AND9232, Rev. B

For more information, please visit www.onsemi.com

AR0542 Register Reference

Table of Contents

List of Tables

Registers

The AR0542 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. The remainder of this section describes these registers in detail.

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)
0x3000–0x3FFF	Manufacturer-specific registers (read-only and read-write dynamic registers)

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 AR0542 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 AR0542 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) has only 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 one 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 2 bytes of address space are aligned to even 16-bit addresses, and registers that occupy 4 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
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 AR0542 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 "Frame Sync'd" column shows which registers or register fields are double-buffered in this way.

Using grouped_parameter_hold

Register grouped_parameter_hold (R0x0104) can be used to inhibit transfers from the pending to the live registers. When the AR0542 is in streaming mode, this register should be written to "1" before making changes to any group of registers where a set of changes is required to take effect simultaneously. When this register is written 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 3.
- 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 Map

Table 3 shows the locations used within the address space. Locations that are not shown in the table are reserved for future use; they return 0x00 on read, but should not be read from or written to maintain compatibility with future designs. Locations that are shown as "Reserved" should not be accessed. The default read values of these registers are subject to change.

Caution **The effect of writing to reserved registers is undefined and may include the possibility of causing permanent electrical damage to the sensor.**

Table 3 through Table 5 on page 15 list sensor registers and their default values. Table 6 on page 35 through Table 8 on page 45 list sensor registers and their descriptions.

Register List and Default Values

SMIA Configuration Register List

Table 1: SMIA Configuration Register List
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)	chip_version_reg	dddd dddd dddd dddd	18432 (0x4800)
R2 (R0x0002)	revision_number	dddd dddd	112 (0x70)
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 00dd dddd dddd	42 (0x002A)
R64 (R0x0040)	frame_format_model_type	???? ????	1 (0x01)
R65 (R0x0041)	frame_format_model_subtype	???? ????	18 (0x12)
R66 (R0x0042)	frame_format_descriptor_0	???? ???? ???? ????	23072 (0x5A20)
R68 (R0x0044)	frame_format_descriptor_1	???? ???? ???? ????	4098 (0x1002)
R70 (R0x0046)	frame_format_descriptor_2	???? ???? ???? ????	22424 (0x5798)
R72 (R0x0048)	frame_format_descriptor_3	???? ???? ???? ????	0 (0x0000)
R74 (R0x004A)	frame_format_descriptor_4	???? ???? ???? ????	0 (0x0000)

Table 1: SMIA Configuration Register List (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)
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)
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	???? ???? ???? ????	511 (0x01FF)
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	???? ????	3 (0x03)
R194 (R0x00C2)	data_format_descriptor_0	???? ???? ???? ????	2570 (0x0A0A)
R196 (R0x00C4)	data_format_descriptor_1	???? ???? ???? ????	2056 (0x0808)
R198 (R0x00C6)	data_format_descriptor_2	???? ???? ???? ????	2568 (0x0A08)

Table 1: SMIA Configuration Register List (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)
R200 (R0x00C8)	data_format_descriptor_3	???? ???? ???? ????	0 (0x0000)
R202 (R0x00CA)	data_format_descriptor_4	???? ???? ???? ????	0 (0x0000)
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)
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	2570 (0x0A0A)
R288 (R0x0120)	gain_mode	0000 000d	0 (0x00)
R512 (R0x0200)	fine_integration_time	dddd dddd dddd ddd0	718 (0x02CE)
R514 (R0x0202)	coarse_integration_time	dddd dddd dddd dddd	16 (0x0010)
R516 (R0x0204)	analogue_gain_code_global	0000 000d dddd dddd	20 (0x0014)
R518 (R0x0206)	analogue_gain_code_greenr	0000 000d dddd dddd	20 (0x0014)
R520 (R0x0208)	analogue_gain_code_red	0000 000d dddd dddd	20 (0x0014)
R522 (R0x020A)	analogue_gain_code_blue	0000 000d dddd dddd	20 (0x0014)
R524 (R0x020C)	analogue_gain_code_greenb	0000 000d dddd dddd	20 (0x0014)
R526 (R0x020E)	digital_gain_greenr	0000 dddd 0000 0000	256 (0x0100)
R528 (R0x0210)	digital_gain_red	0000 dddd 0000 0000	256 (0x0100)
R530 (R0x0212)	digital_gain_blue	0000 dddd 0000 0000	256 (0x0100)
R532 (R0x0214)	digital_gain_greenb	0000 dddd 0000 0000	256 (0x0100)

Table 1: SMIA Configuration Register List (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)
R768 (R0x0300)	vt_pix_clk_div	0000 0000 000d dddd	5 (0x0005)
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	70 (0x0046)
R776 (R0x0308)	op_pix_clk_div	0000 0000 000d dddd	10 (0x000A)
R778 (R0x030A)	op_sys_clk_div	0000 0000 000d dddd	1 (0x0001)
R832 (R0x0340)	frame_length_lines	dddd dddd dddd dddd	2021 (0x07E5)
R834 (R0x0342)	line_length_pck	dddd dddd dddd ddd0	5238 (0x1476)
R836 (R0x0344)	x_addr_start	0000 dddd dddd dddd	8 (0x0008)
R838 (R0x0346)	y_addr_start	0000 dddd dddd dddd	8 (0x0008)
R840 (R0x0348)	x_addr_end	0000 dddd dddd dddd	2599 (0x0A27)
R842 (R0x034A)	y_addr_end	0000 dddd dddd dddd	1951 (0x079F)
R844 (R0x034C)	x_output_size	0000 dddd dddd ddd0	2592 (0x0A20)
R846 (R0x034E)	y_output_size	0000 dddd dddd ddd0	1944 (0x0798)
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 00dd 0000 0ddd	0 (0x0000)

Table 1: SMIA Configuration Register List (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)
R1538 (R0x0602)	test_data_red	0000 00dd dddd dddd	0 (0x0000)
R1540 (R0x0604)	test_data_greenr	0000 00dd dddd dddd	0 (0x0000)
R1542 (R0x0606)	test_data_blue	0000 00dd dddd dddd	0 (0x0000)
R1544 (R0x0608)	test_data_greenb	0000 00dd 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)
R1552 (R0x0610)	vertical_cursor_position	0000 dddd dddd dddd	0 (0x0000)

SMIA Parameter Limit Register List

Table 2: SMIA Parameter Limits Register List
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	718 (0x02CE)
R4106 (R0x100A)	fine_integration_time_max_margin	dddd dddd dddd dddd	482 (0x01E2)
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	???? ???? ???? ???? ???? ???? ???? ????	1073741824 (0x40000000)
R4356 (R0x1104)	max_ext_clk_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1115684864 (0x42800000)

Table 2: SMIA Parameter Limits Register List (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)
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	???? ???? ???? ???? ???? ???? ???? ????	1082130432 (0x40800000)
R4368 (R0x1110)	max_pll_ip_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1103101952 (0x41C00000)
R4372 (R0x1114)	min_pll_multiplier	???? ???? ???? ????	32 (0x0020)
R4374 (R0x1116)	max_pll_multiplier	???? ???? ???? ????	384 (0x0180)
R4376 (R0x1118)	min_pll_op_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1136656384 (0x43C00000)
R4380 (R0x111C)	max_pll_op_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1146224640 (0x44520000)
R4384 (R0x1120)	min_vt_sys_clk_div	???? ???? ???? ????	1 (0x0001)
R4386 (R0x1122)	max_vt_sys_clk_div	???? ???? ???? ????	16 (0x0010)
R4388 (R0x1124)	min_vt_sys_clk_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1103101952 (0x41C00000)
R4392 (R0x1128)	max_vt_sys_clk_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1148846080 (0x447A0000)
R4396 (R0x112C)	min_vt_pix_clk_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1083808154 (0x4099999A)
R4400 (R0x1130)	max_vt_pix_clk_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1121976320 (0x42E00000)
R4404 (R0x1134)	min_vt_pix_clk_div	???? ???? ???? ????	4 (0x0004)
R4406 (R0x1136)	max_vt_pix_clk_div	???? ???? ???? ????	16 (0x0010)
R4416 (R0x1140)	min_frame_length_lines	dddd dddd dddd dddd	79 (0x004F)
R4418 (R0x1142)	max_frame_length_lines	dddd dddd dddd dddd	65535 (0xFFFF)
R4420 (R0x1144)	min_line_length_pck	dddd dddd dddd dddd	1424 (0x0590)
R4422 (R0x1146)	max_line_length_pck	dddd dddd dddd dddd	65534 (0xFFFE)
R4424 (R0x1148)	min_line_blanking_pck	dddd dddd dddd dddd	1102 (0x044E)
R4426 (R0x114A)	min_frame_blanking_lines	dddd dddd dddd dddd	77 (0x004D)
R4448 (R0x1160)	min_op_sys_clk_div	???? ???? ???? ????	1 (0x0001)
R4450 (R0x1162)	max_op_sys_clk_div	???? ???? ???? ????	16 (0x0010)

Table 2: SMIA Parameter Limits Register List (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)
R4452 (R0x1164)	min_op_sys_clk_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1103101952 (0x41C00000)
R4456 (R0x1168)	max_op_sys_clk_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1146224640 (0x44520000)
R4460 (R0x116C)	min_op_pix_clk_div	???? ???? ???? ????	8 (0x0008)
R4462 (R0x116E)	max_op_pix_clk_div	???? ???? ???? ????	10 (0x000A)
R4464 (R0x1170)	min_op_pix_clk_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1075419546 (0x4019999A)
R4468 (R0x1174)	max_op_pix_clk_freq_mhz	???? ???? ???? ???? ???? ???? ???? ????	1118306304 (0x42A80000)
R4480 (R0x1180)	x_addr_min	???? ???? ???? ????	0 (0x0000)
R4482 (R0x1182)	y_addr_min	???? ???? ???? ????	0 (0x0000)
R4484 (R0x1184)	x_addr_max	???? ???? ???? ????	2607 (0x0A2F)
R4486 (R0x1186)	y_addr_max	???? ???? ???? ????	1959 (0x07A7)
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	???? ???? ???? ????	7 (0x0007)
R4608 (R0x1200)	scaling_capability	0000 0000 0000 00??	2 (0x0002)
R4612 (R0x1204)	scaler_m_min	???? ???? ???? ????	16 (0x0010)
R4614 (R0x1206)	scaler_m_max	???? ???? ???? ????	128 (0x0080)
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)

Table 2: SMIA Parameter Limits Register List (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)
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 Register List
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	18432 (0x4800)
R12290 (R0x3002)	y_addr_start_	0000 dddd dddd dddd	8 (0x0008)
R12292 (R0x3004)	x_addr_start_	0000 dddd dddd dddd	8 (0x0008)
R12294 (R0x3006)	y_addr_end_	0000 dddd dddd dddd	1951 (0x079F)
R12296 (R0x3008)	x_addr_end_	0000 dddd dddd dddd	2599 (0x0A27)
R12298 (R0x300A)	frame_length_lines_	dddd dddd dddd dddd	2021 (0x07E5)
R12300 (R0x300C)	line_length_pck_	dddd dddd dddd ddd0	5238 (0x1476)
R12304 (R0x3010)	fine_correction	0ddd dddd dddd dddd	160 (0x00A0)
R12306 (R0x3012)	coarse_integration_time_	dddd dddd dddd dddd	16 (0x0010)
R12308 (R0x3014)	fine_integration_time_	dddd dddd dddd ddd0	718 (0x02CE)
R12310 (R0x3016)	row_speed	0000 0ddd 0ddd 0ddd	273 (0x0111)
R12312 (R0x3018)	extra_delay	dddd dddd dddd ddd0	0 (0x0000)
R12314 (R0x301A)	reset_register	dd0d 0ddd dddd dddd	88 (0x0058)
R12316 (R0x301C)	mode_select_	0000 000d	0 (0x00)

Table 3: Manufacturer Specific Register List (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)
R12317 (R0x301D)	image_orientation_	0000 00dd	0 (0x00)
R12318 (R0x301E)	data_pedestal_	0000 00dd dddd dddd	42 (0x002A)
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 000d dddd dddd	20 (0x0014)
R12330 (R0x302A)	analogue_gain_code_greenr_	0000 000d dddd dddd	20 (0x0014)
R12332 (R0x302C)	analogue_gain_code_red_	0000 000d dddd dddd	20 (0x0014)
R12334 (R0x302E)	analogue_gain_code_blue_	0000 000d dddd dddd	20 (0x0014)
R12336 (R0x3030)	analogue_gain_code_greenb_	0000 000d dddd dddd	20 (0x0014)
R12338 (R0x3032)	digital_gain_greenr_	0000 dddd 0000 0000	256 (0x0100)
R12340 (R0x3034)	digital_gain_red_	0000 dddd 0000 0000	256 (0x0100)
R12342 (R0x3036)	digital_gain_blue_	0000 dddd 0000 0000	256 (0x0100)
R12344 (R0x3038)	digital_gain_greenb_	0000 dddd 0000 0000	256 (0x0100)
R12346 (R0x303A)	smia_version_	???? ????	10 (0x0A)
R12347 (R0x303B)	frame_count_	???? ????	255 (0xFF)
R12348 (R0x303C)	frame_status	0000 0000 0000 00??	0 (0x0000)
R12352 (R0x3040)	read_mode	dd0d dddd dddd dddd	65 (0x0041)
R12358 (R0x3046)	flash	??dd dddd dd0d dddd	1544 (0x0608)
R12360 (R0x3048)	flash_count	dddd dddd dddd dddd	8 (0x0008)
R12362 (R0x304A)	otpm_control	0000 0ddd 0??d d??d	0 (0x0000)
R12364 (R0x304C)	otpm_record	dddd dddd dddd dddd	512 (0x0200)

Table 3: Manufacturer Specific Register List (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)
R12366 (R0x304E)	otpm_status	0000 0??? ???? ????	0 (0x0000)
R12368 (R0x3050)	otpm_manual_control	dddd dddd 0??d 0??d	0 (0x0000)
R12370 (R0x3052)	otpm_config	dd00 000d dddd dddd	0 (0x0000)
R12372 (R0x3054)	otpm_expr	0000 dddd 0000 0dd0	0 (0x0000)
R12374 (R0x3056)	green1_gain	dddd dddd dddd dddd	4176 (0x1050)
R12376 (R0x3058)	blue_gain	dddd dddd dddd dddd	4176 (0x1050)
R12378 (R0x305A)	red_gain	dddd dddd dddd dddd	4176 (0x1050)
R12380 (R0x305C)	green2_gain	dddd dddd dddd dddd	4176 (0x1050)
R12382 (R0x305E)	global_gain	dddd dddd dddd dddd	4176 (0x1050)
R12394 (R0x306A)	datapath_status	0000 0000 00?d dddd	0 (0x0000)
R12398 (R0x306E)	datapath_select	dddd dd00 ?ddd 00dd	44160 (0xAC80)
R12400 (R0x3070)	test_pattern_mode_	0000 00dd 0000 0ddd	0 (0x0000)
R12402 (R0x3072)	test_data_red_	0000 00dd dddd dddd	0 (0x0000)
R12404 (R0x3074)	test_data_greenr_	0000 00dd dddd dddd	0 (0x0000)
R12406 (R0x3076)	test_data_blue_	0000 00dd dddd dddd	0 (0x0000)
R12408 (R0x3078)	test_data_greenb_	0000 00dd 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)
R12456 (R0x30A8)	calib_green1_asc1	0000 0ddd dddd dddd	1024 (0x0400)
R12458 (R0x30AA)	calib_blue_asc1	0000 0ddd dddd dddd	1024 (0x0400)
R12460 (R0x30AC)	calib_red_asc1	0000 0ddd dddd dddd	1024 (0x0400)

