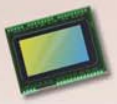
OmniVision



datasheet

PRELIMINARY SPECIFICATION

1/6" color CMOS 1080p (1920x1080) HD
PureCel™ image sensor

OV2740

color CMOS 1080p (1920 x 1080) HD PureCel™ image sensor

datasheet (CSP5)
PRELIMINARY SPECIFICATION

version 1.11
may 2014

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

applications

- smart phones
- ultrabooks and notebooks
- tablets
- digital still cameras (DSC)
- digital video camcorders (DVC)
- PC multimedia

ordering information

- **OV02740-H34A-Z** (color, lead-free)
  34-pin CSP5

features

- 1.4 µm x 1.4 µm pixel
- optical size of 1/6"
- programmable controls for frame rate, mirror and flip, cropping, and windowing
- supports output formats: 10-bit RAW RGB
- supports images sizes: 1080p (1920x1080), 720p (1280x720), VGA (640x480), QVGA (320x240), QQVGA (160x120)
- supports 2x2 binning
- standard serial SCCB interface
- up to 2-lane MIPI serial output interface (supports maximum speed up to 1000 Mbps/lane)
- embedded 4 kilobits of one-time programmable (OTP) memory for customer use
- add staggered HDR raw data output
- interleave row high dynamic range (iHDR) output
- programmable I/O drive capability
- power saving (PSV) mode
- support for LENC color shading correction

note The OV2740 supports LVDS interface. Contact your local FAE for details.

key specifications (typical)

- **active array size:** 1920 x 1080
- **power supply:**
  core: 1.2V
  analog: 2.8V
  I/O: 1.8V
- **power requirements:**
  active: 98.4 mW
  standby: 250 µA
  XSHUTDN: 0.15 µA
- **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 formats:** 10-bit RGB RAW
- **lens size:** 1/6"
- **input clock frequency:** 6~27 MHz
- **lens chief ray angle:** 33° non-linear
- **max S/N ratio:** TBD
- **dynamic range:** TBD
- **maximum image transfer rate:**
  1920x1080: 60 fps (see **table 2-1**)
  1280x720: 90 fps (see **table 2-1**)
- **sensitivity:** TBD
- **scan mode:** progressive
- **maximum exposure interval:** TBD
- **pixel size:** 1.4 µm x 1.4 µm
- **dark current:** TBD
- **image area:** 2728.8 µm x 1549.8 µm
- **package dimensions:** 3855 µm x 2919 µm

note higher junction temperature degrades image quality

table of contents

list of figures

list of tables

1 signal descriptions

table 1-1 lists the signal descriptions and their corresponding pin numbers for the OV2740 image sensor. The package information is shown in section 8.

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

pin number	signal name	pin type	description
A1	AVDD	power	analog power
A2	XSHUTDN2	input	reset and power down (active low with pull down resistor)
A3	DOVDD	power	I/O power
A4	STROBE	output	frame exposure output indicator
A5	GPIO	I/O	general purpose I/O
A6	SID	input	SCCB last bit ID input 0: SCCB ID address = 0x6C 1: SCCB ID address = 0x20
A7	DVDD	power	digital circuit power
B1	DOGND	ground	I/O ground
B2	PWDNB	input	power down (active low)
B3	FSIN	I/O	frame sync
B4	XVCLK	input	system clock input
B5	SDA	I/O	SCCB interface data pin
B6	SCL	input	SCCB interface input clock
B7	AGND	ground	analog ground
C1	AGND	ground	analog ground
C2	TM	input	test mode (active high with pull down resistor)
C3	XSHUTDN	input	reset and power down (active low with pull down resistor)
C4	PVDD	power	PLL analog power
C5	EGND	ground	ground for MIPI TX circuit
C6	DOVDD	power	I/O power
C7	AVDD	power	analog power
D1	VH	input	reference
D2	VN	input	reference
D3	DVDD	power	digital circuit power

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

pin number	signal name	pin type	description
D4	MDN0	output	MIPI data negative output
D5	MCN	output	MIPI clock negative output
D6	MDN1	output	MIPI data negative output
D7	DOGND	ground	I/O ground
E1	AVDD	power	analog power
E3	DOGND	ground	I/O ground
E4	MDP0	output	MIPI data positive output
E5	MCP	output	MIPI clock positive output
E6	MDP1	output	MIPI data positive output
E7	DVDD	power	digital circuit power

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

pin	signal name	RESET[a]	after RESET release[b]	software standby[c]	hardware standby[d]
A2	XSHUTDN2	input	input	input	input
A4	STROBE	low	low	low by default (configurable)	low by default (configurable)
A5	GPIO	low	low	low by default (configurable)	low by default (configurable)
A6	SID	input	input	input	input
B2	PWDNB	input	input	input	input
B3	FSIN	high-z	input	input (configurable)	input (configurable)
B4	XVCLK	high-z	input	input	high-z
B5	SDA	open drain	I/O	I/O	open drain
B6	SCL	high-z	input	input	high-z
C2	TM	input	input	input	input
C3	XSHUTDN	input	input	input	input
D4	MDN0	high-z	high	high by default (configurable)	high by default (configurable)
D5	MCN	high-z	high	high by default (configurable)	high by default (configurable)

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

pin	signal name	RESET[a]	after RESET release[b]	software standby[c]	hardware standby[d]
D6	MDN1	high-z	high	high by default (configurable)	high by default (configurable)
E4	MDP0	high-z	high	high by default (configurable)	high by default (configurable)
E5	MCP	high-z	high	high by default (configurable)	high by default (configurable)
E6	MDP1	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 pin diagram

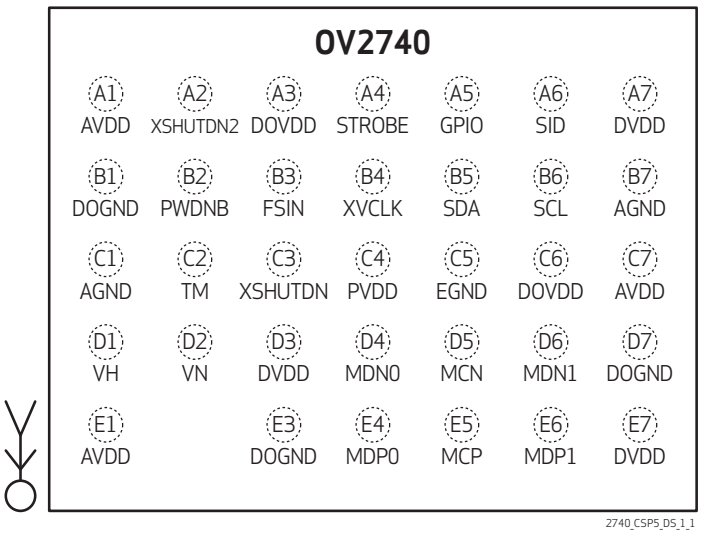


table 1-3 pad symbol and equivalent circuit (sheet 1 of 2)

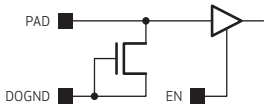
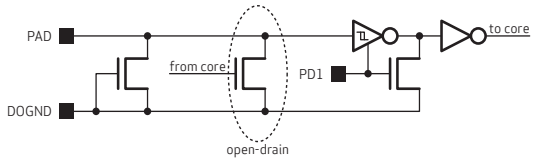
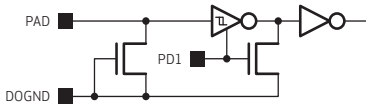
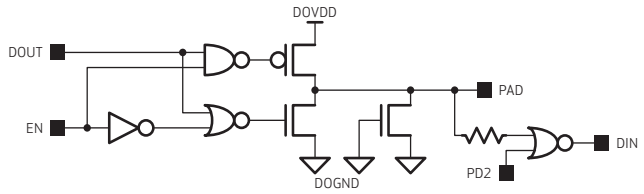
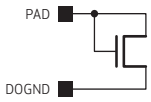
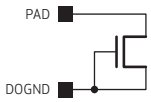
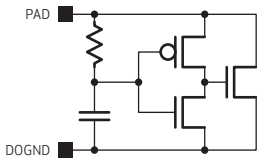
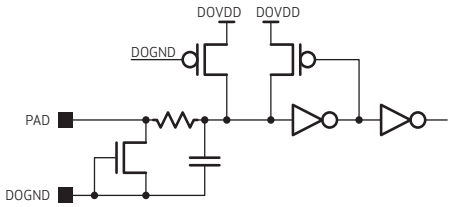
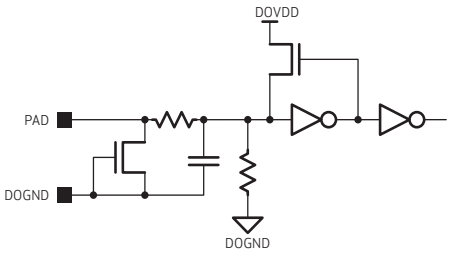
symbol	equivalent circuit
XVCLK	
SDA	
SCL	
STROBE, FSIN, GPIO	
VN	
MDP1, MDP0, MDN1, MDN0, MCP, MCN, EGND, AGND, DOGND, VH	
AVDD, DVDD, DOVDD, PVDD	
PWDNB	

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

symbol	equivalent circuit
XSHUTDN, SID, TM, XSHUTDN2	

2 system level description

2.1 overview

The OV2740 RAW RGB PureCel™ image sensor is a high performance, 1/6-inch 2 megapixel CMOS image sensor that delivers 1920x1080 at 60 fps. 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 OV2740 has a 2 megapixel image array capable of operating at up to 60 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, 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 OV2740 includes 4k bits of one-time programmable (OTP) memory (2k bits are reserved for OmniVision and 2k bits are reserved for customers). The OV2740 has a MIPI interface of up to two lanes.

2.2 architecture

The OV2740 sensor core generates streaming pixel data at a constant frame rate. **figure 2-1** shows the functional block diagram of the OV2740 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 OV2740 block diagram

