedback shift register) 6: horizontal monochrome bars 7: programmable random pattern (linear feedback shift register) Legal values: [1, 7].
258 R0x00000102	31:0	0x0000	test_pxl_red (R/W)
	Value for the red component on the flat field test pattern. Legal values: [0, 1023].		
260 R0x00000104	31:0	0x0000	test_pxl_green1 (R/W)
	Value for the green1 component on the flat field test pattern. Legal values: [0, 1023].		
262 R0x00000106	31:0	0x0000	test_pxl_green2 (R/W)
	Value for the green2 component on the flat field test pattern. Legal values: [0, 1023].		
264 R0x00000108	31:0	0x0000	test_pxl_blue (R/W)
	Value for the blue component on the flat field test pattern. Legal values: [0, 1023].		
266 R0x0000010A	31:0	0x0000	row_start (R/W)
	Starting address of row to be used in generation of frame vaild (TESTP). Legal values: [0, 1951].		
268 R0x0000010C	31:0	0x0000	col_start (R/W)
	Starting address of column to be used in generation of frame vaild (TESTP). Legal values: [0, 2599].		
270 R0x0000010E	31:0	0x0607	row_end (R/W)
	End address of row to be used in generation of frame vaild (TESTP). Legal values: [0, 1951].		
272 R0x00000110	31:0	0x0807	col_end (R/W)
	End address of column to be used in generation of frame vaild (TESTP). Legal values: [0, 2599].		
274 R0x00000112	31:0	0x0014	rx_fifo_control (R/W)
	31:16	X	Reserved
	15	RO	bist_pass 1: RX FIFO BIST has passed Read-only. Legal values: [0,1].
	14	RO	bist_done 1: RX FIFO BIST is done Read-only. Legal values: [0,1].
	13	0x0000	bist_start 1: Start RX FIFO BIST Legal values: [0,1].
	12	0x0000	clr_rxfifo_ovf_unf Clears RXFIFO overflow and underflow Legal values: [0,1].

Table 47: RX_SS Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	11	RO	fifo_underflow Indicates RXFIFO is empty when asserted Read-only.
	10	RO	fifo_overflow Indicates RXFIFO is full when asserted Read-only.
	9:0	0x0014	rx_fifo_watermark Minimum number of data in FIFO before starting read operation Legal values: [0, 1023].
276 R0x00000114	31:0	0x0064	hblank_llead (R/W)
	Number of pixel clocks from leading edge of Frame Valid to leading edge of Line Valid (first row). Number of pixel clocks from column count zero to leading edge of Line Valid. Legal values: [0, 8191].		
278 R0x00000116	31:0	0x000A	hblank_ltrail (R/W)
	Number of pixel clocks from trailing edge of Line Valid use to clear column counter. Number of extra pixels at the end of each line. Legal values: [0, 8191].		
280 R0x00000118	31:0	0x0000	lower_seed_value (R/W)
	When tst_mode = 7, programmable random pattern, this sets the lower bits of the seed value. Legal values: [0, 65535].		
282 R0x0000011A	31:0	0x0000	upper_seed_value (R/W)
	When tst_mode = 7, programmable random pattern, this sets the upper bits of the seed value. Legal values: [0, 15].		
296 R0x00000128	31:0	0x0004	mipi_receiver_control (R/W)
	31:16	X	Reserved
	15	0x0000	sec_sens_en 0: Bring-out rx_word_clk to gpio[2]. 1: Bring-out clock_lp_datan to gpio[2].
	14	0x0000	clr_csi_err_flag Clears MIPI CSI error flags Legal values: [0,1].
	13:12	0x0000	hs_threshold_test_mode Indicates the following: 00: disable parametric testing 11: connect HS receiver output to nand tree 10: connect LP receiver output to nand tree 11: connect ULP receiver output to nand tree Legal values: [0, 3].
	11	0x0000	bypass_csi Indicates the following: 0: disable in test_mode 1: laneN_data[7:0] is routed in pixel_dataN[11:4], pixelValidN, line_valid and frame_valid will not be recovered from the data stream and their values are unknown.
	10	X	Reserved
	9	0x0000	force_reset_sync If mipi_hs_test_en is asserted, then assertion of this bit will force the d1(2)_reset_sync_n outputs: 0: disable in test mode 1: force the laneN_sync_reset outputs when mipi_hs_teset_en is asserted.

Table 47: RX_SS Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	8	0x0000	mipi_hs_test_en When asserted a continuous high speed MIPI data stream will be transmitted without the need to cycle through the LP states as directed by the MIPI protocol first: 0: test mode disable 1: continuous high speed MIPI data stream will be received without the need to have first cycle through the LP states as directed by MIPI protocol.
	7	0x0000	sh_pkt_line_valid_en Indicates the following: 0: line valid is toggled at the start/end of long data packets 1: line valid is toggled by line synchronization short packets only
	6:5	0x0000	lp_filter_cont_cfg Configuration of the MIPI LP mode receiver filter and hysterisis Legal values: [0, 3].
	4	0x0000	digital_timer_sel Indicates the following: 0: the analogue timer in the PHY is used to measure the settle periods. 1: the digital timer in the PHY is used to measure the settle periods.
	3	0x0000	exact_match_sel Receiver synchronization to SoT code (0xb8): 0: synchronize to SoT (0xb8) with a single bit error. 1: synchronize exactly to SoT (0xb8)
	2	0x0001	mipiccp_unpack_en Enable unpacking of received byte data to original pixel format by IP: 0: unpacking of data disable 1: unpacking of data enable
	1	X	Reserved
	0	0x0000	mipiccp_en Enable the Serial Interface. 0: disable 1: enable
300 R0x0000012C	31:0	0x0800	mipi_and_ccp2_pixel_data_out_status_0 (RO)
	31:13	X	Reserved
	12	RO	mipi_pkt_hdr_update Asserted for a single cycle when a new short/long packet header has been detected. The chan_num, mipi_data_type[5:0] and mipiccp_wc[15:0] fields are updated accordingly. Read-only.
	11	RO	mipiccp_stdby Asserted once the transition to ulps (MIPI)/EOF sync code (CCP) is detected. Remains asserted until a LP transition on the MIPI link/a CCP sync code is detected. Read-only.
	10	RO	mipi_embedded_data Asserted if current line contains embedded data (applicable for MIPI interface only - for CCP2 (SMIA) first line of a frame will always contain embedded data and frame format description field within that line will define the number and location of embedded data lines within the frame) Read-only.

Table 47: RX_SS Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	9:4	RO	mipi_data_type Last data type code transmitted as part of the short packet/long packet header (after error correction). Read-only.
	3:1	RO	chan_num Virtual channel number for the current packet of data (bits 1:0 only applicable for MIPI). Read-only.
	0	RO	pixel_data_unpacked Asserted if pixel_data1/2 has been unpacked. If not asserted, then pixel_data1/2 = lane1(/2)_data (with packet header/footer or sync codes stripped off). Read-only.
302 R0x0000012E	31:0	0x0000	mipi_and_ccp2_pixel_data_out_status_1_ (RO)
	31:16	X	Reserved
	15:0	RO	mipiccp_wc Contents of MIPI short packet data field or long packet word count field (after error correction), or the number of bytes detected between CCP start and end sync codes. Read-only.
304 R0x00000130	31:0	0x0000	mipi_error_reporting_ (RO)
	31:11	X	Reserved
	10	RO	mipi_c_esc_ent_err Asserted if an invalid escape mode entry sequence is detected on the clock lane (currently only ULPM supported). Read-only.
	9	RO	mipi_d2_esc_ent_err Asserted if an invalid escape mode sequence is detected on the secondary data lane (currently only ULPM supported). Read-only.
	8	RO	mipi_d1_esc_ent_err Asserted if an invalid escape mode sequence is detected on the primary data lane (currently only ULPM supported). Read-only.
	7	RO	mipi_c_lp_seq_err Asserted if the sequence of LP signal on the clock lane violates the MIPI spec. Read-only.
	6	RO	mipi_d2_lp_seq_err Asserted if the sequence of LP signal on the secondary data lane violates the MIPI spec. Read-only.
	5	RO	mipi_d1_lp_seq_err Asserted if the sequence of LP signal on the primary data lane violates the MIPI spec. Read-only.
	4	RO	mipi_d2_eot_err Asserted if the EoT sequence is not detected on the secondary data lane. Read-only.

Table 47: RX_SS Registers (continued)

Register Dec (Hex)	Bits	Default	Name
	3	RO	mipi_d1_eot_err Asserted if the EoT sequence is not detected on the primary data lane (indicating loss of synchronization and hence almost certainly preceded by an assertion of mipi_checksum_err). Read-only.
	2	RO	mipi_sync_bit_err Asserted when one bit of the MIPI SoT sequence on either data lane is corrupted but it is believed that synchronization has been achieved regardless. Read-only.
	1	RO	mipi_d2_sync_err Asserted when false synchronization of the secondary data lane occurs. Read-only.
	0	RO	mipi_d1_sync_err Asserted when false synchronization of the primary data lane occurs (all data packet will be lost). Read-only.
306 R0x00000132	31:0	0x0000	packet_level_errors_0_ (RO)
	31:11	X	Reserved
	10	RO	mipi_checksum_err Asserted if the checksum calculated by the Rx does not match that in the packet footer. Read-only.
	9	RO	mipi_ecc_warning Asserted if the ecc has detected and corrected a single bit error. Read-only.
	8	RO	mipi_ecc_err Asserted when the ecc detects an unrecoverable error. Read-only.
	7:0	RO	mipi_d1_sot_code The first byte of synchronized data on the primary data lane from the MIPI hs data stream. i.e., the SoT code (expected value = 0xb8). Read-only.
308 R0x00000134	31:0	0x0000	packet_level_errors_1_and_protocol_layer_errors_ (RO)
	31:10	X	Reserved
	9	RO	mipi_line_sync_err Asserted if a line end short packet is not paired with a line start channel. Read-only.
	8	RO	mipi_frame_sync_err Asserted if a frame end short packet is not paired with a frame start channel. Read-only.
	7:0	RO	mipi_d2_sot_code The first byte of synchronized data on the secondary data lane from the MIPI hs data stream. i.e., the SoT code (expected value = 0xb8). Read-only.
310 R0x00000136	31:0	0x0000	data_visibility_0_ (RO)
	31:6	X	Reserved
	5:0	RO	mipi_ecc_rcvd The last ecc field as received as part of a MIPI short/long packet header. Read-only.

Table 47: RX_SS Registers (continued)

Register Dec (Hex)	Bits	Default	Name
312 R0x00000138	31:0	0x0000	data_visibility_1_ (RO)
	31:16	X	Reserved
	15:0	RO	checksum_rcvd The last checksum received in a long packet footer (guaranteed valid from when mipi_checksum_err/ ccp_checksum_err is asserted to flag an error). Read-only.
314 R0x0000013A	31:0	0x0000	data_visibility_2_ (RO)
	31:16	X	Reserved
	15:0	RO	checksum_expected The expected checksum, generated within the protocol layer, based on the last packet of data received (guaranteed valid from when mipi_checksum_err/ ccp_checksum_err is asserted to flag an error). Read-only.
316 R0x0000013C	31:0	0x181F	phy_configuration_control (R/W)
	31:14	X	Reserved
	13:11	0x0003	tclk_settle_cont_cfg T clk_settle timing parameter control.
	10:3	0x0003	ths_settle_cont_cfg Configuration of the analogue timer measuring the Ths- settle.T hs_settle timing parameter control.
	2:0	0x0007	bias_cont_cfg Configuration of bias currents for the HS and LP receivers. LP and HS receivers bias current control.
318 R0x0000013E	31:0	0x00FF	testp_colorbar_width (R/W)
	Indicates the test epattern's colorbar width - 1. Legal values: [0,2600].		
320 R0x00000140	31:0	0x0010	vblank_llead (R/W)
	Number of read clocks to delay Frame Valid output from Frame Valid input calculated using below equation: vblank_llead = |6-rx_fifo_watermark| + 2 Note : Frame Valid leading edge to first Line Valid leading edge is fixed to 6 pixel clocks. Legal values: [0, 8191].		
322 R0x00000142	31:0	0x0006	vblank_ltrail (R/W)
	Number of read clocks (clk_pix_soc) from trailing edge of last Line Valid to trailing edge of Frame Valid. Legal values: [0, 8191].		
336 R0x00000150	31:0	0x0000	crc_lane_1_data_rx_ss (RO)
	31:16	X	Reserved
	15:0	RO	rxss_l1_crc_data Factory Test - Lane 1 CRC Data Read-only. Legal values: [0, 65535].
338 R0x00000152	31:0	0x0000	crc_lane_2_data_rx_ss (RO)
	31:16	X	Reserved
	15:0	RO	rxss_l2_crc_data Factory Test - Lane 2 CRC Data Read-only. Legal values: [0, 65535].
344 R0x00000158	31:0	0x0000	testp_vblank (R/W)
	Extra time added to the end of each frame minus 5. Number of extra rows at the end of frame. Legal values: [0, 65535].		

Table 47: RX_SS Registers (continued)

Register Dec (Hex)	Bits	Default	Name
346 R0x0000015A	31:0	0x047F	eofv_del (R/W)
	Delay count prior to assertion of fv based reset + 64. Legal values: [0, 2047].		

Table 48: XDMA Registers

Register Dec (Hex)	Bits	Default	Name
2434 R0x00000982	31:0	0x0000	access_ctl_stat (R/W)
	31:8	X	Reserved
	7:6	0x0000	phy_region 00: Physical access to Patch RAM 01: UNDEFINED 10: Physical access to SFR address space 11: Physical access to Overlay RAM When physical_access_state=11, this field determines which memory region will be accessed. When physical_access_state=10, the Patch RAM is implicitly selected.
	5	X	Reserved
	4	RO	byte_access_state 1: Byte Access State 0: Word Access State (2 bytes) Read-only.
	3:2	RO	physical_access_state 11: Physical Access State 10: Logical Access State 0x: Indeterminate (DMA address is invalid). The DMA address is made valid either by writing first the tabptr SFR and then the logical_address_access register, or by writing the physical_address_access register. The DMA address is made invalid by reset, or by writing the tabptr SFR whilst physical_access_state=x0 (logical access state). Read-only.
	1	RO	upper_32k_access_state Physical address[15] for current access. When physical_access_state=10, this bit represents the physical address[15] that results from translating the current value of logical_address_access (R0x098E). When physical_access_state=11, this bit is a read-only copy of en_upper_32k_phy_access. Read-only.
	0	0x0000	en_upper_32k_phy_access This bit provides physical address[15] for physical address accesses. physical address[14:0] are provided by R0x098A
	Controls the access and conveys access status		
2442 R0x0000098A	31:0	0x0000	physical_address_access (R/W)
	31:16	X	Reserved
	15	0x0000	physical_byte_access 1: Byte Access 0: Word Access (2 bytes)
	14:0	0x0000	physical_address physical_address[14:0] for current access. physical_address[15] is set by R0x0982[0]. Legal values: [0, 32767].
	Address of physical access; Used for Patch RAM uploads. A write to this address establishes the Physical Access State (See R0x0982[2]). When the Logial Access State is established, a read from this register and from R0x0982[1] will return the physical address translated from the current logical_access_offset, logical_access_drv_num and logical_access_offset.		

