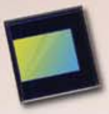
OV8858



datasheet

PRELIMINARY SPECIFICATION

1/4" color CMOS 8 megapixel (3264 x 2448) image sensor
with improved OmniBSI-3™ technology

This document is provided "as is" with no warranties whatsoever, including any warranty of merchantability, non-infringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample.

OmniVision Technologies, Inc. and all its affiliates disclaim all liability, including liability for infringement of any proprietary rights, relating to the use of information in this document. No license, expressed or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

The information contained in this document is considered proprietary to OmniVision Technologies, Inc. and all its affiliates. This information may be distributed to individuals or organizations authorized by OmniVision Technologies, Inc. to receive said information. Individuals and/or organizations are not allowed to re-distribute said information.

Trademark Information

VarioPixel, OmniVision and the OmniVision logo are registered trademarks of OmniVision Technologies, Inc. OmniBSI-3 is a trademark of OmniVision Technologies, Inc.

All other trademarks used herein are the property of their respective owners.

color CMOS 8 megapixel (3264 x 2448) image sensor with OmniBSI-3™ technology

datasheet (COB)
PRELIMINARY SPECIFICATION

version 1.0
july 2013

To learn more about OmniVision Technologies, visit www.ovt.com.
OmniVision Technologies is publicly traded on NASDAQ under the symbol OVTI.

applications

- cellular phones
- PC multimedia
- tablets

ordering information

- **OV08858-G04A-Z** (color, chip probing, 200 µm backgrinding, reconstructed wafer with good die)

features

- 1.12 µm x 1.12 µm pixel with OmniBSI-2™ technology
- optical size of 1/4"
- 32.9° CRA for ~4mm Z-height
- programmable controls for frame rate, mirror and flip, cropping, and windowing
- supports images sizes: 8MP(4:3 - 3264x2448), 8MP (16:9 - 3264x1836), EIS 1080p (2112x1188), 1080p (1920x1080), EIS 720p (1408x792), and more
- 8MP at 30 fps
- two-wire serial bus control (SCCB)
- frame exposure mode for still image (with mechanical shutter)
- 4k bits of embedded one-time programmable (OTP) memory for customer use
- supports Video-in-Video (ViV) mode using an on-chip 1-lane MIPI receiver and a secondary sensor
- special ViV features include: ViV video at up to 30fps, ViV snapshot at up to 15fps, arbitrary positions and shapes for ViV window, separate AWB compensation for secondary sensor, and more
- two on-chip phase lock loops (PLLs)
- image quality control: defect pixel correction, automatic black level calibration, lens shading correction and alternate row HDR
- built-in temperature sensor
- suitable for module size of 8.5mm x 8.5mm x ~4mm

note The OV8858 supports LVDS interface. Contact your local FAE for details.

key specifications (typical)

- **active array size:** 3264 x 2448
- **power supply:**
  analog: 2.6 to 3.0V (2.8V nominal)
  core: 1.14 to 1.26V (1.2V nominal)
  I/O: 1.7 to 3.0V (1.8V or 2.8V nominal)
- **power requirements:**
  active: TBD
  standby: TBD
  XSHUTDN:TBD
- **temperature range:**
  operating: -30°C to +85°C junction temperature (see table 7-2)
  stable image: 0°C to +60°C junction temperature (see table 7-2)
- **output interfaces:** up to 4-lane MIPI serial output
- **output formats:** 10-bit RGB RAW
- **lens chief ray angle:** 32.9° non-linear (see figure 9-2)
- **lens size:** 1/4"
- **input clock frequency:** 6~27 MHz
- **max S/N ratio:** TBD
- **dynamic range:** TBD
- **maximum image transfer rate:**
  3264x2448: 30 fps (see table 2-1)
  3264x1836: 30 fps (see table 2-1)
  2112x1184: 60 fps (see table 2-1)
  1920x1080: 60 fps (see table 2-1)
  1408x792: 90 fps (see table 2-1)
- **sensitivity:** TBD
- **scan mode:** progressive
- **pixel size:** 1.12 µm x 1.12 µm
- **dark current:** TBD
- **image area:** 3678.3 µm x 2767.68 µm
- **die dimensions:** 5040 µm x 4590 µm (COB), 5090 µm x 4640 µm (RW) (see section 8 for details)

note higher junction temperature degrades image quality

note COB refers to whole wafers with known good die and RW refers to singulated good die on a reconstructed wafer. Die size differs between COB and RW.

table of contents

list of figures

list of tables

1 signal descriptions

table 1-1 lists the signal descriptions and their corresponding pad numbers for the OV8858 image sensor. The die information is shown in section 8.

table 1-1 signal descriptions (sheet 1 of 3)

pad number	signal name	pad type	description
1	DOGND	ground	I/O ground
2	DVDD	power	digital circuit power
3	AGND	ground	analog ground
4	AVDD	power	analog power
5	**SID**	input	SCCB last bit ID input 0: SCCB ID address = 0x6C 1: SCCB ID address = 0x20
6	**SIOC**	input	SCCB interface input clock
7	**SIOD**	I/O	SCCB interface data pin
8	NC	–	no connect
9	DOGND	ground	I/O ground
10	**XVCLK**	input	system clock input
11	**VSYNC**	I/O	video output vertical signal
12	**HREF**	I/O	video output horizontal signal
13	DVDD	power	digital circuit power
14	**STROBE**	output	frame exposure output indicator
15	**ILPWM**	output	mechanical shutter output indicator
16	**FSIN**	I/O	frame sync
17	DOVDD	power	I/O power
18	**FREX**	I/O	frame exposure input/mechanical shutter output
19	**GPIO**	I/O	general purpose I/O
20	**XSHUTDN**	input	reset and power down (active low with pull down resistor)
21	DOGND	ground	I/O ground
22	**PWDNB**	input	power down (active low)
23	**TM**	input	test mode (active high iwith pull down resistor)
24	DVDD	power	digital circuit power

table 1-1 signal descriptions (sheet 2 of 3)

pad number	signal name	pad type	description
25	DOGND	ground	I/O ground
26	DOGND	ground	I/O ground
27	ATEST	I/O	analog test pin
28	AVDD	power	analog power
29	AVDD	power	analog power
30	AGND	ground	analog ground
31	AGND	ground	analog ground
32	AVDD	power	analog power
33	AGND	ground	analog ground
34	VH	input	reference
35	VN1	input	reference
36	VN2	input	reference
37	**MDP2**	output	MIPI data positive output
38	**MDN2**	output	MIPI data negative output
39	**MDP0**	output	MIPI data positive output
40	**MDN0**	output	MIPI data negative output
41	DVDD	power	digital circuit power
42	DOGND	ground	I/O ground
43	PVDD	power	PLL analog power
44	DVDD	power	digital circuit power
45	**MCP**	output	MIPI clock positive output
46	**MCN**	output	MIPI clock negative output
47	DOGND	ground	I/O ground
48	**MDP1**	output	MIPI data positive output
49	**MDN1**	output	MIPI data negative output
50	**MDP3**	output	MIPI data positive output
51	**MDN3**	output	MIPI data negative output
52	DVDD	power	digital circuit power
53	RCP	input	MIPI clock positive input
54	RCN	input	MIPI clock negative input

table 1-1 signal descriptions (sheet 3 of 3)

pad number	signal name	pad type	description
55	RDP	input	MIPI data positive input
56	RDN	input	MIPI data negative input
57	DOGND	ground	I/O ground
58	DOVDD	power	I/O power
59	DVDD	power	digital circuit power
60	DVDD	power	digital circuit power
61	DOGND	ground	I/O ground
62	DOGND	ground	I/O ground

table 1-2 configuration under various conditions (sheet 1 of 2)

pad	signal name	RESET[a]	after RESET release[b]	software standby[c]	hardware standby[d]
5	SID	input	input	input	input
6	SIOC	high-z	input	input	high-z
7	SIOD	open drain	I/O	I/O	open drain
10	XVCLK	high-z	input	input	high-z
11	VSYNC	high-z	high-z	high-z by default (configurable)	high-z by default (configurable)
12	HREF	high-z	high-z	high-z by default (configurable)	high-z by default (configurable)
14	STROBE	low	low	low by default (configurable)	low by default (configurable)
15	ILPWM	low	low	low by default (configurable)	low by default (configurable)
16	FSIN	high-z	input	input (configurable)	input (configurable)
18	FREX	high-z	high-z	high-z by default (configurable)	high-z by default (configurable)
19	GPIO	low	low	low by default (configurable)	low by default (configurable)
20	XSHUTDN	input	input	input	input
22	PWDNB	input	input	input	input
23	TM	input	input	input	input

table 1-2 configuration under various conditions (sheet 2 of 2)

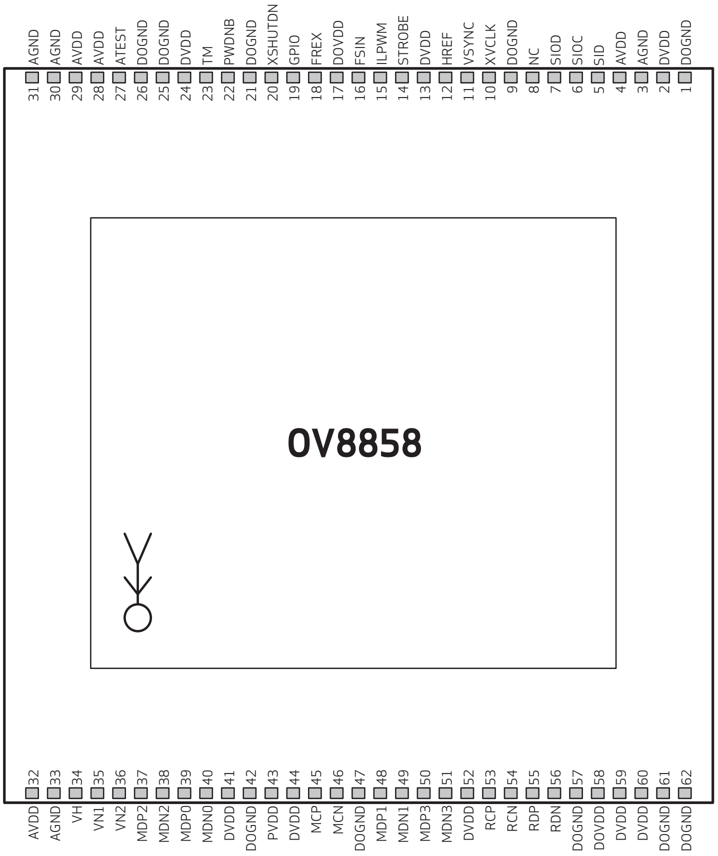
pad	signal name	RESET[a]	after RESET release[b]	software standby[c]	hardware standby[d]
37	MDP2	high-z	high	high by default (configurable)	high by default (configurable)
38	MDN2	high-z	high	high by default (configurable)	high by default (configurable)
39	MDP0	high-z	high	high by default (configurable)	high by default (configurable)
40	MDN0	high-z	high	high by default (configurable)	high by default (configurable)
45	MCP	high-z	high	high by default (configurable)	high by default (configurable)
46	MCN	high-z	high	high by default (configurable)	high by default (configurable)
48	MDP1	high-z	high	high by default (configurable)	high by default (configurable)
49	MDN1	high-z	high	high by default (configurable)	high by default (configurable)
50	MDP3	high-z	high	high by default (configurable)	high by default (configurable)
51	MDN3	high-z	high	high by default (configurable)	high by default (configurable)

a. XSHUTDN = 0

b. XSHUTDN from 0 to 1

c. sensor set to sleep from streaming mode

d. sensor set to hardware standby from streaming mode by pulling PWDNB = 0 figure 1-1 pad diagram



8858_COB_DS_1_1

table 1-3 pad symbol and equivalent circuit (sheet 1 of 2)

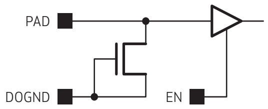
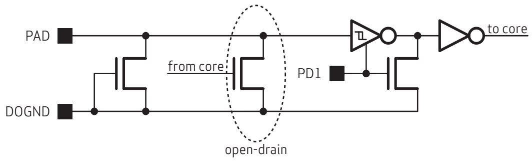
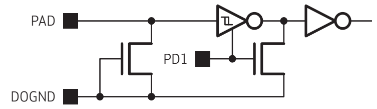
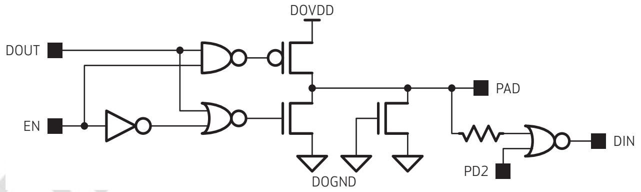
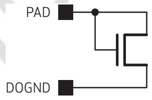
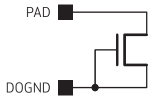
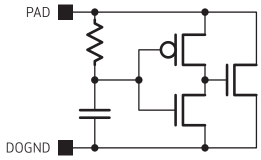
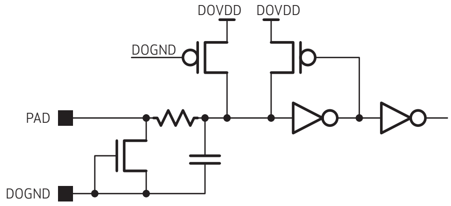
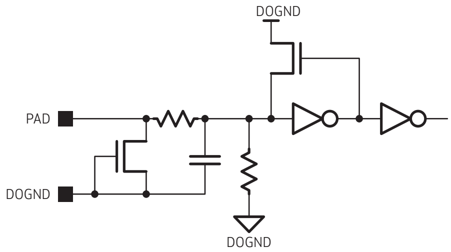
symbol	equivalent circuit
XVCLK	
SIOD	

table 1-3 pad symbol and equivalent circuit (sheet 2 of 2)

symbol	equivalent circuit
SIOC	
VSYNC, STROBE, ILPWM, FREX, FSIN, GPIO, HREF	
VN1, VN2	
MDP3, MDP2, MDP1, MDP0, MDN3, VH, MDN2, MDN1, MDN0, MCP, MCN, EGND, AGND, DOGND	
AVDD, EVDD, DVDD, DOVDD, PVDD	
PWDNB	
XSHUTDN, SID, TM	

2 system level description

2.1 overview

The OV8858 RAW RGB image sensor is a high performance, 1/4-inch 8 megapixel CMOS image sensor that delivers 3264x2448 at 30 fps using OmniBSI-3™ pixel technology. It provides full-frame, sub-sampled, and windowed 10-bit MIPI images in various formats via the control of the Serial Camera Control Bus (SCCB) interface.

The OV8858 has an 8 megapixel image array capable of operating at up to 30 frames per second (fps) in 10-bit resolution with complete user control over image quality, formatting and output data transfer. Some image processing functions, such as defective pixel canceling, lens correction (LENC), etc., are programmable through the SCCB interface.

In addition, OmniVision image sensors use proprietary sensor technology to improve image quality by reducing or eliminating common lighting/electrical sources of image contamination, such as fixed pattern noise, smearing, etc., to produce a clean, fully stable, color image.

For customized information purposes, the OV8858 includes 4k bits of one-time programmable (OTP) memory. The OV8858 has a MIPI interface of up to four lanes.

2.2 architecture

The OV8858 sensor core generates streaming pixel data at a constant frame rate. **figure 2-1** shows the functional block diagram of the OV8858 image sensor.

The timing generator outputs clocks to access the rows of the imaging array, pre-charging and sampling the rows of the array sequentially. In the time between pre-charging and sampling a row, the charge in the pixels decreases with exposure to incident light. This is the exposure time in rolling shutter architecture.

The exposure time is controlled by adjusting the time interval between pre-charging and sampling. After the data of the pixels in the row has been sampled, it is processed through analog circuitry to correct the offset and multiply the data with corresponding gain. Following analog processing is the ADC which outputs up to 10-bit data for each pixel in the array.

figure 2-1 OV8858 block diagram

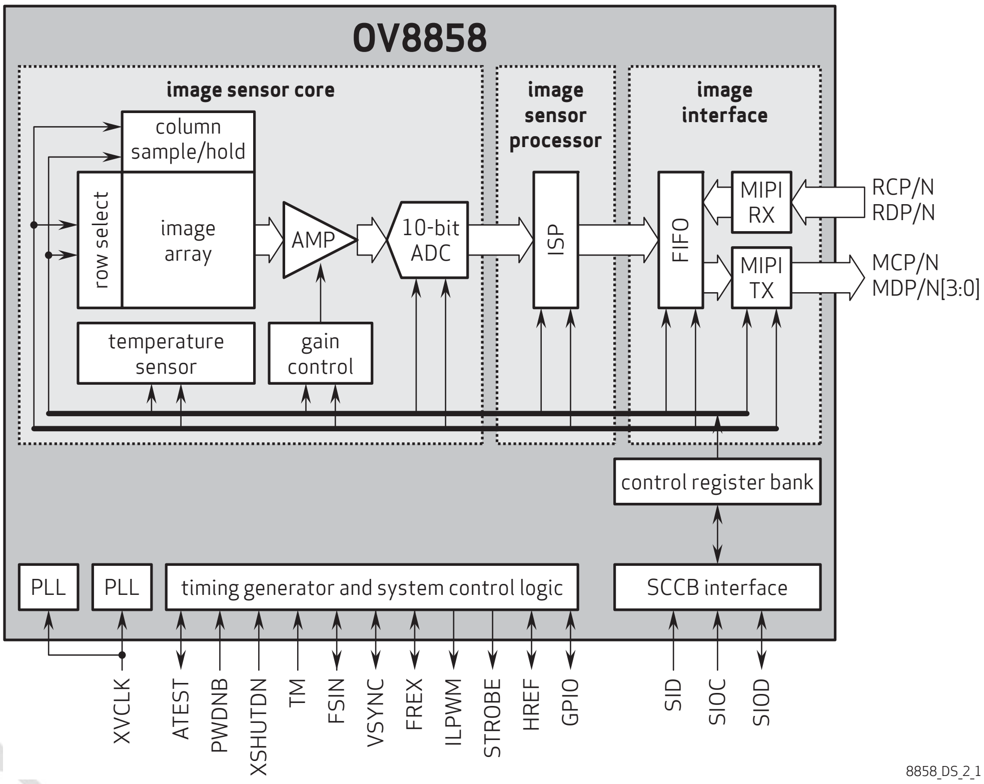
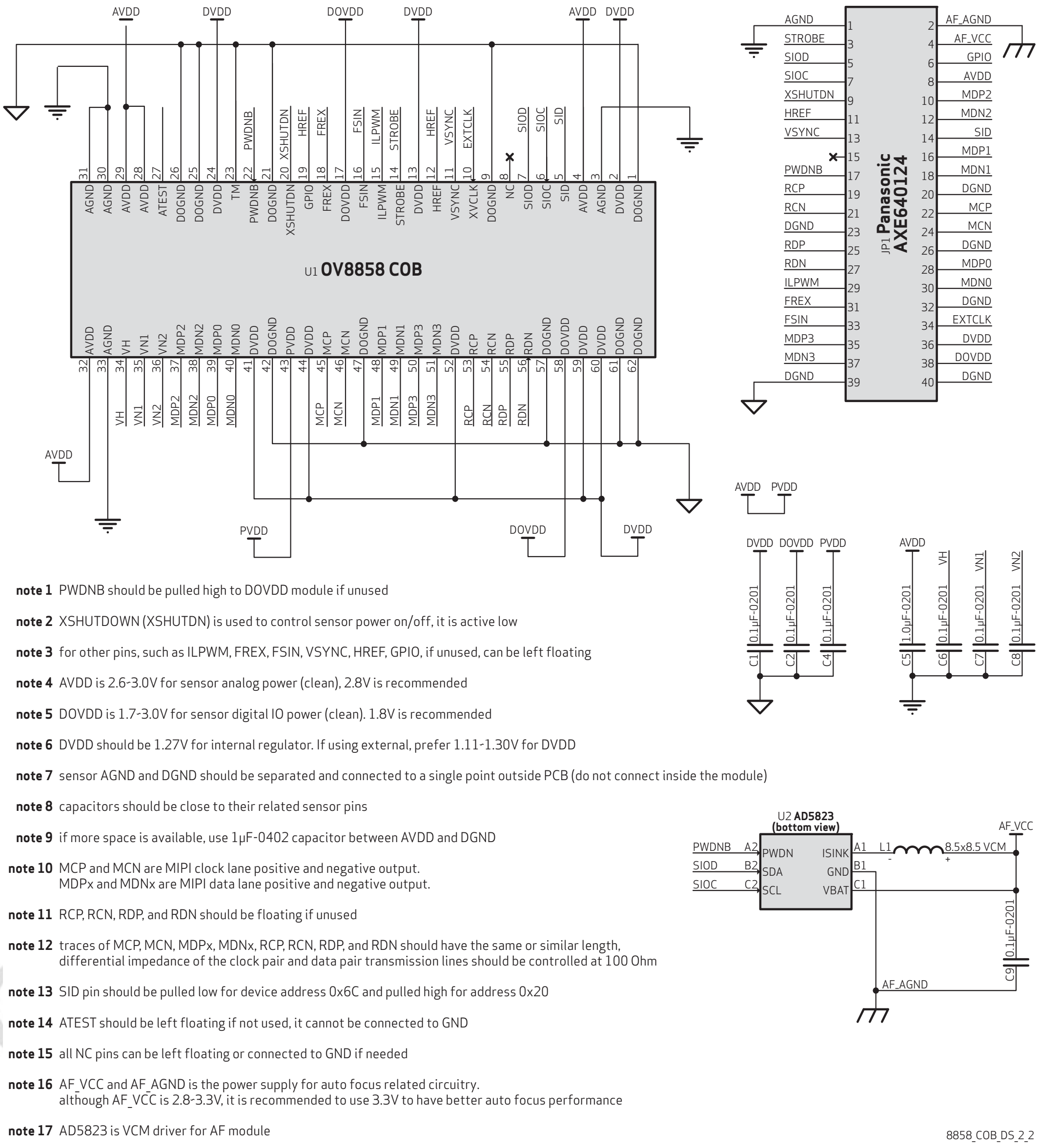


figure 2-2 OV8858 reference schematic



note 1 PWDNB should be pulled high to DOVDD module if unused

note 2 XSHUTDOWN (XSHUTDN) is used to control sensor power on/off, it is active low

note 3 for other pins, such as ILPWM, FREX, FSIN, VSYNC, HREF, GPIO, if unused, can be left floating

note 4 AVDD is 2.6-3.0V for sensor analog power (clean), 2.8V is recommended

note 5 DOVDD is 1.7-3.0V for sensor digital IO power (clean). 1.8V is recommended

note 6 DVDD should be 1.27V for internal regulator. If using external, prefer 1.11-1.30V for DVDD

note 7 sensor AGND and DGND should be separated and connected to a single point outside PCB (do not connect inside the module)

note 8 capacitors should be close to their related sensor pins

note 9 if more space is available, use 1µF-0402 capacitor between AVDD and DGND

note 10 MCP and MCN are MIPI clock lane positive and negative output.
MDPx and MDNx are MIPI data lane positive and negative output.

note 11 RCP, RCN, RDP, and RDN should be floating if unused

note 12 traces of MCP, MCN, MDPx, MDNx, RCP, RCN, RDP, and RDN should have the same or similar length,
differential impedance of the clock pair and data pair transmission lines should be controlled at 100 Ohm

note 13 SID pin should be pulled low for device address 0x6C and pulled high for address 0x20

note 14 ATEST should be left floating if not used, it cannot be connected to GND

note 15 all NC pins can be left floating or connected to GND if needed

note 16 AF_VCC and AF_AGND is the power supply for auto focus related circuitry.
although AF_VCC is 2.8-3.3V, it is recommended to use 3.3V to have better auto focus performance

note 17 AD5823 is VCM driver for AF module

2.3 format and frame

The OV8858 supports RAW RGB output with one/two/four lane MIPI interface.

table 2-1 format and frame rate

format	resolution	output	10-bit output MIPI 4 lanes	methodology
8 MP	3264 x 2448	30 fps	720 Mbps/lane	full resolution (4:3)
6 MP HD	3264 x1836	30 fps	720 Mbps/lane	full resolution (16:9)
EIS 1080p	2112 x 1188	60 fps	720 Mbps/lane	cropping
1080p	1920 x 1080	60 fps	720 Mbps/lane	cropping
EIS 720p	1408 x 792	90 fps	720 Mbps/lane	cropping + 2x binning
720p	1280 x 720	90 fps	720 Mbps/lane	cropping + 2x binning
VGA	640 x 480	120 fps	720 Mbps/lane	cropping + 4x binning

2.4 I/O control

The OV8858 can configure its I/O pads as an input or output. For the output signal, it follows one of two paths: either from the data path or from register control.

table 2-2 I/O control registers (sheet 1 of 2)

function	register	description
output drive capability control	0x3011	Bit[6:5]: I/O pad drive capability 00: 1x 01: 2x 10: 3x 11: 4x
HREF I/O control	0x3002	Bit[6]: HREF output enable 0: input 1: output
HREF output select	0x3010	Bit[6]: enable HREF as GPIO controlled by register
HREF output value	0x300D	Bit[6]: register control HREF output
GPIO I/O control	0x3002	Bit[0]: GPIO output enable 0: input 1: output
GPIO output value	0x300D	Bit[0]: register control GPIO output

table 2-2 I/O control registers (sheet 2 of 2)

function	register	description
VSYNC I/O control	0x3002	Bit[7]: VSYNC output enable 0: input 1: output
VSYNC output select	0x3010	Bit[7]: enable VSYNC as GPIO controlled by register
VSYNC output value	0x300D	Bit[7]: register control VSYNC output

2.5 MIPI interface

The OV8858 supports a MIPI interface of up to 4-lanes. The MIPI interface can be configured for 1/2/4-lane and each lane is capable of a data transfer rate of up to 800 Mbps.

2.6 power management

Based on the system power configuration (XSHUTDN, PWDNB control), the power up sequence will be different. OmniVision recommends cutting off all power supplies, including the external DVDD, when the sensor is not in use.

2.6.1 power up sequence

To avoid any glitch from a strong external noise source, OmniVision recommends controlling XSHUTDN or PWDNB by GPIO and tying the other pin to DOVDD.

Whether or not XSHUTDN is controlled by GPIO, the XSHUTDN rising cannot occur before AVDD and DOVDD.

table 2-3 power up sequence

case	XSHUTDN	PWDNB	power up sequence requirement
1	GPIO	DOVDD	Refer to **figure 2-3** 1. DOVDD rising must occur before DVDD rising 2. AVDD rising can occur before or after DOVDD rising 3. AVDD must occur before DVDD 4. XSHUTDN rising must occur after AVDD, DOVDD and DVDD are stable
2	DOVDD	GPIO	Refer to **figure 2-4** 1. AVDD rising occurs before DOVDD rising 2. DOVDD rising occurs before DVDD 3. PWDNB rising occurs after DVDD rising

table 2-4 power up sequence timing constraints

constraint	label	min	max	unit
AVDD rising – DOVDD rising	t0	0	∞	ns
DOVDD rising – AVDD rising	t1			ns
XSHUTDN rising – first SCCB transaction	t2	8192		XVCLK cycles
minimum number of XVCLK cycles prior to the first SCCB transaction	t3	8192		XVCLK cycles
PLL start up/lock time	t4		0.2	ms
entering streaming mode – first frame start sequence (fixed part)	t5		10	ms
entering streaming mode – first frame start sequence (variable part)	t6	delay is the exposure time value		lines
AVDD or DOVDD, whichever is last – DVDD	t7	0	∞	ns
DVDD – PWDNB rising	t8	0	∞	ns
DVDD – XSHUTDN rising	t9	0	∞	ns

figure 2-3 power up sequence (case 1)

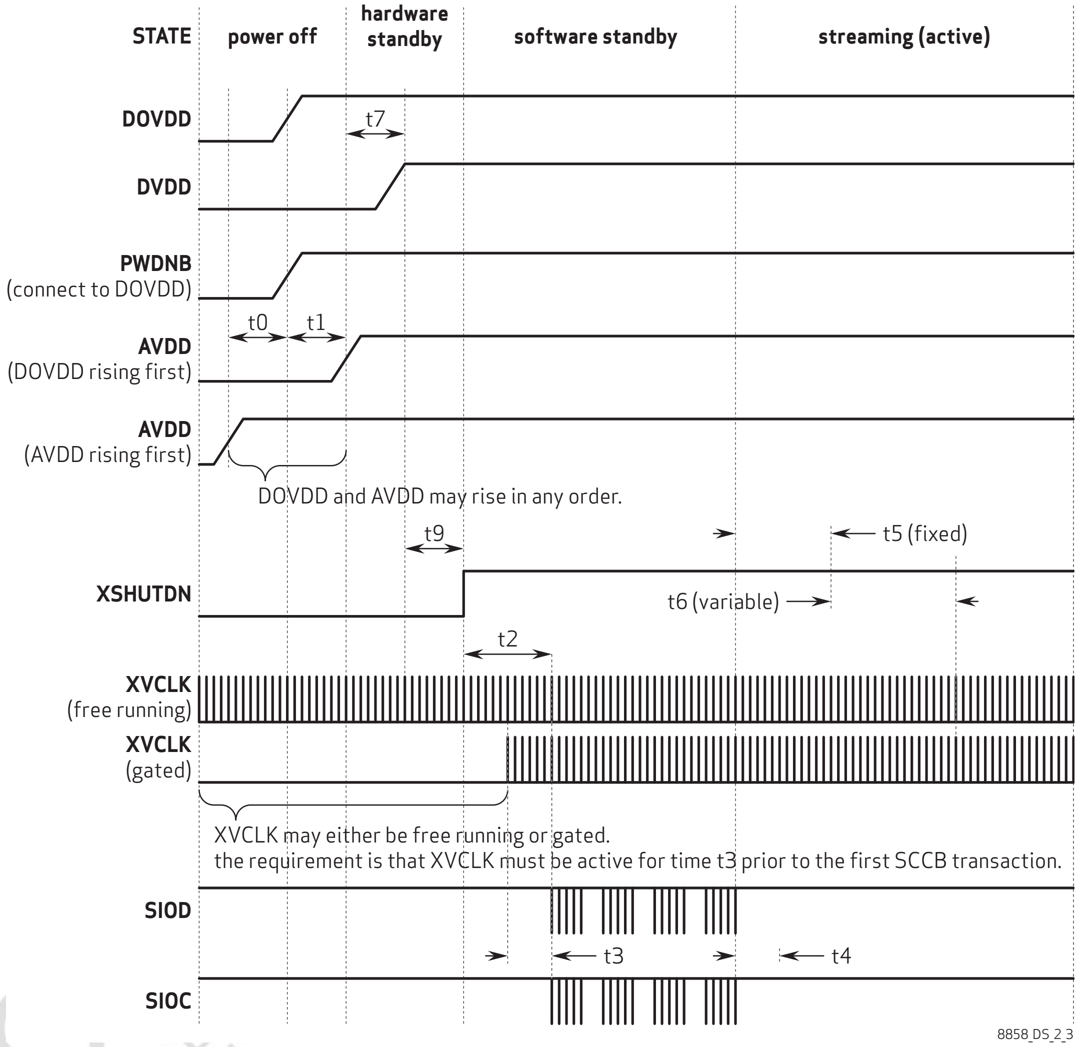
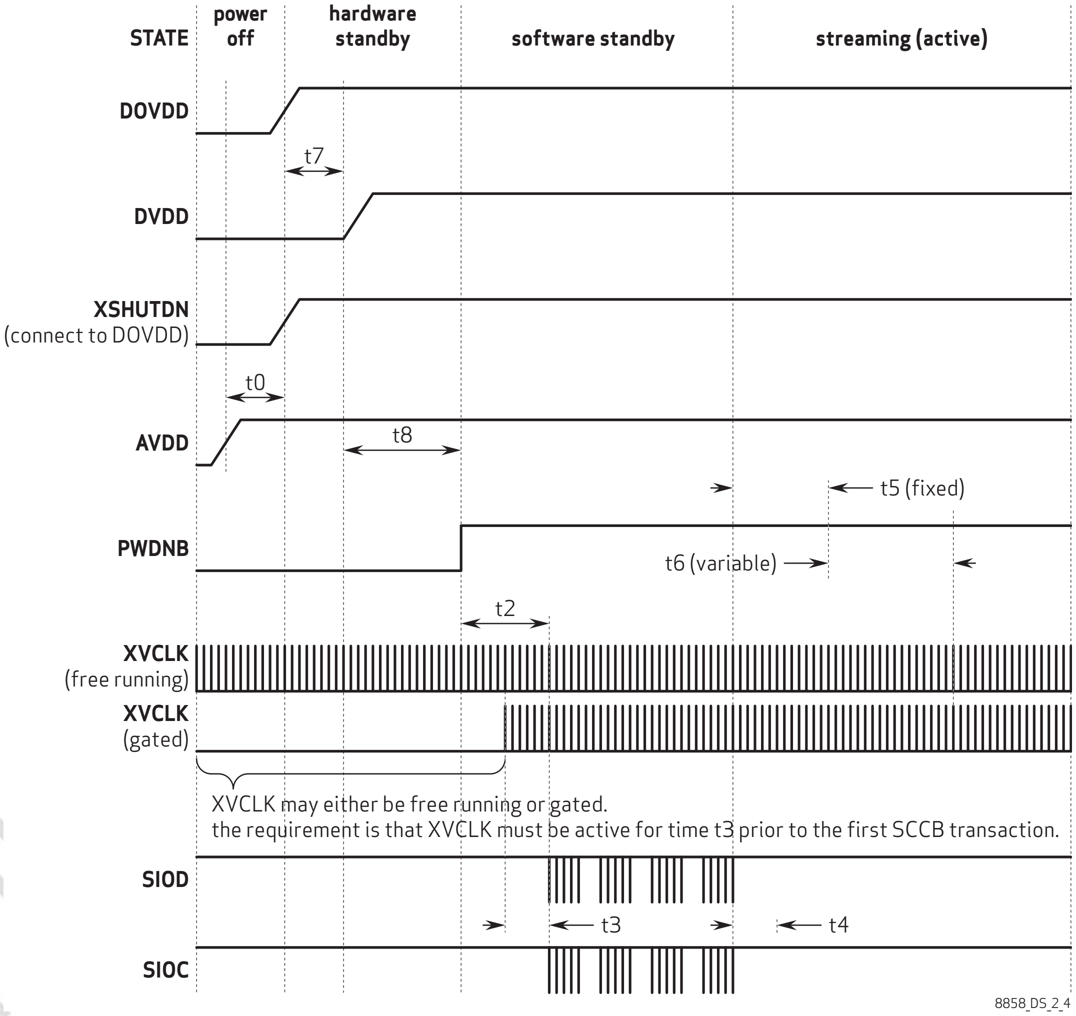


figure 2-4 power up sequence (case 2)



2.6.2 power down sequence

The digital and analog supply voltages can be powered down in any order (e.g. DOVDD, then AVDD or AVDD, then DOVDD). Similar to the power up sequence, the XVCLK input clock may be either gated or continuous. To avoid bad frames from the MIPI, OmniVision recommends to using group hold to send SCCB sleep command.

table 2-5 power down sequence

case	XSHUTDN	PWDNB	power down sequence requirement
1	GPIO	DOVDD	Refer to **figure 2-6** 1. software standby recommended 2. pull XSHUTDN low for minimum power consumption 3. cut off DVDD 4. pull AVDD and DOVDD low in any order
2	DOVDD	GPIO	Refer to **figure 2-7** 1. software standby recommended 2. pull PWDNB low for minimum power consumption 3. cut off DVDD 4. pull DOVDD low (XSHUTDN connected to DOVDD) 5. pull AVDD low

table 2-6 power down sequence timing constraints

constraint	label	min	max	unit
enter software standby SCCB command device in software standby mode	t0	when a frame of MIPI data is output, wait for the MIPI end code before entering the software for standby; otherwise, enter the software standby mode immediately		
minimum of XVCLK cycles after the last SCCB transaction or MIPI frame end	t1	512		XVCLK cycles
last SCCB transaction or MIPI frame end, XSHUTDN falling	t2	512		XVCLK cycles
XSHUTDN falling – AVDD falling or DOVDD falling whichever is first	t3	0.0		ns
AVDD falling – DOVDD falling	t4	AVDD and DOVDD may fall in any order, the falling separation can vary from 0 ns to infinity		ns
DOVDD falling – AVDD falling	t5			ns
PWDNB falling – DOVDD falling	t6	0.0		ns
XSHUTDN falling – DVDD falling	t7	0.0		ns
DVDD falling – AVDD falling or DOVDD falling whichever is first	t8	0.0		ns
PWDNB falling – DVDD falling	t9	0.0		ns

figure 2-5 software standby sequence

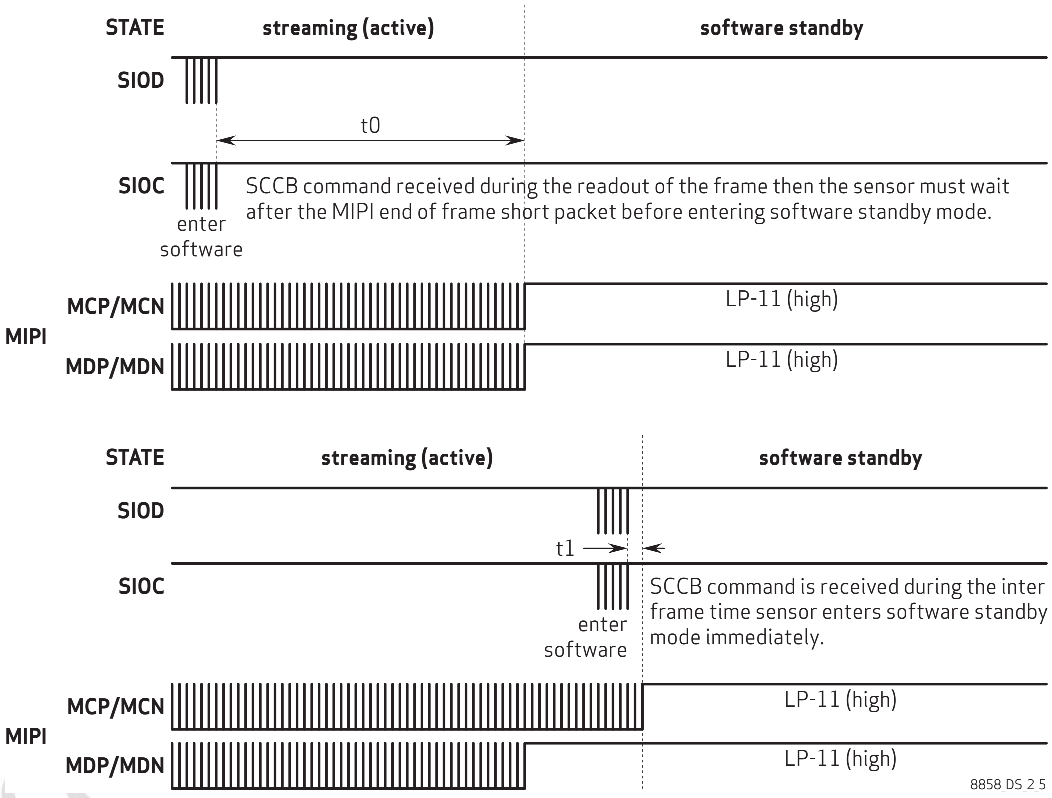

STATE
streaming (active)
software standby
SIOD
t0
SIOC
SCCB command received during the readout of the frame then the sensor must wait after the MIPI end of frame short packet before entering software standby mode.
enter software
MIPI
MCP/MCN
MDP/MDN
LP-11 (high)
LP-11 (high)
STATE
streaming (active)
software standby
SIOD
t1
SIOC
SCCB command is received during the inter frame time sensor enters software standby mode immediately.
enter software
MIPI
MCP/MCN
MDP/MDN
LP-11 (high)
LP-11 (high)
8858_DS_2_5

figure 2-6 power down sequence (case 1)