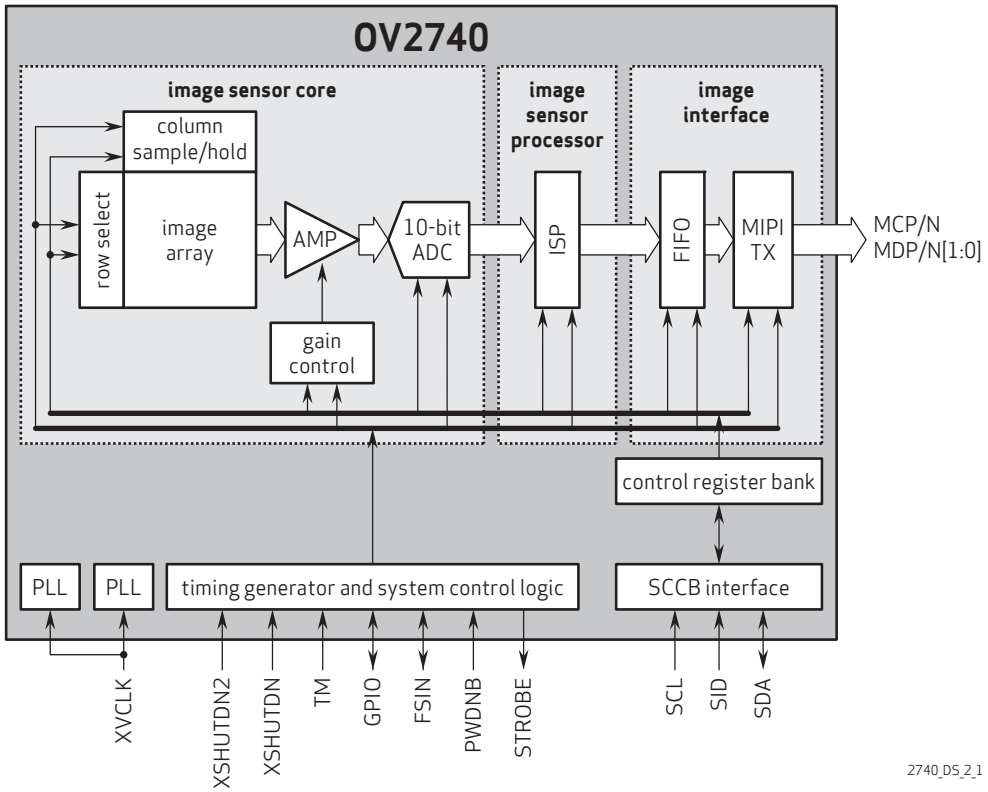
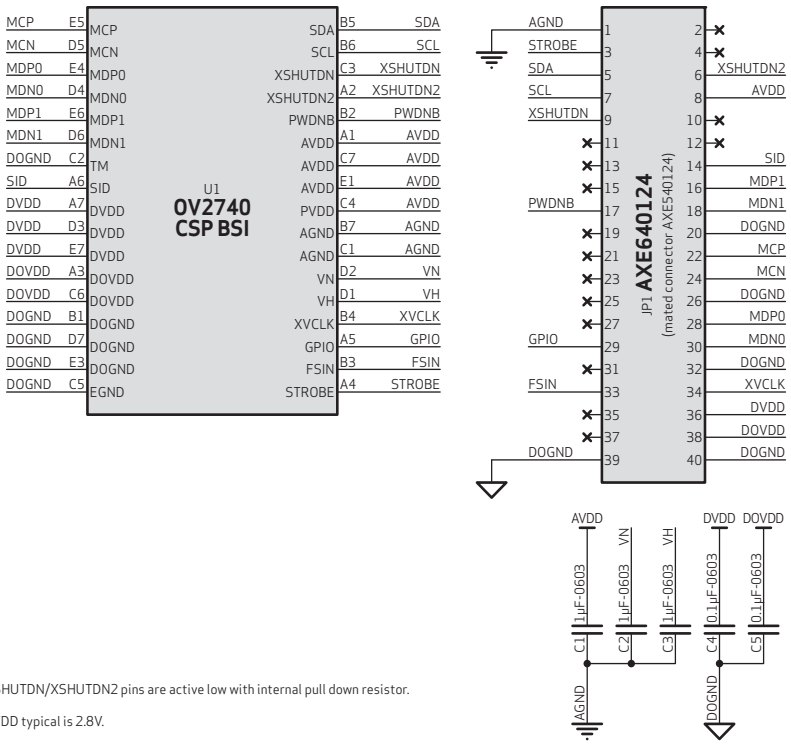


figure 2-2 OV2740 reference schematic



note 1 XSHUTDN/XSHUTDN2 pins are active low with internal pull down resistor.

note 2 AVDD typical is 2.8V.

note 3 DOVDD typical is 1.8V (1.7-3.0V).

note 4 DVDD typical is 1.2V.

note 5 sensor AGND and DOGND should be separated inside the module and connected to a single point outside the module (do not connect inside the module).

note 6 capacitors should be close to their related sensor pins.

note 7 EGND is the ground of MIPI circuit.
MCP and MCN are MIPI clock lane positive and negative output.
MDP0 and MDN0 are MIPI data lane0 positive and negative output.
MDP1 and MDN1 are MIPI data lane1 positive and negative output.

note 8 traces of MCP, MCN, MDP0, MDN0, MDP1 and MDN1 should have the same or similar length.
differential impedance of the clock pair and data pair transmission line should be controlled at 100 Ohm.

note 9 SID is the SCCB ID select input:
SID = 0, SCCB device address = 0x6C
SID = 1, SCCB device address = 0x20
if customer does not need to switch SID, connect it to DOVDD or DOGND depending on which ID the customer would like to use.

note 10 OV2740 uses an external power supply to provide digital core 1.2V DVDD
OV2740 does not have an internal regulator to provide 1.2V DVDD.

2740_CSP5_DS_2_2

2.3 format and frame

The OV2740 supports RAW RGB output with one/two MIPI interface.

table 2-1 format and frame rate

format	resolution	max frame rate	methodology	10-bit output MIPI data rate
2 Mpixel	1920x1080	60 fps	full resolution (16:9)	720 Mbps/lane
720p	1280 x 720	90 fps	cropping	720 Mbps/lane
VGA	640 x 480	120 fps	cropping + 2x binning	720 Mbps/lane
QVGA	320 x 240	240 fps	cropping + 2x binning + 2x skip	720 Mbps/lane
QQVGA (vision mode)	160 x 120	30 fps	cropping + 2x binning + 4x skip	720 Mbps/lane

2.4 I/O control

The OV2740 can configure its I/O pins 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

function	register	description
FSIN I/O control	0x3000	Bit[5]: FSIN output enable 0: input 1: output
FSIN output select	0x300E	Bit[5]: Enable FSIN as GPIO controlled by register
FSIN output value	0x3008	Bit[5]: Register control FSIN output
GPIO I/O control	0x3002	Bit[0]: GPIO output enable 0: input 1: output
GPIO output select	0x3010	Bit[0]: Enable GPIO as GPIO controlled by register
GPIO output value	0x300D	Bit[0]: Register control GPIO output

2.5 MIPI interface

The OV2740 supports an MIPI interface of up to two lanes. The MIPI interface can be configured for 1/2-lane and each lane is capable of a data transfer rate of up to 1000 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 lock period	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)

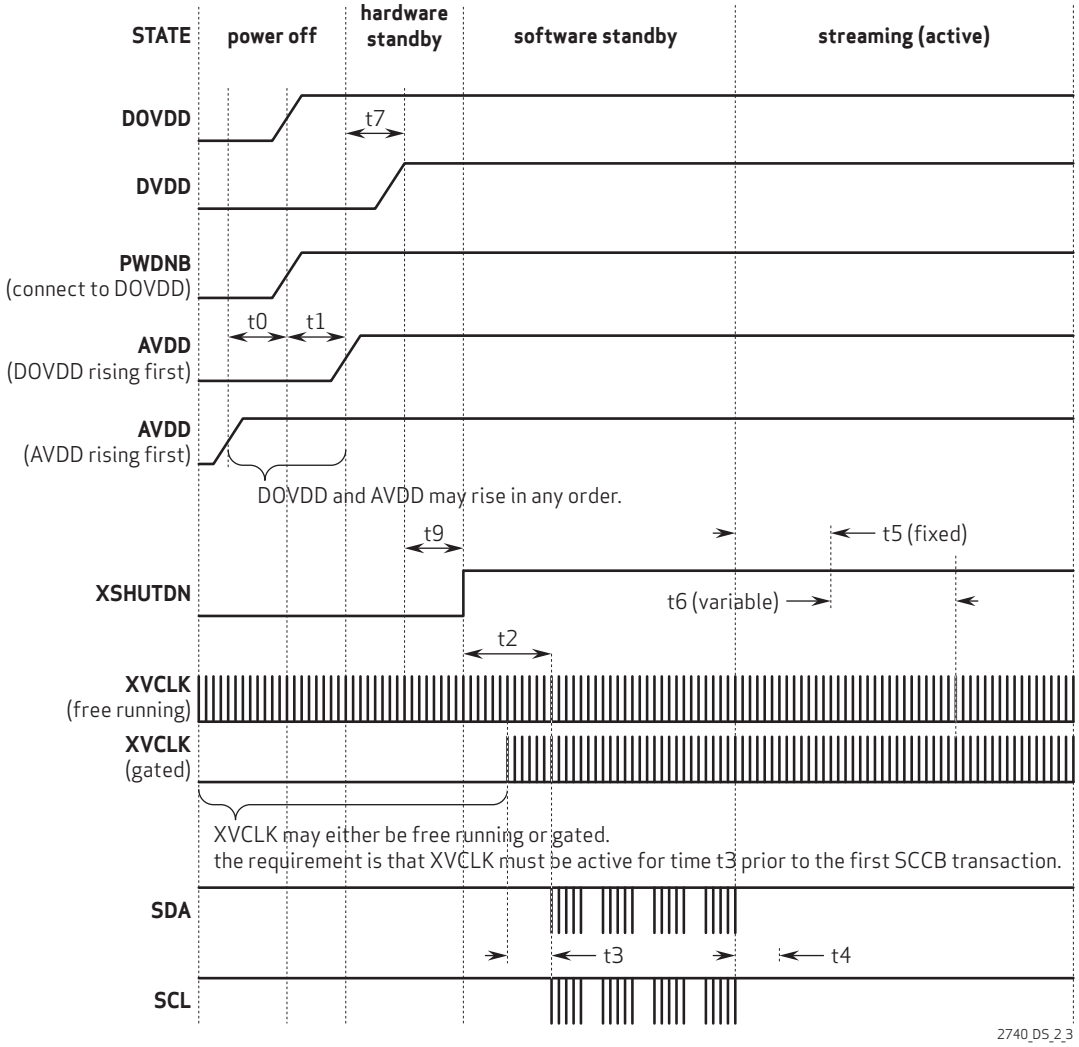
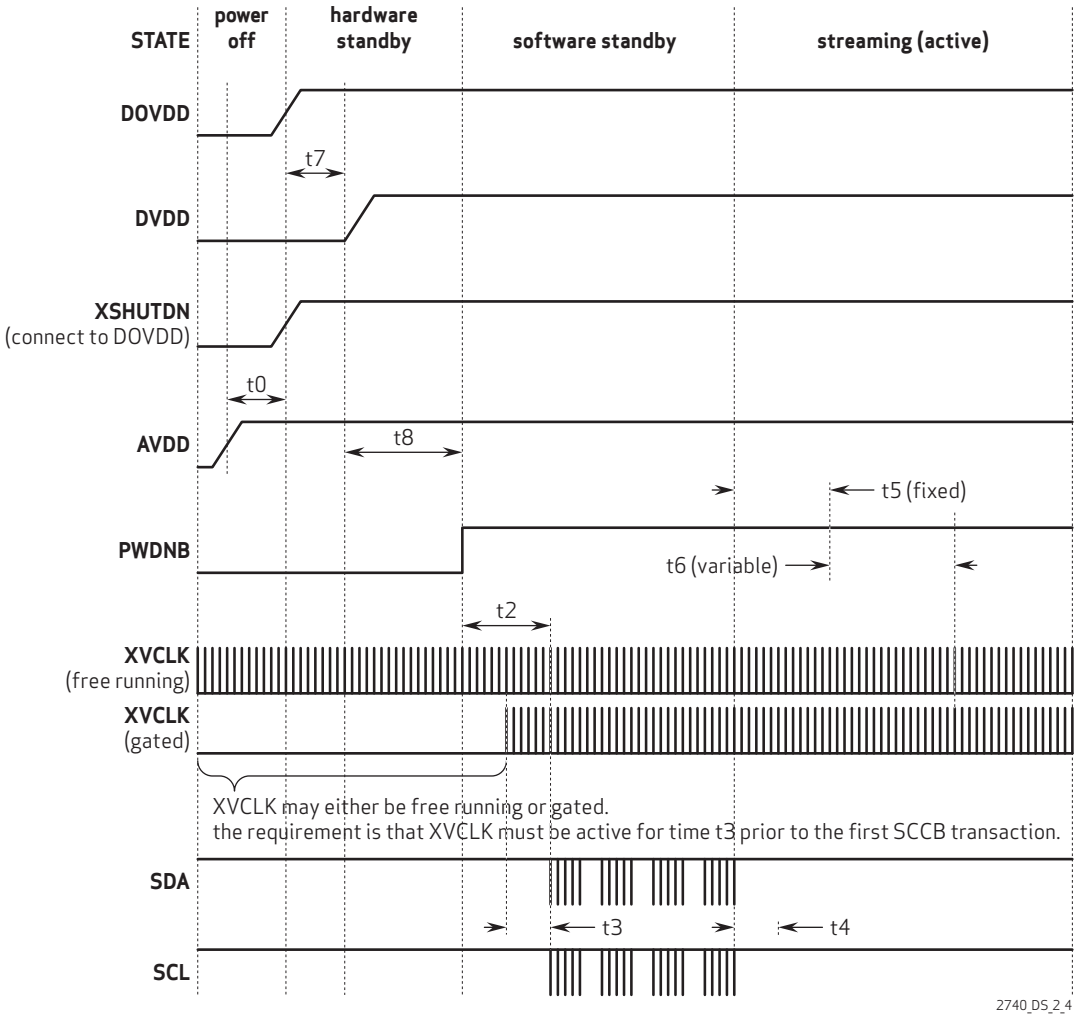


figure 2-4 power up sequence (case 2)