Table 48: XDMA Registers (continued)

Register Dec (Hex)	Bits	Default	Name
2446 R0x0000098E	31:0	0x0000	logical_address_access (R/W)
	31:16	X	Reserved
	15	0x0000	logical_byte_access 1: Byte Access 0: Word Access (2 bytes)
	14:10	0x0000	logical_access_drv_num Address of logical access driver number - logical_address[14:10]. Base address of this driver's variables can be obtained by adding 2*logical_access_drv_num to the value of the tabptr SFR. Physical address of re-directed location can be obtained by adding this offset to the SFR 0x50 return value. Legal values: [0, 31].
	9:0	0x0000	logical_access_offset Address of logical access offset - logical_address[9:0]. Physical address can be obtained by adding this offset to the base address of the selected driver's variables (the driver is selected by logical_access_drv_num). Legal values: [0, 1023].
	Address of logical access; Used for camera control (i.e. register/variable updates) by user. A write to this address establishes Logical Access State (See R0x0982[2]).		
2448 R0x00000990	31:0	0x0000	mcu_variable_data0 (R/W)
	DMA word 0 Legal values: [0, 65535].		
2450 R0x00000992	31:0	0x0000	mcu_variable_data1 (R/W)
	DMA word 1 Legal values: [0, 65535].		
2452 R0x00000994	31:0	0x0000	mcu_variable_data2 (R/W)
	DMA word 2 Legal values: [0, 65535].		
2454 R0x00000996	31:0	0x0000	mcu_variable_data3 (R/W)
	DMA word 3 Legal values: [0, 65535].		
2456 R0x00000998	31:0	0x0000	mcu_variable_data4 (R/W)
	DMA word 4 Legal values: [0, 65535].		
2458 R0x0000099A	31:0	0x0000	mcu_variable_data5 (R/W)
	DMA word 5 Legal values: [0, 65535].		
2460 R0x0000099C	31:0	0x0000	mcu_variable_data6 (R/W)
	DMA word 6 Legal values: [0, 65535].		
2462 R0x0000099E	31:0	0x0000	mcu_variable_data7 (R/W)
	DMA word 7 Legal values: [0, 65535].		
2560 R0x00000A00	31:0	0x0000	jtag_tms (R/W)
	Data for TMS pin		
2562 R0x00000A02	31:0	0x0000	jtag_tdi (R/W)
	Data for TDI pin		

Table 48: XDMA Registers (continued)

Register Dec (Hex)	Bits	Default	Name
2564 R0x00000A04	31:0	0x0000	jtag_ctl (R/W)
	31:7	X	Reserved
	6	0x0000	jtag_enable Enables register control of JTAG to uP.
	5	0x0000	jtag_trstn Value of TRST_ when jtag interface is enabled
	4	0x0000	jtag_go Set to start JTAG interface, cleared by hardware when idle
	3:0	0x0000	jtag_count Number of bits to send, data is sent lsb first.
2566 R0x00000A06	31:0	0x0000	jtag_tdo (RO)
	Data from TDO pin, in the upper -jtag_count- bits Read-only.		

Table 49: TX_SS

Register Dec(Hex)	Bits	Default	Name
13312 R0x00003400	31:0	0x782E	mipi_control (R/W)
	31:16	X	Reserved
	15:10	0x001E	data_type Data Format: 0x1E: YUV422 8-bit 0x22: RGB565 0x21: RGB555 0x20: RGB444 0x2A: RAW8 0x2B: RAW10 0x30: JPEG 0x32: Status Legal values: [1e, 2b].
	9	0x0000	mipi_en 1: Enable MIPI Transmit 0: Disable MIPI Transmit
	8:6	0x0000	tx_chan_num Virtual channel number to be inserted in MIPI packet header (b1:0 only). Legal values: [0,7].
	5	0x0001	mipiccp_pack_en Enable MIPI packing logic Legal values: [0,1].
	4	0x0000	standby_eof Wait until EOF to react to standby; 0: Respond to standby_en at end of current packet/line. 1: Respond to standby_en at end of frame. Legal values: [0,1].
	3	RO	reg_frame_sync Frame boundary sync bit; 1: safe to update frame-synchronized inputs. Read-only. Legal values: [0,1].

Table 49: TX_SS (continued)

Register Dec(Hex)	Bits	Default	Name
	2	0x0001	cont_mipi_clk Continuous MIPI clock; 0: clock active only when data is transmitted 1: transmit a continuous MIPI output clock Legal values: [0,1].
	1	0x0001	standby_en MIPI Standby; Asserted when the chip is to go into standby. The interface completes the transmission of the current packet/frame (as determined by standby_eof), and (for MIPI) sends an end of frame packet and goes into ultra low-power mode. Clock must continue to be supplied until mipi_stdby (0x3402[0]) is asserted. Legal values: [0,1].
	0	0x0000	restart_en MIPI Restart enable; When asserted the interface completes the transmission of the current packet/frame and sends an end of frame packet. Legal values: [0,1].
	Configure MIPI Output Interface		
13314 R0x00003402	31:0	0x0011	mipi_status (R/W)
	31:13	X	Reserved
	12	0x0000	mipi_preamble_err_rst Reset MIPI preamble error; 1: The mipi_preamble_err flag is cleared Legal values: [0,1].
	11	0x0000	mipi_line_byte_err_rst Reset MIPI line byte error; 1: The mipi_line_byte_err flag is cleared Legal values: [0,1].
	10	RO	mipi_preamble_err MIPI Preamble error; Asserted if valid data is presented to the interface before transmission of the protocol preamble signalling has completed. Signal remains asserted until cleared by mipi_preamble_err_rst or reset_n Read-only. Legal values: [0,1].
	9	RO	mipi_line_byte_err MIPI line byte error; Asserted if- pixel_data_rdy is deasserted before the amount of data specified by line_byte_cnt has been received - the interface will pad the transmitted MIPI packet with unspecified data to maintain transmission integrity.- pixel_valid is continuously asserted for more than line_byte_cnt bytes of data - the interface will discard all data after the first line_byte_cnt bytes. Signal remains asserted until cleared by mipi_line_byte_err_rst or reset_n NOTE: It is assumed line_valid_cnt is programmed with a valid value. Read-only. Legal values: [0,1].
	8:6	X	Reserved

Table 49: TX_SS (continued)

Register Dec(Hex)	Bits	Default	Name
	5	RO	mipi_rdy_for_data MIPI ready to receive data; Asserted once the MIPI interface is enabled and has completed its wake-up and initialization cycles. Read-only. Legal values: [0,1].
	4	RO	mipiccp_idle MIPI idle; Asserted if MIPI interface is not currently transmitting packet data. Read-only. Legal values: [0,1].
	3:1	X	Reserved
	0	RO	mipi_stdby MIPI in standby; Asserted when the MIPI interface has completed the transition to ultra low-power state in response to the assertion of standby_en and remains asserted until standby_en is deasserted. Read-only. Legal values: [0,1].
	MIPI status		
13316 R0x00003404	31:0	0x0000	custom_short_pkt (R/W)
	31:14	X	Reserved
	13	RO	custom_short_pkt_rst Reset for custom short packet; Asserted once the custom short packet has been transmitted to reset custom_short_pkt_req. Since the request may be asserted from an asynchronous, non-continuous clock domain (wrt clk) this signal will remain asserted until it detects that custom_short_pkt_req has transitioned to 0 Read-only. Legal values: [0,1].
	12	0x0000	custom_short_pkt_frame_sync Frame sync for custom short packet; When asserted any requested custom short packet will not be transmitted until after the completion of the current frame. Legal values: [0,1].
	11	0x0000	custom_short_pkt_req Custom short packet request; Asserted to request the insertion of a short packet in the output datastream. Should be left asserted until reset by custom_short_pkt_rst. Must go to 0 before it can be reasserted to make a subsequent request. Legal values: [0,1].
	10:8	0x0000	custom_short_pkt_data_type Custom short packet data type; The three LSBs of the data type field of a custom short packet. Legal values: [0,7].

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	7	0x0000	frame_cnt_en Insert frame counter value in WC field; Asserted if the frame counter value (8-bit) is to be inserted in the WC field of a start or end of frame short packet (else WC field = 0). Signal should be static for the duration of the frame and should only toggle between frames; that is when frame_reg_sync is asserted. NOTE: This signal masks the frame counter in the short packet WC field but does not stop the counter internally. To ensure the counter starts at 1, frame_cnt_rst must be asserted and released with frame_cnt_en. Legal values: [0,1].
	6	0x0000	frame_cnt_rst Reset the frame count sent in the frame start or end short packets. Asserted to reset the frame count transmitted in the frame start or end short packets. Signal must be asserted for at least one clk cycle and should only toggle between frames. That is when frame_reg_sync is asserted. Legal values: [0,1].
	5:0	X	Reserved
13318 R0x00003406	31:0	0x3030	txc_mipi_data_format (R/W)
	31:14	X	Reserved
	13:8	0x0030	stat_fmt_data_type Status Format Data Format 0x30: Operating Mode 1 0x32: Operating Mode 2 Legal values: [0, 63].
	7:6	X	Reserved
	5:0	0x0030	tbn_data_type Thumbnail Data Format 0x1E: YUV422 8-bit 0x22: RGB565 0x30: Operating Mode 1 Legal values: [0, 63].
13320 R0x00003408	31:0	0x0000	txc_mipi_line_byte_cnt (R/W)
	The number of data bytes (not including the packet header or footer) in the line. NOTE: Must be programmed with a valid value for the data format, as specified by the MIPI or CCP specs . For 10-bit data, it should be a multiple of 5, for 12-bit data a multiple of 3 and for YUV/RGB a multiple of 2. Legal values: [0, 65535].		
13322 R0x0000340A	31:0	0x000A	txss_mipi_control_addl (R/W)
	31:5	X	Reserved
	4	0x0000	txss_mipi_heavy_lp_load Addl slew control for load above 45pF
	3:2	0x0002	txss_mipi_delay_trim_d1 Addl control on transition time from low power to high speed modes Legal values: [0,3].
	1:0	0x0002	txss_mipi_delay_trim_d0 Addl control on transition time from low power to high speed modes Legal values: [0, 3].

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
13324 R0x0000340C	31:0	0x0000	custom_short_pkt_wc (R/W)
	The contents of the WC field of a custom short packet Value to insert in the WC field of a custom short packet specified by R0x3404. Legal values: [0, 65535].		
13326 R0x0000340E	31:0	0x0000	mipi_test (R/W)
	31:13	X	Reserved
	12	0x0000	io_tst Tristate mipi io's for testing; 1: Force IO lanes to go tri-state for test Legal values: [0,1].
	11:10	X	Reserved
	9	RO	test_checksum_valid 1: Test checksum is outputting a valid checksum Read-only. Legal values: [0,1].
	8	0x0000	test_start_checksum 1: A 16-bit checksum will be calculated over the next frame Legal values: [0,1].
	7	X	Reserved
	6	0x0000	mipi_test_ccp_mipin_sel If mipi_test_enable is asserted; 0: transmit test signals on MIPI link, 1: transmit test signals on CCP link. Legal values: [0,1].
	5	0x0000	prbs_511bit_test_en 511-bit PRBS test enable; When asserted input data is replaced with the 511-bit prbs (x9 + x5 + 1) test sequence. Signal should only toggle between frames ie. when frame_reg_sync is asserted. The sequence will be transmitted over a single data lane(regardless of enable_dp1_out setting) and the packet containing the data is given a data type code of 0x33. Legal values: [0,1].
	4	0x0000	txss_mipi_test_en MIPI test enable; Enables test data to be driven on the phy outputs Legal values: [0,1].
	3	X	Reserved

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	2:0	0x0000	txss_mipi_test_mode MIPI test mode; Defines test mode to be applied if test_en is asserted:MIPI (test_ccp_mipin_sel = 0) 0x0: Force data and clock lane outputs to LP-00 0x1: Force data and clock lane outputs to LP-1 0x2: Force data and clock lane outputs to HS-0 0x3: Force data and clock lane outputs to HS-1 0x4: Drive a fixed frequency (1/2 serial data rate) HS square wave out on the data and clock lanes 0x5: Drive a fixed frequency (1/8, 1/10, 1/12 serial data rate, depending on the setting of data_type) HS square wave out on the data and clock lanes 0x6: Drive a LP square wave at half the clk_op frequency on both the clock and data lanes 0x7: Drive a continuous (non-packetized) HS pseudo random sequence (231 -1 bits long) on the data lane. When test_en is asserted, test data will be driven on both data lanes of a two lane configuration regardless of the state of enable_dp1_out Legal values: [0,7].
13328 R0x00003410	31:0	0x0800	mipi_timing_t_hs_zero (R/W)
	31:12	X	Reserved
	11:8	0x0008	t_hs_zero Time to drive HS-0 before the sync sequence Legal values: [0,15].
	7:0	X	Reserved
13330 R0x00003412	31:0	0x0705	mipi_timing_t_hs_exit_hs_trail (R/W)
	31:14	X	Reserved
	13:8	0x0007	t_hs_exit Time to drive LP-11 after HS burst Legal values: [0,63].
	7:4	X	Reserved
	3:0	0x0005	t_hs_trail Time to drive flipped differential state after last payload data bit of a HS transmission burst Legal values: [0,15].
13332 R0x00003414	31:0	0x0B01	mipi_timing_t_clk_post_clk_pre (R/W)
	31:14	X	Reserved
	13:8	0x000B	t_clk_post Time to drive the HS clock after the data lane has gone into low powermode Legal values: [0,63].
	7:6	X	Reserved
	5:0	0x0001	t_clk_pre Time to drive the HS clock signal before any data lane might start up Legal values: [0,63].

