MT9J003 Registers

For more information, refer to the data sheet on Aptina's Web site: www.aptina.com

MT9J003 Register Reference

Table of Contents

List of Tables

Introduction

This register reference provided for engineers who are designing cameras using the MT9J003.

Register Address Space

The MT9J003 provides a 16-bit register address space accessed through a serial interface. Each register location is 8 or 16 bits in size.

The address space is divided into the five major regions shown in Table 1.

Table 1: Address Space Regions

Address Range	Description
0x0000–0x0FFF	Configuration registers (read-only and read-write dynamic registers)
0x1000–0x1FFF	Parameter limit registers (read-only static registers)
0x2000–0x2FFF	Image statistics registers (none currently defined)
0x3000–0x3FFF	Manufacturer-specific registers (read-only and read-write dynamic registers)
0x4000–0xFFFF	Reserved (undefined)

Identifying the MT9J001 and MT9J003 using Register Settings

The MT9J001 (Rev2) and MT9J003 (Rev3.1) can be identified through the sensor register settings:

MT9J001:

- Register 0x02 = 0x20
- Register 0x31FE = 0x02

MT9J003 (Original Samples):

- Register 0x02 = 0x30
- Register 0x31FE = 0x03

MT9J003 (Engineering Sample, Mass Production):

- Register 0x02 = 0x20
- Register 0x31FE = 0x32

The original sensors samples of the MT9J003 showed Register 0x02 equal to 0x30 and Register 0x31FE equal to 0x03.

Register Notation

The underlying mechanism for reading and writing registers provides byte write capability. However, it is convenient to consider some registers as multiple adjacent bytes. The MT9J003 uses 8-bit, 16-bit, and 32-bit registers, all implemented as 1 or more bytes at naturally aligned, contiguous locations in the address space.

In this document, registers are described either by address or by name. When registers are described by address, the size of the registers is explicit. For example, R0x3024 is an 8-bit register at address 0x3024, and R0x3000–1 is a 16-bit register at address 0x3000–0x3001. When registers are described by name, the size of the register is implicit. It is necessary to refer to the register table to determine that model_id is a 16-bit register.

Register Aliases

A consequence of the internal architecture of the MT9J003 is that some registers are decoded at multiple addresses. Some registers in “configuration space” are also decoded in “manufacturer-specific space.” To provide unique names for all registers, the name of the register within manufacturer-specific register space has a trailing underscore. For example, R0x0000–1 is model_id, and R0x3000–1 is model_id_. The effect of reading or writing a register through any of its aliases is identical.

Bit Fields

Some registers provide control of several different pieces of related functionality, and this makes it necessary to refer to bit fields within registers. As an example of the notation used for this, the least significant 4 bits of the model_id register are referred to as model_id[3:0] or R0x0000–1[3:0].

Bit Field Aliases

In addition to the register aliases described above, some register fields are aliased in multiple places. For example, R0x0100 (mode_select) only has one operational bit, R0x0100[0]. This bit is aliased to R0x301A–B[2]. The effect of reading or writing a bit field through any of its aliases is identical.

Byte Ordering

Registers that occupy more than 1 byte of address space are shown with the lowest address in the highest-order byte lane to match the byte-ordering on the bus. For example, the model_id register is R0x0000–1. In the register table the default value is shown as 0x2600. This means that a READ from address 0x0000 would return 0x26, and a READ from address 0x0001 would return 0x00. When reading this register as two 8-bit transfers on the serial interface, the 0x26 will appear on the serial interface first, followed by the 0x00.

Address Alignment

All register addresses are aligned naturally. Registers that occupy two bytes of address space are aligned to even 16-bit addresses, and registers that occupy four bytes of address space are aligned to 16-bit addresses that are an integer multiple of 4.

Bit Representation

For clarity, 32-bit hex numbers are shown with an underscore between the upper and lower 16 bits. For example: 0x3000_01AB.

Data Format

Most registers represent an unsigned binary value or set of bit fields. For all other register formats, the format is stated explicitly at the start of the register description. The notation for these formats is shown in Table 2.

Table 2: Data Formats

Name	Description
FIX16	Signed fixed-point, 16-bit number: two’s complement number, 8 fractional bits. Examples: 0x0100 = 1.0, 0x8000 = –128, 0xFFFF = –0.0039065

Table 2: Data Formats

Name	Description
UFIX16	Unsigned fixed-point, 16-bit number: 8.8 format. Examples: 0x0100 = 1.0, 0x280 = 2.5
FLP32	Signed floating-point, 32-bit number: IEEE 754 format. Example: 0x4280_0000 = 64.0

Register Behavior

Registers vary from “read-only,” “read/write,” and “read, write-1-to-clear.”

Double-Buffered Registers

Some sensor settings cannot be changed during frame readout. For example, changing R0x0344–5 (x_addr_start) partway through frame readout would result in inconsistent row lengths within a frame. To avoid this, the MT9J003 double-buffers many registers by implementing a “pending” and a “live” version. READs and WRITEs access the pending register; the live register controls the sensor operation.

The value in the pending register is transferred to a live register at a fixed point in the frame timing, called frame start. Frame start is defined as the point at which the first dark row is read out internally to the sensor. In the register tables the “Sync’d” column shows which registers or register fields are double-buffered in this way.

Using grouped_parameter_hold

The register grouped_parameter_hold (R0x0104) can be used to inhibit transfers from the pending to the live registers. When the MT9J003 is in streaming mode, write “1” to this register before making changes to any group of registers where a set of changes is required to take effect simultaneously. When this register is set to “0,” all transfers from pending to live registers take place on the next frame start.

An example of the consequences of failing to set this bit follows:

> An external auto exposure algorithm might want to change both gain and integration time between two frames. If the next frame starts between these operations, it will have the new gain, but not the new integration time, which would return a frame with the wrong brightness that might lead to a feedback loop with the AE algorithm resulting in flickering.

Bad Frames

A bad frame is a frame where all rows do not have the same integration time or where offsets to the pixel values have changed during the frame.

Many changes to the sensor register settings can cause a bad frame. For example, when line_length_pck (R0x0342–3) is changed, the new register value does not affect sensor behavior until the next frame start. However, the frame that would be read out at that frame start will have been integrated using the old row width, so reading it out using the new row width would result in a frame with an incorrect integration time.

By default, bad frames are not masked. If the masked bad frame option is enabled, both LV and FV are inhibited for these frames so that the vertical blanking time between frames is extended by the frame time.

In the register tables, the “Bad Frame” column shows where changing a register or register field will cause a bad frame. This notation is used:

N—No. Changing the register value will not produce a bad frame.
Y—Yes. Changing the register value might produce a bad frame.
YM—Yes; but the bad frame will be masked out when mask_corrupted_frames (R0x0105) is set to “1.”

Changes to Integration Time

If the integration time is changed while FV is asserted for frame *n*, the first frame output using the new integration time is frame *(n + 2)*. The sequence is as follows:

1. During frame *n*, the new integration time is held in the pending register.
2. At the start of frame (*n + 1*), the new integration time is transferred to the live register. Integration for each row of frame (*n + 1*) has been completed using the old integration time.
3. The earliest time that a row can start integrating using the new integration time is immediately after that row has been read for frame (*n + 1*). The actual time that rows start integrating using the new integration time is dependent upon the new value of the integration time.
4. When frame (*n + 2*) is read out, it will have been integrated using the new integration time.

If the integration time is changed on successive frames, each value written will be applied for a single frame; the latency between writing a value and it affecting the frame readout remains at two frames.

Changes to Gain Settings

Usually, when the gain settings are changed, the gain is updated on the next frame start. When the integration time and the gain are changed at the same time, the gain update is held off by one frame so that the first frame output with the new integration time also has the new gain applied. In this case, a new gain should not be set during the extra frame delay. There is an option to turn off the extra frame delay by setting reset_register[14] bit.

Embedded Data

The current values of implemented registers in the address range 0x0000–0x0FFF can be generated as part of the pixel data. This embedded data is enabled by default when the serial pixel data interface is enabled.

The current value of a register is the value that was used for the image data in that frame. In general, this is the live value of the register. The exceptions are:

- The integration time is delayed by one further frame, so that the value corresponds to the integration time used for the image data in the frame. See “Changes to Integration Time” on page 8.
- The PLL timing registers are not double-buffered because the result of changing them in streaming mode is undefined. Therefore, the pending and live values for these registers are equivalent.

Register List and Default Values

SMIA Configuration Register List

Table 1: SMIA Configuration
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R0 (R0x0000)	model_id	dddd dddd dddd dddd	11265 (0x2C01)
R2 (R0x0002)	revision_number	dddd dddd	32 (0x20)
R3 (R0x0003)	manufacturer_id	???? ????	6 (0x06)
R4 (R0x0004)	smia_version	???? ????	10 (0x0A)
R5 (R0x0005)	frame_count	???? ????	255 (0xFF)
R6 (R0x0006)	pixel_order	0000 00??	0 (0x00)
R8 (R0x0008)	data_pedestal	0000 dddd dddd dddd	40 (0x0028)
R64 (R0x0040)	frame_format_model_type	???? ????	1 (0x01)
R65 (R0x0041)	frame_format_model_subtype	???? ????	18 (0x12)
R66 (R0x0042)	frame_format_descriptor_0	???? ???? ???? ????	24144 (0x5E50)
R68 (R0x0044)	frame_format_descriptor_1	???? ???? ???? ????	4098 (0x1002)
R70 (R0x0046)	frame_format_descriptor_2	???? ???? ???? ????	23228 (0x5ABC)
R72 (R0x0048)	frame_format_descriptor_3	???? ???? ???? ????	0 (0x0000)
R74 (R0x004A)	frame_format_descriptor_4	???? ???? ???? ????	0 (0x0000)
R76 (R0x004C)	frame_format_descriptor_5	???? ???? ???? ????	0 (0x0000)
R78 (R0x004E)	frame_format_descriptor_6	???? ???? ???? ????	0 (0x0000)
R80 (R0x0050)	frame_format_descriptor_7	???? ???? ???? ????	0 (0x0000)
R82 (R0x0052)	frame_format_descriptor_8	???? ???? ???? ????	0 (0x0000)
R84 (R0x0054)	frame_format_descriptor_9	???? ???? ???? ????	0 (0x0000)
R86 (R0x0056)	frame_format_descriptor_10	???? ???? ???? ????	0 (0x0000)
R88 (R0x0058)	frame_format_descriptor_11	???? ???? ???? ????	0 (0x0000)

Table 1: SMIA Configuration (continued)
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R90 (R0x005A)	frame_format_descriptor_12	???? ???? ???? ????	0 (0x0000)
R92 (R0x005C)	frame_format_descriptor_13	???? ???? ???? ????	0 (0x0000)
R94 (R0x005E)	frame_format_descriptor_14	???? ???? ???? ????	0 (0x0000)
R128 (R0x0080)	analogue_gain_capability	???? ???? ???? ????	1 (0x0001)
R132 (R0x0084)	analogue_gain_code_min	???? ???? ???? ????	8 (0x0008)
R134 (R0x0086)	analogue_gain_code_max	???? ???? ???? ????	255 (0x00FF)
R136 (R0x0088)	analogue_gain_code_step	???? ???? ???? ????	1 (0x0001)
R138 (R0x008A)	analogue_gain_type	???? ???? ???? ????	0 (0x0000)
R140 (R0x008C)	analogue_gain_m0	???? ???? ???? ????	1 (0x0001)
R142 (R0x008E)	analogue_gain_c0	???? ???? ???? ????	0 (0x0000)
R144 (R0x0090)	analogue_gain_m1	???? ???? ???? ????	0 (0x0000)
R146 (R0x0092)	analogue_gain_c1	???? ???? ???? ????	8 (0x0008)
R192 (R0x00C0)	data_format_model_type	???? ????	1 (0x01)
R193 (R0x00C1)	data_format_model_subtype	???? ????	5 (0x05)
R194 (R0x00C2)	data_format_descriptor_0	???? ???? ???? ????	2570 (0x0A0A)
R196 (R0x00C4)	data_format_descriptor_1	???? ???? ???? ????	2056 (0x0808)
R198 (R0x00C6)	data_format_descriptor_2	???? ???? ???? ????	2568 (0x0A08)
R200 (R0x00C8)	data_format_descriptor_3	???? ???? ???? ????	3084 (0x0C0C)
R202 (R0x00CA)	data_format_descriptor_4	???? ???? ???? ????	3080 (0x0C08)
R204 (R0x00CC)	data_format_descriptor_5	???? ???? ???? ????	0 (0x0000)
R206 (R0x00CE)	data_format_descriptor_6	???? ???? ???? ????	0 (0x0000)
R256 (R0x0100)	mode_select	0000 000d	0 (0x00)
R257 (R0x0101)	image_orientation	0000 00dd	0 (0x00)
R259 (R0x0103)	software_reset	0000 000d	0 (0x00)

Table 1: SMIA Configuration (continued)

1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R260 (R0x0104)	grouped_parameter_hold	0000 000d	0 (0x00)
R261 (R0x0105)	mask_corrupted_frames	0000 000d	0 (0x00)
R272 (R0x0110)	ccp2_channel_identifier	0000 0ddd	0 (0x00)
R273 (R0x0111)	ccp2_signalling_mode	0000 000d	1 (0x01)
R274 (R0x0112)	ccp_data_format	dddd dddd dddd dddd	3084 (0x0C0C)
R288 (R0x0120)	gain_mode	0000 000d	0 (0x00)
R512 (R0x0200)	fine_integration_time	dddd dddd dddd ddd0	1010 (0x03F2)
R514 (R0x0202)	coarse_integration_time	dddd dddd dddd dddd	16 (0x0010)
R516 (R0x0204)	analogue_gain_code_global	0000 0000 dddd dddd	8 (0x0008)
R518 (R0x0206)	analogue_gain_code_greenr	0000 0000 dddd dddd	8 (0x0008)
R520 (R0x0208)	analogue_gain_code_red	0000 0000 dddd dddd	8 (0x0008)
R522 (R0x020A)	analogue_gain_code_blue	0000 0000 dddd dddd	8 (0x0008)
R524 (R0x020C)	analogue_gain_code_greenb	0000 0000 dddd dddd	8 (0x0008)
R526 (R0x020E)	digital_gain_greenr	0000 0ddd 0000 0000	256 (0x0100)
R528 (R0x0210)	digital_gain_red	0000 0ddd 0000 0000	256 (0x0100)
R530 (R0x0212)	digital_gain_blue	0000 0ddd 0000 0000	256 (0x0100)
R532 (R0x0214)	digital_gain_greenb	0000 0ddd 0000 0000	256 (0x0100)
R768 (R0x0300)	vt_pix_clk_div	0000 0000 0000 dddd	3 (0x0003)
R770 (R0x0302)	vt_sys_clk_div	0000 0000 000d dddd	1 (0x0001)
R772 (R0x0304)	pre_pll_clk_div	0000 0000 00dd dddd	2 (0x0002)
R774 (R0x0306)	pll_multiplier	0000 0000 dddd dddd	48 (0x0030)
R776 (R0x0308)	op_pix_clk_div	0000 0000 000d dddd	12 (0x000C)
R778 (R0x030A)	op_sys_clk_div	0000 0000 000d dddd	1 (0x0001)
R832 (R0x0340)	frame_length_lines	dddd dddd dddd dddd	2891 (0x0B4B)