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 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

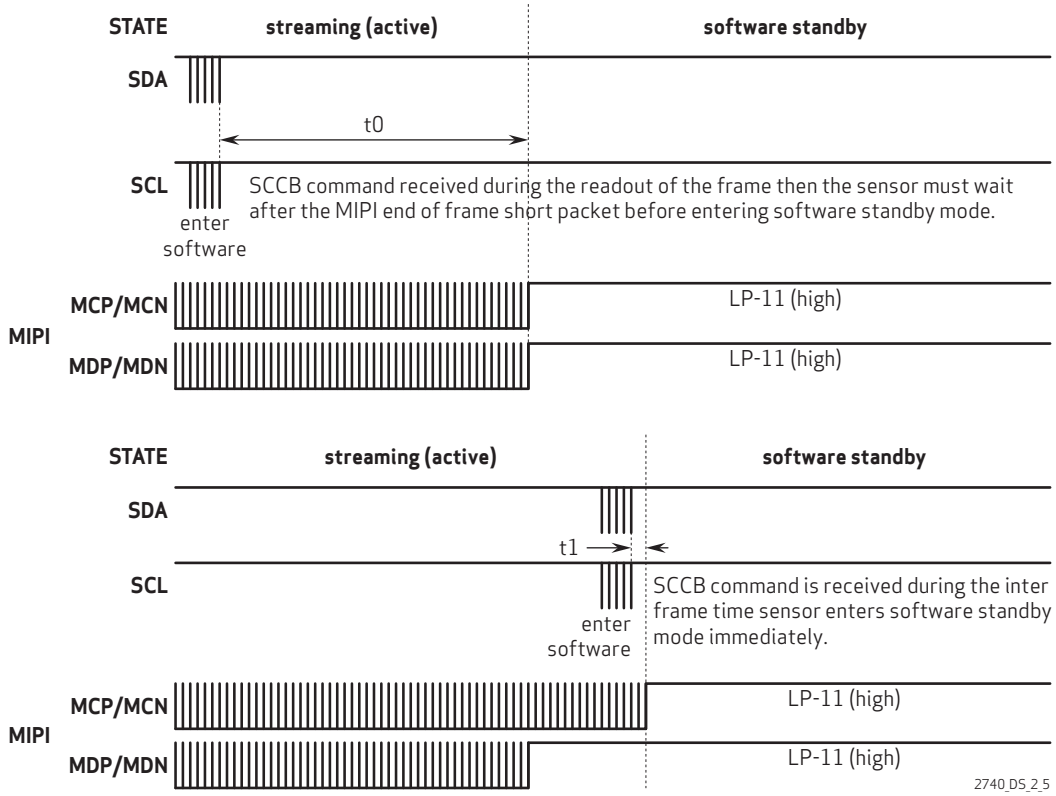


figure 2-6 power down sequence (case 1)

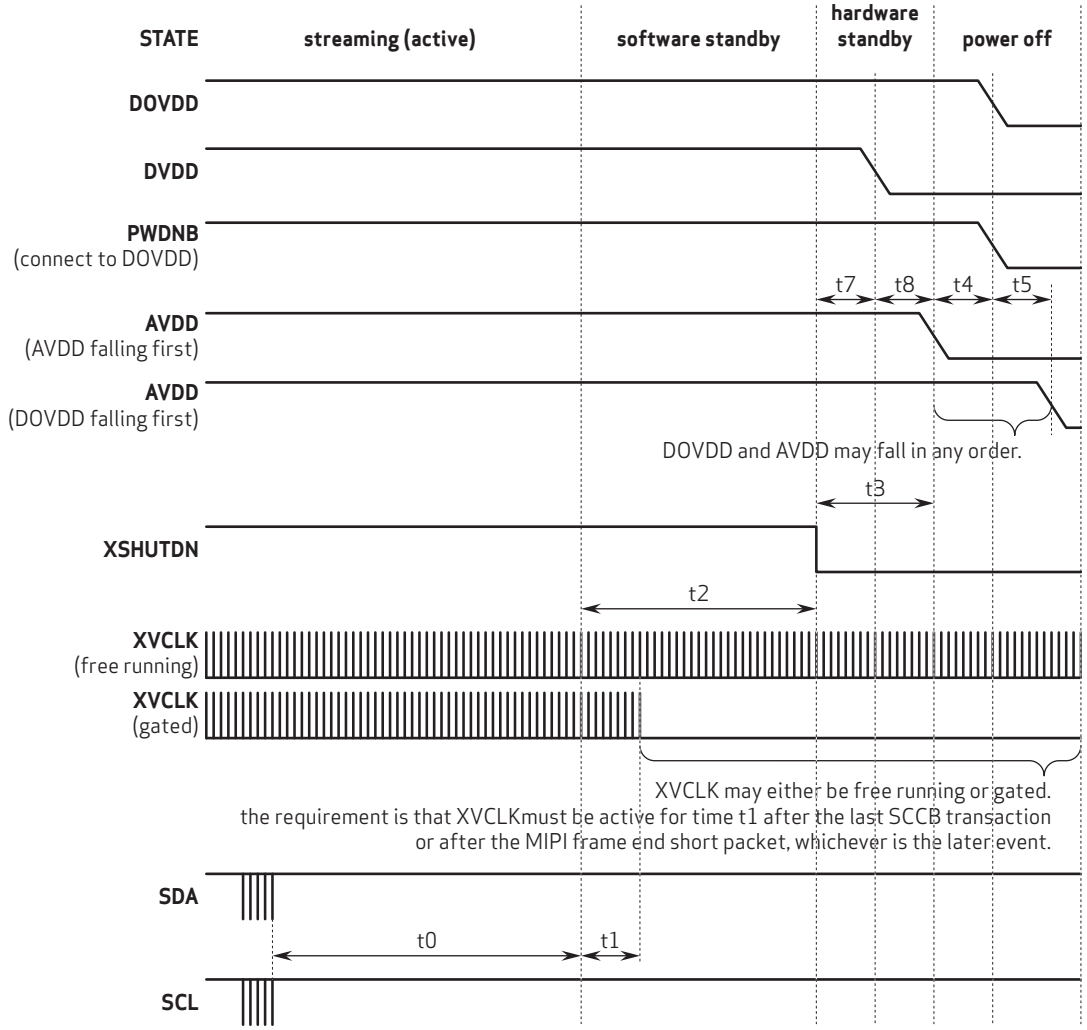
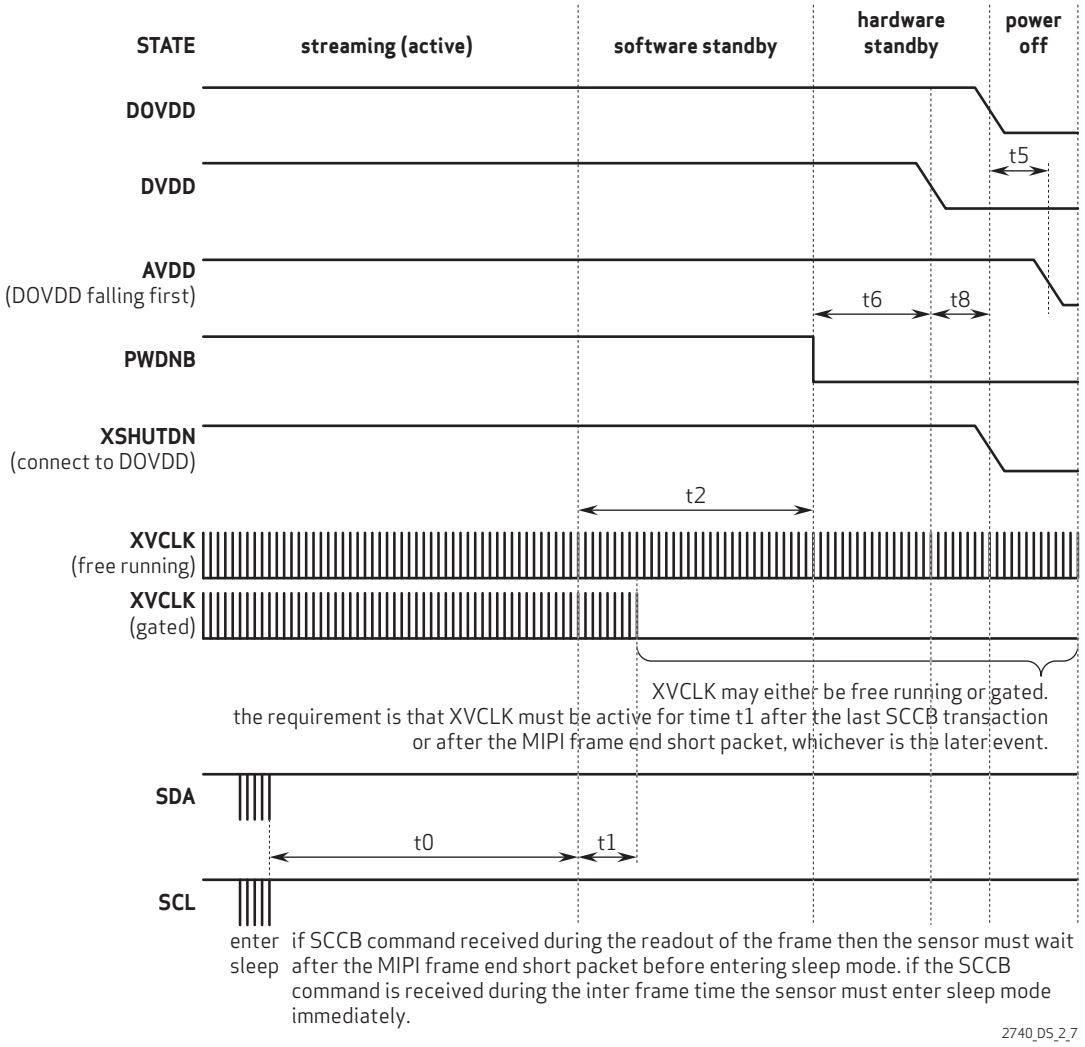


figure 2-7 power down sequence (case 2)



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 OV2740:

- 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 OV2740 internal device clock is halted even when the external clock source is still clocking and all internal counters are reset. When resumed from PWDNB enabled hardware standby, all the registers are restored, while XSHUTDN resets all registers to the default value.

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 6~64 MHz input clock. The VCO range is from 500 MHz to 1200 MHz. A programmable clock is provided to generate different frequencies.