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
13334 R0x00003416	31:0	0x0509	mipi_t_clk_trail_clk_zero (R/W)
	31:12	X	Reserved
	11:8	0x0005	t_clk_trail Time to drive HS differential state after last payload clock bit of a HS transmission burst Legal values: [0,15].
	7:6	X	Reserved
	5:0	0x0009	t_clk_zero Minimum lead HS-0 clock lane drive period before starting clock Legal values: [0,63].
13336 R0x00003418	31:0	0x0004	mipi_timing_t_lpx (R/W)
	31:6	X	Reserved
	5:0	0x0004	mipi_t_lpx Length of any low power state period Legal values: [0,63].
13338 R0x0000341A	31:0	0x0608	mipi_init_timing (R/W)
	31:15	X	Reserved
	14:8	0x0006	t_init Initialisation time when first first entering stop state (LP-11) after power-up/ reset. LP-11 is transmitted for a minimum of 1K (1024) * t_init output clock cycles Legal values: [0,127].
	7	X	Reserved
	6:0	0x0008	t_wake_up Recovery time from ultra low power mode. ULPM is exited by applying a mark state for 8K (8192) * t_wake_up output clock cycles Legal values: [0,127].
13340 R0x0000341C	31:0	0x0000	mipi_ccp__sel_class (R/W)
	31:2	X	Reserved
	1	0x0000	mipi_ccp_class Select ccp class; 0: class 0 1: class 1 / 2 Legal values: [0,1].
	0	0x0000	mipi_ccp_mipin_sel 0: Use mipi interface 1: Use ccp interface Legal values: [0,1].
13342 R0x0000341E	31:0	0x0000	mipi_test_checksum (RO)
	If test_checksum_valid is asserted, this is a valid checksum calculated over a complete frame. Read-only. Legal values: [0, 65535].		

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15360 R0x00003C00	31:0	0x3333	jpssmode (R/W)
	31:15	X	Reserved
	14:12	RO	jpss_mode_fsyncd_jpss_out A frame boundary synchronized version of JPSS_MODE (JPEG Sub System process mode). This reflects the currently active JPSS_MODE at the output of the jpeg_ss module. Read-only. Legal values: [0, 7].
	11	X	Reserved
	10:8	RO	jpss_mode_fsyncd_jpeg_out A frame boundary synchronized version of JPSS_MODE (JPEG Sub System process mode). This reflects the currently active JPSS_MODE at the output of the JPEG compression module. Read-only. Legal values: [0, 7].
	7	X	Reserved
	6:4	RO	jpss_mode_fsyncd_jpss_in A frame boundary synchronized version of JPSS_MODE (JPEG Sub System process mode). This reflects the currently active JPSS_MODE at the input to the jpeg_ss module. Read-only. Legal values: [0, 7].
	3	X	Reserved
	2:0	0x0003	jpss_mode JPEG Sub System process mode. This value is synced to an input frame boundary before being latched and used. 0 - Output disabled. Discard input data. 1 - Bayer data 2 - Reserved (Same as 0) 3 - Processed data, RGB or YUV (Jpeg bypass) 4 - Jpeg active but output discarded 5 - Thumbnail output. Jpeg active but output discarded 6 - Jpeg 7 - Jpeg with embedded thumbnail See JPSS_SOURCE_RX for input source control. Writes are synchronized to frame boundaries. Legal values: [0, 7].
15362 R0x00003C02	31:0	0x0610	jpss_ctrl (R/W)
	31:15	X	Reserved
	14	0x0000	jpss_exif_tn_ptr_en Append JPSS_EXIF_TN_PTR to the end of the status segment This field should only be changed between frames.

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	13	0x0000	jpss_use_datatypes Use separate lines for JPEG/Thumbnail/Status data. In JPSS_MODE 6, this bit will determine whether the status segment is included as part of the last JPEG spoof line or sent in a separate line/packet after the JPEG data. Same for JPSS_MODE 7. Plus, this bit will determine whether the thumbnail data is embedded in the JPEG data or sent in separate lines/packets. This register only applies when JPSS_SPOOF_ENABLE = 1. If JPSS_SPOOF_ENABLE = 0, thumbnail is always embedded in the JPEG data and the status data is always appended to the JPEG data. 0 - Embed (interleave) thumbnail data in JPEG data before 'spoof'ing. Append status segment to end of JPEG data. 1 - Send thumbnail and status data in separate lines/packets from the jpeg data. Thumbnail lines will be interleaved with jpeg lines. Status segment will be after the JPEG and Thumbnail lines. This field should only be changed between frames.
	12	0x0000	jpss_devware_bitremap Remap bit order in JPSS_MODE 3 If set in JPSS_MODE 3, the bit order will be changed from 15:0 to {15:8, 1:0, 3:2, 5:4, 7:6} to support outputting bayer data packed into a 16 bit word in the order expected by the devware system. 0 - No bit remap 1 - Remap bit order This field should only be changed between frames.
	11	0x0000	jpss_jpbyp_byte_swap Byte swap for JPEG bypass mode. JPEG bypass image data is created with 16 bit pixels, typically YCbCr 4:2:2, or RGB 565. The final output bus is 8 bits wide. This control bit swaps the byte order within each pixel. This bit only causes byte swapping in the jpeg bypass data, not the jpeg data, the cpipe bypass nor in the thumbnail data. Note that there is a global byte swap bit, called txc_line_enable_bytes_swap, that swaps the byte order for all output data. 0 - No byte swap: Cb, Y, Cr, Y order. 1 - Swap bytes order: Y, Cb, Y, Cr order. This field should only be changed between frames.
	10	0x0001	jpss_ycc_data Specify format of CPIPE output. Used do define the value to be used for a black pixel during padding. 0 - RGB or RAW input 1 - YCbCr input This field should only be changed between frames.
	9	0x0001	jpss_spoof_height_ignore Output Buffer Spoof Control This register is used in conjunction with JPSS_SPOOF_ENABLE. When set to 1, ignores the JPSS_SPOOF_HEIGHT register. The spoof frame terminates after all received data has been read and padded to the programmable JPSS_SPOOF_WIDTH. This field should only be changed between frames or when JPEG is inactive.

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	8	0x0000	jpss_spoof_enable JPEG Transmit Mode 0 - Continuous Mode 1 - Spoof Mode This bit is only used if JPSS_MODE is either JPEG or JPEG w/ Thumbnail This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 2].
	7	0x0000	jpss_source_rx Muxes jpeg_ss input source between CPIPE_SS and RX_SS. 0 - Take data from CPIPE. 1 - Take data from RX block. This field should only be changed between frames or when JPEG is inactive.
	6	0x0000	jpss_32_bit_compatible If set and configured for jpeg continuous protocol, the jpeg output, is forced to be a multiple of 4 bytes (32 bits) long. Padding bytes of 0xff are added after the last jpeg byte and before the status (if enabled) to meet this requirement. 0 - 8 bit compatible 1 - 32 bit compatible This field should only be changed between frames or when JPEG is inactive.
	5	0x0000	jpss_use_legacy_status Setting this bit swaps the order of the JPEG status bytes and the JPEG data byte count fields of the status segment. It also make the length a 3 byte field instead of 4 and the status a 3 byte field instead of 2. It does not affect any other fields in the status segment. 0 - 4 byte length followed by 2 byte status. 1 - 3 byte status followed by 3 byte length. This field should only be changed between frames or when JPEG is inactive.
	4	0x0001	jpss_enable_status_markers If JPSS_INSERT_JPEG_STATUS and this bit is enabled, the status data will be preceded by the JPSS_JPEG_SOSI_CODE marker and followed by the JPSS_JPEG_EOSI_CODE marker. 0 - No status marker codes. 1 - Use SOSI and EOSI marker codes This field should only be changed between frames or when JPEG is inactive.
	3	0x0000	jpss_enable_legalize_status Enable Legalize Status Fields When this bit is set, all the status segment data will be split into nibbles and every nibble is prefixed with 0xA to avoid possible 0xff code. This field should only be changed between frames or when JPEG is inactive.
	2	0x0000	jpss_enable_resolution Enable JPEG/Thumbnail Resolution If set, the uncompressed JPEG image size (width and height) and the thumbnail size (width and height) will be inserted into the status segment. If thumbnail is not enabled, the thumbnail size will be set to zero. This field should only be changed between frames or when JPEG is inactive.
	1	0x0000	jpss_enable_index_table Enable Thumbnail Index Table When this bit is set, the thumbnail index table will be inserted in the JPEG data steam right after the SOSI in the status segment. This field should only be changed between frames or when JPEG is inactive.

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	0	0x0000	jpss_insert_jpeg_status Enable insertion of JPEG Status Field This bit applies to JPEG compressed stream only. When this bit is set, a status segment is appended at the end of each jpeg segment. The status segment will contain at least a length field and a set of status bits. Other fields and markers can be enabled/disabled. This field should only be changed between frames or when JPEG is inactive.
15364 R0x00003C04	31:0	0x0A20	jpss_inwid (R/W)
	JPEG_SS Input Image Width Width of input image to JPEG subsystem. Image width will be forced to this size by cropping from right or padding on right with black pixels. Must be set for all JPSS_MODE settings. This field should only be changed between frames. Legal values: [8, 4095].		
15366 R0x00003C06	31:0	0x0798	jpss_inhgt (R/W)
	JPEG_SS Input Image Height Height of input image to JPEG subsystem. Image height will be forced to this size by cropping from right or padding on right with black pixels. Must be set for all JPSS_MODE settings. This field should only be changed between frames. Legal values: [8, 4095].		
15368 R0x00003C08	31:0	0x0500	jpss_spoof_width (R/W)
	Output Buffer Spoof Width (in bytes) This register defines the width for the spoof frame in bytes. The maximum value when JPSS_MODE = 6 (JPEG without thumbnail) is 5184. The maximum value when JPSS_MODE = 7 (JPEG with thumbnail) is 2560. The minimum value is 128. Odd values for this register are not allowed. This field should only be changed between frames or when JPEG is inactive. Legal values: [128, 5184].		
15370 R0x00003C0A	31:0	0x01E0	jpss_spoof_height (R/W)
	Output Buffer Spoof Height This field defines the height of the spoof frame in lines. If the JPSS_CTRL[JPSS_SPOOF_HEIGHT_IGNORE] field is set to 1, this field is ignored and the spoof height is made as large as necessary to contain the JPEG data. This field should only be changed between frames or when JPEG is inactive. Legal values: [8, 4095].		
15372 R0x00003C0C	31:0	0xFFBC	jpss_jpeg_sosi_code (R/W)
	JPEG Start of Status Information code This value is used in conjunction with JPEG_EOSI_CODE to encapsulate the status information. This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 65535].		
15374 R0x00003C0E	31:0	0xFFBD	jpss_jpeg_eosi_code (R/W)
	JPEG End of Status Information code This value is used in conjunction with JPEG_SOSI_CODE to encapsulate the status information. This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 65535].		
15376 R0x00003C10	31:0	0xFFFF	jpss_dummy_pattern (R/W)
	Output Buffer Dummy Pattern In spoof mode, dummy pattern shall be padded if the JPEG data is not enough to fill in the size of spoof_width x spoof_height. This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 65535].		

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15378 R0x00003C12	31:0	0x0FFF	jpss_trigger_mark (R/W)
	Output Buffer Trigger Mark In non-JPEG modes and JPEG spoof mode, line output will start when output buffer contains a full output line of data OR when output buffer contains a EOF OR when output buffer contains more than JPSS_TRIGGER_MARK pixels. Set JPSS_TRIGGER_MARK to 0xFFF to disable. It is expected that this register will not need to be changed from the default, disabled value. Legal values: [0, 4095].		
15380 R0x00003C14	31:0	0x0000	jpss_exif_tn_ptr (R/W)
	Optional pointer to add to end of status segment These two bytes will be added to the end of the status segment if enabled by JPSS_EXIF_TN_PTR_EN. Legal values: [0, 65535].		
15382 R0x00003C16	31:0	0x0000	jpss_status (R/W)
	31:16	X	Reserved
	15	0x0000	jpss_inhgt_undersize JPEG_SS Input Width Undersize When asserted, it indicates that the frame input from CPIPE or RX was shorter than JPSS_INHGT and the height was padded to match JPSS_INHGT. The status flag remains set until the firmware writes 1 to clear this bit.
	14	0x0000	jpss_inwid_undersize JPEG_SS Input Width Undersize When asserted, it indicates that the frame input from CPIPE or RX was narrower than JPSS_INWID and the width was padded to match JPSS_INWID. The status flag remains set until the firmware writes 1 to clear this bit.
	13	0x0000	jpss_inhgt_oversize JPEG_SS Input Height Oversize When asserted, it indicates that the frame input from CPIPE or RX was taller than JPSS_INHGT and the height was truncated to match JPSS_INGHT. The status flag remains set until the firmware writes 1 to clear this bit.
	12	0x0000	jpss_inwid_oversize JPEG_SS Input Width Oversize When asserted, it indicates that the frame input from CPIPE or RX was wider than JPSS_INWID and the width was truncated to match JPSS_INWID. The status flag remains set until the firmware writes 1 to clear this bit.
	11:10	0x0000	jpss_flbuf_watermark JPEG BUFFER Watermark 0- It indicates the highest utilization of JPEG Buffer is between 0% and 25%. 1- It indicates the highest utilization of JPEG Buffer is between 25% and 50%. 2- It indicates the highest utilization of JPEG Buffer is between 50% and 75%. 3- It indicates the highest utilization of JPEG Buffer is between 75% and 100%. The watermark is cleared when the firmware writes 2'b11 to this field. Legal values: [0, 3].