Table 3: Manufacturer Specific Register List (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)
R12462 (R0x30AE)	calib_green2_asc1	0000 0ddd dddd dddd	1024 (0x0400)
R12476 (R0x30BC)	calib_global	0000 0ddd dddd dddd	1024 (0x0400)
R12480 (R0x30C0)	calib_control	00dd dddd dddd dddd	544 (0x0220)
R12482 (R0x30C2)	calib_green1	0000 0ddd dddd dddd	1024 (0x0400)
R12484 (R0x30C4)	calib_blue	0000 0ddd dddd dddd	1024 (0x0400)
R12486 (R0x30C6)	calib_red	0000 0ddd dddd dddd	1024 (0x0400)
R12488 (R0x30C8)	calib_green2	0000 0ddd dddd dddd	1024 (0x0400)
R12528 (R0x30F0)	vcm_control	d000 dddd 000d dddd	0 (0x0000)
R12530 (R0x30F2)	vcm_new_code	0000 0000 dddd dddd	0 (0x0000)
R12532 (R0x30F4)	vcm_step_time	dddd dddd dddd dddd	0 (0x0000)
R12544 (R0x3100)	adacd_control	0000 0000 0000 00dd	2 (0x0002)
R12546 (R0x3102)	adacd_noise_model1	0000 00dd dddd dddd	160 (0x00A0)
R12548 (R0x3104)	adacd_noise_model2	0000 dddd dddd dddd	2925 (0x0B6D)
R12550 (R0x3106)	adacd_noise_floor1	dddd dddd dddd dddd	1540 (0x0604)
R12552 (R0x3108)	adacd_noise_floor2	dddd dddd dddd dddd	4618 (0x120A)
R12554 (R0x310A)	adacd_pedestal	0000 00dd dddd dddd	42 (0x002A)
R12556 (R0x310C)	adacd_gain_threshold_0	0000 0ddd dddd dddd	128 (0x0080)
R12558 (R0x310E)	adacd_gain_threshold_1	0000 0ddd dddd dddd	256 (0x0100)
R12560 (R0x3110)	adacd_gain_threshold_2	0000 0ddd dddd dddd	512 (0x0200)
R12592 (R0x3130)	otpm_tcfg_write_01	dddd dddd dddd dddd	30721 (0x7801)
R12596 (R0x3134)	otpm_tcfg_read_01	dddd dddd dddd dddd	3477 (0x0D95)
R12704 (R0x31A0)	descriptor_0	???? ???? ???? ????	257 (0x0101)
R12706 (R0x31A2)	descriptor_1	???? ???? ???? ????	513 (0x0201)
R12708 (R0x31A4)	descriptor_2	???? ???? ???? ????	514 (0x0202)

Table 3: Manufacturer Specific Register List (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)
R12710 (R0x31A6)	descriptor_3	???? ???? ???? ????	0 (0x0000)
R12712 (R0x31A8)	descriptor_4	???? ???? ???? ????	0 (0x0000)
R12714 (R0x31AA)	descriptor_5	???? ???? ???? ????	0 (0x0000)
R12716 (R0x31AC)	descriptor_6	???? ???? ???? ????	0 (0x0000)
R12718 (R0x31AE)	serial_format	dddd dddd dddd dddd	1 (0x0001)
R12720 (R0x31B0)	frame_preamble	0000 0000 dddd dddd	112 (0x0070)
R12722 (R0x31B2)	line_preamble	0000 0000 dddd dddd	73 (0x0049)
R12724 (R0x31B4)	mipi_timing_0	dddd dddd dddd dddd	3175 (0x0C67)
R12726 (R0x31B6)	mipi_timing_1	dddd dddd dddd dddd	784 (0x0310)
R12728 (R0x31B8)	mipi_timing_2	dddd dddd dddd dddd	782 (0x030E)
R12730 (R0x31BA)	mipi_timing_3	dddd dddd dddd dddd	781 (0x030D)
R12732 (R0x31BC)	mipi_timing_4	0ddd dddd 0ddd dddd	11 (0x000B)
R12744 (R0x31C8)	hispi_crc_0	dddd dddd dddd dddd	0 (0x0000)
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)
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)

Table 3: Manufacturer Specific Register List (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)
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)
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)

Table 3: Manufacturer Specific Register List (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)
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)
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)

Table 3: Manufacturer Specific Register List (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)
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)
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)

Table 3: Manufacturer Specific Register List (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)
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)
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)