Table 1: SMIA Configuration (continued)

1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R834 (R0x0342)	line_length_pck	dddd dddd dddd ddd0	7440 (0x1D10)
R836 (R0x0344)	x_addr_start	0000 dddd dddd dddd	112 (0x0070)
R838 (R0x0346)	y_addr_start	0000 dddd dddd dddd	8 (0x0008)
R840 (R0x0348)	x_addr_end	0000 dddd dddd dddd	3775 (0x0EBF)
R842 (R0x034A)	y_addr_end	0000 dddd dddd dddd	2755 (0x0AC3)
R844 (R0x034C)	x_output_size	0000 dddd dddd ddd0	3664 (0x0E50)
R846 (R0x034E)	y_output_size	0000 dddd dddd ddd0	2748 (0x0ABC)
R896 (R0x0380)	x_even_inc	0000 0000 0000 000?	1 (0x0001)
R898 (R0x0382)	x_odd_inc	0000 0000 0000 dddd	1 (0x0001)
R900 (R0x0384)	y_even_inc	0000 0000 0000 000?	1 (0x0001)
R902 (R0x0386)	y_odd_inc	0000 0000 00dd dddd	1 (0x0001)
R1024 (R0x0400)	scaling_mode	0000 0000 0000 00dd	0 (0x0000)
R1026 (R0x0402)	spatial_sampling	0000 0000 0000 000d	0 (0x0000)
R1028 (R0x0404)	scale_m	0000 0000 dddd dddd	16 (0x0010)
R1030 (R0x0406)	scale_n	0000 0000 ???? ????	16 (0x0010)
R1280 (R0x0500)	compression_mode	0000 0000 0000 000?	1 (0x0001)
R1536 (R0x0600)	test_pattern_mode	0000 000d 0000 0ddd	0 (0x0000)
R1538 (R0x0602)	test_data_red	0000 dddd dddd dddd	0 (0x0000)
R1540 (R0x0604)	test_data_greenr	0000 dddd dddd dddd	0 (0x0000)
R1542 (R0x0606)	test_data_blue	0000 dddd dddd dddd	0 (0x0000)
R1544 (R0x0608)	test_data_greenb	0000 dddd dddd dddd	0 (0x0000)
R1546 (R0x060A)	horizontal_cursor_width	0000 dddd dddd dddd	0 (0x0000)
R1548 (R0x060C)	horizontal_cursor_position	0000 dddd dddd dddd	0 (0x0000)
R1550 (R0x060E)	vertical_cursor_width	0000 dddd dddd dddd	0 (0x0000)

Table 1: SMIA Configuration (continued)
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R1552 (R0x0610)	vertical_cursor_position	0000 dddd dddd dddd	0 (0x0000)

SMIA Parameter Limits Register List

Table 2: SMIA Parameter Limits
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R4096 (R0x1000)	integration_time_capability	0000 0000 0000 000?	1 (0x0001)
R4100 (R0x1004)	coarse_integration_time_min	dddd dddd dddd dddd	0 (0x0000)
R4102 (R0x1006)	coarse_integration_time_max_margin	dddd dddd dddd dddd	1 (0x0001)
R4104 (R0x1008)	fine_integration_time_min	dddd dddd dddd dddd	1010 (0x03F2)
R4106 (R0x100A)	fine_integration_time_max_margin	dddd dddd dddd dddd	638 (0x027E)
R4224 (R0x1080)	digital_gain_capability	0000 0000 0000 000?	1 (0x0001)
R4228 (R0x1084)	digital_gain_min	???? ???? ???? ????	256 (0x0100)
R4230 (R0x1086)	digital_gain_max	???? ???? ???? ????	1792 (0x0700)
R4232 (R0x1088)	digital_gain_step_size	???? ???? ???? ????	256 (0x0100)
R4352 (R0x1100)	min_ext_clk_freq_mhz_1	???? ???? ???? ????	16384 (0x4000)
R4354 (R0x1102)	min_ext_clk_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4356 (R0x1104)	max_ext_clk_freq_mhz_1	???? ???? ???? ????	17024 (0x4280)
R4358 (R0x1106)	max_ext_clk_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4360 (R0x1108)	min_pre_pll_clk_div	???? ???? ???? ????	1 (0x0001)
R4362 (R0x110A)	max_pre_pll_clk_div	???? ???? ???? ????	64 (0x0040)
R4364 (R0x110C)	min_pll_ip_freq_mhz_1	???? ???? ???? ????	16512 (0x4080)
R4366 (R0x110E)	min_pll_ip_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4368 (R0x1110)	max_pll_ip_freq_mhz_1	???? ???? ???? ????	16832 (0x41C0)

Table 2: SMIA Parameter Limits (continued)

1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R4370 (R0x1112)	max_pll_ip_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4372 (R0x1114)	min_pll_multiplier	???? ???? ???? ????	32 (0x0020)
R4374 (R0x1116)	max_pll_multiplier	???? ???? ???? ????	128 (0x0080)
R4376 (R0x1118)	min_pll_op_freq_mhz_1	???? ???? ???? ????	17344 (0x43C0)
R4378 (R0x111A)	min_pll_op_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4380 (R0x111C)	max_pll_op_freq_mhz_1	???? ???? ???? ????	17472 (0x4440)
R4382 (R0x111E)	max_pll_op_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4384 (R0x1120)	min_vt_sys_clk_div	???? ???? ???? ????	1 (0x0001)
R4386 (R0x1122)	max_vt_sys_clk_div	???? ???? ???? ????	8 (0x0008)
R4388 (R0x1124)	min_vt_sys_clk_freq_mhz_1	???? ???? ???? ????	16832 (0x41C0)
R4390 (R0x1126)	min_vt_sys_clk_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4392 (R0x1128)	max_vt_sys_clk_freq_mhz_1	???? ???? ???? ????	17472 (0x4440)
R4394 (R0x112A)	max_vt_sys_clk_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4396 (R0x112C)	min_vt_pix_clk_freq_mhz_1	???? ???? ???? ????	16409 (0x4019)
R4398 (R0x112E)	min_vt_pix_clk_freq_mhz_2	???? ???? ???? ????	39322 (0x999A)
R4400 (R0x1130)	max_vt_pix_clk_freq_mhz_1	???? ???? ???? ????	17088 (0x42C0)
R4402 (R0x1132)	max_vt_pix_clk_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4404 (R0x1134)	min_vt_pix_clk_div	???? ???? ???? ????	2 (0x0002)
R4406 (R0x1136)	max_vt_pix_clk_div	???? ???? ???? ????	8 (0x0008)
R4416 (R0x1140)	min_frame_length_lines	dddd dddd dddd dddd	81 (0x0051)
R4418 (R0x1142)	max_frame_length_lines	dddd dddd dddd dddd	65535 (0xFFFF)
R4420 (R0x1144)	min_line_length_pck	dddd dddd dddd dddd	1648 (0x0670)
R4422 (R0x1146)	max_line_length_pck	dddd dddd dddd dddd	65534 (0xFFFE)
R4424 (R0x1148)	min_line_blanking_pck	dddd dddd dddd dddd	1134 (0x046E)

Table 2: **SMIA Parameter Limits (continued)**
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R4426 (R0x114A)	min_frame_blanking_lines	dddd dddd dddd dddd	79 (0x004F)
R4448 (R0x1160)	min_op_sys_clk_div	???? ???? ???? ????	1 (0x0001)
R4450 (R0x1162)	max_op_sys_clk_div	???? ???? ???? ????	1 (0x0001)
R4452 (R0x1164)	min_op_sys_clk_freq_mhz_1	???? ???? ???? ????	16793 (0x4199)
R4454 (R0x1166)	min_op_sys_clk_freq_mhz_2	???? ???? ???? ????	39322 (0x999A)
R4456 (R0x1168)	max_op_sys_clk_freq_mhz_1	???? ???? ???? ????	17472 (0x4440)
R4458 (R0x116A)	max_op_sys_clk_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4460 (R0x116C)	min_op_pix_clk_div	???? ???? ???? ????	8 (0x0008)
R4462 (R0x116E)	max_op_pix_clk_div	???? ???? ???? ????	12 (0x000C)
R4464 (R0x1170)	min_op_pix_clk_freq_mhz_1	???? ???? ???? ????	16409 (0x4019)
R4466 (R0x1172)	min_op_pix_clk_freq_mhz_2	???? ???? ???? ????	39322 (0x999A)
R4468 (R0x1174)	max_op_pix_clk_freq_mhz_1	???? ???? ???? ????	17088 (0x42C0)
R4470 (R0x1176)	max_op_pix_clk_freq_mhz_2	???? ???? ???? ????	0 (0x0000)
R4480 (R0x1180)	x_addr_min	???? ???? ???? ????	24 (0x0018)
R4482 (R0x1182)	y_addr_min	???? ???? ???? ????	0 (0x0000)
R4484 (R0x1184)	x_addr_max	???? ???? ???? ????	3879 (0x0F27)
R4486 (R0x1186)	y_addr_max	???? ???? ???? ????	2763 (0x0ACB)
R4544 (R0x11C0)	min_even_inc	???? ???? ???? ????	1 (0x0001)
R4546 (R0x11C2)	max_even_inc	???? ???? ???? ????	1 (0x0001)
R4548 (R0x11C4)	min_odd_inc	???? ???? ???? ????	1 (0x0001)
R4550 (R0x11C6)	max_odd_inc	???? ???? ???? ????	63 (0x003F)
R4608 (R0x1200)	scaling_capability	0000 0000 0000 00??	2 (0x0002)
R4612 (R0x1204)	scaler_m_min	???? ???? ???? ????	16 (0x0010)
R4614 (R0x1206)	scaler_m_max	???? ???? ???? ????	128 (0x0080)

Table 2: **SMIA Parameter Limits (continued)**
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R4616 (R0x1208)	scaler_n_min	???? ???? ???? ????	16 (0x0010)
R4618 (R0x120A)	scaler_n_max	???? ???? ???? ????	16 (0x0010)
R4864 (R0x1300)	compression_capability	0000 0000 0000 000?	1 (0x0001)
R5120 (R0x1400)	matrix_element_redinred	dddd dddd dddd dddd	578 (0x0242)
R5122 (R0x1402)	matrix_element_greeninred	dddd dddd dddd dddd	65280 (0xFF00)
R5124 (R0x1404)	matrix_element_blueinred	dddd dddd dddd dddd	65470 (0xFFBE)
R5126 (R0x1406)	matrix_element_redingreen	dddd dddd dddd dddd	65460 (0xFFB4)
R5128 (R0x1408)	matrix_element_greeningreen	dddd dddd dddd dddd	512 (0x0200)
R5130 (R0x140A)	matrix_element_blueingreen	dddd dddd dddd dddd	65357 (0xFF4D)
R5132 (R0x140C)	matrix_element_redinblue	dddd dddd dddd dddd	65521 (0xFFF1)
R5134 (R0x140E)	matrix_element_greeninblue	dddd dddd dddd dddd	65332 (0xFF34)
R5136 (R0x1410)	matrix_element_blueinblue	dddd dddd dddd dddd	476 (0x01DC)

Manufacturer-Specific Register List

Table 3: **Manufacturer-Specific**
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R12288 (R0x3000)	model_id_	dddd dddd dddd dddd	11265 (0x2C01)
R12290 (R0x3002)	y_addr_start_	0000 dddd dddd dddd	8 (0x0008)
R12292 (R0x3004)	x_addr_start_	0000 dddd dddd dddd	112 (0x0070)
R12294 (R0x3006)	y_addr_end_	0000 dddd dddd dddd	2755 (0x0AC3)
R12296 (R0x3008)	x_addr_end_	0000 dddd dddd dddd	3775 (0x0EBF)
R12298 (R0x300A)	frame_length_lines_	dddd dddd dddd dddd	2891 (0x0B4B)
R12300 (R0x300C)	line_length_pck_	dddd dddd dddd ddd0	7440 (0x1D10)
R12304 (R0x3010)	fine_correction	0ddd dddd dddd dddd	156 (0x009C)

Table 3: Manufacturer-Specific (continued)

1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R12306 (R0x3012)	coarse_integration_time_	dddd dddd dddd dddd	16 (0x0010)
R12308 (R0x3014)	fine_integration_time_	dddd dddd dddd ddd0	1010 (0x03F2)
R12310 (R0x3016)	row_speed	0000 0ddd 0ddd 0ddd	289 (0x0121)
R12312 (R0x3018)	extra_delay	dddd dddd dddd ddd0	0 (0x0000)
R12314 (R0x301A)	reset_register	dd0d 0ddd dddd dddd	24 (0x0018)
R12316 (R0x301C)	mode_select_	0000 000d	0 (0x00)
R12317 (R0x301D)	image_orientation_	0000 00dd	0 (0x00)
R12318 (R0x301E)	data_pedestal_	0000 dddd dddd dddd	40 (0x0028)
R12321 (R0x3021)	software_reset_	0000 000d	0 (0x00)
R12322 (R0x3022)	grouped_parameter_hold_	0000 000d	0 (0x00)
R12323 (R0x3023)	mask_corrupted_frames_	0000 000d	0 (0x00)
R12324 (R0x3024)	pixel_order_	0000 00??	0 (0x00)
R12326 (R0x3026)	gpi_status	dddd dddd dddd ????	65535 (0xFFFF)
R12328 (R0x3028)	analogue_gain_code_global_	0000 0000 dddd dddd	8 (0x0008)
R12330 (R0x302A)	analogue_gain_code_greenr_	0000 0000 dddd dddd	8 (0x0008)
R12332 (R0x302C)	analogue_gain_code_red_	0000 0000 dddd dddd	8 (0x0008)
R12334 (R0x302E)	analogue_gain_code_blue_	0000 0000 dddd dddd	8 (0x0008)
R12336 (R0x3030)	analogue_gain_code_greenb_	0000 0000 dddd dddd	8 (0x0008)
R12338 (R0x3032)	digital_gain_greenr_	0000 0ddd 0000 0000	256 (0x0100)
R12340 (R0x3034)	digital_gain_red_	0000 0ddd 0000 0000	256 (0x0100)
R12342 (R0x3036)	digital_gain_blue_	0000 0ddd 0000 0000	256 (0x0100)
R12344 (R0x3038)	digital_gain_greenb_	0000 0ddd 0000 0000	256 (0x0100)
R12346 (R0x303A)	smia_version_	???? ????	10 (0x0A)
R12347 (R0x303B)	frame_count_	???? ????	255 (0xFF)