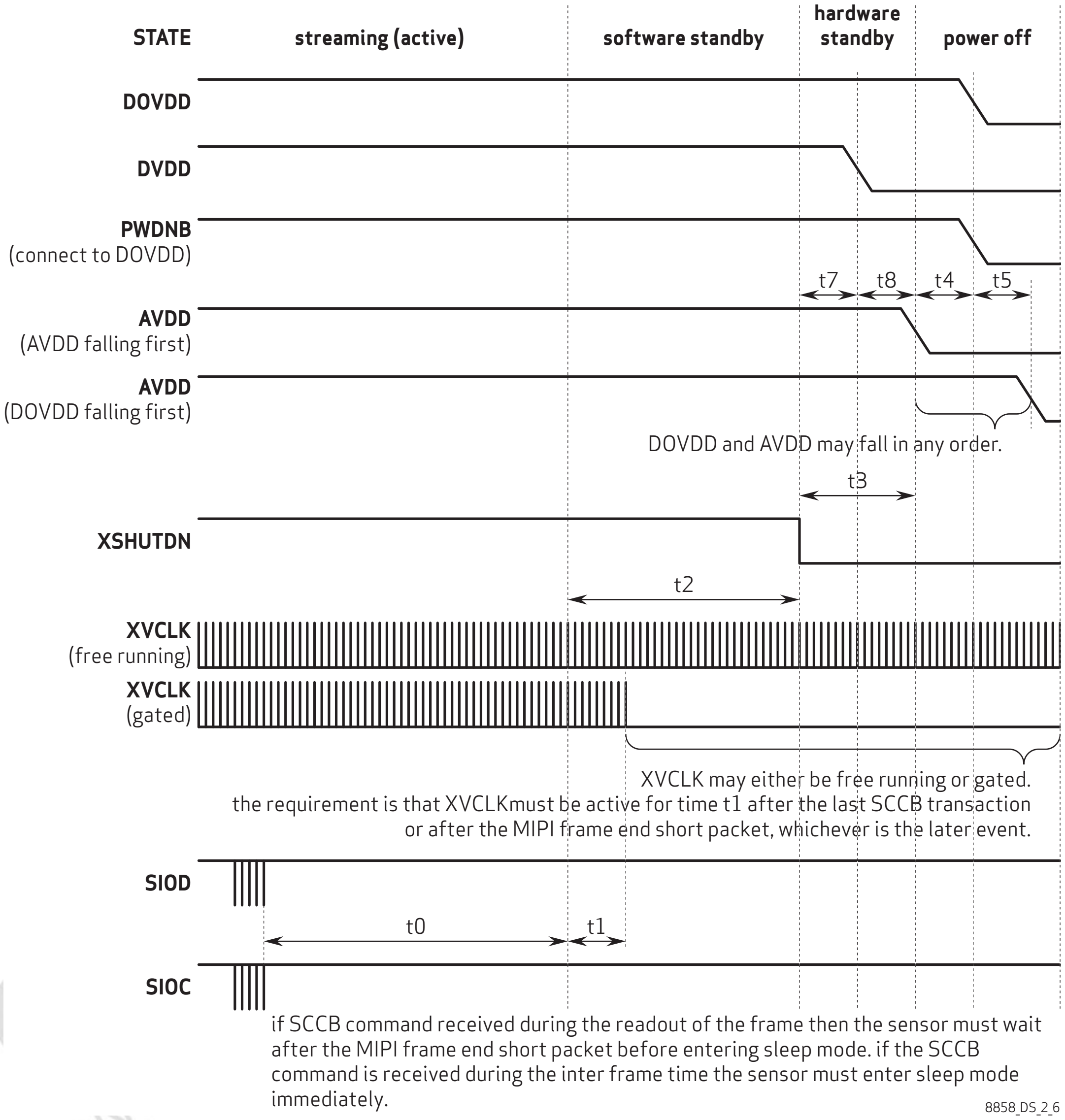
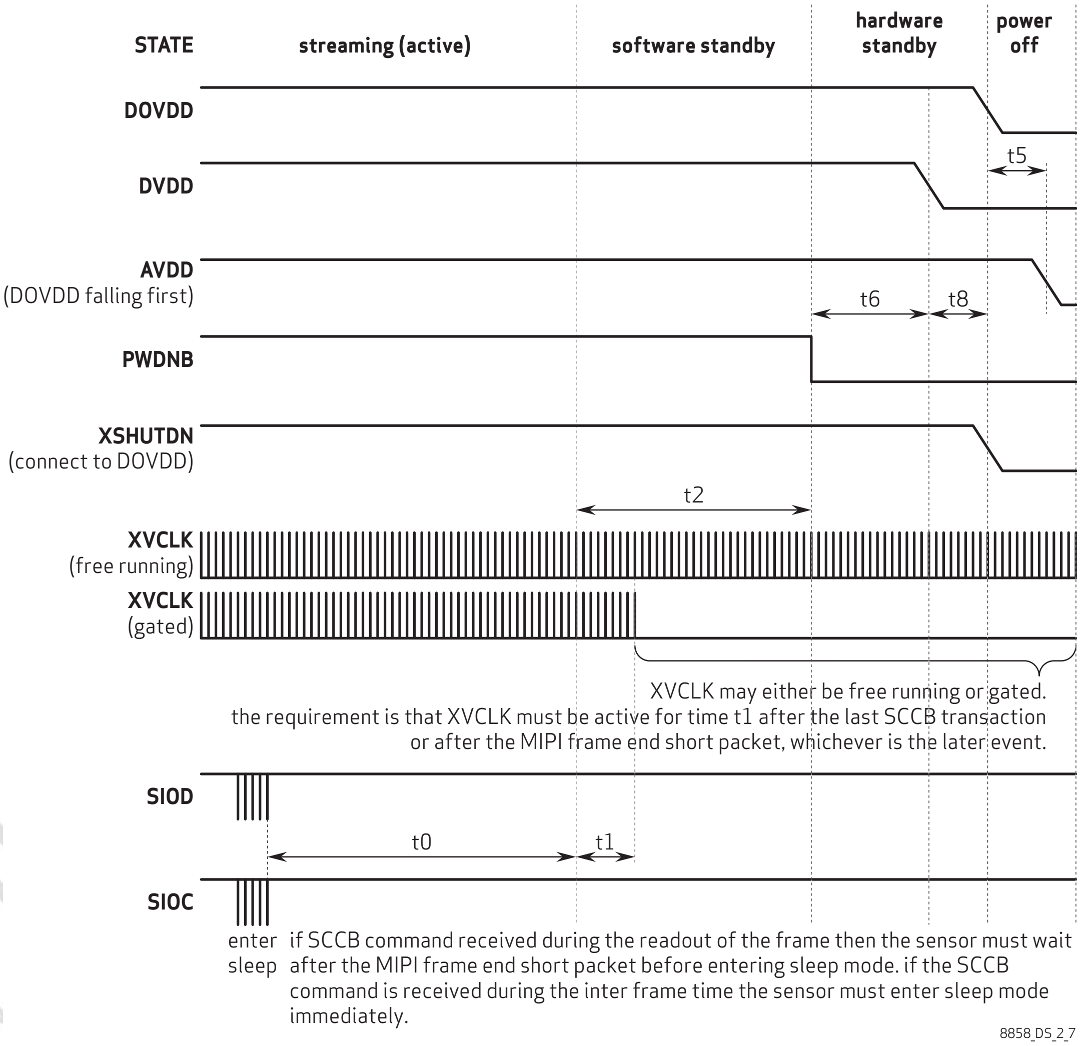

STATE
streaming (active)
software standby
hardware standby
power off
DOVDD
DVDD
PWDNB
(connect to DOVDD)
t7
t8
t4
t5
AVDD
(AVDD falling first)
AVDD
(DOVDD falling first)
DOVDD and AVDD may fall in any order.
t3
XSHUTDN
t2
XVCLK
(free running)
XVCLK
(gated)
XVCLK may either be free running or gated.
the requirement is that XVCLKmust be active for time t1 after the last SCCB transaction
or after the MIPI frame end short packet, whichever is the later event.
SIOD
t0
t1
SIOC
if SCCB command received during the readout of the frame then the sensor must wait
after the MIPI frame end short packet before entering sleep mode. if the SCCB
command is received during the inter frame time the sensor must enter sleep mode
immediately.
8858_DS_2_6

figure 2-7 power down sequence (case 2)


STATE
streaming (active)
software standby
hardware standby
power off
DOVDD
DVDD
t5
AVDD
(DOVDD falling first)
t6
t8
PWDNB
XSHUTDN
(connect to DOVDD)
t2
XVCLK
(free running)
XVCLK
(gated)
XVCLK may either be free running or gated.
the requirement is that XVCLK must be active for time t1 after the last SCCB transaction
or after the MIPI frame end short packet, whichever is the later event.
SIOD
t0
t1
SIOC
enter sleep
if SCCB command received during the readout of the frame then the sensor must wait after the MIPI frame end short packet before entering sleep mode. if the SCCB command is received during the inter frame time the sensor must enter sleep mode immediately.
8858_DS_2_7

2.7 reset

The whole chip will be reset during power up. Manually applying a hardware reset (XSHUTDN=0) upon power up is recommended even though the on-chip power up reset is included. The hardware reset is active low with an asynchronized design. The reset pulse width should be greater than or equal to 2 ms.

2.7.1 power ON reset

The power on reset can be controlled from an external pin. Additionally, in this sensor, a power on reset is generated after the core power becomes stable.

2.7.2 software reset

When register 0x0103[0] is configured as 1, all registers are reset to default value.

2.8 hardware and software standby

Two suspend modes are available for the OV8858:

- hardware standby
- software standby

2.8.1 hardware standby

To initiate hardware standby mode, the XSHUTDN or PWDNB pin must be tied to low. When this occurs, the OV8858 internal device clock is halted even when the external clock source is still clocking and all internal counters are reset.

2.8.2 software standby

Executing a software power down (0x0100[0]) through the SCCB interface suspends internal circuit activity, but does not halt the device clock. All register content is maintained in standby mode. During the resume state, all the registers are restored to their original values.

table 2-7 hardware and standby description (sheet 1 of 2)

mode	description
hardware standby with PWDNB	1. enabled by pulling PWDNB low 2. input clock is gated by PWDNB, no SCCB communication 3. register values are maintained 4. power down all blocks and regulator 5. low power consumption 6. GPIO can be configured as high/low/tri-state
hardware standby with XSHUTDN	1. enabled by pulling XSHUTDN low 2. power down all blocks 3. register values are reset to default values 4. no SCCB communication 5. minimum power consumption

table 2-7 hardware and standby description (sheet 2 of 2)

mode	description
software standby	1. default mode after power on reset 2. power down all blocks except SCCB 3. register values are maintained 4. SCCB communication is available 5. low power consumption 6. GPIO can be configured as high/low/tri-state

2.9 system clock control

2.9.1 PLL1

The PLL1 generates a default 90 MHz pixel clock and 720 MHz MIPI serial clock from a 6~27 MHz input clock. The VCO range is from 500 MHz to 960 MHz. A programmable clock provided is needed to generate different frequencies.

2.9.2 PLL2

The PLL2 generates a default 144 MHz system clock from a 6~27 MHz input clock. The VCO range is from 500 MHz to 960 MHz. A programmable clock divider is provided to generate different frequencies.

figure 2-8 clock scheme

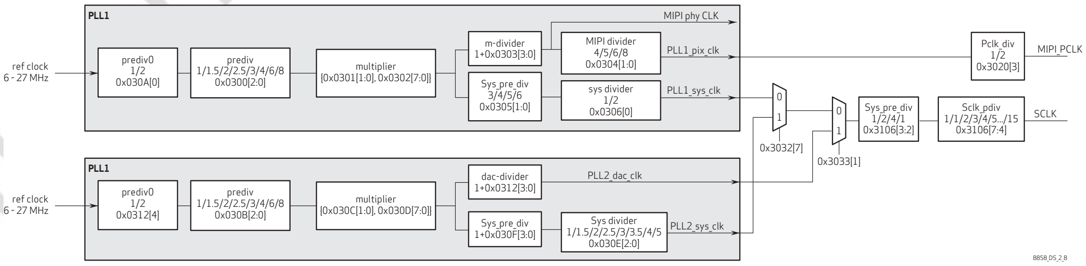


table 2-8 PLL registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x0300	PLL_CTRL_0	0x00	RW	Bit[2:0]: pll1_pre_div
0x0301	PLL_CTRL_1	0x00	RW	Bit[1:0]: pll1_multiplier[9:8]
0x0302	PLL_CTRL_2	0x19	RW	Bit[7:0]: pll1_multiplier[7:0]
0x0303	PLL_CTRL_3	0x00	RW	Bit[3:0]: pll1_divm
0x0304	PLL_CTRL_4	0x03	RW	Bit[1:0]: pll1_div_mipi
0x0305	PLL_CTRL_5	0x01	RW	Bit[1:0]: pll1_div_sp

table 2-8 PLL registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x0306	PLL_CTRL_6	0x01	RW	Bit[0]: pll1_div_s
0x0308	PLL_CTRL_8	0x00	RW	Bit[0]: pll1_bypass
0x0309	PLL_CTRL_9	0x01	RW	Bit[2:0]: pll1_cp
0x030A	PLL_CTRL_A	0x00	RW	Bit[0]: pll1_predivp
0x030B	PLL_CTRL_B	0x00	RW	Bit[2:0]: pll2_pre_div
0x030C	PLL_CTRL_C	0x00	RW	Bit[1:0]: pll2_r_divp[9:8]
0x030D	PLL_CTRL_D	0x1E	RW	Bit[7:0]: pll2_r_divp[7:0]
0x030E	PLL_CTRL_E	0x02	RW	Bit[2:0]: pll2_r_divs
0x030F	PLL_CTRL_F	0x02	RW	Bit[3:0]: pll2_r_divsp
0x0310	PLL_CTRL_10	0x01	RW	Bit[2:0]: pll2_r_cp
0x0311	PLL_CTRL_11	0x00	RW	Bit[0]: pll2_bypass
0x0312	PLL_CTRL_12	0x01	RW	Bit[4]: pll2_pre_div0 Bit[3:0]: pll2_r_divdac

table 2-9 sample PLL configuration (sheet 1 of 2)

control name	address	input clock (XVCLK)	
		24 MHz	6 MHz
PLL1_PREDIVP	0x030A[0]	0x0	0x0
PLL1_PREDIV	0x0300[2:0]	0x0	0x0
PLL1_MULTIPLIER	{0x0301[1:0], 0x0302[7:0]}	0x1E	0x64
PLL1_DIV_MIPI	0x0304[1:0]	0x3	0x3
PLL1_DIVM	0x0303[3:0]	0x0	0x0
PLL1_DIVSP	0x0305[1:0]	0x1	0x1
PLL1_DIVS	0x0306[0]	0x1	0x1
PLL2_PREDIVP	0x3012[0]	0x0	0x0
PLL2_PREDIV	0x030B[2:0]	0x0	0x0
PLL2_MULTIPLIER	{0x030C[1:0], 0x030D[7:0]}	0x1E	0x78
PLL2_DIVSP	0x030F[3:0]	0x4	0x2
PLL2_DIVS	0x030E[2:0]	0x0	0x2

table 2-9 sample PLL configuration (sheet 2 of 2)

control name	address	input clock (XVCLK)	
		24 MHz	6 MHz
SCLK	–	144MHz	120MHz
PHY_SCLK	–	720MHz	600MHz
MIPI_PCLK	–	90MHz	75MHz

2.10 serial camera control bus (SCCB) interface

The Serial Camera Control Bus (SCCB) interface controls the image sensor operation. Refer to the *OmniVision Technologies Serial Camera Control Bus (SCCB) Specification* for detailed usage of the serial control port.

In the OV8858, the SCCB ID is controlled by the SID pin. If SID is low, the sensor's SCCB ID is 0x6C. If SID is high, the sensor's SCCB ID is 0x20. The SCCB ID can also be programmed by registers. When 0x303F[0] is 1, the ID comes from register 0x3004 when SID=0 and register 0x3012 when SID=1.

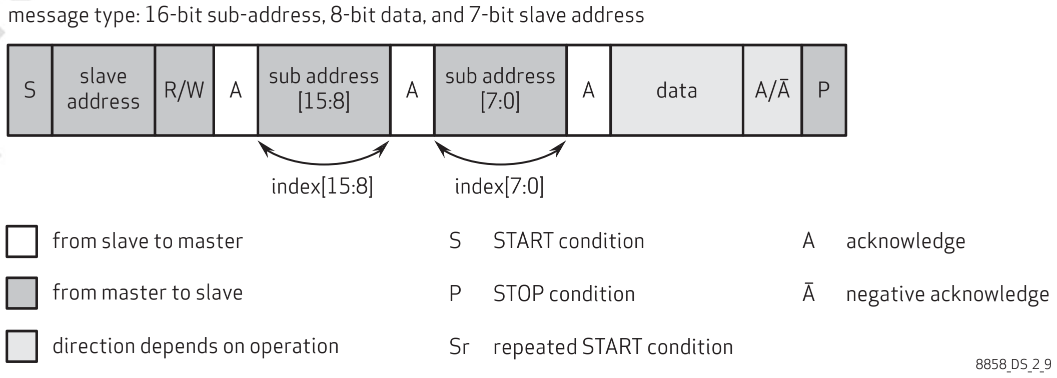
2.10.1 data transfer protocol

The data transfer of the OV8858 follows the SCCB protocol.

2.10.2 message format

The OV8858 supports the message format shown in **figure 2-9**. The repeated START (Sr) condition is not shown in **figure 2-10**, but is shown in **figure 2-11** and **figure 2-12**.

figure 2-9 message type



2.10.3 read / write operation

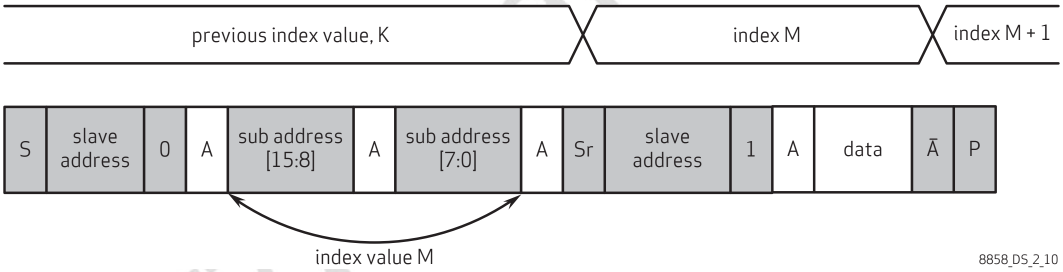
The OV8858 supports four different read operations and two different write operations:

- a single read from random locations
- a sequential read from random locations
- a single read from current location
- a sequential read from current location
- single write to random locations
- sequential write starting from random location

The sub-address in the sensor automatically increases by one after each read/write operation.

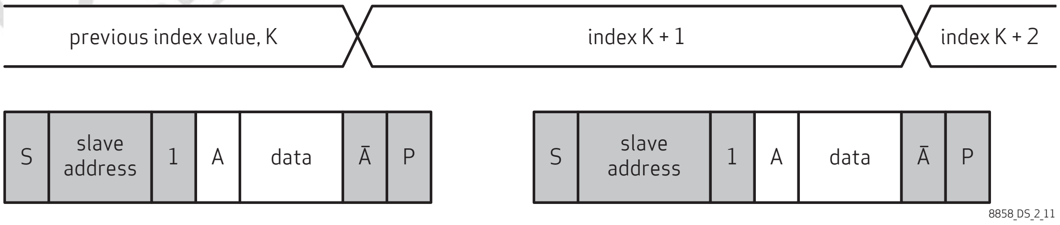
In a single read from random locations, the master does a dummy write operation to desired sub-address, issues a repeated start condition and then addresses the camera again with a read operation. After acknowledging its slave address, the camera starts to output data onto the SIOD line as shown in **figure 2-10**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 2-10 SCCB single read from random location



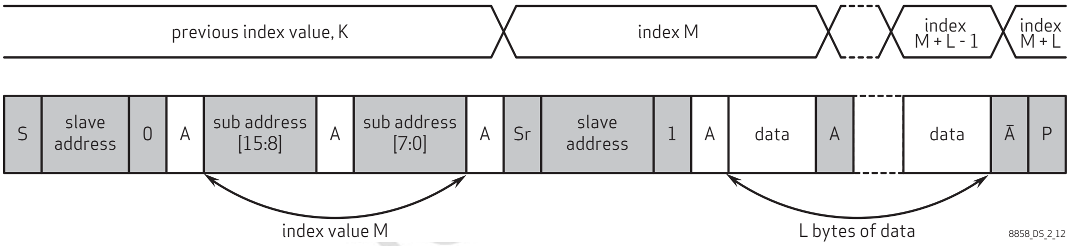
If the host addresses the camera with read operation directly without the dummy write operation, the camera responds by setting the data from last used sub-address to the SIOD line as shown in **figure 2-11**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 2-11 SCCB single read from current location



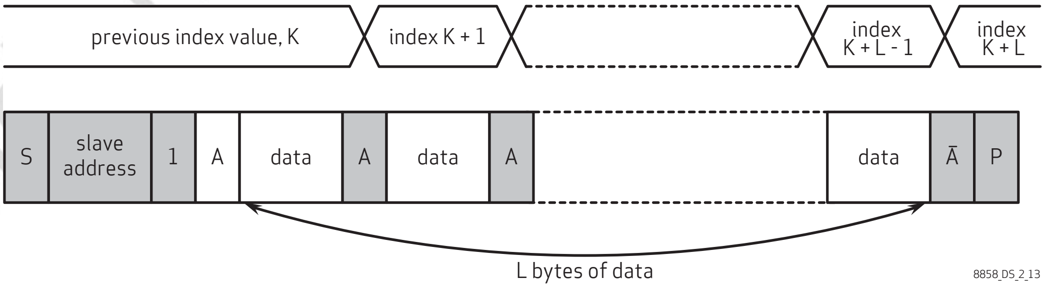
The sequential read from a random location is illustrated in **figure 2-12**. The master does a dummy write to the desired sub-address, issues a repeated start condition after acknowledge from slave and addresses the slave again with read operation. If a master issues an acknowledge after receiving data, it acts as a signal to the slave that the read operation shall continue from the next sub-address. When master has read the last data byte, it issues a negative acknowledge and stop condition.

figure 2-12 SCCB sequential read from random location



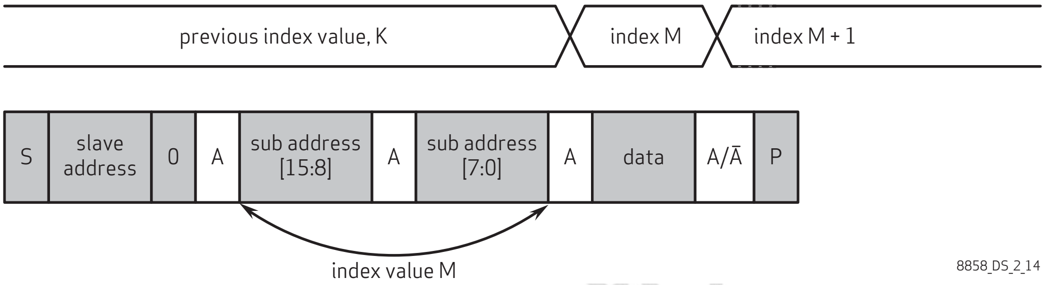
The sequential read from current location is similar to a sequential read from a random location. The only exception is that there is no dummy write operation, as shown in **figure 2-13**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 2-13 SCCB sequential read from current location



The write operation to a random location is illustrated in **figure 2-14**. The master issues a write operation to the slave, sets the sub-address and data correspondingly after the slave has acknowledged. The write operation is terminated with a stop condition from the master.

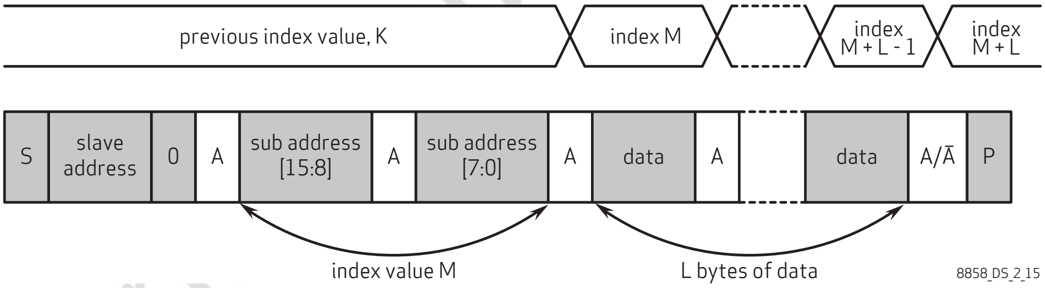
figure 2-14 SCCB single write to random location



8858_DS_2_14

The sequential write is illustrated in **figure 2-15**. The slave automatically increments the sub-address after each data byte. The sequential write operation is terminated with stop condition from the master.

figure 2-15 SCCB sequential write to random location



8858_DS_2_15

2.10.4 SCCB timing

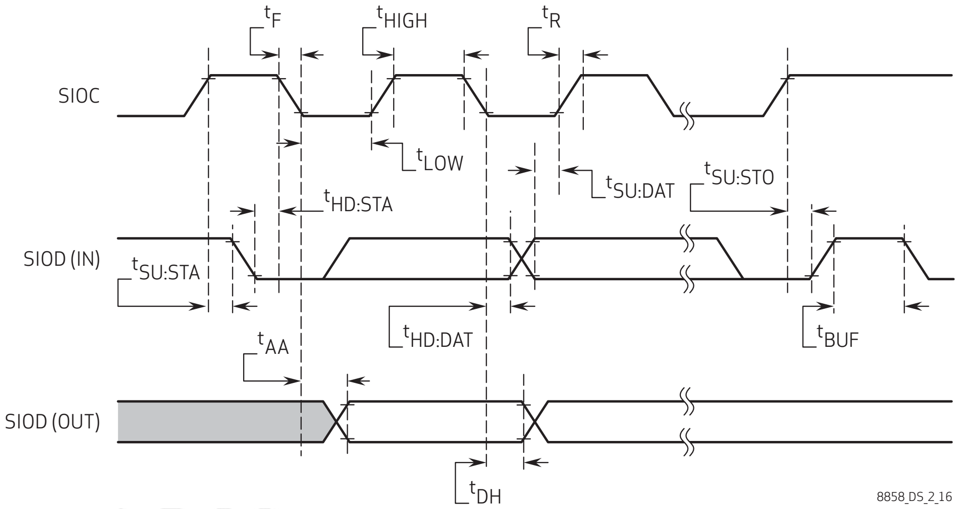


figure 2-16 SCCB interface timing

table 2-10 SCCB interface timing specifications[a][b]

symbol	parameter	min	typ	max	unit
f_{SIOC}	clock frequency			400	kHz
t_{LOW}	clock low period	1.3			μs
t_{HIGH}	clock high period	0.6			μs
t_{AA}	SIOC low to data out valid	0.1		0.9	μs
t_{BUF}	bus free time before new start	1.3			μs
$t_{HD:STA}$	start condition hold time	0.6			μs
$t_{SU:STA}$	start condition setup time	0.6			μs
$t_{HD:DAT}$	data in hold time	0			μs
$t_{SU:DAT}$	data in setup time	0.1			μs
$t_{SU:STO}$	stop condition setup time	0.6			μs
t_R, t_F	SCCB rise/fall times			0.3	μs
t_{DH}	data out hold time	0.05			μs

[a] SCCB timing is based on 400kHz mode

[b] timing measurement shown at the beginning of the rising edge and/or of the falling edge signifies 30%, timing measurement shown in the middle of the rising/falling edge signifies 50%, timing measurement shown at the beginning of the rising edge and/or of the falling edge signifies 70%

2.11 group write

Group write is supported in order to update a group of registers (except 0x31xx) in the same frame. These registers are guaranteed to be written prior to the internal latch at the frame boundary. If more than one group is going to be launched, the second group cannot be recorded or launched before the first group has effectively been launched.

The OV8858 supports up to four groups. These groups share 1024 bytes of memory and the size of each group is programmable by adjusting the start address.

table 2-11 context switching control

address	register name	default value	R/W	description
0x3208	GROUP ACCESS	–	W	Group Access Bit[7:4]: group_ctrl 0000: Group hold start 0001: Group hold end 1010: Group delay launch 1110: Group quick launch Others: Debug mode Bit[3:0]: Group ID 0000: Group bank 0, default start from address 0x00 0001: Group bank 1, default start from address 0x40 0010: Group bank 2, default start from address 0x80 0011: Group bank 3, default start from address 0xB0 Others: Debug mode
0x3209	GRP0_PERIOD	0x00	RW	Frames For Staying in First Group (must be Group 0) 0 Means Always Stay in Group 0
0x320A	GRP1_PERIOD	0x00	RW	Frames For Staying in Second Group (can be Group 1-3) 0 Means Always Stay in Group 1
0x320B	GRP_SWCTRL	0x01	RW	Bit[7]: Auto switch Bit[3]: group_switch_repeat_en Enable the first group (group 0) and second group repeatable switch Bit[2]: context_en Enable to switch from second group back to first group (group 0) automatically Bit[1:0]: Second group selection
0x320D	GRP_ACT	–	R	Indicates Which Group is Active
0x320E	FRAME_CNT_GRP0	–	R	frame_cnt_grp0
0x320F	FRAME_CNT_GRP1	–	R	frame_cnt_grp1

2.11.1 hold

After the groups are configured, users can perform a hold operation to store register settings into the SRAM of each group. The hold of each group starts and ends with the control register 0x3208. The lower 4 bits of register 0x3208 control which group to access, and the upper 4 bits control the start (0x0: hold start) and end (0x1: hold end) of the hold operation.

The example setting below shows the sequence to hold group 0:

```
6C  3208  00    group 0 hold start
6C  3800  11    first register into group 0
6C  3911  22    second register into group 0
6C  3208  10    group 0 hold end
```

2.11.2 launch

After the contents of each group are defined in the hold operation, all registers belonging to each group are stored in SRAM, and ready to be written into target registers (i.e., the launch of that group).

There are five launch modes as described in sections **section 2.11.2.1** to **section 2.11.2.5**.

2.11.2.1 launch mode 1 - quick manual launch

Manual launch is enabled by setting the register 0x320B to 0.

Quick manual launch is achieved by writing to control register 0x3208. The value written into this register is 0xEX, the upper 4 bits (0xE) are the quick launch command and the lower 4 bits (0xX) are the group number. For example, if users want to launch group 0, they just write the value 0xE0 to 0x3208, then the contents of group 0 will be written to the target registers immediately after the sensor gets this command through the SCCB. Below is a setting example.

```
6C  320B  00    manual launch on
6C  3208  E0    quick launch group 0
```

2.11.2.2 launch mode 2 - delay manual launch

Delay manual launch is achieved by writing to the register 0x3208. The value written into this register is 0xAX, where the upper 4 bits (0xA) are the delay launch command and the lower 4 bits (0xX) are the group number. For example, if users want to launch group 1, they just write the value 0xA1 to 0x3208, then the contents of group 1 will be written to the target registers. The difference with mode 1 is that the writing will wait for some internally defined time spot in vertical blanking, thus delayed. Below is a setting example.

```
6C  320B  00    manual launch on
6C  3208  A1    delay launch group 1
```

2.11.2.3 launch mode 3 - quick auto launch

Quick auto launch works like the mode 1, the difference is it will return to a specified group automatically. This is controlled by the register 0x3209, where bit[6:5] controls which group to return and bit[4:0] controls how many frames to stay before returning. The auto launch enable bit is the 0x320B[7].

The operation can be better understood with a setting example:

```
6C  3209  44    Bit[6:5]: 2, return to group 2, Bit[4:0]: 4: stay 4 frames
6C  320B  80    auto launch on
6C  3208  E0    quick launch group 0
```

In this example, sensor will quick launch group 0, stay at group 0 for 4 frames, then return to group 2 after that.

2.11.2.4 launch mode 4: delay auto launch

Delay auto launch works like mode 2 in the delay launch part and like the mode 3 in the return part.

The operation can be better understood with a setting example:

```
6C  3209  44    Bit[6:5]: 2, return to group 2, Bit[4:0]: 4: stay 4 frames
6C  320B  80    auto launch on
6C  3208  A0    delay launch group 0
```

In this example, sensor will delay launch group 0, stay at group 0 for 4 frames, then return to group 2 after that.

2.11.2.5 launch mode 5: repeat launch

Repeat launch is controlled by registers 0x3209, 0x320A, and 0x320B. In this mode, the launch is repeated automatically between the first group (must be group 0) and the second group (can be either one of groups 1-3, which is specified by register 0x320B[1:0]). The register 0x3209 defines how many frames remain at group 0, and register 0x320A defines how many frames remain at the second group.

The operation can be better understood with a setting example:

```
6C  3209  02    Bit[7:0]: 2, stay 2 frames in group 0
6C  320A  03    Bit[7]: 3, stay 3 frames in the second group
6C  320B  0E    Bit[3:2]: 3, repeat launch on, Bit[1:0]: 2, second group select:
                group 2
6C  3208  A0    always use a0 for repeat launch
```

In this example, sensor will delay launch group 0, stay at group 0 for 2 frames, then switch to group 2 for 3 frames, then back to group 0 for 2 frames, group 2 for 3 frames and so on.

Below is another example to apply launch mode 2 (delay manual launch) first, sensor stays at group 2 for an indefinite number of frames, then apply launch mode 5 (repeat launch). The sensor will switch to group 0 for 2 frames, then group 2 for 3 frames, and so on.

```
6C  320B  00    manual launch on
6C  3208  A2    delay launch group 2 stay at group 2 for indefinite frames
6C  3209  02    Bit[7:0]: 2, stay 2 frames in group 0
6C  320A  03    Bit[7:0]: 3, stay 3 frames in the second group
6C  320B  0E    Bit[3:2]: 3, repeat launch on, Bit[1:0]: 2, second group select:
                group 2
6C  3208  A0    always use A0 for repeat launch
```

Switch to group 0 for 2 frames, then group 2 for 3 frames, and so on.

2.12 register re-mapping

The OV8858 supports register re-mapping function to re-map a source address to a continuous destination one. One use is to make some discontinuous address to a consecutive address, so the user can use sequential write through SCCB to access these registers. This will speed up the SCCB access. The OV8858 supports up to 32 registers which can be re-mapped to a continuous address.

```
6C  3101  32 ; Bit[5] enable re-mapping function
6C  3108  90 ; Destination start address
6C  3109  00
6C  3110  35 ; source address0
6C  3111  00 ;
6C  3112  35 ; source address1
6C  3113  01 ;
6C  3114  35 ; source address2
6C  3115  02 ;
6C  3116  35 ; source address3
6C  3117  08 ;
6C  3118  35 ; source address4
6C  3119  09 ;
```

Then if the user wants to write 0x3500~0x3503 and 0x3508-0x3509:

```
6C  9000  AA; will write 0x3500 to 0xAA
6C  9001  BB; will write 0x3501 to 0xBB
6C  9002  CC; will write 0x3502 to 0xCC
6C  9003  DD; will write 0x3508 to 0xDD
6C  9004  EE; will write 0x3509 to 0xEE
```

table 2-12 register re-mapping

address	register name	default value	R/W	description
0x314E	SRC ADDR1F H	0x00	RW	High Byte of Number 1F Source Register Address
0x314F	SRC ADDR1F L	0x00	RW	Low Byte of Number 1F Source Register Address

2.13 video in video (VIV)

The OV8858 comes with a special Video-in-Video (ViV) feature, which is ideal for front-camera/back-camera video conferencing or photo sharing situations where the ViV output can show both the user and their surroundings. Typical examples include a conference call where both the attendees and the person speaking are able to be recorded, or a scenic outdoor photo taken by a solo photographer who can now frame the shot perfectly alongside their own face. ViV is achieved by the OV8858's MIPI input receiver, which allows for it to be paired with a secondary camera sensor to output ViV mode, overlaying the output of the secondary sensor on top of the OV8858 output in a single image. The secondary sensor can be any sensor whose output resolution is up to 2MP, which includes larger resolution sensors that are sub-sampled to 2MP or lower.

ViV can run in three modes which are shown in **table 2-13**, each of which have two recommended configurations. Full resolution snapshot mode runs at 15fps and uses the OV8858 at full resolution (8MP) with the secondary sensor's output stitched in a ViV window. HD video mode runs at 20fps and uses the OV8858 with a 100% horizontal field of view to output 6MP HD (16:9) video with the secondary sensor's output stitched in a ViV window. Preview video mode runs at 30fps and uses the OV8858 sub-sampled to a 2MP output with the secondary sensor's sub-sampled output stitched in a ViV window. Note that the two preview video mode configurations correspond directly to the same field of view but half the size of each of the respective full resolution snapshot mode configurations.

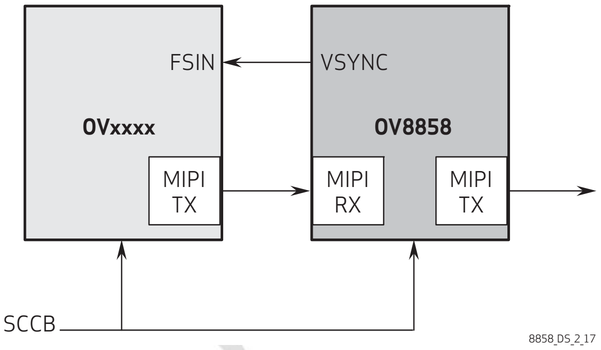
The lens shading correction of the OV8858 output window does not affect the secondary sensor's output window. Special ViV image quality features such as color temperature correction and auto white balance (AWB) compensation for the secondary window make ViV usable even under different lighting conditions for the secondary sensor. The windowed output of the secondary sensor can be moved around, have horizontal borders, and be set to some arbitrary shapes. Furthermore, the OV8858 can also provide statistics on the secondary window which will be valuable for backend ISPs.

The position of the small image can be adjusted by h_offset({0x3A00, 0x3A01}) and v_offset({0x3A02, 0x3A03}), with the frame sync timing of the slave sensor adjusted accordingly. The size of the slave image is configured by width (0x3A04, 0x3A05} and height (0x3A06, 0x3A07), which should be consistent with the external sensor format. The transparency of the background can be adjusted by (0x3A0C, 0x3A0D).

table 2-13 ViV mode details

ViV mode	ViV overlay configuration	OV8858 main sensor resolution	ViV secondary sensor resolution	ViV frame rate
full resolution snapshot	8MP + 2MP	3264x2448	1600x1200	15 fps
	8MP + SVGA	3264x2448	800x600	15 fps
HD video	6MP HD + 1600x900	3264x1836	1600x900	20 fps
	6MP HD + SVGA	3264x1836	800x600	20 fps
preview video	2MP +SVGA	1632x1224	800x600	30 fps
	2MP + 400x300	1632x1224	400x300	30 fps

figure 2-17 video transmission diagram



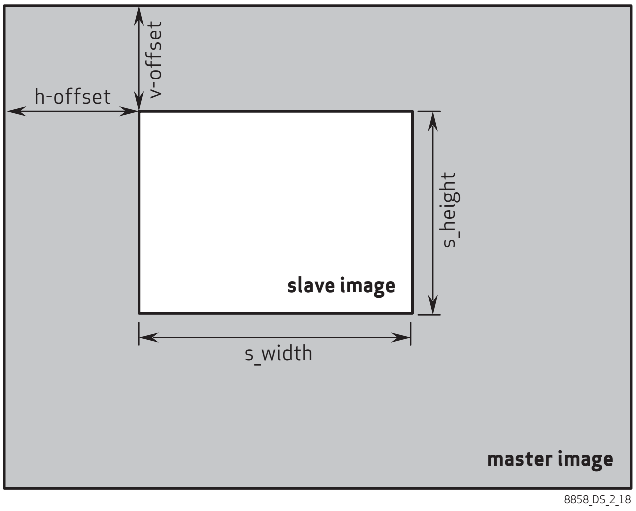
The OV8858 VIV supports the following four modes:

- default mode
- swap mode
- side-by-side mode
- slave sensor bypass mode

2.13.1 default mode

The smaller slave image from external sensor is overlaid on master image from the OV8858 (internal sensor).

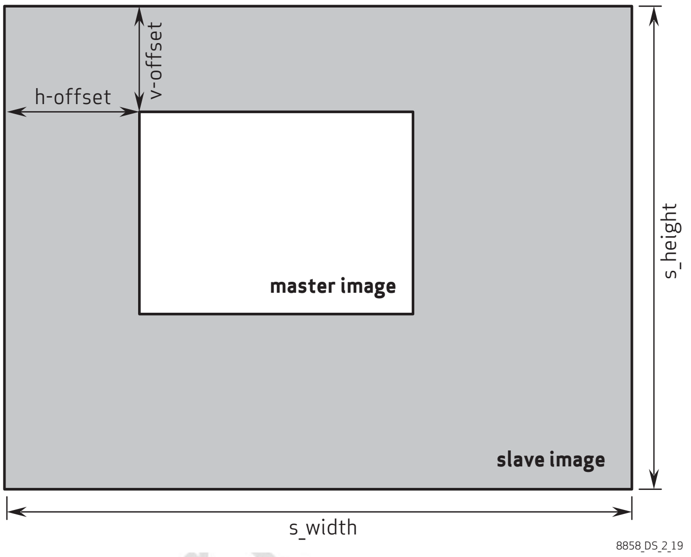
figure 2-18 default mode diagram



2.13.2 swap mode

The smaller master image from the OV8858 (internal sensor) is overlaid on slave image.

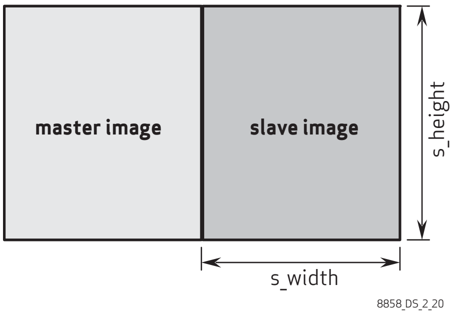
figure 2-19 swap mode diagram



2.13.3 side-by-side mode

The slave image from external sensor is on the side of the master image from the OV8858 (internal sensor).