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	9:8	0x0000	jpss_tnbuf_watermark Thumbnail FIFO Watermark 0- It indicates the highest utilization of Thumbnail FIFO is between 0% and 25%. 1- It indicates the highest utilization of Thumbnail FIFO is between 25% and 50%. 2- It indicates the highest utilization of Thumbnail FIFO is between 50% and 75%. 3- It indicates the highest utilization of Thumbnail FIFO is between 75% and 100%. The watermark is cleared when the firmware writes 2'b11 to this field. Legal values: [0, 3].
	7	0x0000	jpss_spoof_ovsize Spoof Oversize Status When asserted, it indicates that the spoof frame size is too small for the JPEG encoded stream. The status flag remains set until the firmware writes 1 to clear this bit.
	6	0x0000	jpss_frame_ovf Frame Overflow Status If asserted, it indicates the previous frame had not finished when a new frame came in. The new frame is discarded. This status flag remains set until the firmware writes 1 to this field.
	5	0x0000	jpss_flbuf_undf JPEG Buffer Underflow Status When asserted, it indicates that an underflow condition is detected in the JPEG Buffer during the frame transfer. The status flag remains set until the firmware writes 1 to clear this bit.
	4	0x0000	jpss_flbuf_ovf Full Line Buffer Overflow Status When asserted, it indicates that an overflow condition has been detected. If jpeg is enabled, JPEG_L2B_OVF will also be set. The status flag remains set until the firmware writes 1 to clear this bit.
	3	0x0000	jpss_tnbuf_undf Thumbnail Output Buffer Underflow Status When asserted, it indicates that an underflow condition is detected in the Thumbnail FIFO during the frame transfer. The status flag remains set until the firmware writes 1 to clear this bit.
	2	0x0000	jpss_tnbuf_ovf Thumbnail Output Buffer Overflow Status When asserted, it indicates that an overflow condition is detected in the Thumbnail FIFO during the frame transfer and that the Thumbanil transfer has been terminated prematurely. The status flag remains set until the firmware writes 1 to clear this bit.
	1	0x0000	jpss_frame_tx_done Output Buffer Transfer Frame Done Status When asserted, it indicates that the Output Buffer has transferred current frame to outgoing interface. The status flag remains set until the firmware writes 1 to clear this bit.

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	0	0x0000	jpss_frame_rx_done Output Buffer Frame Received Done Status If asserted, it indicates that a frame has been received by Output Buffer. This bit is cleared only when the firmware writes 1 to this field.
15384 R0x00003C18	31:0	0x0000	jpss_frame_length0 (RO)
	Frame byte count (16 LSBs) It is the lower 16 bits of 24-bit frame byte length. The function of this registers varies with JPSS_MODE. Value in this register is updated on each EOF except as noted below. JPSS_MODE 0 (jpeg_ss input discarded) - Holds value from most recently output frame. Not updated for each discarded input frame. 1 (Bayer) - 2 * the number of pixels in the frame. 2 (same as 0) - Same as 0. 3 (YUV/RGB) - 2 * the numbere of pixels in the frame. 4 (jpeg discarded) - The number of bytes in the discarded frame from the JPEG encoder. 5 (thumbnail only) - The number of bytes in the discarded frame from the JPEG encoder. 6 (jpeg only) - The number of bytes in the frame from the JPEG encoder. 7 (jpeg with thumbnail) - If JPSS_USE_DATATYPES then this register holds the number of bytes in the frame from the JPEG encoder. If not JPSS_USE_DATATYPES, then this register holds the sum of the bytes in the JPEG and thumbnail images. In the jpeg modes, if the image is truncated by the spoof logic, this register will include the dropped bytes in the count. Read-only. Legal values: [0, 65535].		
15386 R0x00003C1A	31:0	0x0000	jpss_frame_length1 (RO)
	Frame byte count (8 MSBs) It is the upper 8 bits of 24-bit frame byte length. See JPSS_FRAME_LENGH0 for full description. Read-only. Legal values: [0, 255].		
15388 R0x00003C1C	31:0	0x0000	jpss_rcvd_frame_cnt (R/W)
	Jpeg_ss Received Frame Counter This register is for debug purpose only. It captures the number of frames received by jpeg_ss. When this counter reaches its maximum number, 16'hffff, it rolls over to 0. However, the firmware can write 16'hffff to clear it at any time. Legal values: [0, 65535].		
15390 R0x00003C1E	31:0	0x0000	jpss_rcvd_frame_line_cnt (RO)
	Jpeg_ss Received Frame Line Counter It captures the number of lines in the last frame received by Output Buffer. Read-only. Legal values: [0, 4095].		
15394 R0x00003C22	31:0	0x0000	jpss_tx_frame_cnt (R/W)
	Jpeg_ss Transmitted Frame Counter This register is for debug purpose only. It captures the number of frames sent to the TX_ss by jpeg_ss. When this counter reaches its maximum number, 16'hffff, it rolls over to 0. However, the firmware can write 16'hffff to clear it at any time. Legal values: [0, 65535].		

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15396 R0x00003C24	31:0	0x0000	crc_control (R/W)
	31:2	X	Reserved
	1	0x0000	crc_enable 0: Disable 1: Enable
	0	0x0000	crc_continue_acc 0: Accumulate for one frame 1: Accumulate continuously
15398 R0x00003C26	31:0	0x0000	crc_data (RO)
	Accumulated CRC data for read out. Read-only. Volatile. Legal values: [0, 65535].		
15400 R0x00003C28	31:0	0x0FFF	jpss_between_frame_status (RO)
	31:12	X	Reserved
	11	RO	jpss_out_bf Jpeg_ss Output Data Bus Between Frames Observes image data bus at output of jpeg_ss module. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	10	RO	jpss_spoof_out_bf Jpeg_ss Spoof Gen Output Data Bus Between Frames Observes image data bus at interface between jpeg_ss Spoof Generation and jpeg_ss status generation modules. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	9	RO	jpss_outmux_out_bf Jpeg_ss Outmux Output Data Bus Between Frames Observes image data bus at interface between jpeg_ss outmux and jpeg_ss spoof generation modules. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	8	RO	jpss_tnbuf_out_bf Thumbnail Buffer Output Data Bus Between Frames Observes image data bus at interface between Thumbnail Buffer and jpeg_ss output mux modules. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	7	RO	jpss_tngen_out_bf Thumbnail Generation Output Data Bus Between Frames Observes image data bus at interface between thumbnail generation and thumbnail line buffer modules. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	6	RO	jpss_tngen_in_bf Thumbnail Generation Data Input Bus Between Frames Observes image data bus at input to Thumbnail Generation module. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	5	RO	jpss_flbuf_out_bf Full Line Buffer Output Bus Between Frames Observes image data bus at interface between Full Line Buffer and jpeg_ss output mux modules. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	4	RO	jpss_jpeg_out_bf Jpeg Output Bus Between Frames Observes image data bus at interface between JPEG core and Full Line Buffer modules. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	3	RO	jpss_jpeg_in_bf Jpeg Data Input Bus Between Frames Observes image data bus at interface between JPEG Line to Block and JPEG compression core modules. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	2	RO	jpss_l2b_in_bf Jpeg Line to Block Data Input Bus Between Frames Observes image data bus at input of JPEG Line to Block module. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	1	RO	jpss_flbuf_njp_in_bf Non-JPEG Full Line Buffer Input Data Bus Between Frames Observes image data bus at interface between JPSS_IN_PREP and Full Line Buffer modules. This bus is only active in JPSS_MODES for CPIPE bypass and JPEG bypass. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
	0	RO	jpss_in_bf Jpeg_ss Data Input Bus Between Frames Observes image data bus at input of jpeg_ss. Goes high when an EOF is detected (or at reset) and goes low when a SOF is detected. Read-only.
15424 R0x00003C40	31:0	0x000E	jpeg_ctrl (R/W)
	31:9	X	Reserved
	8	0x0000	jpeg_hdr_pad_en Enable use of JPEG_HDR_PAD_BYTES register. If set, the JPEG header will be padded with JPEG_HDR_PAD_BYTES after the SOF segment and before the SOS segment. 0 - Automatically pad JPEG header so it is a multiple of 8 bytes long. 1 - Pad JPEG header with JPEG_HDR_PAD_BYTES. This field should only be changed between frames or when JPEG is inactive.

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	7	0x0000	jpeg_insert_rajpeg_markers SCALADO/RAJPEG Marker Insertion Enable Enable insertion of RAJPEG markers in the compressed jpeg data stream. Should not be enabled with Restart markers enabled or with embedded thumbnail. 0 - Disabled 1 - Enabled This field should only be changed between frames or when JPEG is inactive.
	6:4	0x0000	jpeg_hdr_ctrl JPEG Header Control 0 = Disabled, no header. 1 = Add SOI (0xff, 0xd8) and EOI (0xff, 0xd9) markers 2 = Add SOI, 4 byte Qtable index header, and EOI 3 = Add JFIF segment to JPEG header 4 = Add EXIF, FW defined segment to JPEG header 5 = Add non-EXIF, FW defined segment to JPEG header 6 = Reserved 7 = Reserved The 4 byte Qtable index header is 0xff, 0xda, 0x00, 'JPEG_CURRENT_QTABLE' This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 4].
	3	0x0001	jpeg_jpop_ofl_en JPEG JPOP Overflow Prevention Enable Limit amount of data from JPEG engine to prevent data FIFO Overflows. 0 - Disabled 1 - Enabled This field should only be changed between frames or when JPEG is inactive.
	2	0x0001	jpeg_jpop_cr_en JPEG JPOP Compression Ratio Limit Enable Limit amount of data from JPEG engine to ensure the image compression ratio is greater than JPOP_LIMIT. 0 - Disabled 1 - Enabled This field should only be changed between frames or when JPEG is inactive.
	1:0	0x0002	jpeg_ds_mode JPEG downsample mode. The down sample format used for JPEG compression. 0 = Grayscale 1 = reserved 2 = 422 3 = 420 This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 3].
15426 R0x00003C42	31:0	0x0000	jpeg_qtable_sel (R/W)
	JPEG Qtable Selector. Selects between the 2 pairs of Y/C qtables. 0 = Use Y Qtable at 0-63 and C Qtable at 64-127. 1 = Use Y Qtable at 128-191 and C Qtable at 192-255. 2 = Reserved 3 = Reserved Changes to this field are sync'd to frame boundaries. Writes are synchronized to frame boundaries.		

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15428 R0x00003C44	31:0	0x0000	jpeg_restart (R/W)
	Number of MCU's per restart marker interval. Set to 0 to have no restart markers inserted in the JPEG stream This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 65535].		
15430 R0x00003C46	31:0	0xFFFF	jpeg_jpop_limit (R/W)
	JPEG JPOP Compression Ratio Limit Control This register defines the minimum compression ratio for the JPEG Overflow Prevention control system (JPOP). If enabled by JPEG_CTRL[JPEG_JPOP_CR_EN], the JPOP system will change the compression process used by JPEG during a frame as needed to achieve a compression ratio greater than the defined minimum. This change to the JPEG process will result in reduced image quality for that frame. It is intended that JPOP only be used to prevent the compressed data from exceeded the system bandwidth or storage limits. For JPEG_DS_MODE of 422 or Grayscale, the value to use for this register is derived based on the equation: Bytes_out = JPEG_JPOP_LIMIT * pixels / 2^16 For JPEG_DS_MODE of 420, the value to use for this register is derived based on the equation: Bytes_out = JPEG_JPOP_LIMIT * pixels * 3 / 2^18 Note that Bytes_out can exceed the amount calculated above by a small amount for 2 reasons. 1. The JPOP function does not count the bytes in the header. 2. The JPOP function allows the "target" defined by JPOP_LIMIT to be exceeded before adjusting the compression process. In the worst case, this could be as much as 2K bytes, but is typically much less. This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 65535].		
15432 R0x00003C48	31:0	0x0000	jpeg_exif_bytes (R/W)
	JPEG EXIF Record size in bytes This register defines the number of bytes to read from the EXIF header RAM for insertion into the JPEG header. The max value is 768. This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 768].		
15434 R0x00003C4A	31:0	0x0000	jpeg_hdr_pad_bytes (R/W)
	Number of bytes to insert into the JPEG header for padding between the SOF and SOS segments. (Padding should result in a total header size that is an even number of bytes.) This field should only be changed between frames or when JPEG is inactive. Legal values: [0, 1023].		

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15454 R0x00003C5E	31:0	0x0000	jpss_debug_sel (R/W)
	PROTO ONLY! Select internal bus to output on debug port. 00000 - JP1_FDCTB output 00001 - JP1_ZZ2B output 00010 - JP1_DCDIFFB output 00011 - JP1_RLEB output 00100 - JP1_VLCB output 00101 - JP1_CODEPACKB output 00110 - JP1_ILBUFB output 00111 - JP1_BITPACK output 01000 - L2B input 01001 - L2B output 01010 - JPCORE output 10000 - TNGEN input 10001 - Scaler output 10010 - YCC2RGB output 10011 - TNCROP output 10100 - TNBUF output 10101 - TN Hdrgen output 11000 - FLBUF input (JPEG bypass) 11001 - FLBUF output 11010 - Frame overflow output 11011 - Embed Mux output 11100 - Spoof gen output 11101 - Nonembed Mux output 11110 - Status gen output Legal values: [0, 10].		
15456 R0x00003C60	31:0	0x0000	jpeg_hw_status (R/W)
	31:9	X	Reserved
	8	0x0000	jpeg_l2b_ovf JPEG Line to Block Overflow Status When asserted, it indicates that an overflow condition has been detected in the JPEG line to block logic. Image data has been corrupted and should be discarded. The status flag remains set until the firmware writes 1 to clear this bit.
	7:4	RO	jpeg_jpop_status JPEG JPOP Max Level In Previous Frame Indicates the maximum level activation in the most recently completed frame. 0 means no JPOP activation. Read-only. Legal values: [0, 15].
	3:2	RO	jpeg_last_qtable JPEG Qtable Index For Previous Frame QTABLE index for the most recently completed frame. This value is updated on EOI out of JPEG. Read-only.
	1:0	RO	jpeg_cur_qtable JPEG Qtable Index For Current Frame Shows which Qtables are being used for the currently processed frame, or if between frames, which Qtables will be used on the next frame. Read-only.

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15488 R0x00003C80	31:0	0x0215	tn_ctrl (R/W)
	31:13	X	Reserved
	12	0x0000	tn_swap_cbcr Swap order of Cb and Cr output of YCbCr thumbnail. 0: Output order is Cb Y Cr Y ... 1: Output order is Cr Y Cb Y ... This field should only be changed between frames or when thumbnail is inactive.
	11	0x0000	tn_c_level_shift Level shift Cb/Cr values of YCbCr thumbnails. 0: Cb/Cr value output range is 0:255 1: Cb/Cr value output range is -128:127 This field should only be changed between frames or when thumbnail is inactive.
	10	0x0000	tn_y_level_shift Level shift Y values of YCbCr thumbnails. 0: Y value output range is 0:255 1: Y value output range is -128:127 This field should only be changed between frames or when thumbnail is inactive.
	9	0x0001	tn_legalize_data Clip max thumbnail data values to prevent 0xff values For YCbCr, the max value of all bytes is 254. For RGB, the Red and Blue data range is 0-30 and the Green data range is 0-62. 0: Disabled 1: Enabled This field should only be changed between frames or when thumbnail is inactive.
	8	0x0000	tn_scale_sharp_enable Set thumbnail scale sharp. 0: Disabled 1: Enabled This field should only be changed between frames or when thumbnail is inactive.
	7	0x0000	tn_scale_hpmode Set thumbnail scaler to high precision mode. This field should only be changed between frames or when thumbnail is inactive.
	6	0x0000	tn_scale_u_first In thumbnail scaler, select U component as the first pixel of row.. This field should only be changed between frames or when thumbnail is inactive.
	5	X	Reserved
	4	0x0001	tn_scale_enable Set thumbnail scale enable. 0: Disabled 1: Enabled This field should only be changed between frames or when thumbnail is inactive.