2.9.2 PLL2

The PLL2 generates a default 72 MHz system clock from a 6~64 MHz input clock. The VCO range is from 500 MHz to 1200 MHz. A programmable clock divider is provided to generate different frequencies.

figure 2-8 clock scheme

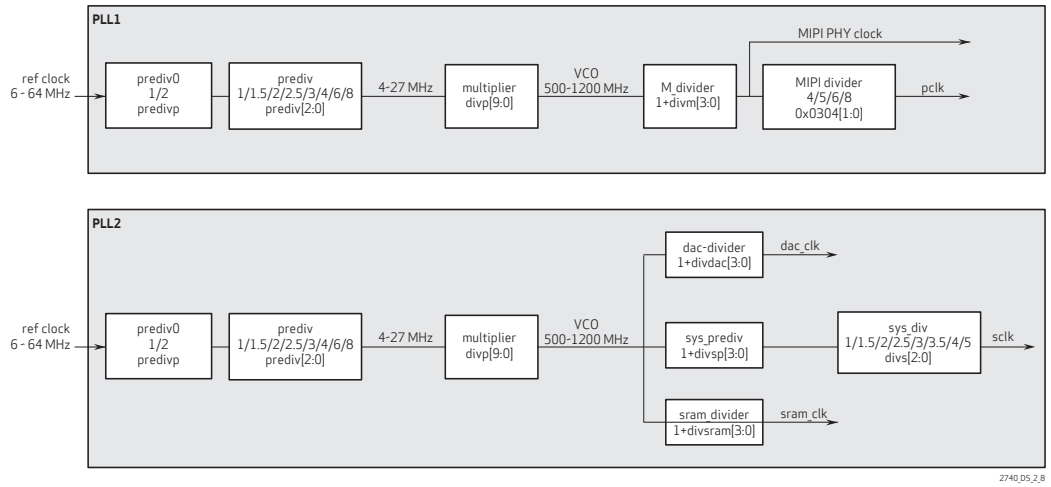


table 2-8 PLL registers

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_divp[9:8]
0x0302	PLL_CTRL_2	0x19	RW	Bit[7:0]: pll1_divp[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
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
0x0313	PLL_CTRL_13	0x04	RW	Bit[4]: pll2_div_rst Bit[3:0]: pll2_divsram

table 2-9 sample PLL configuration

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	0x78
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	0x4
PLL2_DIVS	0x030E[2:0]	0x2	0x2
SCLK	–	72MHz	72MHz
PHY_SCLK	–	720MHz	720MHz
MIPI_PCLK	–	90MHz	90MHz

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 OV2740, 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.

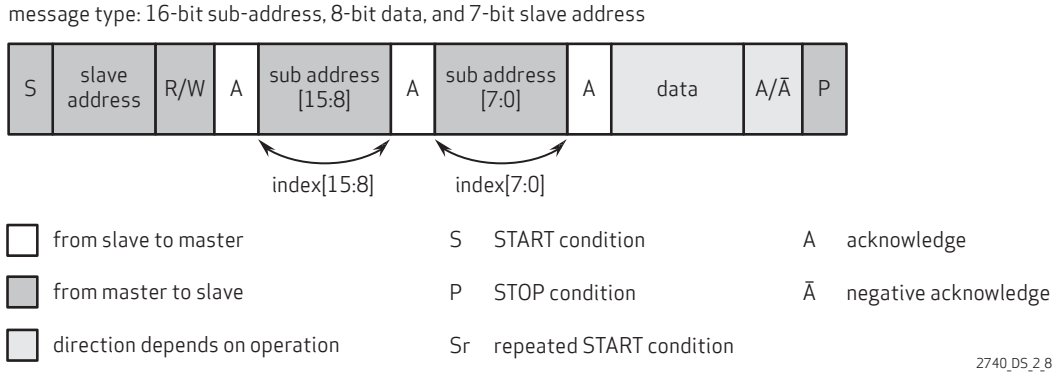
2.10.1 data transfer protocol

The data transfer of the OV2740 follows the SCCB protocol.

2.10.2 message format

The OV2740 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

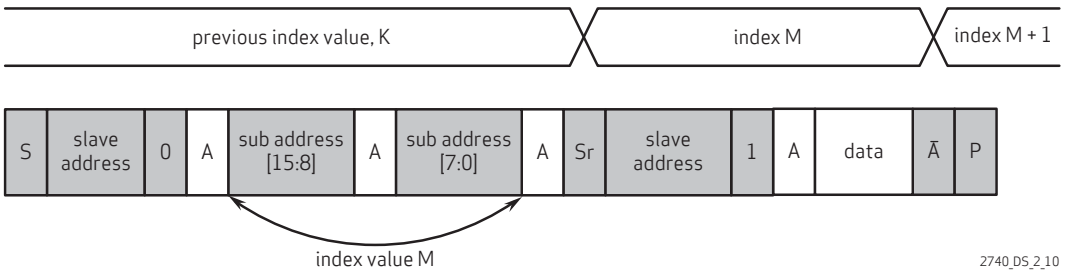
The OV2740 supports four different read operations and two different write operations:

- 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.

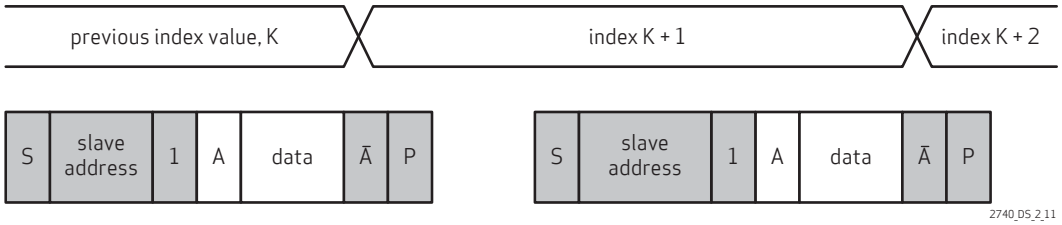
In a single read from random locations, the master does a dummy write operation to desired sub-address, issues a repeated start condition and then addresses the camera again with a read operation. After acknowledging its slave address, the camera starts to output data onto the SDA line as shown in **figure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



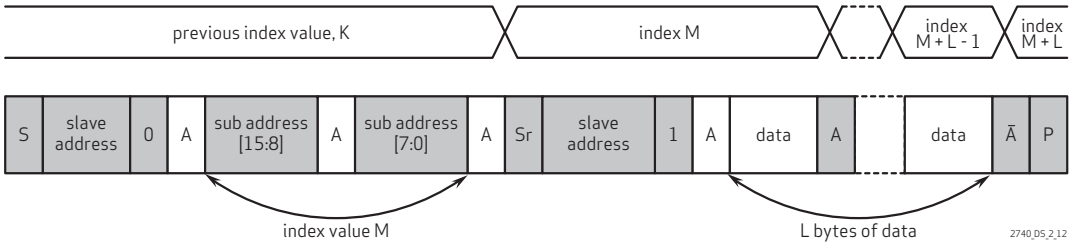
If the host addresses the camera with read operation directly without the dummy write operation, the camera responds by setting the data from last used sub-address to the SDA line as shown in **figure 2-11**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 2-11 SCCB single read from current location



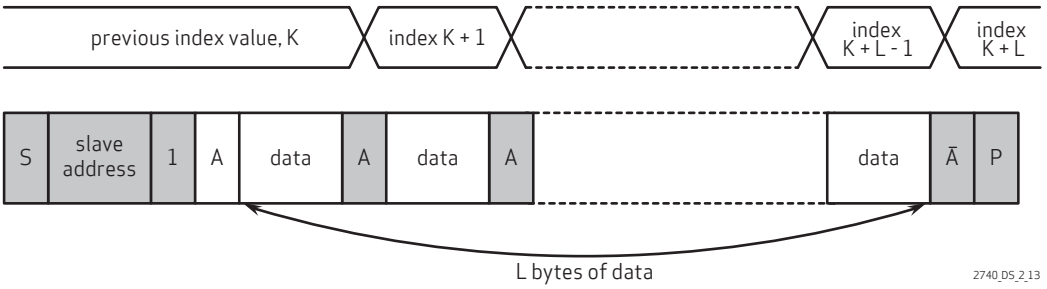
The sequential read from a random location is illustrated in **figure 2-12**. The master does a dummy write to the desired sub-address, issues a repeated start condition after acknowledge from slave and addresses the slave again with read operation. If a master issues an acknowledge after receiving data, 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



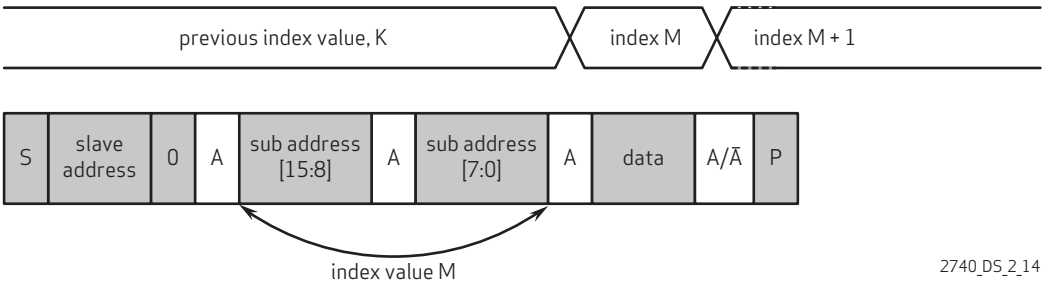
The sequential read from current location is similar to a sequential read from a random location. The only exception is that there is no dummy write operation, as shown in **figure 2-13**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 2-13 SCCB sequential read from current location



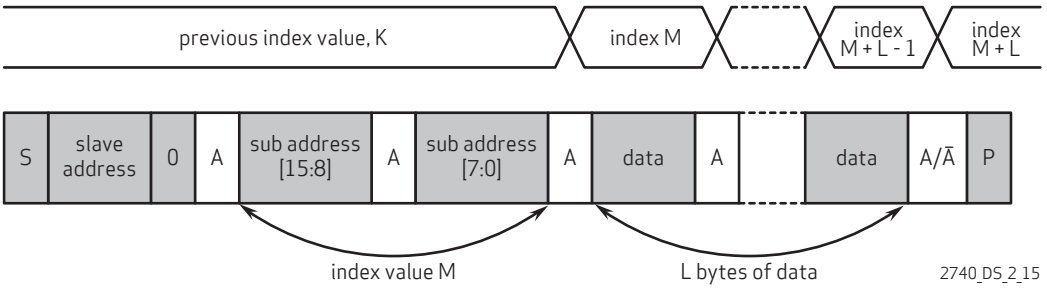
The write operation to a random location is illustrated in **figure 2-14**. The master issues a write operation to the slave, sets the sub-address and data correspondingly after the slave has acknowledged. The write operation is terminated with a stop condition from the master.

figure 2-14 SCCB single write to random location



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



2.10.4 SCCB timing

figure 2-16 SCCB interface timing

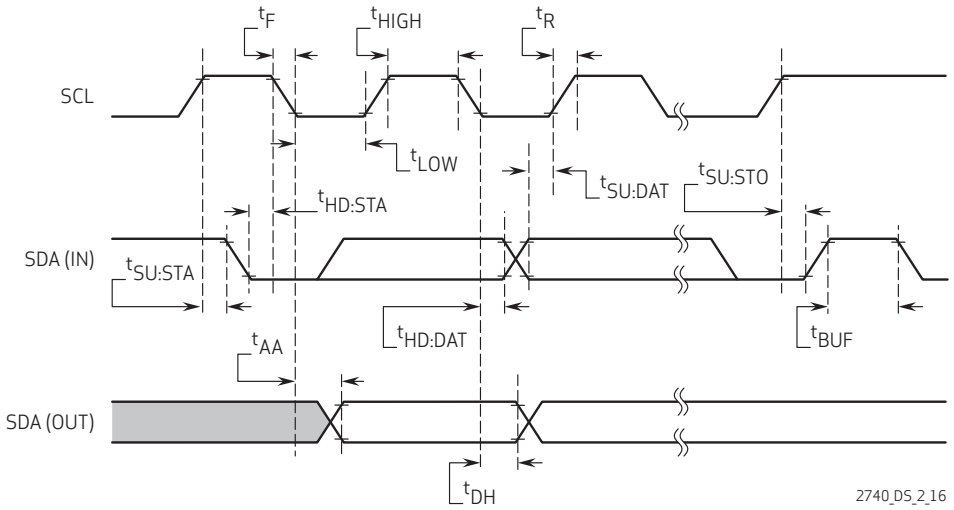


table 2-10 SCCB interface timing specifications[ab]

symbol	parameter	min	typ	max	unit
f_{SCL}	clock frequency			400	kHz
t_{LOW}	clock low period	1.3			µs
t_{HIGH}	clock high period	0.6			µs
t_{AA}	SCL 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 OV2740 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.