Table 3: Manufacturer-Specific (continued)
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R12348 (R0x303C)	frame_status	0000 0000 0000 00??	0 (0x0000)
R12352 (R0x3040)	read_mode	dddd dddd dddd dddd	65 (0x0041)
R12358 (R0x3046)	flash	??dd dddd dddd 0000	1536 (0x0600)
R12360 (R0x3048)	flash_count	dddd dddd dddd dddd	8 (0x0008)
R12374 (R0x3056)	green1_gain	0ddd dddd dddd dddd	4160 (0x1040)
R12376 (R0x3058)	blue_gain	0ddd dddd dddd dddd	4160 (0x1040)
R12378 (R0x305A)	red_gain	0ddd dddd dddd dddd	4160 (0x1040)
R12380 (R0x305C)	green2_gain	0ddd dddd dddd dddd	4160 (0x1040)
R12382 (R0x305E)	global_gain	0ddd dddd dddd dddd	4160 (0x1040)
R12394 (R0x306A)	datapath_status	0000 0000 00?d dddd	0 (0x0000)
R12398 (R0x306E)	datapath_select	dddd dddd ?ddd dddd	36992 (0x9080)
R12400 (R0x3070)	test_pattern_mode_	0000 000d 0000 0ddd	0 (0x0000)
R12402 (R0x3072)	test_data_red_	0000 dddd dddd dddd	0 (0x0000)
R12404 (R0x3074)	test_data_greenr_	0000 dddd dddd dddd	0 (0x0000)
R12406 (R0x3076)	test_data_blue_	0000 dddd dddd dddd	0 (0x0000)
R12408 (R0x3078)	test_data_greenb_	0000 dddd dddd dddd	0 (0x0000)
R12410 (R0x307A)	test_raw_mode	0000 0000 0000 00dd	0 (0x0000)
R12448 (R0x30A0)	x_even_inc_	0000 0000 0000 000?	1 (0x0001)
R12450 (R0x30A2)	x_odd_inc_	0000 0000 0000 dddd	1 (0x0001)
R12452 (R0x30A4)	y_even_inc_	0000 0000 0000 000?	1 (0x0001)
R12454 (R0x30A6)	y_odd_inc_	0000 0000 00dd dddd	1 (0x0001)
R12638 (R0x315E)	global_seq_trigger	dddd 0d?? 00dd 0ddd	0 (0x0000)
R12640 (R0x3160)	global_rst_end	dddd dddd dddd dddd	152 (0x0098)
R12642 (R0x3162)	global_shutter_start	dddd dddd dddd dddd	168 (0x00A8)

Table 3: Manufacturer-Specific (continued)

1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R12644 (R0x3164)	global_shutter_start2	dddd dddd dddd dddd	0 (0x0000)
R12646 (R0x3166)	global_read_start	dddd dddd dddd dddd	184 (0x00B8)
R12648 (R0x3168)	global_read_start2	dddd dddd dddd dddd	0 (0x0000)
R12704 (R0x31A0)	serial_format_descriptor_0	???? ???? ???? ????	257 (0x0101)
R12706 (R0x31A2)	serial_format_descriptor_1	???? ???? ???? ????	513 (0x0201)
R12708 (R0x31A4)	serial_format_descriptor_2	???? ???? ???? ????	514 (0x0202)
R12710 (R0x31A6)	serial_format_descriptor_3	???? ???? ???? ????	769 (0x0301)
R12712 (R0x31A8)	serial_format_descriptor_4	???? ???? ???? ????	770 (0x0302)
R12714 (R0x31AA)	serial_format_descriptor_5	???? ???? ???? ????	772 (0x0304)
R12716 (R0x31AC)	serial_format_descriptor_6	???? ???? ???? ????	0 (0x0000)
R12720 (R0x31B0)	frame_preamble	0000 0000 dddd dddd	99 (0x0063)
R12722 (R0x31B2)	line_preamble	0000 0000 dddd dddd	57 (0x0039)
R12724 (R0x31B4)	mipi_timing_0	???? dddd dddd dddd	3415 (0x0D57)
R12726 (R0x31B6)	mipi_timing_1	??dd dddd ??dd dddd	2832 (0x0B10)
R12728 (R0x31B8)	mipi_timing_2	??dd dddd ??dd dddd	269 (0x010D)
R12730 (R0x31BA)	mipi_timing_3	??dd dddd ?ddd dddd	1293 (0x050D)
R12732 (R0x31BC)	mipi_timing_4	???? ???? ?ddd dddd	11 (0x000B)
R12736 (R0x31C0)	hispi_timing	0ddd dddd dddd dddd	0 (0x0000)
R12742 (R0x31C6)	hispi_control_status	dddd dddd dddd dddd	32768 (0x8000)
R12776 (R0x31E8)	horizontal_cursor_position_	0000 dddd dddd dddd	0 (0x0000)
R12778 (R0x31EA)	vertical_cursor_position_	0000 dddd dddd dddd	0 (0x0000)
R12780 (R0x31EC)	horizontal_cursor_width_	0000 dddd dddd dddd	0 (0x0000)
R12782 (R0x31EE)	vertical_cursor_width_	0000 dddd dddd dddd	0 (0x0000)
R12786 (R0x31F2)	i2c_ids_mipi_default	dddd dddd dddd dddd	28268 (0x6E6C)

Table 3: Manufacturer-Specific (continued)
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R12796 (R0x31FC)	i2c_ids	dddd dddd dddd dddd	12320 (0x3020)
R13824 (R0x3600)	p_gr_p0q0	dddd dddd dddd dddd	0 (0x0000)
R13826 (R0x3602)	p_gr_p0q1	dddd dddd dddd dddd	0 (0x0000)
R13828 (R0x3604)	p_gr_p0q2	dddd dddd dddd dddd	0 (0x0000)
R13830 (R0x3606)	p_gr_p0q3	dddd dddd dddd dddd	0 (0x0000)
R13832 (R0x3608)	p_gr_p0q4	dddd dddd dddd dddd	0 (0x0000)
R13834 (R0x360A)	p_rd_p0q0	dddd dddd dddd dddd	0 (0x0000)
R13836 (R0x360C)	p_rd_p0q1	dddd dddd dddd dddd	0 (0x0000)
R13838 (R0x360E)	p_rd_p0q2	dddd dddd dddd dddd	0 (0x0000)
R13840 (R0x3610)	p_rd_p0q3	dddd dddd dddd dddd	0 (0x0000)
R13842 (R0x3612)	p_rd_p0q4	dddd dddd dddd dddd	0 (0x0000)
R13844 (R0x3614)	p_bl_p0q0	dddd dddd dddd dddd	0 (0x0000)
R13846 (R0x3616)	p_bl_p0q1	dddd dddd dddd dddd	0 (0x0000)
R13848 (R0x3618)	p_bl_p0q2	dddd dddd dddd dddd	0 (0x0000)
R13850 (R0x361A)	p_bl_p0q3	dddd dddd dddd dddd	0 (0x0000)
R13852 (R0x361C)	p_bl_p0q4	dddd dddd dddd dddd	0 (0x0000)
R13854 (R0x361E)	p_gb_p0q0	dddd dddd dddd dddd	0 (0x0000)
R13856 (R0x3620)	p_gb_p0q1	dddd dddd dddd dddd	0 (0x0000)
R13858 (R0x3622)	p_gb_p0q2	dddd dddd dddd dddd	0 (0x0000)
R13860 (R0x3624)	p_gb_p0q3	dddd dddd dddd dddd	0 (0x0000)
R13862 (R0x3626)	p_gb_p0q4	dddd dddd dddd dddd	0 (0x0000)
R13888 (R0x3640)	p_gr_p1q0	dddd dddd dddd dddd	0 (0x0000)
R13890 (R0x3642)	p_gr_p1q1	dddd dddd dddd dddd	0 (0x0000)
R13892 (R0x3644)	p_gr_p1q2	dddd dddd dddd dddd	0 (0x0000)

Table 3: **Manufacturer-Specific (continued)**
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R13894 (R0x3646)	p_gr_p1q3	dddd dddd dddd dddd	0 (0x0000)
R13896 (R0x3648)	p_gr_p1q4	dddd dddd dddd dddd	0 (0x0000)
R13898 (R0x364A)	p_rd_p1q0	dddd dddd dddd dddd	0 (0x0000)
R13900 (R0x364C)	p_rd_p1q1	dddd dddd dddd dddd	0 (0x0000)
R13902 (R0x364E)	p_rd_p1q2	dddd dddd dddd dddd	0 (0x0000)
R13904 (R0x3650)	p_rd_p1q3	dddd dddd dddd dddd	0 (0x0000)
R13906 (R0x3652)	p_rd_p1q4	dddd dddd dddd dddd	0 (0x0000)
R13908 (R0x3654)	p_bl_p1q0	dddd dddd dddd dddd	0 (0x0000)
R13910 (R0x3656)	p_bl_p1q1	dddd dddd dddd dddd	0 (0x0000)
R13912 (R0x3658)	p_bl_p1q2	dddd dddd dddd dddd	0 (0x0000)
R13914 (R0x365A)	p_bl_p1q3	dddd dddd dddd dddd	0 (0x0000)
R13916 (R0x365C)	p_bl_p1q4	dddd dddd dddd dddd	0 (0x0000)
R13918 (R0x365E)	p_gb_p1q0	dddd dddd dddd dddd	0 (0x0000)
R13920 (R0x3660)	p_gb_p1q1	dddd dddd dddd dddd	0 (0x0000)
R13922 (R0x3662)	p_gb_p1q2	dddd dddd dddd dddd	0 (0x0000)
R13924 (R0x3664)	p_gb_p1q3	dddd dddd dddd dddd	0 (0x0000)
R13926 (R0x3666)	p_gb_p1q4	dddd dddd dddd dddd	0 (0x0000)
R13952 (R0x3680)	p_gr_p2q0	dddd dddd dddd dddd	0 (0x0000)
R13954 (R0x3682)	p_gr_p2q1	dddd dddd dddd dddd	0 (0x0000)
R13956 (R0x3684)	p_gr_p2q2	dddd dddd dddd dddd	0 (0x0000)
R13958 (R0x3686)	p_gr_p2q3	dddd dddd dddd dddd	0 (0x0000)
R13960 (R0x3688)	p_gr_p2q4	dddd dddd dddd dddd	0 (0x0000)
R13962 (R0x368A)	p_rd_p2q0	dddd dddd dddd dddd	0 (0x0000)
R13964 (R0x368C)	p_rd_p2q1	dddd dddd dddd dddd	0 (0x0000)

Table 3: **Manufacturer-Specific (continued)**
1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R13966 (R0x368E)	p_rd_p2q2	dddd dddd dddd dddd	0 (0x0000)
R13968 (R0x3690)	p_rd_p2q3	dddd dddd dddd dddd	0 (0x0000)
R13970 (R0x3692)	p_rd_p2q4	dddd dddd dddd dddd	0 (0x0000)
R13972 (R0x3694)	p_bl_p2q0	dddd dddd dddd dddd	0 (0x0000)
R13974 (R0x3696)	p_bl_p2q1	dddd dddd dddd dddd	0 (0x0000)
R13976 (R0x3698)	p_bl_p2q2	dddd dddd dddd dddd	0 (0x0000)
R13978 (R0x369A)	p_bl_p2q3	dddd dddd dddd dddd	0 (0x0000)
R13980 (R0x369C)	p_bl_p2q4	dddd dddd dddd dddd	0 (0x0000)
R13982 (R0x369E)	p_gb_p2q0	dddd dddd dddd dddd	0 (0x0000)
R13984 (R0x36A0)	p_gb_p2q1	dddd dddd dddd dddd	0 (0x0000)
R13986 (R0x36A2)	p_gb_p2q2	dddd dddd dddd dddd	0 (0x0000)
R13988 (R0x36A4)	p_gb_p2q3	dddd dddd dddd dddd	0 (0x0000)
R13990 (R0x36A6)	p_gb_p2q4	dddd dddd dddd dddd	0 (0x0000)
R14016 (R0x36C0)	p_gr_p3q0	dddd dddd dddd dddd	0 (0x0000)
R14018 (R0x36C2)	p_gr_p3q1	dddd dddd dddd dddd	0 (0x0000)
R14020 (R0x36C4)	p_gr_p3q2	dddd dddd dddd dddd	0 (0x0000)
R14022 (R0x36C6)	p_gr_p3q3	dddd dddd dddd dddd	0 (0x0000)
R14024 (R0x36C8)	p_gr_p3q4	dddd dddd dddd dddd	0 (0x0000)
R14026 (R0x36CA)	p_rd_p3q0	dddd dddd dddd dddd	0 (0x0000)
R14028 (R0x36CC)	p_rd_p3q1	dddd dddd dddd dddd	0 (0x0000)
R14030 (R0x36CE)	p_rd_p3q2	dddd dddd dddd dddd	0 (0x0000)
R14032 (R0x36D0)	p_rd_p3q3	dddd dddd dddd dddd	0 (0x0000)
R14034 (R0x36D2)	p_rd_p3q4	dddd dddd dddd dddd	0 (0x0000)
R14036 (R0x36D4)	p_bl_p3q0	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer-Specific (continued)