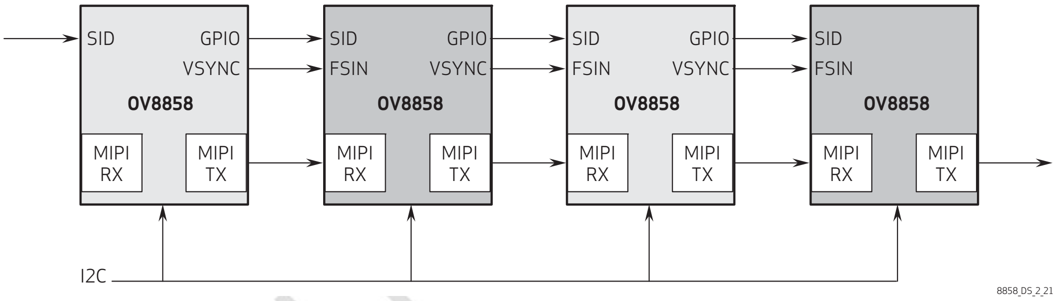
figure 2-20 side-by-side mode diagram



This mode can be combined with a OV8858 MIPI daisy chain for panoramic view applications, in which each sensor would receive the MIPI line data on MIPI RX, combine it with its own line data, and transmit data to the next sensor through MIPI TX.

The daisy chain connection is shown in **figure 2-21**. In this application, each sensor needs both global and individual SCCB IDs. Initially, all sensors in the chain have the same ID1. After the first sensor SID is configured to high, the ID2 is then selected and configured to individual IDx. GPIO is then configured to high to enable the SID of the next sensor and so on. The individual ID can either be loaded from OTP during the initial setting or be programmed right after its SID enabled. Each sensor in the chain will have its individual ID after the configuration and a shared ID with the secondary SCCB ID3.

figure 2-21 MIPI daisy chain diagram

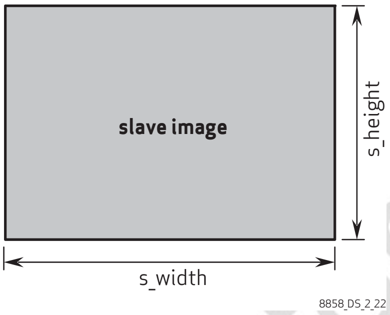


```
20 303f 01; [0] program sensor1 id enable

20 3012 IDx1; change sensor1 ID from ID2(20) to individual IDx1

IDx1 300d 01; [0] set gpio to 1 to enable SID of sensor2

20 303f 01; [0] program sensor2 id enable

20 3012 IDx2; change sensor2 ID from ID2(20) to individual IDx2

IDx2 300d 01; [0] set gpio to 1 to enable SID of sensor3

20 303f 01 ; [0] program sensor3 id enable

20 3012 IDx3; change sensor3 ID from ID2(20) to individual IDx3

IDx3 300d 01; [0] set gpio to 1 to enable SID of sensor4

20 303f 01; [0] program sensor4 id enable

20 3012 IDx4; change sensor4 ID from ID2(20) to individual IDx4

After the ID configuration, each sensor has its own ID (IDx1, IDx2, IDx3, IDx4).
```

2.13.4 slave sensor bypass mode

The OV8858 can work as a bridge chip and only bypass data from the slave sensor.

figure 2-22 slave sensor output diagram

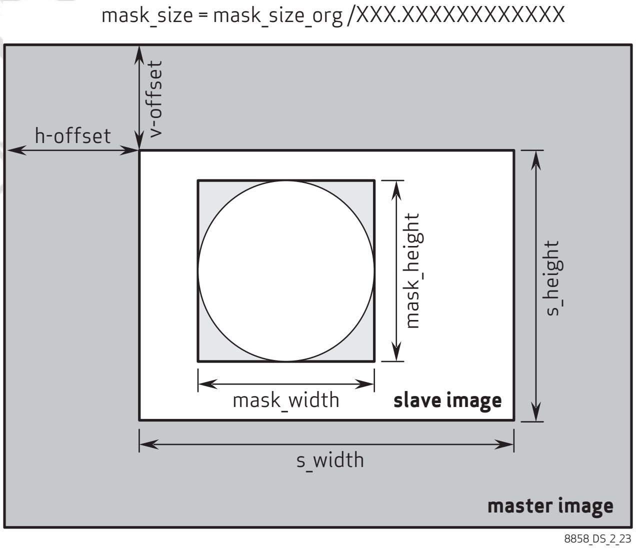


2.13.5 shape function

The OV8858 supports shape function which is enabled by 0x3A12[1]. When this function is enabled, an external slave sensor video, which is contoured by the shape programmed, is overlaid on the OV8858 video. The position of the mask can be controlled by mask_offset ({0x3A08, 0x3A09}) and v_offset ({0x3A0A, 0x3A0B}). The size of the mask is defined in the shape header, but can be manually configured by width (0x3A04, 0x3A05) and height (0x3A06, 0x3A07). The OV8858 supports shape scale function which is enabled by 0x3A0E[0] with 3.13 (3 bit integer and 13 bit decimal) precision controlled by (0x3A0F, 0x3A10).

A border can be added on to the shape, enabled by register 0x3A12[2]. The border width is set by register 0x3A20.

figure 2-23 shape function diagram



The OV8858 has a 4Kx10 bits SRAM to store shape data which is generated by shape compiler. Before programming, the shape function must be turned off and the SRAM program enable bit must be turned on. The SRAM program address would be reset by triggering 0x3A14[0]. Addresses other than the VIV SRAM address (0xA000~0xAFFF) can be accessed in the middle of programming.

```
6c 3a12 01 ; [1] shape function disable
6c 3a0e 14 ; [4] program SRAM enable
6c 3a14 01 ; [0] reset SRAM address
6c a000 byte1
6c a000 byte2
6c a000 byte3
6c a000 byte4
; I2c access other registers
6c REG1 XX
6c REG2 XX
6c REG3 XX
; continue programming
6c a000 byte5
6c a000 byte6
6c a000 byte7
```

table 2-14 register re-mapping (sheet 1 of 3)

address	register name	default value	R/W	description
0x3A00	H_OFFSET	0x00	RW	Bit[3:0]: Small image horizontal offset from large image[11:8]
0x3A01	H_OFFSET	0x00	RW	Bit[7:0]: Small image horizontal offset from large image [7:0]
0x3A02	V_OFFSET	0x00	RW	Bit[3:0]: Small image vertical offset from large image[11:8]
0x3A03	V_OFFSET	0x00	RW	Bit[7:0]: Small image vertical offset from large image[7:0]
0x3A04	EXT_WIDTH	0x02	RW	Bit[3:0]: External image width[11:8]
0x3A05	EXT_WIDTH	0x80	RW	Bit[7:0]: External image width[7:0]

table 2-14 register re-mapping (sheet 2 of 3)

address	register name	default value	R/W	description
0x3A06	EXT_HEIGHT	0x01	RW	Bit[3:0]: External image height[11:8]
0x3A07	EXT_HEIGHT	0xE0	RW	Bit[7:0]: External image height[7:0]
0x3A08	MASK_H_OFFSET	0x00	RW	Bit[3:0]: Mask horizontal offset[11:8]
0x3A09	MASK_H_OFFSET	0x00	RW	Bit[7:0]: Mask horizontal offset[7:0]
0x3A0A	MASK_V_OFFSET	0x00	RW	Bit[3:0]: Mask vertical offset[11:8]
0x3A0B	MASK_V_OFFSET	0x00	RW	Bit[7:0]: Mask vertical offset[7:0]
0x3A0C	DATA_RATIO	0x00	RW	Bit[7:0]: Ratio of large image to small image[15:8]
0x3A0D	DATA_RATIO	0x00	RW	Bit[7:0]: Ratio of large image to small image[7:0]
0x3A0E	CTRL_REG	0x04	RW	Bit[4]: program_sram_en Bit[2]: always_load_header_en Bit[1]: manual_mask_size Bit[0]: manual_scale_en
0x3A0F	SCALE_RATIO	0x00	RW	Bit[7:0]: Mask scale ratio[15:8]
0x3A10	SCALE_RATIO	0x00	RW	Bit[7:0]: Mask scale ratio[7:0]
0x3A11	VIV_CTRL	0x00	RW	Bit[7]: VIV_block_en Bit[6]: swap_en Bit[5]: side_by_side_en
0x3A12	VIV_CTRL	0x00	RW	Bit[2]: border_en Bit[1]: shape_en Bit[0]: VIV_en
0x3A14	SRAM_TRIG	0x00	CS	Bit[0]: SRAM read/write address reset
0x3A16	DATA_OFFSET0	0x00	RW	Bit[7:0]: Data_offset0[7:0]
0x3A17	DATA_OFFSET1	0x00	RW	Bit[7:0]: Data_offset1[7:0]
0x3A18	DATA_OFFSET2	0x00	RW	Bit[7:0]: Data_offset2[7:0]
0x3A19	DATA_OFFSET3	0x00	RW	Bit[7:0]: Data_offset3[7:0]
0x3A1A	DATA_OFFSET	0x00	RW	Bit[7:6]: Data_offset0[9:8] Bit[5:4]: Data_offset1[9:8] Bit[3:2]: Data_offset2[9:8] Bit[1:0]: Data_offset3[9:8]
0x3A1B	DATA_BORDER0	0x00	RW	Bit[7:0]: Data_border0[7:0]
0x3A1C	DATA_BORDER1	0x00	RW	Bit[7:0]: Data_border1[7:0]
0x3A1D	DATA_BORDER2	0x00	RW	Bit[7:0]: Data_border2[7:0]
0x3A1E	DATA_BORDER3	0x00	RW	Bit[7:0]: Data_border3[7:0]

table 2-14 register re-mapping (sheet 3 of 3)

address	register name	default value	R/W	description
0x3A1F	DATA_BORDER	0x00	RW	Bit[7:6]: Data_border0[9:8] Bit[5:4]: Data_border1[9:8] Bit[3:2]: Data_border2[9:8] Bit[1:0]: Data_border3[9:8]
0x3A20	BORDER_WIDTH	0x00	RW	Border_width
0x3A21	MASK_WIDTH	0x00	RW	Bit[3:0]: Manual_mask_width[11:8]
0x3A22	MASK_WIDTH	0x00	RW	Bit[7:0]: Manual_mask_width[7:0]
0x3A23	MASK_HEIGHT	0x00	RW	Bit[3:0]: Manual_mask_height[11:8]
0x3A24	MASK_HEIGHT	0x00	RW	Bit[7:0]: Manual_mask_height[7:0]

3 block level description

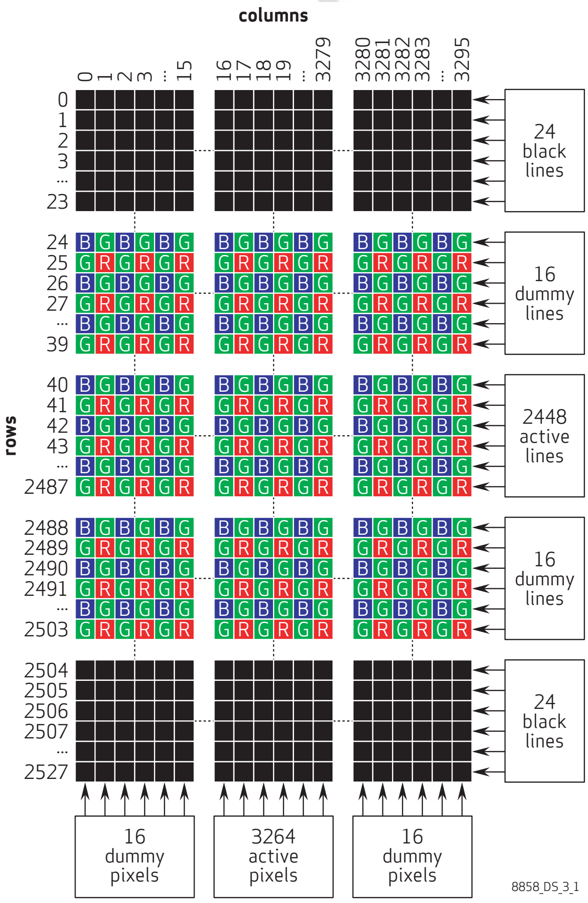
3.1 pixel array structure

The OV8858 sensor has an image array of 3296 columns by 2528 rows (8,332,288 pixels including 48 black lines). **figure 3-1** shows a cross-section of the image sensor array.

The color filters are arranged in a Bayer pattern. The primary color BG/GR array is arranged in line-alternating fashion. Of the 8,332,288 pixels, 7,990,272 (3264x2448) are active pixels and can be output. The other pixels are used for black level calibration and interpolation. The center 3264x2448 pixels is suggested to be output from the whole active pixel array. The backend processor can use the boundary pixels for additional processing.

The sensor array design is based on a field integration readout system with line-by-line transfer and an electronic shutter with a synchronous pixel readout scheme.

figure 3-1 sensor array region color filter layout



3.2 subsampling

The OV8858 supports a binning mode to provide a lower resolution output while maintaining the field of view. With binning mode ON, the voltage levels of adjacent pixels (of the same color) are averaged before being sent to the ADC. The OV8858 supports 2x2 binning, which is illustrated in **figure 3-2**, where the voltage levels of two horizontal (2x1) adjacent same-color pixels are averaged.

figure 3-2 example of 2x2 binning

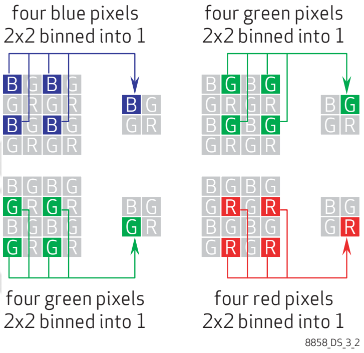


table 3-1 binning-related registers

address	register name	default value	R/W	description
0x3821	TIMING_FORMAT2	0x08	RW	Bit[7]: Vertical sum Bit[5]: Vertical binning Bit[4]: Horizontal binning Bit[3]: ISP horizontal VAR2
0x3814	X_ODD_INC	0x01	RW	Bit[3:0]: Horizontal increase number at odd pixel
0x3815	X_EVEN_INC	0x01	RW	Bit[3:0]: Horizontal increase number at even pixel
0x382A	Y_ODD_INC	0x01	RW	Bit[3:0]: Vertical increase number at odd row
0x382B	Y_EVEN_INC	0x01	RW	Bit[3:0]: Vertical increase number at even row

3.3 analog amplifier

When the column sample/hold circuit has sampled one row of pixels, the pixel data will shift out one-by-one into an analog amplifier.

3.4 10-bit A/D converters

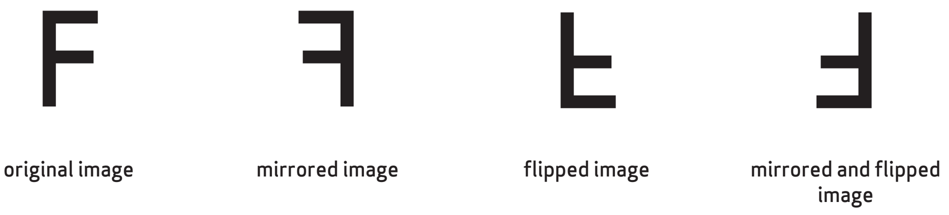
The balanced signal is then digitized by the on-chip 10-bit ADC.

4 image sensor core digital functions

4.1 mirror and flip

The OV8858 provides mirror and flip readout modes, which respectively reverse the sensor data readout order horizontally and vertically (see **figure 4-1**).

figure 4-1 mirror and flip samples



8858_DS_4_1

table 4-1 mirror and flip registers

address	register name	default value	R/W	description
0x3820	FORMAT1	0x00	RW	Timing Control Register Bit[2]: Digital vertical flip enable 0: Normal 1: Vertical flip Bit[1]: Array vertical flip enable 0: Normal 1: Vertical flip
0x3821	FORMAT2	0x00	RW	Timing Control Register Bit[2]: Digital horizontal mirror enable 0: Normal 1: Horizontal mirror Bit[1]: Array horizontal mirror enable 0: Normal 1: Horizontal mirror

4.2 image windowing

An image windowing area is defined by four parameters, horizontal start (HS), horizontal end (HE), vertical start (VS), and vertical end (VE). By properly setting the parameters, any portion within the sensor array size can output as a visible area. Windowing is achieved by masking off the pixels outside of the window; thus, the original timing is not affected.

The OV8858 also supports auto size mode which is controlled by 0x3841[5:0]. Setting it to 0x3F will enable the auto size function and will output the center of the image by default. The user only has to configure the H/V output size (0x3808~0x380B) and registers 0x3842~0x3845 are used to control the offset for auto size mode.

figure 4-2 image windowing

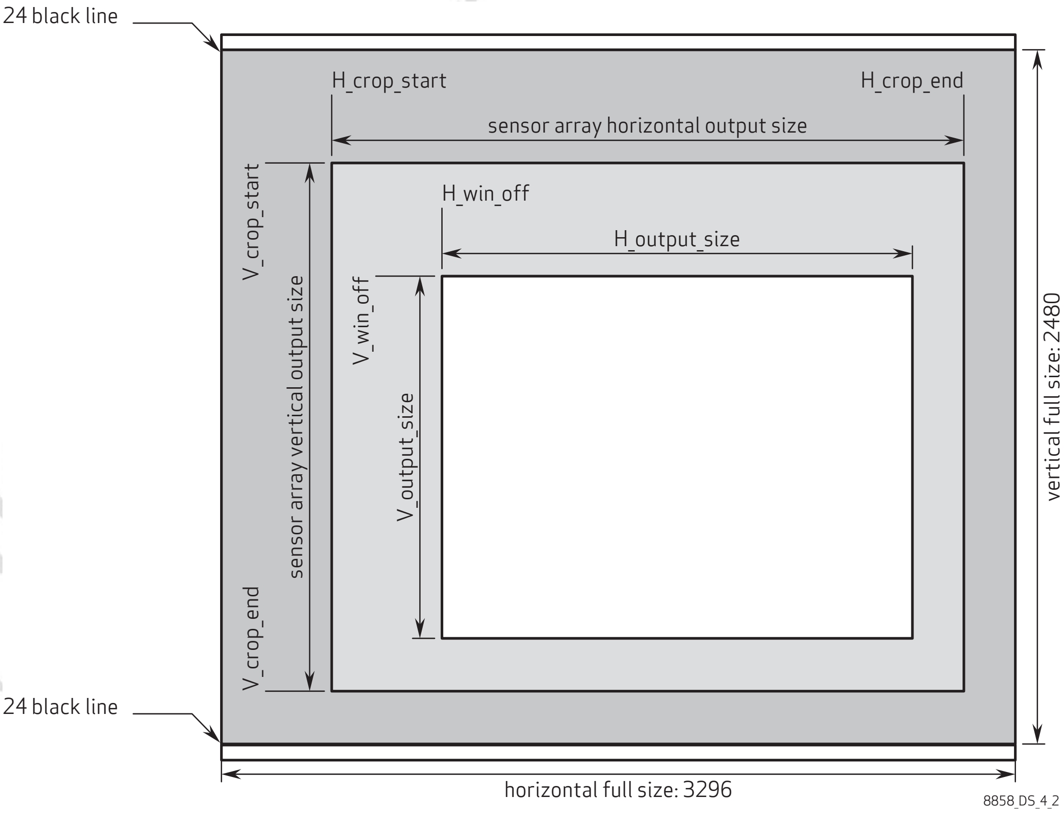


table 4-2 image windowing control functions (sheet 1 of 2)

address	register name	default value	R/W	description
0x3800	H_CROP_START	0x00	RW	Bit[3:0]: Manual horizontal crop start address[11:8]
0x3801	H_CROP_START	0x0C	RW	Bit[7:0]: Manual horizontal crop start address[7:0]
0x3802	V_CROP_START	0x00	RW	Bit[3:0]: Manual vertical crop start address[11:8]
0x3803	V_CROP_START	0x0C	RW	Bit[7:0]: Manual vertical crop start address[7:0]
0x3804	H_CROP_END	0x0C	RW	Bit[3:0]: Manual horizontal crop end address[11:8]
0x3805	H_CROP_END	0xD3	RW	Bit[7:0]: Manual horizontal crop end address[7:0]
0x3806	V_CROP_END	0x09	RW	Bit[3:0]: Manual vertical crop end address[11:8]
0x3807	V_CROP_END	0xA3	RW	Bit[7:0]: Manual vertical crop end address[7:0]
0x3808	H_OURPUT_SIZE	0xDC	RW	Bit[3:0]: Horizontal output size[11:8]
0x3809	H_OUTPUT_SIZE	0xC0	RW	Bit[7:0]: Horizontal output size[7:0]
0x380A	V_OUTPUT_SIZE	0x09	RW	Bit[3:0]: Vertical output size[11:8]
0x380B	V_OUTPUT_SIZE	0x90	RW	Bit[7:0]: Vertical output size[7:0]
0x380C	TIMING_HTS	0x07	RW	Bit[7:0]: Horizontal total size[15:8]
0x380D	TIMING_HTS	0x4C	RW	Bit[7:0]: Horizontal total size[7:0]
0x380E	TIMING_VTS	0x0A	RW	Bit[6:0]: Vertical total size[14:8]
0x380F	TIMING_VTS	0x74	RW	Bit[7:0]: Vertical total size[7:0]
0x3810	H_WIN_OFF	0x00	RW	Bit[3:0]: Manual horizontal windowing offset[11:8]
0x3811	H_WIN_OFF	0x04	RW	Bit[7:0]: Manual horizontal windowing offset[7:0]
0x3812	V_WIN_OFF	0x00	RW	Bit[3:0]: Manual vertical windowing offset[11:8]
0x3813	V_WIN_OFF	0x02	RW	Bit[7:0]: Manual vertical windowing offset[7:0]
0x3814	H_INC_ODD	0x01	RW	Bit[3:0]: Horizontal sub-sample odd increase number
0x3815	H_INC_EVEN	0x01	RW	Bit[3:0]: Horizontal sub-sample even increase number
0x382A	V_INC_ODD	0x01	RW	Bit[3:0]: Vertical sub-sample odd increase number
0x382B	V_INC_EVEN	0x01	RW	Bit[3:0]: Vertical sub-sample even increase number
0x3841	AUTO_SIZE_CTRL	0xFF	RW	Bit[5]: V window auto enable Bit[4]: H window auto enable Bit[3]: V end size auto enable Bit[2]: H end size auto enable Bit[1]: V start size auto enable Bit[0]: H start size auto enable

table 4-2 image windowing control functions (sheet 2 of 2)

address	register name	default value	R/W	description
0x3842	H_AUTO_OFF_H	0x00	RW	Bit[3:0]: H_offset[11:8] for auto size mode Offset is complemental code 0x0001 is to right shift 1 pixel 0xFFFF is to left shift 1 pixel
0x3843	H_AUTO_OFF_L	0x00	RW	Bit[3:0]: H_offset[7:0] for auto size mode
0x3844	V_AUTO_OFF_H	0x00	RW	Bit[3:0]: V_offset[11:8] for auto size mode Offset is complemental code 0x0001 is to up shift 1 row 0xFFFF is to down shift 1 row
0x3845	V_AUTO_OFF_L	0x00	RW	Bit[3:0]: V_offset[7:0] for auto size mode

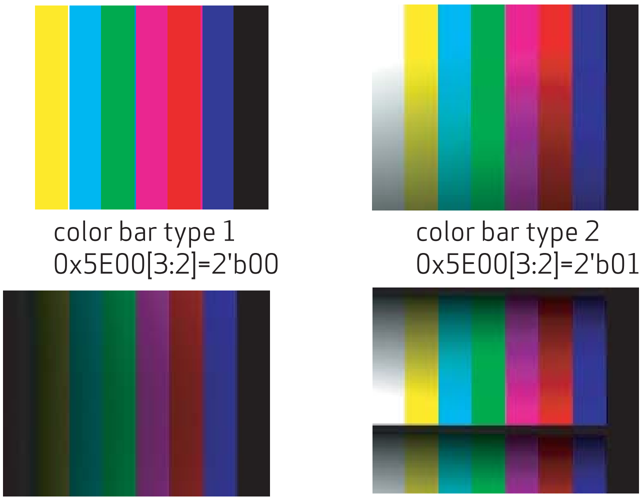
4.3 test pattern

For testing purposes, the OV8858 offers three types of test patterns: color bar, square and random data. The OV8858 also offers two digital effects: transparent effect and rolling bar effect. The output type of digital test pattern is controlled by the test_pattern_type register (0x5E00[3:2]). The digital test pattern function is controlled by register 0x5E00[7].

4.3.1 color bar

There are four types of color bars which are switched by bar-style in register 0x5E00[3:2] (see **figure 4-3**).

figure 4-3 color bar types



color bar type 1
0x5E00[3:2]=2'b00

color bar type 2
0x5E00[3:2]=2'b01

color bar type 3
0x5E00[3:2]=2'10

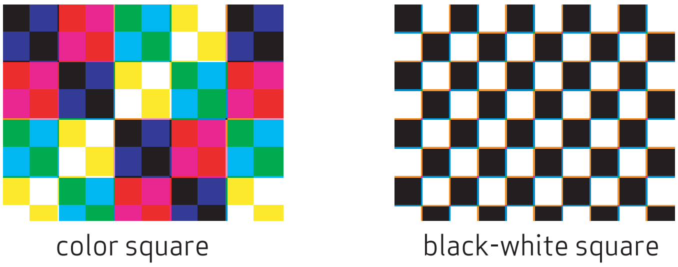
color bar type 4
0x5E00[3:2]=2'b11

8858_DS_4_3

4.3.2 square

There are two types of squares: color square and black-white square. The squ_bw register (0x5E00[4]) decides which type of square will be output.

figure 4-4 color, black and white square bars



color square

black-white square

8858 DS 4 4

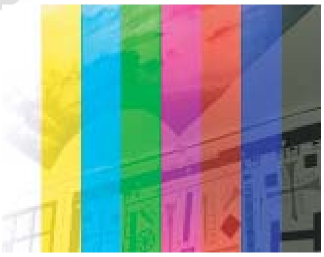
4.3.3 random data

There are two types of random data test patterns: frame-changing and frame-fixed random data.

4.3.4 transparent effect

The transparent effect is enabled by transparent_en register (0x5E00[5]). If this register is set, the transparent test pattern will be displayed. The following image is an example showing a transparent color bar image (see **figure 4-5**).

figure 4-5 transparent effect

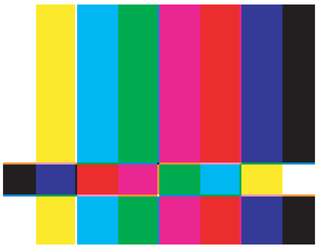


8858_DS_4_5

4.3.5 rolling bar effect

The rolling bar is set by rolling_bar_en register (0x5E00[6]). If it is set, an inverted-color rolling bar will roll from up to down. The following image is an example showing a rolling bar on color bar image (see **figure 4-6**).

figure 4-6 rolling bar effect



8858_DS_4_6

table 4-3 test pattern registers

address	register name	default value	R/W	description
0x5E00	PRE CTRL00	0x00	RW	Bit[7]: Test pattern enable Bit[6]: Rolling bar function enable Bit[5]: Transparent enable 0: Disable transparent effect function 1: Enable transparent effect function Bit[4]: Square mode 0: Color square 1: Black-white square Bit[3:2]: Color bar style 00: Standard color bar 01: Top-bottom darker color bar 10: Right-left darker color bar 11: Bottom-top darker color bar Bit[1:0]: Test pattern mode 00: Color bar 01: Random data 10: Square pattern 11: Black image
0x5E01	PRE CTRL01	0x41	RW	Bit[6]: Window cut enable 0: Do not cut the redundant pixels 1: Cut the redundant pixels Bit[5]: two_lsb_0_en When set, two LSBs of output data are 0 Bit[4]: Same seed enable When set, the seed used to generate the random data are same which is set in seed register Bit[3:0]: Random seed Seed used in generating random data

4.4 average luminance (YAVG)

Exposure time control is based on a frame brightness average value. The OV8858 supports the average image luminance calculation. By properly setting X_start, Y_start, and window_width and window_height, the user can adjust the average based window. **table 4-4** lists the corresponding registers.

figure 4-7 average-based window definition

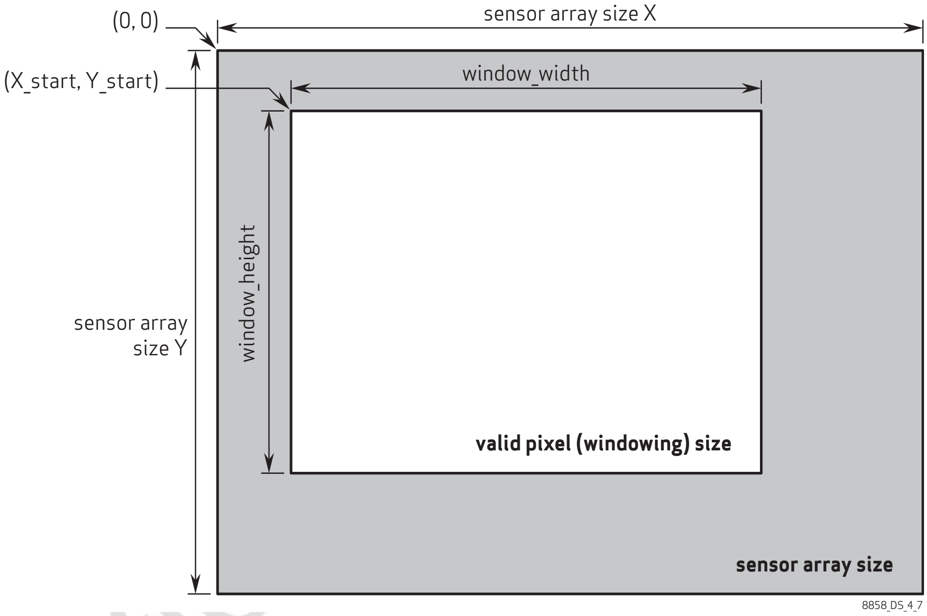


table 4-4 AVG registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5041	ISP CTRL41	0x14	RW	Bit[2]: AVG function enable 0: Disable 1: Enable
0x5680	AVG CTRL00	0x00	RW	Bit[4:0]: X_start_avg[12:8]
0x5681	AVG CTRL01	0x00	RW	Bit[7:0]: X_start_avg[7:0]
0x5682	AVG CTRL02	0x00	RW	Bit[3:0]: Y_start_avg[11:8]
0x5683	AVG CTRL03	0x00	RW	Bit[7:0]: Y_start_avg[7:0]
0x5684	AVG CTRL04	0x0C	RW	Bit[4:0]: Window_width_avg[12:8]
0x5685	AVG CTRL05	0xC0	RW	Bit[7:0]: Window_width_avg[7:0]

table 4-4 AVG registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x5686	AVG CTRL06	0x09	RW	Bit[3:0]: Window_height_avg[11:8]
0x5687	AVG CTRL07	0x90	RW	Bit[7:0]: Window_height_avg[7:0]
0x5688	AVG CTRL08	0x02	RW	Bit[1]: Sum option 0: Sum=(4×B+9×G×2 +10×R)/8 1: Sum=B+G×2+R Bit[0]: Sub-window function enable 0: Use whole output window for average 1: Use registers 0x5680~0x5687 to define window for average
0x568A	AVG RO0A	–	R	Bit[7:0]: High 8 bits of whole image average output

4.5 black level calibration (BLC)

The pixel array contains several optically shielded (black) lines and optically shielded (black) pixels on the right side. These lines and columns are used as reference for black level calibration. The main function of the BLC is to adjust all normal pixel values based on the values of the black levels.

Black level adjustments can be made with registers 0x4000, 0x4004, and 0x4005.

table 4-5 BLC registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x4000	BLC CTRL00	0xF1	RW	Bit[7]: Offset out of range triggers BLC enable Bit[6]: Format change triggers BLC enable Bit[5]: Gain change triggers BLC enable Bit[4]: Exposure change triggers BLC enable Bit[3]: Manually trigger BLC signal Its rising edge will trigger BLC Bit[2]: BLC freeze function enable When set, BLC will be frozen and the offsets will keep the pre-frame values Bit[1]: BLC always triggered enable When set, the BLC will be triggered every frame unless register bit 0x4000[2] is enabled Bit[0]: Five points median filter function enable

table 4-5 BLC registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x4001	BLC CTRL01	0x86	RW	Bit[7]: Offset dithering function enable Bit[6]: Enable difference between black line column with zero line to cancel horizontal noise Bit[5:4]: Column shift option 00: Left 256 columns 01: Left 128 columns 10: Left 64 columns 11: Left 32 columns Bit[2]: Final BLC offset limitation enable Bit[1]: BLC column cancel function enable
0x4002	BLC CTRL02	0x40	RW	Bit[7:0]: Up threshold of cut range function
0x4004	BLC CTRL04	0x00	RW	Bit[7:0]: Target[15:8] High byte of BLC target
0x4005	BLC CTRL05	0x40	RW	Bit[7:0]: Target[7:0] Low byte of BLC target
0x4009	BLC CTRL09	0x29	RW	Bit[7:4]: Line number for BLC initial function Bit[3]: Bypass cut range function enable Bit[2:0]: BLC column added offset[10:8]
0x400A	BLC CTRL0A	0x00	RW	Bit[7:0]: BLC column added offset[7:0]
0x400B	BLC CTRL0B	0x0C	RW	Bit[7:4]: Start line for BLC initial function Bit[3]: Offset limitation function enable Bit[2]: Cut range function enable Bit[1:0]: BLC last line select
0x4011	BLC CTRL11	0x00	RW	Bit[7:6]: Dithering offset Bit[5]: offset_man_same When it is enabled, the manual offsets will be same. They are all defined by manual_offset00. Bit[4]: Offset manual mode enable
0x401E	BLC CTRL1E	0x20	RW	Bit[7:0]: Down threshold of cut range function
0x401F	BLC CTRL1F	0x06	RW	Bit[3]: Rblue BLC reverse Bit[2]: Interpolation x enable Bit[1]: Interpolation y enable Bit[0]: Anchor one enable
0x4030	DCBLC K1	0x01	RW	Bit[3:0]: Dark current BLC top K coefficient[11:8]
0x4031	DCBLC K1	0x00	RW	Bit[7:0]: Dark current BLC top K coefficient[7:0]
0x4032	DCBLC K2	0x01	RW	Bit[3:0]: Dark current BLC bottom K coefficient[11:8]
0x4033	DCBLC K2	0x00	RW	Bit[7:0]: Dark current BLC bottom K coefficient[7:0]

4.6 one time programmable (OTP) memory

The OV8858 supports a maximum of 1024 bytes of one-time programmable (OTP) memory to store chip identification and manufacturing information, which can be used to update the sensor's default setting and can be controlled through the SCCB (see **table 4-6**). Out of 8k bits (1024 bytes), 4k bits are reserved for OmniVision and 4k bits are reserved for customers.

4.6.1 OTP other functions

OTP loading data can be triggered when power up or writing 0x01 to register 0x3D81. Power up loading data is enabled by register 0x3D85[2], by default it is off. Auto mode and manual mode can be chosen by setting register 0x3D84[6] to 0 and 1, respectively, and by default, it is in auto mode. In auto mode, all data in the OTP will be loaded to the OTP buffer; while in manual mode, part of the data which is defined by the start address ({0x3D88,0x3D89}) and the end address ({0x3D8A,0x3D8B}) of the OTP will be loaded to the OTP buffer.

The OV8858 supports loading setting. When 0xDD as a head byte is read out from the start address, which is set by {0x3D8C, 0x3D8D}, setting is recognized. While the setting is being read out from the OTP, it is being written to the OTP buffer, and at the same time, interpreting to the register write command. Loading setting is controlled by registers 0x3D85[1] and 0x3D85[0], which enable power up loading setting and writing register loading setting, respectively.

OTP data can be loaded from 0x7100 to 0x74FF through SCCB interface using a total of 1k bytes. 0x7000 ~ 0x700F and 0x7210 ~ 0x73FF are reserved for OmniVision, while 0x7010 ~ 0x720F (512 bytes) are reserved for customer use.

There are two types of setting format:

1. AX Start Address MSB, Start Address LSB, data0, data1,.... dataX
2. 5X (X can be 0x0 ~ 0xF) data0, data1,.... dataX

Neither AX nor 5X means the end of the setting. 5X means the start address is from the previous end address. X means number of registers is (x+1).

Example: store the setting table in address 0x0100 of OTP. The table content is: DD A3 30 00 11 22 33 44 53 55 66 77 88

which is: 3000-11, 3001-22, 3002-33, 3003-44, 3004-55, 3005-66, 3006-77, 3007-88

To program the OTP:

```
6C 3D84 40; [6]manual mode enable
6C 3D85 00
6C 3D88 71; manual OTP start address for access
6C 3D89 00
6C 3D8A 71; manual OTP end address for access
6C 3D8B 0C
6C 0100 01; stream mode enable
;delay 20ms
6C 7100 DD
6C 7101 A3
6C 7102 30
6C 7103 00
```

```
6C 7104 11
6C 7105 22
6C 7106 33
6C 7107 44
6C 7108 53
6C 7109 55
6C 710A 66
6C 710B 77
6C 710C 88
6C 3D80 01;[0] program enable
;delay 200ms
6C 3D80 00
```

Setting for loading:

```
...
6C 3D88 71; manual OTP start address for access
6C 3D89 00
6C 3D8A 71; manual OTP end address for access
6C 3D8B 0C
6C 3D85 06; [2] OTP load data enable
; [1] OTP load setting enable
6C 3D8C 01; Start address OTP setting table, the first byte of OTP setting table should
be 0xDD
6C 3D8D 00;
6C 0100 01; stream mode enable, after streaming of the first power up, OV8858 will load
setting from OTP if 3D85[2:1]=2'b11
```

The OV8858 supports OTP BIST. When register 0x3D85[4] is set to 1, the BIST function is enabled. When OTP loading data, the data which is read out from the OTP can be compared with zero or the data with the same address in the register, which can be controlled by setting register 0x3D85[5] to 1 or 0, respectively. After the BIST done, the BIST done flag can be read out from register 0x3D81[4], the BIST error flag can be read out from 0x3D81[5], and the address of the first error can be read out from {0x3D8E, 0x3D8F}.

table 4-6 OTP control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x7000~ 0x75FF	OTP_SRAM	0x00	RW	Bit[7:0]: OTP buffer
0x3D80	OTP_PROGRAM_CTRL	–	RW	Bit[7]: OTP_wr_busy (read only) Bit[0]: OTP_program_enable (write only)

table 4-6 OTP control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x3D81	OTP_LOAD_CTRL	–	RW	Bit[7]: OTP_rd_busy (read only) Bit[5]: OTP_bist_error (read only) Bit[4]: OTP_bist_done (read only) Bit[0]: OTP_load_enable (read and write)
0x3D84	OTP_MODE_CTRL	0x00	RW	Bit[7]: Program disable 1: Disable Bit[6]: Mode select 0: Auto mode 1: Manual mode
0x3D85	OTP_REG85	0x13	RW	Bit[5]: OTP_bist_select 0: Compare with SRAM 1: Compare with zero Bit[4]: OTP_bist_enable Bit[2]: OTP power up load data enable Bit[1]: OTP power up load setting enable Bit[0]: OTP write register load setting enable
0x3D88	OTP_START_ADDRESS	0x00	RW	OTP Start High Address for Manual Mode
0x3D89	OTP_START_ADDRESS	0x00	RW	OTP Start Low Address for Manual Mode
0x3D8A	OTP_END_ADDRESS	0x00	RW	OTP End High Address For Manual Mode
0x3D8B	OTP_END_ADDRESS	0x00	RW	OTP End Low Address For Manual Mode
0x3D8C	OTP_SETTING_STT_ADDRESS	0x00	RW	OTP Start High Address For Load Setting
0x3D8D	OTP_SETTING_STT_ADDRESS	0x00	RW	OTP Start Low Address For Load Setting
0x3D8E	OTP_BIST_ERR_ADDRESS	–	R	OTP Check Error Address High
0x3D8F	OTP_BIST_ERR_ADDRESS	–	R	OTP Check Error Address Low
0x3D90	OTP_STROBE_GAP_PGM	0x12	RW	Gap Between Strobe Pulse When Programming
0x3D91	OTP_STROBE_GAP_LOAD	0x06	RW	Gap Between Strobe Pulse When Loading

4.7 temperature sensor

TheOV8858 supports an on-chip temperature sensor that covers -40~192°C with an error range of 5°C. It can be controlled through the SCCB interface (see **table 4-7**). When the readout data is lower than 0xC0, the temperature is a positive value.