3 block level description

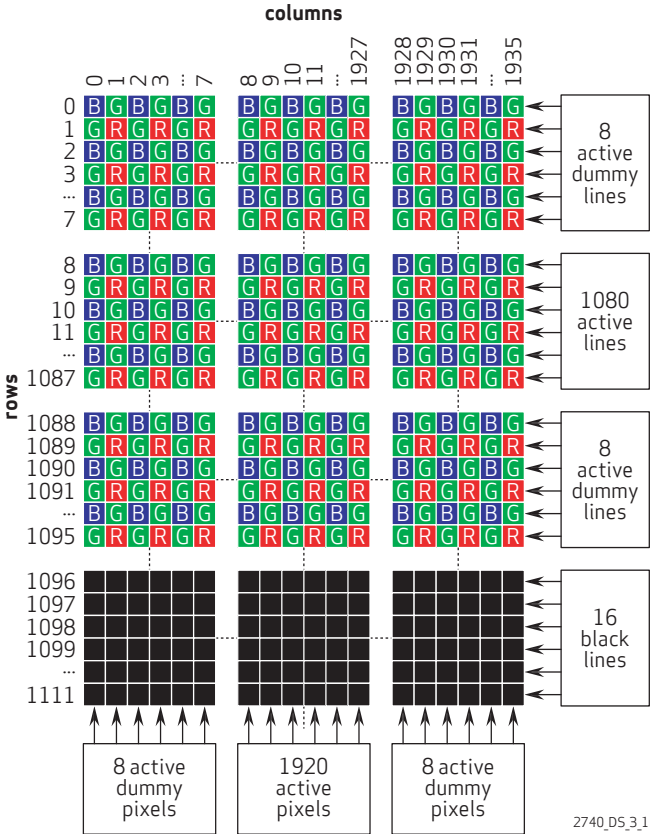
3.1 pixel array structure

The OV2740 sensor has an image array of 1936 columns by 1112 rows (2,152,832 pixels including 20 black lines). figure 3-1shows 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 2,152,832 pixels, 2,121,856 (1936x1096) are active pixels and can be output. The other pixels are used for black level calibration and interpolation. The center 1920x1080 pixels are 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 OV2740 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 OV2740 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

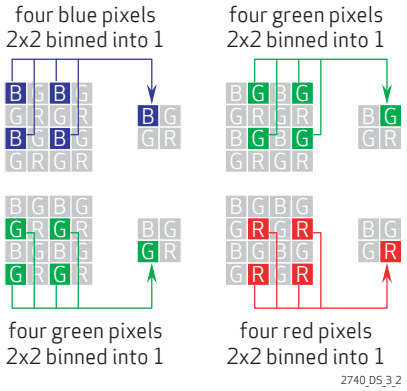


table 3-1 binning-related registers

address	register name	default value	R/W	description
0x3814	X_ODD_INC	0x01	RW	Bit[4:0]: Horizontal increase number at odd pixel
0x3815	X_EVEN _INC	0x01	RW	Bit[4:0]: Horizontal increase number at even pixel
0x3821	TIMING_FORMAT2	0x08	RW	Bit[5]: Vertical binning Bit[0]: horizontal binning
0x382A	Y_ODD_INC	0x01	RW	Bit[4:0]: Vertical increase number at odd row
0x382B	Y_EVEN_INC	0x01	RW	Bit[4:0]: 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 OV2740 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

original image | mirrored image | flipped image | mirrored and flipped image

2740_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.

figure 4-2 image windowing

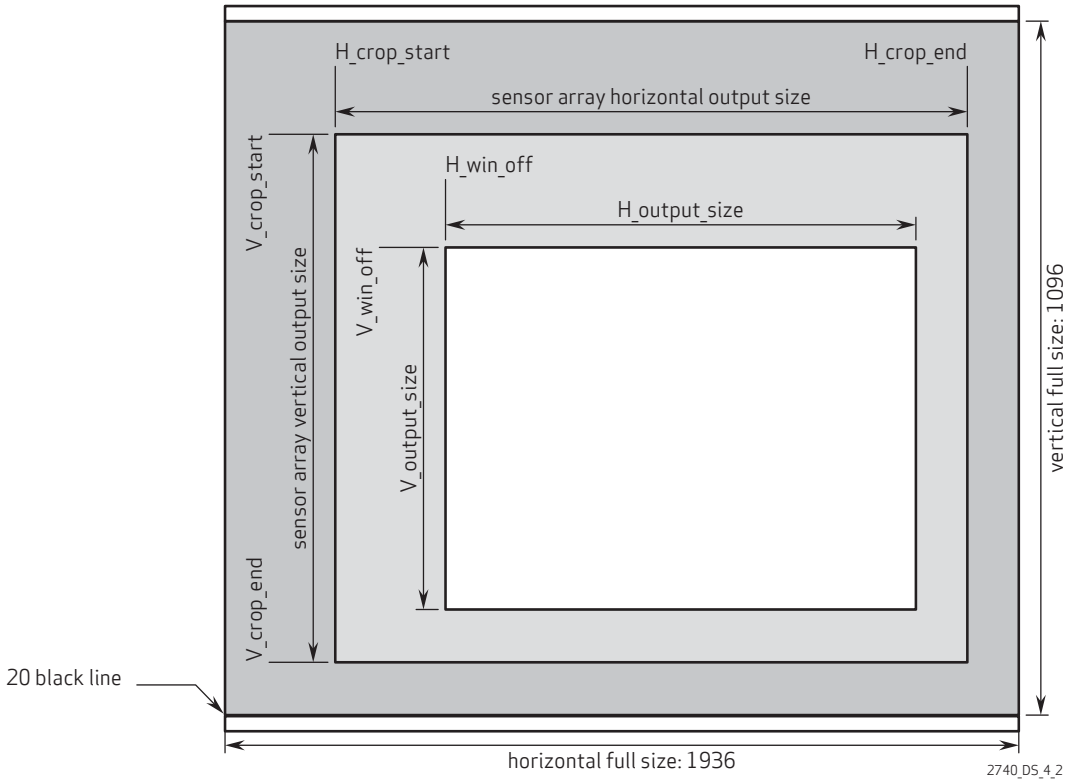


table 4-2 image windowing control functions

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	0x00	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	0x04	RW	Bit[7:0]: Manual vertical crop start address[7:0]
0x3804	H_CROP_END	0x07	RW	Bit[3:0]: Manual horizontal crop end address[11:8]
0x3805	H_CROP_END	0xF3	RW	Bit[7:0]: Manual horizontal crop end address[7:0]
0x3806	V_CROP_END	0x04	RW	Bit[3:0]: Manual vertical crop end address[11:8]
0x3807	V_CROP_END	0x43	RW	Bit[7:0]: Manual vertical crop end address[7:0]
0x3808	H_OURPUT_SIZE	0x07	RW	Bit[3:0]: Horizontal output size[11:8]
0x3809	H_OUTPUT_SIZE	0x80	RW	Bit[7:0]: Horizontal output size[7:0]
0x380A	V_OUTPUT_SIZE	0x04	RW	Bit[3:0]: Vertical output size[11:8]
0x380B	V_OUTPUT_SIZE	0x38	RW	Bit[7:0]: Vertical output size[7:0]
0x380C	TIMING_HTS	0x04	RW	Bit[7:0]: Horizontal total size[15:8]
0x380D	TIMING_HTS	0x38	RW	Bit[7:0]: Horizontal total size[7:0]
0x380E	TIMING_VTS	0x04	RW	Bit[6:0]: Vertical total size[14:8]
0x380F	TIMING_VTS	0x58	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	0x08	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	0x04	RW	Bit[7:0]: Manual vertical windowing offset[7:0]
0x3814	X_ODD_INC	0x01	RW	Bit[4:0]: Horizontal increase number at odd pixel
0x3815	X_EVEN_INC	0x01	RW	Bit[4:0]: Horizontal increase number at even pixel
0x382A	Y_ODD_INC	0x01	RW	Bit[4:0]: Vertical increase number at odd row
0x382B	Y_EVEN_INC	0x01	RW	Bit[4:0]: Vertical increase number at even row

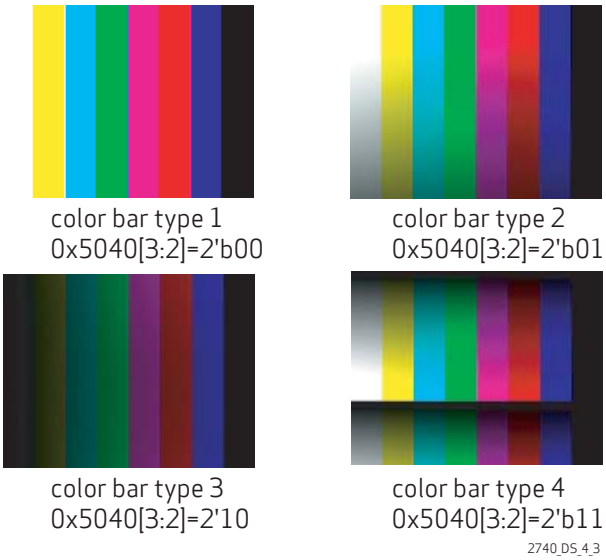
4.3 test pattern

For testing purposes, the OV2740 offers three types of test patterns: color bar, square and random data. The OV2740 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 (0x5040[3:2]). The digital test pattern function is controlled by register 0x5040[7].

4.3.1 color bar

There are four types of color bars which are switched by bar-style in register 0x5040[3:2] (see **figure 4-3**).

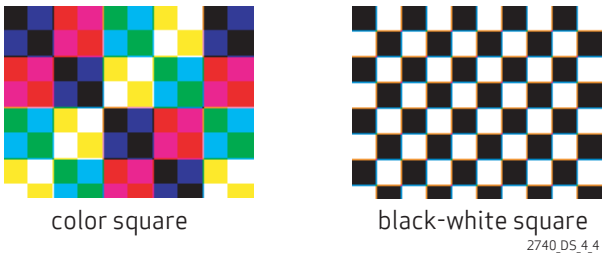
figure 4-3 color bar types



4.3.2 square

There are two types of squares: color square and black-white square. The squ_bw register (0x5040[4]) determines which type of square will be output.

figure 4-4 color, black and white square bars



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 (0x5040[5]). If this register is set, the transparent test pattern will be displayed. **figure 4-5** is an example showing a transparent color bar image.

figure 4-5 transparent effect

2740_DS_4_5

4.3.5 rolling bar effect

The rolling bar is set by rolling_bar_en register (0x5040[6]). If it is set, an inverted-color rolling bar will roll from up to down. **figure 4-6** is an example showing a rolling bar on color bar image.

figure 4-6 rolling bar effect

2740_DS_4_6

table 4-3 test pattern registers

address	register name	default value	R/W	description
0x5040	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
0x5041	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 OV2740 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.