Table 3: Manufacturer Specific Register List (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)
R14210 (R0x3782)	poly_origin_c	0000 dddd dddd dddd	0 (0x0000)
R14212 (R0x3784)	poly_origin_r	0000 dddd dddd dddd	0 (0x0000)
R14272 (R0x37C0)	p_gr_q5	dddd dddd dddd dddd	0 (0x0000)
R14274 (R0x37C2)	p_rd_q5	dddd dddd dddd dddd	0 (0x0000)
R14276 (R0x37C4)	p_bl_q5	dddd dddd dddd dddd	0 (0x0000)
R14278 (R0x37C6)	p_gb_q5	dddd dddd dddd dddd	0 (0x0000)
R14336 (R0x3800)	otpm_data_0	dddd dddd dddd dddd	0 (0x0000)
R14338 (R0x3802)	otpm_data_1	dddd dddd dddd dddd	0 (0x0000)
R14340 (R0x3804)	otpm_data_2	dddd dddd dddd dddd	0 (0x0000)
R14342 (R0x3806)	otpm_data_3	dddd dddd dddd dddd	0 (0x0000)
R14344 (R0x3808)	otpm_data_4	dddd dddd dddd dddd	0 (0x0000)
R14346 (R0x380A)	otpm_data_5	dddd dddd dddd dddd	0 (0x0000)
R14348 (R0x380C)	otpm_data_6	dddd dddd dddd dddd	0 (0x0000)
R14350 (R0x380E)	otpm_data_7	dddd dddd dddd dddd	0 (0x0000)
R14352 (R0x3810)	otpm_data_8	dddd dddd dddd dddd	0 (0x0000)
R14354 (R0x3812)	otpm_data_9	dddd dddd dddd dddd	0 (0x0000)
R14356 (R0x3814)	otpm_data_10	dddd dddd dddd dddd	0 (0x0000)
R14358 (R0x3816)	otpm_data_11	dddd dddd dddd dddd	0 (0x0000)
R14360 (R0x3818)	otpm_data_12	dddd dddd dddd dddd	0 (0x0000)
R14362 (R0x381A)	otpm_data_13	dddd dddd dddd dddd	0 (0x0000)
R14364 (R0x381C)	otpm_data_14	dddd dddd dddd dddd	0 (0x0000)
R14366 (R0x381E)	otpm_data_15	dddd dddd dddd dddd	0 (0x0000)
R14368 (R0x3820)	otpm_data_16	dddd dddd dddd dddd	0 (0x0000)
R14370 (R0x3822)	otpm_data_17	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14372 (R0x3824)	otpm_data_18	dddd dddd dddd dddd	0 (0x0000)
R14374 (R0x3826)	otpm_data_19	dddd dddd dddd dddd	0 (0x0000)
R14376 (R0x3828)	otpm_data_20	dddd dddd dddd dddd	0 (0x0000)
R14378 (R0x382A)	otpm_data_21	dddd dddd dddd dddd	0 (0x0000)
R14380 (R0x382C)	otpm_data_22	dddd dddd dddd dddd	0 (0x0000)
R14382 (R0x382E)	otpm_data_23	dddd dddd dddd dddd	0 (0x0000)
R14384 (R0x3830)	otpm_data_24	dddd dddd dddd dddd	0 (0x0000)
R14386 (R0x3832)	otpm_data_25	dddd dddd dddd dddd	0 (0x0000)
R14388 (R0x3834)	otpm_data_26	dddd dddd dddd dddd	0 (0x0000)
R14390 (R0x3836)	otpm_data_27	dddd dddd dddd dddd	0 (0x0000)
R14392 (R0x3838)	otpm_data_28	dddd dddd dddd dddd	0 (0x0000)
R14394 (R0x383A)	otpm_data_29	dddd dddd dddd dddd	0 (0x0000)
R14396 (R0x383C)	otpm_data_30	dddd dddd dddd dddd	0 (0x0000)
R14398 (R0x383E)	otpm_data_31	dddd dddd dddd dddd	0 (0x0000)
R14400 (R0x3840)	otpm_data_32	dddd dddd dddd dddd	0 (0x0000)
R14402 (R0x3842)	otpm_data_33	dddd dddd dddd dddd	0 (0x0000)
R14404 (R0x3844)	otpm_data_34	dddd dddd dddd dddd	0 (0x0000)
R14406 (R0x3846)	otpm_data_35	dddd dddd dddd dddd	0 (0x0000)
R14408 (R0x3848)	otpm_data_36	dddd dddd dddd dddd	0 (0x0000)
R14410 (R0x384A)	otpm_data_37	dddd dddd dddd dddd	0 (0x0000)
R14412 (R0x384C)	otpm_data_38	dddd dddd dddd dddd	0 (0x0000)
R14414 (R0x384E)	otpm_data_39	dddd dddd dddd dddd	0 (0x0000)
R14416 (R0x3850)	otpm_data_40	dddd dddd dddd dddd	0 (0x0000)
R14418 (R0x3852)	otpm_data_41	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14420 (R0x3854)	otpm_data_42	dddd dddd dddd dddd	0 (0x0000)
R14422 (R0x3856)	otpm_data_43	dddd dddd dddd dddd	0 (0x0000)
R14424 (R0x3858)	otpm_data_44	dddd dddd dddd dddd	0 (0x0000)
R14426 (R0x385A)	otpm_data_45	dddd dddd dddd dddd	0 (0x0000)
R14428 (R0x385C)	otpm_data_46	dddd dddd dddd dddd	0 (0x0000)
R14430 (R0x385E)	otpm_data_47	dddd dddd dddd dddd	0 (0x0000)
R14432 (R0x3860)	otpm_data_48	dddd dddd dddd dddd	0 (0x0000)
R14434 (R0x3862)	otpm_data_49	dddd dddd dddd dddd	0 (0x0000)
R14436 (R0x3864)	otpm_data_50	dddd dddd dddd dddd	0 (0x0000)
R14438 (R0x3866)	otpm_data_51	dddd dddd dddd dddd	0 (0x0000)
R14440 (R0x3868)	otpm_data_52	dddd dddd dddd dddd	0 (0x0000)
R14442 (R0x386A)	otpm_data_53	dddd dddd dddd dddd	0 (0x0000)
R14444 (R0x386C)	otpm_data_54	dddd dddd dddd dddd	0 (0x0000)
R14446 (R0x386E)	otpm_data_55	dddd dddd dddd dddd	0 (0x0000)
R14448 (R0x3870)	otpm_data_56	dddd dddd dddd dddd	0 (0x0000)
R14450 (R0x3872)	otpm_data_57	dddd dddd dddd dddd	0 (0x0000)
R14452 (R0x3874)	otpm_data_58	dddd dddd dddd dddd	0 (0x0000)
R14454 (R0x3876)	otpm_data_59	dddd dddd dddd dddd	0 (0x0000)
R14456 (R0x3878)	otpm_data_60	dddd dddd dddd dddd	0 (0x0000)
R14458 (R0x387A)	otpm_data_61	dddd dddd dddd dddd	0 (0x0000)
R14460 (R0x387C)	otpm_data_62	dddd dddd dddd dddd	0 (0x0000)
R14462 (R0x387E)	otpm_data_63	dddd dddd dddd dddd	0 (0x0000)
R14464 (R0x3880)	otpm_data_64	dddd dddd dddd dddd	0 (0x0000)
R14466 (R0x3882)	otpm_data_65	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14468 (R0x3884)	otpm_data_66	dddd dddd dddd dddd	0 (0x0000)
R14470 (R0x3886)	otpm_data_67	dddd dddd dddd dddd	0 (0x0000)
R14472 (R0x3888)	otpm_data_68	dddd dddd dddd dddd	0 (0x0000)
R14474 (R0x388A)	otpm_data_69	dddd dddd dddd dddd	0 (0x0000)
R14476 (R0x388C)	otpm_data_70	dddd dddd dddd dddd	0 (0x0000)
R14478 (R0x388E)	otpm_data_71	dddd dddd dddd dddd	0 (0x0000)
R14480 (R0x3890)	otpm_data_72	dddd dddd dddd dddd	0 (0x0000)
R14482 (R0x3892)	otpm_data_73	dddd dddd dddd dddd	0 (0x0000)
R14484 (R0x3894)	otpm_data_74	dddd dddd dddd dddd	0 (0x0000)
R14486 (R0x3896)	otpm_data_75	dddd dddd dddd dddd	0 (0x0000)
R14488 (R0x3898)	otpm_data_76	dddd dddd dddd dddd	0 (0x0000)
R14490 (R0x389A)	otpm_data_77	dddd dddd dddd dddd	0 (0x0000)
R14492 (R0x389C)	otpm_data_78	dddd dddd dddd dddd	0 (0x0000)
R14494 (R0x389E)	otpm_data_79	dddd dddd dddd dddd	0 (0x0000)
R14496 (R0x38A0)	otpm_data_80	dddd dddd dddd dddd	0 (0x0000)
R14498 (R0x38A2)	otpm_data_81	dddd dddd dddd dddd	0 (0x0000)
R14500 (R0x38A4)	otpm_data_82	dddd dddd dddd dddd	0 (0x0000)
R14502 (R0x38A6)	otpm_data_83	dddd dddd dddd dddd	0 (0x0000)
R14504 (R0x38A8)	otpm_data_84	dddd dddd dddd dddd	0 (0x0000)
R14506 (R0x38AA)	otpm_data_85	dddd dddd dddd dddd	0 (0x0000)
R14508 (R0x38AC)	otpm_data_86	dddd dddd dddd dddd	0 (0x0000)
R14510 (R0x38AE)	otpm_data_87	dddd dddd dddd dddd	0 (0x0000)
R14512 (R0x38B0)	otpm_data_88	dddd dddd dddd dddd	0 (0x0000)
R14514 (R0x38B2)	otpm_data_89	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14516 (R0x38B4)	otpm_data_90	dddd dddd dddd dddd	0 (0x0000)
R14518 (R0x38B6)	otpm_data_91	dddd dddd dddd dddd	0 (0x0000)
R14520 (R0x38B8)	otpm_data_92	dddd dddd dddd dddd	0 (0x0000)
R14522 (R0x38BA)	otpm_data_93	dddd dddd dddd dddd	0 (0x0000)
R14524 (R0x38BC)	otpm_data_94	dddd dddd dddd dddd	0 (0x0000)
R14526 (R0x38BE)	otpm_data_95	dddd dddd dddd dddd	0 (0x0000)
R14528 (R0x38C0)	otpm_data_96	dddd dddd dddd dddd	0 (0x0000)
R14530 (R0x38C2)	otpm_data_97	dddd dddd dddd dddd	0 (0x0000)
R14532 (R0x38C4)	otpm_data_98	dddd dddd dddd dddd	0 (0x0000)
R14534 (R0x38C6)	otpm_data_99	dddd dddd dddd dddd	0 (0x0000)
R14536 (R0x38C8)	otpm_data_100	dddd dddd dddd dddd	0 (0x0000)
R14538 (R0x38CA)	otpm_data_101	dddd dddd dddd dddd	0 (0x0000)
R14540 (R0x38CC)	otpm_data_102	dddd dddd dddd dddd	0 (0x0000)
R14542 (R0x38CE)	otpm_data_103	dddd dddd dddd dddd	0 (0x0000)
R14544 (R0x38D0)	otpm_data_104	dddd dddd dddd dddd	0 (0x0000)
R14546 (R0x38D2)	otpm_data_105	dddd dddd dddd dddd	0 (0x0000)
R14548 (R0x38D4)	otpm_data_106	dddd dddd dddd dddd	0 (0x0000)
R14550 (R0x38D6)	otpm_data_107	dddd dddd dddd dddd	0 (0x0000)
R14552 (R0x38D8)	otpm_data_108	dddd dddd dddd dddd	0 (0x0000)
R14554 (R0x38DA)	otpm_data_109	dddd dddd dddd dddd	0 (0x0000)
R14556 (R0x38DC)	otpm_data_110	dddd dddd dddd dddd	0 (0x0000)
R14558 (R0x38DE)	otpm_data_111	dddd dddd dddd dddd	0 (0x0000)
R14560 (R0x38E0)	otpm_data_112	dddd dddd dddd dddd	0 (0x0000)
R14562 (R0x38E2)	otpm_data_113	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14564 (R0x38E4)	otpm_data_114	dddd dddd dddd dddd	0 (0x0000)
R14566 (R0x38E6)	otpm_data_115	dddd dddd dddd dddd	0 (0x0000)
R14568 (R0x38E8)	otpm_data_116	dddd dddd dddd dddd	0 (0x0000)
R14570 (R0x38EA)	otpm_data_117	dddd dddd dddd dddd	0 (0x0000)
R14572 (R0x38EC)	otpm_data_118	dddd dddd dddd dddd	0 (0x0000)
R14574 (R0x38EE)	otpm_data_119	dddd dddd dddd dddd	0 (0x0000)
R14576 (R0x38F0)	otpm_data_120	dddd dddd dddd dddd	0 (0x0000)
R14578 (R0x38F2)	otpm_data_121	dddd dddd dddd dddd	0 (0x0000)
R14580 (R0x38F4)	otpm_data_122	dddd dddd dddd dddd	0 (0x0000)
R14582 (R0x38F6)	otpm_data_123	dddd dddd dddd dddd	0 (0x0000)
R14584 (R0x38F8)	otpm_data_124	dddd dddd dddd dddd	0 (0x0000)
R14586 (R0x38FA)	otpm_data_125	dddd dddd dddd dddd	0 (0x0000)
R14588 (R0x38FC)	otpm_data_126	dddd dddd dddd dddd	0 (0x0000)
R14590 (R0x38FE)	otpm_data_127	dddd dddd dddd dddd	0 (0x0000)
R14592 (R0x3900)	otpm_data_128	dddd dddd dddd dddd	0 (0x0000)
R14594 (R0x3902)	otpm_data_129	dddd dddd dddd dddd	0 (0x0000)
R14596 (R0x3904)	otpm_data_130	dddd dddd dddd dddd	0 (0x0000)
R14598 (R0x3906)	otpm_data_131	dddd dddd dddd dddd	0 (0x0000)
R14600 (R0x3908)	otpm_data_132	dddd dddd dddd dddd	0 (0x0000)
R14602 (R0x390A)	otpm_data_133	dddd dddd dddd dddd	0 (0x0000)
R14604 (R0x390C)	otpm_data_134	dddd dddd dddd dddd	0 (0x0000)
R14606 (R0x390E)	otpm_data_135	dddd dddd dddd dddd	0 (0x0000)
R14608 (R0x3910)	otpm_data_136	dddd dddd dddd dddd	0 (0x0000)
R14610 (R0x3912)	otpm_data_137	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14612 (R0x3914)	otpm_data_138	dddd dddd dddd dddd	0 (0x0000)
R14614 (R0x3916)	otpm_data_139	dddd dddd dddd dddd	0 (0x0000)
R14616 (R0x3918)	otpm_data_140	dddd dddd dddd dddd	0 (0x0000)
R14618 (R0x391A)	otpm_data_141	dddd dddd dddd dddd	0 (0x0000)
R14620 (R0x391C)	otpm_data_142	dddd dddd dddd dddd	0 (0x0000)
R14622 (R0x391E)	otpm_data_143	dddd dddd dddd dddd	0 (0x0000)
R14624 (R0x3920)	otpm_data_144	dddd dddd dddd dddd	0 (0x0000)
R14626 (R0x3922)	otpm_data_145	dddd dddd dddd dddd	0 (0x0000)
R14628 (R0x3924)	otpm_data_146	dddd dddd dddd dddd	0 (0x0000)
R14630 (R0x3926)	otpm_data_147	dddd dddd dddd dddd	0 (0x0000)
R14632 (R0x3928)	otpm_data_148	dddd dddd dddd dddd	0 (0x0000)
R14634 (R0x392A)	otpm_data_149	dddd dddd dddd dddd	0 (0x0000)
R14636 (R0x392C)	otpm_data_150	dddd dddd dddd dddd	0 (0x0000)
R14638 (R0x392E)	otpm_data_151	dddd dddd dddd dddd	0 (0x0000)
R14640 (R0x3930)	otpm_data_152	dddd dddd dddd dddd	0 (0x0000)
R14642 (R0x3932)	otpm_data_153	dddd dddd dddd dddd	0 (0x0000)
R14644 (R0x3934)	otpm_data_154	dddd dddd dddd dddd	0 (0x0000)
R14646 (R0x3936)	otpm_data_155	dddd dddd dddd dddd	0 (0x0000)
R14648 (R0x3938)	otpm_data_156	dddd dddd dddd dddd	0 (0x0000)
R14650 (R0x393A)	otpm_data_157	dddd dddd dddd dddd	0 (0x0000)
R14652 (R0x393C)	otpm_data_158	dddd dddd dddd dddd	0 (0x0000)
R14654 (R0x393E)	otpm_data_159	dddd dddd dddd dddd	0 (0x0000)
R14656 (R0x3940)	otpm_data_160	dddd dddd dddd dddd	0 (0x0000)
R14658 (R0x3942)	otpm_data_161	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14660 (R0x3944)	otpm_data_162	dddd dddd dddd dddd	0 (0x0000)
R14662 (R0x3946)	otpm_data_163	dddd dddd dddd dddd	0 (0x0000)
R14664 (R0x3948)	otpm_data_164	dddd dddd dddd dddd	0 (0x0000)
R14666 (R0x394A)	otpm_data_165	dddd dddd dddd dddd	0 (0x0000)
R14668 (R0x394C)	otpm_data_166	dddd dddd dddd dddd	0 (0x0000)
R14670 (R0x394E)	otpm_data_167	dddd dddd dddd dddd	0 (0x0000)
R14672 (R0x3950)	otpm_data_168	dddd dddd dddd dddd	0 (0x0000)
R14674 (R0x3952)	otpm_data_169	dddd dddd dddd dddd	0 (0x0000)
R14676 (R0x3954)	otpm_data_170	dddd dddd dddd dddd	0 (0x0000)
R14678 (R0x3956)	otpm_data_171	dddd dddd dddd dddd	0 (0x0000)
R14680 (R0x3958)	otpm_data_172	dddd dddd dddd dddd	0 (0x0000)
R14682 (R0x395A)	otpm_data_173	dddd dddd dddd dddd	0 (0x0000)
R14684 (R0x395C)	otpm_data_174	dddd dddd dddd dddd	0 (0x0000)
R14686 (R0x395E)	otpm_data_175	dddd dddd dddd dddd	0 (0x0000)
R14688 (R0x3960)	otpm_data_176	dddd dddd dddd dddd	0 (0x0000)
R14690 (R0x3962)	otpm_data_177	dddd dddd dddd dddd	0 (0x0000)
R14692 (R0x3964)	otpm_data_178	dddd dddd dddd dddd	0 (0x0000)
R14694 (R0x3966)	otpm_data_179	dddd dddd dddd dddd	0 (0x0000)
R14696 (R0x3968)	otpm_data_180	dddd dddd dddd dddd	0 (0x0000)
R14698 (R0x396A)	otpm_data_181	dddd dddd dddd dddd	0 (0x0000)
R14700 (R0x396C)	otpm_data_182	dddd dddd dddd dddd	0 (0x0000)
R14702 (R0x396E)	otpm_data_183	dddd dddd dddd dddd	0 (0x0000)
R14704 (R0x3970)	otpm_data_184	dddd dddd dddd dddd	0 (0x0000)
R14706 (R0x3972)	otpm_data_185	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14708 (R0x3974)	otpm_data_186	dddd dddd dddd dddd	0 (0x0000)
R14710 (R0x3976)	otpm_data_187	dddd dddd dddd dddd	0 (0x0000)
R14712 (R0x3978)	otpm_data_188	dddd dddd dddd dddd	0 (0x0000)
R14714 (R0x397A)	otpm_data_189	dddd dddd dddd dddd	0 (0x0000)
R14716 (R0x397C)	otpm_data_190	dddd dddd dddd dddd	0 (0x0000)
R14718 (R0x397E)	otpm_data_191	dddd dddd dddd dddd	0 (0x0000)
R14720 (R0x3980)	otpm_data_192	dddd dddd dddd dddd	0 (0x0000)
R14722 (R0x3982)	otpm_data_193	dddd dddd dddd dddd	0 (0x0000)
R14724 (R0x3984)	otpm_data_194	dddd dddd dddd dddd	0 (0x0000)
R14726 (R0x3986)	otpm_data_195	dddd dddd dddd dddd	0 (0x0000)
R14728 (R0x3988)	otpm_data_196	dddd dddd dddd dddd	0 (0x0000)
R14730 (R0x398A)	otpm_data_197	dddd dddd dddd dddd	0 (0x0000)
R14732 (R0x398C)	otpm_data_198	dddd dddd dddd dddd	0 (0x0000)
R14734 (R0x398E)	otpm_data_199	dddd dddd dddd dddd	0 (0x0000)
R14736 (R0x3990)	otpm_data_200	dddd dddd dddd dddd	0 (0x0000)
R14738 (R0x3992)	otpm_data_201	dddd dddd dddd dddd	0 (0x0000)
R14740 (R0x3994)	otpm_data_202	dddd dddd dddd dddd	0 (0x0000)
R14742 (R0x3996)	otpm_data_203	dddd dddd dddd dddd	0 (0x0000)
R14744 (R0x3998)	otpm_data_204	dddd dddd dddd dddd	0 (0x0000)
R14746 (R0x399A)	otpm_data_205	dddd dddd dddd dddd	0 (0x0000)
R14748 (R0x399C)	otpm_data_206	dddd dddd dddd dddd	0 (0x0000)
R14750 (R0x399E)	otpm_data_207	dddd dddd dddd dddd	0 (0x0000)
R14752 (R0x39A0)	otpm_data_208	dddd dddd dddd dddd	0 (0x0000)
R14754 (R0x39A2)	otpm_data_209	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14756 (R0x39A4)	otpm_data_210	dddd dddd dddd dddd	0 (0x0000)
R14758 (R0x39A6)	otpm_data_211	dddd dddd dddd dddd	0 (0x0000)
R14760 (R0x39A8)	otpm_data_212	dddd dddd dddd dddd	0 (0x0000)
R14762 (R0x39AA)	otpm_data_213	dddd dddd dddd dddd	0 (0x0000)
R14764 (R0x39AC)	otpm_data_214	dddd dddd dddd dddd	0 (0x0000)
R14766 (R0x39AE)	otpm_data_215	dddd dddd dddd dddd	0 (0x0000)
R14768 (R0x39B0)	otpm_data_216	dddd dddd dddd dddd	0 (0x0000)
R14770 (R0x39B2)	otpm_data_217	dddd dddd dddd dddd	0 (0x0000)
R14772 (R0x39B4)	otpm_data_218	dddd dddd dddd dddd	0 (0x0000)
R14774 (R0x39B6)	otpm_data_219	dddd dddd dddd dddd	0 (0x0000)
R14776 (R0x39B8)	otpm_data_220	dddd dddd dddd dddd	0 (0x0000)
R14778 (R0x39BA)	otpm_data_221	dddd dddd dddd dddd	0 (0x0000)
R14780 (R0x39BC)	otpm_data_222	dddd dddd dddd dddd	0 (0x0000)
R14782 (R0x39BE)	otpm_data_223	dddd dddd dddd dddd	0 (0x0000)
R14784 (R0x39C0)	otpm_data_224	dddd dddd dddd dddd	0 (0x0000)
R14786 (R0x39C2)	otpm_data_225	dddd dddd dddd dddd	0 (0x0000)
R14788 (R0x39C4)	otpm_data_226	dddd dddd dddd dddd	0 (0x0000)
R14790 (R0x39C6)	otpm_data_227	dddd dddd dddd dddd	0 (0x0000)
R14792 (R0x39C8)	otpm_data_228	dddd dddd dddd dddd	0 (0x0000)
R14794 (R0x39CA)	otpm_data_229	dddd dddd dddd dddd	0 (0x0000)
R14796 (R0x39CC)	otpm_data_230	dddd dddd dddd dddd	0 (0x0000)
R14798 (R0x39CE)	otpm_data_231	dddd dddd dddd dddd	0 (0x0000)
R14800 (R0x39D0)	otpm_data_232	dddd dddd dddd dddd	0 (0x0000)
R14802 (R0x39D2)	otpm_data_233	dddd dddd dddd dddd	0 (0x0000)

Table 3: Manufacturer Specific Register List (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)
R14804 (R0x39D4)	otpm_data_234	dddd dddd dddd dddd	0 (0x0000)
R14806 (R0x39D6)	otpm_data_235	dddd dddd dddd dddd	0 (0x0000)
R14808 (R0x39D8)	otpm_data_236	dddd dddd dddd dddd	0 (0x0000)
R14810 (R0x39DA)	otpm_data_237	dddd dddd dddd dddd	0 (0x0000)
R14812 (R0x39DC)	otpm_data_238	dddd dddd dddd dddd	0 (0x0000)
R14814 (R0x39DE)	otpm_data_239	dddd dddd dddd dddd	0 (0x0000)
R14816 (R0x39E0)	otpm_data_240	dddd dddd dddd dddd	0 (0x0000)
R14818 (R0x39E2)	otpm_data_241	dddd dddd dddd dddd	0 (0x0000)
R14820 (R0x39E4)	otpm_data_242	dddd dddd dddd dddd	0 (0x0000)
R14822 (R0x39E6)	otpm_data_243	dddd dddd dddd dddd	0 (0x0000)
R14824 (R0x39E8)	otpm_data_244	dddd dddd dddd dddd	0 (0x0000)
R14826 (R0x39EA)	otpm_data_245	dddd dddd dddd dddd	0 (0x0000)
R14828 (R0x39EC)	otpm_data_246	dddd dddd dddd dddd	0 (0x0000)
R14830 (R0x39EE)	otpm_data_247	dddd dddd dddd dddd	0 (0x0000)
R14832 (R0x39F0)	otpm_data_248	dddd dddd dddd dddd	0 (0x0000)
R14834 (R0x39F2)	otpm_data_249	dddd dddd dddd dddd	0 (0x0000)
R14836 (R0x39F4)	otpm_data_250	dddd dddd dddd dddd	0 (0x0000)
R14838 (R0x39F6)	otpm_data_251	dddd dddd dddd dddd	0 (0x0000)
R14840 (R0x39F8)	otpm_data_252	dddd dddd dddd dddd	0 (0x0000)
R14842 (R0x39FA)	otpm_data_253	dddd dddd dddd dddd	0 (0x0000)
R14844 (R0x39FC)	otpm_data_254	dddd dddd dddd dddd	0 (0x0000)
R14846 (R0x39FE)	otpm_data_255	dddd dddd dddd dddd	0 (0x0000)