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	3	0x0000	tn_byte_swap Swap thumbnail byte order. Thumbnail image data is created with 16 bit pixels, either YCbCr 4:2:2, or RGB 565. The final output bus is 8 bits wide. This control bit swaps the byte order within each pixel. See TN_FORMAT_RGB565 for the unswapped order. This bit can be used to reverse the order of the Y and Cb/Cr bytes in the thumbnail data steam. This bit only causes byte swapping in the thumbnail data, not the jpeg data it may be embedded in nor any other non-thumbnail data. It also does not affect the order of the TN_START/END_CODEs. This field should only be changed between frames or when thumbnail is inactive.
	2	0x0001	tn_format_rgb565 Set thumbnail format to RGB565. If 0, thumbnail format will be YCbCr422 packed in CbY,CrY order. If 1, thumbnail format will be RGB565 packed in {R[7:3],G[7:5]}, {G[4:2],B[7:3]} order. This field should only be changed between frames or when thumbnail is inactive.
	1:0	0x0001	tn_line_marker_format Determines marker format for each thumbnail line that is inserted/embedded in the JPEG data stream. 0 - No TN markers 1 - Insert a TN_START_CODE at the start of each TN line. Insert a TN_END_CODE at the end of each TN line 2 - Like 1, except insert a 4 byte image size header after the TN_START_CODE marker of each thumbnail line 3 - insert a TN_START_CODE at the start of each TN followed by a 2 byte length field that specifies the number of bytes in the Line + 2. Length_field = 2 * thumbnail_width + 2 If TN_START_CODE is set to 0xFFFE, then this looks like a JPEG comment which may be ignored in some decoders. This field should only be changed between frames or when thumbnail is inactive.
15490 R0x00003C82	31:0	0x01FA	tn_scalar_x_ratio (R/W)
	Horizontal Thumbnail Scalar Factor This field should only be changed between frames or when thumbnail is inactive. Horizontal Scaling Weight = int_roundup((Output_X_Size / ScaleInput_X_Size) * 2048). Weight = 2048 implies unity scale Legal values: [1, 2048].		
15492 R0x00003C84	31:0	0x01FA	tn_scalar_y_ratio (R/W)
	Vertical Thumbnail Scalar Factor This field should only be changed between frames or when thumbnail is inactive. Vertical Scaling Weight = int_roundup((Output_Y_Size / ScaleInput_Y_Size) * 2048). Weight = 2048 implies unity scale Legal values: [1, 2048].		
15494 R0x00003C86	31:0	0x0280	tn_crop_wid (R/W)
	Thumbnail Scalar Output Image Width Width of output image from Thumbnail Scalar. Image will be cropped to this size after scalr. Max value is 640. Thumbnail width is required to be a multiple of 2. This field should only be changed between frames or when thumbnail is inactive. Legal values: [8, 640].		

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15496 R0x00003C88	31:0	0x01E0	tn_crop_hgt (R/W)
	Thumbnail Scalar Output Image Height Height of output image from Thumbnail Scalar. Max value is 480. This field should only be changed between frames or when thumbnail is inactive. Legal values: [8, 480].		
15498 R0x00003C8A	31:0	0x0000	tn_crop_tl (R/W)
	31:16	X	Reserved
	15:8	0x0000	tn_crop_top Thumbnail Crop Top Number of rows to remove from top edge of thumbnail image (after thumbnail scaler). This register is for use when the JPEG aspect ratio and the thumbnail aspect ratio do not match. The thumbnail should be cropped so that the thumbnail does not look distored. This field should only be changed between frames or when thumbnail is inactive. Legal values: [0, 255].
	7:0	0x0000	tn_crop_left Thumbnail Crop Left Number of columns to remove from left edge of thumbnail image (after thumbnail scalar). This register is for use when the JPEG aspect ratio and the thumbnail aspect ratio do not match. The thumbnail should be cropped so that the thumbnail does not look distored. This field should only be changed between frames or when thumbnail is inactive. Legal values: [0, 255].
15500 R0x00003C8C	31:0	0xFFBE	tn_start_code (R/W)
	Thumbnail Embedded Data Start Code This 2 byte code will be inserted at the start of each thumbnail line that is embedded inside the JPEG data. This field should only be changed between frames or when thumbnail is inactive. Legal values: [0, 65535].		
15502 R0x00003C8E	31:0	0xFFBF	tn_end_code (R/W)
	Thumbnail Embedded Data End Code This 2 byte code will be inserted at the end of each thumbnail line that is embedded inside the JPEG data. This field should only be changed between frames or when thumbnail is inactive. Legal values: [0, 65535].		
15504 R0x00003C90	31:0	0x1208	tn_scale_sharp_ctrl (R/W)
	31:14	X	Reserved
	13:11	0x0002	tn_scale_sharp_gain_exp Scale Sharp Exponent Legal values: [0, 7].
	10:8	0x0002	tn_scale_sharp_gain_mant Scale Sharp Gain Legal values: [0, 7].
	7:0	0x0008	tn_scale_sharp_knee Scale Sharp Threshold Legal values: [0, 255].

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15520 R0x00003CA0	31:0	0x0040	txss_parameters (R/W)
	31:7	X	Reserved
	6	0x0001	txss_invert_pclk At the output of the pad: 0: Non-inverted PCLK 1: Inverted PCLK
	5	0x0000	txss_enable_fv_trail_pclk 0: FV synced with PCLK 1: Add 8 pixel clock cycles after the falling edge of frame valid when it is turned off during vertical blanking.
	4:3	0x0000	txss_output_data_select 00: Normal image data 01: Reserved 10: Moto Test Pattern data 11: GPIO Signal Legal values: [0, 3].
	2:1	X	Reserved
	0	0x0000	txss_output_interface TX Output Interface: 0: Parallel Output, 1: MIPI/CCP Output Legal values: [0,1].
15522 R0x00003CA2	31:0	0x0007	txc_parameters (R/W)
	31:8	X	Reserved
	7	0x0000	txc_mipi_enable_line_byte_cnt Asserted for color pipe bypass mode, MIPI output. User to configure 0x3408 with 1.25 * and 1.5 * line_pixel_count for RAW10 and RAW16 data, respectively. Legal values: [0,1].
	6	0x0000	txc_line_insert_ccir_codes Enable insertion of CCIR markers in frame. Legal values: [0,1].
	5	0x0000	txc_cont_dup_fv_to_be_lv Duplicate frame valid signal to be line valid signal also. Applies to Parallel Output and JPEG continuous data only. Legal values: [0,1].
	4	0x0000	txc_line_enable_byte_swap Swap byte order between a pair of bytes. Legal values: [0,1].
	3	0x0000	txc_po_enable_adaptive_clk Adapt clock frequency to fifo fullness level. Legal values: [0,1].
	2	0x0001	txc_po_enable_clk_invalid_data When set, enable PCLK for all data when frame valid is asserted (JPEG continuous data). User should disable for non-JPEG mode. Legal values: [0,1].
	1	0x0001	txc_po_enable_clk_betwn_lines PCLK is runiing between assertion of line valid, during line blanking, for PO only. User should disable for JPEG continuous mode. Legal values: [0,1].

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
	0	0x0001	txc_po_enable_clk_betwn_frames PCLK is running between assertion of frame valid, during frame blanking, for PO only. Legal values: [0,1].
15524 R0x00003CA4	31:0	0x0001	txc_po_pclk1_config (R/W)
	31:8	X	Reserved
	7:5	0x0000	txc_po_pclk1_slew Slew rate of PCLK when PCLK1 is source. Legal values: [0, 7].
	4	X	Reserved
	3:0	0x0001	txc_po_pclk1_divider Integer divisor used to produce PCLK when PCLK1 is source. Legal values: [1, 15].
	For PO only		
15526 R0x00003CA6	31:0	0x0001	txc_po_pclk2_config (R/W)
	31:8	X	Reserved
	7:5	0x0000	txc_po_pclk2_slew Slew rate of PCLK when PCLK2 is source. Legal values: [0, 7].
	4	X	Reserved
	3:0	0x0001	txc_po_pclk2_divider Integer divisor used to produce PCLK when PCLK2 is source. Legal values: [1, 15].
	For PO only		
15528 R0x00003CA8	31:0	0x0001	txc_po_pclk3_config (R/W)
	31:8	X	Reserved
	7:5	0x0000	txc_po_pclk3_slew Slew rate of PCLK when PCLK3 is source. Legal values: [0, 7].
	4	X	Reserved
	3:0	0x0001	txc_po_pclk3_divider Integer divisor used to produce PCLK when PCLK3 is source. Legal values: [1, 15].
	For PO only		
15530 R0x00003CAA	31:0	0x0505	txc_timing (R/W)
	31:16	X	Reserved
	15:8	0x0005	txc_lv_trail Back porch of line timing; at least 5 for PO and 8 for MIPI. It can be 0 for continuous input data. Legal values: [5, 255].
	7:0	0x0005	txc_lv_lead Front porch of line timing; at least 5 for PO and 8 for MIPI. It can be 0 for continuous input data. Legal values: [5, 255].
15536 R0x00003CB0	31:0	0xFF00	txc_line_ccir_sof01_code (R/W)
	CCIR code, SOF LSB Legal values: [1, 65535].		

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15538 R0x00003CB2	31:0	0x00AB	txc_line_ccir_sof23_code (R/W)
	CCIR code, SOF MSB Legal values: [0, 65535].		
15540 R0x00003CB4	31:0	0xFF00	txc_line_ccir_eof01_code (R/W)
	CCIR code, EOF LSB Legal values: [0, 65535].		
15542 R0x00003CB6	31:0	0x00B6	txc_line_ccir_eof23_code (R/W)
	CCIR code, EOF MSB Legal values: [0, 65535].		
15544 R0x00003CB8	31:0	0xFF00	txc_line_ccir_sol01_code (R/W)
	CCIR code, SOL LSB Legal values: [0, 65535].		
15546 R0x00003CBA	31:0	0x0080	txc_line_ccir_sol23_code (R/W)
	CCIR code, SOL MSB Legal values: [0, 65535].		
15548 R0x00003CBC	31:0	0xFF00	txc_line_ccir_eol01_code (R/W)
	CCIR code, EOL LSB Legal values: [0, 65535].		
15550 R0x00003CBE	31:0	0x009D	txc_line_ccir_eol23_code (R/W)
	CCIR code, EOL MSB Legal values: [0, 65535].		
15552 R0x00003CC0	31:0	0x0000	txss_status (RO)
	31:3	X	Reserved
	2	RO	txss_fifo_undf 1: Txss (psync) fifo underflow error Read-only.
	1	X	Reserved
	0	RO	txss_frame_done 1: Frame done in txss Read-only.
	Fifo and frame status from Txss Read-only.		
15554 R0x00003CC2	31:0	0x0000	txss_clr_status (R/W)
	Clear txss_fifo_undf and txss_frame_done		

Table 49: TX_SS (continued)

15584 R0x00003CE0	31:0	0x0001	indirect_ram_access_ctrl (R/W)
	31:3	X	Reserved
	2	RO	indirect_busy Indirect RAM access in process This bit indicates that an indirect RAM access is happening. Read-only.
	1	0x0000	indirect_error Indirect Addressing Error This bit indicates an error in the processing of an indirect RAM access request. An error will occur if the write of the INDIRECT_ADDRESS register or a R/W of the INDIRECT_DATA register occurs while a previous indirect access is not complete. If an error occurs, the previous access will be completed and any INDIRECT_ADDRESS or INDIRECT_DATA R/W will be ignored. This status flag remains set until the firmware writes 1 to clear it.
	0	0x0001	indirect_enable_auto_inc Indirect Auto Address Increment Enable 0 - Disable Address Auto Increment 1 - Enable Address Auto Increment If auto-increment is enabled, the address is incremented at the completion of a write (which is initiated by a write to the INDIRECT_DATA register.) or at the beginning of a pre-read (which is initiated by a read of the INDIRECT_DATA register.)

Table 49: TX_SS (continued)

Register Dec (Hex)	Bits	Default	Name
15586 R0x00003CE2	31:0	0xFFFF	indirect_address (R/W)
	This register defines the indirect address It is the FW's responsibility to access valid addresses for each of the specific memories. Writing to non-exiting address is ignored, and reading from non_existing address gets invalid data (although in both cases, the address will continue to auto increment (if enabled). In order to access these RAMs, the hardware clks must be enabled. Bits [15:11] controls the RAM selection: 5'b000x_x: L2B3 Y0 RAM (5184x16) 5'b001x_x: L2B3 Y1 RAM (5184x16) 5'b010x_x: L2B3 Cb RAM (5184x16) 5'b011x_x: L2B3 Cr RAM (5184x16) 5'b1000_0: Qtable RAM (768x16). This RAM has multiple sections. The definitions are: 0x8000 - 0x803f : Qtable Y0 (non header) (big endian) 0x8040 - 0x807f : Qtable Cb0/Cr0 (non header) (big endian) 0x8080 - 0x80bf : Qtable Y1 (non header) (big endian) 0x80c0 - 0x80ff : Qtable Cb1/Cr1 (non header) (big endian) 0x8100 - 0x811f : Q Table Header Y0 (little endian) 0x8120 - 0x813f : Q Table Header Cb0/Cr0 (little endian) 0x8140 - 0x815f : Q Table Header Y1 (little endian) 0x8160 - 0x817f : Q Table Header Cb1/Cr1 (little endian) 0x8180 - 0x82ff : APP1 segment (little endian) 5'b1000_1: Reserved 5'b1001_0: JPEG Transpose RAM 0 (64x24) 5'b1001_1: JPEG Transpose RAM 1 (64x24) 5'b1010_0: JPEG Zigzag RAM 0 (64x24) 5'b1010_1: JPEG Zigzag RAM 1 (64x24) 5'b1011_0: JPEG Interleave Buffer RAM 0 (64x28 or 192x28) 5'b1011_1: JPEG Interleave Buffer RAM 1 (64x28 or 192x28) 5'b110x_x: Reserved 5'b1111_1: Reserved Reading or writing to a RAMs during image processing may corrupt the image data (except for the non JPEG_QTABLE_SEL portions of the Qtable RAM) A write to this register triggers a pre-read of the RAM address defined by the INDIRECT_ADDRESS value. See INDIRECT_DATA register for more info. A write to this register is required before reading a RAM location or sequence of locations. Legal values: [0, 65535].		
15588 R0x00003CE4	31:0	0x0000	indirect_data (R/W)
	This register holds the data written from ICB during an indirect write or the data read from RAM during an indirect read. Writing to this register triggers a write to the RAM address in INDIRECT_ADDRESS. Reading from this register returns the value from the previous pre-read. It also triggers a pre-read from the RAM address in INDIRECT_ADDRESS (If INDIRECT_ENABLE_AUTO_INCREMENT is set, INDIRECT_ADDRESS is incremented before the pre-read.) Not recommended, but if the previous indirect access operation was a write, not a pre-read, then reading this register will return the value previously written, then do a pre-read (with address auto increment, if enabled) Legal values: [0, 65535].		