figure 4-7 average based window definition

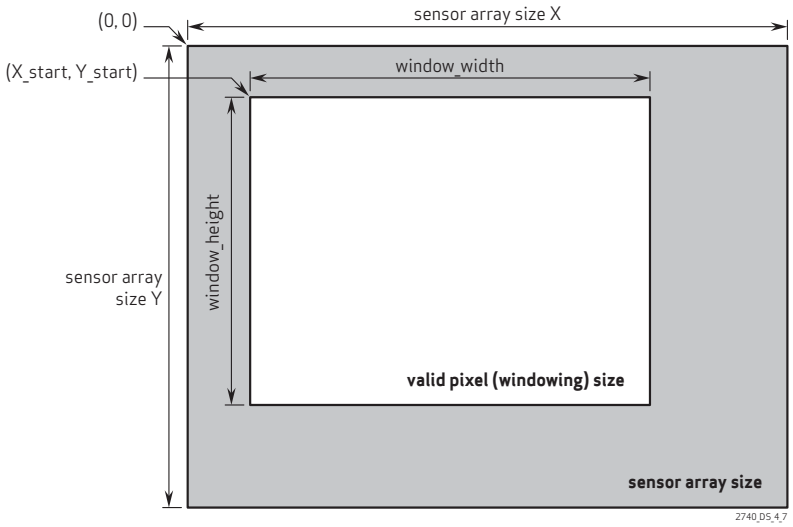


table 4-4 average control registers

address	register name	default value	R/W	description
0x5680	AVG CTRL00	0x00	RW	Bit[0]: X_start_avg[8]
0x5681	AVG CTRL01	0x00	RW	Bit[7:0]: X_start_avg[7:0]
0x5683	AVG CTRL03	0x00	RW	Bit[7:0]: Y_start_avg[7:0]
0x5684	AVG CTRL04	0x01	RW	Bit[0]: Window_width_avg[8]
0x5685	AVG CTRL05	0x40	RW	Bit[7:0]: Window_width_avg[7:0]
0x5687	AVG CTRL07	0xF0	RW	Bit[7:0]: Window_height_avg[7:0]
0x5693	AVG B	–	R	Average B Channel Read Out
0x5694	AVG G	–	R	Average G Channel Read Out
0x5695	AVG R	–	R	Average R Channel Read Out
0x5696	AVG Y	–	R	Average Y Channel Read Out

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.

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

address	register name	default value	R/W	description
0x4000	BLC CTRL 00	0xCF	RW	Bit[7:4]: Avg_weight Bit[3]: Target adjust disable Use offset to adjust target 0: Enable 1: Disable Bit[2]: Compensation enable Adjust on offset due to gain change 0: Enable 1: Disable Bit[1]: Dither_en 1 bit dithering 0: Enable 1: Disable Bit[0]: Median_en Median filter function enable
0x4001	BLC CTRL 01	0x20	RW	Bit[7:6]: r_hdr_option Bit[5]: Kcoef manual enable Bit[4]: off_man_en Bit[3]: Zero line out enable Bit[2]: Black line out enable Bit[1:0]: Bypass mode
0x4002	BLC CTRL 02	0x00	RW	Bit[1:0]: Blacklevel target[9:8]
0x4003	BLC CTRL 03	0x10	RW	Bit[7:0]: Blacklevel target[7:0]
0x4004	BLC CTRL 04	0x00	RW	Bit[3:0]: Horizontal win start[11:8]
0x4005	BLC CTRL 05	0x02	RW	Bit[7:0]: Horizontal win start[7:0]
0x4006	BLC CTRL 06	0x00	RW	Bit[3:0]: Horizontal win pad[11:8]
0x4007	BLC CTRL 07	0x10	RW	Bit[7:0]: Horizontal win pad[7:0]
0x4008	BLC CTRL 08	0x00	RW	Bit[7:0]: Black line start line
0x4009	BLC CTRL 09	0x0B	RW	Bit[7:0]: Black line end line
0x400A	BLC CTRL 0A	0xFF	RW	Bit[7:0]: Offset trigger threshold[15:8]
0x400B	BLC CTRL 0B	0xFF	RW	Bit[7:0]: Offset trigger threshold[7:0]
0x400C	BLC CTRL 0C	0x00	RW	Bit[7:0]: CVDN black lines start

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

address	register name	default value	R/W	description
0x400D	BLC CTRL 0D	0x00	RW	Bit[7:0]: CVDN black lines end
0x400E	BLC CTRL 0E	0x00	RW	Bit[1:0]: Kcoef man value[9:8]
0x400F	BLC CTRL 0F	0x80	RW	Bit[7:0]: Kcoef man value[7:0]
0x4010	BLC CTRL 10	0x60	RW	Bit[7]: Offset trigger enable Bit[6]: Gain change trigger enable Bit[5]: Format change trigger enable Bit[4]: Reset trigger enable Bit[3]: Manual average enable Bit[2]: Manual trigger Bit[1]: Freeze enable Bit[0]: Offset always update
0x4011	BLC CTRL 11	0x70	RW	Bit[6]: Offset trigger multiframe enable Bit[5]: Format trigger multiframe enable Bit[4]: Gain trigger multiframe enable Bit[3]: Reset trigger multiframe enable Bit[2]: Offset trigger multiframe mode Bit[1]: Format trigger multiframe mode Bit[0]: Gain trigger multiframe mode
0x4012	BLC CTRL 12	0x00	RW	Bit[5:0]: Reset trigger frame number
0x4013	BLC CTRL 13	0x02	RW	Bit[5:0]: Format trigger frame number
0x4014	BLC CTRL 14	0x02	RW	Bit[5:0]: Gain trigger frame number
0x4015	BLC CTRL 15	0x02	RW	Bit[5:0]: Offset trigger frame number
0x4016	BLC CTRL 16	0x00	RW	Bit[1:0]: Offset trigger threshold[9:8]
0x4017	BLC CTRL 17	0x00	RW	Bit[7:0]: Offset trigger threshold[7:0]
0x4020	BLC CTRL 20	0x00	RW	Bit[5:0]: Offset compensation th000
0x4021	BLC CTRL 21	0x00	RW	Bit[5:0]: Offset compensation k000
0x4022	BLC CTRL 22	0x00	RW	Bit[5:0]: Offset compensation th001
0x4023	BLC CTRL 23	0x00	RW	Bit[5:0]: Offset compensation k001
0x4024	BLC CTRL 24	0x00	RW	Bit[5:0]: Offset compensation th010
0x4025	BLC CTRL 25	0x00	RW	Bit[5:0]: Offset compensation k010
0x4026	BLC CTRL 26	0x00	RW	Bit[5:0]: Offset compensation th011
0x4027	BLC CTRL 27	0x00	RW	Bit[5:0]: Offset compensation k011
0x4028	BLC CTRL 28	0x00	RW	Bit[5:0]: Offset compensation th100
0x4029	BLC CTRL 29	0x00	RW	Bit[5:0]: Offset compensation k100

table 4-5 BLC registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x402A	BLC CTRL 2A	0x00	RW	Bit[5:0]: Offset compensation th101
0x402B	BLC CTRL 2B	0x00	RW	Bit[5:0]: Offset compensation k101
0x402C	BLC CTRL 2C	0x00	RW	Bit[5:0]: Offset compensation th110
0x402D	BLC CTRL 2D	0x00	RW	Bit[5:0]: Offset compensation k110
0x402E	BLC CTRL 2E	0x00	RW	Bit[5:0]: Offset compensation th111
0x402F	BLC CTRL 2F	0x00	RW	Bit[5:0]: Offset compensation k111
0x4030	BLC CTRL 30	0x00	RW	Bit[1:0]: Offset man 000[9:8]
0x4031	BLC CTRL 31	0x00	RW	Bit[7:0]: Offset man 000[7:0]
0x4032	BLC CTRL 32	0x00	RW	Bit[1:0]: Offset man 001[9:8]
0x4033	BLC CTRL 33	0x00	RW	Bit[7:0]: Offset man 001[7:0]
0x4034	BLC CTRL 34	0x00	RW	Bit[1:0]: Offset man 010[9:8]
0x4035	BLC CTRL 35	0x00	RW	Bit[7:0]: Offset man 010[7:0]
0x4036	BLC CTRL 36	0x00	RW	Bit[1:0]: Offset man 011[9:8]
0x4037	BLC CTRL 37	0x00	RW	Bit[7:0]: Offset man 011[7:0]
0x4038	BLC CTRL 38	0x00	RW	Bit[1:0]: Offset man 100[9:8]
0x4039	BLC CTRL 39	0x00	RW	Bit[7:0]: Offset man 100[7:0]
0x403A	BLC CTRL 3A	0x00	RW	Bit[1:0]: Offset man 101[9:8]
0x403B	BLC CTRL 3B	0x00	RW	Bit[7:0]: Offset man 101[7:0]
0x403C	BLC CTRL 3C	0x00	RW	Bit[1:0]: Offset man 110[9:8]
0x403D	BLC CTRL 3D	0x00	RW	Bit[7:0]: Offset man 110[7:0]
0x403E	BLC CTRL 3E	0x00	RW	Bit[1:0]: Offset man 111[9:8]
0x403F	BLC CTRL 3F	0x00	RW	Bit[7:0]: Offset man 111[7:0]
0x4042	BLC CTRL 42	0x10	RW	Bit[7]: r_format_trig_beh Bit[6]: r_gain_trig_beh Bit[5]: r_stagger_ls_man Bit[4]: r_blk_col_out_en Bit[3:2]: r_manu_cvdn_out_en Bit[1]: r_lim_off_en Bit[0]: r_cut_blkline_en

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

address	register name	default value	R/W	description
0x4043	BLC CTRL 43	0x00	RW	Bit[5]: r_man_ln_en Bit[4]: r_output_sel Bit[3]: r_cvdn_blc_en_man Bit[2]: r_cvdn_blc_en Bit[1]: r_dc_blc_en_man Bit[0]: r_dc_blc_en
0x4044	BLC CTRL 44	0x02	RW	Bit[1:0]: r_md_gain_th[9:8]
0x4045	BLC CTRL 45	0x00	RW	Bit[7:0]: r_md_gain_th[7:0]
0x4046	BLC CTRL 46	0x00	RW	Bit[7:5]: r_zln_num Bit[4:0]: r_bln_num
0x4060	BLC CTRL 60	–	R	Bit[1:0]: BLC_offset000[9:8]
0x4061	BLC CTRL 61	–	R	Bit[7:0]: BLC_offset000[7:0]
0x4062	BLC CTRL 62	–	R	Bit[1:0]: BLC_offset001[9:8]
0x4063	BLC CTRL 63	–	R	Bit[7:0]: BLC_offset001[7:0]
0x4064	BLC CTRL 64	–	R	Bit[1:0]: BLC_offset010[9:8]
0x4065	BLC CTRL 65	–	R	Bit[7:0]: BLC_offset010[7:0]
0x4066	BLC CTRL 66	–	R	Bit[1:0]: BLC_offset011[9:8]
0x4067	BLC CTRL 67	–	R	Bit[7:0]: BLC_offset011[7:0]
0x4068	BLC CTRL 68	–	R	Bit[1:0]: BLC_offset100[9:8]
0x4069	BLC CTRL 69	–	R	Bit[7:0]: BLC_offset100[7:0]
0x406A	BLC CTRL 6A	–	R	Bit[1:0]: BLC_offset101[9:8]
0x406B	BLC CTRL 6B	–	R	Bit[7:0]: BLC_offset101[7:0]
0x406C	BLC CTRL 6C	–	R	Bit[1:0]: BLC_offset110[9:8]
0x406D	BLC CTRL 6D	–	R	Bit[7:0]: BLC_offset110[7:0]
0x406E	BLC CTRL 6E	–	R	Bit[1:0]: BLC_offset111[9:8]
0x406F	BLC CTRL 6F	–	R	Bit[7:0]: BLC_offset111[7:0]