Register Descriptions

SMIA Configuration Register Description

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
0 R0x0000	15:0	0x4800	**chip_version_reg (R/W)**	N	N
	This register is an alias of R0x3000-1. Read-only. Can be made read/write by clearing R0x301A-B[3].				
2 R0x0002	7:0	0x70	**revision_number (R/W)**	N	N
	Aptina-assigned revision number. Read-only. Can be made read/write by clearing R0x301A-B[3].				
3 R0x0003	7:0	0x06	**manufacturer_id (RO)**	N	N
	Manufacturer ID assigned to Aptina. Read-only. Can be made read/write by clearing R0x301A-B[3].				
4 R0x0004	7:0	0x0A	**smia_version (RO)**	N	N
	This register is an alias of R0x303A. Read-only.				
5 R0x0005	7:0	0xFF	**frame_count (RO)**	Y	N
	This register is an alias of R0x303B. Read-only.				
6 R0x0006	7:0	0x00	**pixel_order (RO)**	N	N
	This register is an alias of R0x3024. Read-only.				
8 R0x0008	15:0	0x002A	**data_pedestal (R/W)**	N	Y
	This register is an alias of R0x301E-F. Read-only. Can be made read/write by clearing R0x301A-B[3].				
64 R0x0040	7:0	0x01	**frame_format_model_type (RO)**	N	N
	Type 1. 2-byte Generic Frame Format Description. Read-only.				
65 R0x0041	7:0	0x12	**frame_format_model_subtype (RO)**	N	N
	Number of descriptors: 1 X (column) descriptor and two Y (row) descriptors. Read-only.				
66 R0x0042	15:0	0x5A20	**frame_format_descriptor_0 (RO)**	Y	N
	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 R0x0044	15:0	0x1002	**frame_format_descriptor_1 (RO)**	Y	N
	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 R0x3070-1) this register will return 0x1000. Read-only.				
70 R0x0046	15:0	0x5798	**frame_format_descriptor_2 (RO)**	Y	N
	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 R0x0048	15:0	0x0000	**frame_format_descriptor_3 (RO)**	N	N
	Read-only.				
74 R0x004A	15:0	0x0000	**frame_format_descriptor_4 (RO)**	N	N
	Read-only.				
76 R0x004C	15:0	0x0000	**frame_format_descriptor_5 (RO)**	N	N
	Read-only.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
78 R0x004E	15:0	0x0000	**frame_format_descriptor_6 (RO)**	N	N
	Read-only.				
80 R0x0050	15:0	0x0000	**frame_format_descriptor_7 (RO)**	N	N
	Read-only.				
82 R0x0052	15:0	0x0000	**frame_format_descriptor_8 (RO)**	N	N
	Read-only.				
84 R0x0054	15:0	0x0000	**frame_format_descriptor_9 (RO)**	N	N
	Read-only.				
86 R0x0056	15:0	0x0000	**frame_format_descriptor_10 (RO)**	N	N
	Read-only.				
88 R0x0058	15:0	0x0000	**frame_format_descriptor_11 (RO)**	N	N
	Read-only.				
90 R0x005A	15:0	0x0000	**frame_format_descriptor_12 (RO)**	N	N
	Read-only.				
92 R0x005C	15:0	0x0000	**frame_format_descriptor_13 (RO)**	N	N
	Read-only.				
94 R0x005E	15:0	0x0000	**frame_format_descriptor_14 (RO)**	N	N
	Read-only.				
128 R0x0080	15:0	0x0001	**analogue_gain_capability (RO)**	N	N
	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 R0x0084	15:0	0x0008	**analogue_gain_code_min (RO)**	N	N
	Minimum gain code. Read-only.				
134 R0x0086	15:0	0x01FF	**analogue_gain_code_max (RO)**	N	N
	Maximum gain code. Read-only.				
136 R0x0088	15:0	0x0001	**analogue_gain_code_step (RO)**	N	N
	Gain code step size. Read-only.				
138 R0x008A	15:0	0x0000	**analogue_gain_type (RO)**	N	N
	Indicates support for analog gain coding type 0 (baseline SMIA). Read-only.				
140 R0x008C	15:0	0x0001	**analogue_gain_m0 (RO)**	N	N
	Constants for the gain equation. Read-only.				
142 R0x008E	15:0	0x0000	**analogue_gain_c0 (RO)**	N	N
	Constants for the gain equation. Read-only.				
144 R0x0090	15:0	0x0000	**analogue_gain_m1 (RO)**	N	N
	Constants for the gain equation. Read-only.				
146 R0x0092	15:0	0x0008	**analogue_gain_c1 (RO)**	N	N
	Constants for the gain equation. Read-only.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
192 R0x00C0	7:0	0x01	**data_format_model_type (RO)**	N	N
	Indicates the use of 2-byte data format. Read-only.				
193 R0x00C1	7:0	0x03	**data_format_model_subtype (RO)**	N	N
	Indicates the provision of 3 data format descriptors. Read-only.				
194 R0x00C2	15:0	0x0A0A	**data_format_descriptor_0 (RO)**	N	N
	Indicates support for RAW10, uncompressed data format. Read-only.				
196 R0x00C4	15:0	0x0808	**data_format_descriptor_1 (RO)**	N	N
	Indicates support for RAW8 data format in which the two LSB of each 10-bit pixel data value are discarded. Read-only.				
198 R0x00C6	15:0	0x0A08	**data_format_descriptor_2 (RO)**	N	N
	Indicates support for RAW8 data format in which each 10-bit pixel data value is compressed to an 8-bit value. Read-only.				
200 R0x00C8	15:0	0x0000	**data_format_descriptor_3 (RO)**	N	N
	Read-only.				
202 R0x00CA	15:0	0x0000	**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	**vert_flip** This register field is an alias of R0x3040-1[15].	Y	YM
	0	0x00	**horiz_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 three-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 two bits of this three-bit field supply the Virtual Channel (VC) identifer in the Data Identifier (DI) byte which forms part of the short and long packet headers.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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	0x0A0A	**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	0x02CE	**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	0x0014	**analogue_gain_code_global (R/W)**	Y	N
	This register is an alias of R0x3028-9.				
518 R0x0206	15:0	0x0014	**analogue_gain_code_greenr (R/W)**	Y	N
	This register is an alias of R0x302A-B.				
520 R0x0208	15:0	0x0014	**analogue_gain_code_red (R/W)**	Y	N
	This register is an alias of R0x302C-D.				
522 R0x020A	15:0	0x0014	**analogue_gain_code_blue (R/W)**	Y	N
	This register is an alias of R0x302E-F.				
524 R0x020C	15:0	0x0014	**analogue_gain_code_greenb (R/W)**	Y	N
	This register is an alias of R0x3030-1.				
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 R0x3036-7.				
532 R0x0214	15:0	0x0100	**digital_gain_greenb (R/W)**	Y	N
	This register is an alias of R0x3038-9.				
768 R0x0300	15:0	0x0005	**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.				
772 R0x0304	15:0	0x0002	**pre_pll_clk_div (R/W)**	N	Y
	Clock divisor applied to EXTCLK to generate PLL input clock.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
774 R0x0306	15:0	0x0046	**pll_multiplier (R/W)**	N	Y
	Clock multiplier applied to PLL input clock.				
776 R0x0308	15:0	0x000A	**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.				
832 R0x0340	15:0	0x07E5	**frame_length_lines (R/W)**	Y	YM
	This register is an alias of R0x300A-B.				
834 R0x0342	15:0	0x1476	**line_length_pck (R/W)**	Y	YM
	This register is an alias of R0x300C-D.				
836 R0x0344	15:0	0x0008	**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	0x0A27	**x_addr_end (R/W)**	Y	N
	This register is an alias of R0x3008-9.				
842 R0x034A	15:0	0x079F	**y_addr_end (R/W)**	Y	YM
	This register is an alias of R0x3006-7.				
844 R0x034C	15:0	0x0A20	**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	0x0798	**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 register is 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].				
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				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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 compression uses 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 (R0x0012-3).				
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 R0x3074-5.				
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 Descriptions

Table 5: SMIA Parameter Limits Register Descriptions
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. The result will be a graceful degradation of frame rate, like pre-mi3120 products.				
4104 R0x1008	15:0	0x02CE	**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	0x01E2	**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	31:0	0x40000000	**min_ext_clk_freq_mhz (RO)**	N	N
	FLP32. Minimum external clock frequency into PLL is 2.0 MHz. Read-only.				
4356 R0x1104	31:0	0x42800000	**max_ext_clk_freq_mhz (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	31:0	0x40800000	**min_pll_ip_freq_mhz (RO)**	N	N
	FLP32. Minimum clock frequency into the PFD of the PLL is 4.0 MHz. Read-only.				
4368 R0x1110	31:0	0x41C00000	**max_pll_ip_freq_mhz (RO)**	N	N
	FLP32. Maximum clock frequency into the PFD of the PLL is 24 MHz. Read-only.				

Table 5: SMIA Parameter Limits Register Descriptions (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	31:0	0x43C00000	**min_pll_op_freq_mhz (RO)**	N	N
	FLP32. Minimum output frequency supported by the PLL is 384.0 MHz. Read-only.				
4380 R0x111C	31:0	0x44520000	**max_pll_op_freq_mhz (RO)**	N	N
	FLP32. Maximum output frequency supported by the PLL is 1000.0 MHz. Read-only.				
4384 R0x1120	15:0	0x0001	**min_vt_sys_clk_div (RO)**	N	N
	Minimum divisor for the video timing sys_clk. Read-only.				
4386 R0x1122	15:0	0x0010	**max_vt_sys_clk_div (RO)**	N	N
	Maximum divisor for the video timing sys_clk. Read-only.				
4388 R0x1124	31:0	0x41C00000	**min_vt_sys_clk_freq_mhz (RO)**	N	N
	FLP32. Minimum frequency for the video timing sys_clk is 24.0 MHz.				
4392 R0x1128	31:0	0x447A0000	**max_vt_sys_clk_freq_mhz (RO)**	N	N
	FLP32. Maximum frequency for the video timing sys_clk is 1000.0 MHz. Read-only.				
4396 R0x112C	31:0	0x4099999A	**min_vt_pix_clk_freq_mhz (RO)**	N	N
	FLP32. Minimum frequency for video timing pix_clk is 4.8 MHz. Read-only.				
4400 R0x1130	31:0	0x42E00000	**max_vt_pix_clk_freq_mhz (RO)**	N	N
	FLP32. Maximum frequency for video timing pix_clk is 200.0 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	0x0010	**max_vt_pix_clk_div (RO)**	N	N
	Maximum divisor for the video timing pix_clk. Read-only.				
4416 R0x1140	15:0	0x004F	**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	0x0590	**min_line_length_pck (R/W)**	N	N
	Minimum line length. Read-only. Can be made read/write by clearing R0x301A-B[3].				
4422 R0x1146	15:0	0xFFFE	**max_line_length_pck (R/W)**	N	N
	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 R0x1148	15:0	0x044E	**min_line_blanking_pck (R/W)**	N	N
	Minimum line blanking time. Read-only. Can be made read/write by clearing R0x301A-B[3].				
4426 R0x114A	15:0	0x004D	**min_frame_blanking_lines (R/W)**	N	N
	Minimum frame blanking time. Read-only. Can be made read/write by clearing R0x301A-B[3].				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
4448 R0x1160	15:0	0x0001	**min_op_sys_clk_div (RO)**	N	N
	Minimum divisor for the output sys_clk. Read-only.				
4450 R0x1162	15:0	0x0010	**max_op_sys_clk_div (RO)**	N	N
	Maximum divisor for the output sys_clk. Read-only.				
4452 R0x1164	31:0	0x41C00000	**min_op_sys_clk_freq_mhz (RO)**	N	N
	FLP32. Minimum frequency for output sys_clk is 24.0 MHz. Read-only.				
4456 R0x1168	31:0	0x44520000	**max_op_sys_clk_freq_mhz (RO)**	N	N
	FLP32. Maximum frequency for output sys_clk is 1000.0 MHz. Read-only.				
4460 R0x116C	15:0	0x0008	**min_op_pix_clk_div (RO)**	N	N
	Minimum divisor for output pix_clk. Read-only.				
4462 R0x116E	15:0	0x000A	**max_op_pix_clk_div (RO)**	N	N
	Maximum divisor for output pix_clk. Read-only.				
4464 R0x1170	31:0	0x4019999A	**min_op_pix_clk_freq_mhz (RO)**	N	N
	FLP32. Minimum frequency for output pix_clk is 2.4 MHz. Read-only.				
4468 R0x1174	31:0	0x42A80000	**max_op_pix_clk_freq_mhz (RO)**	N	N
	FLP32. Maximum frequency for output pix_clk is 100.0 MHz. Read-only.				
4480 R0x1180	15:0	0x0000	**x_addr_min (RO)**	N	N
	Minimum value for x_addr_start, x_addr_end. Read-only.				
4482 R0x1182	15:0	0x0000	**y_addr_min (RO)**	N	N
	Minimum value for y_addr_start, y_addr_end. Read-only.				
4484 R0x1184	15:0	0x0A2F	**x_addr_max (RO)**	N	N
	Maximum value for x_addr_start, x_addr_end. Read-only.				
4486 R0x1186	15:0	0x07A7	**y_addr_max (RO)**	N	N
	Maximum value for y_addr_start, y_addr_end. Read-only.				
4544 R0x11C0	15:0	0x0001	**min_even_inc (RO)**	N	N
	Minimum value for increment of even X/Y addresses when subsampling is enabled. Read-only.				
4546 R0x11C2	15:0	0x0001	**max_even_inc (RO)**	N	N
	Maximum value for increment of even X/Y addresses when subsampling is enabled. Read-only.				
4548 R0x11C4	15:0	0x0001	**min_odd_inc (RO)**	N	N
	Minimum value for increment of odd X/Y addresses when subsampling is enabled. Read-only.				
4550 R0x11C6	15:0	0x0007	**max_odd_inc (RO)**	N	N
	Maximum value for increment of odd X/Y addresses when subsampling is enabled. Read-only. Higher increment values are supported by the sensor, but only the values 1, 3 and 7 for x_odd_inc and 1, 3, 7, 15 and 31 for y_odd_inc. A value of 3 gives 2x subsampling and a value of 7 gives 4x subsampling.				
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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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 Descriptions

Table 6: Manufacturer-Specific Register Descriptions
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	0x4800	**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	0x0008	**x_addr_start_ (R/W)**	Y	N
	The first column of visible pixels to be read out (not counting any dark columns that may be read). To move the image window, set this register to the starting X value.				
12294 R0x3006	15:0	0x079F	**y_addr_end_ (R/W)**	Y	YM
	The last row of visible pixels to be read out.				
12296 R0x3008	15:0	0x0A27	**x_addr_end_ (R/W)**	Y	N
	The last column of visible pixels to be read out.				
12298 R0x300A	15:0	0x07E5	**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	0x1476	**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	0x00A0	**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	0x02CE	**fine_integration_time_ (R/W)**	Y	N
	Integration time specified as a number of pixel clocks.				