If the readout data is higher than 0xC0, the temperature is lower than 0°C and the readout data is twos complement code. Before reading the temperature, the temperature sensor should be triggered by a 0 to 1 transition of register 0x4D12[0].

table 4-7 temperature sensor functions

address	register name	R/W	description
0x4D12	TPM TRIGGER	RW	Bit[0]: Temperature sensor trigger
0x4D13	TPM READ	R	Bit[7:0]: Temperature readout

4.8 strobe flash and frame exposure

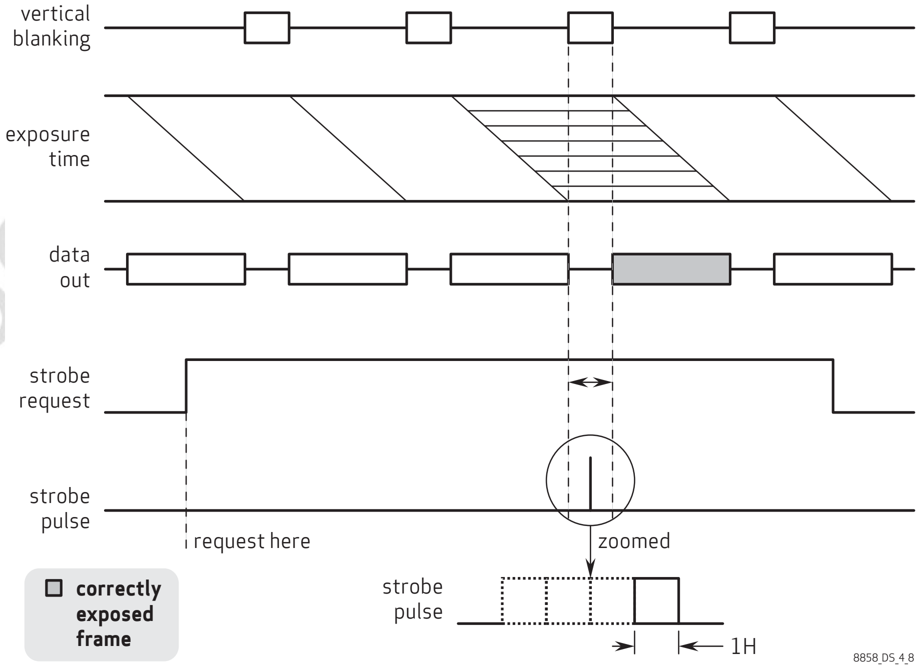
4.8.1 strobe flash control

The strobe signal is programmable using register 0x3B00[2:0]. It supports both LED and Xenon modes. The polarity of the pulse can be changed. The strobe signal is enabled (turned high/low depending on the pulse's polarity) by requesting the signal via the SCCB interface using register bit 0x3B00[7]. Flash modules are triggered by the rising edge by default or by the falling edge if the signal polarity is changed. it supports the following flashing modes: xenon flash control, LED mode 1, LED mode 2, LED mode 3, and LED mode 4.

4.8.1.1 xenon flash control

After a strobe request is submitted, the strobe pulse will be activated at the beginning of the third frame (see figure 4-8). The third frame will be correctly exposed. The pulse width can be changed in Xenon mode between 1H and 4H using register 0x3B00[5:4], where H is one row period.

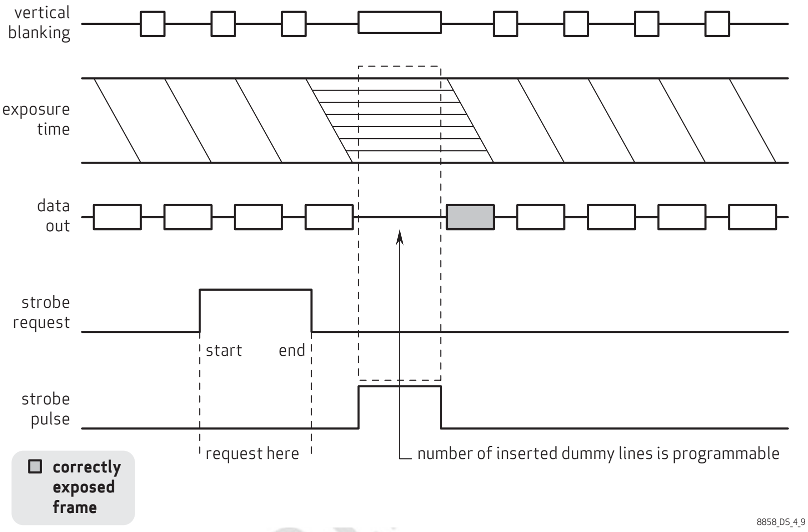
figure 4-8 xenon flash mode



4.8.1.2 LED 1

In LED 1 mode, the strobe signal stays active until the strobe end request is sent (see LED 1 mode).

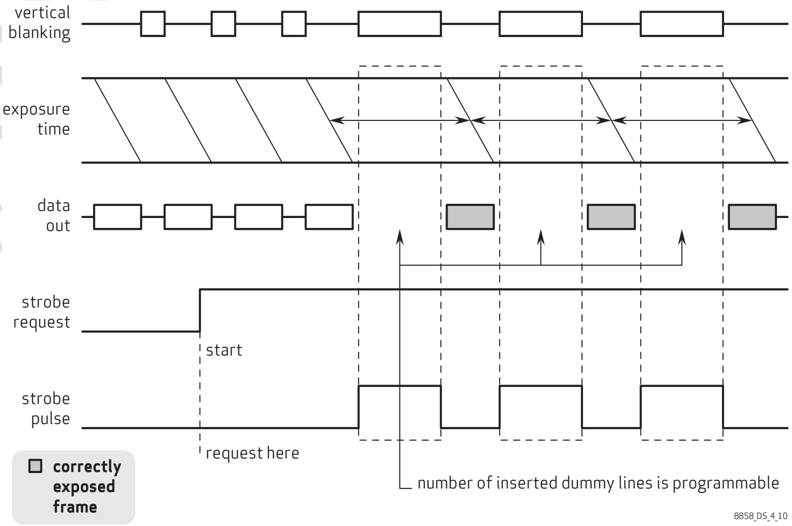
figure 4-9 LED 1



4.8.1.3 LED 2 mode

In LED 2 mode, the strobe signal width can be added by inserting dummy lines which is controlled by register {0x3B02, 0x3B03} (see **section 4-10**).

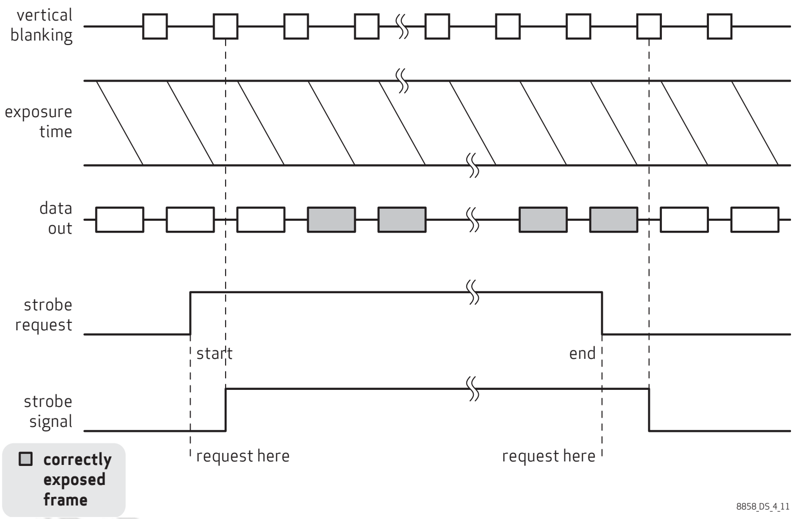
figure 4-10 LED2 mode



4.8.1.4 LED 3 mode

In LED 3 mode, the strobe signal stays active until the strobe end request is sent (see **figure 4-11**).

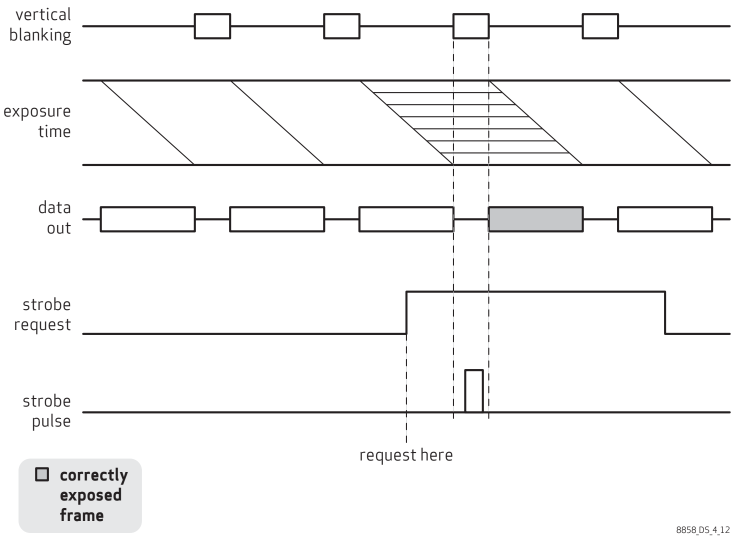
figure 4-11 LED 3 mode



4.8.1.5 LED 4 mode

In LED 4 mode, the strobe signal width is controlled by register 0x3B05 (see **figure 4-12**). Strobe width = 128 × (2^0x3B05[1:0]) × (0x3B05[7:2] + 1) × sclk_period. The maximum value of 0x3B05[7:2] is 6'b111110.

figure 4-12 LED 4 mode



4.9 frame exposure (FREX) mode

In FREX mode, all pixels in the frame start integration at the same time, rather than integrating row by row. After a user-defined exposure time, the mechanical shutter should be closed, preventing further integration, and then the image begins to read out. After the readout finishes, the shutter opens again and the sensor resumes normal mode, waiting for the next FREX request.

The OV8858 supports two modes of FREX (see **figure 4-13** and **figure 4-14**):

figure 4-13 FREX mode 1

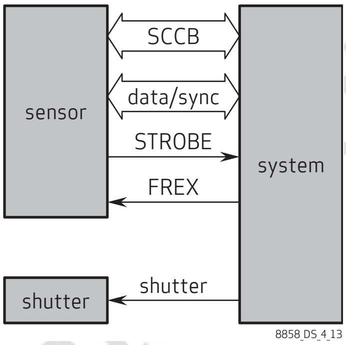
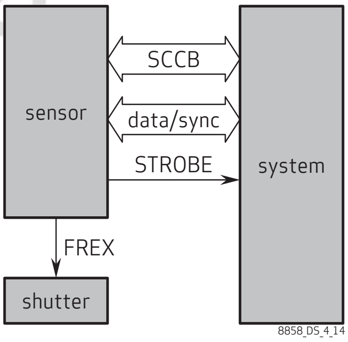


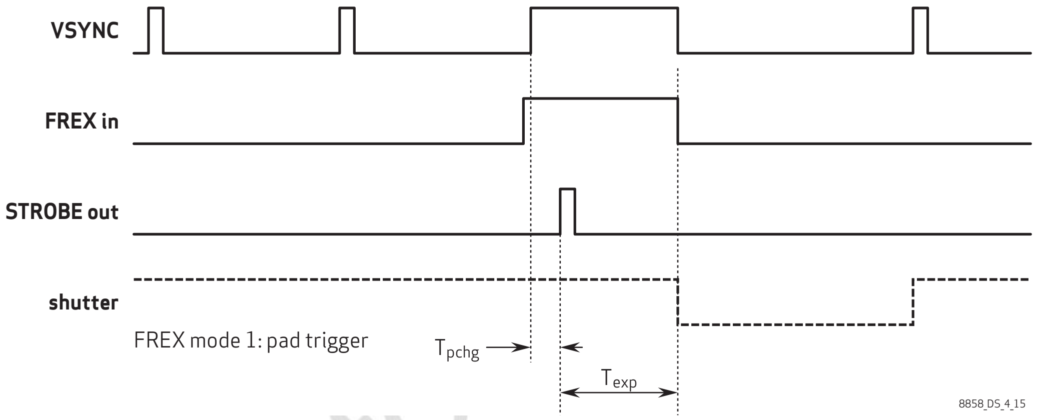
figure 4-14 FREX mode 2



In mode 1, the FREX pin is configured as an input while it is configured as an output in mode 2. In both mode 1 and mode 2, the strobe output is irrelevant with the rolling strobe function. When in rolling shutter mode, the strobe function and this FREX/shutter control function do not work at the same time.

The timing diagram for mode 1 is shown in **figure 4-15**.

figure 4-15 FREX mode 1 timing diagram



In mode 1, the host asserts FREX at any time in preview mode (mechanical shutter is open at this time). The sensor will trigger STROBE to indicate the start of exposure time. Exposure time is calculated from the STROBE rising edge to when the mechanical shutter closes. The host will control when to close the mechanical shutter (shutter delay is handled by the host). The host can re-open the shutter after receiving the entire image data or the next VSYNC signal.

The timing diagrams for mode 2 are shown in **figure 4-16** and **figure 4-17**.

figure 4-16 FREX mode 2 (shutter delay = 0) timing diagram

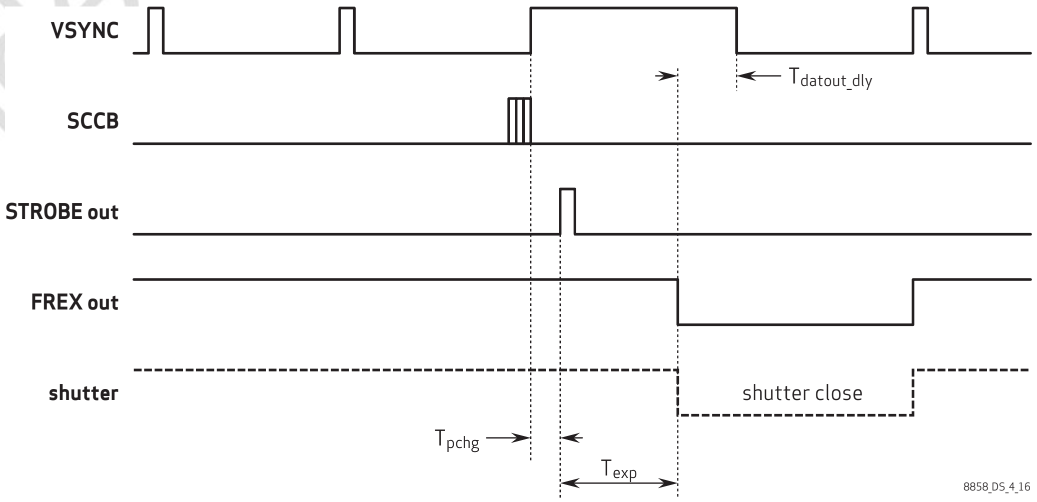
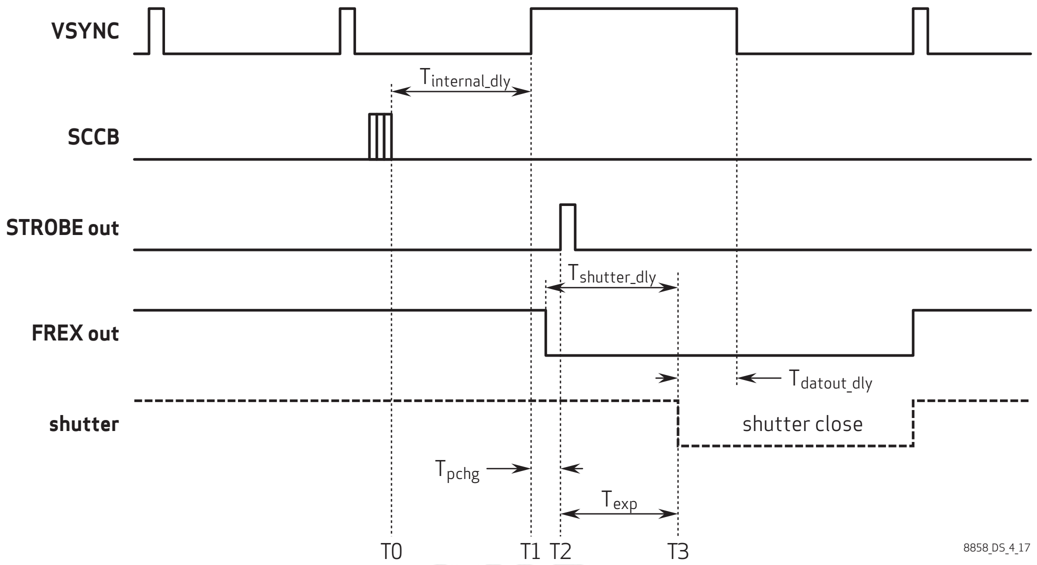


figure 4-17 FREX mode 2 (shutter delay > 0) timing diagram



Before using mode 2, the host needs to program exposure time (registers 0x37C5, 0x37C6, 0x37C7), shutter delay (registers 0x37CC, 0x37CD), strobe width (registers 0x37C9, 0x37CA, 0x37CB), and data output delay. The host triggers this mode by SCCB at any time in preview mode (mechanical shutter is open at this time). The sensor can either start frame exposure right away (since the current data packet is broken, the receiver may get a packet error) or wait for the current frame to finish (controlled by register 0x37DF[0]). If there is no STROBE delay, the sensor will trigger STROBE to indicate the start of exposure time. Exposure time is calculated from STROBE rising edge to when the mechanical shutter closes. Otherwise, the STROBE signal will be sent out even before the sensor begins to pre-charge. The host can control the sensor to start sending image data after a certain delay (registers 0x37D0, 0x37D1) after FREX goes low. The host can re-open the shutter after receiving the entire image data or the next VSYNC signal.

See **table 4-8** for FREX strobe control functions.

table 4-8 FREX strobe control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x3B00	STROBE CTRL	0x00	RW	Bit[7]: Strobe ON/OFF Bit[6]: Strobe polarity 0: Active high 1: Active low Bit[5:4]: width_in_xenon Bit[2:0]: Strobe mode 000: Xenon 001: LED1 010: LED2 011: LED3 100: LED4

table 4-8 FREX strobe control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x3B02	STROBE H	0x00	RW	Dummy Lines Added at Strobe Mode, MSB
0x3B03	STROBE L	0x00	RW	Dummy Lines Added at Strobe Mode, LSB
0x3B04	STROBE CTRL	0x00	RW	Bit[3]: start_point_sel Bit[2]: Strobe repeat enable Bit[1:0]: Strobe latency 00: Strobe generated at next frame 01: Strobe generated 2 frames later 10: Strobe generated 3 frames later 11: Strobe generated 4 frames later
0x3B05	STROBE WIDTH	0x00	RW	Bit[7:2]: Strobe pulse width step Bit[1:0]: Strobe pulse width gain strobe_pulse_width = 128×(2^gain)×(step+1)×Tsclk
0x37C5	FREX REG5	0x00	RW	Bit[7:0]: Frame exposure[23:16] MSB of frame exposure time in mode 2 Exposure time in units of 128 system clock cycles
0x37C6	FREX REG6	0x00	RW	Bit[7:0]: Frame exposure[15:8] Middle byte of frame exposure time in mode 2
0x37C7	FREX REG7	0x08	RW	Bit[7:0]: Frame exposure[7:0] LSB of frame exposure time in mode 2
0x37C9	FREX REG9	0x00	RW	Bit[3:0]: strobe_width[19:16] MSB of strobe width in mode 2. Strobe width in units of 1 system clock cycle
0x37CA	FREX REGA	0x06	RW	Bit[7:0]: strobe_width[15:8] Middle byte of strobe width in mode 2
0x37CB	FREX REGB	0x00	RW	Bit[7:0]: strobe_width[7:0] LSB of strobe width in mode 2
0x37CC	FREX REGC	0x00	RW	Bit[4:0]: shutter_dly[12:8] MSB of shutter delay in mode 2 Shutter delay is in units of 128 system clock cycles
0x37CD	FREX REGD	0x44	RW	Bit[7:0]: shutter_dly[7:0] LSB of shutter delay in mode 2
0x37CE	FREX REGE	0x1F	RW	Bit[7:0]: frex_pre_charge_width[15:8] MSB of sensor precharge in mode 2 Sensor precharge is in units of 1 system clock cycle
0x37CF	FREX REGF	0x40	RW	Bit[7:0]: frex_pre_charge_width[7:0] LSB of sensor precharge in mode 2

table 4-8 FREX strobe control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x37D0	FREX REG10	0x00	RW	Bit[7:0]: Readout delay[15:8] MSB of readout delay time in mode 2 Readout delay time is in units of 128 system clock cycles
0x37D1	FREX REG11	0x01	RW	Bit[7:0]: Readout delay[7:0] LSB of readout delay time in mode 2
0x37D2	FREX_REG12	0x00	RW	Bit[4:0]: Strobe delay[12:8] MSB of strobe delay time
0x37D3	FREX_REG13	0x00	RW	Bit[7:0]: Strobe delay[7:0] LSB of strobe delay time
0x37D4	FREX_REG14	0x32	RW	Bit[7:0]: bst_pchg_en_width /8
0x37D5	FREX_REG15	0x32	RW	Bit[7:0]: bst_pchg_en_gap /8
0x37D6	FREX_REG16	0x00	RW	Bit[3]: bst_prechg_en_sel_eco1234 Bit[2]: bst_prechg_en_all0 Bit[1]: bst_prechg_en_pol
0x379E	FREX REG1E	0x01	RW	Bit[0]: frex_sccb_req_repeat_trig_sel 0: SOF 1: EOF
0x37DF	FREX REG1F	0x04	RW	Bit[7]: frex_sccb_req, self clearing Bit[5]: frex_strobe_out_sel 0: Strobe for rolling mode 1: Strobe for frame mode Bit[4]: frex_nopchg Bit[3]: frex_strobe polarity 0: Active high 1: Active low Bit[2]: frex_shutter polarity Bit[1]: frex_pad_in_enable 0: Frame mode is triggered by register 1: Frame mode is triggered by FREX pad Bit[0]: no_latch at SOF for frex_sccb_req 0: Trigger frame mode in SOF 1: Trigger frame mode immediately

4.9.1 exposure time control

registers: r_frame_exp = {0x37C5, 0x37C6, 0x37C7}, 24 bits, 1 step = 128 clock cycles

minimum exposure time: 0x37C5 = 0x00, 0x37C6 = 0x00, 0x37C7 = 0x00

If the OV8858 works at 160 MHz, the minimum exposure time is 0 and minimum step is 800 ns

maximum exposure time: 0x37C5 = 0xFF, 0x37C6 = 0xFF, 0x37C7 = 0xFF

If the OV8858 works at 160 MHz, the maximum exposure time is 13.42 sec

4.9.2 shutter delay control

registers: r_shutter_dly = {0x37CC[4:0], 0x37CD[7:0]}, 13 bits, 1 step = 128 clock cycles

minimum shutter delay time: 0x37CC = 0x00, 0x37CD = 0x00

Minimum step is 800 ns.

maximum shutter delay time: 0x37CC = 0x1F, 0x37CD = 0xFF.

If the OV8858 works at 160 MHz, the maximum shutter delay time is 6.55 ms.

4.9.3 sensor pre charge control

registers: r_frex_pchg = {0x37CE[7:0], 0x37CF[7:0]}, 16 bits, 1 step = 1 system clock cycle

These registers affect sensor performance. It is for internal use and not recommended for customer to change.

4.9.4 strobe control

Registers: r_strobe_width = {0x37C9[3:0], 0x37CA[7:0], 0x37CB[7:0]}, 20 bits, 1 step = 1 clock cycle.

These registers control the strobe signal output width.

4.10 embedded data

The *MIPI Camera Serial Interface 2 (CSI-2) specification* provides an option to embed sensor internal status information in the picture frame to be delivered to the MIPI host. This feature is especially useful for MIPI host image process. Parameters in sensor, such as ADC gain and exposure time, can help MIPI host to fine tune image processor settings for better image presentation.

The embedded data can be at the beginning or the end of each picture frame by setting register 0x5A08[2]. If embedded information exists, then the lines containing the embedded data must use the embedded data packet data type in the data identifier. It can be configured through register 0x4816 and its default value is 0x13.

table 4-9 embedded data control registers

address	register name	default value	R/W	description
0x4816	EMBEDED DT	0x53	RW	Bit[5:0]: Embedded line data type
0x5A08	EMBEDED FLAG	0x06	RW	Bit[2]: Embedded line flag 0: At start of frame (frame header) 1: At end of frame (frame footer)

table 4-10 describes all data in sensor embedded line for MIPI host image processing. The number in the first column indicates the position of the data with unit in byte, while the second column is the register in that position.

table 4-10 embedded line position data (sheet 1 of 2)

byte number	register name
0	sensor info
1	reserved
2	digital gain
3	not used
4	analog gain[10:8]
5	analog gain[7:0]
6	course integration time[15:8]
7	course integration time[7:0]
8~9	reserved
10	DPC threshold[9:2]
11~14	reserved
15	x_output_size[15:8]

table 4-10 embedded line position data (sheet 2 of 2)

byte number	register name
16	x_output_size[7:0]
17	y_output_size[15:8]
18	y_output_size[7:0]
19~22	reserved
23	MIPI header revision number
24~30	reserved
31	{6'h0, vflip, mirror}
32	frame duration A
33	frame duration B
34	context count
35	context select
36~53	reserved
54	data pedestal[9:2]
55~62	reserved
63	frame average[9:2]
64	digital_gain_red
65	digital_gain_red
66	digital_gain_greenR
67	digital_gain_greenR
68	digital_gain_blue
69	digital_gain_blue
70	digital_gain_greenB
71	digital_gain_greenB
72~88	reserved
89	frame counter
90~94	reserved
95	die temperature
96	temperature decimal

5 image sensor processor digital functions

5.1 ISP top

The main purpose of the ISP top includes:

- integrate all sub-modules
- create necessary control signals

table 5-1 ISP top registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xFE	RW	ISP Control 00 (0: disable; 1: enable) Bit[7]: Lens correction (LENC) function enable Bit[6]: Slave sensor AWB gain function enable Bit[5]: Slave sensor AWB statistics function enable Bit[4]: Master sensor AWB gain function enable Bit[3]: Master sensor AWB statistics function enable Bit[2]: Black DPC function enable Bit[1]: White DPC function enable
0x5001	ISP CTRL01	0x01	RW	Bit[0]: BLC function enable
0x5002	ISP CTRL02	0x28	RW	Bit[7]: Horizontal scale function enable Bit[6]: WBMATCH bypass mode 0: Select slave sensor's gain 1: Select master sensor's gain Bit[5]: WBMATCH function enable Bit[4]: Master MWB gain support RGBC Bit[3]: OTP_DPC function enable Bit[2]: Manual mode of VarioPixel® function enable Bit[1]: Manual enable of VarioPixel® function enable Bit[0]: Use VSYNC to latch ISP modules' function enable signals

table 5-1 ISP top registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x5003	ISP CTRL03	0x20	RW	Bit[6]: Bypass mode Bypass all ISP modules after BLC module Bit[5]: DPC and DBC buffer control enable Bit[4]: WBMATCH VSYNC selection 0: Select master sensor's VSYNC fall 1: Select slave sensor's VSYNC fall Bit[1]: Select master AWB gain to embedded line 0: Select master AWB gain before manual mode 1: Select master AWB gain after manual mode Bit[0]: Enable BLC's input flip_i signal 0: Disable BLC's input flip_i signal 1: Enable BLC's input flip_i signal
0x5004	ISP CTRL4	0x0C	RW	Bit[7]: Bypass master sensor's MWB gain Bit[6]: Manual enable of WBMATCH gain Bit[5]: Manual mode of frame counter for master and slave sensor's average Bit[4]: Manual enable of frame counter for master and slave sensor's average Bit[3]: Auto mode of master sensor's input size Bit[2]: Auto mode of slave sensor's input size Bit[1]: Reverse Gfirst signal of master sensor Bit[0]: Reverse Rblue signal of master sensor
0x501E	ISP CTRL1E	0x91	RW	Bit[7:6]: Select master sensor's average input 00: From pre_DSP 01: From binning post processing 1x: From VarioPixel Bit[5]: Select slave sensor's average input 0: Before slave MWB gain 1: After slave MWB gain Bit[4]: Master sensor AWB statistics input data selection 0: Before master MWB gain 1: After master MWB gain Bit[3]: Manual enable of master AWB statistics Bit[2]: Manual enable of slave AWB statistics Bit[1]: Digital gain function enable, shared with master sensor's MWB gain Bit[0]: Manual mode of master sensor's MWB gain

table 5-1 ISP top registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x501F	ISP CTRL1F	0x00	RW	Bit[5]: Bypass ISP enable Only bypass pre_DSP, LENC, MWB gain, OTP, DPC, bin and hscale Bit[4]: Bit shift enable, when ISP bypass is disabled Bit[3]: Bit shift direction 0: Left shift 1: Right shift Bit[2:0]: Bit shift number
0x5026	BLC BIAS SLV	0x10	RW	Bit[7:0]: Slave sensor’s BLC target
0x5041	ISP CTRL41	0x1C	RW	Bit[4]: Binning post processing function enable Bit[3]: VIV crop window enable in average module Bit[2]: Average module function enable Bit[1]: Do average or sub-sample before AWB statistics for both slave and master sensor 0: Horizontal sub-sample 1: Horizontal average Bit[0]: Embedded line enable
0x5043	ISP CTRL43	0x08	RW	Bit[4:3]: Subtract offset for average module's height Bit[2]: Horizontal post binning enable Bit[1]: Vertical post binning enable Bit[0]: Manual mode of post binning function enable (in auto mode, post binning will be disabled automatically if image size is larger than buffer size)
0x5048	BLC BIAS MAN	0x10	RW	Bit[7:0]: Master sensor's manual BLC target
0x5064	MASTER AVG	–	R	Bit[7:0]: High 8 bits of master sensor's average
0x5065	SLAVE AVG	–	R	Bit[7:0]: High 8 bits of slave sensor's average

5.2 pre_DSP

The main purposes of the pre_DSP module include:

- adjust HREF, valid, RBlue signals and data
- create color bar image
- determine the sizes of input image by removing redundant data
- create control signals

table 5-2 pre_DSP registers

address	register name	default value	R/W	description
0x5E00	PRE CTRL00	0x00	RW	Bit[7]: Test pattern enable Bit[6]: Rolling bar function enable Bit[5]: Transparent enable 0: Disable transparent effect function 1: Enable transparent effect function Bit[4]: Square mode 0: Color square 1: Black-white square Bit[3:2]: Color bar style 00: Standard color bar 01: Top-bottom darker color bar 10: Right-left darker color bar 11: Bottom-top darker color bar Bit[1:0]: Test pattern mode 00: Color bar 01: Random data 10: Square pattern 11: Black image
0x5E01	PRE CTRL01	0x41	RW	Bit[6]: Window cut enable Bit[5]: two_lsb_0_en Set lowest two bits to 0 Bit[4]: Same seed enable Reset seed to 0x5E01[3:0] each frame Bit[3:0]: Random seed Seed used in generating random data

5.3 defective pixel cancellation (DPC)

The DPC uses a one line buffer and removes defect pixels. It also supports black/white mode.

table 5-3 DPC control registers

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0x96	RW	Bit[2]: Black DPC function enable Bit[1]: White DPC function enable

5.4 window cut (WINC)

The main purpose of the WINC module is to make the image size to be real size by removing offset.

table 5-4 WINC registers

address	register name	default value	R/W	description
0x5A00	WINC CTRL00	0x00	RW	Bit[3:0]: X_start offset[11:8]
0x5A01	WINC CTRL01	0x00	RW	Bit[7:0]: X_start offset[7:0]
0x5A02	WINC CTRL02	0x00	RW	Bit[3:0]: Y_start offset[11:8]
0x5A03	WINC CTRL03	0x00	RW	Bit[7:0]: Y_start offset[7:0]
0x5A04	WINC CTRL04	0x0C	RW	Bit[3:0]: Window width[11:8]
0x5A05	WINC CTRL05	0xE0	RW	Bit[7:0]: Window width[7:0]
0x5A06	WINC CTRL06	0x09	RW	Bit[3:0]: Window height[11:8]
0x5A07	WINC CTRL07	0xB0	RW	Bit[7:0]: Window height[7:0]
0x5A08	WINC CTRL08	0x06	RW	Bit[2]: Select embed line flag 0: Select first line as embed flag 1: Select last line as embed flag Bit[1]: Window enable option 0: Disable window after last valid line 1: Get enable from register Bit[0]: Manual window enable

5.5 white balance gain match (WBMATCH)

The main purpose of the WBMATCH module is for adjusting the slave sensor's color temperature to match the master sensor based on the AWB statistics results from both master sensor and slave sensor. Thus, both sensors have the same color temperature before VIV merging. There is an option to select the target color temperature, either before master MWB gain or after it.

There is also a manual mode of WBMATCH gain. The automatically calculated WBMATCH gain can be read out from read only registers.

WBMATCH only supports RGB Bayer pattern.

table 5-5 WBMATCH control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xFE	RW	Bit[6]: Slave sensor AWB gain function enable Bit[5]: Slave sensor AWB statistics function enable Bit[4]: Master sensor AWB gain function enable Bit[3]: Master sensor AWB statistics function enable
0x5004	ISP CTRL4	0x0C	RW	Bit[6]: Manual enable of WBMATCH gain
0x501E	ISP CTRL1E	0x91	RW	Bit[4]: Select master sensor's AWB statistics input 0: Before master MWB gain 1: After master MWB gain Bit[3]: Manual enable of master AWB statistics Bit[2]: Manual enable of slave AWB statistics
0x5012	WBMATCH R GAIN	0x04	RW	Bit[3:0]: Manual WBMATCH R gain for slave sensor[11:8]
0x5013	WBMATCH R GAIN	0x00	RW	Bit[7:0]: Manual WBMATCH R gain for slave sensor[7:0]
0x5014	WBMATCH G GAIN	0x04	RW	Bit[3:0]: Manual WBMATCH G gain for slave sensor[11:8]
0x5015	WBMATCH G GAIN	0x00	RW	Bit[7:0]: Manual WBMATCH G gain for slave sensor[7:0]
0x5016	WBMATCH B GAIN	0x04	RW	Bit[3:0]: Manual WBMATCH B gain for slave sensor[11:8]

table 5-5 WBMATCH control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x5017	WBMATCH B GAIN	0x00	RW	Bit[7:0]: Manual WBMATCH B gain for slave sensor[7:0]
0x5056	AUTO CALCULATED WBMATCH RED GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Auto calculated WBMATCH red gain[11:8]
0x5057	AUTO CALCULATED WBMATCH RED GAIN	–	R	Bit[7:0]: Auto calculated WBMATCH red gain[7:0]
0x5058	AUTO CALCULATED WBMATCH GREEN GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Auto calculated WBMATCH green gain[11:8]
0x5059	AUTO CALCULATED WBMATCH GREEN GAIN	–	R	Bit[7:0]: Auto calculated WBMATCH green gain[7:0]
0x505A	AUTO CALCULATED WBMATCH BLUE GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Auto calculated WBMATCH blue gain[11:8]
0x505B	AUTO CALCULATED WBMATCH BLUE GAIN	–	R	Bit[7:0]: Auto Calculated WBMATCH blue gain[7:0]

5.6 AVG

The main function of the AVG module is to calculate the luminance average using special filters.

table 5-6 AVG control registers

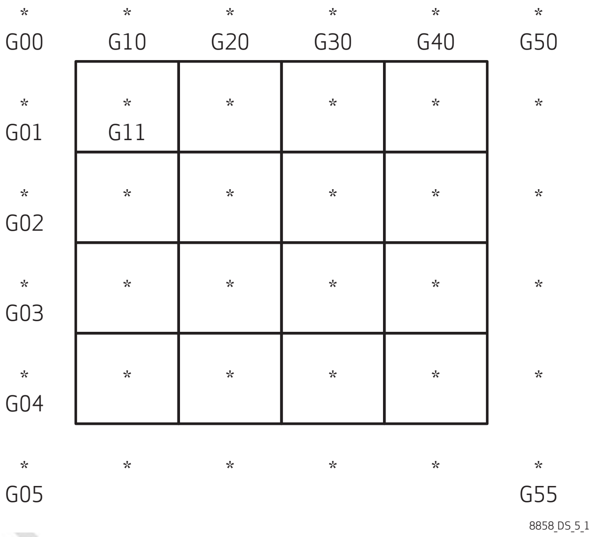
address	register name	default value	R/W	description
0x5004	ISP CTRL04	0x0C	RW	Bit[5]: Manual value of frame counter for AVG Bit[4]: Manual enable of frame counter for AVG
0x5041	ISP CTRL41	0x1C	RW	Bit[3]: VIV window enable for AVG Bit[2]: AVG function enable 0: Disable 1: Enable
0x5680	AVG CTRL00	0x00	RW	Bit[4:0]: X_start_avg[12:8]
0x5681	AVG CTRL01	0x00	RW	Bit[7:0]: X_start_avg[7:0]
0x5682	AVG CTRL02	0x00	RW	Bit[3:0]: Y_start_avg[11:8]
0x5683	AVG CTRL03	0x00	RW	Bit[7:0]: Y_start_avg[7:0]
0x5684	AVG CTRL04	0x0C	RW	Bit[4:0]: Window_width_avg[12:8]
0x5685	AVG CTRL05	0xC0	RW	Bit[7:0]: Window_width_avg[7:0]
0x5686	AVG CTRL06	0x09	RW	Bit[3:0]: Window_height_avg[11:8]
0x5687	AVG CTRL07	0x90	RW	Bit[7:0]: Window_height_avg[7:0]
0x5688	AVG CTRL08	0x02	RW	Bit[1]: Sum option 0: Sum=(4×B+9×G×2 +10×R)/8 1: Sum=B+G×2+R Bit[0]: Sub-window function enable 0: Use whole output window for average 1: Use registers 0x5680~0x5687 to define window for average
0x568A	AVG RO0A	–	R	Bit[7:0]: High 8 bits of whole image average output
0x5064	MASTER AVG	–	R	Bit[7:0]: High 8 bits of master sensor's average
0x5065	SLAVE AVG	–	R	Bit[7:0]: High 8 bits of slave sensor's average

5.7 lens correction (LENC)

The LENC algorithm compensates for the illumination drop off in the corners due to the lens. Based on the radius of each pixel to the lens, the algorithm calculates a gain for each pixel and then corrects each pixel with the calculated gain to compensate for the light distribution due to the lens curvature. Additionally, the LENC supports subsampling in both the horizontal and vertical directions. LENC is performed in the RGB domain.