4.6 one time programmable (OTP) memory

The OV2740 supports a maximum of 512 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 4k bits (512 bytes), 2k bits are reserved for OmniVision and 2k 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 OV2740 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 0x71FF through SCCB interface using a total of 1k bytes. 0x7000 ~ 0x700F and 0x7110 ~ 0x71FF are reserved for OmniVision, while 0x7010 ~ 0x710F (256 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, OV2740 will load
setting from OTP if 3D85[2:1]=2'b11
```

The OV2740 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~ 0x71FF	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	0x80	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 strobe flash and frame exposure

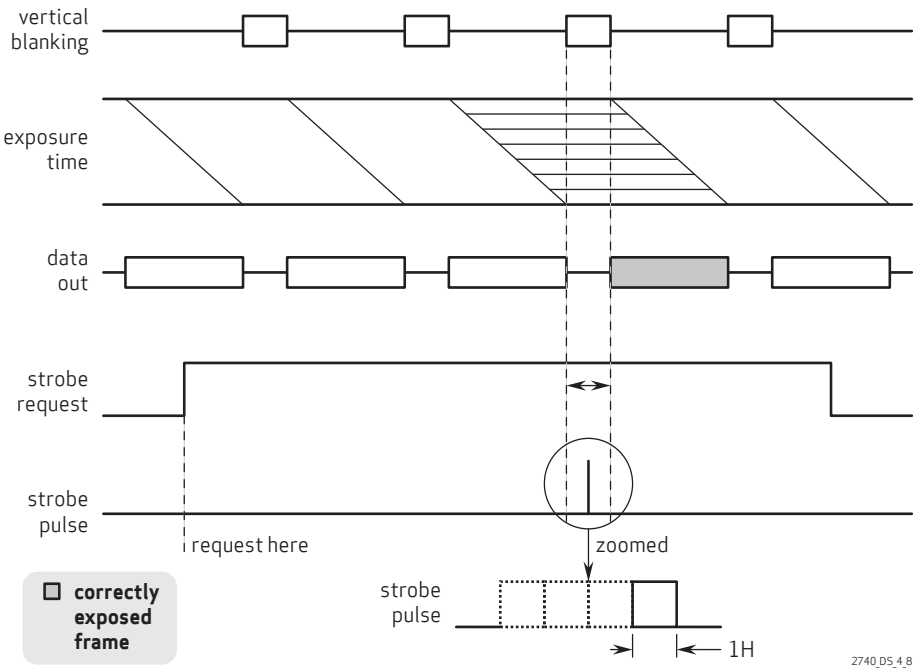
4.7.1 strobe flash control

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.

4.7.1.1 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.

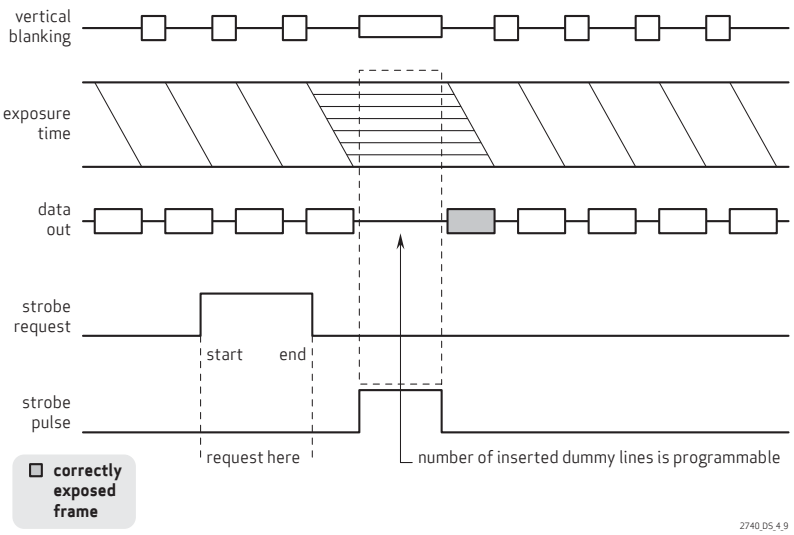
figure 4-8 xenon flash mode



4.7.1.2 LED 1 mode

In LED 1 mode, the strobe signal stays active until the strobe end request is sent (see **figure 4-9**).

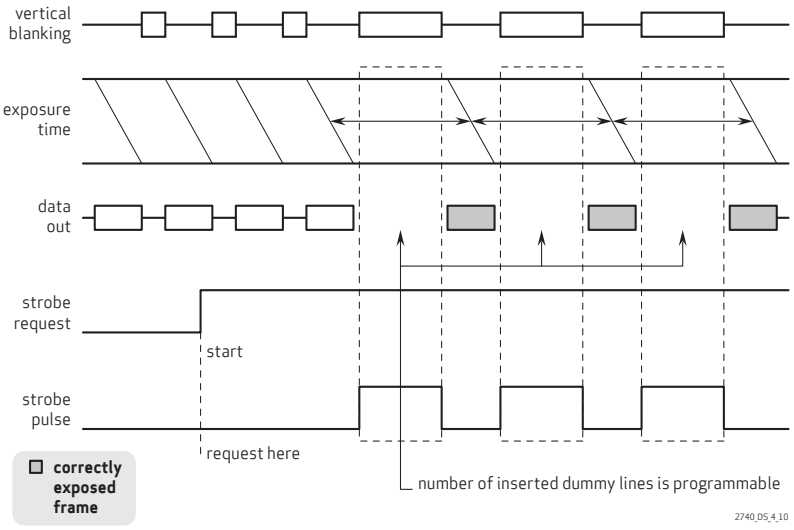
figure 4-9 LED 1 mode



4.7.1.3 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 **figure 4-10**).

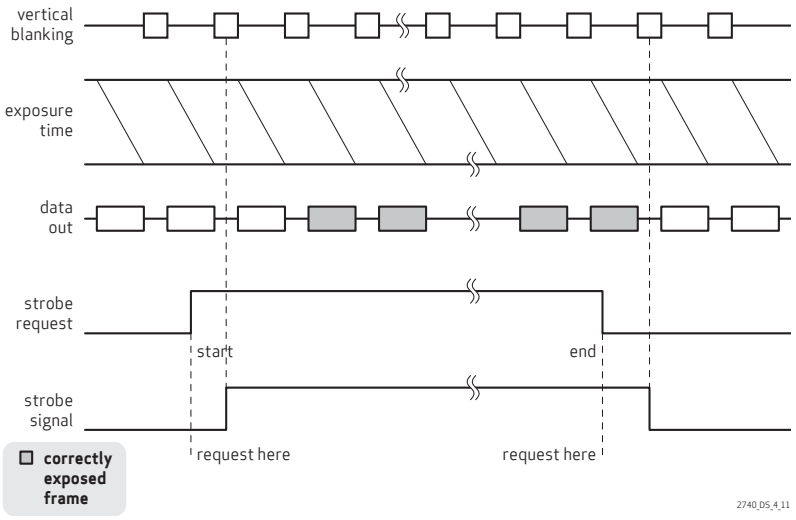
figure 4-10 LED2 mode



4.7.1.4 LED 3 mode

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

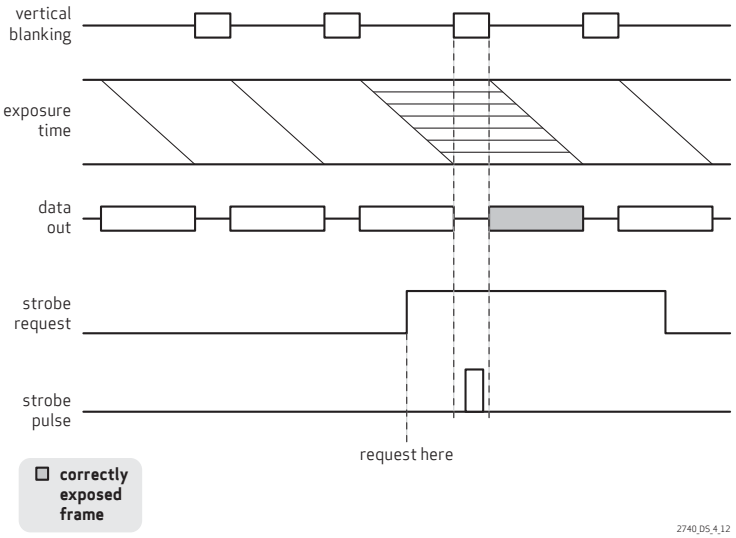
figure 4-11 LED 3 mode



4.7.1.5 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



See **table 4-7** for FREX strobe control functions.

table 4-7 FREX strobe control registers

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
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

4.8 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.

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 4-8 embedded data control registers

address	register name	default value	R/W	description
0x4816	EMBEDED DT	0x53	RW	Bit[5:0]: Embedded line data type

table 4-9 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 4-9 embedded line position data (sheet 1 of 2)

byte number	register name
0	SENSOR INFO
1	RSVD
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	RSVD
10	DPC THRESHOLD[9:2]
11~14	RSVD
15	X_OUTPUT_SIZE[15:8]
16	X_OUTPUT_SIZE[7:0]
17	Y_OUTPUT_SIZE[15:8]
18	Y_OUTPUT_SIZE[7:0]

table 4-9 embedded line position data (sheet 2 of 2)

byte number	register name
19~22	RSVD
23	MIPI HEADER REVISION NUMBER
24~30	RSVD
31	{6'H0, VFLIP, MIRROR}
32	FRAME DURATION A
33	FRAME DURATION B
34	CONTEXT COUNT
35	CONTEXT SELECT
36~53	RSVD
54	DATA PEDESTAL[9:2]
55~62	RSVD
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	RSVD
89	FRAME COUNTER
90~94	RSVD
95	RSVD
96	RSVD

4.9 power saving (PSV) mode

The sensor provides a new operating mode called power saving (PSV) mode. While it features extremely low power consumption, its functionality is very different from conventional video streaming modes. In PSV mode, all digital functions are shut down. The image sensor serves as an ambient light sensor. Therefore, there is no video streaming output.

All the photodiodes are pinned together to become one big passive pixel. It measures photo current and converts it to voltage pulse signals. The more the sensor is exposed to light, the higher the frequency of pulses. A digital logic will take in and count these pulse signals, and generate corresponding interrupts if some user-defined criteria are met.

To enable PSV, pad XSHUTDN2 must be set to high. Also note that SCCB ID in PSV is 0x7C when SID is 0 and 0x30 when SID is 1.

table 4-10 PSV control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x2000	PSV CTRL00	0x20	RW	Bit[0]: PSV enable 0: Disable PSV mode 1: Enable PSV mode
0x2001	PSV CTRL01	0x01	RW	Bit[6:0]: Frame interval Minimum number of frames between two PSV interrupt signals
0x2004	THRESHOLD0 HIGH	0x40	RW	Bit[7:0]: Threshold0[15:8]
0x2005	THRESHOLD0 LOW	0x00	RW	Bit[7:0]: Threshold0[7:0]
0x2006	THRESHOLD1 HIGH	0x00	RW	Bit[7:0]: Threshold1[15:8]
0x2007	THRESHOLD1 LOW	0x00	RW	Bit[7:0]: Threshold1[7:0]
0x2010	PSV CTRL10	0x9A	RW	Bit[3]: Interrupt polarity select 0: Active low 1: Active high

table 4-10 PSV control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x2013	PSV CTRL13	0x00	RW	Bit[7:6]: Interrupt calculation mode select 00: Compare pulse count of current frame with Threshold0 01: Compare absolute difference count against previous frame with Threshold0 10: Combine mode0 and mode1: compare pulse count of current frame with Threshold0 and compare absolute difference count against previous frame with Threshold1 Bit[5:4]: PSV output select 00: Interrupt is a level signal (will be self-cleared) 10: Interrupt is a pulse
0x2016	INTR CLEAR	–	W	Bit[0]: Level interrupt clear

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

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xBF	RW	ISP Control 00 (0: disable; 1: enable) Bit[7]: Lens correction (LENC) function enable Bit[6]: var_pix enable Bit[5]: OTP_DPC function enable Bit[4]: Manual AWB gain enable Bit[3]: Black DPC function enable Bit[2]: White DPC function enable Bit[1]: BLC function enable Bit[0]: ISP enable
0x5001	ISP CTRL01	0x01	RW	Bit[0]: avg_size_man_en
0x5005	DSP CTRL05	0x00	RW	Bit[7]: Pre_awb Average enable Bit[5]: DCBLC enable Bit[0]: Long/short reverse for AWB gain
0x5008	FMT_MUX_CTRL	0x00	RW	Bit[7:0]: fmt_mux_ctrl

5.2 high dynamic range (HDR)

HDR mode increases image dynamic range by capturing multiple exposures of a similar scene and then combining them into one single image.

The OV2740 supports two HDR modes:

- HDR-lite
- staggered HDR

5.2.1 HDR-lite

In HDR-lite mode, long exposure lines and short exposure lines are interlaced in the pixel array (see **figure 5-1**). The exposure time for long/short exposure can be set by registers. The whole image with interlacing exposure time outputs through the MIPI interface. The HDR-combine function operates through the backend chip to combine two half-sized images into a single half sized HDR image. The OV2740 does not support an on-chip HDR combine function.

figure 5-1 HDR-lite exposure lines

table 5-2 ISP top register

address	register name	default value	R/W	description
0x3820	FORMAT	0x00	RW	Bit[0]: hdr_lite_en

To change long/short exposure/gain, please refer to **section 6-3,** AEC control registers.

5.2.2 staggered HDR

In staggered HDR mode, long/short exposure frames overlap each other. The overlapping reduces the timing delay between various exposure frames that combine to form one HDR frame. The overlapping also reduces the frame/line buffer needed for backend sensors. The frame timing for staggered HDR is shown in **figure 5-2**.

The OV2740 uses a MIPI channel to differentiate different exposure frames. Long/short frames use MIPI virtual channel vc0/vc1 respectively, in order for different exposure frames to not mix signals on the MIPI receiver side.

figure 5-2 2 set staggered HDR

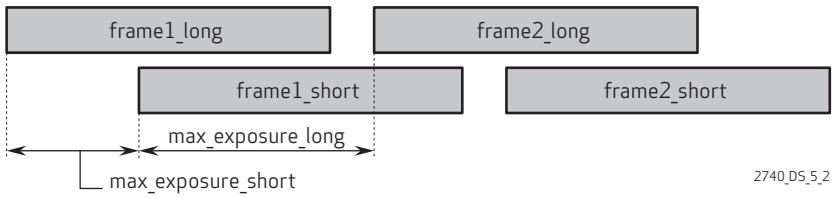


table 5-3 ISP top register

address	register name	default value	R/W	description
0x3829	TIMING_REG_36	0x00	RW	Bit[4]: r_stg_hdr_en

To change long/short exposure/gain, please refer to **section 6-3,** AEC control registers.

5.3 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 images by removing redundant data
- create control signals

table 5-4 pre_DSP registers

address	register name	default value	R/W	description
0x5040	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
0x5041	PRE CTRL01	0x01	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

5.4 defective pixel cancellation (DPC)

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

table 5-5 DPC control register