Table 50: OTPM

Register Dec (Hex)	Bits	Default	Name
14336 R0x00003800	31:0	0x0000	otpm_status (RO)
	31:9	X	Reserved
	8	RO	ded_parity_failure Double error-detect parity failure, data bad Read-only. Volatile.
	7	RO	sec_used ECC single bit error correction activated. Read-only. Volatile.
	6:1	RO	ecc_check_bits Check bits produced by ECC Read-only. Volatile. Legal values: [0, 63].
	0	RO	op_done Last operation (pgm or read) complete Read-only. Volatile.
14338 R0x00003802	31:0	0x0004	otpm_control (R/W)
	31:4	X	Reserved
	3	0x0000	otpm_op_done_clr Clear "operation done" status bit Write-only.
	2:0	0x0004	otpm_mode otmp mode 000 = idle 001 = read 010 = program 1xx = power down Legal values: [0, 7].
14340 R0x00003804	31:0	0x0000	otpm_addr (R/W)
	Row address Legal values: [0, 255].		
14342 R0x00003806	31:0	0x0000	otpm_data_pgm_l (R/W)
	31:16	X	Reserved
	15:8	0x0000	otpm_data_pgm_1 otpm data to program byte 1 Legal values: [0, 255].
	7:0	0x0000	otpm_data_pgm_0 otpm data to program byte 0 Legal values: [0, 255].
14344 R0x00003808	31:0	0x0000	otpm_data_pgm_h (R/W)
	31:16	X	Reserved
	15:8	0x0000	otpm_data_pgm_3 otpm data to program byte 3 Legal values: [0, 255].
	7:0	0x0000	otpm_data_pgm_2 otpm data to program byte 2 Legal values: [0, 255].

Table 50: OTPM (continued)

Register Dec (Hex)	Bits	Default	Name
14346 R0x0000380A	31:0	0x0000	otpm_data_pgm_extra (R/W)
	31:8	X	Reserved
	7:0	0x0000	otpm_data_pgm_4 otpm data to program byte 4 (used only when ecc bypassed) Legal values: [0, 255].
14348 R0x0000380C	31:0	0x0000	otpm_data_read_l (RO)
	31:16	X	Reserved
	15:8	RO	otpm_data_read_1 otpm read data byte 1 Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	otpm_data_read_0 otpm read data byte 0 Read-only. Volatile. Legal values: [0, 255].
14350 R0x0000380E	31:0	0x0000	otpm_data_read_h (RO)
	31:16	X	Reserved
	15:8	RO	otpm_data_read_3 otpm read data byte 3 Read-only. Volatile. Legal values: [0, 255].
	7:0	RO	otpm_data_read_2 otpm read data byte 2 Read-only. Volatile. Legal values: [0, 255].
14352 R0x00003810	31:0	0x0000	otpm_data_read_extra (RO)
	31:8	X	Reserved
	7:0	RO	otpm_data_read_4 otpm read data byte 4 (used only when ecc bypassed) Read-only. Volatile. Legal values: [0, 255].
14354 R0x00003812	31:0	0x215C	otpm_cfg (R/W)
	31:15	X	Reserved
	14	0x0000	ecc_bypass Set to bypass ECC.
	13:11	0x0004	npix_idac_code define bit program current Legal values: [0, 7].
	10	0x0000	npix_en_vaa Use vaa level in column to protect bit cell from being blown
	9:8	0x0001	npix_vln_code define vln current Legal values: [0, 3].
	7:3	0x000B	npix_vdac_cmpr_code define comparison threshold voltage Legal values: [0, 31].
	2:0	0x0004	npix_vdac_bst_code define boosting level Legal values: [0, 7].

Table 50: OTPM (continued)

Register Dec (Hex)	Bits	Default	Name
14356 R0x00003814	31:0	0x0703	otpm_tcfg_01 (R/W)
	31:16	X	Reserved
	15:8	0x0007	tcfg_1 TC timing parameter 1 program mode: T6 readout mode: T24 Legal values: [0, 255].
	7:0	0x0003	tcfg_0 TC timing parameter 0 program mode: T1 readout mode: T23 Legal values: [0, 255].
14358 R0x00003816	31:0	0x0707	otpm_tcfg_23 (R/W)
	31:16	X	Reserved
	15:8	0x0007	tcfg_3 TC timing parameter 3 program mode: T8 readout mode: T26 Legal values: [0, 255].
	7:0	0x0007	tcfg_2 TC timing parameter 2 program mode: T7 readout mode: T25 Legal values: [0, 255].
14360 R0x00003818	31:0	0x4407	otpm_tcfg_4b (R/W)
	31:16	X	Reserved
	15:12	0x0004	tcfg_b1 booster timing parameter T44 Legal values: [0, 15].
	11:8	0x0004	tcfg_b0 booster timing parameter T41 Legal values: [0, 15].
	7:0	0x0007	tcfg_4 TC timing parameter 4 readout mode: T27 Legal values: [0, 255].
14362 R0x0000381A	31:0	0x0000	otpm_expr (R/W)
	31:3	X	Reserved
	2	0x0000	npix_comp_test_en Causes npix coladd to be driven to 2 during first comparison of a readout, and 3 ("invalid") during second comparison. Experimental: Firmware leaves at zero.
	1	0x0000	npix_vln_en_always_on Forces npix vln en to be high in readout mode Experimental: Firmware leaves at zero.
	0	0x0000	npix_tx_en_all Enable TX transistors in all the rows of the array. Experimental: Firmware leaves at zero.

Table 50: OTPM (continued)

Register Dec (Hex)	Bits	Default	Name
14364 R0x0000381C	31:0	0x0000	otpm_cfg2 (R/W)
	31:10	X	Reserved
	9	0x0000	npix_anatest_en Enables analog test mode. 0: Normal mode 1: Analog test mode. npix_tmux_code3 has effect.
	8	0x0000	npix_digtest_en 0: Normal mode 1: Places 0xa_ce3f on npix_frdout.
	7:6	X	Reserved
	5:0	0x0000	npix_tmux_code3 Select analog signals in test mode. 011011: Vref comparator on ATEST bus. 011010: Pixout voltage on ATEST bus. Legal values: [0, 63].

Timing Specifications

Power-up Sequence for Rev3 Silicon

Powering up the sensor is independent of voltages applied in a particular order, as shown in Figure 35. The timing requirements for other signals are shown in Table 51. It is advised that the user manually assert a hard reset upon power up.

Figure 35: Power-Up Sequence

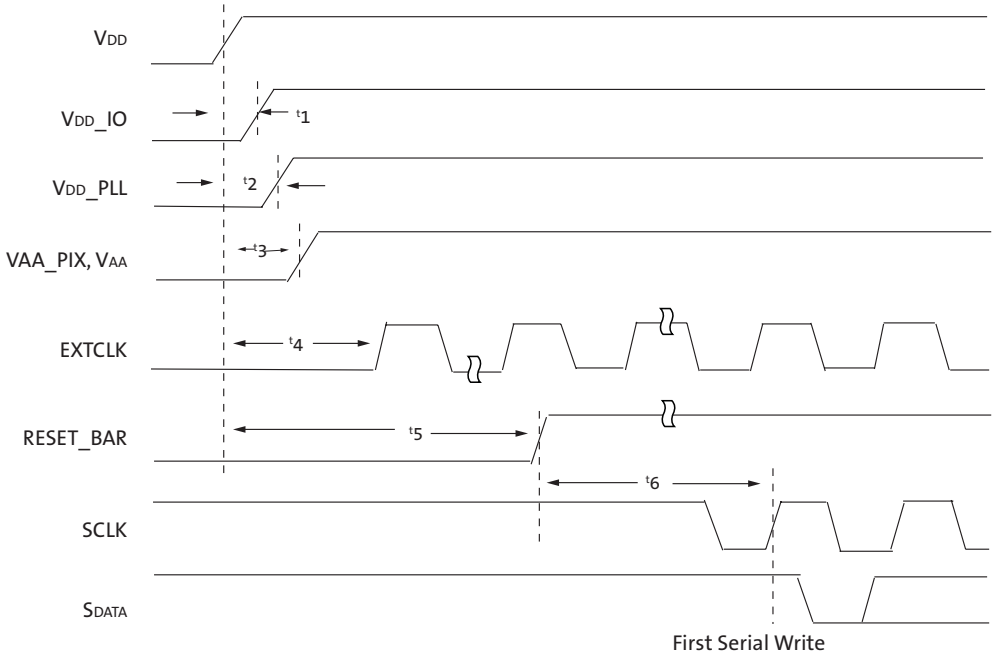


Table 51: Power-Up Signal Timing

Parameter	Symbol	Min	Typ	Max	Unit
VDD to VDD_IO	t_1	0	–	–	ms
VDD to VDD_PLL	t_2	1	–	–	
VDD to VAA_PIX	t_3	0	–	–	
VDD to EXTCLK Activation	t_4	–	500	–	
RESET_BAR activation time	t_5	70	–	–	EXTCLKs
First serial write	t_6	100	–	–	EXTCLKs

Note: All supplies are referenced to VDD.

Reset

Two types of reset are available:

- A hard reset is issued by toggling RESET_BAR.
- A soft reset is issued by writing commands through the two-wire serial interface.

Hard Reset

After hard reset, the output FIFO is configured for operation but disabled and all outputs are tri-stated. These outputs can be enabled through the two-wire serial interface. After hard reset, the output FIFO is configured for operation but disabled and all outputs are tri-stated. These outputs can be enabled through the two-wire serial interface. The hard reset signal sequence is shown in Figure 36 on page 275. Hard reset timing is shown in Table 52 on page 275.

Figure 36: Hard Reset Signal Sequence

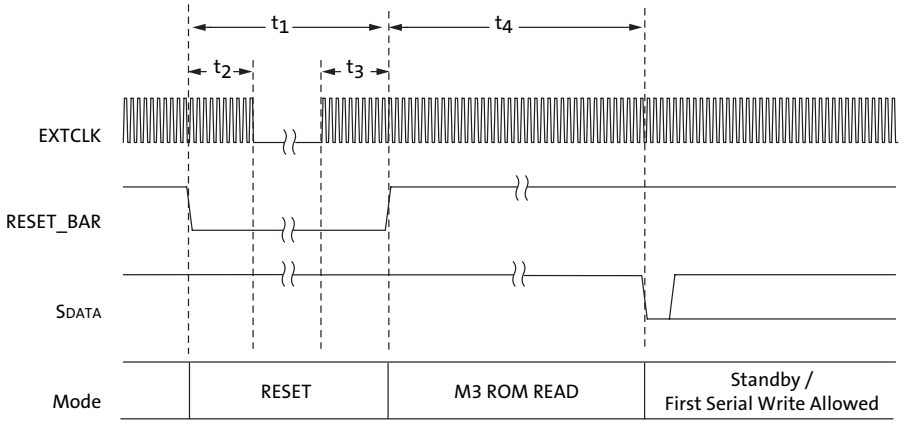


Table 52: Hard Reset Signal Timing

Parameter	Symbol	Min	Typ	Max	Unit
RESET_BAR pulse width	t_1	70	–	–	EXTCLKs
Active ECXTCLK after RESET_BAR is asserted	t_2	10	–	–	
Active EXTCLK before RESET_BAR is de-asserted	t_3	10	–	–	
First two-wire serial interface communication after RESET is HIGH	t_4	–	100	–	

Soft Reset

A soft reset sequence to the sensor has the same affect as the hard reset and can be activated writing to a register through the two-wire serial interface. The soft reset signal sequence is shown in Figure 37. Soft reset timing is shown in Table 53.

Figure 37: Soft Reset Signal Sequence

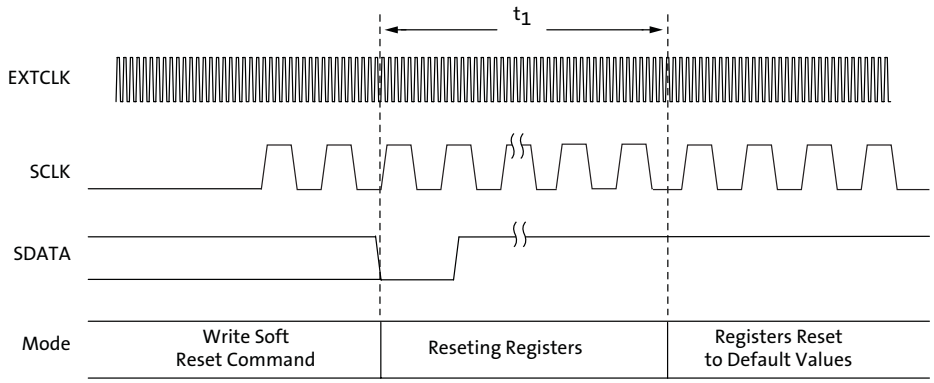


Table 53: Soft Reset Signal Timing

Parameter	Symbol	Min	Typ	Max	Unit
Active EXTCLK after soft reset command is asserted	t_1	–	100	–	EXTCLKs

Standby Modes

The MT9P111 supports three different standby modes:

- Hard standby with shutdown
- Hard standby with memory retention
- Soft standby with state retention

For all hard and soft standby modes, entry can be inhibited by programming the standby_control register.