Both luminance channel and color channel consists of 36 control points.

figure 5-1 control points of luminance and color channels



note There is a lens calibration tool that can be used for calibrating these settings required for a specific module. Contact your local OmniVision FAE for generating these settings.

figure 5-2 luminance compensation level calculation

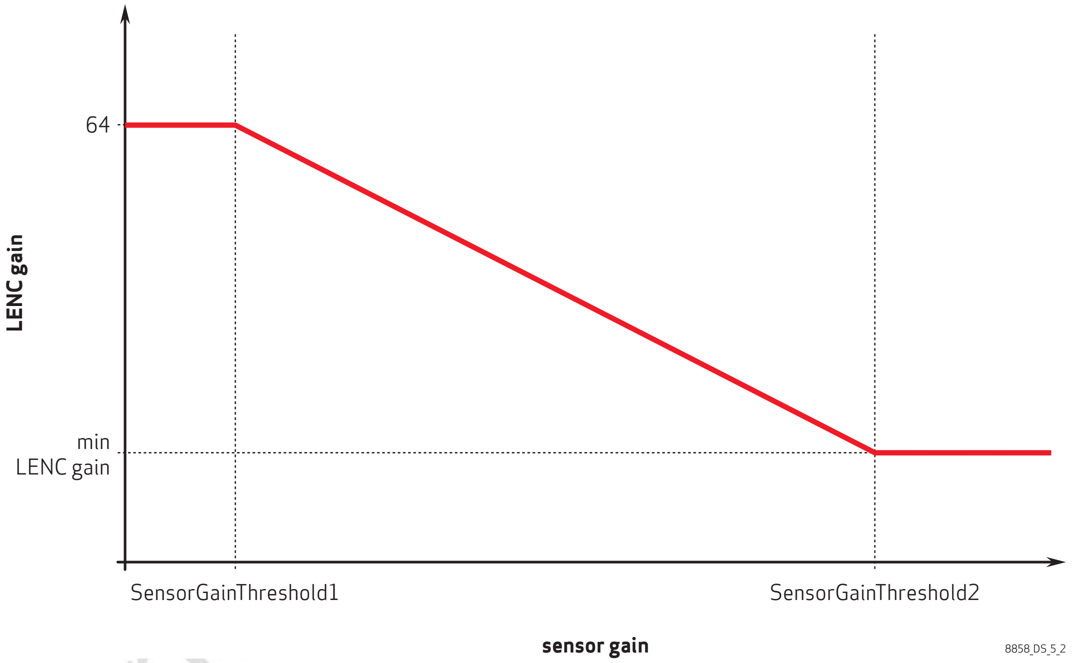


table 5-7 LENC control registers (sheet 1 of 4)

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xFE	RW	Bit[7]: Lens correction (LENC) function enable 0: Disable 1: Enable
0x5800	LENC G00	0x10	RW	Bit[5:0]: Control point G00 for luminance compensation
0x5801	LENC G01	0x10	RW	Bit[5:0]: Control point G01 for luminance compensation
0x5802	LENC G02	0x10	RW	Bit[5:0]: Control point G02 for luminance compensation
0x5803	LENC G03	0x10	RW	Bit[5:0]: Control point G03 for luminance compensation
0x5804	LENC G04	0x10	RW	Bit[5:0]: Control point G04 for luminance compensation
0x5805	LENC G05	0x10	RW	Bit[5:0]: Control point G05 for luminance compensation

table 5-7 LENC control registers (sheet 2 of 4)

address	register name	default value	R/W	description
0x5806	LENC G10	0x10	RW	Bit[5:0]: Control point G10 for luminance compensation
0x5807	LENC G11	0x08	RW	Bit[5:0]: Control point G11 for luminance compensation
0x5808	LENC G12	0x08	RW	Bit[5:0]: Control point G12 for luminance compensation
0x5809~ 0x5822	LENC G13~LENC G54	–	RW	Bit[5:0]: Control point G13~G54 for luminance compensation
0x5823	LENC G55	0x10	RW	Bit[5:0]: Control point G55 for luminance compensation
0x5824	LENC B00	0x14	RW	Bit[5:0]: Control point B00 for blue channel compensation
0x5825	LENC B01	0x14	RW	Bit[5:0]: Control point B01 for blue channel compensation
0x5826	LENC B02	0x14	RW	Bit[5:0]: Control point B02 for blue channel compensation
0x5827	LENC B03	0x14	RW	Bit[5:0]: Control point B03 for blue channel compensation
0x5828	LENC B04	0x14	RW	Bit[5:0]: Control point B04 for blue channel compensation
0x5829	LENC B05	0x14	RW	Bit[5:0]: Control point B05 for blue channel compensation
0x582A	LENC B10	0x14	RW	Bit[5:0]: Control point B10 for blue channel compensation
0x582B	LENC B11	0x12	RW	Bit[5:0]: Control point B11 for blue channel compensation
0x582C	LENC B12	0x12	RW	Bit[5:0]: Control point B12 for blue channel compensation
0x582D~ 0x5846	LENC B13~LENC B54	–	RW	Bit[5:0]: Control point B13~B54 for blue channel compensation
0x5847	LENC B55	0x14	RW	Bit[5:0]: Control point B55 for blue channel compensation
0x5848	LENC R00	0x14	RW	Bit[5:0]: Control point R00 for red channel compensation
0x5849	LENC R01	0x14	RW	Bit[5:0]: Control point R01 for red channel compensation
0x584A	LENC R02	0x14	RW	Bit[5:0]: Control point R02 for red channel compensation

table 5-7 LENC control registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x584B	LENC R03	0x14	RW	Bit[5:0]: Control point R03 for red channel compensation
0x584C	LENC R04	0x14	RW	Bit[5:0]: Control point R04 for red channel compensation
0x584D	LENC R05	0x14	RW	Bit[5:0]: Control point R05 for red channel compensation
0x584E	LENC R10	0x14	RW	Bit[5:0]: Control point R10 for red channel compensation
0x584F	LENC R11	0x12	RW	Bit[5:0]: Control point R11 for red channel compensation
0x5850	LENC R12	0x12	RW	Bit[5:0]: Control point R12 for red channel compensation
0x5851~ 0x586A	LENC R13~LENC R54	–	RW	Bit[5:0]: Control point R13~R54 for red channel compensation
0x586B	LENC R55	0x14	RW	Bit[5:0]: Control point R55 for red channel compensation
0x586C	LENC BOFFSET	0x30	RW	Bit[6:0]: Base value for all blue channel control points
0x586D	LENC ROFFSET	0x30	RW	Bit[6:0]: Base value for all red channel control points
0x586E	LENC MAXGAIN	0x40	RW	Bit[7:0]: If AutoLensSwitchEnable is true and sensor gain is larger than this threshold, luminance compensation amplitude will be the minimum value (min LENC gain). Register value is 16 times sensor gain
0x586F	LENC MINGAIN	0x20	RW	Bit[7:0]: If AutoLensSwitchEnable is true and sensor gain is larger than this threshold, luminance compensation amplitude will start to decrease; otherwise, the amplitude will not change. Register value is 16 times sensor gain.
0x5870	LENC MINQ	0x18	RW	Bit[6:0]: This value indicates the minimum amplitude which luminance channel compensates when AutoLensSwitchEnable is true. Value should be in the range [0~64]

table 5-7 LENC control registers (sheet 4 of 4)

address	register name	default value	R/W	description
0x5871	LENC CTRL	0x0D	RW	Bit[3]: Add BLC target after applying compensation Bit[2]: Enable BLC target for LENC 0: Disable BLC target 1: Enable BLC target Bit[0]: AutoLensSwitchEnable 0: Luminance compensation amplitude does not change with sensor gain 1: Luminance compensation amplitude changes with sensor gain
0x5872	LENC HSCALE	0x01	RW	Bit[2:0]: HScale[10:8] For horizontal gain calculation, this value indicates the step between two connected horizontal pixels, where hscale = 4×2^18 / image width
0x5873	LENC HSCALE	0x3E	RW	Bit[7:0]: HScale[7:0]
0x5874	LENC VSCALE	0x00	RW	Bit[2:0]: VScale[10:8] For vertical gain calculation, this value indicates the step between two connected vertical pixels, where vscale = 4×2^17 / image height
0x5875	LENC VSCALE	0xD3	RW	Bit[7:0]: VScale[7:0]

5.8 manual exposure compensation/ manual gain compensation (MEC/MGC)

Manual exposure provides exposure time settings and sensor gain.The exposure value in register 0x3500~0x3502 and 0x3510~0x3512 are in units of 1/16 line.

Manual gain provides analog gain settings. The OV8858 has a maximum 16x analog gain.

note
For optimal performance, maximum exposure should be 200ms. For more details, contact your local OmniVision FAE.

table 5-8 MEC/MGC control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x3500	LONG EXPO	0x00	RW	Bit[3:0]: Long exposure[19:16]
0x3501	LONG EXPO	0x02	RW	Bit[7:0]: Long exposure[15:8]
0x3502	LONG EXPO	0x00	RW	Bit[7:0]: Long Exposure[7:0] Low 4 bits are fraction bits
0x3503	AEC MANUAL	0x00	RW	Bit[6]: Digital fraction gain delay option 0: Delay 1 frame 1: Not delay 1 frame Bit[5]: Gain change delay option 0: Delay 1 frame 1: Not delay 1 frame Bit[4]: Gain delay option 0: Delay 1 frame 1: Not delay 1 frame Bit[2]: Gain manual as sensor gain 0: Input gain as real gain format 1: Input gain as sensor gain format Bit[1]: Exposure delay option (must be 0) 0: Delay 1 frame 1: Not used Bit[0]: Exposure change delay option (must be 0) 0: Delay 1 frame 1: Not used
0x3505	GCVT OPTION	0x80	RW	Gain Conversation Option Bit[7]: DAC fixed gain bit Bit[5:4]: Sensor gain fixed bit Bit[3:2]: Sensor gain pregain option (debug only, always set it to 0) Bit[1:0]: Sensor gain option for transferring real gain to sensor gain format
0x3507	AEC GAIN SHIFT	0x00	RW	Bit[1:0]: Gain shift option 00: Not shift 01: Left shift 1 bit 10: Left shift 2 bits 11: Left shift 3 bits
0x3508	LONG GAIN	0x00	RW	Bit[4:0]: Long gain[12:8]

table 5-8 MEC/MGC control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x3509	LONG GAIN	0x80	RW	Bit[7:0]: Long gain[7:0] 0x3503[2]=0, gain[7:0] is real gain format, where low 4 bits are fraction bits, for example, 0x10 is 1x gain, 0x28 is 2.5x gain If 0x3503[2]=1, gain[7:0] is sensor gain format, gain[7:4] is coarse gain, 00000: 1x, 00001: 2x, 00011: 4x, 00111: 8x, gain[7] is 1, gain[3:0] is fine gain. For example, 0x10 is 1x gain, 0x30 is 2x gain, 0x70 is 4x gain
0x350A	LONG DIGIGAIN	0x08	RW	Bit[7:0]: Long digital gain[13:6]
0x350B	LONG DIGIGAIN	0x00	RW	Bit[5:0]: Long digital gain[5:0] Low 10 bits are fraction bits
0x350C	SHORT GAIN	0x00	RW	Bit[4:0]: Short gain[12:8]
0x350D	SHORT GAIN	0x80	RW	Bit[7:0]: Short gain[7:0]
0x350E	SHORT DIGIGAIN	0x08	RW	Bit[7:0]: Short digital gain[13:6]
0x350F	SHORT DIGIGAIN	0x00	RW	Bit[5:0]: Short digital gain[5:0] Low 10 bits are fraction bits
0x3510	SHORT EXPO	0x00	RW	Bit[3:0]: Short exposure[19:16]
0x3511	SHORT EXPO	0x02	RW	Bit[7:0]: Short exposure[15:8]
0x3512	SHORT EXPO	0x00	RW	Bit[7:0]: Short exposure[7:0] Low 4 bits are fraction bits

5.9 hscale

Horizontal scale down module only supports four kinds of scale mode: 0.6x, 0.5x, 0.4x, 1/3x. Both auto and manual modes support choosing the width of after-scale image. When manual mode is enabled, the output image size will be given by register; otherwise, it will be determined by input.

table 5-9 MEC/MGC control registers

address	register name	default value	R/W	description
0x5600	SCALE CTRL00	0x00	RW	Bit[3]: scale_manual_en For manual mode enable Bit[2]: scale_gain_2x_en For 2x gain enable Bit[1:0]: scale_mode For scale mode, select 00: 0.6x 01: 0.5x 10: 0.4x 11: 1/3x
0x5602	SCALE WIDTH	0x01	RW	Bit[4:0]: output_width[12:8] for manual mode
0x5603	SCALE WIDTH	0x80	RW	Bit[7:0]: output_width[7:0] for manual mode

6 register tables

The following tables provide descriptions of the device control registers contained in the OV8858. For all register enable/disable bits, ENABLE = 1 and DISABLE = 0. The device slave addresses are 0x6C for write and 0x6D for read (when SID=1, 0x20 for write and 0x21 for read).

6.1 PLL control [0x0100, 0x0103, 0x0300 - 0x031E]

table 6-1 PLL control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x0100	SC_CTRL0100	0x00	RW	Bit[7:1]: Debug mode Bit[0]: software_standby 0: software_standby 1: Streaming
0x0103	SC_CTRL0103	–	W	Bit[7:1]: Debug mode Bit[0]: software_reset
0x0300	PLL_CTRL_0	0x00	RW	Bit[7:3]: Not used Bit[2:0]: pll1_pre_div 000: /1 001: /1.5 010: /2 011: /2.5 100: /3 101: /4 110: /6 111: /8
0x0301	PLL_CTRL_1	0x00	RW	Bit[7:2]: Not used Bit[1:0]: pll1_multiplier[9:8]
0x0302	PLL_CTRL_2	0x19	RW	Bit[7:0]: pll1_multiplier[7:0]
0x0303	PLL_CTRL_3	0x00	RW	Bit[7:4]: Not used Bit[3:0]: pll1_divm 1+pll1_divm
0x0304	PLL_CTRL_4	0x03	RW	Bit[7:2]: Not used Bit[1:0]: pll1_div_mipi 00: /4 01: /5 10: /6 11: /8
0x0305	PLL_CTRL_5	0x01	RW	Bit[7:2]: Not used Bit[1:0]: pll1_div_sp 00: /3 01: /4 10: /5 11: /6

table 6-1 PLL control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x0306	PLL_CTRL_6	0x01	RW	Bit[7:1]: Not used Bit[0]: pll1_div_s 0: /1 1: /2
0x0308	PLL_CTRL_8	0x00	RW	Bit[7:1]: Not used Bit[0]: pll1_bypass
0x0309	PLL_CTRL_9	0x01	RW	Bit[7:3]: Not used Bit[2:0]: pll1_cp
0x030A	PLL_CTRL_A	0x00	RW	Bit[7:4]: Not used Bit[3:1]: pll1_reserve Bit[0]: pll1_predivp 0: /1 1: /2
0x030B	PLL_CTRL_B	0x00	RW	Bit[7:3]: Not used Bit[2:0]: pll2_pre_div 000: /1 001: /1.5 010: /2 011: /2.5 100: /3 101: /4 110: /6 111: /8
0x030C	PLL_CTRL_C	0x00	RW	Bit[7:2]: Not used Bit[1:0]: pll2_r_divp[9:8]
0x030D	PLL_CTRL_D	0x1E	RW	Bit[7:0]: pll2_r_divp[7:0]
0x030E	PLL_CTRL_E	0x02	RW	Bit[7:3]: Not used Bit[2:0]: pll2_r_divs 000: /1 001: /1.5 010: /2 011: /2.5 100: /3 101: /3.5 110: /4 111: /5
0x030F	PLL_CTRL_F	0x02	RW	Bit[7:4]: Not used Bit[3:0]: pll2_r_divsp 1+pll2_r_divsp
0x0310	PLL_CTRL_10	0x01	RW	Bit[7:3]: Not used Bit[2:0]: pll2_r_cp
0x0311	PLL_CTRL_11	0x00	RW	Bit[7:1]: Not used Bit[0]: pll2_bypass

table 6-1 PLL control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x0312	PLL_CTRL_12	0x01	RW	Bit[7:5]: pll2_reserve Bit[4]: pll2_pre_div0 0: /1 1: /2 Bit[3:0]: pll2_r_divdac 1+pll2_r_divdac
0x031B	PLL_CTRL_1B	0x00	RW	Bit[7:1]: Not used Bit[0]: pll1_rst
0x031C	PLL_CTRL_1C	0x00	RW	Bit[7:1]: Not used Bit[0]: pll2_rst
0x031E	PLL_CTRL_1E	0x04	RW	Bit[7:4]: Not used Bit[3]: pll1_no_lat Bit[2]: Not used Bit[1:0]: mipi_bitsel_man

6.2 system control [0x3000 - 0x3043]

table 6-2 system control registers (sheet 1 of 6)

address	register name	default value	R/W	description
0x3000	PAD OEN0	0x00	RW	Bit[7:6]: Not used Bit[5]: io_fsin_oen Bit[4:0]: Not used
0x3002	PAD OEN2	0x20	RW	Bit[7]: io_vsync_oen Bit[6]: io_href_oen Bit[5]: Not used Bit[4]: io_frex_oen Bit[3:1]: Not used Bit[0]: io_gpio_oen
0x3003	GPIO IN	–	R	Bit[7:4]: Not used Bit[3]: io_fsin_i Bit[2]: io_href_i Bit[1]: io_vsync_i Bit[0]: io_gpio_i
0x3004	SCCB ID	0x6C	RW	Bit[7:0]: sccb_id SCCB programmed ID when SID = 0

table 6-2 system control registers (sheet 2 of 6)

address	register name	default value	R/W	description
0x3005	CLKRST5	0xF0	RW	Bit[7:6]: Not used Bit[5]: sclk_psram Bit[4]: sclk_syncfifo Bit[3:2]: Not used Bit[1]: rst_psram Bit[0]: rst_syncfifo
0x3006	SCCB ID2	0x42	RW	Bit[7:0]: sccb_id2 SCCB ID2, SCCB will ack ID2 and ID1
0x3007	R ISPOUT BITSEL	0x20	RW	Bit[7]: pll12_daclk_sel Bit[6]: r_pump_clk_sel 0: pll2_sclk 1: pll1_sclk Bit[5]: r_ispin_array_addr_sel Bit[4]: r_ilpwm_out_sel Bit[3]: r_rst_pll_sleep_dis Bit[2]: r_db_out_en Bit[1:0]: r_vsync_sel
0x300A	CHIP ID	0x00	R	Bit[7:0]: chip_id[23:16]
0x300B	CHIP ID	0x88	R	Bit[7:0]: chip_id[15:8]
0x300C	CHIP ID	0x65	R	Bit[7:0]: chip_id[7:0]
0x300D	PAD OUT2	0x00	RW	Bit[7]: io_vsync_o Bit[6]: io_href_o Bit[5]: Not used Bit[4]: io_frex_o Bit[3]: io_strobe_o Bit[2]: io_sda_o Bit[1]: io_ilpwm_o Bit[0]: io_gpio_o
0x3010	PAD SEL2	0x00	RW	Bit[7]: io_vsync_sel Bit[6]: io_href_sel Bit[5]: Not used Bit[4]: io_frex_sel Bit[3]: io_strobe_sel Bit[2]: io_sda_sel Bit[1]: io_ilpwm_sel Bit[0]: io_gpio_sel
0x3011	PAD	0x00	RW	Bit[7]: Not used Bit[6:5]: Pad drive strength 00: 1x 01: 2x 10: 3x 11: 4x
0x3012	SCCB R12	0x20	RW	Bit[7:0]: sccb_id when SID = 1

table 6-2 system control registers (sheet 3 of 6)

address	register name	default value	R/W	description
0x3015	PUMP CLK DIV	0x00	RW	Bit[7]: Not used Bit[6:4]: Npump clock div 000: /2 001: /4 010: /8 011: /16 100: /32 1xx: Disable pump_clk Bit[3]: Not used Bit[2:0]: Ppump clock div 000: /2 001: /4 010: /8 011: /16 100: /32 1xx: Disable pump_clk
0x3018	MIPI SC CTRL	0x72	RW	Bit[7:5]: mipi_lane_mode N+1 lane Bit[4]: mipi_lvds_sel 0: LVDS enable 1: MIPI enable Bit[3:2]: r_phy_pd_mipi_man Bit[1]: Reserved Bit[0]: lane_dis option 1: Disable lanes when pd_mipi
0x3019	MIPI SC CTRL	0x00	RW	Bit[7:0]: MIPI lane disable manual
0x301A	CLKRST0	0xF0	RW	Bit[7]: Reserved Bit[6]: sclk_stb Bit[5]: sclk_ac Bit[4]: sclk_tc Bit[3]: mipi_phy_rst_o Bit[2]: rst_stb Bit[1]: rst_ac Bit[0]: rst_tc
0x301B	CLKRST1	0xF0	RW	Bit[7]: sclk_blc Bit[6]: sclk_isp Bit[5]: sclk_testmode Bit[4]: sclk_vfifo Bit[3]: rst_blc Bit[2]: rst_isp Bit[1]: rst_testmode Bit[0]: rst_vfifo

table 6-2 system control registers (sheet 4 of 6)

address	register name	default value	R/W	description
0x301C	CLKRST2	0xF0	RW	Bit[7]: Not used Bit[6]: sclk_mipi Bit[5]: sclk_dpcm Bit[4]: sclk_otp Bit[3]: Not used Bit[2]: rst_mipi Bit[1]: rst_dpcm Bit[0]: rst_otp
0x301D	CLKRST3	0xF0	RW	Bit[7]: sclk_asram_tst Bit[6]: sclk_grp Bit[5]: sclk_bist Bit[4]: Reserved Bit[3]: rst_asram_tst Bit[2]: rst_grp Bit[1]: rst_bist Bit[0]: rst_aec_pk
0x301E	CLKRST4	0xF0	RW	Bit[7]: sclk_ilpwm Bit[6]: pclk_lvds Bit[5]: pclk_vfifo Bit[4]: pclk_mipi Bit[3]: rst_ilpwm Bit[2]: rst_lvds Bit[1:0]: Not used
0x301F	FREX RST MASK0	0x00	RW	Bit[7]: frex_mask_aec_pk Bit[6]: frex_mask_blc/sync FIFO/psram Bit[5]: frex_mask_isp Bit[4]: Not used Bit[3]: frex_mask_mipi Bit[2]: frex_mask_vfifo Bit[1]: frex_mask_testmode Bit[0]: frex_mask_mipi_phy
0x3020	CLOCK SEL	0x93	RW	Bit[7]: Clock switch output 0: Padclk 1: Normal Bit[6:4]: Not used Bit[3]: pclk_div 0: /1 1: /2 Bit[2:0]: Not used

table 6-2 system control registers (sheet 5 of 6)

address	register name	default value	R/W	description
0x3021	MISC CTRL	0x23	RW	Bit[7]: Not used Bit[6]: Sleep no latch option 1: No latch Bit[5]: fst_stby_ctr 0: Software standby enter at v_blk 1: Software standby enter at l_blk or v_blk Bit[4:1]: Not used Bit[0]: cen_global_o Global control all CEN of SRAM
0x3022	MIPI SC CTRL	0x01	RW	Bit[7:4]: Not used Bit[3]: lvds_mode_o Bit[2]: Not used Bit[1]: Clock lane disable when pd_mipi Bit[0]: pd_mipi enable when rst_sync
0x3023	MIPI LPTX SEL	0x00	RW	Bit[7:0]: Not used
0x3024	REG24	0x10	RW	Bit[7:3]: Not used Bit[2]: rst_ana manual Bit[1:0]: Not used
0x302A	SUB ID	0xB0	R	Bit[7:4]: Process Bit[3:0]: Version
0x3030	REG30	0x00	RW	Bit[7:6]: Not used Bit[5]: SCLK inv Bit[4]: PCLK inv Bit[3:0]: Not used
0x3031	REG31	0x0A	RW	Bit[7:5]: Not used Bit[4:0]: mipi_bit_sel 0x8: 8-bit mode 0xA: 10-bit mode Others: Not used
0x3032	REG32	0x80	RW	Bit[7]: pll2_sysclk_sel Bit[6]: asram_clk_sel Bit[5]: array_hskip_man_en Bit[4]: r_rst_otp_sleep_dis Bit[3]: r_rst_ana_sleep_dis Bit[2:0]: array_hskip_man[3:1]

table 6-2 system control registers (sheet 6 of 6)

address	register name	default value	R/W	description
0x3033	REG33	0x24	RW	Bit[7]: array_hskip_man[0] Bit[6]: Not used Bit[5]: r_fmt_eof_sel 0: isp_eof 1: mipi_eof Bit[4]: sync_point_sel Bit[3]: rip_sof_en Bit[2]: Not used Bit[1]: pll_sysclk_sel Bit[0]: lvds_ck_data_sel
0x3037	PLL1 CTR1	0x00	RW	Bit[7:1]: Not used Bit[0]: sid_rev
0x303E	ENCLK_SEL	0x03	RW	Bit[7:3]: Not used Bit[2:0]: Stream on delay 2^9/2^10.../2^16
0x303F	CTRL3F	0x00	RW	Bit[7:2]: Not used Bit[1]: sccb_id2_nack Nack to ID2 Bit[0]: sccb_pgm_id_en
0x3040	CTRL00	0xF0	RW	Bit[7]: sclk_isp_fc_en Bit[6]: sclk_fc_en Bit[5]: sclk_tpm_en Bit[4]: sclk_fmt_en Bit[3]: rst_isp_fc Bit[2]: rst_fc Bit[1]: rst_tpm Bit[0]: rst_fmt
0x3043	FREX RST MASK1	0xF0	RW	Bit[7]: frex_mask_isp_fc Bit[6]: frex_mask_fc Bit[5]: frex_mask_tpm Bit[4]: frex_mask_fmt Bit[3:0]: Not used

6.3 SCCB control [0x3100 - 0x3111, 0x314E-0x314F]

table 6-3 SCCB registers

address	register name	default value	R/W	description
0x3100	SB_SCCB_CTRL	0x00	RW	Bit[7:0]: Debug mode
0x3101	SCCB OPT	0x32	RW	Bit[7:6]: Not used Bit[5]: Re-mapping enable Bit[4]: en_ss_addr_inc Bit[3:0]: Debug mode
0x3102	SCCB FILTER	0x00	RW	Bit[7:0]: Debug mode
0x3103	SCCB SYSREG	0x00	RW	Bit[7:0]: Not used
0x3104	PWUP DIS	0x01	RW	Bit[7:0]: Not used
0x3105	SB_PADCLK_DIV	0x11	RW	Bit[7:6]: Debug mode Bit[5:0]: Chip debug
0x3106	SRB HOST INPUT DIS	0x01	RW	Bit[7:4]: sclk_div /1/1/2/3.../15 Bit[3:2]: sclk_pre_div 00: /1 01: /2 10: /4 11: /1 Bit[1:0]: Chip debug
0x3108	DESTINATION RE-MAPPING ADDR	0x90	RW	High Byte of Destination Address for Re-mapping Function
0x3109	DESTINATION RE-MAPPING ADDR	0x00	RW	Low Byte of Destination Address for Re-mapping Function
0x3110	SRC ADDR00 H	0x00	RW	High Byte of Number 00 Source Register Address
0x3111	SRC ADDR00 L	0x00	RW	Low Byte of Number 00 Source Register Address
0x314E	SRC ADDR1F H	0x00	RW	High Byte of Number 1F Source Register Address
0x314F	SRC ADDR1F L	0x00	RW	Low Byte of Number 1F Source Register Address

6.4 group hold [0x3200 - 0x320F]

table 6-4 group hold registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x3200	GROUP ADR0	0x00	RW	Group0 Start Address in SRAM, actual address is {0x3200[5:0], 4'h0}
0x3201	GROUP ADR1	0x10	RW	Group1 Start Address in SRAM, Actual Address is {0x3201[5:0], 4'h0}
0x3202	GROUP ADR2	0x20	RW	Group2 Start Address in SRAM, Actual Address is {0x3202[5:0], 4'h0}
0x3203	GROUP ADR3	0x30	RW	Group3 Start Address in SRAM, Actual Address is {0x3203[5:0], 4'h0}
0x3204	GROUP LEN0	–	R	Length of Group0
0x3205	GROUP LEN1	–	R	Length of Group1
0x3206	GROUP LEN2	–	R	Length of Group2
0x3207	GROUP LEN3	–	R	Length of Group3
0x3208	GROUP ACCESS	–	W	Bit[7:4]: group_ctrl 0000: Group hold start 0001: Group hold end 1010: Group launch 1110: Fast group launch Others: Reserved Bit[3:0]: Group ID 0000: Group bank 0 0001: Group bank 1 0010: Group bank 2 0011: Group bank 3 Others: Reserved
0x3209	GROUP0 PERIOD	0x00	RW	Bit[7]: Not used Bit[6:5]: Switch back group Bit[4:0]: Number of frames to stay in group 0
0x320A	GROUP1 PERIOD	0x00	RW	Number of Frames to Stay in Group 1
0x320B	GRP_SW_CTRL	0x11	RW	Bit[7]: auto_sw Bit[6:5]: Not used Bit[4]: frame_cnt_trig Bit[3]: group_switch_repeat Bit[2]: context_en Bit[1:0]: Second group select
0x320C	SRAM TEST	0x0A	RW	Bit[7:5]: Not used Bit[4]: Group hold SRAM test enable Bit[3:0]: Group hold SRAM RM[3:0]

table 6-4 group hold registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x320D	GRP_ACT	–	R	Active Group Indicator
0x320E	FM_CNT_GRP0	–	R	Group 0 Frame Count
0x320F	FM_CNT_GRP1	–	R	Group 1 Frame Count

6.5 MEC/MGC control [0x3500 - 0x3512]

table 6-5 MEC/MGC control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x3500	LONG EXPO	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Long exposure[19:16]
0x3501	LONG EXPO	0x02	RW	Bit[7:0]: Long exposure[15:8]
0x3502	LONG EXPO	0x00	RW	Bit[7:0]: Long exposure[7:0] Low 4 bits are fraction bits
0x3503	AEC MANUAL	0x00	RW	Bit[7]: Not used Bit[6]: Digital fraction gain delay option 0: Delay 1 frame 1: Not delay 1 frame Bit[5]: Gain change delay option 0: Delay 1 frame 1: Not delay 1 frame Bit[4]: Gain delay option 0: Delay 1 frame 1: Not delay 1 frame Bit[3]: Not used Bit[2]: Gain manual as sensor gain 0: Input gain as real gain format 1: Input gain as sensor gain format Bit[1]: Exposure delay option (must be 0) 0: Delay 1 frame 1: Not used Bit[0]: Exposure change delay option (must be 0) 0: Delay 1 frame 1: Not used

table 6-5 MEC/MGC control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x3505	GCVT OPTION	0x80	RW	Gain Conversation Option Bit[7]: DAC fixed gain bit Bit[6]: Not used Bit[5:4]: Sensor gain fixed bit Bit[3:2]: Sensor gain pregain option (debug only, always set it to 0) Bit[1:0]: Sensor gain option for transferring real gain to sensor gain format
0x3507	GAIN SHIFT	0x00	RW	Bit[7:2]: Not used Bit[1:0]: Gain shift option 00: No shift 01: Left shift 1 bit 10: Left shift 2 bit 11: Left shift 3 bit
0x3508	LONG GAIN	0x00	RW	Bit[7:5]: Not used Bit[4:0]: Long gain[12:8]
0x3509	LONG GAIN	0x80	RW	Bit[7:0]: Long gain[7:0] 0x3503[2]=0, gain[7:0] is real gain format, where low 4 bits are fraction bits, for example, 0x10 is 1x gain, 0x28 is 2.5x gain If 0x3503[2]=1, gain[7:0] is sensor gain format, gain[7:4] is coarse gain, 00000: 1x, 00001: 2x, 00011: 4x, 00111: 8x, gain[7] is 1, gain[3:0] is fine gain. For example, 0x10 is 1x gain, 0x30 is 2x gain, 0x70 is 4x gain
0x350A	LONG DIGIGAIN	0x08	RW	Bit[7:0]: Long digital gain[13:6]
0x350B	LONG DIGIGAIN	0x00	RW	Bit[7:6]: Not used Bit[5:0]: Long digital gain[5:0] Low 10 bits are fraction bits
0x350C	SHORT GAIN	0x00	RW	Bit[7:5]: Not used Bit[4:0]: Short gain[12:8]
0x350D	SHORT GAIN	0x80	RW	Bit[7:0]: Short gain[7:0]
0x350E	SHORT DIGIGAIN	0x08	RW	Bit[7:0]: Short digital gain[13:6]
0x350F	SHORT DIGIGAIN	0x00	RW	Bit[7:6]: Not used Bit[5:0]: Short digital gain[5:0] Low 10 bits are fraction bits
0x3510	SHORT EXPO	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Short exposure[19:16]
0x3511	SHORT EXPO	0x02	RW	Bit[7:0]: Short exposure[15:8]

table 6-5 MEC/MGC control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x3512	SHORT EXPO	0x00	RW	Bit[7:0]: Short exposure[7:0] Low 4 bits are fraction bits

6.6 FREX control [0x37C5 - 0x37DF]

table 6-6 FREX control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x37C5	FREX EXP	0x00	RW	Bit[7:0]: frex_exp[23:16]
0x37C6	FREX EXP	0x00	RW	Bit[7:0]: frex_exp[15:8]
0x37C7	FREX EXP	0x05	RW	Bit[7:0]: frex_exp[7:0]
0x37C9	STROBE WIDTH	0x00	RW	Bit[7:4]: Not used Bit[3:0]: strobe_width[19:16]
0x37CA	STROBE WIDTH	0x06	RW	Bit[7:0]: strobe_width[15:8]
0x37CB	STROBE WIDTH	0x00	RW	Bit[7:0]: strobe_width[7:0]
0x37CC	SHUTTER DLY	0x00	RW	Bit[7:5]: Not used Bit[4:0]: shutter_dly[12:8]
0x37CD	SHUTTER DLY	0x44	RW	Bit[7:0]: shutter_dly[7:0]
0x37CE	FREX PCHG WIDTH	0x1F	RW	Bit[7:0]: frex_pchg_width[15:8]
0x37CF	FREX PCHG WIDTH	0x40	RW	Bit[7:0]: frex_pchg_width[7:0]
0x37D0	DATOUT DLY	0x00	RW	Bit[7:0]: datout_dly[15:8]
0x37D1	DATOUT DLY	0x01	RW	Bit[7:0]: datout_dly[7:0]
0x37D2	STROBE DLY	0x00	RW	Bit[7:5]: Not used Bit[4:0]: strobe_dly[12:8]
0x37D3	STROBE DLY	0x00	RW	Bit[7:0]: strobe_dly[7:0]
0x37D4	BST PCHG EN WIDTH	0x32	RW	Bit[7:0]: bst_pchg_en_width[7:0] bst_pchg_en width / 8
0x37D5	BST PCHG EN GAP	0x32	RW	Bit[7:0]: bst_pchg_en_gap[7:0] bst_pchg_en gap / 8

table 6-6 FREX control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x37D6	REG14	0x00	RW	Bit[7:4]: Not used Bit[3]: bst_prechg_en_sel_eco1234 Bit[2]: bst_prechg_en_all0 Bit[1]: bst_prechg_en_pol Bit[0]: frex_end option 0: Use frame_exp 1: Use falling edge of FREX input
0x37DE	R1E	0x01	RW	Bit[7:1]: Not used Bit[0]: frex_sccb_req_repeat_trig_sel 0: SOF 1: EOF
0x37DF	FREX REQ	0x04	RW	Bit[7]: frex_sccb_req (self clearing) Bit[6]: frex_sccb_req_repeat (debug) Bit[5]: frex_strobe_out_sel Bit[4]: frex_nopchg Bit[3]: frex_strobe polarity Bit[2]: frex_shutter polarity Bit[1]: frex_i from pad in Bit[0]: no_latch at SOF for frex_sccb_req

6.7 timing control [0x3800 - 0x382B]

table 6-7 timing control registers (sheet 1 of 4)

address	register name	default value	R/W	description
0x3800	X ADDR START	0x00	RW	Bit[7:4]: Not used Bit[3:0]: x_addr_start[11:8] Array horizontal start point high byte
0x3801	X ADDR START	0x0C	RW	Bit[7:0]: x_addr_start[7:0] Array horizontal start point low byte
0x3802	Y ADDR START	0x00	RW	Bit[7:4]: Not used Bit[3:0]: y_addr_start[11:8] Array vertical start point high byte
0x3803	Y ADDR START	0x0C	RW	Bit[7:0]: y_addr_start[7:0] Array vertical start point low byte
0x3804	X ADDR END	0x0C	RW	Bit[7:4]: Not used Bit[3:0]: x_addr_end[11:8] Array horizontal end point high byte

table 6-7 timing control registers (sheet 2 of 4)

address	register name	default value	R/W	description
0x3805	X ADDR END	0xD3	RW	Bit[7:0]: x_addr_end[7:0] Array horizontal end point low byte
0x3806	Y ADDR END	0x09	RW	Bit[7:4]: Not used Bit[3:0]: y_addr_end[11:8] Array vertical end point high byte
0x3807	Y ADDR END	0xA3	RW	Bit[7:0]: y_addr_end[7:0] Array vertical end point low byte
0x3808	X OUTPUT SIZE	0x0C	RW	Bit[7:4]: Not used Bit[3:0]: x_output_size[11:8] ISP horizontal output width high byte
0x3809	X OUTPUT SIZE	0xC0	RW	Bit[7:0]: x_output_size[7:0] ISP horizontal output width low byte
0x380A	Y OUTPUT SIZE	0x09	RW	Bit[7:4]: Not used Bit[3:0]: y_output_size[11:8] ISP vertical output height high byte
0x380B	Y OUTPUT SIZE	0x90	RW	Bit[7:0]: y_output_size[7:0] ISP vertical output height low byte
0x380C	HTS	0x07	RW	Bit[7:4]: Not used Bit[3:0]: HTS[15:8] Total horizontal timing size high byte
0x380D	HTS	0x4C	RW	Bit[7:0]: HTS[7:0] Total horizontal timing size low byte
0x380E	VTS	0x0A	RW	Bit[7:0]: VTS[15:8] Total vertical timing size high byte
0x380F	VTS	0x74	RW	Bit[7:0]: VTS[7:0] Total vertical timing size low byte
0x3810	ISP X WIN	0x00	RW	Bit[7:0]: isp_x_win[15:8] ISP horizontal windowing offset high byte
0x3811	ISP X WIN	0x04	RW	Bit[7:0]: isp_x_win[7:0] ISP horizontal windowing offset low byte
0x3812	ISP Y WIN	0x00	RW	Bit[7:4]: Not used Bit[3:0]: isp_y_win[11:8] ISP vertical windowing offset high byte
0x3813	ISP Y WIN	0x02	RW	Bit[7:0]: isp_y_win[7:0] ISP vertical windowing offset low byte
0x3814	X INC ODD	0x01	RW	Bit[7:5]: Not used Bit[4:0]: x_odd_inc

table 6-7 timing control registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x3815	X INC EVEN	0x01	RW	Bit[7:5]: Not used Bit[4:0]: x_even_inc
0x3816	VSYNC START	0x00	RW	Bit[7:0]: vsync_start[15:8] VSYNC start point high byte
0x3817	VSYNC START	0x00	RW	Bit[7:0]: vsync_start[7:0] VSYNC start point low byte
0x3818	VSYNC END	0x00	RW	Bit[7:0]: vsync_end[15:8] VSYNC end point high byte
0x3819	VSYNC END	0x00	RW	Bit[7:0]: vsync_end[7:0] VSYNC end point low byte
0x381A	HSYNC FIRST H	0x04	RW	Bit[7:0]: hsync_first[15:8] HSYNC first active row start position high byte
0x381B	HSYNC FIRST L	0x00	RW	Bit[7:0]: hsync_first[7:0] HSYNC first active row start position low byte
0x3820	FORMAT1	0x00	RW	Format1 Bit[7]: vsub48_blc Bit[6]: vflip_blc Bit[5:4]: Not used Bit[3]: byp_isp_o Bit[2]: vflip_dig Bit[1]: vflip_arr Bit[0]: hdr_en
0x3821	FORMAT2	0x40	RW	Format2 Bit[7]: dig_hbin4 Bit[6]: hsync_en_o Bit[5]: fst_vbin Bit[4]: fst_hbin Bit[3]: isp_hvar2 Bit[2]: mirror_dig Bit[1]: mirror_arr Bit[0]: dig_hbin2
0x3822	REG22	0x88	RW	Bit[7:5]: addr0_num[3:1] Bit[4:0]: ablc_num[5:1]
0x3823	REG23	0x08	RW	Bit[7]: ext_vs_re Bit[6]: ext_vs_en Bit[5]: vts_no_latch Bit[4]: init_man Bit[3:0]: r_grp_adj
0x3824	CS RST FSIN	0x00	RW	Bit[7:0]: cs_rst_fsin[15:8] CS reset value high byte at vs_ext

table 6-7 timing control registers (sheet 4 of 4)

address	register name	default value	R/W	description
0x3825	CS RST FSIN	0x20	RW	Bit[7:0]: cs_rst_fsin[7:0] CS reset value low byte at vs_ext
0x3826	R RST FSIN	0x00	RW	Bit[7:0]: r_rst_fsin[15:8] R reset value high byte at vs_ext
0x3827	R RST FSIN	0x04	RW	Bit[7:0]: r_rst_fsin[7:0] R reset value low byte at vs_ext
0x3828	REG28	0x00	RW	Bit[7]: ext_hs_re Bit[6]: ext_hs_en Bit[5]: asp_start_sel 0: Use sync output 1: Use sensor output Bit[4]: hts_inc_en Bit[3]: r_gate_vs_b Bit[2]: VSYNC polarity Bit[1:0]: href_w
0x382A	Y INC ODD	0x01	RW	Bit[7:5]: Not used Bit[4:0]: y_odd_inc
0x382B	Y INC EVEN	0x01	RW	Bit[7:5]: Not used Bit[4:0]: y_even_inc