1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R14038 (R0x36D6)	p_bl_p3q1	dddd dddd dddd dddd	0 (0x0000)
R14040 (R0x36D8)	p_bl_p3q2	dddd dddd dddd dddd	0 (0x0000)
R14042 (R0x36DA)	p_bl_p3q3	dddd dddd dddd dddd	0 (0x0000)
R14044 (R0x36DC)	p_bl_p3q4	dddd dddd dddd dddd	0 (0x0000)
R14046 (R0x36DE)	p_gb_p3q0	dddd dddd dddd dddd	0 (0x0000)
R14048 (R0x36E0)	p_gb_p3q1	dddd dddd dddd dddd	0 (0x0000)
R14050 (R0x36E2)	p_gb_p3q2	dddd dddd dddd dddd	0 (0x0000)
R14052 (R0x36E4)	p_gb_p3q3	dddd dddd dddd dddd	0 (0x0000)
R14054 (R0x36E6)	p_gb_p3q4	dddd dddd dddd dddd	0 (0x0000)
R14080 (R0x3700)	p_gr_p4q0	dddd dddd dddd dddd	0 (0x0000)
R14082 (R0x3702)	p_gr_p4q1	dddd dddd dddd dddd	0 (0x0000)
R14084 (R0x3704)	p_gr_p4q2	dddd dddd dddd dddd	0 (0x0000)
R14086 (R0x3706)	p_gr_p4q3	dddd dddd dddd dddd	0 (0x0000)
R14088 (R0x3708)	p_gr_p4q4	dddd dddd dddd dddd	0 (0x0000)
R14090 (R0x370A)	p_rd_p4q0	dddd dddd dddd dddd	0 (0x0000)
R14092 (R0x370C)	p_rd_p4q1	dddd dddd dddd dddd	0 (0x0000)
R14094 (R0x370E)	p_rd_p4q2	dddd dddd dddd dddd	0 (0x0000)
R14096 (R0x3710)	p_rd_p4q3	dddd dddd dddd dddd	0 (0x0000)
R14098 (R0x3712)	p_rd_p4q4	dddd dddd dddd dddd	0 (0x0000)
R14100 (R0x3714)	p_bl_p4q0	dddd dddd dddd dddd	0 (0x0000)
R14102 (R0x3716)	p_bl_p4q1	dddd dddd dddd dddd	0 (0x0000)
R14104 (R0x3718)	p_bl_p4q2	dddd dddd dddd dddd	0 (0x0000)
R14106 (R0x371A)	p_bl_p4q3	dddd dddd dddd dddd	0 (0x0000)
R14108 (R0x371C)	p_bl_p4q4	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer-Specific (continued)

1 = read-only, always 1; 0 = read-only, always 0; d = programmable;? = read-only, dynamic

Register Dec(Hex)	Name	Data Format (Binary)	Default Value Dec(Hex)
R14110 (R0x371E)	p_gb_p4q0	dddd dddd dddd dddd	0 (0x0000)
R14112 (R0x3720)	p_gb_p4q1	dddd dddd dddd dddd	0 (0x0000)
R14114 (R0x3722)	p_gb_p4q2	dddd dddd dddd dddd	0 (0x0000)
R14116 (R0x3724)	p_gb_p4q3	dddd dddd dddd dddd	0 (0x0000)
R14118 (R0x3726)	p_gb_p4q4	dddd dddd dddd dddd	0 (0x0000)
R14208 (R0x3780)	poly_sc_enable	d000 0000 0000 0000	0 (0x0000)
R14210 (R0x3782)	poly_origin_c	0000 dddd dddd dddd	0 (0x0000)
R14212 (R0x3784)	poly_origin_r	0000 dddd dddd dddd	0 (0x0000)

Register Descriptions

SMIA Configuration Register Description

Table 4: SMIA Configuration
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
0	15:0	0x2C01	model_id (R/W)	N	N
R0x0000	This register is an alias of R0x3000-1. Read-only. Can be made read/write by clearing R0x301A-B[3].				
2	7:0	0x20	revision_number (R/W)	N	N
R0x0002	Aptina-assigned revision number. Read-only. Can be made read/write by clearing R0x301A-B[3].				
3	7:0	0x06	manufacturer_id (RO)	N	N
R0x0003	Manufacturer ID assigned to Aptina. Read-only. Can be made read/write by clearing R0x301A-B[3].				
4	7:0	0x0A	smia_version (RO)	N	N
R0x0004	This register is an alias of R0x303A. Read-only.				
5	7:0	0xFF	frame_count (RO)	Y	N
R0x0005	This register is an alias of R0x303B. Read-only.				
6	7:0	0x00	pixel_order (RO)	N	N
R0x0006	This register is an alias of R0x3024. Read-only.				
8	15:0	0x0028	data_pedestal (R/W)	N	Y
R0x0008	This register is an alias of R0x301E-F. Read-only. Can be made read/write by clearing R0x301A-B[3].				
64	7:0	0x01	frame_format_model_type (RO)	N	N
R0x0040	Type 1. 2-byte Generic Frame Format Description. Read-only.				
65	7:0	0x12	frame_format_model_subtype (RO)	N	N
R0x0041	Number of descriptors: 1 X (column) descriptor and two Y (row) descriptors. Read-only.				
66	15:0	0x5E50	frame_format_descriptor_0 (RO)	Y	N
R0x0042	X descriptor: Bits[11:0] of this register reflect the current value of x_output_size[11:0]. Upper 4 bits is the pixel code; 5=Visible Pixel Data. Read-only, dynamic.				
68	15:0	0x1002	frame_format_descriptor_1 (RO)	Y	N
R0x0044	Y descriptor: In normal operation, returns 0x1002 to indicate that 2 rows of embedded data are present in the output image. If embedded data is disabled (by selecting the PN9 test pattern using Reg0x3070-1) this register will return 0x1000. Read-only.				
70	15:0	0x5ABC	frame_format_descriptor_2 (RO)	Y	N
R0x0046	Y descriptor: Bits[11:0] of this register reflect the current value of y_output_size[11:0]. Upper 4 bits is the pixel code; 5=Visible Pixel Data. Read-only, dynamic.				
72	15:0	0x0000	frame_format_descriptor_3 (RO)	N	N
R0x0048	Read-only.				
74	15:0	0x0000	frame_format_descriptor_4 (RO)	N	N
R0x004A	Read-only.				
76	15:0	0x0000	frame_format_descriptor_5 (RO)	N	N
R0x004C	Read-only.				
78	15:0	0x0000	frame_format_descriptor_6 (RO)	N	N
R0x004E	Read-only.				
80	15:0	0x0000	frame_format_descriptor_7 (RO)	N	N
R0x0050	Read-only.				
82	15:0	0x0000	frame_format_descriptor_8 (RO)	N	N
R0x0052	Read-only.				

Table 4: SMIA Configuration (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
84	15:0	0x0000	frame_format_descriptor_9 (RO)	N	N
R0x0054	Read-only.				
86	15:0	0x0000	frame_format_descriptor_10 (RO)	N	N
R0x0056	Read-only.				
88	15:0	0x0000	frame_format_descriptor_11 (RO)	N	N
R0x0058	Read-only.				
90	15:0	0x0000	frame_format_descriptor_12 (RO)	N	N
R0x005A	Read-only.				
92	15:0	0x0000	frame_format_descriptor_13 (RO)	N	N
R0x005C	Read-only.				
94	15:0	0x0000	frame_format_descriptor_14 (RO)	N	N
R0x005E	Read-only.				
128	15:0	0x0001	analogue_gain_capability (RO)	N	N
R0x0080	Indicates the provision of separate (per-color) analog gain control. The sensor supports both global and separate (per-color) analog gain control. Read-only.				
132	15:0	0x0008	analogue_gain_code_min (RO)	N	N
R0x0084	Minimum gain code. Read-only.				
134	15:0	0x00FF	analogue_gain_code_max (RO)	N	N
R0x0086	Maximum gain code. Read-only.				
136	15:0	0x0001	analogue_gain_code_step (RO)	N	N
R0x0088	Gain code step size. Read-only.				
138	15:0	0x0000	analogue_gain_type (RO)	N	N
R0x008A	Indicates support for analog gain coding type 0 (baseline SMIA). Read-only.				
140	15:0	0x0001	analogue_gain_m0 (RO)	N	N
R0x008C	Constants for the gain equation. Read-only.				
142	15:0	0x0000	analogue_gain_c0 (RO)	N	N
R0x008E	Constants for the gain equation. Read-only.				
144	15:0	0x0000	analogue_gain_m1 (RO)	N	N
R0x0090	Constants for the gain equation. Read-only.				
146	15:0	0x0008	analogue_gain_c1 (RO)	N	N
R0x0092	Constants for the gain equation. Read-only.				
192	7:0	0x01	data_format_model_type (RO)	N	N
R0x00C0	Indicates the use of 2-byte data format. Read-only.				
193	7:0	0x05	data_format_model_subtype (RO)	N	N
R0x00C1	Indicates the provision of 3 data format descriptors. Read-only.				
194	15:0	0x0A0A	data_format_descriptor_0 (RO)	N	N
R0x00C2	Indicates support for RAW10, uncompressed data format. Read-only.				
196	15:0	0x0808	data_format_descriptor_1 (RO)	N	N
R0x00C4	Indicates support for RAW8 data format in which the two LSB of each 10-bit pixel data value are discarded. Read-only.				
198	15:0	0x0A08	data_format_descriptor_2 (RO)	N	N
R0x00C6	Indicates support for RAW8 data format in which each 10-bit pixel data value is compressed to an 8-bit value. Read-only.				
200	15:0	0x0C0C	data_format_descriptor_3 (RO)	N	N
R0x00C8	Read-only.				

Table 4: SMIA Configuration (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
202 R0x00CA	15:0	0x0C08	data_format_descriptor_4 (RO)	N	N
	Read-only.				
204 R0x00CC	15:0	0x0000	data_format_descriptor_5 (RO)	N	N
	Read-only.				
206 R0x00CE	15:0	0x0000	data_format_descriptor_6 (RO)	N	N
	Read-only.				
256 R0x0100	7:0	0x00	mode_select (R/W)	Y	N
	This register field is an alias of R0x301A[2].				
257 R0x0101	7:0	0x00	image_orientation (R/W)		
	7:2	X	Reserved		
	1	0x00	image_orientation_vertical_flip This register field is an alias of R0x3040-1[15].	Y	YM
	0	0x00	image_orientation_horizontal_mirror This register field is an alias of R0x3040-1[14].	Y	YM
259 R0x0103	7:0	0x00	software_reset (R/W)	N	Y
	This register field is an alias of R0x301A-B[0].				
260 R0x0104	7:0	0x00	grouped_parameter_hold (R/W)	N	N
	This register field is an alias of R0x301A-B[15].				
261 R0x0105	7:0	0x00	mask_corrupted_frames (R/W)	N	Y
	This register field is an alias of R0x301A-B[9].				
272 R0x0110	7:0	0x00	ccp2_channel_identifier (R/W)	Y	N
	When the CCP2 serial pixel data interface is in use, this 3-bit field supplies the DMA channel identifier that will be used within the CCP2 embedded synchronization codes. When the MIPI serial pixel data interface is in use, the low 2 bits of this 3-bit field supply the Virtual Channel (VC) identiifer in the Data Identifier (DI) byte which forms part of the short and long packet headers.				
273 R0x0111	7:0	0x01	ccp2_signalling_mode (R/W)	Y	N
	0: Use Data/Clock signaling on the CCP2 serial interface. 1: Use Data/Strobe signaling on the CCP2 serial interface.				
274 R0x0112	15:0	0x0C0C	ccp_data_format (R/W)	Y	N
	[7:0]: The bit-width of the compressed pixel data [15:8]: The bit-width of the uncompressed pixel data The value in this register must match one of the valid data_format_descriptor registers (R0x00C2-R0x00C7).				
288 R0x0120	7:0	0x00	gain_mode (R/W)	N	N
	This read/write bit has no function.				
512 R0x0200	15:0	0x03F2	fine_integration_time (R/W)	Y	N
	Integration time programmed in units of pck. This register is an alias of R0x3014-5.				
514 R0x0202	15:0	0x0010	coarse_integration_time (R/W)	Y	N
	Integration time programmed in units of line_length_pck. This register is an alias of R0x3012-3.				
516 R0x0204	15:0	0x0008	analogue_gain_code_global (R/W)	Y	N
	This register is an alias of R0x3028-9.				
518 R0x0206	15:0	0x0008	analogue_gain_code_greenr (R/W)	Y	N
	This register is an alias of R0x302A-B.				
520 R0x0208	15:0	0x0008	analogue_gain_code_red (R/W)	Y	N
	This register is an alias of R0x302C-D.				
522 R0x020A	15:0	0x0008	analogue_gain_code_blue (R/W)	Y	N
	This register is an alias of R0x302E-F.				

Table 4: **SMIA Configuration (continued)**
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
524 R0x020C	15:0	0x0008	analogue_gain_code_greenb (R/W)	Y	N
	This register is an alias of Reg0x3030-1.				
526 R0x020E	15:0	0x0100	digital_gain_greenr (R/W)	Y	N
	This register is an alias of R0x3032-3.				
528 R0x0210	15:0	0x0100	digital_gain_red (R/W)	Y	N
	This register is an alias of R0x3034-5.				
530 R0x0212	15:0	0x0100	digital_gain_blue (R/W)	Y	N
	This register is an alias of Reg0x3036-7.				
532 R0x0214	15:0	0x0100	digital_gain_greenb (R/W)	Y	N
	This register is an alias of Reg0x3038-9.				
768 R0x0300	15:0	0x0003	vt_pix_clk_div (R/W)	N	Y
	Clock divisor applied to video timing system clock to generate video timing pixel clock.				
770 R0x0302	15:0	0x0001	vt_sys_clk_div (R/W)	N	N
	Clock divisor applied to PLL output clock to generate video timing system clock. Read-only.				
772 R0x0304	15:0	0x0002	pre_pll_clk_div (R/W)	N	Y
	Clock divisor applied to EXTCLK to generate PLL input clock.				
774 R0x0306	15:0	0x0030	pll_multiplier (R/W)	N	Y
	Clock multiplier applied to PLL input clock.				
776 R0x0308	15:0	0x000C	op_pix_clk_div (R/W)	N	Y
	Clock divisor applied to the output system clock to generate the output pixel clock.				
778 R0x030A	15:0	0x0001	op_sys_clk_div (R/W)	N	Y
	Clock divisor applied to PLL output clock to generate output system clock. Read-only.				
832 R0x0340	15:0	0x0B4B	frame_length_lines (R/W)	Y	YM
	This register is an alias of R0x300A-B.				
834 R0x0342	15:0	0x1D10	line_length_pck (R/W)	Y	YM
	This register is an alias of R0x300C-D.				
836 R0x0344	15:0	0x0070	x_addr_start (R/W)	Y	N
	This register is an alias of R0x3004-5.				
838 R0x0346	15:0	0x0008	y_addr_start (R/W)	Y	YM
	This register is an alias of R0x3002-5.				
840 R0x0348	15:0	0x0EBF	x_addr_end (R/W)	Y	N
	This register is an alias of R0x3008-9.				
842 R0x034A	15:0	0x0AC3	y_addr_end (R/W)	Y	YM
	This register is an alias of R0x3006-7.				
844 R0x034C	15:0	0x0E50	x_output_size (R/W)	Y	N
	Set X output size of displayed image. Bit[0] is read-only 0. The default value of this register is set to be consistent with the default values of x_addr_end and x_addr_start.				
846 R0x034E	15:0	0x0ABC	y_output_size (R/W)	Y	N
	Set Y output size of the displayed image. Bit[0] is read-only 0. The default value of this registers set to be consistent with the default values of y_addr_end and y_addr_start. The output image will have two additional rows containing embedded data, in accordance with the frame format descriptors.				
896 R0x0380	15:0	0x0001	x_even_inc (RO)	N	N
	Read-only. The fixed value of 1 constrains subsampling operation to use adjacent pixels of a pixel quad.				
898 R0x0382	15:0	0x0001	x_odd_inc (R/W)	Y	YM
	This register field is an alias of R0x3040-1[8:6].				