Table 6: Manufacturer-Specific Register Descriptions (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	0x0111	**row_speed (R/W)**		
	15:11	X	**Reserved**		
	10:8	0x0001	**opclk_speed** Slows down the output pixel clock frequency relative to the system clock frequency. A programmed value of N gives a output pixel clock period of N system clocks. Only values 1, 2 and 4 are supported. A value of 0 is illegal: it causes the clock to stop.	N	N
	7	X	**Reserved**		
	6:4	0x0001	**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	0x0001	**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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12314 R0x301A	15:0	0x0058	**reset_register (R/W)**		
	15	0x0000	**grouped_parameter_hold** **0 = Update of many of the registers is synchronized to frame start.** **1 = Inhibit register updates; register changes will remain pending until this bit is returned to 0. When this bit is returned to 0, all pending register updates will be made on the next frame start.**	N	N
	14	0x0000	**gain_insert** **When set, the gain values will always take effect the following frame independent of the integration time. When not set, gain will not update if an integration time change is in progress.**	N	Y
	13	X	**Reserved**		
	12	0x0000	**smia_serialiser_dis** **This bit disables the SMIA high-speed serialiser and differential output buffers.**	N	N
	11	X	**Reserved**		
	10	0x0000	**restart_bad** **1 = A restart is forced any time a bad frame is detected. This can shorten the delay when waiting for a good frame, since the delay for masking out a bad frame will be the integration time rather than the full-frame time.**	N	N
	9	0x0000	**mask_bad** **0 = The sensor will produce bad (corrupted) frames as a result of some register changes.** **1 = Bad (corrupted) frames are masked within the sensor by extending the vertical blanking time for the duration of the bad frame.**	N	N
	8	0x0000	**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	0x0000	**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	0x0001	**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	0x0000	**Reserved**		
	4	0x0001	**stdby_eof** **0 = Transition to standby is synchronized to the end of a sensor row readout (held-off until LINE_VALID has fallen).** **1 = Transition to standby is synchronized to the end of a frame.**	N	Y
	3	0x0001	**lock_reg** **Many SMIA registers that are specified as read-only are actually implemented as read/write registers. Clearing this bit allows such registers to be written.**	N	N
	2	0x0000	**stream** **Setting this bit places the sensor in streaming mode. Clearing this bit places the sensor in a low power mode. The result of clearing this bit depends upon the operating mode of the sensor.** **Entry and exit from streaming mode can also be controlled from the signal interface.**	Y	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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	**vert_flip** **This bit is an alias of R0x3040[15].**	Y	YM
	0	0x00	**horiz_mirror** **This bit is an alias of R0x3040[14].**	Y	YM
12318 R0x301E	15:0	0x002A	**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].				
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 Register Descriptions (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	0x0007	**standby_pin_select** **Associate the standby function with an active-high input pin** **0 = associate with GPI0** **1 = associate with GPI1** **2 = associate with GPI2** **3 = associate with GPI3** **4-6 = RESERVED** **7 = standby function cannot be controlled by any pin** **Must be set to 7 if reset[8]=0.**	N	N
	12:10	0x0007	**oe_n_pin_selct** **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	0x0007	**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	0x0007	**saddr_pin_select** **Associate the SADDR function with an active-high input pin** **0 = associate with GPI0** **1 = associate with GPI1** **2 = associate with GPI2** **3 = associate with GPI3** **4-6 = RESERVED** **7 = SADDR function is not controlled by any pin** **Must be set to 7 if R0x301A-B[8]=0.**	N	N
	3	RO	**gpi3** **Read-only. Return the current state of the GPI3 input pin. Invalid if R0x301A-B[8]=0.**	N	N
	2	RO	**gpi2** **Read-only. Return the current state of the GPI2 input pin. Invalid if R0x301A-B[8]=0.**	N	N
	1	RO	**gpi1** **Read-only. Return the current state of the GPI1 input pin. Invalid if R0x301A-B[8]=0.**	N	N
	0	RO	**gpi0** **Read-only. Return the current state of the GPI0 input pin. Invalid if R0x301A-B[8]=0.**	N	N
12328 R0x3028	15:0	0x0014	**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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12330 R0x302A	15:0	0x0014	**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	0x0014	**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	0x0014	**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	0x0014	**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 [11:8] are significant and are an alias of R0x3056[15:12].				
12340 R0x3034	15:0	0x0100	**digital_gain_red_ (R/W)**	Y	N
	Digital gain applied to red pixels on red/green rows of the pixel array. The value is an unsigned 8.8 fixed-point format. Bits [11:8] are significant and are an alias of R0x3056[15:12].				
12342 R0x3036	15:0	0x0100	**digital_gain_blue_ (R/W)**	Y	N
	Digital gain applied to blue pixels on blue/green rows of the pixel array. The value is an unsigned 8.8 fixed-point format. Bits [11:8] are significant and are an alias of R0x3056[15:12].				
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 [11:8] are significant and are an alias of R0x3056[15:12].				
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)**		
	15:2	X	**Reserved**		
	1	RO	**standby_status** **This bit tells you whether the sensor is in standby state. Can be polled after standby is entered to see when the real low-power state is entered; which can happen at the end of row or frame depending on bit 0x301A[4].**	N	N
	0	RO	**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 Register Descriptions (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	0x0000	**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	0x0000	**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	X	**Reserved**		
	12	0x0000	**reserved_y_sum_bin** **RESERVED Do not change from default**	Y	N
	11	0x0000	**x_bin_en** **Enable analogue binning in X (column) direction.** **When set, y_odd_inc must be set to 1 and x_odd_inc must be set to 3 for column binning or 7 for column skipping and binning, along with other register changes.**	Y	N
	10	0x0000	**xy_bin_en** **Enable analogue 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 binning or 7 for binning and skipping, along with other changes.**	Y	N
	9	0x0000	**low_power** **Enables low power mode. This will automatically half the pixel clock speed and the low power DAC values from R0x3EEC-R0x3EF1 will be used. Cannot be used when pc_speed[2:0] = 4.**	Y	YM
	8:6	0x0001	**x_odd_inc** **Increment applied to odd addresses in X (column) direction.** **1= Normal readout** **3 = Read out alternate pixel pairs to halve the amount of horizontal data in a frame.** **7 = Read out 1 of 4 pixel pairs to reduce the amount of horizontal data in a frame by 4.**	Y	YM
	5:0	0x0001	**y_odd_inc** **Increment applied to odd addresses in Y (row) direction.** **1= Normal readout** **3 = Read out alternate pixel pairs to halve the amount of vertical data in a frame.** **7 = Read out 1 of 4 pixel pairs to reduce the amount of vertical data in a frame by 4.** **15= Read out 1 out of 8 pixel pairs to reduce the amount of vertical data in a frame by 8.** **31 = Read out 1 out of 16 pixel pairs to reduce the amount of vertical data in a frame by 16.** **63 = Read out 1 out of 32 pixel pairs to reduce the amount of vertical data in a frame by 32.**	Y	YM

Table 6: Manufacturer-Specific Register Descriptions (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	0x0608	**flash (R/W)**		
	15	RO	**strobe** **Reflects the current state of the FLASH output signal. Read-only.**	N	N
	14	RO	**triggered** **Indicates that the FLASH output signal was asserted for the current frame. Read-only.**	N	N
	13	0x0000	**xenon_flash** **Enable Xenon flash. When set, the FLASH output signal will assert for the programmed period (bits [7:0]) during vertical blanking. This is achieved by keeping the integration time equal to one frame, and the pulse width less than the vertical blanking time.**	Y	N
	12:11	0x0000	**frame_delay** **Flash pulse delay measured in frames.**	N	N
	10	0x0001	**end_of_reset** **1 = In Xenon mode, the flash is triggered after resetting a frame.** **0 = In Xenon mode, the flash is triggered after a frame readout.**	N	N
	9	0x0001	**every_frame** **1 = Flash should be enabled every frame.** **0 = Flash should be enabled for 1 frame only.**	N	N
	8	0x0000	**led_flash** **Enable LED flash. When set, the FLASH output signal will assert prior to the start of the resetting of a frame and will remain asserted until the end of the frame readout.**	Y	Y
	7	0x0000	**invert_flash** **Invert flash output signal. When set, the FLASH output signal will be active low.**	N	N
	6	0x0000	**xenon_no_delay** **At the start of streaming, no frame delay will occur before the xenon flash pulse is triggered.**	N	N
	5	X	**Reserved**		
	4	0x0000	**trigger_timed** **1 = Enable Flash Count**	N	N
	3:0	0x0008	**Reserved**		
12360 R0x3048	15:0	0x0008	**flash_count (R/W)**		
	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 (0xFFFF), the flash pulse will automatically be truncated prior to the readout of the first row, giving the longest pulse possible.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12362 R0x304A	15:0	0x0000	**otpm_control (R/W)**		
	15:11	X	**Reserved**		
	10	0x0000	**enable_standby** **When set, enable otpm control logic to go into standby**	N	N
	9	0x0000	**single_record_only** **When set, automatic read sequence will end after one record is read out.**	N	N
	8	0x0000	**auto_rd_next_start** **To be used when single_record = 1 and bypass_record = 0. Triggers automatic OTPM read sequence to read the next record; bit 4 and bit 9 should both be set to 1.**	N	N
	7	X	**Reserved**		
	6	RO	**auto_rd_success** **Indicates whether the automatic read sequence was successful.**	N	N
	5	RO	**auto_rd_end** **Indicates whether the automatic read sequence is finished.**	N	N
	4	0x0000	**auto_rd_start** **Trigger automatic OTPM read sequence.** **When bypass_record=0, record(s) may be read out by record type (otpm_record[15:8]) or by OTPM address (otpm_config[8:0]). The payload of the record(s) will appear in the otpm_data* registers.** **When bypass_record = 1, the OTPM address to start from will be taken from otpm_config[8:0]. The length of the data to read is taken from otpm_record[7:0]. The data read from OTPM will appear in the otpm_data* registers.**	N	N
	3	0x0000	**otpm_control_disable_autord** **When register bit is set to 1, disable automatic OTPM read sequence.**	N	N
	2	RO	**auto_wr_success** **Indicates whether the automatic write sequence was successful.**	N	N
	1	RO	**auto_wr_end** **Indicates whether the automatic write sequence is finished.**	N	N
	0	0x0000	**auto_wr_start** **Trigger automatic OTPM write sequence. The high voltage must be available on the high voltage pad before this sequence is triggered. The OTPM address to start from will be taken from otpm_config[8:0]. The length of the data to program is taken from otpm_record[7:0]. The data is taken from the otpm_data* registers.**	N	N
12364 R0x304C	15:0	0x0200	**otpm_record (R/W)**	N	N
	15:8	0x0002	**auto_type** **Type of record. Currently supported types are:** **0x02 - Default registers loaded before software standby.** **0x1n - Default registers loaded after software standby.** **0x2n - Register sets.** **Other types are only available for readback from the OTPM RAM through the otpm_data_* registers.**	N	N
	7:0	0x0000	**auto_length** **Length of record payload in 16-bit words (between 1 and 255)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12366 R0x304E	15:0	0x0000	**otpm_status (RO)**		
	15:11	X	**Reserved**		
	10	RO	**otpm_insufficient** **Insufficient OTPM space to include a record**	N	N
	9	RO	**otpm_full** **OTPM is full**	N	N
	8	RO	**ded_parity_failure** **Double error-detect parity failure, data bad**	N	N
	7	RO	**sec_used** **ECC single bit error correction activated**	N	N
	6:1	RO	**ecc_check_bits** **Check bits produced by ECC**	N	N
	0	RO	**op_done** **Read/Write Operation complete**	N	N
12368 R0x3050	15:0	0x0000	**otpm_manual_control (R/W)**		
	15:7	X	**Reserved**		
	6	RO	**single_rd_success** **Indicates whether the single read sequence was successful.**	N	N
	5	RO	**single_rd_end** **Indicates whether the single read sequence is finished.**	N	N
	4	0x0000	**single_rd_start** **Trigger single OTPM read sequence from the OTPM address programmed in otpm_config[8:0].**	N	N
	3	X	**Reserved**		
	2	RO	**single_wr_success** **Indicates whether the single write sequence was successful.**	N	N
	1	RO	**single_wr_end** **Indicates whether the single write sequence is finished.**	N	N
	0	0x0000	**single_wr_start** **Trigger single OTPM write sequence. The high voltage must be available on the high voltage pad before this sequence is triggered. otpm_config[8:0] is the OTPM address that will be programmed with otpm_data_manual_*. A single read sequence will automatically be triggered for the same address. This can be used to determine whether the program was successful.**	N	N
12370 R0x3052	15:0	0x0000	**otpm_config (R/W)**		
	15	0x0000	**read_cfg_auto_scale_n** **Not used.**	N	N
	14	0x0000	**read_cfg_manual_scale** **Not used.**	N	N
	13:9	X	**Reserved**		
	8:0	0x0000	**single_address** **Address of the OTPM used for single writes/reads as well as auto writes/reads.**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12372 R0x3054	15:0	0x0000	**otpm_expr (R/W)**		
	15:12	X	**Reserved**		
	11	0x0000	**trigger_auto_ram_load** **Load current content of the OTPM RAM (otpm_data_*) to registers.**	N	N
	10	0x0000	**disable_auto_ram_load** **Disable automatic RAM load for record types supporting RAM load.**	N	N
	9	0x0000	**ecc_bypass** **When set the ECC logic will be bypassed.**	N	N
	8	0x0000	**bypass_record** **When enabled the record structure will be bypassed. Data in otpm_data* will be written directly to the OTPM.**	N	N
	7:3	X	**Reserved**		
	2	0x0000	**comp_test_en** **Not used.**	N	N
	1	0x0000	**vln_en_always_on** **Not used.**	N	N
	0	X	**Reserved**		
12374 R0x3056	15:0	0x1050	**green1_gain (R/W)**		
	15:12	0x0001	**digital_gain** **Digital Gain. Legal values 1-15**	Y	N
	11:10	0x0000	**col_gain** **This is actually the column gain. Valid values for bits[11:10] are** **00 : 1x** **01 : 3x** **10 : 2x** **11 : 4x**	N	N
	9:8	0x0000	**asc1_gain** **This is the ASC1 gain.** **Valid values for bits[9:8] are** **00 : 1x ,** **01 : 1.3x** **10 : 2x**	N	N
	7	0x0000	**Reserved**		
	6:0	0x0050	**initial_gain** **Initial gain = bits [6:0] * 1/32.**	Y	N
	Gain = Column Gain*ASC1 Gain* Initial_gain				