6.8 VIV control registers [0x3A00 - 0x3A24, 0x3A40 - 0x3A43]

table 6-8 VIV control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x3A00	H_OFFSET	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Small image horizontal offset from large image[11:8]
0x3A01	H_OFFSET	0x00	RW	Bit[7:0]: Small image horizontal offset from large image[7:0]
0x3A02	V_OFFSET	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Small image vertical offset from large image[11:8]
0x3A03	V_OFFSET	0x00	RW	Bit[7:0]: Small image vertical offset from large image[7:0]
0x3A04	EXT_WIDTH	0x02	RW	Bit[7:4]: Not used Bit[3:0]: External image width[11:8]

table 6-8 VIV control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x3A05	EXT_WIDTH	0x80	RW	Bit[7:0]: External image width[7:0]
0x3A06	EXT_HEIGHT	0x01	RW	Bit[7:4]: Not used Bit[3:0]: External image height[11:8]
0x3A07	EXT_HEIGHT	0xE0	RW	Bit[7:0]: External image height[7:0]
0x3A08	MASK_H_OFFSET	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Mask horizontal offset[11:8]
0x3A09	MASK_H_OFFSET	0x00	RW	Bit[7:0]: Mask horizontal offset7:0]
0x3A0A	MASK_V_OFFSET	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Mask vertical offset[11:8]
0x3A0B	MASK_V_OFFSET	0x00	RW	Bit[7:0]: Mask vertical offset[7:0]
0x3A0C	DATA_RATIO	0x00	RW	Bit[7:0]: Ratio of large image to small image[15:8]
0x3A0D	DATA_RATIO	0x00	RW	Bit[7:0]: Ratio of large image to small image[7:0]
0x3A0E	CTRL_REG	0x04	RW	Bit[7:5]: Not used Bit[4]: program_sram_en Bit[3]: Not used Bit[2]: always_load_header_en Bit[1]: manual_mask_size Bit[0]: manual_scale_en
0x3A0F	SCALE_RAIO	0x00	RW	Bit[7:0]: Mask scale ratio[15:8]
0x3A10	SCALE_RAIO	0x00	RW	Bit[7:0]: Mask scale ratio[7:0]
0x3A11	VIV_CTRL	0x00	RW	Bit[7]: VIV_block_en Bit[6]: swap_en Bit[5]: side_by_side_en Bit[4:0]: Not used
0x3A12	VIV_CTRL	0x00	RW	Bit[7:3]: Not used Bit[2]: border_en Bit[1]: shape_en Bit[0]: VIV_en
0x3A14	SRAM_TRIG	0x00	RW	Bit[7:1]: Not used Bit[0]: SRAM read/write address reset
0x3A16	DATA_OFFSET0	0x00	RW	Bit[7:0]: Data_offset0[7:0]
0x3A17	DATA_OFFSET1	0x00	RW	Bit[7:0]: Data_offset1[7:0]
0x3A18	DATA_OFFSET2	0x00	RW	Bit[7:0]: Data_offset2[7:0]
0x3A19	DATA_OFFSET3	0x00	RW	Bit[7:0]: Data_offset3[7:0]

table 6-8 VIV control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x3A1A	DATA_OFFSET	0x00	RW	Bit[7:6]: Data_offset0[9:8] Bit[5:4]: Data_offset1[9:8] Bit[3:2]: Data_offset2[9:8] Bit[1:0]: Data_offset3[9:8]
0x3A1B	DATA_BORDER0	0x00	RW	Bit[7:0]: Data_border0[7:0]
0x3A1C	DATA_BORDER1	0x00	RW	Bit[7:0]: Data_border1[7:0]
0x3A1D	DATA_BORDER2	0x00	RW	Bit[7:0]: Data_border2[7:0]
0x3A1E	DATA_BORDER3	0x00	RW	Bit[7:0]: Data_border3[7:0]
0x3A1F	DATA_BORDER	0x00	RW	Bit[7:6]: Data_border0[9:8] Bit[5:4]: Data_border1[9:8] Bit[3:2]: Data_border2[9:8] Bit[1:0]: Data_border3[9:8]
0x3A20	BORDER_WIDTH	0x00	RW	Border_width
0x3A21	MASK_WIDTH	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Manual_mask_width[11:8]
0x3A22	MASK_WIDTH	0x00	RW	Bit[7:0]: Manual_mask_width[7:0]
0x3A23	MASK_HEIGHT	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Manual_mask_height[11:8]
0x3A24	MASK_HEIGHT	0x00	RW	Bit[7:0]: Manual_mask_height[7:0]
0x3A40	VIV_BUF_CTRL	0x01	RW	Bit[7:2]: Not used Bit[1]: HREF clear enable Bit[0]: SOF clear enable
0x3A42	VIV_RD_LINE_OUT	–	R	High Byte Start Line Counter at VIV Read
0x3A43	VIV_RD_LINE_OUT	–	R	Low Byte Start Line Counter at VIV Read

6.9 strobe [0x3B00 - 0x3B05]

table 6-9 strobe control registers

address	register name	default value	R/W	description
0x3B00	RSTRB	0x00	RW	Bit[7]: Strobe ON/OFF Bit[6]: Strobe polarity 0: Active high 1: Active low Bit[5:4]: width_in_xenon Bit[3]: Not used Bit[2:0]: Strobe mode 000: Xenon 001: LED1 010: LED2 011: LED3 100: LED4
0x3B02	STROBE ADD DUMMY	0x00	RW	Bit[7:0]: strobe_add_dummy[15:8] Dummy line number added at strobe high byte
0x3B03	STROBE ADD DUMMY	0x00	RW	Bit[7:0]: strobe_add_dummy[7:0] Dummy line number added at strobe low byte
0x3B04	STROBE CTL1	0x00	RW	Bit[7]: strobe_valid (read only) Bit[6:4]: Not used Bit[3]: start_point_sel Bit[2]: Strobe repeat enable Bit[1:0]: Strobe lantency 00: Strobe generated at next frame 01: Delay one frame, strobe generated 2 frame later 10: Delay one frame, strobe generated 3 frame later 11: Delay one frame, strobe generated 4 frame later
0x3B05	STROBE WIDTH	0x00	RW	Bit[7:2]: Strobe pulse width step Bit[1:0]: Strobe pulse width gain Strobe pulse width = 128×(2^gain)×(step+1)×sclk_period

6.10 OTP control registers [0x3D80 - 0x3D91]

table 6-10 OTP control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x3D80	OTP_PROGRAM_ CTRL	–	RW	Bit[7]: OTP_wr_busy (read only) Bit[6:1]: Not used Bit[0]: OTP_program_enable (write only)
0x3D81	OTP_LOAD_CTRL	–	RW	Bit[7]: OTP_rd_busy (read only) Bit[6]: Not used Bit[5]: OTP_bist_error (read only) Bit[4]: OTP_bist_done (read only) Bit[3:1]: Not used Bit[0]: OTP_load_enable (write only)
0x3D82	OTP_PGM_PULSE	0x55	RW	Program Strobe Pulse Width Unit: 8×system clock period
0x3D83	OTP_LOAD_PULSE	0x08	RW	Load Strobe Pulse Width Unit: system clock period
0x3D84	OTP_MODE_CTRL	0x00	RW	Bit[7]: Program disable 0: Not used 1: Disable Bit[6]: Mode select 0: Auto mode 1: Manual mode Bit[5:0]: Debug mode
0x3D85	OTP_REG85	0x13	RW	Bit[7:3]: Debug mode Bit[2]: OTP power up load data enable Bit[1]: OTP power up load setting enable Bit[0]: OTP write register load setting enable
0x3D86	SRAM_TEST_ SIGNALS	0x02	RW	Bit[7:4]: Debug mode Bit[3]: rst_otp_manual Bit[2]: r_test Bit[1:0]: r_rm
0x3D87	OTP_PS2CS	0x0A	RW	OTP PS to CSB Delay Unit: system clock period
0x3D88	OTP_START_ ADDRESS	0x00	RW	OTP Start High Address for Manual Mode
0x3D89	OTP_START_ ADDRESS	0x00	RW	OTP Start Low Address for Manual Mode
0x3D8A	OTP_EN_ADDRESS	0x00	RW	OTP End High Address for Manual Mode
0x3D8B	OTP_END_ADDRESS	0x00	RW	OTP End Low Address for Manual Mode

table 6-10 OTP control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x3D8C	OTP_SETTING_STT_ ADDRESS	0x00	RW	OTP Start High Address for Load Setting
0x3D8D	OTP_SETTING_STT_ ADDRESS	0x00	RW	OTP Start Low Address for Load Setting
0x3D8E	OTP_BIST_ERR_ ADDRESS	–	R	OTP Check Error Address High
0x3D8F	OTP_BIT_ERR_ ADDRESS	–	R	OTP Check Error Address Low
0x3D90	OTP_STROBE_GAP_ PGM	0x12	RW	Gap Between Strobe Pulse When Programming
0x3D91	OTP_STROBE_GAP_ LOAD	0x06	RW	Gap Between Strobe Pulse When Loading

6.11 PSRAM control [0x3F00 - 0x3F0F]

table 6-11 PSRAM control registers

address	register name	default value	R/W	description
0x3F00~ 0x3F0F	PSRAM_CTRL	–	–	PSRAM Control Register

6.12 BLC control [0x4000 - 0x402F, 0x403C-0x406F]

table 6-12 BLC control registers (sheet 1 of 7)

address	register name	default value	R/W	description
0x4000	BLC CTRL00	0xF1	RW	Bit[7]: out_range_trig_en Offset out of range trigger function enable signal Bit[6]: format_chg_en Format change trigger function enable signal Bit[5]: gain_chg_en Gain change trigger function enable signal Bit[4]: exp_chg_en Exposure change trigger function enable signal Bit[3]: manual_trig Manual trigger signal Its rising edge will trigger BLC Bit[2]: freeze_en BLC freeze function enable signal When it is set, the BLC will be frozen. Offsets will keep their previous frame values. Bit[1]: always_do BLC always trigger signal When it is set, the BLC will be triggered every frame unless the freeze_en is enabled. Bit[0]: median_en 5-point median filter function enable signal

table 6-12 BLC control registers (sheet 2 of 7)

address	register name	default value	R/W	description
0x4001	BLC CTRL01	0x86	RW	Bit[7]: Reserved Bit[6]: Blkcol zeroline diff enable Enable difference between black line column with zero line to cancel horizontal noise Bit[5:4]: Column shift option 00: Total 256 black columns 01: Total 128 black columns 10: Total 64 black columns 11: Total 32 black columns Bit[3]: RGBC pattern enable Bit[2]: BLC column cancel function enable Bit[1]: BLC cut range function enable Bit[0]: Remove row offset enable Column delta offset remove function enable signal 0: Used offset includes no column delta offset 1: Used offset includes column delta offset
0x4002	BLC CTRL02	0x04	RW	Bit[7:0]: Value used to limit BLC offset (Append 4'b1111 to get the real limitation value)
0x4003	BLC CTRL03	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Black line number used to calculate the BLC offsets
0x4004	BLC CTRL04	0x00	RW	Bit[7:0]: BLC target[15:8]
0x4005	BLC CTRL05	0x10	RW	Bit[7:0]: BLC target[7:0]
0x4006	BLC CTRL06	0x1F	RW	Bit[7:0]: Format change frame number
0x4007	BLC CTRL07	0x1F	RW	Bit[7:0]: Reset trigger frame number
0x4008	BLC CTRL08	0x01	RW	Bit[7:0]: Manual trigger frame number

table 6-12 BLC control registers (sheet 3 of 7)

address	register name	default value	R/W	description
0x4009	BLC CTRL09	0x01	RW	Bit[7]: Final BLC offset limitation function enable Bit[6]: BLC offset limitation function enable Bit[5:4]: BLC bypass output mode selection 00: Limit or clip to 10 bits 01: Low 10 bits (no bit shift) 10: Right shift 1 bit 11: Right shift 2 bits Bit[3]: Bypass cut range function enable Bit[2:1]: Reserved Bit[0]: Black column off flag option for median filter
0x400A	RSVD	–	–	Reserved
0x400B	BLC CTRL0B	0x0C	RW	Bit[7:4]: Start line for BLC initial function Bit[3]: Dark current BLC function enable Bit[2]: Dark current BLC manual mode enable Bit[1:0]: BLC last line select 00: Last bottom black line 01: Last bottom zero line 10: Last top black line 11: Last top zero line
0x400C	BLC CTRL0C	0x00	RW	Bit[7:0]: Offset trigger threshold[15:8] When abs(line_current_offset – blc_line_offset) bigger than offset_trig_thresh, the BLC update will be set
0x400D	BLC CTRL0D	0x20	RW	Bit[7:0]: Offset trigger threshold[7:0]
0x400E	BLC CTRL0E	0x00	RW	Bit[7:0]: BLC bypass offset[15:8]
0x400F	BLC CTRL0F	0x00	RW	Bit[7:0]: BLC bypass offset[7:0]
0x4010	BLC CTRL10	0xFF	RW	Bit[7:0]: max_offset It defines top limitation for offsets. The really used max_offset is {max_offset, {(IM_DW+2-8){1'b1}}}, where IM_DW=10

table 6-12 BLC control registers (sheet 4 of 7)

address	register name	default value	R/W	description
0x4011	BLC CTRL11	0x00	RW	Bit[7:6]: Dithering offset Bit[5]: offset_man_same When it is enabled, manual offsets will be the same and are all defined by manual_offset00. Bit[4]: Offset manual mode enable When it is enabled, used offsets will be defined manually with registers manual_offset00 ~ manual_offset11 Bit[3]: Data dithering function enable Bit[2]: out_range trigger option 0: Trigger all channel 1: Trigger the signal channel Bit[1]: Black column output enable Bit[0]: Black line output enable
0x4012	BLC CTRL12	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: manual_offset00[11:8] Manual offset for normal even-line and even-column pixels for long exposure
0x4013	BLC CTRL13	0x00	RW	Bit[7:0]: manual_offset000[7:0]
0x4014	BLC CTRL14	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: manual_offset001[11:8] Manual offset for normal even-line and odd-column pixels for long exposure
0x4015	BLC CTRL15	0x00	RW	Bit[7:0]: manual_offset001[7:0]
0x4016	BLC CTRL16	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: manual_offset10[11:8] Manual offset for normal odd-line and even-column pixels for long exposure
0x4017	BLC CTRL17	0x00	RW	Bit[7:0]: manual_offset010[7:0]
0x4018	BLC CTRL18	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: manual_offset011[11:8] Manual offset for normal odd-line and odd-column pixels for long exposure
0x4019	BLC CTRL19	0x00	RW	Bit[7:0]: manual_offset011[7:0]
0x401A	BLC CTRL1A	0x00	RW	Bit[7:0]: Zero line R coefficient[15:8]

table 6-12 BLC control registers (sheet 5 of 7)

address	register name	default value	R/W	description
0x401B	BLC CTRL1B	0x10	RW	Bit[7:0]: Zero line R coefficient[7:0]
0x401C	BLC CTRL1C	0x00	RW	Bit[7:0]: Zero line T coefficient[15:8]
0x401D	BLC CTRL1D	0x40	RW	Bit[7:0]: Zero line T coefficient[7:0]
0x401E	BLC CTRL1E	0x20	RW	Bit[7:0]: Cut range down threshold
0x401F	BLC CTRL1F	0x06	RW	Bit[7:4]: Initial line number Bit[3]: Rblue BLC reverse Bit[2]: Interpolation x enable Bit[1]: Interpolation y enable Bit[0]: Anchor one enable
0x4020	ANCHOR LEFT START	0x02	RW	Bit[7:4]: Reserved Bit[3:0]: Anchor left start[11:8]
0x4021	ANCHOR LEFT START	0x40	RW	Bit[7:0]: Anchor left start[7:0]
0x4022	ANCHOR LEFT END	0x03	RW	Bit[7:4]: Reserved Bit[3:0]: Anchor left end[11:8]
0x4023	ANCHOR LEFT END	0x3F	RW	Bit[7:0]: Anchor left end[7:0]
0x4024	ANCHOR RIGHT START	0x07	RW	Bit[7:4]: Reserved Bit[3:0]: Anchor right start[11:8]
0x4025	ANCHOR RIGHT START	0xC0	RW	Bit[7:0]: Anchor right start[7:0]
0x4026	ANCHOR RIGHT END	0x08	RW	Bit[7:4]: Reserved Bit[3:0]: Anchor right end[11:8]
0x4027	ANCHOR RIGHT END	0xBF	RW	Bit[7:0]: Anchor right end[7:0]
0x4028	TOP ZLINE ST	0x00	RW	Bit[7:6]: Not used Bit[5:0]: Top zero line start
0x4029	TOP ZLINE NUM	0x02	RW	Bit[7:5]: Not used Bit[4:0]: Top zero line number
0x402A	TOP BLKLINE ST	0x06	RW	Bit[7:6]: Not used Bit[5:0]: Top black line start
0x402B	TOP BLKLINE NUM	0x04	RW	Bit[7:5]: Not used Bit[4:0]: Top black line number
0x402C	BOT ZLINE ST	0x02	RW	Bit[7:6]: Not used Bit[5:0]: Bottom zero line start
0x402D	BOT ZLINE NUM	0x02	RW	Bit[7:5]: Not used Bit[4:0]: Bottom zero line number
0x402E	BOT BLKLINE ST	0x0E	RW	Bit[7:6]: Not used Bit[5:0]: Bottom black line start

table 6-12 BLC control registers (sheet 6 of 7)

address	register name	default value	R/W	description
0x402F	BOT BLKLINE NUM	0x04	RW	Bit[7:5]: Not used Bit[4:0]: Bottom black line number
0x403C	BLC CTRL3C	0x00	RW	Bit[7:0]: Cut range data width
0x403D	BLC CTRL3D	0x00	RW	Bit[7]: lsb_sub Bit[6]: round_precision_en_1_sign Bit[5]: round_precision_en_0_sign Bit[4]: load_seed Bit[3]: vfpn_precision_en Bit[2]: rnd_precision_en Bit[1]: round_precision_en_1 Bit[0]: Reserved
0x403E	BLC CTRL3E	0x00	RW	Bit[7:6]: Not used Bit[5:0]: gain_th1[5:0]
0x403F	BLC CTRL3F	0x00	RW	Bit[7]: Not used Bit[6:0]: round_value[6:0]
0x4040	BLC CTRL40	0x00	RW	Bit[7:6]: Not used Bit[5:0]: gain_th2[5:0]
0x4041	BLC CTRL41	0x07	RW	Bit[7:3]: Reserved Bit[2]: gain_comp_man_inty Bit[1]: gain_comp_man_intx Bit[0]: gain_comp_man_dither
0x4042	BLC CTRL42	0x01	RW	Bit[7:3]: Not used Bit[2:0]: col_add[10:8]
0x4043	BLC CTRL43	0x01	RW	Bit[7:0]: col_add[7:0]
0x4044~ 0x404F	RSVD	–	–	Reserved
0x4050	BLC CTRL50	–	R	Bit[7:0]: bline_offset_c000[15:8]
0x4051	BLC CTRL51	–	R	Bit[7:0]: bline_offset_c000[7:0]
0x4052	BLC CTRL52	–	R	Bit[7:0]: bline_offset_c001[15:8]
0x4053	BLC CTRL53	–	R	Bit[7:0]: bline_offset_c001[7:0]
0x4054	BLC CTRL54	–	R	Bit[7:0]: bline_offset_c010[15:8]
0x4055	BLC CTRL55	–	R	Bit[7:0]: bline_offset_c010[7:0]
0x4056	BLC CTRL56	–	R	Bit[7:0]: bline_offset_c011[15:8]
0x4057	BLC CTRL57	–	R	Bit[7:0]: bline_offset_c011[7:0]
0x4058~ 0x405F	BLINE_OFFSET_D000 ~ BLINE_OFFSET_D011	–	R	bline_offset_d000 ~ bline_offset_d011

table 6-12 BLC control registers (sheet 7 of 7)

address	register name	default value	R/W	description
0x4060~ 0x4067	BLINE_OFFSET_E000 ~ BLINE_OFFSET_E011	–	R	bline_offset_e000 ~ bline_offset_e011
0x4068~ 0x406F	BLINE_OFFSET_F000 ~ BLINE_OFFSET_F011	–	R	bline_offset_f000 ~ bline_offset_f011

6.13 frame control [0x4200 - 0x4203]

table 6-13 frame control registers

address	register name	default value	R/W	description
0x4200	R0	0x08	RW	Bit[7:4]: Not used Bit[3]: sof_after_line0 Bit[2]: fcnt_eof_sel Bit[1]: fcnt_mask_dis Bit[0]: fcnt_reset
0x4201	R1	0x00	RW	Bit[7:4]: Not used Bit[3:0]: frame_on_number
0x4202	R2	0x00	RW	Bit[7:4]: Not used Bit[3:0]: frame_off_number 0x4201, 0x4202= x, 0: on x frames and off 0x4201, 0x4202= 0, x: off x frames and on 0x4201, 0x4202= x, y: on x frames and off y frames
0x4203	R3	0x80	RW	Bit[7]: zero_line_mask_dis Bit[6]: rblue_mask_dis Bit[5]: data_mask_dis Bit[4]: valid_mask_dis Bit[3]: href_mask_dis Bit[2]: eof_mask_dis Bit[1]: sof_mask_dis Bit[0]: all_mask_dis

6.14 format control [0x4300 - 0x4302, 0x4320 - 0x4329]

table 6-14 format control registers

address	register name	default value	R/W	description
0x4300	CLIP MAX HI	0xFF	RW	Bit[7:0]: clip_max[11:4]
0x4301	CLIP MIN HI	0x00	RW	Bit[7:0]: clip_min[11:4]
0x4302	CLIP LO	0x0F	RW	Bit[7:4]: clip_min[3:0] Bit[3:0]: clip_max[3:0]
0x4316	CTRL16	0x00	RW	Bit[7:1]: r_seof_vsync_delay[7:1] Bit[0]: r_dpcm_en
0x4320	TEST PATTERN CTRL	0x80	RW	Bit[7:6]: pixel_order 00: GR/BG 01: RG/GB 10: BG/GR 11: GB/RG Bit[5]: byte_swap Bit[4]: bit_reverse Bit[3:2]: Not used Bit[1]: solid_color_en Bit[0]: pn31_enable
0x4321	PN31 CTRL	0x00	RW	Bit[7:4]: Not used Bit[3]: PN31 LSB first enable Bit[2]: PN31 reset by sof enable Bit[1]: PN31 reset by HREF enable Bit[0]: PN9 enable
0x4322	SOLID COLOR B	0x00	RW	Bit[7:2]: Not used Bit[1:0]: solid_color_b[9:8]
0x4323	SOLID COLOR B	0x00	RW	Bit[7:0]: solid_color_b[7:0]
0x4324	SOLID COLOR GB	0x00	RW	Bit[7:2]: Not used Bit[1:0]: solid_color_gb[9:8]
0x4325	SOLID COLOR GB	0x00	RW	Bit[7:0]: solid_color_gb[7:0]
0x4326	SOLID COLOR R	0x00	RW	Bit[7:2]: Not used Bit[1:0]: solid_color_r[9:8]
0x4327	SOLID COLOR R	0x00	RW	Bit[7:0]: solid_color_r[7:0]
0x4328	SOLID COLOR GR	0x00	RW	Bit[7:2]: Not used Bit[1:0]: solid_color_gr[9:8]
0x4329	SOLID COLOR GR	0x00	RW	Bit[7:0]: solid_color_gr[7:0]

6.15 MIPI control [0x4800 - 0x4851]

table 6-15 MIPI control registers (sheet 1 of 9)

address	register name	default value	R/W	description
0x4800	MIPI CTRL00	0x4C	RW	Bit[7]: Writing '1' to this bit will stop clock lane once at vblk Bit[6]: gate_sc_vblk_en 0: Not used 1: Enable gate clock lane only when vblanking Bit[5]: gate_sc_en 0: Clock lane is free running 1: Gate clock lane when there is no packet to transmit Bit[4]: line_sync_en 0: Do not send line short packet for each line 1: Send line short packet for each line Bit[3]: Enable clock lane stop at hblk when in sleep mode Bit[2:0]: Not used
0x4801	MIPI CTRL01	0x00	RW	Bit[7]: Not used Bit[6]: spkt_dt_sel 0: Not used 1: Use dt_spkt as short packet data Bit[5]: first_bit Change clk_lane first bit 0: Output 0x05 1: Output 0xAA Bit[4:2]: Not used Bit[1]: LPX_select for PCLK domain 0: Auto calculate t_lpx_p, unit pclk2x cycle 1: Use lpx_p_min[7:0] Bit[0]: Not used

table 6-15 MIPI control registers (sheet 2 of 9)

address	register name	default value	R/W	description
0x4802	MIPI CTRL02	0x00	RW	Bit[7]: hs_prepare_sel 0: Auto calculate T_hs_prepare, unit pclk2x 1: Use hs_prepare_min_o[7:0] Bit[6]: clk_prepare_sel 0: Auto calculate T_clk_prepare, unit pclk2x 1: Use clk_prepare_min_o[7:0] Bit[5]: clk_post_sel 0: Auto calculate T_clk_post, unit pclk2x 1: Use clk_post_min_o[7:0] Bit[4]: clk_trail_sel 0: Auto calculate T_clk_trail, unit pclk2x 1: Use clk_trail_min_o[7:0] Bit[3]: hs_exit_sel 0: Auto calculate T_hs_exit, unit pclk2x 1: Use hs_exit_min_o[7:0] Bit[2]: hs_zero_sel 0: Auto calculate T_hs_zero, unit pclk2x 1: Use hs_zero_min_o[7:0] Bit[1]: hs_trail_sel 0: Auto calculate T_hs_trail, unit pclk2x 1: Use hs_trail_min_o[7:0] Bit[0]: clk_zero_sel 0: Auto calculate T_clk_zero, unit pclk2x 1: Use clk_zero_min_o[7:0]
0x4803	MIPI CTRL03	0x00	RW	Bit[7:4]: Not used Bit[3]: manu_ofset_o t_period manual offset SMIA Bit[2]: r_manu_half2one t_period half to 1 SMIA Bit[1]: clk_pre_half Bit[0]: hs_pre_half
0x4804	MIPI CTRL04	0x04	RW	Bit[7:4]: man_lane_num Bit[3]: lane_num_manaul_enable Bit[2]: lane4_6b_en 0: Not used 1: Support 4, 7, 8 lane 6-bit Bit[1]: Vsub select 0: Valid in behind 1: Valid in front Bit[0]: vfifo_8x

table 6-15 MIPI control registers (sheet 3 of 9)

address	register name	default value	R/W	description
0x4805	MIPI CTRL05	0x00	RW	Bit[7:4]: Not used Bit[3]: lpda_retim_manu_o Bit[2]: lpda_retim_sel_o 0: Not used 1: Manual Bit[1]: lpck_retim_manu_o Bit[0]: lpck_retim_sel_o 0: Not used 1: Manual
0x4806	MIPI CTRL06	0x10	RW	Bit[7]: Not used Bit[6]: Suspend latch at horizontal blanking Bit[5]: Suspend latch at vertical blanking Bit[4]: pu_mark_en_o Power up mark1 enable Bit[3]: mipi_remot_rst Bit[2]: mipi_susp Bit[1]: smia_lane_ch_en Bit[0]: tx_lsb_first 0: High bit first 1: Low power transmit low bit first
0x4807	MIPI CTRL07	0x03	RW	Bit[7:4]: Not used Bit[3:0]: sw_t_lpx Ultra low power T_lpx
0x4808	MIPI CTRL08	0x0A	RW	Bit[7:0]: wkup_dly Mark1 wakeup delay/2^10
0x4810	FCNT MAX	0xFF	RW	Bit[7:0]: fcnt_max[15:8] High byte of maximum frame counter of frame sync short packet
0x4811	FCNT MAX	0xFF	RW	Bit[7:0]: fcnt_max[7:0] Low byte of maximum frame counter of frame sync short packet
0x4813	MIPI CTRL13	0x00	RW	Bit[7:3]: Not used Bit[2]: vc_sel Bit[1:0]: VC ID
0x4814	MIPI CTRL14	0x2A	RW	Bit[7]: Not used Bit[6]: lpkt_dt_sel 0: Use mipi_dt 1: Use dt_man_o as long packet data Bit[5:0]: dt_man Manual data type

table 6-15 MIPI control registers (sheet 4 of 9)

address	register name	default value	R/W	description
0x4815	MIPI CTRL15	0x00	RW	Bit[7]: Not used Bit[6]: pclk_inv 0: Using falling edge of mipi_pclk_o to generate MIPI bus to PHY 1: Using rising edge of mipi_pclk_o to generate MIPI bus to PHY Bit[5:0]: manu_dt_short Manual type for short packet
0x4816	EMB DT	0x53	RW	Bit[7:6]: Not used Bit[5:0]: emb_dt Manual set embedded data type
0x4818	HS ZERO MIN	0x00	RW	Bit[7:2]: Not used Bit[1:0]: hs_zero_min[9:8] High byte of minimum value of hs_zero, unit ns
0x4819	HS ZERO MIN	0x70	RW	Bit[7:0]: hs_zero_min[7:0] Low byte of minimum value of hs_zero hs_zero_real = hs_zero_min_o + Tui*ui_hs_zero_min_o
0x481A	HS TRAIL MIN	0x00	RW	Bit[7:2]: Not used Bit[1:0]: hs_trail_min[9:8] High byte of minimum value of hs_trail, unit ns
0x481B	HS TRAIL MIN	0x3C	RW	Bit[7:0]: hs_trail_min[7:0] Low byte of minimum value of hs_trail hs_trail_real = hs_trail_min_o + Tui*ui_hs_trail_min_o
0x481C	CLK ZERO MIN	0x01	RW	Bit[7:2]: Not used Bit[1:0]: clk_zero_min[9:8] High byte of minimum value of clk_zero, unit ns
0x481D	CLK ZERO MIN	0x2C	RW	Bit[7:0]: clk_zero_min[7:0] Low byte of minimum value of clk_zero clk_zero_real = clk_zero_min_o + Tui*ui_clk_zero_min_o
0x481E	CLK PREPARE MAX	0x5F	RW	Bit[7:0]: clk_prepare_max[7:0] Maximum value of clk_prepare, unit ns
0x481F	CLK PREPARE MIN	0x26	RW	Bit[7:0]: clk_prepare_min[7:0] Minimum value of clk_prepare clk_prepare_real = clk_prepare_min_o + Tui*ui_clk_prepare_min_o

table 6-15 MIPI control registers (sheet 5 of 9)

address	register name	default value	R/W	description
0x4820	CLK POST MIN	0x00	RW	Bit[7:2]: Not used Bit[1:0]: clk_post_min[9:8] High byte of minimum value of clk_post, unit ns
0x4821	CLK POST MIN	0x3C	RW	Bit[7:0]: clk_post_min[7:0] Low byte of minimum value of clk_post clk_post_real = clk_post_min_o + Tui*ui_clk_post_min_o
0x4822	CLK TRAIL MIN	0x00	RW	Bit[7:2]: Not used Bit[1:0]: clk_trail_min[9:8] High byte of minimum value of clk_trail, unit ns
0x4823	CLK TRAIL MIN	0x3C	RW	Bit[7:0]: clk_trail_min[7:0] Low byte of minimum value of clk_trail clk_trail_real = clk_trail_min_o + Tui*ui_clk_trail_min_o
0x4824	LPX P MIN	0x00	RW	Bit[7:2]: Not used Bit[1:0]: lpx_p_min[9:8] High byte of minimum value of lpx_p, unit ns
0x4825	LPX P MIN	0x32	RW	Bit[7:0]: lpx_p_min[7:0] Low byte of minimum value of lpx_p lpx_p_real = lpx_p_min_o + Tui*ui_lpx_p_min_o
0x4826	HS PREPARE MIN	0x32	RW	Bit[7:0]: hs_prepare_min[7:0] Minimum value of hs_prepare, unit ns
0x4827	HS PREPARE MAX	0x55	RW	Bit[7:0]: hs_prepare_max[7:0] Maximum value of hs_prepare hs_prepare_real = hs_prepare_max_o + Tui*ui_hs_prepare_max_o
0x4828	HS EXIT MIN	0x00	RW	Bit[7:2]: Not used Bit[1:0]: hs_exit_min[9:8] High byte of minimum value of hs_exit, unit ns
0x4829	HS EXIT MIN	0x64	RW	Bit[7:0]: hs_exit_min[7:0] Low byte of minimum value of hs_exit hs_exit_real = hs_exit_min_o + Tui*ui_hs_exit_min_o
0x482A	UI HS ZERO MIN	0x06	RW	Bit[7:6]: Not used Bit[5:0]: ui_hs_zero_min[5:0] Minimum UI value of hs_zero, unit UI

table 6-15 MIPI control registers (sheet 6 of 9)

address	register name	default value	R/W	description
0x482B	UI HS TRAIL MIN	0x04	RW	Bit[7:6]: Not used Bit[5:0]: ui_hs_trail_min[5:0] Minimum UI value of hs_trail, unit UI
0x482C	UI CLK ZERO MIN	0x00	RW	Bit[7:6]: Not used Bit[5:0]: ui_clk_zero_min[5:0] Minimum UI value of clk_zero, unit UI
0x482D	UI CLK PREPARE	0x00	RW	Bit[7:4]: ui_clk_prepare_max Maximum UI value of clk_prepare, unit UI Bit[3:0]: ui_clk_prepare_min Minimum UI value of clk_prepare, unit UI
0x482E	UI CLK POST MIN	0x34	RW	Bit[7:6]: Not used Bit[5:0]: ui_clk_post_min[5:0] Minimum UI value of clk_post, unit UI
0x482F	UI CLK TRAIL MIN	0x00	RW	Bit[7:6]: Not used Bit[5:0]: ui_clk_trail_min[5:0] Minimum UI value of clk_trail, unit UI
0x4830	UI LPX P MIN	0x00	RW	Bit[7:6]: Not used Bit[5:0]: ui_lpx_p_min[5:0] Minimum UI value of lpx_p (pclk2x domain), unit UI
0x4831	UI HS PREPARE	0x64	RW	Bit[7:4]: ui_hs_prepare_max Maximum UI value of hs_prepare, unit UI Bit[3:0]: ui_hs_prepare_min Minimum UI value of hs_prepare, unit UI
0x4832	UI HS EXIT MIN	0x00	RW	Bit[7:6]: Not used Bit[5:0]: ui_hs_exit_min[5:0] Minimum UI value of hs_exit, unit UI
0x4833	MIPI PKT STAR SIZE	0x08	RW	Bit[7:6]: Not used Bit[5:0]: r_rdy_mark
0x4837	PCLK PERIOD	0x1A	RW	Bit[7:0]: pclk_period[7:0] Period of pclk2x, pclk_div=1, and 1 bit decimal

table 6-15 MIPI control registers (sheet 7 of 9)

address	register name	default value	R/W	description
0x4838	MIPI LP GPIO0	0x00	RW	Bit[7]: lp_sel0 0: Auto generate mipi_lp_dir0_o 1: Use lp_dir_man0 to be mipi_lp_dir0_o Bit[6]: lp_dir_man0 0: Input 1: Output Bit[5]: lp_p0_o Bit[4]: lp_n0_o Bit[3]: lp_sel1 0: Auto generate mipi_lp_dir1_o 1: Use lp_dir_man1 to be mipi_lp_dir1_o Bit[2]: lp_dir_man1 0: Input 1: Output Bit[1]: lp_p1_o Bit[0]: lp_n1_o
0x4839	MIPI LP GPIO1	0x00	RW	Bit[7]: lp_sel2 0: Auto generate mipi_lp_dir2_o 1: Use lp_dir_man2 to be mipi_lp_dir2_o Bit[6]: lp_dir_man2 0: Input 1: Output Bit[5]: lp_p2_o Bit[4]: lp_n2_o Bit[3]: lp_sel3 0: Auto generate mipi_lp_dir3_o 1: Use lp_dir_man3 to be mipi_lp_dir3_o Bit[2]: lp_dir_man3 0: Input 1: Output Bit[1]: lp_p3_o Bit[0]: lp_n3_o
0x483C	MIPI CTRL3C	0x02	RW	Bit[7:4]: Not used Bit[3:0]: t_clk_pre Unit: pclk2x cycle

table 6-15 MIPI control registers (sheet 8 of 9)

address	register name	default value	R/W	description
0x483D	MIPI LP GPIO4	0x00	RW	Bit[7]: lp_ck_sel0 0: Auto generate mipi_ck_lp_dir0_o 1: Use lp_ck_dir_man0 to be mipi_ck_lp_dir0_o Bit[6]: lp_ck_dir_man0 0: Input 1: Output Bit[5]: lp_ck_p0_o Bit[4]: lp_ck_n0_o Bit[3]: lp_ck_sel1 0: Auto generate mipi_ck_lp_dir1_o 1: Use lp_ck_dir_man1 to be mipi_ck_lp_dir1_o Bit[2]: lp_ck_dir_man1 0: Input 1: Output Bit[1]: lp_ck_p1_o Bit[0]: lp_ck_n1_o
0x484A	SEL MIPI CTRL4A	0x27	RW	Bit[7:6]: Not used Bit[5]: slp_lp_pon_man_o Set for power up Bit[4]: slp_lp_pon_da Bit[3]: slp_lp_pon_ck Bit[2]: mipi_slp_man_st MIPI bus status manual control enable in sleep mode Bit[1]: clk_lane_state Bit[0]: data_lane_state
0x484B	SMIA OPTION	0x07	RW	Bit[7:3]: Not used Bit[2]: line_st_sel_o 0: Line starts after HREF 1: Line starts after fifo_st Bit[1]: clk_start_sel_o 0: Clock starts after SOF 1: Clock starts after reset Bit[0]: sof_sel_o 0: Frame starts after HREF occurs 1: Frame starts after SOF
0x484C	SEL MIPI CTRL4C	0x03	RW	Bit[7]: Not used Bit[6]: smia_fcnt_i select Bit[5]: prbs_enable Bit[4]: hs_test_only MIPI high speed only test mode enable Bit[3]: set_frame_cnt_0 Set frame count to inactive mode (keep 0) Bit[2:0]: Not used

table 6-15 MIPI control registers (sheet 9 of 9)

address	register name	default value	R/W	description
0x484D	TEST PATTEN DATA	0xB6	RW	Bit[7:0]: test_patten_data[7:0] Data lane test pattern
0x484E	FE DLY	0x10	RW	Bit[7:0]: r_fe_dly_o Last packet to frame end delay / 2
0x484F	TEST PATTEN CK DATA	0x55	RW	Bit[7:2]: Not used Bit[1:0]: clk_test_patten_reg
0x4850	LANE SEL01	0x12	RW	Bit[7]: Not used Bit[6:4]: lane1_sel Bit[3]: Not used Bit[2:0]: lane0_sel
0x4851	LANE SEL23	0x03	RW	Bit[7]: Not used Bit[6:4]: lane3_sel Bit[3]: Not used Bit[2:0]: lane2_sel

6.16 ISPFC [0x4900 - 0x4903]

table 6-16 ISPFC control registers

address	register name	default value	R/W	description
0x4900	R0	0x00	RW	Bit[7:4]: Not used Bit[3]: sof_after_line0 Bit[2]: fcnt_eof_sel Bit[1]: fcnt_mask_dis Bit[0]: fcnt_reset
0x4901	R1	0x00	RW	Bit[7:4]: Not used Bit[3:0]: frame_on_number
0x4902	R2	0x00	RW	Bit[7:4]: Not used Bit[3:0]: frame_off_number 0x4901, 0x 4902=x,0: on x frames and off 0x4901, 0x4902=0,x: off x frames and on 0x4901, 0x4902=x,y: on x frames and off y frames
0x4903	R3	0x00	RW	Bit[7]: zero_line_mask_dis Bit[6]: rblue_mask_dis Bit[5]: data_mask_dis Bit[4]: valid_mask_dis Bit[3]: href_mask_dis Bit[2]: eof_mask_dis Bit[1]: sof_mask_dis Bit[0]: all_mask_dis

6.17 MIPI_RX control [0x4A00 - 0x4A3A]

table 6-17 MIPI control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x4A00	MIPI_RX_REG0	0x01	RW	Bit[7]: mipi_rx_tst_en Bit[6:4]: mipi_rx_tst_ctrl Bit[3]: sync_delay_byp Bit[2:1]: delay_num Bit[0]: syn_sigerr_dis
0x4A01	MIPI_RX_REG1	0x00	RW	Bit[7:4]: Not used Bit[3]: rx_vld_dly_sel Bit[2]: Not used Bit[1:0]: lane_number
0x4A02	MIPI_RX_REG2	0x00	RW	Bit[7:6]: rx_vc Bit[5]: Not used Bit[4]: ecc_byp 0: Not used 1: Do not check ECC Bit[3]: ph_bit_order Bit[2:1]: ph_order Bit[0]: spkt_line_sync
0x4A03	MIPI_RX_TYPE	0x00	RW	Bit[7]: Not used Bit[6]: type_select 0: Auto data type 1: Select r_data_type Bit[5:0]: r_data_type
0x4A04	MIPI_RX_PRBS	0x00	RW	Bit[7]: prbs_sel Check data generate from 0: Auto generate 1: Input data Bit[6:1]: Not used Bit[0]: prbs_en
0x4A05	MIPI_RX_VALID_ CNT	0x0A	RW	Bit[7:0]: rx_vld_dly_cnt
0x4A06	MIPI_RX_REG6	0x04	RW	Bit[7:5]: Not used Bit[4]: h_l_swap 0: Data output fromdata_o[19:10] 1: Data output from data_o[9:0] Bit[3]: pclk_inv Bit[2]: cif_2pix_en Bit[1:0]: Not used
0x4A0A	MIPI_RX_PRBS_ ERR	0x10	RW	Bit[7:0]: PRBS error counter

table 6-17 MIPI control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x4A0C	MIPI_RX_ERR_CLR	0x00	RW	Bit[7]: byte_num clear Bit[6]: data_type_rx clear Bit[5]: rx_prbs_err clear Bit[4]: rx_prbs_flag_err clear Bit[3]: rx_byte_err clear Bit[2]: rx_crc_err clear Bit[1]: rx_ecc_err clear Bit[0]: rx_sync_err clear
0x4A20	MIPI_RX_REG20	–	R	Bit[7:0]: byte_num[15:8]
0x4A21	MIPI_RX_REG21	–	R	Bit[7:0]: byte_num[7:0]
0x4A22	MIPI_RX_REG22	–	R	Bit[7:6]: Not used Bit[5:0]: data_type_rx
0x4A23	MIPI_RX_REG23	–	R	Bit[7:0]: data_cif_i[15:8]
0x4A24	MIPI_RX_REG24	–	R	Bit[7:0]: data_cif_i[7:0]
0x4A25	MIPI_RX_REG25	–	R	Bit[7]: dec_href Bit[6: rcv_href Bit[5:4]: dec_valid[1:0] Bit[3:2]: rcv_valid[1:0] Bit[1]: sync_err_1 Bit[0]: sync_sigerr_1
0x4A26	MIPI_RX_REG26	–	R	Bit[7]: sync_err_0 Bit[6]: sync_sigerr_0 Bit[5]: ecc_single_err Bit[4]: ecc_fatal_err Bit[3]: crc_err Bit[2]: byte_cnt_err Bit[1]: rx_prbs_pass1 Bit[0]: rx_prbs_pass0
0x4A27	MIPI_RX_REG27	–	R	Bit[7]: rx_prbs_fail1 Bit[6]: rx_prbs_fail0 Bit[5]: prbs_error1 Bit[4]: prbs_error0 Bit[3]: rx_pvalid_0_i Bit[2]: rx_pvalid_1_i Bit[1]: sof_cif Bit[0]: eof_o
0x4A30	MIPI_RX_ERR_FLAG	–	R	Bit[7:0]: sync_err[7:0]
0x4A31	MIPI_RX_ERR_FLAG	–	R	Bit[7:6]: ecc_err Bit[5]: crc_err Bit[4]: byte_err Bit[3:0]: rx_prbs_err
0x4A32	MIPI_RX_ERR_FLAG	–	R	Bit[7:0]: prbs_flag_err[7:0]

table 6-17 MIPI control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x4A33	MIPI_RX_ERR_FLAG	–	R	Bit[7:2]: data_type_rx[5:0] Bit[1:0]: Not used
0x4A34	MIPI_RX_REG34	–	R	Bit[7:0]: byte_num[15:8]
0x4A35	MIPI_RX_REG35	–	R	Bit[7:0]: byte_num[7:0]
0x4A36	MIPI_RX_REG36	–	R	Bit[7:0]: prbs_err0_cnt
0x4A37	MIPI_RX_REG37	–	R	Bit[7:0]: prbs_err1_cnt
0X4A38	MIPI_RX_REG38	–	R	Bit[7:0]: prbs_err2_cnt
0x4A39	MIPI_RX_REG39	–	R	Bit[7:0]: prbs_err3_cnt
0x4A3A	MIPI_RX_REG3A	–	R	Bit[7:0]: pn31_tst_err_cnt