Table 4: SMIA Configuration (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
900 R0x0384	15:0	0x0001	y_even_inc (RO)	N	N
	Read-only. The fixed value of 1 constrains subsampling operation to use adjacent pixels of a pixel quad.				
902 R0x0386	15:0	0x0001	y_odd_inc (R/W)	Y	YM
	This register field is an alias of R0x3040-1[5:0].				
1024 R0x0400	15:0	0x0000	scaling_mode (R/W)	Y	N
	0: Disable scaler 1: Enable horizontal scaling 2: Enable horizontal and vertical scaling 3: Reserved				
1026 R0x0402	15:0	0x0000	spatial_sampling (R/W)	Y	N
	0: Bayer sampling 1: Co-sited sampling				
1028 R0x0404	15:0	0x0010	scale_m (R/W)	Y	N
	Scale factor M.				
1030 R0x0406	15:0	0x0010	scale_n (RO)	N	N
	Scale factor N. Read-only.				
1280 R0x0500	15:0	0x0001	compression_mode (RO)	N	Y
	0x0001 = 10-bit to 8-bit and 12-bit to 8-bit compression use the DPCM/PCM Simple Predictor algorithm. Read-only. This register controls the algorithm that is to be used for compression. The sensor only supports a single algorithm and therefore this register is read-only. This register does not control whether data compression is enabled; that is controlled by the ccp_data_format register (Reg0x0012-3).				
1536 R0x0600	15:0	0x0000	test_pattern_mode (R/W)	N	Y
	This register is an alias of R0x3070-1.				
1538 R0x0602	15:0	0x0000	test_data_red (R/W)	N	Y
	This register is an alias of R0x3072-3.				
1540 R0x0604	15:0	0x0000	test_data_greenr (R/W)	N	Y
	This register is an alias of Reg0x3074-5.				
1542 R0x0606	15:0	0x0000	test_data_blue (R/W)	N	Y
	This register is an alias of R0x3076-7.				
1544 R0x0608	15:0	0x0000	test_data_greenb (R/W)	N	Y
	This register is an alias of R0x3078-8.				
1546 R0x060A	15:0	0x0000	horizontal_cursor_width (R/W)	N	N
	This register is an alias of R0x31EC-D.				
1548 R0x060C	15:0	0x0000	horizontal_cursor_position (R/W)	N	N
	This register is an alias of R0x31E8-9.				
1550 R0x060E	15:0	0x0000	vertical_cursor_width (R/W)	N	N
	This register is an alias of R0x31EE-F.				
1552 R0x0610	15:0	0x0000	vertical_cursor_position (R/W)	N	N
	This register is an alias of R0x31EA-B.				

SMIA Parameter Limits Register Description

Table 5: SMIA Parameter Limits

R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
4096 R0x1000	15:0	0x0001	integration_time_capability (RO)	N	N
	Indicates the provision of coarse and fine integration time control. Read-only. Can be made read/write by clearing R0x301A-B[3].				
4100 R0x1004	15:0	0x0000	coarse_integration_time_min (R/W)	N	N
	The minimum coarse integration time. Read-only. Can be made read/write by clearing R0x301A-B[3].				
4102 R0x1006	15:0	0x0001	coarse_integration_time_max_margin (R/W)	N	N
	The maximum coarse integration time is (frame_length_lines - coarse_integration_time_max_margin). Read-only. Can be made read/write by clearing R0x301A-B[3]. In the sensor, this limit can be broken.				
4104 R0x1008	15:0	0x03F2	fine_integration_time_min (R/W)	N	N
	The minimum fine integration time. Read-only. Can be made read/write by clearing R0x301A-B[3].				
4106 R0x100A	15:0	0x027E	fine_integration_time_max_margin (R/W)	N	N
	The maximum fine integration time is (line_length_pck - fine_integration_time_max_margin). Read-only. Can be made read/write by clearing R0x301A-B[3].				
4224 R0x1080	15:0	0x0001	digital_gain_capability (RO)	N	N
	Indicates the provision of separate (per-color) digital gain control. Read-only.				
4228 R0x1084	15:0	0x0100	digital_gain_min (RO)	N	N
	UFIX16. Minimum value of digital gain is 1.0. Read-only.				
4230 R0x1086	15:0	0x0700	digital_gain_max (RO)	N	N
	UFIX16. Maximum value of digital gain is 7.0. Read-only.				
4232 R0x1088	15:0	0x0100	digital_gain_step_size (RO)	N	N
	UFIX16. Step size for digital gain is 1.0. Read-only.				
4352 R0x1100	15:0	0x4000	min_ext_clk_freq_mhz_1 (RO)	N	N
	FLP32. Minimum external clock frequency into PLL is 2.0 MHz. Read-only.				
4354 R0x1102	15:0	0x0000	min_ext_clk_freq_mhz_2 (RO)	N	N
	FLP32. Minimum external clock frequency into PLL is 2.0 MHz. Read-only.				
4356 R0x1104	15:0	0x4280	max_ext_clk_freq_mhz_1 (RO)	N	N
	FLP32. Maximum external clock frequency into PLL is 64.0 MHz. Read-only.				
4358 R0x1106	15:0	0x0000	max_ext_clk_freq_mhz_2 (RO)	N	N
	FLP32. Maximum external clock frequency into PLL is 64.0 MHz. Read-only.				
4360 R0x1108	15:0	0x0001	min_pre_pll_clk_div (RO)	N	N
	Minimum clock divisor applied to PLL input clock. Read-only.				
4362 R0x110A	15:0	0x0040	max_pre_pll_clk_div (RO)	N	N
	Maximum clock divisor applied to PLL input clock. Read-only.				
4364 R0x110C	15:0	0x4080	min_pll_ip_freq_mhz_1 (RO)	N	N
	FLP32. Minimum clock frequency into the PFD of the PLL is 4.0 MHz. Read-only.				
4366 R0x110E	15:0	0x0000	min_pll_ip_freq_mhz_2 (RO)	N	N
	FLP32. Minimum clock frequency into the PFD of the PLL is 4.0 MHz. Read-only.				
4368 R0x1110	15:0	0x41C0	max_pll_ip_freq_mhz_1 (RO)	N	N
	FLP32. Maximum clock frequency into the PFD of the PLL is 24 MHz. Read-only.				
4370 R0x1112	15:0	0x0000	max_pll_ip_freq_mhz_2 (RO)	N	N
	FLP32. Maximum clock frequency into the PFD of the PLL is 24 MHz. Read-only.				

Table 5: SMIA Parameter Limits (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
4372 R0x1114	15:0	0x0020	min_pll_multiplier (RO)	N	N
	Minimum multiplier applied by PLL. Read-only.				
4374 R0x1116	15:0	0x0180	max_pll_multiplier (RO)	N	N
	Maximum multiplier applied by PLL. Read-only.				
4376 R0x1118	15:0	0x43C0	min_pll_op_freq_mhz_1 (RO)	N	N
	FLP32. Minimum output frequency supported by the PLL is 384.0 MHz. Read-only.				
4378 R0x111A	15:0	0x0000	min_pll_op_freq_mhz_2 (RO)	N	N
	FLP32. Minimum output frequency supported by the PLL is 384.0 MHz. Read-only.				
4380 R0x111C	15:0	0x4440	max_pll_op_freq_mhz_1 (RO)	N	N
	FLP32. Maximum output frequency supported by the PLL is 768.0 MHz. Read-only.				
4382 R0x111E	15:0	0x0000	max_pll_op_freq_mhz_2 (RO)	N	N
	FLP32. Maximum output frequency supported by the PLL is 768.0 MHz. Read-only.				
4384 R0x1120	15:0	0x0001	min_vt_sys_clk_div (RO)	N	N
	The video timing sys_clk has a fixed divisor. Read-only.				
4386 R0x1122	15:0	0x0001	max_vt_sys_clk_div (RO)	N	N
	The video timing sys_clk has a fixed divisor. Read-only.				
4388 R0x1124	15:0	0x41C0	min_vt_sys_clk_freq_mhz_1 (RO)	N	N
	FLP32. Minimum frequency for the video timing sys_clk is 24.0 MHz.				
4390 R0x1126	15:0	0x0000	min_vt_sys_clk_freq_mhz_2 (RO)	N	N
	FLP32. Minimum frequency for the video timing sys_clk is 24.0 MHz.				
4392 R0x1128	15:0	0x4440	max_vt_sys_clk_freq_mhz_1 (RO)	N	N
	Maximum frequency for the video timing sys_clk is 768.0 MHz. Read-only.				
4394 R0x112A	15:0	0x0000	max_vt_sys_clk_freq_mhz_2 (RO)	N	N
	Maximum frequency for the video timing sys_clk is 768.0 MHz. Read-only.				
4396 R0x112C	15:0	0x4019	min_vt_pix_clk_freq_mhz_1 (RO)	N	N
	FLP32. Minimum frequency for video timing pix_clk is 2.4 MHz. Read-only.				
4398 R0x112E	15:0	0x999A	min_vt_pix_clk_freq_mhz_2 (RO)	N	N
	FLP32. Minimum frequency for video timing pix_clk is 2.4 MHz. Read-only.				
4400 R0x1130	15:0	0x42C0	max_vt_pix_clk_freq_mhz_1 (RO)	N	N
	FLP32. Maximum frequency for video timing pix_clk is 80.0 MHz. Read-only.				
4402 R0x1132	15:0	0x0000	max_vt_pix_clk_freq_mhz_2 (RO)	N	N
	FLP32. Maximum frequency for video timing pix_clk is 80.0 MHz. Read-only.				
4404 R0x1134	15:0	0x0004	min_vt_pix_clk_div (RO)	N	N
	Minimum divisor for the video timing pix_clk. Read-only.				
4406 R0x1136	15:0	0x0006	max_vt_pix_clk_div (RO)	N	N
	Maximum divisor for the video timing pix_clk. Read-only.				
4416 R0x1140	15:0	0x0091	min_frame_length_lines (R/W)	N	N
	Minimum frame length. Read-only. Can be made read/write by clearing R0x301A-B[3].				
4418 R0x1142	15:0	0xFFFF	max_frame_length_lines (R/W)	N	N
	Maximum frame length. The maximum frame length is only constrained by the size of the read/write field in the frame_length_lines register (16-bits). Read-only. Can be made read/write by clearing R0x301A-B[3].				
4420 R0x1144	15:0	0x0670	min_line_length_pck (R/W)	N	N
	Minimum line length. Read-only. Can be made read/write by clearing R0x301A-B[3].				

Table 5: SMIA Parameter Limits (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
4422	15:0	0xFFFE	max_line_length_pck (R/W)	N	N
R0x1146	Maximum line length. The maximum line length is only constrained by the size of the read/write field in the line_length_pck register (16 bits). Read-only. Can be made read/write by clearing R0x301A-B[3].				
4424	15:0	0x046E	min_line_blanking_pck (R/W)	N	N
R0x1148	Minimum line blanking time. Read-only. Can be made read/write by clearing Reg0x301A-B[3].				
4426	15:0	0x008F	min_frame_blanking_lines (R/W)	N	N
R0x114A	Minimum frame blanking time. Read-only. Can be made read/write by clearing Reg0x301A-B[3].				
4448	15:0	0x0001	min_op_sys_clk_div (RO)	N	N
R0x1160	Minimum divisor for the output sys_clk. Read-only.				
4450	15:0	0x0001	max_op_sys_clk_div (RO)	N	N
R0x1162	Maximum divisor for the output sys_clk. Read-only.				
4452	15:0	0x4199	min_op_sys_clk_freq_mhz_1 (RO)	N	N
R0x1164	FLP32. Minimum frequency for output sys_clk is 19.2 MHz. Read-only.				
4454	15:0	0x999A	min_op_sys_clk_freq_mhz_2 (RO)	N	N
R0x1166	FLP32. Minimum frequency for output sys_clk is 19.2 MHz. Read-only.				
4456	15:0	0x4440	max_op_sys_clk_freq_mhz_1 (RO)	N	N
R0x1168	FLP32. Maximum frequency for output sys_clk is 768.0 MHz. Read-only.				
4458	15:0	0x0000	max_op_sys_clk_freq_mhz_2 (RO)	N	N
R0x116A	FLP32. Maximum frequency for output sys_clk is 768.0 MHz. Read-only.				
4460	15:0	0x0008	min_op_pix_clk_div (RO)	N	N
R0x116C	Minimum divisor for output pix_clk. Read-only.				
4462	15:0	0x000C	max_op_pix_clk_div (RO)	N	N
R0x116E	Maximum divisor for output pix_clk. Read-only.				
4464	15:0	0x4019	min_op_pix_clk_freq_mhz_1 (RO)	N	N
R0x1170	FLP32. Minimum frequency for output pix_clk is 2.4 MHz. Read-only.				
4466	15:0	0x999A	min_op_pix_clk_freq_mhz_2 (RO)	N	N
R0x1172	FLP32. Minimum frequency for output pix_clk is 2.4 MHz. Read-only.				
4468	15:0	0x42C0	max_op_pix_clk_freq_mhz_1 (RO)	N	N
R0x1174	FLP32. Maximum frequency for output pix_clk is 96.0 MHz. Read-only.				
4470	15:0	0x0000	max_op_pix_clk_freq_mhz_2 (RO)	N	N
R0x1176	FLP32. Maximum frequency for output pix_clk is 96.0 MHz. Read-only.				
4480	15:0	0x0018	x_addr_min (RO)	N	N
R0x1180	Minimum value for x_addr_start, x_addr_end. Read-only.				
4482	15:0	0x0000	y_addr_min (RO)	N	N
R0x1182	Minimum value for y_addr_start, y_addr_end. Read-only.				
4484	15:0	0x0F27	x_addr_max (RO)	N	N
R0x1184	Maximum value for x_addr_start, x_addr_end. Read-only.				
4486	15:0	0x0ACB	y_addr_max (RO)	N	N
R0x1186	Maximum value for y_addr_start, y_addr_end. Read-only.				
4544	15:0	0x0001	min_even_inc (RO)	N	N
R0x11C0	Minimum value for increment of even X/Y addresses when subsampling is enabled. Read-only.				
4546	15:0	0x0001	max_even_inc (RO)	N	N
R0x11C2	Maximum value for increment of even X/Y addresses when subsampling is enabled. Read-only.				
4548	15:0	0x0001	min_odd_inc (RO)	N	N
R0x11C4	Minimum value for increment of odd X/Y addresses when subsampling is enabled. Read-only.				