Hard Standby with Shutdown Mode

The hard standby with shutdown mode uses STANDBY to shut down digital power (VDD) and ensure the lowest power consumption. All the two-wire serial interface settings and firmware variables including patches will be lost in this mode. Starting up from this mode is similar to performing a power-up. The host will not have to reload the PLL and clock divider settings but other sensor settings such as sensor timing, LSC, CCM, and so forth will have to be reloaded. The two-wire serial interface will be inactive and the sensor must be started up by de-asserting STANDBY.

Hard Standby With Memory Retention Mode

Hard standby with memory retention mode without the loss of variable data can also be achieved. This mode stores the variables and state of the sensor before entering standby (similar to soft standby). The power consumption is lower than that of soft standby, as internal clocks are turned off, and the two-wire serial interface will be inactive.

Since the hard standby with memory retention mode is activated by STANDBY, the en_vdd_dis_soft register needs to be programmed to indicate the selection of this mode before STANDBY is asserted. De-asserting STANDBY will cause the sensor to come out of standby mode. This also causes the sensor to resume operation from the state before the STANDBY signal was asserted. By default, asserting the STANDBY signal causes the hard standby mode described above.

The signal sequence for both modes in shown in Figure 38. The timing for both modes is shown in Table 54 on page 278.

Figure 38: Hard Standby Signal Sequence Mode

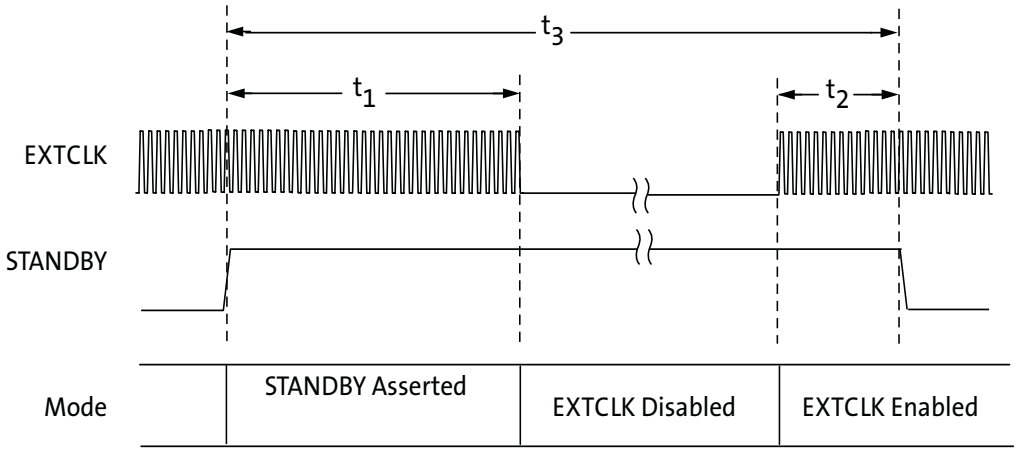


Table 54: Hard Standby Signal Timing

Parameter	Symbol	Min	Typ	Max	Unit
Standby entry complete	t_1	20	–	–	?s
Active EXTCLK before STANDBY de-asserted	t_2	10	–	–	
STANDBY pulse width	t_3	100	–	–	
STANDBY de-assertion to First Allowable Serial Write	t_4	20	–	–	

Soft Standby with State Retention

Soft standby with state retention can be enabled by register access and disables the sensor core and most of the digital logic. The two-wire serial interface is still active and the sensor can be programmed through register commands. All register settings and RAM content will be preserved. Soft standby can be performed in any sequencer state after all AE, AWB, histogram, and flicker calculations are finished and the sensor core has been disabled.

The execution of standby will take place after the completion of the current line by default. It is possible to synchronize the execution of standby with the end of frame through the standby_control register. The soft standby signal sequence is shown in Figure 39. The timing for the signals is shown in Table 55.

Figure 39: Soft Standby Signal Sequence

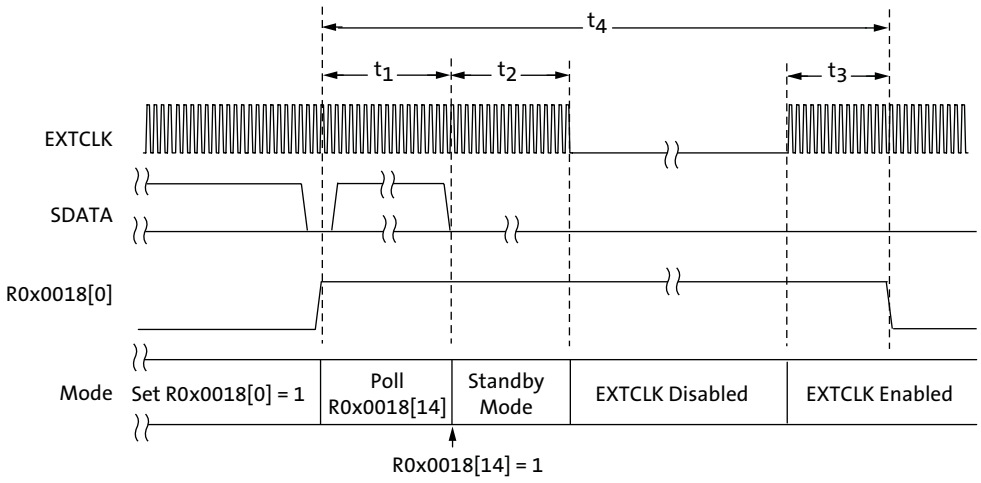


Table 55: Soft Standby Signal Timing

Parameter	Symbol	Min	Typ	Max	Unit
Standby entry complete	t_1	20	–	–	?s
Active EXTCLK before soft standby de-activates	t_2	10	–	–	
Minimum standby time	t_3	100	–	–	

Shutdown Mode

The shutdown mode is entered when the SHUTDOWN pin is asserted. All power to the MT9P111 is disabled and no state, register or patch information is retained. De-assertion of the SHUTDOWN pin will cause a full POR.

Table 56: DC Electrical Definitions and Characteristics—Parallel Mode

^{f}EXTCLK = 54 MHz; ^{f}PIXCLK = 96 MHz; V_{DD} = 1.8V; V_{DD_IO} = 1.8V or 2.8V; V_{AA} = 2.8V; VAA_PIX = 2.8V; V_{DD_PLL} = 2.8V; V_{DD_MIPI} = 2.8V; T_J = 70°C; power consumption values are reliminary estimates only

Symbol	Parameter	Condition	Min	Typ	Max	Unit
V_{DD}	Core digital voltage		1.7	1.8	1.95	V
V_{DD_IO1}	I/O digital voltage		1.7	1.8	1.95	V
V_{DD_IO2}	I/O digital voltage		2.5	2.8	3.1	V
V_{AA}	Analog voltage		2.5	2.8	3.1	V
VAA_PIX	Pixel supply voltage		2.5	2.8	3.1	V
V_{DD_PLL}	PLL supply voltage		2.5	2.8	3.1	V
I_{DD}	Digital operating current	Context A	–	91	–	mA
I_{AA}	Analog operating current	Context A	–	93	–	mA
I_{AA_PIX}	Pixel supply current	Context A	–	2	–	mA
I_{DD_PLL}	PLL supply current	Context A	–	31	–	mA
I_{DDIO_MIPI}	MIPI supply current	Context A	–	NA	–	mA
	Total supply current	Context A	–	217	–	mA
	Total power consumption	Context A	–	555	–	mW
I_{DD}	Digital operating current	Context B	–	58	–	mA
I_{AA}	Analog operating current	Context B	–	47	–	mA
IAA_PIX	Pixel supply current	Context B	–	2	–	mA
I_{DD_PLL}	PLL supply current	Context B	–	0	–	mA
I_{DDIO_MIPI}	MIPI supply current	Context B	–	NA	–	mA
	Total supply current	Context B	–	127	–	mA
	Total power consumption	Context B	–	308	–	mW
Hard standby	Total standby current when asserting the STANDBY signal	VDD Disable ON R0x0028[0] = 1 (T_J = 70°C)			20	?A
Hard standby	Total standby current when asserting the STANDBY signal	VDD Disable OFF R0x0028[0] = 0 (T_J = 70°C)		170		?A
Soft standby (clock on at 24 MHz)	Total standby current when asserting R0x0018[0] = 1	VDD Disable ON R0x0028[0] = 1		1.7		mA
Soft standby (clock OFF)	Total Standby Current when asserting R0x0018[0] = 1	VDD Disable ON R0x0028[0] = 1		180		?A
SHUTDOWN	Total standby current when asserting the SHUTDOWN signal				10	?A

Notes: 1. Context A: 30 fps preview mode.
11. Context B: 15 fps full resolution mode.

Table 57: I/O Parameters

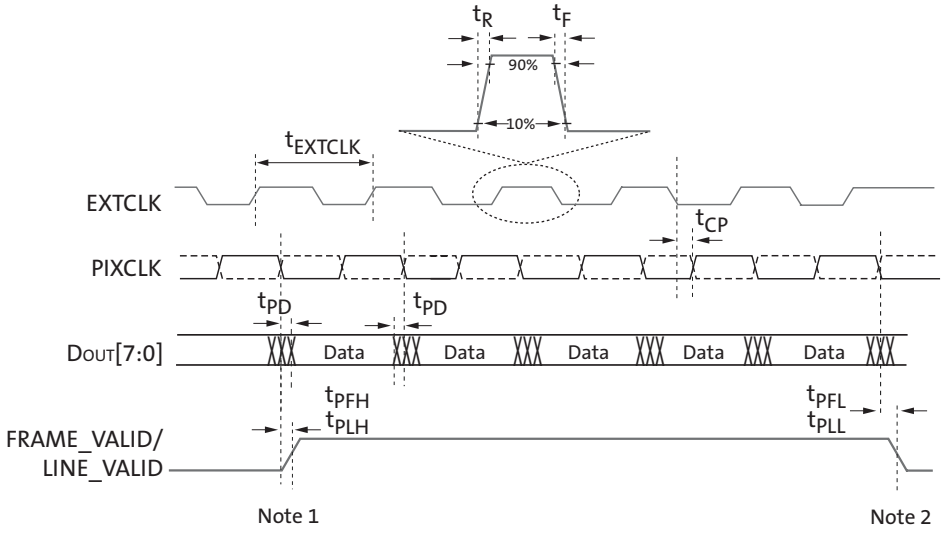
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	Input HIGH voltage	At specified I_{IN}	V_{DD_IO} – 0.4		V_{DD_IO} + 0.3	V
V_{IL}	Input LOW voltage	V_{DD_IO} = 2.8V at specified I_{IN}	–0.3		0.6	V
		V_{DD_IO} = 1.8V at specified I_{IN}	–0.3		0.4	V
I_{IN}	Input leakage current	No pull-up resistor; V_{IN} = V_{DD} or D_{GND}	–10		10	?A
V_{OH}	Output HIGH voltage	At specified I_{OH}	V_{DD_IO} – 0.4		–	V
V_{OL}	Output LOW voltage	At specified I_{OL}	–		0.4	V

Table 57: I/O Parameters

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{OH}	Output HIGH current	At minimum of 1.4V V_{OH}	–		–13	mA
		At minimum of 2.4V V_{OH}	–		–20	mA
I_{OL}	Output LOW current	At maximum of 0.4V V_{OL}	–		12	mA
		At specified V_{OL}	–		15	mA
I_{OZ}	Tri-state output leakage current				5	?A

Notes: 1. High speed parameters are not included

Figure 40: I/O Timing Diagram



Notes: 1. FV leads LV by 6 PIXCLKs.
12. FV trails LV by 6 PIXCLKs.
13. PLL disabled for t_{CP}

Table 58: I/O Timing Specifications

^{f}EXTCLK = 54 MHz; ^{f}PIXCLK = 96 MHz; V_{DD} = 1.8V; V_{DD}_IO = 1.8V or 2.8V; V_{AA} = 2.8V; VAA_PIX = 2.8V; V_{DD}_PLL = 2.8V; V_{DD}_MIPI = 2.8V; T_J = 70°C

Symbol	Parameter	Conditions	Min	Typ	Max	Unit	Note
^{f}EXTCLK	External clock frequency		8	–	54	MHz	
^{t}EXTCLK	External clock period		18.52		166.7	ns	
	EXTCLK duty cycle		45	50	55	%	
^{t}R	EXTCLK rise time				0.05 * ^{t}EXTCLK	ns	
^{t}F	EXTCLK fall time				0.05 * ^{t}EXTCLK	ns	
^{t}JITTER	EXTCLK jitter (peak-peak cycle jitter)		–	0.5	1	ns	1
^{t}CP	EXTCLK to PIXCLK propagation delay					ns	3
^{f}PIXCLK	PIXCLK frequency	Default	6	–	96	MHz	
^{t}RPIXCLK	PIXCLK rise time	C_{load} = 15pF	–	2	5	ns	
^{t}FPIXCLK	PIXCLK fall time	C_{load} = 15pF	–	2	5	ns	

Table 58: I/O Timing Specifications (continued)

^{f}EXTCLK = 54 MHz; ^{f}PIXCLK = 96 MHz; VDD = 1.8V; VDD_IO = 1.8V or 2.8V; VAA = 2.8V; VAA_PIX = 2.8V; VDD_PLL = 2.8V; VDD_MIPI = 2.8V; T_J = 70°C

Symbol	Parameter	Conditions	Min	Typ	Max	Unit	Note
^{t}PD	PIXCLK to data valid	Default	0.4 * ^{t}PIXCLK	0.5 * ^{t}PIXCLK	0.6 * ^{t}PIXCLK	ns	2
^{t}PFH	PIXCLK to FV HIGH	Default	0.4 * ^{t}PIXCLK	0.5 * ^{t}PIXCLK	0.6 * ^{t}PIXCLK	ns	2
^{t}PLH	PIXCLK to LV HIGH	Default	0.4 * ^{t}PIXCLK	0.5 * ^{t}PIXCLK	0.6 * ^{t}PIXCLK	ns	2
^{t}PFL	PIXCLK to FV LOW	Default	0.4 * ^{t}PIXCLK	0.5 * ^{t}PIXCLK	0.6 * ^{t}PIXCLK	ns	2
^{t}PLL	PIXCLK to LV LOW	Default	0.4 * ^{t}PIXCLK	0.5 * ^{t}PIXCLK	0.6 * ^{t}PIXCLK	ns	2
PIXCLK signal slew	R0x001E[10:8] = 000	VDD_IO: Typ C_{load} = 45pF		0.21		V/ns	4
	R0x001E[10:8] = 100	VDD_IO: Typ C_{load} = 45pF		0.66		V/ns	4
	R0x001E[10:8] = 111	VDD_IO: Typ C_{load} = 45pF		1.2		V/ns	4
DOUT[7:0] signal slew	R0x001E[2:0] = 000	VDD_IO: Typ C_{load} = 45pF		0.21		V/ns	4
	R0x001E[2;0] = 100	VDD_IO: Typ C_{load} = 45pF		0.66		V/ns	4
	R0x001E[2;0] = 111	VDD_IO: Typ C_{load} = 45pF		1.2		V/ns	4
CIN	Input pin capacitance		–	2.5	–	pF	
RIN	Input pin impedance		–	400	–	KΩ	

Notes:
1. Measured in terms of standard deviation.
14. From falling edge of PIXCLK.
15. PLL disabled for t_{CP}.
16. PIXCLK = 24 MHz.