Table 6: Manufacturer-Specific Register Descriptions (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	0x1050	**blue_gain (R/W)**		
	15:12	0x0001	**digital_gain** **Digital Gain. Legal values 1-7.**	Y	N
	11:10	0x0000	**col_gain** **This is the column gain** **Valid values for bits[11:10] are** **00 : 1x** **01 : 3x** **10 : 2x** **11 : 4x**	N	N
	9:8	0x0000	**asc1_gain** **This is the ASC1 gain** **Valid values for bits[9:8] are** **00 : 1x** **01 : 1.3x** **10 : 2x**	N	N
	7	0x0000	**Reserved**		
	6:0	0x0050	**initial_gain** **Initial gain = bits [6:0] * 1/32.**	Y	N
	Gain = Column Gain*ASC1 Gain* Initial_gain				
12378 R0x305A	15:0	0x1050	**red_gain (R/W)**		
	15:12	0x0001	**digital_gain** **Digital Gain. Legal values 1-7.**	Y	N
	11:10	0x0000	**col_gain** **This is the column gain** **Valid values for bits[11:10] are** **00 : 1x** **01 : 3x** **10 : 2x** **11 : 4x**	N	N
	9:8	0x0000	**asc1_gain** **This is the ASC1 gain** **Valid values for bits[9:8] are** **00 : 1x** **01 : 1.3x** **10 : 2x**	N	N
	7	0x0000	**Reserved**		
	6:0	0x0050	**initial_gain** **Initial gain = bits [6:0] * 1/32.**	Y	N
	Gain = Column Gain*ASC1 Gain* Initial_gain				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12380 R0x305C	15:0	0x1050	**green2_gain (R/W)**		
	15:12	0x0001	**digital_gain** **Digital Gain. Legal values 1-7.**	Y	N
	11:10	0x0000	**col_gain** **This is the column gain** **Valid values for bits[11:10] are** **00 : 1x** **01 : 3x** **10 : 2x** **11 : 4x**	N	N
	9:8	0x0000	**asc1_gain** **This is the ASC1 gain** **Valid values for bits[9:8] are** **00 : 1x** **01 : 1.3x** **10 : 2x**	N	N
	7	0x0000	**Reserved**		
	6:0	0x0050	**initial_gain** **Initial gain = bits [6:0] * 1/32.**	Y	N
	Gain = Column Gain*ASC1 Gain* Initial_gain				
12382 R0x305E	15:0	0x1050	**global_gain (R/W)**		
	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	15:0	0x0000	**datapath_status (RO)**		
	15:6	X	**Reserved**		
	5	RO	**Reserved**		
	4	0x0000	**line_length_mismatch**	N	N
	3	0x0000	**frameover**	N	N
	2	0x0000	**lineover**	N	N
	1	0x0000	**fifo_overflow**	N	N
	0	0x0000	**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, behaviour 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 Register Descriptions (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	0xAC80	**datapath_select (R/W)**	N	N
	15:13	0x0005	**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 electro-magnetic emissions.**	N	N
	12:10	0x0003	**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	**profile** **SMIA Profile Mode:** **0 = Profile 0** **1 = Profile 1/2** **(this bit is read-only)**	N	N
	6	0x0000	**Reserved**		
	5	0x0000	**Reserved**		
	4	0x0000	**true_bayer** **Enables true Bayer scaling mode.**	N	N
	3:2	X	**Reserved**		
	1:0	0x0000	**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 1's test pattern (10 bit) 257 = Walking 1's test pattern (8 bit) other = Reserved.				
12402 R0x3072	15:0	0x0000	**test_data_red_ (R/W)**		
	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)**		
	The value for blue pixels in the bayer data used for the solid color test pattern and the test cursors.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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.				
12410 R0x307A	15:0	0x0000	**test_raw_mode (R/W)**		
	15:2	X	**Reserved**		
	1	0x0000	**Reserved**		
	0	0x0000	**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]				
12456 R0x30A8	15:0	0x0400	**calib_green1_asc1 (R/W)**	N	Y
	The calib_<color>_asc1 registers are associated with unused logic. They are similar to the calib_<color> registers, but only the frame average status values are useful.				
12458 R0x30AA	15:0	0x0400	**calib_blue_asc1 (R/W)**		
	The calib_<color>_asc1 registers are associated with unused logic. They are similar to the calib_<color> registers, but only the frame average status values are useful.				
12460 R0x30AC	15:0	0x0400	**calib_red_asc1 (R/W)**		
	The calib_<color>_asc1 registers are associated with unused logic. They are similar to the calib_<color> registers, but only the frame average status values are useful.				
12462 R0x30AE	15:0	0x0400	**calib_green2_asc1 (R/W)**	N	Y
	The calib_<color>_asc1 registers are associated with unused logic. They are similar to the calib_<color> registers, but only the frame average status values are useful.				
12476 R0x30BC	15:0	0x0400	**calib_global (R/W)**		
	When written all calib_<color><0/1> registers are updated (with the same value), when read value of calib_greenR (asc0) is returned.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12480 R0x30C0	15:0	0x0220	**calib_control (R/W)**	N	Y
	15:14	X	**Reserved**		
	13	0x0000	**force_recalculate** **When set, the offset calibration algorithm is triggered.**	Y	N
	12	0x0000	**recalculate** **When set, the offset calibration algorithm is triggered. This bit is write-1 but always reads back as 0.**	Y	N
	11:10	0x0000	**rd_calib_sel** **Selects what to read back from the calib_<color>* registers:** **00 = Offset calibration value.** **01 = Dark average.** **10 = Target.** **11 = Undefined.**	N	N
	9:8	0x0002	**rapid_step** **Factor to multiply offset calibration algorithm loop gain by:** **00 = 1x (default)** **01 = 2x** **10 = 4x** **11 = 0.5x**	N	N
	7	0x0000	**rapid_enable** **Not used.**	N	Y
	6:4	0x0002	**rapid_samples** **Number of samples used in the rapid offset calibration algorithm (after removing 1 max and 1 min pixel values per 8 used pixels).** **000 = 8 samples** **001 = 16 samples** **010 = 32 samples** **011 = 64 samples** **100 = 128 samples** **101 = 256 samples**	N	Y
	3	0x0000	**same_asc** **When this bit is set, values of the same color independent of which ASC they correspond to will be used to generate the offset calibration setting.**	N	Y
	2	0x0000	**same_redblue** **When this bit is set, the same calibration value will be used for red and blue pixels: Calib blue = calib red.**	N	Y
	1	0x0000	**same_green** **When this bit is set, the same calibration value will be used for all green pixels: Calib green2 = calib green1.**	N	Y
	0	0x0000	**manual_override** **0 = Normal operation** **1 = Override automatic offset calibration values with values programmed into R0x30C2-R0x30C8 and R0x30A8-R0x30AE.**	N	Y

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12482 R0x30C2	15:0	0x0400	**calib_green1 (R/W)**	N	Y
	Calibration target and offset register for green1 pixels in ASC0. Green1 pixels share rows with red pixels. The value in the register depends on the rd_calib_sel setting in calib_control: 00 = Analog offset calibration value represented as a two's complement signed 8-bit value (if [8] is clear, the offset is positive and the magnitude is given by [7:0]. If [8] is set, the offset is negative and the magnitude is given by not([7:0]) + 1). If R0x30C0[0] = 0, this register is read-only and returns the current value computed by the offset calibration algorithm. If R0x30C0[0] = 1, this register is read/write, and can be used to set the calibration offset manually. 01 = Dark average. Average of the offset calibration pixels used by the offset calibration algorithm. 10 = Target value. Corresponds to the ADC code the offset calibration pixels should correspond to. 11 = Undefined.				
12484 R0x30C4	15:0	0x0400	**calib_blue (R/W)**	N	Y
	Calibration target and offset register for blue pixels in ASC0. The value in the register depends on the rd_calib_sel setting in calib_control: 00 = Analog offset calibration value represented as a two's complement signed 8-bit value (if [8] is clear, the offset is positive and the magnitude is given by [7:0]. If [8] is set, the offset is negative and the magnitude is given by not([7:0]) + 1). If R0x30C0[0] = 0, this register is read-only and returns the current value computed by the offset calibration algorithm. If R0x30C0[0] = 1, this register is read/write, and can be used to set the calibration offset manually. 01 = Dark average. Average of the offset calibration pixels used by the offset calibration algorithm. 10 = Target value. Corresponds to the ADC code the offset calibration pixels should correspond to. 11 = Undefined.				
12486 R0x30C6	15:0	0x0400	**calib_red (R/W)**	N	Y
	Calibration target and offset register for red pixels in ASC0. The value in the register depends on the rd_calib_sel setting in calib_control: 00 = Analog offset calibration value represented as a two's complement signed 8-bit value (if [8] is clear, the offset is positive and the magnitude is given by [7:0]. If [8] is set, the offset is negative and the magnitude is given by not([7:0]) + 1). If R0x30C0[0] = 0, this register is read-only and returns the current value computed by the offset calibration algorithm. If R0x30C0[0] = 1, this register is read/write, and can be used to set the calibration offset manually. 01 = Dark average. Average of the offset calibration pixels used by the offset calibration algorithm. 10 = Target value. Corresponds to the ADC code the offset calibration pixels should correspond to. 11 = Undefined.				
12488 R0x30C8	15:0	0x0400	**calib_green2 (R/W)**	N	Y
	Calibration target and offset register for green2 pixels in ASC0. Green2 pixels share rows with blue pixels. The value in the register depends on the rd_calib_sel setting in calib_control: 00 = Analog offset calibration value represented as a two's complement signed 8-bit value (if [8] is clear, the offset is positive and the magnitude is given by [7:0]. If [8] is set, the offset is negative and the magnitude is given by not([7:0]) + 1). If R0x30C0[0] = 0, this register is read-only and returns the current value computed by the offset calibration algorithm. If R0x30C0[0] = 1, this register is read/write, and can be used to set the calibration offset manually. 01 = Dark average. Average of the offset calibration pixels used by the offset calibration algorithm. 10 = Target value. Corresponds to the ADC code the offset calibration pixels should correspond to. 11 = Undefined.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12528 R0x30F0	15:0	0x0000	**vcm_control (R/W)**	N	N
	15	0x0000	**vcm_en** **Disables VCM driver.**	N	N
	14:12	X	**Reserved**		
	11:8	0x0000	**vcm_ctrl** **Bit8: null code feature enable** **0 = disabled (default), enable VCM driver regardless of VCM_DATA<7:0>** **1 = enabled, shutdown Vcm driver when VCM_DATA<7:0>=0** **Bit9 = low TC bias current enable** **0 = disabled (default), CTAT current, -700ppm/degC** **1 = enabled, PTAT/CTAT mix, ~ -150ppm/degC**	N	N
	7:5	X	**Reserved**		
	4	0x0000	**disable_pd** **Disable vcm power down. It overrides disabling the VCM when streaming is turned off. Set this bit with vcm_en (bit 15)=1 to prevent the vcm dac codes from resetting. Note also analog_control6 bit 13, ana_power_up, needs to be set when streaming is turned off to ensure the bias currents to vcm is not removed.**	N	N
	3:0	0x0000	**vcm_slew** **Programmable counter to define the mode and the step transition time to refresh the target code to VCM DAC.** **vcm_slew[2:0] = 0: mode 0, refresh the code directly to target code** **vcm_slew[2:0] > 0: mode 1, increment/decrement 1 code every step transition time to target code** **step transition time = Tsysclk * 16 * (vcm_step_time[15:0] + 1) * (2 ^ (vcm_slew[2:0] -1))**	N	N
12530 R0x30F2	15:0	0x0000	**vcm_new_code (R/W)**	N	N
	New target code to VCM DAC.				
12532 R0x30F4	15:0	0x0000	**vcm_step_time (R/W)**	N	N
	Programmable counter to define how many system clocks for each step time. vcm step time = Tsysclk * 16 * (vcm_step_time[15:0] + 1)				
12544 R0x3100	15:0	0x0002	**adacd_control (R/W)**		
	15:2	X	**Reserved**		
	1	0x0001	**adacd_control_filter_en** **Setting this bit enables the adacd**	N	N
	0	0x0000	**adacd_control_low_light** **Setting this bit initiates enables the low light mode of adacd. In this mode filter window sizes are set as following: 5x9 for G, 5x14 for R and B (as opposed to regular light condition where the filter window sizes are set as 5x5 for G and 5x10 for R and B**	N	N
	reserved				
12546 R0x3102	15:0	0x00A0	**adacd_noise_model1 (R/W)**		
	Value of noise model coefficient to be programmed				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12548 R0x3104	15:0	0x0B6D	**adacd_noise_model2 (R/W)**	N	N
	15:12	X	**Reserved**		
	11:6	0x002D	**adacd_noise_model2_ratio_bg_code** **noise ratio parameter that reflects the ratio between the noise levels of the difference/sum signal and the blue signal.**	N	N
	5:0	0x002D	**adacd_noise_model2_ratio_rg_code** **noise ratio parameter that reflects the ratio between the noise levels of the difference/sum signal and the red signal.**	N	N
	ADACD_NOISE_MODEL2				
12550 R0x3106	15:0	0x0604	**adacd_noise_floor1 (R/W)**	N	N
	Noise floor values are used internally in sensor based on if(g1_analog_gain < adacd_gain_threshold_0) then noise_floor = noise_floor1[7:0], if(g1_analog_gain < adacd_gain_threshold_1) then noise_floor = noise_floor1[15:8], if(g1_analog_gain < adacd_gain_threshold_2) then noise_floor = noise_floor2[7:0], else noise_floor = noise_floor2[15:8]				
12552 R0x3108	15:0	0x120A	**adacd_noise_floor2 (R/W)**		
	Noise floor values are used internally in sensor based on if(g1_analog_gain < adacd_gain_threshold_0) then noise_floor = noise_floor1[7:0], if(g1_analog_gain < adacd_gain_threshold_1) then noise_floor = noise_floor1[15:8], if(g1_analog_gain < adacd_gain_threshold_2) then noise_floor = noise_floor2[7:0], else noise_floor = noise_floor2[15:8]				
12554 R0x310A	15:0	0x002A	**adacd_pedestal (R/W)**	N	N
	ADACD_PEDESTAL				
12556 R0x310C	15:0	0x0080	**adacd_gain_threshold_0 (R/W)**	N	N
	adacd_gain_threshold_0				
12558 R0x310E	15:0	0x0100	**adacd_gain_threshold_1 (R/W)**	N	N
	adacd_gain_threshold_1				
12560 R0x3110	15:0	0x0200	**adacd_gain_threshold_2 (R/W)**	N	N
	adacd_gain_threshold_2				
12592 R0x3130	15:0	0x7801	**otpm_tcfg_write_01 (R/W)**		
	15:8	0x0078	**write_program** **Duration of TMG_WR_PROGRAM state in the RTL module otpm_core_tmg_40is.** **This value multiplied by 256 equals the duration hvstate equals 3'b110.**	N	N
	7:4	0x0000	**write_pre2** **Duration of TMG_WR_PRE2 state in the RTL module otpm_core_tmg_40is.** **This equals the delay between vcmn_write going high and the address being assigned to the OTPM addr port.**	N	N
	3:0	0x0001	**write_pre1** **Duration of TMG_WR_PRE1 state in the RTL module otpm_core_tmg_40is.** **This equals the delay between hvstate=3'b010 and vcmn_write going high.**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12596 R0x3134	15:0	0x0D95	**otpm_tcfg_read_01 (R/W)**		
	15	0x0000	**read_grab** **This bit determines when to grab the dout data from the OTPM:** **0 = dout is sampled when dout_ready goes high.** **1 = dout is sampled after a time determined by bits 14:12.**	N	N
	14:12	0x0000	**read_read** **When bit 15 is set these bits multiplied by 8 determines the duration of TMG_RD_READ state in the RTL module otpm_core_tmg_40is.** **This equals the delay between hvstate=3'b101 and the dout data from the OTPM is grabbed.**	N	N
	11:8	0x000D	**read_post3** **Duration of TMG_WR_POST3 state in the RTL module otpm_core_tmg_40is.** **This equals the delay between vcmn_standby going high and hvstate=3'b000.**	N	N
	7:4	0x0009	**read_post2** **Duration of TMG_RD_POST2 state in the RTL module otpm_core_tmg_40is.** **This equals the delay between vcmn_read going low and vcmn_standby going high.**	N	N
	3:0	0x0005	**read_start2** **Duration of TMG_RD_START2 state in the RTL module otpm_core_tmg_40is.** **This equals the delay between vcmn_read going high and hvstate=3'b001.**	N	N
12704 R0x31A0	15:0	0x0101	**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	**descriptor_1 (RO)**		
	The value of this descriptor indicates that a single-lane MIPI interface is available on this device.				
12708 R0x31A4	15:0	0x0202	**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	0x0000	**descriptor_3 (RO)**	N	N
	The value of this descriptor indicates that it is unused.				
12712 R0x31A8	15:0	0x0000	**descriptor_4 (RO)**	N	N
	The value of this descriptor indicates that it is unused.				
12714 R0x31AA	15:0	0x0000	**descriptor_5 (RO)**		
	The value of this descriptor indicates that it is unused.				
12716 R0x31AC	15:0	0x0000	**descriptor_6 (RO)**		
	The value of this descriptor indicates that it is unused.				
12718 R0x31AE	15:0	0x0001	**serial_format (R/W)**	Y	N
	When the serial interface is enabled (reset_register[12]=0), this register controls which serial interface is in use. The upper byte of this register (interface_type[9:8]) is read-only when lock_bit[15] has been set. (lock_bit[15] is write-only, reads zero, resets after hard/soft chip reset). The lower byte is read/write.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12720 R0x31B0	15:0	0x0070	**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	0x0049	**line_preamble (R/W)**		
	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.				
12724 R0x31B4	15:0	0x0C67	**mipi_timing_0 (R/W)**	N	N
	15:12	X	**Reserved**		
	11:8	0x000C	**t_hs_zero** **Time, in op_pix_clk periods, to drive HS-0 before the sync sequence**	N	N
	7:4	0x0006	**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	0x0007	**t_clk_trail** **Time, in op_pix_clk periods, to drive HS differential state after last payload clock bit of an HS transmission burst**	N	N
12726 R0x31B6	15:0	0x0310	**mipi_timing_1 (R/W)**		
	15:12	X	**Reserved**		
	11:6	0x000C	**t_hs_exit** **Time, in op_pix_clk periods, to drive LP-11 after HS burst**	N	N
	5:0	0x0010	**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	0x030E	**mipi_timing_2 (R/W)**	N	N
	15:12	X	**Reserved**		
	11:6	0x000C	**t_clk_pre** **Time, in op_pix_clk periods, to drive the HS clock before any data lane might start up**	N	N
	5:0	0x000E	**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