Table 5: SMIA Parameter Limits (continued)

R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
4550 R0x11C6	15:0	0x003F	max_odd_inc (RO)	N	N
	This register shows the maximum capability for the subsampling mode. The formula to understand the subsample factor is (1+odd_inc)/2. A value of 3 will lead to a skip2x. The available values in the x-direction are 1, 3, and 7. For the y-direction, the available values are 1, 3, 7, 15, 31, and 63.				
4608 R0x1200	15:0	0x0002	scaling_capability (RO)	N	N
	Indicates the provision of a full (horizontal and vertical) scaler. Read-only.				
4612 R0x1204	15:0	0x0010	scaler_m_min (RO)	N	N
	Indicates the minimum M value for the scaler. Read-only.				
4614 R0x1206	15:0	0x0080	scaler_m_max (RO)	N	N
	Indicates the maximum M value for the scaler. Read-only.				
4616 R0x1208	15:0	0x0010	scaler_n_min (RO)	N	N
	Indicates the minimum N value for the scaler. Read-only.				
4618 R0x120A	15:0	0x0010	scaler_n_max (RO)	N	N
	Indicates the maximum N value for the scaler. Read-only.				
4864 R0x1300	15:0	0x0001	compression_capability (RO)	N	N
	Indicates the capability for performing 10-bit to 8-bit pixel data compression. Read-only.				
5120 R0x1400	15:0	0x0242	matrix_element_redinred (R/W)	N	N
	Read-only. Can be made read/write by clearing R0x301A-B[3].				
5122 R0x1402	15:0	0xFF00	matrix_element_greeninred (R/W)	N	N
	Color-correction matrix. Read-only. Can be made read/write by clearing R0x301A-B[3].				
5124 R0x1404	15:0	0xFFBE	matrix_element_blueinred (R/W)	N	N
	Color-correction matrix. Read-only. Can be made read/write by clearing R0x301A-B[3].				
5126 R0x1406	15:0	0xFFB4	matrix_element_redingreen (R/W)	N	N
	Color-correction matrix. Read-only. Can be made read/write by clearing R0x301A-B[3].				
5128 R0x1408	15:0	0x0200	matrix_element_greeningreen (R/W)	N	N
	Color-correction matrix. Read-only. Can be made read/write by clearing R0x301A-B[3].				
5130 R0x140A	15:0	0xFF4D	matrix_element_blueingreen (R/W)	N	N
	Color-correction matrix. Read-only. Can be made read/write by clearing R0x301A-B[3].				
5132 R0x140C	15:0	0xFFF1	matrix_element_redinblue (R/W)	N	N
	Color-correction matrix. Read-only. Can be made read/write by clearing R0x301A-B[3].				
5134 R0x140E	15:0	0xFF34	matrix_element_greeninblue (R/W)	N	N
	Color-correction matrix. Read-only. Can be made read/write by clearing R0x301A-B[3].				
5136 R0x1410	15:0	0x01DC	matrix_element_blueinblue (R/W)	N	N
	Color-correction matrix. Read-only. Can be made read/write by clearing R0x301A-B[3].				

Manufacturer-Specific Register Description

Table 6: Manufacturer-Specific
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12288 R0x3000	15:0	0x2C01	model_id_ (R/W)	N	N
	Model ID. Read-only. Can be made read/write by clearing R0x301A-B[3].				
12290 R0x3002	15:0	0x0008	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 R0x3004	15:0	0x0070	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. The minimum value for x_addr is 24. This is the beginning of the 8x border pixels at each edge of the array. The recommended start of the array is 32 and the recommended finish is 3872.				
12294 R0x3006	15:0	0x0AC3	y_addr_end_ (R/W)	Y	YM
	The last row of visible pixels to be read out.				
12296 R0x3008	15:0	0x0EBF	x_addr_end_ (R/W)	Y	N
	The last column of visible pixels to be read out. The minimum value for x_addr is 24. This is the beginning of the 8x border pixels at each edge of the array. The recommended start of the array is 32 and the recommended finish is 3872.				
12298 R0x300A	15:0	0x0B4B	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 R0x300C	15:0	0x1D10	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.				
12304 R0x3010	15:0	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 R0x3012	15:0	0x0010	coarse_integration_time_ (R/W)	Y	N
	Integration time specified in multiples of line_length_pck_.				
12308 R0x3014	15:0	0x03F2	fine_integration_time_ (R/W)	Y	N
	Integration time specified as a number of pixel clocks.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12310 R0x3016	15:0	0x0121	row_speed (R/W)		
	15:11	X	Reserved		
	10:8	0x01	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	0x02	row_speed_opclk_delay Number of half-system-clock-cycle increments to delay the rising edge of PIXCLK relative to transitions on FRAME_VALID, LINE_VALID, and DOUT.	N	N
	3	X	Reserved		
	2:0	0x01	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 R0x3018	15:0	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 R0x301A	15:0	0x0018	reset_register (R/W)		
	15	0x00	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	0x00	reset_register_gain_insert 1: Gain values will al 0: Gain will not update if an integration time change is in progress.	N	Y
	13	X	Reserved		
	12	0x00	reset_register_smia_serialiser_dis This bit disables the SMIA high-speed serializer and differential output buffers.	N	N
	11	X	Reserved		
	10	0x00	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	0x00	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

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
	8	0x00	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 R0x3026-7.	N	N
	7	0x00	reset_register_parallel_en 0: The parallel data interface (DOUT[9:0], LINE_VALID, FRAME_VALID, and PIXCLK) is disabled and the outputs are placed in a high-impedance state. 1: The parallel data interface is enabled. The output signals can be switched between a driven and a high-impedance state using output-enable control.	N	N
	6	0x00	reset_register_drive_pins 0: The parallel data interface (DOUT[9:0], LINE_VALID, FRAME_VALID, and PIXCLK) may enter a high-impedance state (depending upon the configuration of R0x3026). 1: The parallel data interface is driven. This bit is "don't-care" unless bit[7]=1.	N	N
	5	0x00	Reserved		
	4	0x01	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	0x01	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	0x00	reset_register_stream Setting this bit places the sensor in streaming mode. Clearing this bit places the sensor in a low power mode. Entry and exit from streaming mode can also be controlled from the signal interface.	Y	N
	1	0x00	reset_register_restart 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. Default value is zero.	N	Y
	0	0x00	reset_register_reset Setting this bit initiates a reset sequence: the frame being generated will be truncated. Default value is zero.	N	Y
12316 R0x301C	7:0	0x00	mode_select_ (R/W)	Y	N
	This bit is an alias of R0x301A-B[2].				
12317 R0x301D	7:0	0x00	image_orientation_ (R/W)		
	7:2	X	Reserved		
	1	0x00	image_orientation_vert_flip This bit is an alias of R0x3040[15].	Y	YM
	0	0x00	image_orientation_horiz_mirror This bit is an alias of R0x3040[14].	Y	YM
12318 R0x301E	15:0	0x0028	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 R0x301A-B[3].				

Table 6: **Manufacturer-Specific (continued)**
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12321 R0x3021	7:0	0x00	software_reset_ (R/W)	N	Y
	This bit is an alias of R0x301A-B[0].				
12322 R0x3022	7:0	0x00	grouped_parameter_hold_ (R/W)	N	N
	This bit is an alias of R0x301A-B[15].				
12323 R0x3023	7:0	0x00	mask_corrupted_frames_ (R/W)	N	N
	This bit is an alias of R0x301A-B[9].				
12324 R0x3024	7:0	0x00	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 R0x3040[1:0].				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12326 R0x3026	15:0	0xFFFF	gpi_status (R/W)		
	15:13	0x07	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	0x07	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
	9:7	0x07	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 R0x301A-B[8]=0.	N	N
	6:4	0x07	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: SADDRR function is not controlled by any pin Must be set to 7 if R0x301A-B[8]=0.	N	N
	3	RO	gpi_status_gpi3 Read-only. Return the current state of the GPI3 input pin. Invalid if R0x301A-B[8]=0.	N	N
	2	RO	gpi_status_gpi2 Read-only. Return the current state of the GPI2 input pin. Invalid if R0x301A-B[8]=0.	N	N
	1	RO	gpi_status_gpi1 Read-only. Return the current state of the GPI1 input pin. Invalid if R0x301A-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

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12328 R0x3028	15:0	0x0008	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 R0x302A	15:0	0x0008	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 R0x302C	15:0	0x0008	analogue_gain_code_red_ (R/W)	Y	N
	The gain code written to this register sets the gain for red pixels.				
12334 R0x302E	15:0	0x0008	analogue_gain_code_blue_ (R/W)	Y	N
	The gain code written to this register sets the gain for blue pixels.				
12336 R0x3030	15:0	0x0008	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 R0x3032	15:0	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 R0x3056[14:12].				
12340 R0x3034	15:0	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 R0x305A[14:12].				
12342 R0x3036	15:0	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 R0x3058[14:12].				
12344 R0x3038	15:0	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 R0x305C[14:12].				
12346 R0x303A	7:0	0x0A	smia_version_ (RO)	N	N
	Return the value 10 to indicate an implementation of revision 1.0 of the SMIA specification. Read-only.				
12347 R0x303B	7:0	0xFF	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 R0x301A-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 R0x303C	15:0	0x0000	frame_status (RO)		
	7:2	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 R0x301A[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 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12352 R0x3040	15:0	0x0041	read_mode (R/W)		
	15	0x00	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 R0x3024).	Y	YM
	14	0x00	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 R0x3024).	Y	YM
	13	0x00	read_mode_y_sum_en Enable Y summing mode for binning. The register y_odd_inc must be set to 3 to enable this bit.	Y	N
	12	0x00	read_mode_bin_sum Enable X summing mode for binning. The register x_odd_inc must be set to 3 to enable this bit.	Y	N
	11	0x00	read_mode_x_bin_en Enable analog 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	0x00	read_mode_xy_bin_en Enable analog binning in X and Y (column and row) directions. When set, both x_odd_inc and y_odd_inc must be set to 3 for bining or 7 for binning and skipping, along with other changes.	Y	N
	9	0x00	read_mode_low_power Enable low power mode. This will automatically halve the pixel clock speed. Can not be used when pc_speed[2:0] = 4.	Y	YM
	8:6	0x01	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 1/4.	Y	YM
	5:0	0x01	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 1/4. 15= Read out 1 out of 8 pixel pairs to reduce the amount of vertical data in a frame by 1/8. 31 = Read out 1 out of 16 pixel pairs to reduce the amount of vertical data in a frame by 1/16. 63 = Read out 1 out of 32 pixel pairs to reduce the amount of vertical data in a frame by 1/32.	Y	YM

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12358 R0x3046	15:0	0x0600	flash (R/W)		
	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	0x00	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	0x00	flash_frame_delay Flash pulse delay measured in frames.	N	N
	10	0x01	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	0x01	flash_every_frame 1: Flash should be enabled every frame. 0: Flash should be enabled for 1 frame only.	N	N
	8	0x00	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	0x00	flash_invert_flash Invert flash output signal. When set, the FLASH output signal will be active low.	N	N
	6:0	X	Reserved		
12360 R0x3048	15:0	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 PIXCLK cycle increments (by default, PIXCLK = 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 R0x3056	15:0	0x1040	green1_gain (R/W)		
	15	X	Reserved		
	14:12	0x01	green1_gain_digital_gain Green1 Digital Gain. Legal values 1-7.	Y	N
	11:9	0x00	green1_gain_analog_gain_2 Analog gain = [8/(8 - bits [11:9])] * Analog gain 1 Valid values for bits[11:9] are 0 , 4 ,and 6.	N	N
	8:7	0x00	green1_gain_analog_gain_1 Analog gain 1 = { (bit [8] + 1) * (bit [7] + 1) * initial gain }	Y	N
	6:0	0x40	green1_gain_initial_gain Initial gain = bits [6:0] * 1/64.	Y	N

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12376 R0x3058	15:0	0x1040	blue_gain (R/W)		
	15	X	Reserved		
	14:12	0x01	blue_gain_digital_gain Blue Digital Gain. Legal values 1-7.	Y	N
	11:9	0x00	blue_gain_analog_gain_2 Analog gain = [8/(8 - bits [11:9])] * Analog gain 1 Valid values for bits[11:9] are 0 , 4, and 6.	N	N
	8:7	0x00	blue_gain_analog_gain_1 Analog gain 1 = { (bit [8] + 1) * (bit [7] + 1) * initial gain }	Y	N
	6:0	0x40	blue_gain_initial_gain Initial gain = bits [6:0] * 1/64.	Y	N
12378 R0x305A	15:0	0x1040	red_gain (R/W)		
	15	X	Reserved		
	14:12	0x01	red_gain_digital_gain Red Digital Gain. Legal values 1-7.	Y	N
	11:9	0x00	red_gain_analog_gain_2 Analog gain = [8/(8 - bits [11:9])] * Analog gain 1 Valid values for bits[11:9] are 0 , 4, and 6.	N	N
	8:7	0x00	red_gain_analog_gain_1 Analog gain 1 = { (bit [8] + 1) * (bit [7] + 1) * initial gain }	Y	N
	6:0	0x40	red_gain_initial_gain Initial gain = bits [6:0] * 1/64.	Y	N
12380 R0x305C	15:0	0x1040	green2_gain (R/W)		
	15	X	Reserved		
	14:12	0x01	green2_gain_digital_gain Green2 Digital Gain. Legal values 1-7.	Y	N
	11:9	0x00	green2_gain_analog_gain_2 Analog gain = [8/(8 - bits [11:9])] * Analog gain 1 Valid values for bits[11:9] are 0 , 4, and 6.	N	N
	8:7	0x00	green2_gain_analog_gain_1 Analog gain 1 = { (bit [8] + 1) * (bit [7] + 1) * initial gain }	Y	N
	6:0	0x40	green2_gain_initial_gain Initial gain = bits [6:0] * 1/64.	Y	N
12382 R0x305E	15:0	0x1040	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.				
12394 R0x306A	15:6	X	Reserved		
	5	RO	Reserved		
	4	0x0000	datapath_status_line_length_mismatch A fatal error occurred because the line length of the pixel data that the MIPI serializer expected to transmit did not match the line length set by X_OUTPUT_SIZE. The most likely reason for this error is that the clocking is configured incorrectly or that some register values that should remain static were changed while the sensor was in its streaming system state.	N	N