Two-Wire Serial Bus Timing

Figure 41 and Table 59 on page 284 describe the timing for the two-wire serial interface.

Figure 41: Two-Wire Serial Bus Timing Parameters

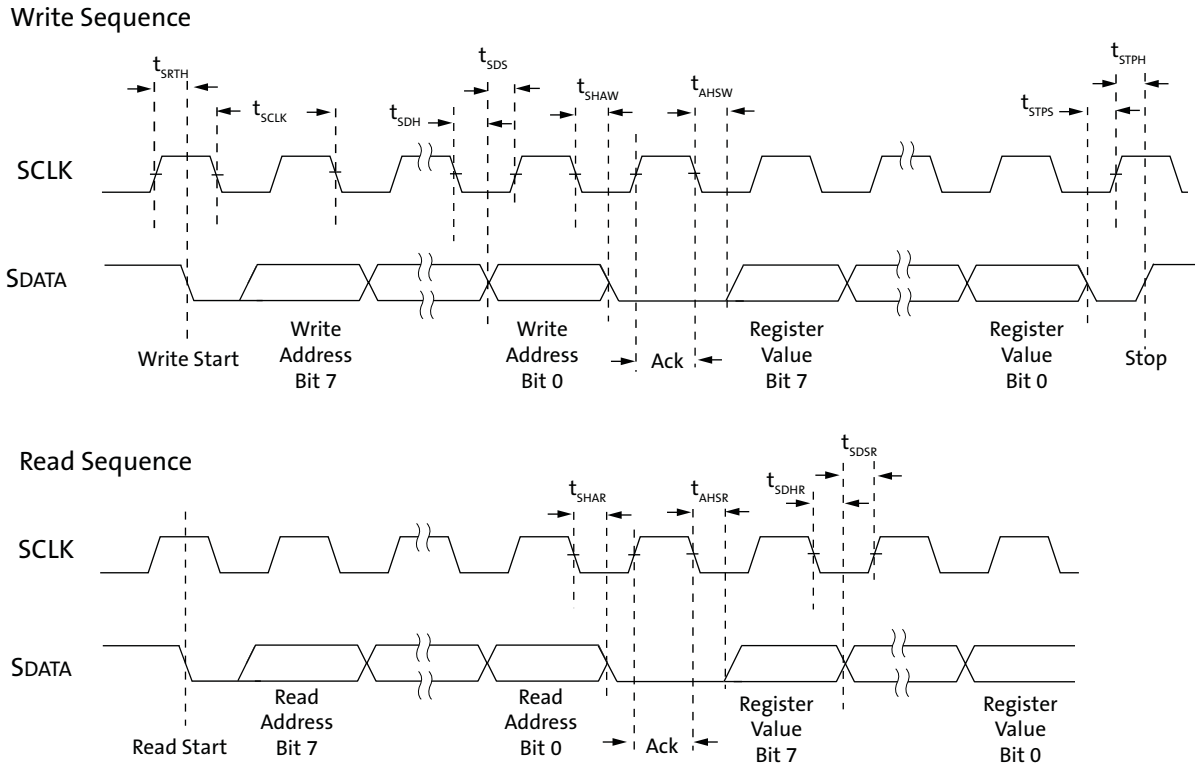


Table 59: Two-Wire Serial Bus Characteristics

^{f}EXTCLK = 8–54 MHz; V_{DD} = 1.8V; V_{DD_IO} = 1.8V or 2.8V; V_{AA} = 2.8V; V_{AA_PIX} = 2.8V; V_{DD_PLL} = 2.8V; V_{DD_PHY} = 2.8V; T_J = 70°C

Symbol	Parameter	Conditions	Min	Typ	Max	Unit	Note
^{f}SCLK	Serial interface input clock frequency		100	–	400	kHz	
^{t}SCLK	Serial interface input clock period		2.5	–	10	?s	Master clock cycle units or PLL cycles if enabled
	SCLK duty cycle		40	50	60	%	
^{t}SRTH	Start hold time	Write/Read	0.3	–	–	?s	Master clock cycle units or PLL cycles if enabled
^{t}SDH	SDATA hold	Write	0.3	–	–	?s	Master clock cycle units or PLL cycles if enabled
^{t}SDS	SDATA setup	Write	0.3	–	–	?s	Master clock cycle units or PLL cycles if enabled
^{t}SHAW	SDATA hold to ack	Write	0.15	–	0.75	?s	Master clock cycle units or PLL cycles if enabled
^{t}AHSW	Ack hold to SDATA	Write	0.3	–	–	?s	Master clock cycle units or PLL cycles if enabled
^{t}STPS	Stop setup time	Write/Read	0.3	–	–	?s	Master clock cycle units or PLL cycles if enabled
^{t}STPH	Stop hold time	Write/Read	0.6	–	–	?s	Master clock cycle units or PLL cycles if enabled
^{t}SHAR	SDATA hold to ack	Read	0.15	–	0.75	?s	Master clock cycle units or PLL cycles if enabled
^{t}AHSR	Ack hold to SDATA	Read	0.15	–	1	?s	Master clock cycle units or PLL cycles if enabled
^{t}SDHR	SDATA hold	Read	0.3	–	–	?s	Master clock cycle units or PLL cycles if enabled
^{t}SDSR	SDATA setup	Read	0.3	–	–	?s	Master clock cycle units or PLL cycles if enabled
C_{IN_SI}	Serial interface input pin capacitance		–	–	3.3	pF	
C_{LOAD_SD}	SDATA max load capacitance		–	–	30	pF	
R_{SD}	SDATA pull-up resistor		–	1.5	–	KΩ	

Caution **Table 60 shows stress ratings only, and functional operation of the device at these or any other conditions above those indicated in the product specification is not implied. Stresses above those listed may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.**

Table 60: Absolute Maximum Ratings

Symbol	Parameter	Rating		Unit
		Min	Max	
VDD_MAX	Core digital voltage	–0.3	2.4	V
VDD_IO_MAX	I/O digital voltage	–0.3	4.0	V
VAA_MAX	Analog voltage	–0.3	4.0	V
VAA_PIX_MAX	Pixel supply voltage	–0.3	4.0	V
VDD_PLL_MAX	PLL supply voltage	–0.3	4.0	V
VIH_MAX	Input HIGH voltage		VDD_IO + 0.3	V
VIL_MAX	Input LOW voltage	–0.3		V
T_OP	Operating temperature (measured at junction)	–30	75	°C
T_ST	Storage temperature	–40	85	°C

One-Time Programmable Memory Programming Sequence

Figure 42 shows the sequence of signals to be used for OTP memory programming sequence. The supply voltages and EXTCLK to be used are shown in Table 61 on page 286.

Figure 42: Sequence of Signals for OTP Memory Operation

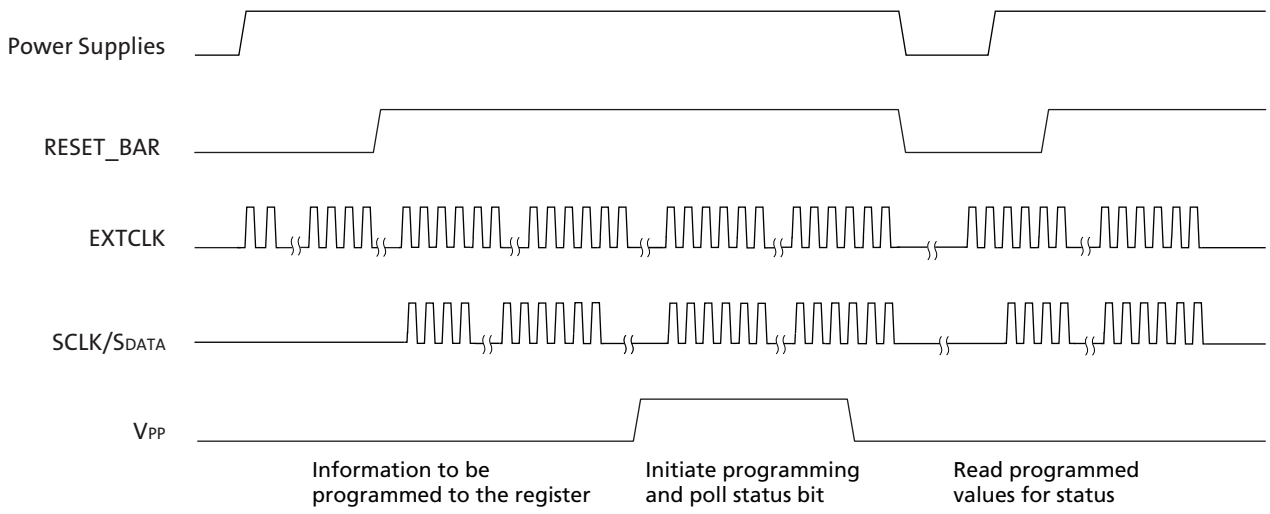


Table 61: Supplies Voltages and Clock Frequency for OTP Memory Programming

Symbol	Parameter	Min	Typ	Max	Unit
^{f}CLKIN	Input clock frequency		12		MHz
VDD	Core digital voltage		1.8		V
VDD_IO	I/O digital voltage		1.8 or 2.8		V
VAA	Analog voltage	2.6	2.8	3.1	V
VAA_PIX	Pixel supply voltage		2.8		V
VDD_PLL	PLL supply voltage		2.8		V
VPP	Programming voltage		8.5		V
VDDIO_TX	MIPI supply voltage		2.8		V

Signal States

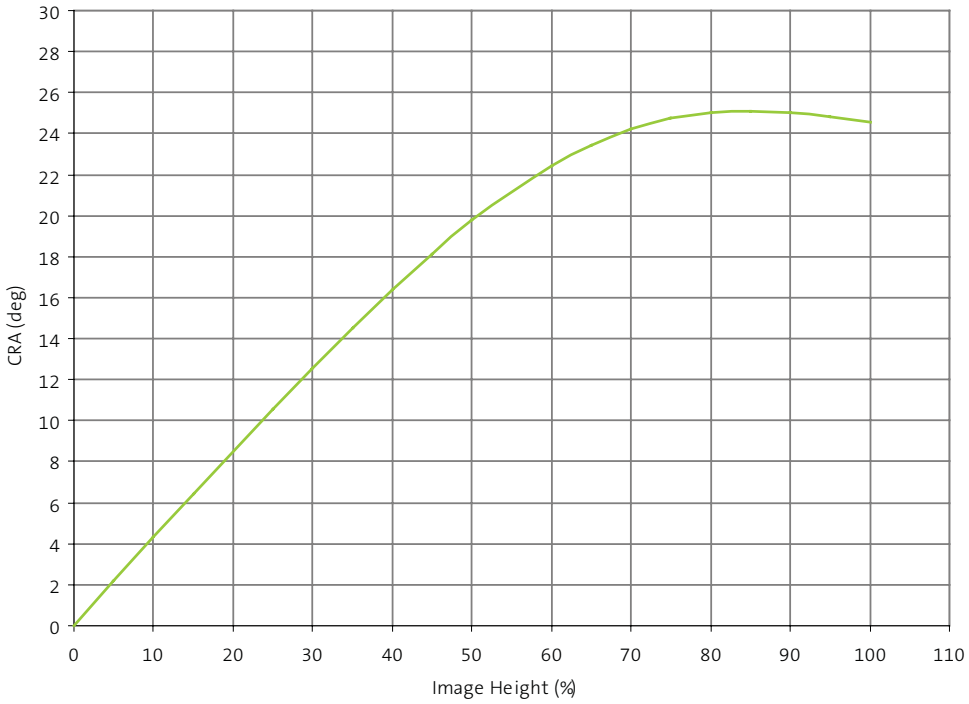
Table 62: Status of Signals During Different States

Signal	Reset	Post-Reset	Standby	Standby with SHUTDOWN	Power Down
DOUT[7:0]	High-Z	High-Z	High-Z by default (configurable via OE_BAR or two-wire serial interface reg)	High-Z by default	X
PIXCLK	High-Z	High-Z	High-Z by default (configurable)	High-Z by default	X
LV	High-Z	High-Z	High-Z by default (configurable)	High-Z by default	X
FV	High-Z	High-Z	High-Z by default (configurable)	High-Z by default	X
DOUT_N	0	0	0	0	X
DOUT_P	0	0	0	0	X
CLK_N	0	0	0	0	X
CLK_P	0	0	0	0	X
VGPIO[7:0]	High-Z	High-Z	Depending on how system uses them	Depending on how system uses them	X
SADDR	Input	Input	Input	Input	
SDATA	Input	I/O	Input	Input	
SCLK	Input	Input	Input	Input	
S_SCL	High-Z	High-Z	High-Z by default (configurable)	High-Z by default (configurable)	X
S_SDA	Input	I/O	Input	Input	

Note: X = “Don’t Care.”

Spectral Characteristics

Figure 43: Chief Ray Angle (CRA) vs. Image Height



CRA vs. Image Height Plot	Image Height		CRA
	(%)	(mm)	(deg)
	0	0	0
	5	0.113	2.19
	10	0.227	4.33
	15	0.340	6.43
	20	0.454	8.50
	25	0.567	10.55
	30	0.680	12.57
	35	0.794	14.52
	40	0.907	16.39
	45	1.021	18.15
	50	1.134	19.76
	55	1.247	21.20
	60	1.361	22.43
	65	1.474	23.44
	70	1.588	24.21
	75	1.701	24.74
	80	1.814	25.03
	85	1.928	25.11
	90	2.041	25.01
	95	2.155	24.80
	100	2.268	24.55

Revision History

Rev. A, Advance 7/11/2008

- Initial release

3080 North 1st Street, San Jose, CA 95134, prodmktg@aptina.com www.aptina.com