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
12730 R0x31BA	15:0	0x030D	**mipi_timing_3 (R/W)**	N	N
	15:13	X	**Reserved**		
	12:7	0x0006	**t_lpx** **Time, in op_pix_clk periods, of any low-power state period**	N	N
	6:0	0x000D	**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)**	N	N
	15	X	**Reserved**		
	14	0x0000	**heavy_lp_load_in** **Control of the port heavy_lp_load_in on the mipiccp PHY.**	N	N
	13:12	0x0000	**delay_trim_d2** **Control of the port delay_trim_d2 on the mipiccp PHY.**	N	N
	11:10	0x0000	**delay_trim_d1** **Control of the port delay_trim_d1 on the mipiccp PHY.**	N	N
	9:8	0x0000	**delay_trim_d0** **Control of the port delay_trim_d0 on the mipiccp PHY.**	N	N
	7	X	**Reserved**		
	6:0	0x000B	**t_init** **Initialisation time when first entering stop state (LP-11) after powerup or reset. LP-11 is transmitted for a minimum of (1024) * T_INIT * op_pix_clk.**	N	N
12744 R0x31C8	15:0	0x0000	**hispi_crc_0 (R/W)**	N	N
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)**		
	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)**		
	Specify the width, in columns, of the vertical test cursor. A width of 0 disables the cursor.				
12786 R0x31F2	15:0	0x6E6C	**i2c_ids_mipi_default (WO)**		
	Programmable two-wire serial interface slave addresses for MIPI operation.				
12796 R0x31FC	15:0	0x3020	**i2c_ids (R/W)**		
	I2C 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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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)**		
	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)**		
	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)**		
	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)**		
	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)**		
	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)**		
	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)**		
	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)**		
	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)**		
	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 Register Descriptions (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)**		
	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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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.				
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)**		
	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)**		
	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)**		
	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)**		
	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)**		
	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)**		
	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)**		
	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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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.				
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)**		
	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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
13978 R0x369A	15:0	0x0000	**p_bl_p2q3 (R/W)**		
	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.				
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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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.				
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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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.				
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.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
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)**	N	N
	15	0x0000	**enable** **Turn on lens shading correction.**	N	N
	14:0	X	**Reserved**		
	When LSC_EN bit is set lens shading correction will generate polynomial function and correct stream of pixels. When not set LSC 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.				
14272 R0x37C0	15:0	0x0000	**p_gr_q5 (R/W)**	N	N
	Parameter for parabolic roll-off algorithm for greenR pixels.				
14274 R0x37C2	15:0	0x0000	**p_rd_q5 (R/W)**	N	N
	Parameter for parabolic roll-off algorithm for red pixels.				
14276 R0x37C4	15:0	0x0000	**p_bl_q5 (R/W)**	N	N
	Parameter for parabolic roll-off algorithm for blue pixels.				