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
1	3	0x0000	datapath_status_frameover A fatal error occurred because a new frame started before the current frame had completed. The usual reason for this is that the Y_OUTPUT_SIZE has been set too large so that the sensor output is still padding the frame with rows of undefined pixel data when the next frame starts. An alternative reason is that clock ratio between the VT clock domain and the OP clock domain does not allow sufficient time for the OP domain to complete a frame before the next frame starts. The first step in avoiding this error is to set Y_OUTPUT_SIZE so that it matches the number of rows generated from the pixel array (Y_ADDR_END - Y_ADDR_START + 1, taking into account any subsampling/binning and any scaling). If the error remains, the next step is to increase FRAME_LENGTH_LINES.	N	N
	2	0x0000	datapath_status_lineover A fatal error occurred because the sensor output was unable to generate a full line of pixel data in the time budget provided by the setting of LINE_LENGTH_PCK and the vt_pix_clk period. The usual reason for this is that the X_OUTPUT_SIZE has been set too large so that the sensor output is still padding the row with undefined pixel data when the next row starts. An alternative reason is that clock ratio between the VT clock domain and the OP clock domain does not allow sufficient time for the OP domain to complete a row before the next row starts. The first step in avoiding this error is to set X_OUTPUT_SIZE so that it matches the number of pixels generated from the pixel array (X_ADDR_END - X_ADDR_START + 1, taking into account any subsampling/binning and any scaling). If the error remains, the next step is to increase LINE_LENGTH_PCK.	N	N
	1	0x0000	datapath_status_fifo_overflow A fatal error occurred because the output FIFO overflowed. The FIFO is sized to accommodate a full-length line from the pixel array, so this error can only occur when X_OUTPUT_SIZE is unnecessarily large, or when the ratio between the VT and OP clock domains has been set incorrectly. The first step in avoiding this error is to set X_OUTPUT_SIZE so that it matches the number of rows generated from the pixel array (X_ADDR_END - X_ADDR_START + 1, taking into account any subsampling/binning and any scaling).	N	N
	0	0x0000	datapath_status_fifo_underflow This fatal error condition is flagged if the output FIFO detects a data underflow.	N	N
	This register flags fatal error conditions associated with incorrect configuration of the sensor. Once any bit in this register has been set, behavior of the sensor is UNDEFINED and a reset may be required to restore correct operation. All bits are cleared automatically on the transition from the software standby system state to the streaming system state.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12398 R0x306E	15:0	0x9080	datapath_select (R/W)		
	15:13	0x04	datapath_select_slew_rate_ctrl_parallel Selects the slew (edge) rate for the DOUT[9:0], SHUTTER, FRAME_VALID, LINE_VALID and FLASH outputs. Only affects SHUTTER and FLASH outputs when parallel data output is disabled. The value 7 results in the fastest edge rates on these signals. Slowing down the edge rate can reduce ringing and electromagnetic emissions.	N	N
	12:10	0x04	datapath_select_slew_rate_ctrl_pixclk Selects the slew (edge) rate for the PIXCLK output. Has no effect when parallel data output is disabled. The value 7 results in the fastest edge rates on this signal. Slowing down the edge rate can reduce ringing and electromagnetic emissions.	N	N
	9:8	X	Reserved		
	7	RO	datapath_select_profile SMIA Profile Mode: 0: Profile 0 1: Profile 1/2 (this bit is read-only)	N	N
	6	X	Reserved		
	5	0x00	Reserved		
	4	0x00	datapath_select_true_bayer Enables true Bayer scaling mode.	N	N
	3:2	X	Reserved		
	1:0	0x00	datapath_select_special_line_valid 00: Normal behavior of LINE_VALID 01: LINE_VALID is driven continuously (continue generating LINE_VALID during vertical blanking) 10: LINE_VALID is driven continuously as LINE_VALID XOR FRAME_VALID	N	N
12400 R0x3070	15:0	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 : Walking 1s test pattern (10-bit) 257: Walking 1s test pattern (8-bit) other = Reserved.				
12402 R0x3072	15:0	0x0000	test_data_red_ (R/W)	N	Y
	The value for red pixels in the Bayer data used for the solid color test pattern and the test cursors.				
12404 R0x3074	15:0	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 R0x3076	15:0	0x0000	test_data_blue_ (R/W)	N	Y
	The value for blue pixels in the bByer data used for the solid color test pattern and the test cursors.				
12408 R0x3078	15:0	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.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12410 R0x307A	15:0	0x0000	test_raw_mode (R/W)		
	7:2	X	Reserved		
	1	0x00	Reserved		
	0	0x00	Reserved		
12448 R0x30A0	15:0	0x0001	x_even_inc_ (RO)	N	N
	Read-only.				
12450 R0x30A2	15:0	0x0001	x_odd_inc_ (R/W)	Y	YM
	This register field is an alias of R0x3040[8:6]				
12452 R0x30A4	15:0	0x0001	y_even_inc_ (RO)	N	N
	Read-only.				
12454 R0x30A6	15:0	0x0001	y_odd_inc_ (R/W)	Y	YM
	This register field is an alias of R0x3040[5:0]				
12638 R0x315E	15:0	0x0000	global_seq_trigger (R/W)		
	15:10	X	Reserved		
	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
	7:6	X	Reserved		
	5:4	0x00	global_seq_trigger_global_scale This field is used as the stepsize for duration of integration time/shutter starting from end of row reset phase of Global reset. The field is decoded as 0: 512 1: 2048 2: 128 3: 32 The GRR integration time is determined as: (global_read_start - global_read_end)*global_scale_factor/vt_pix_clk_freq_mhz where vt_pix_clk_freq_mhz is the array clock of the sensor. The scale factor ranging from 32 to 2048 enables up to a 64x scale on the integration time.	N	N
	3	X	Reserved		
	2	0x00	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
	1	0x00	global_seq_trigger_global_bulb 0: The integration of the GRR is determined by the global_rst_end, global_shutter_start, and global_read_start registers. 1: The integration is controlled by holding the trigger to '1' and finished by setting the trigger to '0'. This can be done using the global_seq_trigger_global_trigger bit or by configuring an external IO.	N	Y
	0	0x00	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

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12640 R0x3160	15:0	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 * global_scale / vt_pix_clk_freq_mhz.				
12642 R0x3162	15:0	0x00A8	global_shutter_start (R/W)	N	N
	Bits 15-0 of a 24-bit value which 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*global_scale / vt_pix_clk_freq_mhz timed from the end of row that was in progress when the global reset sequence was triggered.				
12644 R0x3164	15:0	0x0000	global_shutter_start2 (R/W)	N	N
	Bits 31-16of a 32-bit value which 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* global_scale / vt_pix_clk_freq_mhz timed from the end of row that was in progress when the global reset sequence was triggered.				
12646 R0x3166	15:0	0x00B8	global_read_start (R/W)	N	N
	Bits 15-0 of a 32-bit value which 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 time of N *global_scale / vt_pix_clk_freq_mhz. The integration time is given by (global_read_start-global_rst_end)* global_scale / vt_pix_clk_freq_mhz.				
12648 R0x3168	15:0	0x0000	global_read_start2 (R/W)	N	N
	Bits 23-16 of a 32-bit value which 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 * global_scale/ vt_pix_clk_freq_mhz. The integration time is given by (global_read_start-global_rst_end) * global_scale/ vt_pix_clk_freq_mhz.				
12704 R0x31A0	15:0	0x0101	serial_format_descriptor_0 (RO)	N	N
	The value of this descriptor indicates that a single-lane CCP2 interface is available on this device.				
12706 R0x31A2	15:0	0x0201	serial_format_descriptor_1 (RO)	N	N
	The value of this descriptor indicates that a single-lane MIPI interface is available on this device.				
12708 R0x31A4	15:0	0x0202	serial_format_descriptor_2 (RO)	N	N
	The value of this descriptor indicates that a dual-lane MIPI interface is available on this device.				
12710 R0x31A6	15:0	0x0301	serial_format_descriptor_3 (RO)	N	N
	The value of this descriptor indicates that a single-lane HiSPi interface is available on this device.				
12712 R0x31A8	15:0	0x0302	serial_format_descriptor_4 (RO)	N	N
	The value of this descriptor indicates that a 2-lane HiSPi interface is available on this device.				
12714 R0x31AA	15:0	0x0304	serial_format_descriptor_5 (RO)	N	N
	The value of this descriptor indicates that a 4-lane HiSPi interface is available on this device.				
12716 R0x31AC	15:0	0x0000	serial_format_descriptor_6 (RO)	N	N
	Not used.				
12720 R0x31B0	15:0	0x0063	frame_preamble (R/W)	N	N
	This timing value, expressed in op_pix_clk periods, must be large enough to allow the MIPI wakeup and start-of-frame short packet to be transmitted prior to the start of a frame of pixel data. The default value should be correct for most applications. Too small a value will result in an INSUFFICIENT_FRAME_PREAMBLE error being flagged in the DATAPATH_STATUS register.				
12722 R0x31B2	15:0	0x0039	line_preamble (R/W)	N	N
	This timing value, expressed in op_pix_clk periods, must be large enough to allow the MIPI long packet header to be transmitted prior to the start of a line of pixel data. The default value should be correct for most applications. Too small a value will result in an INSUFFICIENT_LINE_PREAMBLE error being flagged in the DATAPATH_STATUS register.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12724 R0x31B4	15:0	0x0D57	mipi_timing_0 (R/W)		
	15:12	RO	mipi_timing_0_reserved Reserved. Read as 0	N	N
	11:8	0x0D	mipi_timing_0_t_hs_zero Time, in op_pix_clk periods, to drive HS-0 before the sync sequence	N	N
	7:4	0x05	mipi_timing_0_t_hs_trail Time, in op_pix_clk periods, to drive flipped differential state after last payload data bit of an HS transmission burst	N	N
	3:0	0x07	mipi_timing_0_t_clk_trail Time, in op_pix_clk periods, to drive HS differentialstate after last payload clock bit of an HS transmission burst	N	N
12726 R0x31B6	15:0	0x0B10	mipi_timing_1 (R/W)		
	15:14	RO	mipi_timing_1_reserved_1 Reserved. Read as 0	N	N
	13:8	0x0B	mipi_timing_1_t_hs_exit Time, in op_pix_clk periods, to drive LP-11 after HS burst	N	N
	7:6	RO	mipi_timing_1_reserved_0 Reserved. Read as 0	N	N
	5:0	0x10	mipi_timing_1_t_clk_zero Minimum time, in op_pix_clk periods, to drive HS-0 on clock lane prior to starting clock	N	N
12728 R0x31B8	15:0	0x010D	mipi_timing_2 (R/W)		
	15:14	RO	mipi_timing_2_reserved_1 Reserved. Read as 0	N	N
	13:8	0x01	mipi_timing_2_t_clk_pre Time, in op_pix_clk periods, to drive the HS clock before any data lane might start up	N	N
	7:6	RO	mipi_timing_2_reserved_0 Reserved. Read as 0	N	N
	5:0	0x0D	mipi_timing_2_t_clk_post Time, in op_pix_clk periods, to drive the HS clock after the data lane has gone into low-power mode	N	N
12730 R0x31BA	15:0	0x050D	mipi_timing_3 (R/W)		
	15:14	RO	mipi_timing_3_reserved_1 Reserved. Read as 0	N	N
	13:8	0x05	mipi_timing_3_t_lpx Time, in op_pix_clk periods, of any low-power state period	N	N
	7	RO	mipi_timing_3_reserved_0 Reserved. Read as 0	N	N
	6:0	0x0D	mipi_timing_3_t_wake_up Time to recover from ultra low-power mode (ULPM). ULPM is exited by applying a mark state for (8192) * T_WAKE_UP * op_pix_clk	N	N
12732 R0x31BC	15:0	0x000B	mipi_timing_4 (R/W)		
	15:7	RO	mipi_timing_4_reserved Reserved. Read as 0	N	N
	6:0	0x0B	mipi_timing_4_t_init Initialization time when first entering stop state (LP-11) after power up or reset. LP-11 is transmitted for a minimum of (1024) * T_INIT * op_pix_clk.	N	N

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12736 R0x31C0	15:0	0x0000	hispi_timing (R/W)		
	15	X	Reserved		
	14:12	0x00	hispi_timing_clock_del Delay applied to the clock lane in 1/8 unit interval (UI) steps.	N	N
	11:9	0x00	hispi_timing_data3_del Delay applied to Data Lane 3 in 1/8 unit interval (UI) steps.	N	N
	8:6	0x00	hispi_timing_data2_del Delay applied to Data Lane 2 in 1/8 unit interval (UI) steps.	N	N
	5:3	0x00	hispi_timing_data1_del Delay applied to Data Lane 1 in 1/8 unit interval (UI) steps.	N	N
	2:0	0x00	hispi_timing_data0_del Delay applied to Data Lane 0 in 1/8 unit interval (UI) steps.	N	N
	Within the HiSPi PHY there is a DLL connected to the clock lane and each data lane, which acts as a control master for the output delay buffers. This additional delay allows the user to increase the setup or hold time at the receiver circuits and can be used to compensate for skew introduced in PCB design. If the DLL timing adjustment is not required, the data and clock lane delay settings should be set to a default code of 0x000 to reduce jitter, skew, and power dissipation.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12742 R0x31C6	15:0	0x8000	hispi_control_status (R/W)		
	15	0x01	Reserved		
	14	0x00	Reserved		
	13	X	Reserved		
	12	0x00	Reserved		
	11:10	0x00	mode_select Will select the HiSPi output protocol: 00: Streaming S 01: Packetized SP	N	N
	9	0x00	Reserved		
	8	0x00	Reserved		
	7	0x00	test_enable When asserted, the test pattern is output through the HiSPi PHY interface.	N	N
	6:4	0x00	test_mode Define the test mode to be applied if the test mode is enabled. 0: Transmit a constant 0 on all enabled data lanes. 1: Transmit a constant 1 on all enabled data lanes. 2: Transmit a square wave at half the serial data rate on all enabled data lanes. 3:Transmit a square wave at the pixel rate on all enabled data lanes. 4: Transmit a continuous sequence of pseudorandom data, with no SAV code, copied on all enabled data lanes. 5: Replace data from the sensor with a known sequence (SMIA-defined PN9) copied on all enabled data lanes.	N	N
	3	0x00	blanking_data_enable 0: The default pattern (constant 1) is output during horizontal and vertical blanking periods 1: The pattern defined by the blanking_data input is output during horizontal and vertical blanking periods NOTE: For the HiSPi Streaming protocol the blanking data pattern cannot be changed from the default.	N	N
	2	0x00	streaming_mode Global enable for the SLVS interface.	N	N
	1	0x00	output_msb_first Output MSB first	N	N
	0	0x00	vert_left_bar_en An optional filler code of "1" may be padded after the sync code. When filler codes are enabled, the receiver must window the received image to eliminate first 4 data words (columns per PHYs).	N	N
	HiSPi Control status for the output PHY.				
12776 R0x31E8	15:0	0x0000	horizontal_cursor_position_ (R/W)	N	N
	Specify the start row for the test cursor.				
12778 R0x31EA	15:0	0x0000	vertical_cursor_position_ (R/W)	N	N
	Specify the start column for the test cursor.				
12780 R0x31EC	15:0	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 R0x31EE	15:0	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.				