6.18 temperature sensor [0x4D00 - 0x4D14]

table 6-18 temperature sensor registers

address	register name	default value	R/W	description
0x4D00~ 0x4D0F	TPM_CTRL_REG	–	–	Temperature Sensor Control Registers
0x4D10	TPM_CTRL_10	0x00	RW	Bit[7:0]: r_tpm_min
0x4D11	TPM_CTRL_11	0xFF	RW	Bit[7:0]: r_tpm_max
0x4D12	TPM_CTRL_12	–	W	Writing 0x4D12[0] to '1' will trigger temperature calculating, then 0x4D12 and 0x4D13 will be the latched temperature value
0x4D13	TPM_CTRL_13	–	R	Latched Temperature Value, Integer Part
0x4D14	TPM_CTRL_14	–	R	Latched Temperature Value, Decimal Part

6.19 EXT_LENC control [0x4E00 - 0x4A3A]

table 6-19 EXT_LENC control registers (sheet 1 of 5)

address	register name	default value	R/W	description
0x4E00	EXT LENC G00	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G00 for luminance compensation
0x4E01	EXT LENC G01	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G01 for luminance compensation
0x4E02	EXT LENC G02	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G02 for luminance compensation
0x4E03	EXT LENC G03	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G03 for luminance compensation
0x4E04	EXT LENC G04	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G04 for luminance compensation
0x4E05	EXT LENC G05	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G05 for luminance compensation
0x4E06	EXT LENC G10	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G10 for luminance compensation
0x4E07	EXT LENC G11	0x08	RW	Bit[7:6]: Not used Bit[5:0]: Control point G11 for luminance compensation
0x4E08	EXT LENC G12	0x08	RW	Bit[7:6]: Not used Bit[5:0]: Control point G12 for luminance compensation
0x4AE9~ 0x4E22	EXT LENC G13~ LENC EXT G54	–	RW	Bit[7:6]: Not used Bit[5:0]: Control point G13~G54 for luminance compensation
0x4E23	EXT LENC G55	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G55 for luminance compensation
0x4E24	EXT LENC BR00	0xAA	RW	Bit[7:4]: Control point B00 for blue channel compensation Bit[3:0]: Control point R00 for red channel compensation

table 6-19 EXT_LENC control registers (sheet 2 of 5)

address	register name	default value	R/W	description
0x4E25	EXT LENC BR01	0xAA	RW	Bit[7:4]: Control point B01 for blue channel compensation Bit[3:0]: Control point R01 for red channel compensation
0x4E26	EXT LENC BR02	0xAA	RW	Bit[7:4]: Control point B02 for blue channel compensation Bit[3:0]: Control point R02 for red channel compensation
0x4E27	EXT LENC BR03	0xAA	RW	Bit[7:4]: Control point B03 for blue channel compensation Bit[3:0]: Control point R03 for red channel compensation
0x4E28	EXT LENC BR04	0xAA	RW	Bit[7:4]: Control point B04 for blue channel compensation Bit[3:0]: Control point R04 for red channel compensation
0x4E29~ 0x4E3C	EXT LENC BR10~ LENC BR44	–	RW	Bit[7:4]: Control point B10~B44 for blue channels compensation Bit[3:0]: Control point R10~R44 for red channels compensation
0x4E3D	EXT LENC BROFFSET	0x88	RW	Bit[7:4]: Base value for all blue channel control points Bit[3:0]: Base value for all red channel control points
0x4E3E	EXT LENC MAXGAIN	0x40	RW	Bit[7:0]: If AutoLensSwitchEnable is true and sensor gain is larger than this threshold, luminance compensation amplitude will be the minimum value (min LENC gain). Register value is 16 times sensor gain.
0x4E3F	EXT LENC EXT MINGAIN	0x20	RW	Bit[7:0]: If AutoLensSwitchEnable is true and sensor gain is larger than this threshold, luminance compensation amplitude will start to decrease; otherwise, the amplitude will not change. Register value is 16 times sensor gain.

table 6-19 EXT_LENC control registers (sheet 3 of 5)

address	register name	default value	R/W	description
0x4E40	EXT LENC MINQ	0x18	RW	Bit[7]: Not used Bit[6:0]: This value indicates the minimum amplitude which luminance channel compensates when AutoLensSwitchEnable is true. Value should be in the range [0~64]
0x4E41	EXT LENC CTRL	0x0D	RW	Bit[7:4]: Not used Bit[3]: Add BLC target 0: Do not add BLC target after applying compensation 1: Add BLC target after applying compensation Bit[2]: Subtract BLC target 0: Do not subtract BLC target after applying compensation 1: Subtract BLC target after applying compensation Bit[1]: Reserved Bit[0]: AutoLensSwitchEnable 0: Luminance compensation amplitude does not change with sensor gain 1: Luminance compensation amplitude changes with sensor gain
0x4E42	EXT_LENC BRHSCALE	0x00	RW	Bit[7:3]: Not used Bit[2:0]: BRHScale[10:8] For horizontal color gain calculation, this value indicates the step between two connected horizontal pixels, where BRHScale = 3×2^18 / ImageWidth, (In OV8858, ImageWidth=3296, ImageHeight=2480)
0x4E43	EXT_LENC BRHSCALE	0xEE	RW	Bit[7:0]: BRHScale[7:0]

table 6-19 EXT_LENC control registers (sheet 4 of 5)

address	register name	default value	R/W	description
0x4E44	EXT_LENC BRVSCALE	0x01	RW	Bit[7:3]: not used Bit[2:0]: BRVScale[10:8] For vertical color gain calculation, this value indicates the step between two connected vertical pixels, where BRVScale = 3×2^18 / ImageHeight
0x4E45	EXT_LENC BRVSCALE	0x3D	RW	Bit[7:0]: BRVScale[7:0]
0x4E46	EXT_LENC GHSCALE	0x01	RW	Bit[7:3]: Not used Bit[2:0]: GHScale[10:8] For horizontal luminance gain calculation, this value indicates the step between two connected horizontal pixels, where GHScale = 4×2^18 / ImageWidth
0x4E47	EXT_LENC GHSCALE	0x3E	RW	Bit[7:0]: GHScale[7:0]
0x4E48	EXT_LENC GVSCALE	0x00	RW	Bit[7:3]: Not used Bit[2:0]: GVScale[10:8] For vertical luminance gain calculation, this value indicates the step between two connected horizontal pixels, where GVScale = 4×2^17 / ImageHeight
0x4E49	EXT_LENC GVSCALE	0xD3	RW	Bit[7:0]: GVScale[7:0]
0x4E4A~ 0x4E4F	NOT USED	–	–	Not Used
0x4E50	EXT_LENC XOFFSET	–	R	Bit[7:4]: Not used Bit[3:0]: Input sensor horizontal offset[11:8]
0x4E51	EXT_LENC XOFFSET	–	R	Bit[7:0]: Input sensor horizontal offset[7:0]
0x4E52	EXT_LENC YOFFSET	–	R	Bit[7:4]: Not used Bit[3:0]: Input sensor vertical offset[11:8]
0x4E53	EXT_LENC YOFFSET	–	R	Bit[7:0]: Input sensor vertical offset[7:0]

table 6-19 EXT_LENC control registers (sheet 5 of 5)

address	register name	default value	R/W	description
0x4E54	EXT_LENC INPUT	–	R	Bit[7:6]: Not used Bit[5]: Input sensor flip Bit[4]: Input sensor mirror Bit[3:2]: Input sensor Y skip Bit[1:0]: Input sensor X skip
0x4E55	EXT_LENC OVERFLOW	–	R	Bit[7:4]: Not used Bit[3]: GH overflow for debug Bit[2]: BRH overflow for debug Bit[1]: GV overflow for debug Bit[0]: BRV overflow for debug
0x4E56	EXT_LENC QVALUE	–	R	Bit[7]: Not used Bit[6:0]: Real amplitude Q value
0x4E80	EXT_LENC REG80	0x00	RW	Bit[7:6]: Not used Bit[5:4]: px_order Bit[3]: Not used Bit[2]: Flip Bit[1]: Mirror Bit[0]: lenc_en
0x4E81	EXT_LENC REG81	0x00	RW	Bit[7:3]: Not used Bit[2:0]: real_gain[10:8]
0x4E82	EXT_LENC REG82	0x00	RW	Bit[7:0]: real_gain[7:0]
0x4E83	EXT_LENC REG83	0x00	RW	Bit[7:0]: Bias[7:0]
0x4E84	EXT_LENC REG84	0x00	RW	Bit[7:4]: Not used Bit[3:0]: x_offset[11:8]
0x4E85	EXT_LENC REG85	0x00	RW	Bit[7:0]: x_offset[7:0]
0x4E86	EXT_LENC REG86	0x00	RW	Bit[7:4]: Not used Bit[3:0]: y_offset[11:8]
0x4E87	EXT_LENC REG87	0x00	RW	Bit[7:0]: y_offset[7:0]
0x4E88	EXT_LENC REG88	0x00	RW	Bit[7:6]: Not used Bit[5:4]: y_skip Bit[3:2]: Not used Bit[1:0]: x_skip

6.20 ISP control [0x5000 - 0x5065]

table 6-20 ISP control registers (sheet 1 of 10)

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xFE	RW	Bit[7]: Lens correction (LENC) function enable Bit[6]: Slave sensor AWB Gain function enable Bit[5]: Slave sensor AWB Statistics function enable Bit[4]: Master sensor AWB Gain function enable Bit[3]: Master sensor AWB Statistics function enable Bit[2]: Black DPC function enable Bit[1]: White DPC function enable Bit[0]: Not used
0x5001	ISP CTRL01	0x01	RW	Bit[7:1]: Not used Bit[0]: BLC function enable
0x5002	ISP CTRL02	0x28	RW	Bit[7]: Horizontal scale function enable Bit[6]: WBMATCH bypass mode 0: Select slave sensor's gain 1: Select master sensor's gain Bit[5]: WBMATCH function enable Bit[4]: Master MWB gain support RGBC Bit[3]: OTP_DPC function enable Bit[2]: Manual mode of VarioPixel® function enable Bit[1]: Manual enable of VarioPixel® function enable Bit[0]: Use VSYNC to latch ISP modules' function enable signals
0x5003	ISP CTRL03	0x20	RW	Bit[7]: Not used Bit[6]: Bypass mode Bypass all ISP modules after BLC module Bit[5]: DPC_DBC buffer control enable Bit[4]: WBMATCH VSYNC selection 0: Select master sensor's VSYNC fall 1: Select slave sensor's VSYNC fall Bit[3:2]: Not used Bit[1]: Select master AWB gain to embed line 0: Select master AWB gain before manual mode 1: Select master AWB gain after manual mode Bit[0]: Enable BLC's input flip_i signal 0: Disable BLC's input flip_i signal 1: Enable BLC's input flip_i signal

table 6-20 ISP control registers (sheet 2 of 10)

address	register name	default value	R/W	description
0x5004	ISP CTRL04	0x0C	RW	Bit[7]: Bypass master sensor's MWB gain Bit[6]: Manual enable of WBMATCH gain Bit[5]: Manual mode of frame counter for master and slave sensor's average Bit[4]: Manual enable of frame counter for master and slave sensor's average Bit[3]: Auto mode of master sensor's input size Bit[2]: Auto mode of slave sensor's input size Bit[1]: Reverse Gfirst signal of master sensor Bit[0]: Reverse Rblue signal of master sensor
0x5005	ISP CTRL05	0xF0	RW	Bit[7]: Enable hscale bias Subtract BLC target before 2x gain mode, and add it back after 2x gain mode Bit[6]: Enable LENC bias Subtract BLC target before LENC gain, and add it back after LENC gain Bit[5]: Enable master sensor's MWB bias Subtract BLC target before MWB gain, and add it back after MWB gain Bit[4]: Reserved Bit[3:0]: Post binning option Reverse the mirror / flip / pixel_order signals to post binning module
0x5006	ISP HSIZE IN	0x0C	RW	Bit[7:4]: Reserved Bit[3:0]: Manual mode of master ISP input width[11:8]
0x5007	ISP HSIZE IN	0xE0	RW	Bit[7:0]: Manual mode of master ISP input width[7:0]
0x5008	ISP VSIZE IN	0x09	RW	Bit[7:4]: Reserved Bit[3:0]: Manual mode of master ISP input height[11:8]
0x5009	ISP VSIZE IN	0xB0	RW	Bit[7:0]: Manual mode of master ISP input height[7:0]
0x500A	SLAVE HSIZE IN	0x08	RW	Bit[7:4]: Reserved Bit[3:0]: Manual mode of slave ISP input width[11:8]
0x500B	SLAVE HSIZE IN	0x00	RW	Bit[7:0]: Manual mode of slave ISP input width[7:0]
0x500C	SLAVE VSIZE IN	0x06	RW	Bit[7:4]: Reserved Bit[3:0]: Manual mode of slave ISP input height[11:8]
0x500D	SLAVE VSIZE IN	0x00	RW	Bit[7:0]: Manual mode of slave ISP input height[7:0]

table 6-20 ISP control registers (sheet 3 of 10)

address	register name	default value	R/W	description
0x500E	DPC HSIZE IN	0x0C	RW	Bit[7:4]: Reserved Bit[3:0]: Manual mode of DPC input width[11:8]
0x500F	DPC HSIZE IN	0xE0	RW	Bit[7:0]: Manual mode of DPC input width[7:0]
0x5010	DPC VSIZE IN	0x09	RW	Bit[7:4]: Reserved Bit[3:0]: Manual mode of DPC input height[11:8]
0x5011	DPC VSIZE IN	0xB0	RW	Bit[7:0]: Manual mode of DPC input height[7:0]
0x5012	WBMATCH R	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual WBMATCH red gain[11:8]
0x5013	WBMATCH R	0x00	RW	Bit[7:0]: Manual WBMATCH red gain[7:0]
0x5014	WBMATCH G	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual WBMATCH green gain[11:8]
0x5015	WBMATCH G	0x00	RW	Bit[7:0]: Manual WBMATCH green gain[7:0]
0x5016	WBMATCH B	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual WBMATCH blue gain[11:8]
0x5017	WBMATCH B	0x00	RW	Bit[7:0]: Manual WBMATCH blue gain[7:0]
0x5018	AWBM R GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual red gain[11:8] for master AWB statistics
0x5019	AWBM R GAIN	0x00	RW	Bit[7:0]: Manual red gain[7:0] for master AWB statistics
0x501A	AWBM G GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual green gain[11:8] for master AWB statistics
0x501B	AWBM G GAIN	0x00	RW	Bit[7:0]: Manual green gain[7:0] for master AWB Statistics
0x501C	AWBM B GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual blue gain[11:8] for master AWB statistics
0x501D	AWBM B GAIN	0x00	RW	Bit[7:0]: Manual blue gain[7:0] for master AWB statistics

table 6-20 ISP control registers (sheet 4 of 10)

address	register name	default value	R/W	description
0x501E	ISP CTRL1E	0x91	RW	Bit[7:6]: Select master sensor's average input 00: Use the data after pre_DSP to calculate average 01: Use the data after binning post Processing to calculate average 1x: Use the data after VarioPixel to calculate average Bit[5]: Select slave sensor's average input 0: Before slave MWB gain 1: After slave MWB gain Bit[4]: Master sensor MWB input data selection 0: Before master MWB gain 1: After master MWB gain Bit[3]: Manual enable of master AWB statistics Bit[2]: Manual enable of slave AWB statistics Bit[1]: Digital gain function enable, shared with master sensor's MWB gain Bit[0]: Manual mode of master sensor's MWB gain
0x501F	ISP CTRL1F	0x00	RW	Bit[7:6]: Reserved Bit[5]: Bypass ISP enable Only bypass pre_DSP, LENC, MWB gain, OTP, DPC, BIN and hscale Bit[4]: Bit shift enable when ISP bypass is disabled Bit[3]: Bit shift direction 0: Left shift 1: Right shift Bit[2:0]: Bit shift number
0x5020	AWBS R GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual red gain[11:8] for slave AWB statistics
0x5021	AWBS R GAIN	0x00	RW	Bit[7:0]: Manual red gain[7:0] for slave AWB statistics
0x5022	AWBS G GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual green gain[11:8] for slave AWB statistics
0x5023	AWBS G GAIN	0x00	RW	Bit[7:0]: Manual green gain[7:0] for slave AWB statistics
0x5024	AWBS B GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual blue gain[11:8] for slave AWB statistics
0x5025	AWBS B GAIN	0x00	RW	Bit[7:0]: Manual blue gain[7:0] for slave AWB statistics

table 6-20 ISP control registers (sheet 5 of 10)

address	register name	default value	R/W	description
0x5026	BLC BIAS SLV	0x10	RW	Bit[7:0]: Slave sensor's BLC target
0x5027~ 0x5029	RSVD	–	–	Reserved
0x502A	ISP CTRL2A	0x00	RW	Bit[7]: VIV swap manual enable Bit[6]: VIV swap manual value Bit[5]: Append 0/1 for bit select module 0: Append 0 when left shift 1: Append 1 when left shift Bit[4]: Bit select manual enable Bit[3:2]: Bit select manual value Bit[1]: Manual mode of DPC input size (for manual input size, see registers 0x500E~0x5011) Bit[0]: Reserved
0x502B~ 0x502F	NOT USED	–	–	Not Used
0x5030	ISP CTRL30	0x30	RW	Bit[7]: Reserved Bit[6:4]: Average done mask Bit[3:0]: Reserved
0x5031	NOT USED	–	–	Not Used
0x5032	MWB R GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual red gain[11:8] for master MWB gain
0x5033	MWB R GAIN	0x00	RW	Bit[7:0]: Manual red gain[7:0] for master MWB gain
0x5034	MWB G GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual green gain[11:8] for master MWB gain
0x5035	MWB G GAIN	0x00	RW	Bit[7:0]: Manual green gain[7:0] for master MWB gain
0x5036	MWB B GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual blue gain[11:8] for master MWB gain
0x5037	MWB B GAIN	0x00	RW	Bit[7:0]: Manual blue gain[7:0] for master MWB gain
0x5038	MWB C GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Manual clear gain[11:8] for master MWB gain
0x5039	MWB C GAIN	0x00	RW	Bit[7:0]: Manual clear gain[7:0] for master MWB gain

table 6-20 ISP control registers (sheet 6 of 10)

address	register name	default value	R/W	description
0x503A~ 0x503C	NOT USED	–	–	Not Used
0x503D	ISP CTRL3D	0x58	RW	Bit[7]: Reserved Bit[6]: Auto mode of scale enable to pre_DSP module Disable scale module automatically when bypass ISP Bit[5]: Manual scale enable to pre_DSP ISP module, when auto mode is disabled Bit[4]: Window cut using the output size from pre_DSP Bit[3]: ISP raw enable (for pre_DSP to adjust window cut module's Y offset) Bit[2:0]: Adjust value of window cut module's Y offset
0x503E	ISP CTRL3E	0x00	RW	Bit[7]: Adjust enable for auto ISP input width 0: Plus adjust value 1: Minus adjust value Bit[6:0]: Adjust value for auto ISP input width
0x503F	ISP CTRL3F	0x00	RW	Bit[7]: Adjust enable for auto ISP input height 0: Plus adjust value 1: Minus adjust value Bit[6:0]: Adjust value for auto ISP input height
0x5040	NOT USED	–	–	Not Used
0x5041	ISP CTRL41	0x1C	RW	Bit[7:5]: Reserved Bit[4]: Post binning function enable Bit[3]: VIV crop window enable in average module Bit[2]: Average module function enable Bit[1]: Do average or sub-sample before AWB statistics for both slave and master sensor 0: Horizontal sub-sample 1: Horizontal average Bit[0]: Embedded line enable
0x5042	NOT USED	–	–	Not Used

table 6-20 ISP control registers (sheet 7 of 10)

address	register name	default value	R/W	description
0x5043	ISP CTRL43	0x08	RW	Bit[7:5]: Reserved Bit[4:3]: Subtract offset for average module's height Bit[2]: Horizontal post binning enable Bit[1]: Vertical post binning enable Bit[0]: Manual mode of post binning function enable In auto mode, post binning will disabled automatically if image size is larger than buffer size
0x5044	ISP CTRL44	0x00	RW	Bit[7]: Manual mode of input sensor Y offset (for manual value, see registers 0x504C and 0x504D) Bit[6]: Manual mode of input sensor X offset (for manual value, see registers 0x504A and 0x504B) Bit[5]: Reserved Bit[4]: Auto dummy line enable Bit[3:2]: Reserved Bit[1:0]: ISP EOF select 00: Auto mode of EOF 01: Last HREF falling from window cut module 10: EOF from timing control module 11: EOF from window cut module
0x5045	ISP CTRL45	0x01	RW	Bit[7:5]: Reserved Bit[4]: VSYNC plus for slave sensor 0: Using input VSYNC's falling edge as ISP's VSYNC 1: Using input VSYNC pulse as ISP's VSYNC Bit[3]: VSYNC plus for master sensor 0: Using input VSYNC's falling edge as ISP's VSYNC 1: Using input VSYNC pulse as ISP's VSYNC Bit[2]: Manual BLC target enable for master sensor Bit[1:0]: AEC trigger select 00: EOF from timing control 01: Average done signal from average 10: Last HREF fall from window cut 11: EOF from timing control (same as 00)

table 6-20 ISP control registers (sheet 8 of 10)

address	register name	default value	R/W	description
0x5046	ISP CTRL46	0x12	RW	Bit[7:6]: ISP SOF select 00: Auto mode of SOF 01: Use VSYNC as SOF 10: SOF from timing control 11: SOF from Pre_DSP module Bit[5]: Manual enable of post binning's pixel order Bit[4]: Manual value of post binning's pixel order Bit[3]: Manual enable of average's pixel order Bit[2]: Not used Bit[1:0]: Manual value of average's pixel order
0x5047	ISP CTRL47	0x92	RW	Bit[7:6]: Salve sensor's input pixel order Bit[5]: Manual enable of master MWB's pixel order Bit[4:3]: Manual value of master MWB's pixel order Bit[2]: Manual enable of pre_DSP's pixel order Bit[1:0]: Manual value of pre_DSP's pixel order
0x5048	ISP CTRL48	0x10	RW	Bit[7:0]: Master sensor's manual BLC target
0x5049	ISP CTRL49	0x20	RW	Bit[7]: DPC data switch 0: Not switch 1: Switch even and odd channel of DPC's input data Bit[6]: Manual enable of DPC's pixel order Bit[5:4]: Manual value of DPC's pixel order Bit[3:2]: Reserved Bit[1]: Manual enable of bin_mode in post binning module Bit[0]: Manual value of bin_mode in post binning module
0x504A	ISP CTRL4A	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Manual input sensor's X offset[11:8]
0x504B	ISP CTRL4B	0x00	RW	Bit[7:0]: Manual input sensor's X offset[7:0]
0x504C	ISP CTRL4C	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Manual input sensor's Y offset[11:8]
0x504D	ISP CTRL4D	0x00	RW	Bit[7:0]: Manual input sensor's Y offset[7:0]
0x504E~ 0x504F	NOT USED	–	–	Not Used
0x5050	BLC BIAS MST	–	R	Bit[7:0]: Master sensor's BLC target
0x5051	ISP CTRL51	–	R	Bit[7:0]: Real sensor gain

table 6-20 ISP control registers (sheet 9 of 10)

address	register name	default value	R/W	description
0x5052	ISP CTRL52	–	R	Bit[7:5]: Not used Bit[4:0]: Adjust master sensor's X offset [12:8] for OTP DPC module
0x5053	ISP CTRL53	–	R	Bit[7:0]: Adjust master sensor's X offset [7:0] for OTP DPC module
0x5054	ISP CTRL54	–	R	Bit[7:4]: Not used Bit[3:0]: Adjust master sensor's Y offset [11:8] for OTP DPC module
0x5055	ISP CTRL55	–	R	Bit[7:0]: Adjust master sensor's Y offset [7:0] for OTP DPC module
0x5056	AUTO CALCULATED WBMATCH RED GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Auto calculated WBMATCH red gain[11:8]
0x5057	AUTO CALCULATED WBMATCH RED GAIN	–	R	Bit[7:0]: Auto calculated WBMATCH red gain[7:0]
0x5058	AUTO CALCULATED WBMATCH GREEN GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Auto calculated WBMATCH green gain[11:8]
0x5059	AUTO CALCULATED WBMATCH GREEN GAIN	–	R	Bit[7:0]: Auto calculated WBMATCH green gain[7:0]
0x505A	AUTO CALCULATED WBMATCH BLUE GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Auto calculated WBMATCH blue gain[11:8]
0x505B	AUTO CALCULATED WBMATCH BLUE GAIN	–	R	Bit[7:0]: Auto Calculated WBMATCH blue gain[7:0]
0x505C	REAL USED RED GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Real used red gain[11:8] in master MWB gain module
0x505D	REAL USED RED GAIN	–	R	Bit[7:0]: Real used red gain[7:0] in master MWB gain module
0x505E	REAL USED GREEN GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Real used green gain[11:8] in master MWB gain module

table 6-20 ISP control registers (sheet 10 of 10)

address	register name	default value	R/W	description
0x505F	REAL USED GREEN GAIN	–	R	Bit[7:0]: Real used green gain[7:0] in master MWB gain module
0x5060	REAL USED BLUE GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Real used blue gain[11:8] in master MWB gain module
0x5061	REAL USED BLUE GAIN	–	R	Bit[7:0]: Real used blue gain[7:0] in master MWB gain module
0x5062~ 0x5063	REAL USED CLEAR GAIN	–	R	Bit[7:4]: Not used Bit[3:0]: Real used clear gain[11:8] in master MWB gain module
0x5063	REAL USED CLEAR GAIN	–	R	Bit[7:0]: Real used clear gain[7:0] in master MWB gain module
0x5064	MASTER AVG	–	R	Bit[7:0]: High 8 bits of master sensor's average
0x5065	SLAVE AVG	–	R	Bit[7:0]: High 8 bits of slave sensor's average

6.21 AWB_M control [0x5180 ~ 0x51AB]

table 6-21 AWB_M control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x5180	AWBM CTRL00	0x61	RW	Bit[7:6]: step_local Bit[5:4]: step_fast Bit[3]: slop_8x Bit[2]: slop_4x Bit[1]: Debug mode (manual enable) Bit[0]: avg_all
0x5181	AWBM CTRL01	0x11	RW	Bit[7:4]: max_local_cnt Bit[3:0]: max_fast_cnt
0x5182	AWBM CTRL02	0x41	RW	Bit[7]: freeze enable Bit[6]: fast_enable Bit[5:4]: win_idx Bit[3:1]: min_white_pixel_sel Bit[0]: bias_stat
0x5183	AWBM CTRL03	0x42	RW	Bit[7:4]: Unstable stable range Bit[3:0]: Stable range
0x5184	AWBM CTRL04	0x20	RW	Bit[7:0]: awb_x0(cwf_x) X position of center point in cwf

table 6-21 AWB_M control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x5185	AWBM CTRL05	0x20	RW	Bit[7:0]: awb_y0(cwf_y) Y position of center point in cwf
0x5186	AWBM CTRL06	0x00	RW	Bit[7:0]: awb_kx Slope of the connection line between the center point of A zone and cwf zone
0x5187	AWBM CTRL07	0x00	RW	Bit[7:0]: awb_ky Slope of the connection line between the center point of day zone and cwf zone
0x5188	AWBM CTRL08	0x10	RW	Bit[7:0]: Range of cwf zone
0x5189	AWBM CTRL09	0x10	RW	Bit[7:0]: Range of A zone
0x518A	AWBM CTRL0A	0x10	RW	Bit[7:0]: Range of day zone
0x518B	AWBM CTRL0B	0x00	RW	Bit[7:0]: Left boundary of day zone
0x518C	AWBM CTRL0C	0x00	RW	Bit[7:0]: Bottom boundary of A zone
0x518D	AWBM CTRL0D	0xF0	RW	Bit[7:0]: Top threshold of input data If input pixel is larger than it, input pixel is ignored
0x518E	AWBM CTRL0E	0x10	RW	Bit[7:0]: Bottom threshold of input data If input pixel is less than it, input pixel is ignored
0x518F	AWBM CTRL0F	0xFF	RW	Bit[7:0]: Gain max value for R channel
0x5190	AWBM CTRL10	0x00	RW	Bit[7:0]: Gain min value for R channel
0x5191	AWBM CTRL11	0xFF	RW	Bit[7:0]: Gain max value for G channel
0x5192	AWBM CTRL12	0x00	RW	Bit[7:0]: Gain min value for G channel
0x5193	AWBM CTRL13	0xFF	RW	Bit[7:0]: Gain max value for B channel
0x5194	AWBM CTRL14	0x00	RW	Bit[7:0]: Gain min value for B channel
0x5195	AWBM CTRL15	0x04	RW	Bit[7:4]: Not used Bit[3:0]: awb_r_gain_m[11:8]
0x5196	AWBM CTRL16	0x00	RW	Bit[7:0]: awb_r_gain_m[7:0]
0x5197	AWBM CTRL17	0x04	RW	Bit[7:4]: Not used Bit[3:0]: awb_g_gain_m[11:8]
0x5198	AWBM CTRL18	0x00	RW	Bit[7:0]: awb_g_gain_m[7:0].
0x5199	AWBM CTRL19	0x04	RW	Bit[7:4]: Not used Bit[3:0]: awb_b_gain_m[11:8]
0x519A	AWBM CTRL1A	0x00	RW	Bit[7:0]: awb_b_gain_m[7:0]

table 6-21 AWB_M control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x519B	AWBM CTRL1B	0x04	RW	Bit[7:1]: Not used Bit[0]: local_limit_sel
0x51A0	AWBM CTRL20	–	R	Bit[7:0]: ar_i[9:2]
0x51A1	AWBM CTRL21	–	R	Bit[7:2]: Not used Bit[1:0]: ar_i[1:0]
0x51A2	AWBM CTRL22	–	R	Bit[7:0]: ag_i[9:2]
0x51A3	AWBM CTRL23	–	R	Bit[7:2]: Not used Bit[1:0]: ag_i[1:0]
0x51A4	AWBM CTRL24	–	R	Bit[7:0]: ab_i[9:2]
0x51A5	AWBM CTRL25	–	R	Bit[7:2]: Not used Bit[1:0]: ab_i[1:0]
0x51A6	AWBM CTRL26	–	R	Bit[7:4]: Not used Bit[3:0]: r_center_i[11:8]
0x51A7	AWBM CTRL27	–	R	Bit[7:0]: r_center_i[7:0]
0x51A8	AWBM CTRL28	–	R	Bit[7:4]: Not used Bit[3:0]: g_center_i[11:8]
0x51A9	AWBM CTRL29	–	R	Bit[7:0]: g_center_i[7:0]
0x51AA	AWBM CTRL2A	–	R	Bit[7:5]: Not used Bit[4:0]: b_center_i[11:8]
0x51AB	AWBM CTRL2B	–	R	Bit[7:0]: b_center_i[7:0]

6.22 AWB_S control [0x5200 - 0x522B]

table 6-22 AWB_S control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x5200	AWBS CTRL00	0x61	RW	Bit[7:6]: step_local Bit[5:4]: step_fast Bit[3]: slop_8x Bit[2]: slop_4x Bit[1]: Debug mode (manual enable) Bit[0]: avg_all
0x5201	AWBS CTRL01	0x11	RW	Bit[7:4]: max_local_cnt Bit[3:0]: max_fast_cnt

table 6-22 AWB_S control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x5202	AWBS CTRL02	0x41	RW	Bit[7]: freeze enable Bit[6]: fast_enable Bit[5:4]: win_idx Bit[3:1]: min_white_pixel_sel Bit[0]: bias_stat
0x5203	AWBS CTRL03	0x42	RW	Bit[7:4]: Unstable stable range Bit[3:0]: Stable range
0x5204	AWBS CTRL04	0x20	RW	Bit[7:0]: awb_x0 (cwf_x) X position of center point in cwf
0x5205	AWBS CTRL05	0x20	RW	Bit[7:0]: awb_y0 (cwf_y) Y position of center point in cwf
0x5206	AWBS CTRL06	0x00	RW	Bit[7:0]: awb_kx Slope of the connection line between the center point of A zone and cwf zone
0x5207	AWBS CTRL07	0x00	RW	Bit[7:0]: awb_ky Slope of the connection line between the center point of day zone and cwf zone
0x5208	AWBS CTRL08	0x10	RW	Bit[7:0]: Range of cwf zone
0x5209	AWBS CTRL09	0x10	RW	Bit[7:0]: Range of A zone
0x520A	AWBS CTRL0A	0x10	RW	Bit[7:0]: Range of day zone
0x520B	AWBS CTRL0B	0x00	RW	Bit[7:0]: Left boundary of day zone
0x520C	AWBS CTRL0C	0x00	RW	Bit[7:0]: Bottom boundary of A zone
0x520D	AWBS CTRL0D	0xF0	RW	Bit[7:0]: Top threshold of input data If input pixel is larger than it, input pixel is ignored
0x520E	AWBS CTRL0E	0x10	RW	Bit[7:0]: Bottom threshold of input data If input pixel is less than it, input pixel is ignored
0x520F	AWBS CTRL0F	0xFF	RW	Bit[7:0]: Gain max value for R channel
0x5210	AWBS CTRL10	0x00	RW	Bit[7:0]: Gain min value for R channel
0x5211	AWBS CTRL11	0xFF	RW	Bit[7:0]: Gain max value for G channel
0x5212	AWBS CTRL12	0x00	RW	Bit[7:0]: Gain min value for G channel
0x5213	AWBS CTRL13	0xFF	RW	Bit[7:0]: Gain max value for B channel
0x5214	AWBS CTRL14	0x00	RW	Bit[7:0]: Gain min value for B channel
0x5215	AWBS CTRL15	0x04	RW	Bit[7:4]: Not used Bit[3:0]: awb_r_gain_m[11:8]

table 6-22 AWB_S control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x5216	AWBS CTRL16	0x00	RW	Bit[7:0]: awb_r_gain_m[7:0]
0x5217	AWBS CTRL17	0x04	RW	Bit[7:4]: Not used Bit[3:0]: awb_g_gain_m[11:8]
0x5218	AWBS CTRL18	0x00	RW	Bit[7:0]: awb_g_gain_m[7:0]
0x5219	AWBS CTRL19	0x04	RW	Bit[7:4]: Not used Bit[3:0]: awb_b_gain_m[11:8]
0x521A	AWBS CTRL1A	0x00	RW	Bit[7:0]: awb_b_gain_m[7:0]
0x521B	AWBS CTRL1B	0x04	RW	Bit[7:1]: Not used Bit[0]: local_limit_sel
0x5220	AWBS CTRL20	–	R	Bit[7:0]: ar_i[9:2]
0x5221	AWBS CTRL21	–	R	Bit[7:2]: Not used Bit[1:0]: ar_i[1:0]
0x5222	AWBS CTRL22	–	R	Bit[7:0]: ag_i[9:2]
0x5223	AWBS CTRL23	–	R	Bit[7:2]: Not used Bit[1:0]: ag_i[1:0]
0x5224	AWBS CTRL24	–	R	Bit[7:0]: ab_i[9:2]
0x5225	AWBS CTRL25	–	R	Bit[7:2]: Not used Bit[1:0]: ab_i[1:0]
0x5226	AWBS CTRL26	–	R	Bit[7:4]: Not used Bit[3:0]: r_center_i[11:8]
0x5227	AWBS CTRL27	–	R	Bit[7:0]: r_center_i[7:0]
0x5228	AWBS CTRL28	–	R	Bit[7:4]: Not used Bit[3:0]: g_center_i[11:8]
0x5229	AWBS CTRL29	–	R	Bit[7:0]: g_center_i[7:0]
0x522A	AWBS CTRL2A	–	R	Bit[7:4]: Not used Bit[3:0]: b_center_i[11:8]
0x522B	AWBS CTRL2B	–	R	Bit[7:0]: b_center_i[7:0]

6.23 hscale control [0x5600 ~ 0x5603]

table 6-23 hscale control registers

address	register name	default value	R/W	description
0x5600	SCALE CTRL00	0x00	RW	Bit[7:4]: Not used Bit[3]: scale_manual_en For manual mode enable Bit[2]: scale_gain_2x_en, For 2x gain enable Bit[1:0]: scale_mode For scale mode select 00: 0.6x 01: 0.5x 10: 0.4x 11: 1/3x
0x5602	SCALE WIDTH	0x01	RW	Bit[7:5]: Not used Bit[4:0]: output_width[12:8] for manual mode
0x5603	SCALE WIDTH	0x80	RW	Bit[7:0]: output_width[7:0] for manual mode

6.24 AVG control [0x5680 ~ 0x568A]

table 6-24 AVG control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5680	AVG CTRL00	0x00	RW	Bit[7:5]: Not used Bit[4:0]: x_start_avg[12:8] AVG sub-window horizontal start position high byte
0x5681	AVG CTRL01	0x00	RW	Bit[7:0]: x_start_avg[7:0] AVG sub-window horizontal start position low byte
0x5682	AVG CTRL02	0x00	RW	Bit[7:4]: Not used Bit[3:0]: y_start_avg[11:8] AVG sub-window vertical start position high byte
0x5683	AVG CTRL03	0x00	RW	Bit[7:0]: y_start_avg[7:0] AVG sub-window vertical start position low byte

table 6-24 AVG control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x5684	AVG CTRL04	0x0C	RW	Bit[7:5]: Not used Bit[4:0]: window_width_avg[12:8] Sub-window width high byte
0x5685	AVG CTRL05	0xC0	RW	Bit[7:0]: window_width_avg[7:0] Sub-window width low byte
0x5686	AVG CTRL06	0x09	RW	Bit[7:4]: Not used Bit[3:0]: window_height_avg[11:8] Sub-window height high byte
0x5687	AVG CTRL07	0x90	RW	Bit[7:0]: window_height_avg[7:0] Sub-window height low byte
0x5688	AVG CTRL08	0x02	RW	Bit[7:2]: Not used Bit[1]: Sum option 0: Sum = (4×B+9×G×2+10×R)/8 1: Sum = B+G×2+R Bit[0]: Sub-window function enable 0: Use whole output window for average 1: Use registers 0x5680~0x5687 to define window for average
0x5689	AVG RO09	–	R	Bit[7:1]: Not used Bit[0]: Average calculated indicating signal for SCCB read
0x568A	AVG RO0A	–	R	Bit[7:0]: High 8 bits of whole image's average output

6.25 DPC control [0x5780 - 0x57A7]

table 6-25 DPC control registers

address	register name	default value	R/W	description
0x5780~ 0x57A7	DPC CTRL	–	–	DPC Control Registers