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14278 R0x37C6	15:0	0x0000	**p_gb_q5 (R/W)**	N	N
	Parameter for parabolic roll-off algorithm for greenB pixels.				
14336 R0x3800	15:0	0x0000	**otpm_data_0 (R/W)**	N	N
	Corresponds to OTPM RAM address 0. OTPM data to be written to the OTPM through auto write should be stored here. After OTPM auto read the data will be available here. The exception is for record using automatic upload to registers. In this case the OTPM data will not be available in the OTPM RAM unless the automatic upload feature is turned off.				
14338 R0x3802	15:0	0x0000	**otpm_data_1 (R/W)**	N	N
14340 R0x3804	15:0	0x0000	**otpm_data_2 (R/W)**	N	N
14342 R0x3806	15:0	0x0000	**otpm_data_3 (R/W)**	N	N
14344 R0x3808	15:0	0x0000	**otpm_data_4 (R/W)**	N	N
14346 R0x380A	15:0	0x0000	**otpm_data_5 (R/W)**	N	N
14348 R0x380C	15:0	0x0000	**otpm_data_6 (R/W)**	N	N
14350 R0x380E	15:0	0x0000	**otpm_data_7 (R/W)**	N	N
14352 R0x3810	15:0	0x0000	**otpm_data_8 (R/W)**	N	N
14354 R0x3812	15:0	0x0000	**otpm_data_9 (R/W)**	N	N
14356 R0x3814	15:0	0x0000	**otpm_data_10 (R/W)**	N	N
14358 R0x3816	15:0	0x0000	**otpm_data_11 (R/W)**	N	N
14360 R0x3818	15:0	0x0000	**otpm_data_12 (R/W)**	N	N
14362 R0x381A	15:0	0x0000	**otpm_data_13 (R/W)**	N	N
14364 R0x381C	15:0	0x0000	**otpm_data_14 (R/W)**	N	N
14366 R0x381E	15:0	0x0000	**otpm_data_15 (R/W)**	N	N
14368 R0x3820	15:0	0x0000	**otpm_data_16 (R/W)**	N	N
14370 R0x3822	15:0	0x0000	**otpm_data_17 (R/W)**	N	N
14372 R0x3824	15:0	0x0000	**otpm_data_18 (R/W)**	N	N
14374 R0x3826	15:0	0x0000	**otpm_data_19 (R/W)**	N	N
14376 R0x3828	15:0	0x0000	**otpm_data_20 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14378 R0x382A	15:0	0x0000	**otpm_data_21 (R/W)**	N	N
14380 R0x382C	15:0	0x0000	**otpm_data_22 (R/W)**	N	N
14382 R0x382E	15:0	0x0000	**otpm_data_23 (R/W)**	N	N
14384 R0x3830	15:0	0x0000	**otpm_data_24 (R/W)**	N	N
14386 R0x3832	15:0	0x0000	**otpm_data_25 (R/W)**	N	N
14388 R0x3834	15:0	0x0000	**otpm_data_26 (R/W)**	N	N
14390 R0x3836	15:0	0x0000	**otpm_data_27 (R/W)**	N	N
14392 R0x3838	15:0	0x0000	**otpm_data_28 (R/W)**	N	N
14394 R0x383A	15:0	0x0000	**otpm_data_29 (R/W)**	N	N
14396 R0x383C	15:0	0x0000	**otpm_data_30 (R/W)**	N	N
14398 R0x383E	15:0	0x0000	**otpm_data_31 (R/W)**	N	N
14400 R0x3840	15:0	0x0000	**otpm_data_32 (R/W)**	N	N
14402 R0x3842	15:0	0x0000	**otpm_data_33 (R/W)**	N	N
14404 R0x3844	15:0	0x0000	**otpm_data_34 (R/W)**	N	N
14406 R0x3846	15:0	0x0000	**otpm_data_35 (R/W)**	N	N
14408 R0x3848	15:0	0x0000	**otpm_data_36 (R/W)**	N	N
14410 R0x384A	15:0	0x0000	**otpm_data_37 (R/W)**	N	N
14412 R0x384C	15:0	0x0000	**otpm_data_38 (R/W)**	N	N
14414 R0x384E	15:0	0x0000	**otpm_data_39 (R/W)**	N	N
14416 R0x3850	15:0	0x0000	**otpm_data_40 (R/W)**	N	N
14418 R0x3852	15:0	0x0000	**otpm_data_41 (R/W)**	N	N
14420 R0x3854	15:0	0x0000	**otpm_data_42 (R/W)**	N	N
14422 R0x3856	15:0	0x0000	**otpm_data_43 (R/W)**	N	N
14424 R0x3858	15:0	0x0000	**otpm_data_44 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14426 R0x385A	15:0	0x0000	**otpm_data_45 (R/W)**	N	N
14428 R0x385C	15:0	0x0000	**otpm_data_46 (R/W)**	N	N
14430 R0x385E	15:0	0x0000	**otpm_data_47 (R/W)**	N	N
14432 R0x3860	15:0	0x0000	**otpm_data_48 (R/W)**	N	N
14434 R0x3862	15:0	0x0000	**otpm_data_49 (R/W)**	N	N
14436 R0x3864	15:0	0x0000	**otpm_data_50 (R/W)**	N	N
14438 R0x3866	15:0	0x0000	**otpm_data_51 (R/W)**	N	N
14440 R0x3868	15:0	0x0000	**otpm_data_52 (R/W)**	N	N
14442 R0x386A	15:0	0x0000	**otpm_data_53 (R/W)**	N	N
14444 R0x386C	15:0	0x0000	**otpm_data_54 (R/W)**	N	N
14446 R0x386E	15:0	0x0000	**otpm_data_55 (R/W)**	N	N
14448 R0x3870	15:0	0x0000	**otpm_data_56 (R/W)**	N	N
14450 R0x3872	15:0	0x0000	**otpm_data_57 (R/W)**	N	N
14452 R0x3874	15:0	0x0000	**otpm_data_58 (R/W)**		
14454 R0x3876	15:0	0x0000	**otpm_data_59 (R/W)**	N	N
14456 R0x3878	15:0	0x0000	**otpm_data_60 (R/W)**	N	N
14458 R0x387A	15:0	0x0000	**otpm_data_61 (R/W)**	N	N
14460 R0x387C	15:0	0x0000	**otpm_data_62 (R/W)**	N	N
14462 R0x387E	15:0	0x0000	**otpm_data_63 (R/W)**	N	N
14464 R0x3880	15:0	0x0000	**otpm_data_64 (R/W)**	N	N
14466 R0x3882	15:0	0x0000	**otpm_data_65 (R/W)**	N	N
14468 R0x3884	15:0	0x0000	**otpm_data_66 (R/W)**	N	N
14470 R0x3886	15:0	0x0000	**otpm_data_67 (R/W)**	N	N
14472 R0x3888	15:0	0x0000	**otpm_data_68 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14474 R0x388A	15:0	0x0000	**otpm_data_69 (R/W)**	N	N
14476 R0x388C	15:0	0x0000	**otpm_data_70 (R/W)**	N	N
14478 R0x388E	15:0	0x0000	**otpm_data_71 (R/W)**	N	N
14480 R0x3890	15:0	0x0000	**otpm_data_72 (R/W)**	N	N
14482 R0x3892	15:0	0x0000	**otpm_data_73 (R/W)**	N	N
14484 R0x3894	15:0	0x0000	**otpm_data_74 (R/W)**	N	N
14486 R0x3896	15:0	0x0000	**otpm_data_75 (R/W)**	N	N
14488 R0x3898	15:0	0x0000	**otpm_data_76 (R/W)**	N	N
14490 R0x389A	15:0	0x0000	**otpm_data_77 (R/W)**	N	N
14492 R0x389C	15:0	0x0000	**otpm_data_78 (R/W)**	N	N
14494 R0x389E	15:0	0x0000	**otpm_data_79 (R/W)**	N	N
14496 R0x38A0	15:0	0x0000	**otpm_data_80 (R/W)**	N	N
14498 R0x38A2	15:0	0x0000	**otpm_data_81 (R/W)**	N	N
14500 R0x38A4	15:0	0x0000	**otpm_data_82 (R/W)**	N	N
14502 R0x38A6	15:0	0x0000	**otpm_data_83 (R/W)**	N	N
14504 R0x38A8	15:0	0x0000	**otpm_data_84 (R/W)**	N	N
14506 R0x38AA	15:0	0x0000	**otpm_data_85 (R/W)**	N	N
14508 R0x38AC	15:0	0x0000	**otpm_data_86 (R/W)**	N	N
14510 R0x38AE	15:0	0x0000	**otpm_data_87 (R/W)**	N	N
14512 R0x38B0	15:0	0x0000	**otpm_data_88 (R/W)**	N	N
14514 R0x38B2	15:0	0x0000	**otpm_data_89 (R/W)**	N	N
14516 R0x38B4	15:0	0x0000	**otpm_data_90 (R/W)**	N	N
14518 R0x38B6	15:0	0x0000	**otpm_data_91 (R/W)**	N	N
14520 R0x38B8	15:0	0x0000	**otpm_data_92 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14522 R0x38BA	15:0	0x0000	**otpm_data_93 (R/W)**	N	N
14524 R0x38BC	15:0	0x0000	**otpm_data_94 (R/W)**	N	N
14526 R0x38BE	15:0	0x0000	**otpm_data_95 (R/W)**	N	N
14528 R0x38C0	15:0	0x0000	**otpm_data_96 (R/W)**	N	N
14530 R0x38C2	15:0	0x0000	**otpm_data_97 (R/W)**	N	N
14532 R0x38C4	15:0	0x0000	**otpm_data_98 (R/W)**	N	N
14534 R0x38C6	15:0	0x0000	**otpm_data_99 (R/W)**	N	N
14536 R0x38C8	15:0	0x0000	**otpm_data_100 (R/W)**	N	N
14538 R0x38CA	15:0	0x0000	**otpm_data_101 (R/W)**	N	N
14540 R0x38CC	15:0	0x0000	**otpm_data_102 (R/W)**	N	N
14542 R0x38CE	15:0	0x0000	**otpm_data_103 (R/W)**	N	N
14544 R0x38D0	15:0	0x0000	**otpm_data_104 (R/W)**	N	N
14546 R0x38D2	15:0	0x0000	**otpm_data_105 (R/W)**	N	N
14548 R0x38D4	15:0	0x0000	**otpm_data_106 (R/W)**	N	N
14550 R0x38D6	15:0	0x0000	**otpm_data_107 (R/W)**	N	N
14552 R0x38D8	15:0	0x0000	**otpm_data_108 (R/W)**	N	N
14554 R0x38DA	15:0	0x0000	**otpm_data_109 (R/W)**	N	N
14556 R0x38DC	15:0	0x0000	**otpm_data_110 (R/W)**	N	N
14558 R0x38DE	15:0	0x0000	**otpm_data_111 (R/W)**	N	N
14560 R0x38E0	15:0	0x0000	**otpm_data_112 (R/W)**	N	N
14562 R0x38E2	15:0	0x0000	**otpm_data_113 (R/W)**	N	N
14564 R0x38E4	15:0	0x0000	**otpm_data_114 (R/W)**	N	N
14566 R0x38E6	15:0	0x0000	**otpm_data_115 (R/W)**	N	N
14568 R0x38E8	15:0	0x0000	**otpm_data_116 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14570 R0x38EA	15:0	0x0000	**otpm_data_117 (R/W)**	N	N
14572 R0x38EC	15:0	0x0000	**otpm_data_118 (R/W)**	N	N
14574 R0x38EE	15:0	0x0000	**otpm_data_119 (R/W)**	N	N
14576 R0x38F0	15:0	0x0000	**otpm_data_120 (R/W)**	N	N
14578 R0x38F2	15:0	0x0000	**otpm_data_121 (R/W)**	N	N
14580 R0x38F4	15:0	0x0000	**otpm_data_122 (R/W)**	N	N
14582 R0x38F6	15:0	0x0000	**otpm_data_123 (R/W)**	N	N
14584 R0x38F8	15:0	0x0000	**otpm_data_124 (R/W)**	N	N
14586 R0x38FA	15:0	0x0000	**otpm_data_125 (R/W)**	N	N
14588 R0x38FC	15:0	0x0000	**otpm_data_126 (R/W)**	N	N
14590 R0x38FE	15:0	0x0000	**otpm_data_127 (R/W)**	N	N
14592 R0x3900	15:0	0x0000	**otpm_data_128 (R/W)**	N	N
14594 R0x3902	15:0	0x0000	**otpm_data_129 (R/W)**	N	N
14596 R0x3904	15:0	0x0000	**otpm_data_130 (R/W)**	N	N
14598 R0x3906	15:0	0x0000	**otpm_data_131 (R/W)**	N	N
14600 R0x3908	15:0	0x0000	**otpm_data_132 (R/W)**	N	N
14602 R0x390A	15:0	0x0000	**otpm_data_133 (R/W)**	N	N
14604 R0x390C	15:0	0x0000	**otpm_data_134 (R/W)**	N	N
14606 R0x390E	15:0	0x0000	**otpm_data_135 (R/W)**	N	N
14608 R0x3910	15:0	0x0000	**otpm_data_136 (R/W)**	N	N
14610 R0x3912	15:0	0x0000	**otpm_data_137 (R/W)**	N	N
14612 R0x3914	15:0	0x0000	**otpm_data_138 (R/W)**	N	N
14614 R0x3916	15:0	0x0000	**otpm_data_139 (R/W)**	N	N
14616 R0x3918	15:0	0x0000	**otpm_data_140 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14618 R0x391A	15:0	0x0000	**otpm_data_141 (R/W)**	N	N
14620 R0x391C	15:0	0x0000	**otpm_data_142 (R/W)**	N	N
14622 R0x391E	15:0	0x0000	**otpm_data_143 (R/W)**	N	N
14624 R0x3920	15:0	0x0000	**otpm_data_144 (R/W)**	N	N
14626 R0x3922	15:0	0x0000	**otpm_data_145 (R/W)**	N	N
14628 R0x3924	15:0	0x0000	**otpm_data_146 (R/W)**	N	N
14630 R0x3926	15:0	0x0000	**otpm_data_147 (R/W)**	N	N
14632 R0x3928	15:0	0x0000	**otpm_data_148 (R/W)**	N	N
14634 R0x392A	15:0	0x0000	**otpm_data_149 (R/W)**	N	N
14636 R0x392C	15:0	0x0000	**otpm_data_150 (R/W)**	N	N
14638 R0x392E	15:0	0x0000	**otpm_data_151 (R/W)**	N	N
14640 R0x3930	15:0	0x0000	**otpm_data_152 (R/W)**	N	N
14642 R0x3932	15:0	0x0000	**otpm_data_153 (R/W)**	N	N
14644 R0x3934	15:0	0x0000	**otpm_data_154 (R/W)**	N	N
14646 R0x3936	15:0	0x0000	**otpm_data_155 (R/W)**	N	N
14648 R0x3938	15:0	0x0000	**otpm_data_156 (R/W)**	N	N
14650 R0x393A	15:0	0x0000	**otpm_data_157 (R/W)**	N	N
14652 R0x393C	15:0	0x0000	**otpm_data_158 (R/W)**	N	N
14654 R0x393E	15:0	0x0000	**otpm_data_159 (R/W)**	N	N
14656 R0x3940	15:0	0x0000	**otpm_data_160 (R/W)**	N	N
14658 R0x3942	15:0	0x0000	**otpm_data_161 (R/W)**	N	N
14660 R0x3944	15:0	0x0000	**otpm_data_162 (R/W)**	N	N
14662 R0x3946	15:0	0x0000	**otpm_data_163 (R/W)**	N	N
14664 R0x3948	15:0	0x0000	**otpm_data_164 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14666 R0x394A	15:0	0x0000	**otpm_data_165 (R/W)**	N	N
14668 R0x394C	15:0	0x0000	**otpm_data_166 (R/W)**	N	N
14670 R0x394E	15:0	0x0000	**otpm_data_167 (R/W)**	N	N
14672 R0x3950	15:0	0x0000	**otpm_data_168 (R/W)**	N	N
14674 R0x3952	15:0	0x0000	**otpm_data_169 (R/W)**	N	N
14676 R0x3954	15:0	0x0000	**otpm_data_170 (R/W)**	N	N
14678 R0x3956	15:0	0x0000	**otpm_data_171 (R/W)**	N	N
14680 R0x3958	15:0	0x0000	**otpm_data_172 (R/W)**	N	N
14682 R0x395A	15:0	0x0000	**otpm_data_173 (R/W)**	N	N
14684 R0x395C	15:0	0x0000	**otpm_data_174 (R/W)**	N	N
14686 R0x395E	15:0	0x0000	**otpm_data_175 (R/W)**	N	N
14688 R0x3960	15:0	0x0000	**otpm_data_176 (R/W)**	N	N
14690 R0x3962	15:0	0x0000	**otpm_data_177 (R/W)**	N	N
14692 R0x3964	15:0	0x0000	**otpm_data_178 (R/W)**	N	N
14694 R0x3966	15:0	0x0000	**otpm_data_179 (R/W)**	N	N
14696 R0x3968	15:0	0x0000	**otpm_data_180 (R/W)**	N	N
14698 R0x396A	15:0	0x0000	**otpm_data_181 (R/W)**	N	N
14700 R0x396C	15:0	0x0000	**otpm_data_182 (R/W)**	N	N
14702 R0x396E	15:0	0x0000	**otpm_data_183 (R/W)**	N	N
14704 R0x3970	15:0	0x0000	**otpm_data_184 (R/W)**	N	N
14706 R0x3972	15:0	0x0000	**otpm_data_185 (R/W)**	N	N
14708 R0x3974	15:0	0x0000	**otpm_data_186 (R/W)**	N	N
14710 R0x3976	15:0	0x0000	**otpm_data_187 (R/W)**	N	N
14712 R0x3978	15:0	0x0000	**otpm_data_188 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14714 R0x397A	15:0	0x0000	**otpm_data_189 (R/W)**	N	N
14716 R0x397C	15:0	0x0000	**otpm_data_190 (R/W)**	N	N
14718 R0x397E	15:0	0x0000	**otpm_data_191 (R/W)**	N	N
14720 R0x3980	15:0	0x0000	**otpm_data_192 (R/W)**	N	N
14722 R0x3982	15:0	0x0000	**otpm_data_193 (R/W)**	N	N
14724 R0x3984	15:0	0x0000	**otpm_data_194 (R/W)**	N	N
14726 R0x3986	15:0	0x0000	**otpm_data_195 (R/W)**	N	N
14728 R0x3988	15:0	0x0000	**otpm_data_196 (R/W)**	N	N
14730 R0x398A	15:0	0x0000	**otpm_data_197 (R/W)**	N	N
14732 R0x398C	15:0	0x0000	**otpm_data_198 (R/W)**	N	N
14734 R0x398E	15:0	0x0000	**otpm_data_199 (R/W)**	N	N
14736 R0x3990	15:0	0x0000	**otpm_data_200 (R/W)**	N	N
14738 R0x3992	15:0	0x0000	**otpm_data_201 (R/W)**	N	N
14740 R0x3994	15:0	0x0000	**otpm_data_202 (R/W)**	N	N
14742 R0x3996	15:0	0x0000	**otpm_data_203 (R/W)**	N	N
14744 R0x3998	15:0	0x0000	**otpm_data_204 (R/W)**	N	N
14746 R0x399A	15:0	0x0000	**otpm_data_205 (R/W)**	N	N
14748 R0x399C	15:0	0x0000	**otpm_data_206 (R/W)**	N	N
14750 R0x399E	15:0	0x0000	**otpm_data_207 (R/W)**	N	N
14752 R0x39A0	15:0	0x0000	**otpm_data_208 (R/W)**	N	N
14754 R0x39A2	15:0	0x0000	**otpm_data_209 (R/W)**	N	N
14756 R0x39A4	15:0	0x0000	**otpm_data_210 (R/W)**	N	N
14758 R0x39A6	15:0	0x0000	**otpm_data_211 (R/W)**	N	N
14760 R0x39A8	15:0	0x0000	**otpm_data_212 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14762 R0x39AA	15:0	0x0000	**otpm_data_213 (R/W)**	N	N
14764 R0x39AC	15:0	0x0000	**otpm_data_214 (R/W)**	N	N
14766 R0x39AE	15:0	0x0000	**otpm_data_215 (R/W)**	N	N
14768 R0x39B0	15:0	0x0000	**otpm_data_216 (R/W)**	N	N
14770 R0x39B2	15:0	0x0000	**otpm_data_217 (R/W)**	N	N
14772 R0x39B4	15:0	0x0000	**otpm_data_218 (R/W)**	N	N
14774 R0x39B6	15:0	0x0000	**otpm_data_219 (R/W)**	N	N
14776 R0x39B8	15:0	0x0000	**otpm_data_220 (R/W)**	N	N
14778 R0x39BA	15:0	0x0000	**otpm_data_221 (R/W)**	N	N
14780 R0x39BC	15:0	0x0000	**otpm_data_222 (R/W)**	N	N
14782 R0x39BE	15:0	0x0000	**otpm_data_223 (R/W)**	N	N
14784 R0x39C0	15:0	0x0000	**otpm_data_224 (R/W)**	N	N
14786 R0x39C2	15:0	0x0000	**otpm_data_225 (R/W)**	N	N
14788 R0x39C4	15:0	0x0000	**otpm_data_226 (R/W)**	N	N
14790 R0x39C6	15:0	0x0000	**otpm_data_227 (R/W)**	N	N
14792 R0x39C8	15:0	0x0000	**otpm_data_228 (R/W)**	N	N
14794 R0x39CA	15:0	0x0000	**otpm_data_229 (R/W)**	N	N
14796 R0x39CC	15:0	0x0000	**otpm_data_230 (R/W)**	N	N
14798 R0x39CE	15:0	0x0000	**otpm_data_231 (R/W)**	N	N
14800 R0x39D0	15:0	0x0000	**otpm_data_232 (R/W)**	N	N
14802 R0x39D2	15:0	0x0000	**otpm_data_233 (R/W)**	N	N
14804 R0x39D4	15:0	0x0000	**otpm_data_234 (R/W)**	N	N
14806 R0x39D6	15:0	0x0000	**otpm_data_235 (R/W)**	N	N
14808 R0x39D8	15:0	0x0000	**otpm_data_236 (R/W)**	N	N

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

Register Dec(Hex)	Bits	Default	Name	Frame Sync'd	Bad Frame
14810 R0x39DA	15:0	0x0000	**otpm_data_237 (R/W)**	N	N
14812 R0x39DC	15:0	0x0000	**otpm_data_238 (R/W)**	N	N
14814 R0x39DE	15:0	0x0000	**otpm_data_239 (R/W)**	N	N
14816 R0x39E0	15:0	0x0000	**otpm_data_240 (R/W)**	N	N
14818 R0x39E2	15:0	0x0000	**otpm_data_241 (R/W)**	N	N
14820 R0x39E4	15:0	0x0000	**otpm_data_242 (R/W)**	N	N
14822 R0x39E6	15:0	0x0000	**otpm_data_243 (R/W)**	N	N
14824 R0x39E8	15:0	0x0000	**otpm_data_244 (R/W)**	N	N
14826 R0x39EA	15:0	0x0000	**otpm_data_245 (R/W)**	N	N
14828 R0x39EC	15:0	0x0000	**otpm_data_246 (R/W)**	N	N
14830 R0x39EE	15:0	0x0000	**otpm_data_247 (R/W)**	N	N
14832 R0x39F0	15:0	0x0000	**otpm_data_248 (R/W)**	N	N
14834 R0x39F2	15:0	0x0000	**otpm_data_249 (R/W)**	N	N
14836 R0x39F4	15:0	0x0000	**otpm_data_250 (R/W)**	N	N
14838 R0x39F6	15:0	0x0000	**otpm_data_251 (R/W)**	N	N
14840 R0x39F8	15:0	0x0000	**otpm_data_252 (R/W)**	N	N
14842 R0x39FA	15:0	0x0000	**otpm_data_253 (R/W)**	N	N
14844 R0x39FC	15:0	0x0000	**otpm_data_254 (R/W)**	N	N
14846 R0x39FE	15:0	0x0000	**otpm_data_255 (R/W)**	N	N

Revision History

Rev. B 6/30/15

- Converted to ON Semiconductor template
- Removed Confidential marking

Rev. A 7/26/11

- Initial release