Table 6: **Manufacturer-Specific (continued)**
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12786 R0x31F2	15:0	0x6E6C	i2c_ids_mipi_default (R/W)	N	N
	Programmable two-wire serial interface slave addresses for MIPI operation.				
12796 R0x31FC	15:0	0x3020	i2c_ids (R/W)	N	N
	Two-wire serial interface addresses.				
13824 R0x3600	15:0	0x0000	p_gr_p0q0 (R/W)	N	N
	P0 coefficient for Q0 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13826 R0x3602	15:0	0x0000	p_gr_p0q1 (R/W)	N	N
	P0 coefficient for Q1 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13828 R0x3604	15:0	0x0000	p_gr_p0q2 (R/W)	N	N
	P0 coefficient for Q2 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13830 R0x3606	15:0	0x0000	p_gr_p0q3 (R/W)	N	N
	P0 coefficient for Q3 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13832 R0x3608	15:0	0x0000	p_gr_p0q4 (R/W)	N	N
	P0 coefficient for Q4 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13834 R0x360A	15:0	0x0000	p_rd_p0q0 (R/W)	N	N
	P0 coefficient for Q0 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13836 R0x360C	15:0	0x0000	p_rd_p0q1 (R/W)	N	N
	P0 coefficient for Q1 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13838 R0x360E	15:0	0x0000	p_rd_p0q2 (R/W)	N	N
	P0 coefficient for Q2 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13840 R0x3610	15:0	0x0000	p_rd_p0q3 (R/W)	N	N
	P0 coefficient for Q3 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13842 R0x3612	15:0	0x0000	p_rd_p0q4 (R/W)	N	N
	P0 coefficient for Q4 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13844 R0x3614	15:0	0x0000	p_bl_p0q0 (R/W)	N	N
	P0 coefficient for Q0 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13846 R0x3616	15:0	0x0000	p_bl_p0q1 (R/W)	N	N
	P0 coefficient for Q1 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13848 R0x3618	15:0	0x0000	p_bl_p0q2 (R/W)	N	N
	P0 coefficient for Q2 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13850 R0x361A	15:0	0x0000	p_bl_p0q3 (R/W)	N	N
	P0 coefficient for Q3 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
13852 R0x361C	15:0	0x0000	p_bl_p0q4 (R/W)	N	N
	P0 coefficient for Q4 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13854 R0x361E	15:0	0x0000	p_gb_p0q0 (R/W)	N	N
	P0 coefficient for Q0 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13856 R0x3620	15:0	0x0000	p_gb_p0q1 (R/W)	N	N
	P0 coefficient for Q1 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13858 R0x3622	15:0	0x0000	p_gb_p0q2 (R/W)	N	N
	P0 coefficient for Q2 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13860 R0x3624	15:0	0x0000	p_gb_p0q3 (R/W)	N	N
	P0 coefficient for Q3 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13862 R0x3626	15:0	0x0000	p_gb_p0q4 (R/W)	N	N
	P0 coefficient for Q4 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13888 R0x3640	15:0	0x0000	p_gr_p1q0 (R/W)	N	N
	P1 coefficient for Q0 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13890 R0x3642	15:0	0x0000	p_gr_p1q1 (R/W)	N	N
	P1 coefficient for Q1 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13892 R0x3644	15:0	0x0000	p_gr_p1q2 (R/W)	N	N
	P1 coefficient for Q2 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13894 R0x3646	15:0	0x0000	p_gr_p1q3 (R/W)	N	N
	P1 coefficient for Q3 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13896 R0x3648	15:0	0x0000	p_gr_p1q4 (R/W)	N	N
	P1 coefficient for Q4 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13898 R0x364A	15:0	0x0000	p_rd_p1q0 (R/W)	N	N
	P1 coefficient for Q0 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13900 R0x364C	15:0	0x0000	p_rd_p1q1 (R/W)	N	N
	P1 coefficient for Q1 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13902 R0x364E	15:0	0x0000	p_rd_p1q2 (R/W)	N	N
	P1 coefficient for Q2 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13904 R0x3650	15:0	0x0000	p_rd_p1q3 (R/W)	N	N
	P1 coefficient for Q3 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
13906 R0x3652	15:0	0x0000	p_rd_p1q4 (R/W)	N	N
	P1 coefficient for Q4 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13908 R0x3654	15:0	0x0000	p_bl_p1q0 (R/W)	N	N
	P1 coefficient for Q0 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13910 R0x3656	15:0	0x0000	p_bl_p1q1 (R/W)	N	N
	P1 coefficient for Q1 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13912 R0x3658	15:0	0x0000	p_bl_p1q2 (R/W)	N	N
	P1 coefficient for Q2 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13914 R0x365A	15:0	0x0000	p_bl_p1q3 (R/W)	N	N
	P1 coefficient for Q3 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13916 R0x365C	15:0	0x0000	p_bl_p1q4 (R/W)	N	N
	P1 coefficient for Q4 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13918 R0x365E	15:0	0x0000	p_gb_p1q0 (R/W)	N	N
	P1 coefficient for Q0 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13920 R0x3660	15:0	0x0000	p_gb_p1q1 (R/W)	N	N
	P1 coefficient for Q1 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13922 R0x3662	15:0	0x0000	p_gb_p1q2 (R/W)	N	N
	P1 coefficient for Q2 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13924 R0x3664	15:0	0x0000	p_gb_p1q3 (R/W)	N	N
	P1 coefficient for Q3 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13926 R0x3666	15:0	0x0000	p_gb_p1q4 (R/W)	N	N
	P1 coefficient for Q4 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13952 R0x3680	15:0	0x0000	p_gr_p2q0 (R/W)	N	N
	P2 coefficient for Q0 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13954 R0x3682	15:0	0x0000	p_gr_p2q1 (R/W)	N	N
	P2 coefficient for Q1 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13956 R0x3684	15:0	0x0000	p_gr_p2q2 (R/W)	N	N
	P2 coefficient for Q2 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13958 R0x3686	15:0	0x0000	p_gr_p2q3 (R/W)	N	N
	P2 coefficient for Q3 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
13960 R0x3688	15:0	0x0000	p_gr_p2q4 (R/W)	N	N
	P2 coefficient for Q4 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
13962 R0x368A	15:0	0x0000	p_rd_p2q0 (R/W)	N	N
	P2 coefficient for Q0 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13964 R0x368C	15:0	0x0000	p_rd_p2q1 (R/W)	N	N
	P2 coefficient for Q1 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13966 R0x368E	15:0	0x0000	p_rd_p2q2 (R/W)	N	N
	P2 coefficient for Q2 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13968 R0x3690	15:0	0x0000	p_rd_p2q3 (R/W)	N	N
	P2 coefficient for Q3 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13970 R0x3692	15:0	0x0000	p_rd_p2q4 (R/W)	N	N
	P2 coefficient for Q4 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
13972 R0x3694	15:0	0x0000	p_bl_p2q0 (R/W)	N	N
	P2 coefficient for Q0 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13974 R0x3696	15:0	0x0000	p_bl_p2q1 (R/W)	N	N
	P2 coefficient for Q1 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13976 R0x3698	15:0	0x0000	p_bl_p2q2 (R/W)	N	N
	P2 coefficient for Q2 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13978 R0x369A	15:0	0x0000	p_bl_p2q3 (R/W)	N	N
	P2 coefficient for Q3 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13980 R0x369C	15:0	0x0000	p_bl_p2q4 (R/W)	N	N
	P2 coefficient for Q4 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
13982 R0x369E	15:0	0x0000	p_gb_p2q0 (R/W)	N	N
	P2 coefficient for Q0 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13984 R0x36A0	15:0	0x0000	p_gb_p2q1 (R/W)	N	N
	P2 coefficient for Q1 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13986 R0x36A2	15:0	0x0000	p_gb_p2q2 (R/W)	N	N
	P2 coefficient for Q2 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
13988 R0x36A4	15:0	0x0000	p_gb_p2q3 (R/W)	N	N
	P2 coefficient for Q3 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
13990 R0x36A6	15:0	0x0000	p_gb_p2q4 (R/W)	N	N
	P2 coefficient for Q4 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
14016 R0x36C0	15:0	0x0000	p_gr_p3q0 (R/W)	N	N
	P3 coefficient for Q0 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14018 R0x36C2	15:0	0x0000	p_gr_p3q1 (R/W)	N	N
	P3 coefficient for Q1 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14020 R0x36C4	15:0	0x0000	p_gr_p3q2 (R/W)	N	N
	P3 coefficient for Q2 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14022 R0x36C6	15:0	0x0000	p_gr_p3q3 (R/W)	N	N
	P3 coefficient for Q3 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14024 R0x36C8	15:0	0x0000	p_gr_p3q4 (R/W)	N	N
	P3 coefficient for Q4 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14026 R0x36CA	15:0	0x0000	p_rd_p3q0 (R/W)	N	N
	P3 coefficient for Q0 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
14028 R0x36CC	15:0	0x0000	p_rd_p3q1 (R/W)	N	N
	P3 coefficient for Q1 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
14030 R0x36CE	15:0	0x0000	p_rd_p3q2 (R/W)	N	N
	P3 coefficient for Q2 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
14032 R0x36D0	15:0	0x0000	p_rd_p3q3 (R/W)	N	N
	P3 coefficient for Q3 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
14034 R0x36D2	15:0	0x0000	p_rd_p3q4 (R/W)	N	N
	P3 coefficient for Q4 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
14036 R0x36D4	15:0	0x0000	p_bl_p3q0 (R/W)	N	N
	P3 coefficient for Q0 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
14038 R0x36D6	15:0	0x0000	p_bl_p3q1 (R/W)	N	N
	P3 coefficient for Q1 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
14040 R0x36D8	15:0	0x0000	p_bl_p3q2 (R/W)	N	N
	P3 coefficient for Q2 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
14042 R0x36DA	15:0	0x0000	p_bl_p3q3 (R/W)	N	N
	P3 coefficient for Q3 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14044 R0x36DC	15:0	0x0000	p_bl_p3q4 (R/W)	N	N
	P3 coefficient for Q4 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.				
14046 R0x36DE	15:0	0x0000	p_gb_p3q0 (R/W)	N	N
	P3 coefficient for Q0 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
14048 R0x36E0	15:0	0x0000	p_gb_p3q1 (R/W)	N	N
	P3 coefficient for Q1 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
14050 R0x36E2	15:0	0x0000	p_gb_p3q2 (R/W)	N	N
	P3 coefficient for Q2 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
14052 R0x36E4	15:0	0x0000	p_gb_p3q3 (R/W)	N	N
	P3 coefficient for Q3 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
14054 R0x36E6	15:0	0x0000	p_gb_p3q4 (R/W)	N	N
	P3 coefficient for Q4 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.				
14080 R0x3700	15:0	0x0000	p_gr_p4q0 (R/W)	N	N
	P4 coefficient for Q0 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14082 R0x3702	15:0	0x0000	p_gr_p4q1 (R/W)	N	N
	P4 coefficient for Q1 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14084 R0x3704	15:0	0x0000	p_gr_p4q2 (R/W)	N	N
	P4 coefficient for Q2 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14086 R0x3706	15:0	0x0000	p_gr_p4q3 (R/W)	N	N
	P4 coefficient for Q3 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14088 R0x3708	15:0	0x0000	p_gr_p4q4 (R/W)	N	N
	P4 coefficient for Q4 for Gr. P_GR_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gr pixels.				
14090 R0x370A	15:0	0x0000	p_rd_p4q0 (R/W)	N	N
	P4 coefficient for Q0 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
14092 R0x370C	15:0	0x0000	p_rd_p4q1 (R/W)	N	N
	P4 coefficient for Q1 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
14094 R0x370E	15:0	0x0000	p_rd_p4q2 (R/W)	N	N
	P4 coefficient for Q2 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				
14096 R0x3710	15:0	0x0000	p_rd_p4q3 (R/W)	N	N
	P4 coefficient for Q3 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.				

Aptina reserves the right to change products or specifications without notice.

Table 6: Manufacturer-Specific (continued)
R/W (Read or Write) bit; RO (Read Only) bit

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14098 R0x3712	15:0	0x0000	p_rd_p4q4 (R/W)	N	N
			P4 coefficient for Q4 for Rd. P_RD_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Rd pixels.		
14100 R0x3714	15:0	0x0000	p_bl_p4q0 (R/W)	N	N
			P4 coefficient for Q0 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.		
14102 R0x3716	15:0	0x0000	p_bl_p4q1 (R/W)	N	N
			P4 coefficient for Q1 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.		
14104 R0x3718	15:0	0x0000	p_bl_p4q2 (R/W)	N	N
			P4 coefficient for Q2 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.		
14106 R0x371A	15:0	0x0000	p_bl_p4q3 (R/W)	N	N
			P4 coefficient for Q3 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.		
14108 R0x371C	15:0	0x0000	p_bl_p4q4 (R/W)	N	N
			P4 coefficient for Q4 for Bl. P_BL_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Bl pixels.		
14110 R0x371E	15:0	0x0000	p_gb_p4q0 (R/W)	N	N
			P4 coefficient for Q0 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.		
14112 R0x3720	15:0	0x0000	p_gb_p4q1 (R/W)	N	N
			P4 coefficient for Q1 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.		
14114 R0x3722	15:0	0x0000	p_gb_p4q2 (R/W)	N	N
			P4 coefficient for Q2 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.		
14116 R0x3724	15:0	0x0000	p_gb_p4q3 (R/W)	N	N
			P4 coefficient for Q3 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.		
14118 R0x3726	15:0	0x0000	p_gb_p4q4 (R/W)	N	N
			P4 coefficient for Q4 for Gb. P_GB_PpQq registers are read successively when the row polynomial (Q) coefficients are calculated during the horizontal blanking period before a row containing Gb pixels.		
14208 R0x3780	15:0	0x0000	poly_sc_enable (R/W)		
	15	0x00	poly_sc_en Turn on shading correction.	N	N
	14:0	X	Reserved		
			When poly_sc_en bit is set, poly_sc_enable will generate polynomial function and correct stream of pixels. When poly_sc_en bit is not set, poly_sc_enable will bypass data.		
14210 R0x3782	15:0	0x0000	poly_origin_c (R/W)	N	N
			Origin of polynomial function: applied as offset to X (col) coordinate of pixel.		
14212 R0x3784	15:0	0x0000	poly_origin_r (R/W)	N	N
			Origin of polynomial function: applied as offset to Y (row) coordinate of pixel.		

Revision History

Rev. A 7/14/10

- Initial release

10 Eunos Road 8 13-40, Singapore Post Center, Singapore 408600 prodmktg@aptina.com www.aptina.com

This data sheet contains minimum and maximum limits specified over the power supply and temperature range set forth herein. Although considered final, these specifications are subject to change, as further product development and data characterization sometimes occur.