6.26 LENC control [0x5800 - 0x587C]

table 6-26 LENC control registers (sheet 1 of 5)

address	register name	default value	R/W	description
0x5800	LENC G00	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G00 for luminance compensation
0x5801	LENC G01	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G01 for luminance compensation
0x5802	LENC G02	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G02 for luminance compensation
0x5803	LENC G03	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G03 for luminance compensation
0x5804	LENC G04	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G04 for luminance compensation
0x5805	LENC G05	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G05 for luminance compensation
0x5806	LENC G10	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G10 for luminance compensation
0x5807	LENC G11	0x08	RW	Bit[7:6]: Not used Bit[5:0]: Control point G11 for luminance compensation
0x5808	LENC G12	0x08	RW	Bit[7:6]: Not used Bit[5:0]: Control point G12 for luminance compensation
0x5809~ 0x5822	LENC G13~ LENC G54	–	RW	Bit[7:6]: Not used Bit[5:0]: Control point G13~G54 for luminance compensation
0x5823	LENC G55	0x10	RW	Bit[7:6]: Not used Bit[5:0]: Control point G55 for luminance compensation
0x5824	LENC B00	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point B00 for blue channel compensation

table 6-26 LENC control registers (sheet 2 of 5)

address	register name	default value	R/W	description
0x5825	LENC B01	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point B01 for blue channel compensation
0x5826	LENC B02	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point B02 for blue channel compensation
0x5827	LENC B03	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point B03 for blue channel compensation
0x5828	LENC B04	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point B04 for blue channel compensation
0x5829	LENC B05	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point B05 for blue channel compensation
0x582A	LENC B10	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point B10 for blue channel compensation
0x582B	LENC B11	0x12	RW	Bit[7:6]: Not used Bit[5:0]: Control point B11 for blue channel compensation
0x582C	LENC B12	0x12	RW	Bit[7:6]: Not used Bit[5:0]: Control point B12 for blue channel compensation
0x582D~ 0x5846	LENC B13~LENC B54	–	RW	Bit[7:6]: Not used Bit[5:0]: Control point B13~B54 for blue channel compensation
0x5847	LENC R00	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point R00 for red channel compensation
0x5849	LENC R01	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point R01 for red channel compensation
0x584A	LENC R02	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point R02 for red channel compensation
0x584B	LENC R03	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point R03 for red channel compensation

table 6-26 LENC control registers (sheet 3 of 5)

address	register name	default value	R/W	description
0x584C	LENC R04	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point R04 for red channel compensation
0x584D	LENC R05	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point R05 for red channel compensation
0x584E	LENC R10	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point R10 for red channel compensation
0x584F	LENC R11	0x12	RW	Bit[7:6]: Not used Bit[5:0]: Control point R11 for red channel compensation
0x5850	LENC R12	0x12	RW	Bit[7:6]: Not used Bit[5:0]: Control point R12 for red channel compensation
0x5851~ 0x586A	LENC R13~LENC R54	–	RW	Bit[7:6]: Not used Bit[5:0]: Control point R13~R54 for red channel compensation
0x586B	LENC R55	0x14	RW	Bit[7:6]: Not used Bit[5:0]: Control point R55 for red channel compensation
0x586C	LENC BOFFSET	0x30	RW	Bit[7]: Not used Bit[6:0]: Base value for all blue channel control points
0x586D	LENC ROFFSET	0x30	RW	Bit[7]: Not used Bit[6:0]: Base value for all red channel control points
0x586E	LENC MAXGAIN	0x40	RW	Bit[7:0]: If AutoLensSwitchEnable is true and sensor gain is larger than this threshold, luminance compensation amplitude will be the minimum value (min LENC gain). Register value is 16 times sensor gain
0x586F	LENC MINGAIN	0x20	RW	Bit[7:0]: If AutoLensSwitchEnable is true and sensor gain is larger than this threshold, luminance compensation amplitude will start to decrease; otherwise, the amplitude will not change. Register value is 16 times sensor gain.

table 6-26 LENC control registers (sheet 4 of 5)

address	register name	default value	R/W	description
0x5870	LENC MINQ	0x18	RW	Bit[7]: Not used Bit[6:0]: This value indicates the minimum amplitude which luminance channel compensates when AutoLensSwitchEnable is true. Value should be in the range [0~64]
0x5871	LENC CTRL	0x0D	RW	Bit[7:4]: Not used Bit[3]: Add BLC target after applying compensation Bit[2]: Enable BLC target for LENC 0: Disable BLC target 1: Enable BLC target Bit[1]: Not used Bit[0]: AutoLensSwitchEnable 0: Luminance compensation amplitude does not change with sensor gain 1: Luminance compensation amplitude changes with sensor gain
0x5872	LENC HSCALE	0x01	RW	Bit[7:3]: Not used Bit[2:0]: HScale[10:8] For horizontal gain calculation, this value indicates the step between two connected horizontal pixels, where hscale = 4×2^18 / image width
0x5873	LENC HSCALE	0x3E	RW	Bit[7:0]: HScale[7:0]
0x5874	LENC VSCALE	0x00	RW	Bit[7:3]: Not used Bit[2:0]: VScale[10:8] For vertical gain calculation, this value indicates the step between two connected vertical pixels, where vscale = 4×2^17 / image height
0x5875	LENC VSCALE	0xD3	RW	Bit[7:0]: VScale[7:0]
0x5876	LENC XOFFSET	–	R	Bit[7:4]: Not used Bit[3:0]: Input sensor horizontal offset[11:8]
0x5877	LENC XOFFSET	–	R	Bit[7:0]: Input sensor horizontal offset[7:0]
0x5878	LENC YOFFSET	–	R	Bit[7:4]: Not used Bit[3:0]: Input sensor vertical offset[11:8]
0x5879	LENC YOFFSET	–	R	Bit[7:0]: Input sensor vertical offset[7:0]

table 6-26 LENC control registers (sheet 5 of 5)

address	register name	default value	R/W	description
0x587A	LENC INPUT	–	R	Bit[7:6]: Not used Bit[5]: Input sensor flip Bit[4]: Input sensor mirror Bit[3:2]: Input sensor Y skip Bit[1:0]: Input sensor X skip
0x587B	LENC OVERFLOW	–	R	Bit[7:4]: Not used Bit[3]: GH overflow for debug Bit[2]: BRH overflow for debug Bit[1]: GV overflow for debug Bit[0]: BRV overflow for debug
0x587C	LENC QVALUE	–	R	Bit[7]: Not used Bit[6:0]: Real amplitude Q value

6.27 VAP [0x5900 ~ 0x5901]

table 6-27 VAP control registers

address	register name	default value	R/W	description
0x5900	VAP CTRL00	0x01	RW	Bit[7]: R channel average enable Bit[6]: Gr channel average enable Bit[5]: Gb channel average enable Bit[4]: B channel average enable Bit[3]: Debug mode enable Bit[2]: Single channel enable Bit[1:0]: Add option 00: sum mode 01: average mode 1x: drop mode
0x5901	VAP CTRL01	0x00	RW	Bit[7:4]: Not used Bit[3:2]: Hskip (only support 1:2, hskip=1) Bit[1:0]: Vskip (only support 1:2, vskip=1)

6.28 WINC [0x5A00 - 0x5A0C]

table 6-28 WINC control registers

address	register name	default value	R/W	description
0x5A00	WINC CTRL00	0x00	RW	Bit[7:4]: Not used Bit[3:0]: x_start_offset[11:8] Start address in horizontal
0x5A01	WINC CTRL01	0x00	RW	Bit[7:0]: x_start_offset[7:0]
0x5A02	WINC CTRL02	0x00	RW	Bit[7:4]: Not used Bit[3:0]: y_start_offset[11:8] Start address in vertical
0x5A03	WINC CTRL03	0x00	RW	Bit[7:0]: y_start_offset[7:0]
0x5A04	WINC CTRL04	0x0C	RW	Bit[7:4]: Not used Bit[3:0]: window_width[11:8] Select whole zone width high byte
0x5A05	WINC CTRL05	0xE0	RW	Bit[7:0]: window_width[7:0] Select whole zone width low byte
0x5A06	WINC CTRL06	0x09	RW	Bit[7:4]: Not used Bit[3:0]: window_height[11:8] Select whole zone height high byte
0x5A07	WINC CTRL07	0xB0	RW	Bit[7:0]: window_height[7:0] Select whole zone height low byte
0x5A08	WINC CTRL08	0x06	RW	Bit[7:4]: Reserved Bit[3]: Window valid select option (for debug) 0: Select new valid_1d 1: Select original valid_1d Bit[2]: Select embed line flag 0: Select first line as embedded flag 1: Select last line as embedded flag Bit[1]: Window enable option 0: Disable after last valid line 1: Original enable signal from register Bit[0]: Manual window enable 0: Window size from window top 1: Window size from 0x5A00 to 0x5A07
0x5A09	WINC RO09	–	R	Bit[7:4]: Not used Bit[3:0]: Pixel count[11:8] for debug
0x5A0A	WINC RO0A	–	R	Bit[7:0]: Pixel count[7:0] for debug
0x5A0B	WINC RO0B	–	R	Bit[7:4]: Not used Bit[3:0]: Line count[11:8] for debug
0x5A0C	WINC RO0C	–	R	Bit[7:0]: Line count[7:0] for debug

6.29 OTP DPC control [0x5B00 - 0x5B23]

table 6-29 OTP DPC registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x5B00	OTP CTRL00	0x00	RW	Bit[7:2]: Not used Bit[1:0]: Memory start address[9:8]
0x5B01	OTP CTRL01	0x00	RW	Bit[7:0]: Memory start address[7:0]
0x5B02	OTP CTRL02	0x01	RW	Bit[7:2]: Not used Bit[1:0]: Memory end address[9:8]
0x5B03	OTP CTRL03	0xFF	RW	Bit[7:0]: Memory end address[7:0]
0x5B04	OTP CTRL04	0x42	RW	Bit[7]: Select xy_end signal for debug 0: xy_end keep 0 1: xy_end keep 1 after last cluster is read out Bit[6]: VSYNC reset enable for debug 0: Do not use VSYNC to reset 3 enable signals 1: Use VSYNC to reset 3 enable signals to fix a bug Bit[5]: Threshold function enable 0: Disable the recover threshold in register 0x5B09 (can recover black cluster) 1: Enable the recover threshold in register 0x5B09 (can not recover black cluster) Bit[4]: Manual increase step enable Bit[3]: Disable mirror and flip Bit[2]: Disable OTP offset Bit[1]: Mirror option enable Bit[0]: Disable binning mode

table 6-29 OTP DPC registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x5B05	OTP CTRL05	0x6C	RW	Bit[7]: Not used Bit[6:5]: Recover method select 00: Left 1 neighbor pixel (on same channel) 01: Minimum of left 2 neighbor pixels 10: Average of left and right 1 neighbor pixel 11: Maximum between the minimum of left 2 neighbor pixels and the minimum of right 2 neighbor pixels Bit[4]: Use fixed pattern to recover cluster Bit[3]: Fixed pattern mode 0: Use 0x00 to recover cluster 1: Use 0x3FF to recover cluster Bit[2]: Flip option enable Bit[1]: Sensor exposure constrain enable Bit[0]: Sensor gain constrain enable
0x5B06	OTP CTRL06	0x00	RW	Bit[7]: Not used Bit[6:5]: Constrain exposure threshold[9:8] Bit[4:0]: Not used
0x5B07	OTP CTRL07	0x00	RW	Bit[7:0]: Constrain exposure threshold[7:0] (disable OTP function when the sensor exposure is smaller than the constrain exposure threshold)
0x5B08	OTP CTRL08	0x07	RW	Bit[7:6]: Not used Bit[5:0]: Constrain gain threshold (disable OTP function when the sensor gain is smaller than the constrain gain threshold)
0x5B09	OTP CTRL09	0x08	RW	Bit[7:4]: Not used Bit[3:0]: Recover threshold (recover when the high 8-bits of the recovered data is bigger than the original one by this threshold)
0x5B0A	OTP CTRL0A	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Manual horizontal even increase step
0x5B0B	OTP CTRL0B	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Manual horizontal odd increase step
0x5B0C	OTP CTRL0C	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Manual vertical even increase step
0x5B0D	OTP CTRL0D	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Manual vertical odd increase step

table 6-29 OTP DPC registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x5B10	OTP RO10	–	R	Bit[7:4]: Not used Bit[3:0]: Horizontal offset[11:8]
0x5B11	OTP RO11	–	R	Bit[7:0]: Horizontal offset[7:0]
0x5B12	OTP RO12	–	R	Bit[7:4]: Not used Bit[3:0]: Vertical offset[11:8]
0x5B13	OTP RO13	–	R	Bit[7:0]: Vertical offset[7:0]
0x5B14	OTP RO14	–	R	Bit[7:5]: Not used Bit[4:0]: Horizontal even increase step
0x5B15	OTP RO15	–	R	Bit[7:5]: Not used Bit[4:0]: Horizontal odd increase step
0x5B16	OTP RO16	–	R	Bit[7:5]: Not used Bit[4:0]: Vertical even increase step
0x5B17	OTP RO17	–	R	Bit[7:5]: Not used Bit[4:0]: Vertical odd increase step
0x5B18~ 0x5B1F	NOT USED	–	–	Not Used
0x5B20	OTP CTRL20	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Manual X offset[11:8]
0x5B21	OTP CTRL21	0x00	RW	Bit[7:0]: Manual X offset[7:0]
0x5B22	OTP CTRL22	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Manual Y offset[11:8]
0x5B23	OTP CTRL23	0x00	RW	Bit[7:0]: Manual Y offset[7:0]

6.30 pre_DSP control [0x5E00 - 0x5E2E]

table 6-30 pre_DSP control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x5E00	PRE CTRL00	0x00	RW	Bit[7]: Test pattern enable Bit[6]: Rolling bar function enable Bit[5]: Transparent enable Bit[4]: Square mode 0: Color square 1: Black-white square Bit[3:2]: Color bar style 00: Standard color bar 01: Top-bottom darker color bar 10: Right-left darker color bar 11: Bottom-top darker color bar Bit[1:0]: Test pattern mode 00: Color bar 01: Random data 10: Square 11: Black image
0x5E01	PRE CTRL01	0x41	RW	Bit[7]: Reserved Bit[6]: Window cut enable Bit[5]: two_lsb_0_en When set, two LSBs of output data are 0 Bit[4]: Same seed enable When set, the seed used to generate the random data are same which is set in seed register Bit[3:0]: Random seed Seed used in generating random data
0x5E02	PRE CTRL02	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Line number interrupt[11:8]
0x5E03	PRE CTRL03	0x01	RW	Bit[7:0]: Line number interrupt[7:0]
0x5E04~ 0x5E07	RSVD	–	–	Reserved
0x5E08	PRE CTRL08	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Horizontal manual offset[11:8]]
0x5E09	PRE CTRL09	0x00	RW	Bit[7:0]: Horizontal manual offset[7:0]
0x5E0A	PRE CTRL0A	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Vertical manual offset[11:8]
0x5E0B	PRE CTRL0B	0x00	RW	Bit[7:0]: Vertical manual offset[7:0]
0x5E0C	PRE RO0C	–	R	Bit[7:4]: Reserved Bit[3:0]: Input image pixel number[11:8]

table 6-30 pre_DSP control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x5E0D	PRE RO0D	–	R	Bit[7:0]: Input image pixel number[7:0]
0x5E0E	PRE RO0E	–	R	Bit[7:4]: Reserved Bit[3:0]: Input image line number[11:8]
0x5E0F	PRE RO0F	–	R	Bit[7:0]: Input image line number[7:0]
0x5E10	PRE CTRL10	0x3C	RW	Bit[7]: Window X offset option Bit[6]: Window Y offset option Bit[5]: Take the first pixel in the same position with no mirror image enable Bit[4]: Take the first pixel in the same position with no flip image enable Bit[3]: Mirror option from window 0: First pixel is Gb or R with window output 1: First pixel is B or Gr with window output Bit[2]: Flip option from window 0: First line is GR with window output 1: First line is BG with window output Bit[1]: Offset manual enable Bit[0]: Reserved
0x5E11	PRE CTRL11	0x00	RW	Bit[7]: Manual clock/valid ratio enable Bit[6:4]: Manual dummy line number Bit[3]: Reduce HREF low length by half Bit[2:0]: Manual clock/valid ratio for dummy line
0x5E12	PRE RO12	–	R	Bit[7:0]: HREF blank length for dummy line[15:8]
0x5E13	PRE RO13	–	R	Bit[7:0]: HREF blank length for dummy line[7:0]
0x5E14	PRE RO14	–	R	Bit[7:0]: HREF length for dummy line[15:8]
0x5E15	PRE RO15	–	R	Bit[7:0]: HREF length for dummy line[7:0]
0x5E16	PRE RO16	–	R	Bit[7:5]: Reserved Bit[4]: Dummy error indicating signal Bit[3]: Reserved Bit[2:0]: Dummy line clock ratio output
0x5E17	PRE RO17	–	R	Bit[7:4]: Horizontal odd increase step Bit[3:0]: Vertical odd increase step
0x5E18	PRE RO18	–	R	Bit[7:4]: Reserved Bit[3:0]: Horizontal sensor offset[11:8]
0x5E19	PRE RO19	–	R	Bit[7:0]: Horizontal sensor offset[7:0]
0x5E1A	PRE RO1A	–	R	Bit[7:4]: Reserved Bit[3:0]: Vertical sensor offset[11:8]
0x5E1B	PRE RO1B	–	R	Bit[7:0]: Vertical sensor offset[7:0]

table 6-30 pre_DSP control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x5E1C	PRE RO1C	–	R	Bit[7:4]: Reserved Bit[3:0]: Horizontal window offset[11:8]
0x5E1D	PRE RO1D	–	R	Bit[7:0]: Horizontal window offset[7:0]
0x5E1E	PRE RO1E	–	R	Bit[7:4]: Reserved Bit[3:0]: Vertical window offset[11:8]
0x5E1F	PRE RO1F	–	R	Bit[7:0]: Vertical window offset[7:0]
0x5E20	PRE RO20	–	R	Bit[7:5]: Reserved Bit[4:0]: Horizontal window output size[12:8]
0x5E21	PRE RO21	–	R	Bit[7:0]: Horizontal window output size[7:0]
0x5E22	PRE RO22	–	R	Bit[7:4]: Reserved Bit[3:0]: Vertical window output size[11:8]
0x5E23	PRE RO23	–	R	Bit[7:0]: Vertical window output size[7:0]
0x5E24	PRE RO24	–	R	Bit[7:6]: Reserved Bit[5:4]: Horizontal skip Bit[3:2]: Reserved Bit[1:0]: Vertical skip
0x5E25	PRE RO25	–	R	Bit[7:4]: Horizontal even increase step Bit[3:0]: Vertical even increase step
0x5E26	NOT USED	–	–	Not Used
0x5E27	PRE RO27	–	R	Bit[7:4]: Reserved Bit[3:0]: Cut top offset for bi-linear BLC[11:8]
0x5E28	PRE RO28	–	R	Bit[7:0]: Cut top offset for bi-linear BLC[7:0]
0x5E29	PRE RO29	–	R	Bit[7:4]: Reserved Bit[3:0]: Cut bottom offset for bi-linear BLC[11:8]
0x5E2A	PRE RO2A	–	R	Bit[7:0]: Cut bottom offset for bi-linear BLC[7:0]
0x5E2B	PRE CTRL2B	0x09	RW	Bit[7:4]: Reserved Bit[3:0]: Array height for bi-linear BLC[11:8]
0x5E2C	PRE CTRL2C	0xB0	RW	Bit[7:0]: Array height for bi-linear BLC[7:0]
0x5E2D	PRE CTRL2D	0x00	RW	Bit[7:6]: Reserved Bit[5]: Manual horizontal skip enable Bit[4:0]: Manual horizontal skip for RGBC pattern
0x5E2E	PRE CTRL2E	0x00	RW	Bit[7:6]: Reserved Bit[5]: Manual vertical skip enable Bit[4:0]: Manual vertical skip for RGBC pattern

OV8858

color CMOS 8 megapixel (3264 x 2448) image sensor with OmniBSI-3™ technology

7 operating specifications

7.1 absolute maximum ratings

table 7-1 absolute maximum ratings

parameter		absolute maximum rating[a]
ambient storage temperature		-40°C to +125°C
supply voltage (with respect to ground)	V_{DD-A}	4.5V
	V_{DD-D}	3V
	V_{DD-IO}	4.5V
electro-static discharge (ESD)	human body model	2000V
	machine model	200V
all input/output voltages (with respect to ground)		-0.3V to V_{DD-IO} + 1V
I/O current on any input or output pin		± 200 mA

a. exceeding the absolute maximum ratings shown above invalidates all AC and DC electrical specifications and may result in permanent damage to the device. Exposure to absolute maximum rated conditions for extended periods may affect device reliability.

7.2 functional temperature

table 7-2 functional temperature

parameter	range
operating temperature (for applications up to 90 fps)[a]	-30°C to +85°C junction temperature
stable image temperature[b]	0°C to +60°C junction temperature

a. sensor functions but image quality may be noticeably different at temperatures outside of stable image range

b. image quality remains stable throughout this temperature range

7.3 DC characteristics

table 7-3 DC characteristics (-30°C < T_J < 85°C)

symbol	parameter	min	typ	max[a]	unit
supply					
$V_{DD\text{-}A}$	supply voltage (analog)	2.6	2.8	3.0	V
$V_{DD\text{-}D}$	supply voltage (digital core for 4-lane MIPI up to 1000 Mbps/lane)	1.1	1.2	1.3	V
$V_{DD\text{-}IO}$	supply voltage (digital I/O)	1.7	1.8	3.0	V
$I_{DD\text{-}A}$	active (operating) current[a]		TBD	TBD	mA
$I_{DD\text{-}IO}$			TBD	TBD	mA
$I_{DD\text{-}D}$			TBD	TBD	mA
$I_{DDS\text{-}SCCB}$	standby current[b]		TBD	TBD	μA
$I_{DDS\text{-}PWDN}$			TBD	TBD	μA
$I_{DDS\text{-}XSHUTDN}$			TBD	TBD	μA
digital inputs (typical conditions: AVDD = 2.8V, DVDD = 1.2V, DOVDD = 1.8V)					
V_{IL}	input voltage LOW			0.54	V
V_{IH}	input voltage HIGH	1.26			V
C_{IN}	input capacitor			10	pF
digital outputs (standard loading 25 pF)					
V_{OH}	output voltage HIGH	1.62			V
V_{OL}	output voltage LOW			0.18	V
serial interface inputs					
V_{IL}[c]	SIOC and SIOD	-0.5	0	0.54	V
V_{IH}	SIOC and SIOD	1.28	1.8	3.0	V

a. maximum active current is measured under typical supply voltage

a. DVDD is provided by external regulator for lower power consumption. DVDD and EVDD are tied together. DOVDD = 1.8V

b. standby current is measured at room temperature with external clock off

c. based on DOVDD = 1.8V

7.4 timing characteristics

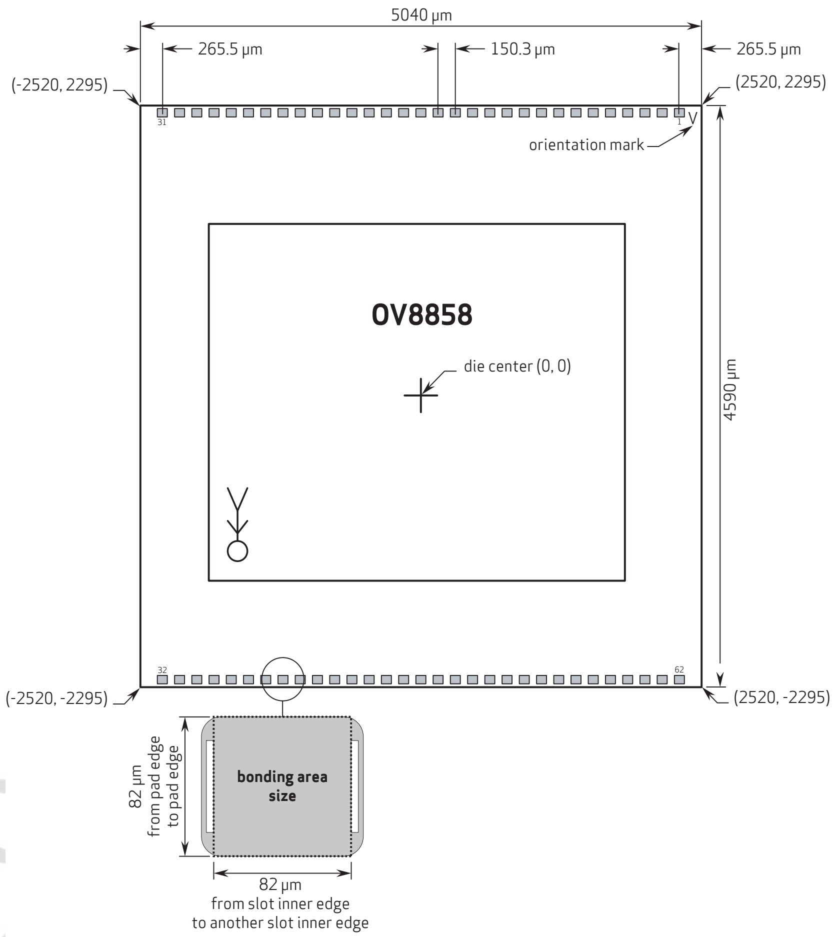
table 7-4 timing characteristics

symbol	parameter	min	typ	max	unit
oscillator and clock input					
f_{OSC}	frequency (XVCLK)	6	24	27	MHz
t_r, t_f	clock input rise/fall time			TBD	ns
	clock input duty cycle	45	50	55	%

8 mechanical specifications

8.1 COB physical specifications

figure 8-1 COB die specifications



note 1 all dimensions and coordinates are in µm unless otherwise specified

note 2 bonding outside the defined bonding area is prohibited as it may cause failure in reliability or functionality

8858_COB_DS_8_1

table 8-1 pad location coordinates (sheet 1 of 3)

pad number	pad name	x coordinate	y coordinate	bonding area size
1	DOGND	2254.50	2220.21	82x82
2	DVDD	2104.20	2220.21	82x82

table 8-1 pad location coordinates (sheet 2 of 3)

pad number	pad name	x coordinate	y coordinate	bonding area size
3	AGND	1953.90	2220.21	82x82
4	AVDD	1803.60	2220.21	82x82
5	SID	1653.30	2220.21	82x82
6	SIOC	1503.00	2220.21	82x82
7	SIOD	1352.70	2220.21	82x82
8	NC	1202.40	2220.21	82x82
9	DOGND	1052.10	2220.21	82x82
10	XVCLK	901.80	2220.21	82x82
11	VSYNC	751.50	2220.21	82x82
12	HREF	601.20	2220.21	82x82
13	DVDD	450.90	2220.21	82x82
14	STROBE	300.60	2220.21	82x82
15	ILPWM	150.30	2220.21	82x82
16	FSIN	0.00	2220.21	82x82
17	DOVDD	-150.30	2220.21	82x82
18	FREX	-300.60	2220.21	82x82
19	GPIO	-450.90	2220.21	82x82
20	XSHUTDN	-601.20	2220.21	82x82
21	DOGND	-751.50	2220.21	82x82
22	PWDNB	-901.80	2220.21	82x82
23	TM	-1052.10	2220.21	82x82
24	DVDD	-1202.40	2220.21	82x82
25	DOGND	-1352.70	2220.21	82x82
26	DOGND	-1503.00	2220.21	82x82
27	ATEST	-1653.30	2220.21	82x82
28	AVDD	-1803.60	2220.21	82x82
29	AVDD	-1953.90	2220.21	82x82
30	AGND	-2104.20	2220.21	82x82
31	AGND	-2254.50	2220.21	82x82
32	AVDD	-2254.50	-2220.21	82x82

table 8-1 pad location coordinates (sheet 3 of 3)

pad number	pad name	x coordinate	y coordinate	bonding area size
33	AGND	-2104.20	-2220.21	82x82
34	VH	-1953.90	-2220.21	82x82
35	VN1	-1803.60	-2220.21	82x82
36	VN2	-1653.30	-2220.21	82x82
37	MDP2	-1503.00	-2220.21	82x82
38	MDN2	-1352.70	-2220.21	82x82
39	MDP0	-1202.40	-2220.21	82x82
40	MDN0	-1052.10	-2220.21	82x82
41	DVDD	-901.80	-2220.21	82x82
42	DOGND	-751.50	-2220.21	82x82
43	PVDD	-601.20	-2220.21	82x82
44	DVDD	-450.90	-2220.21	82x82
45	MCP	-300.60	-2220.21	82x82
46	MCN	-150.30	-2220.21	82x82
47	DOGND	0.00	-2220.21	82x82
48	MDP1	150.30	-2220.21	82x82
49	MDN1	300.60	-2220.21	82x82
50	MDP3	450.90	-2220.21	82x82
51	MDN3	601.20	-2220.21	82x82
52	DVDD	751.50	-2220.21	82x82
53	RCP	901.80	-2220.21	82x82
54	RCN	1052.10	-2220.21	82x82
55	RDP	1202.40	-2220.21	82x82
56	RDN	1352.70	-2220.21	82x82
57	DOGND	1503.00	-2220.21	82x82
58	DOVDD	1653.30	-2220.21	82x82
59	DVDD	1803.60	-2220.21	82x82
60	DVDD	1953.90	-2220.21	82x82
61	DOGND	2104.20	-2220.21	82x82
62	DOGND	2254.50	-2220.21	82x82

8.2 reconstructed wafer (RW) physical specifications

maximum total die count:	880
film frame:	Compact Disco stainless SUS420
dicing tape:	UV tape

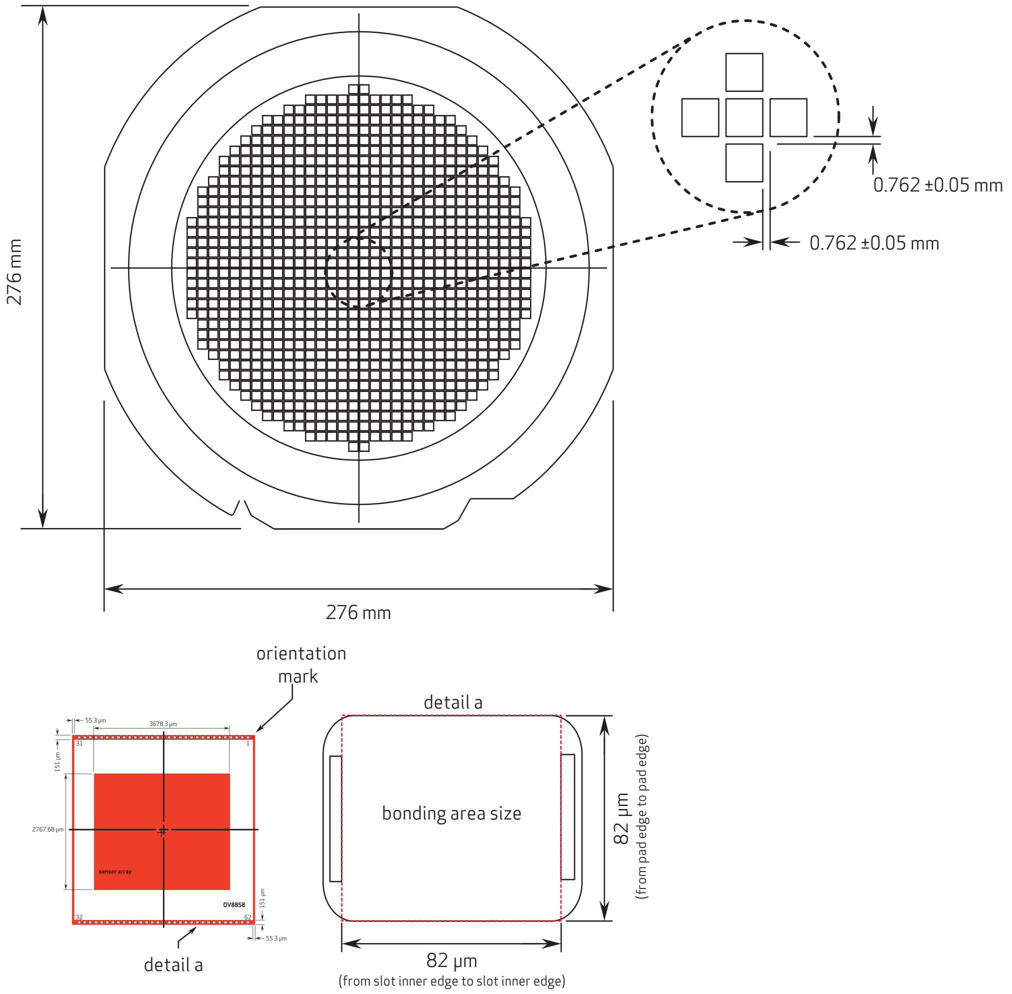
note
Actual die count varies and the absent die may be less than 10% of the maximum total die count (excluding the last frame of the wafer lot).

table 8-2 RW physical dimensions

feature	dimensions
RW physical dimensions	8" RW on 12" frame
wafer thickness (OVXXXXX-ABCD)	
C=4	200 µm ± 10 µm (7.9 mil ± 0.4 mil)
reconstructed wafer street width	0.762 mm (30 mil) ± 0.05 mm
placement accuracy x, y, theta	± 50 µm (± 2 mil), <1.0 degree
singulated die size	
width	5090 µm ± 20 µm (200.4 mil ± 0.8 mil)
length	4640 µm ± 20 µm (182.7 mil ± 0.8 mil)
bond pad size	96 µm × 82 µm (3.8 mil × 3.2 mil)
minimum bond pad pitch	150.3 µm (5.9 mil)
bonding area size	82 µm × 82 µm (3.2 mil × 3.2 mil)
optical array	
die center	(0, 0)
optical center from die center[a]	-27 µm, -54 µm (-1.1 mil, -2.1 mil)

[a] based on die orientation on frame with notch facing down position

figure 8-2 OV8858 RW physical diagram



note 1 bonding outside the defined bonding area is prohibited, it may potentially induce reliablity issues or functionality failure

note 2 keep-out-of-contact areas are highlighted in red color for related process fixtures/tools (e.g., nozzle, collets, etc).

8858_COB_DS_8_2

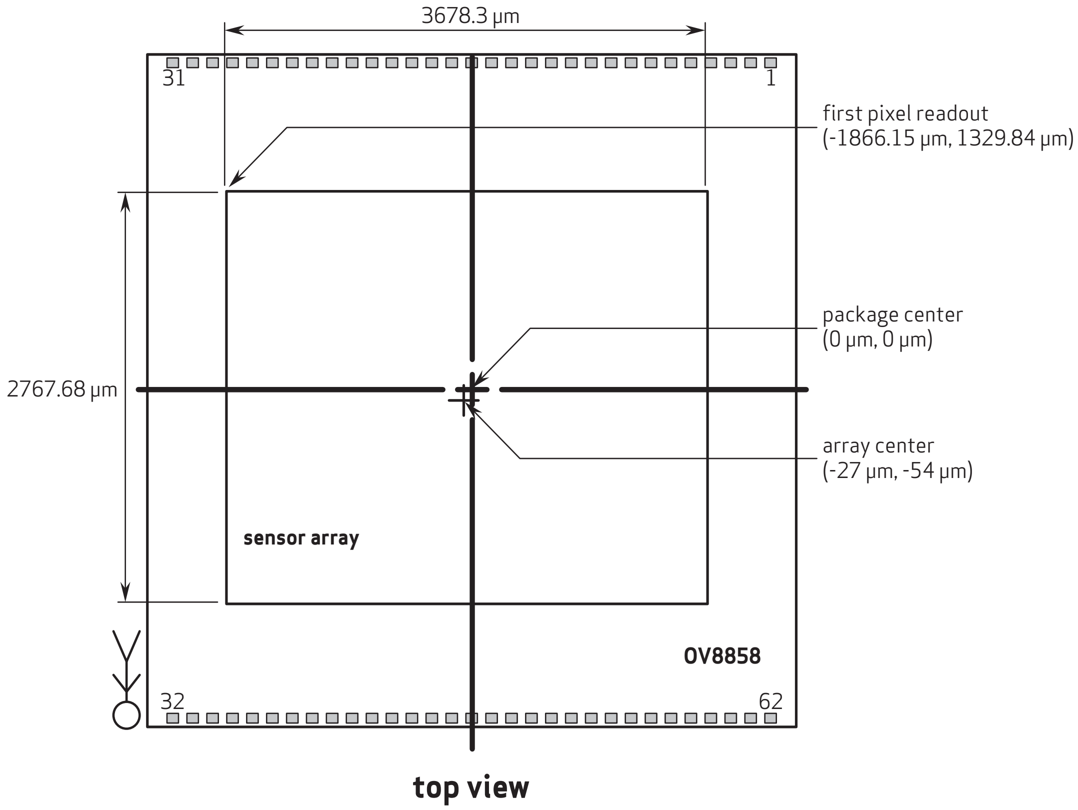
OV8858

color CMOS 8 megapixel (3264 x 2448) image sensor with OmniBSI-3™ technology

9 optical specifications

9.1 sensor array center

figure 9-1 sensor array center



note 1 this drawing is not to scale and is for reference only.

note 2 as most optical assemblies invert and mirror the image, the chip is typically mounted with pad 1 oriented down on the PCB.

8858_COB_DS_9_1

9.2 lens chief ray angle (CRA)

figure 9-2 chief ray angle (CRA)

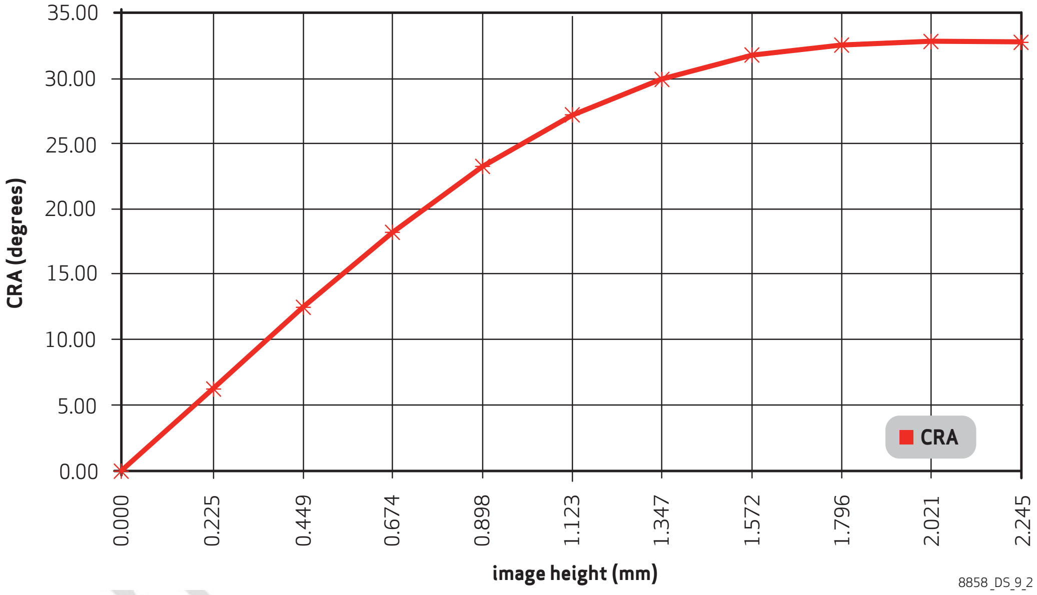


table 9-1 CRA versus image height plot

field (%)	image height (mm)	CRA (degrees)
0.00	0.000	0.00
0.10	0.225	6.35
0.20	0.449	12.55
0.30	0.674	18.30
0.40	0.898	23.29
0.50	1.123	27.26
0.60	1.347	30.07
0.70	1.572	31.80
0.80	1.796	32.63
0.90	2.021	32.86
1.00	2.245	32.78

appendix A handling of RW devices

A.1 ESD /EOS prevention

1. Ensure that there is 500V ESD control in all work areas.
2. Use ESD safety shoes, ground strap, and static control smocks in test areas.
3. Use grounded work carts and tables in inspection areas.
4. OmniVision recommends the use of ionized air in all work areas.

A.2 particles and cleanliness of environment

1. All production, inspection and packaging areas should meet Class10 environment requirements.
2. Use optical microscopes with 50X and 100X magnifications for particle inspection.
3. Ensure that there is good cassette sealing for particle protection during storage.
4. OmniVision recommends air blowing to remove removable particles.
5. RW die should be stored in nitrogen gas purged cabinets with temperature less than 30°C and relative humidity of 60% before assembly.

A.3 other requirements

1. Reliability assurance of RW or COB bare die is certified by product reliability of the bare die in a CLCC, CSP or QFP package form factor. Precautions should be taken if the packaging form factor of the bare die is other than these specified.
2. Avoid exposure to strong sunlight for extended periods of time as the color filter of the image sensor may become discolored.
3. Avoid direct exposure of the sensor bare die to high temperature and/or humidity environment as sensor characteristics will be affected. Extra precautions should be exercised if the bare die experiences temperatures exceeding 260°C for more than 75 seconds.

revision history

version 1.0 07.30.2013

- initial release

OV8858

color CMOS 8 megapixel (3264 x 2448) image sensor with OmniBSI-3™ technology

defining the future of digital imaging™

OmniVision Technologies, Inc.

UNITED STATES

4275 Burton Drive
Santa Clara, CA 95054

tel: + 1 408 567 3000

fax: + 1 408 567 3001

email: salesamerican@ovt.com

UNITED KINGDOM

Hampshire + 44 1256 744 610

GERMANY

Munich +49 89 63 81 99 88

INDIA

Bangalore +91 988 008 0140

CHINA

Beijing + 86 10 6580 1690

Shanghai + 86 21 6175 9888

Shenzhen + 86 755 8384 9733

Hong Kong + 852 2403 4011

JAPAN

Yokohama +81 45 478 7977

Osaka +81 6 4964 2606

KOREA

Seoul + 82 2 3478 2812

SINGAPORE +65 6220 1335

TAIWAN

Taipei +886 2 2657 9800

Hsinchu +886 3 6110933

website: **www.ovt.com**