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xD6	RW	Bit[3]: Black DPC function enable Bit[2]: White DPC function enable

5.5 window cut (WINC)

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

table 5-6 WINC registers

address	register name	default value	R/W	description
0x5980	WINC CTRL00	0x00	RW	Bit[4:0]: X_start offset[12:8]
0x5981	WINC CTRL01	0x10	RW	Bit[7:0]: X_start offset[7:0]
0x5982	WINC CTRL02	0x00	RW	Bit[3:0]: Y_start offset[11:8]
0x5983	WINC CTRL03	0x04	RW	Bit[7:0]: Y_start offset[7:0]
0x5984	WINC CTRL04	0x07	RW	Bit[4:0]: Window width[12:8]
0x5985	WINC CTRL05	0x80	RW	Bit[7:0]: Window width[7:0]
0x5986	WINC CTRL06	0x04	RW	Bit[3:0]: Window height[11:8]
0x5987	WINC CTRL07	0x38	RW	Bit[7:0]: Window height[7:0]
0x5988	WINC CTRL08	0x00	RW	Bit[2]: Select embedded line flag 0: Select first line as embedded flag 1: Select last line as embedded flag Bit[0]: Manual window enable

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

Manual exposure provides exposure time settings and sensor gain. Manual gain provides analog gain settings. For optimal performance, maximum exposure should be 200ms. For more details, contact your local OmniVision FAE.

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

table 5-7 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	0x00	RW	Bit[7:0]: Long exposure[15:8]
0x3502	LONG EXPO	0x40	RW	Bit[7:0]: Long exposure[7:0] Low 4 bits are fraction bits
0x3503	AEC CTRL03	0x88	RW	Bit[7]: agc_manual_en 0: real_gain auto 1: real_gain manual Bit[5]: Gain change delay option 0: Delay 1 frame 1: Do not delay 1 frame Bit[4]: Gain delay option 0: Delay 1 frame 1: Do not delay 1 frame Bit[3]: aec_manual_en 0: aec_auto 1: aec_manual 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[6:4]: 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

table 5-7 MEC/MGC control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x3507	AEC GAIN SHIFT	0x00	RW	Bit[2]: Priority_6 switch_snr_gain_en 0: dac[2:0]_comp[1:0] 1: comp[1:0]_dac[2:0] Bit[1:0]: Gain shift option 00: Do 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]: Gain[12:8]
0x3509	LONG GAIN	0x80	RW	Bit[7:0]: Gain[7:0] If 0x3503[2] = 0, then Gain[12:0] is real gain format, where low 7 bits are fraction bits. real_gain = Gain[12:0]/128 For example, 0x080 is 1x gain, 0x140 is 2.5x gain. Maximum gain is 15.5x; 0x7C0. If 0x3503[2]=1, then Gain[12:0] is sensor gain format, where: Gain[12:8]: Coarse gain 00000: 1x 00001: 2x 00011: 4x 00111: 8x Gain[7]: 1'b1 Gain[6:3]: Fine gain Gain[2:0]: Always 0 For example, 0x080 is 1x gain, 0x180 is 2x gain, and 0x380 is 4x gain.
0x350C	SHORT GAIN	0x00	RW	Bit[5:0]: Short gain[13:8]
0x350D	SHORT GAIN	0x80	RW	Bit[7:0]: Short gain[7:0]
0x3510	SHORT EXPO	0x00	RW	Bit[3:0]: Short exposure[19:16]
0x3511	SHORT EXPO	0x00	RW	Bit[7:0]: Short exposure[15:8]
0x3512	SHORT EXPO	0x20	RW	Bit[7:0]: Short exposure[7:0] Low 4 bits are fraction bits

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.

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.

table 5-8 LENC control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x5000	ISP CTRL0	0x08	RW	Bit[7]: Lens correction (LENC) function enable 0: Disable 1: Enable
0x5800	LENC G00	0x00	RW	Bit[7:0]: Control point G00 for luminance compensation
0x5801	LENC G01	0x00	RW	Bit[7:0]: Control point G01 for luminance compensation
0x5802	LENC G02	0x00	RW	Bit[7:0]: Control point G02 for luminance compensation
0x5803	LENC G03	0x00	RW	Bit[7:0]: Control point G03 for luminance compensation
0x5804	LENC G04	0x00	RW	Bit[7:0]: Control point G04 for luminance compensation
0x5805	LENC G05	0x00	RW	Bit[7:0]: Control point G05 for luminance compensation
0x5806	LENC G06	0x00	RW	Bit[7:0]: Control point G06 for luminance compensation
0x5807	LENC G07	0x00	RW	Bit[7:0]: Control point G07 for luminance compensation
0x5808	LENC G10	0x00	RW	Bit[7:0]: Control point G10 for luminance compensation
0x5809~ 0x584E	LENC G11~ LENC G96	–	RW	Bit[7:0]: Control point G11~G96 for luminance compensation
0x584F	LENC G97	0x80	RW	Bit[7:0]: Control point G97 for luminance compensation
0x5850	LENC B00	0x80	RW	Bit[7:0]: Control point B00 for blue channel compensation
0x5851	LENC B01	0x80	RW	Bit[7:0]: Control point B01 for blue channel compensation

table 5-8 LENC control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x5852	LENC B02	0x80	RW	Bit[7:0]: Control point B02 for blue channel compensation
0x5853	LENC B03	0x80	RW	Bit[7:0]: Control point B03 for blue channel compensation
0x5854	LENC B04	0x80	RW	Bit[7:0]: Control point B04 for blue channel compensation
0x5855	LENC B05	0x80	RW	Bit[7:0]: Control point B05 for blue channel compensation
0x5856	LENC B06	0x80	RW	Bit[7:0]: Control point B06 for blue channel compensation
0x5857	LENC B07	0x80	RW	Bit[7:0]: Control point B07 for blue channel compensation
0x5858	LENC B10	0x80	RW	Bit[7:0]: Control point B10 for blue channel compensation
0x5859~ 0x589E	LENC B11~LENC B96	–	RW	Bit[7:0]: Control point B11~B96 for blue channel compensation
0x589F	LENC B97	0x80	RW	Bit[7:0]: Control point B97 for blue channel compensation
0x58A0	LENC R00	0x80	RW	Bit[7:0]: Control point R00 for red channel compensation
0x58A1	LENC R01	0x80	RW	Bit[7:0]: Control point R01 for red channel compensation
0x58A2	LENC R02	0x80	RW	Bit[7:0]: Control point R02 for red channel compensation
0x58A3	LENC R03	0x80	RW	Bit[7:0]: Control point R03 for red channel compensation
0x58A4	LENC R04	0x80	RW	Bit[7:0]: Control point R04 for red channel compensation
0x58A5	LENC R05	0x80	RW	Bit[7:0]: Control point R05 for red channel compensation
0x58A6	LENC R06	0x80	RW	Bit[7:0]: Control point R06 for red channel compensation
0x58A7	LENC R07	0x80	RW	Bit[7:0]: Control point R07 for red channel compensation
0x58A8	LENC R10	0x80	RW	Bit[7:0]: Control point R10 for red channel compensation
0x58A9~ 0x58EE	LENC R11~LENC R96	–	RW	Bit[7:0]: Control point R11~R96 for red channel compensation

table 5-8 LENC control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x58EF	LENC R97	0x80	RW	Bit[7:0]: Control point R97 for red channel compensation
0x58F5	LENC HSCALE	0x04	RW	Bit[4:0]: HScale[12:8] For horizontal gain calculation, this value indicates the step between two connected horizontal pixels, where HScale = 4*2^18 / image width
0x58F6	LENC HSCALE	0x3B	RW	Bit[7:0]: HScale[7:0]
0x58F7	LENC VSCALE	0x02	RW	Bit[4:0]: VScale[12:8] For vertical gain calculation, this value indicates the step between two connected vertical pixels, where VScale = 4*2^17 / image height
0x58F8	LENC VSCALE	0xCD	RW	Bit[7:0]: VScale[7:0]

6 register tables

The following tables provide descriptions of the device control registers contained in the OV2740. 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 - 0x0313, 0x031B - 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	0x40	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_divp[9:8]
0x0302	PLL_CTRL_2	0x7D	RW	Bit[7:0]: pll1_dvp[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
0x0307	RSVD	–	–	Reserved
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	0x06	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

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

address	register name	default value	R/W	description
0x0311	PLL_CTRL_11	0x00	RW	Bit[7:1]: Not used Bit[0]: pll2_bypass
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
0x0313	PLL_CTRL_13	0x04	RW	Bit[4]: pll2_div_rst Bit[3:0]: pll2_divsram
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
0x031D	RSVD	–	–	Reserved
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 PSV control [0x2000 - 0x2013, 0x2016 - 0x2018, 0x2020 - 0x2029]

The SCCB ID in PSV is 0x7C when SID is 0, and 0x30 when SID is 1.

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

address	register name	default value	R/W	description
0x2000	PSV CTRL00	0x20	RW	Bit[7:6]: Debug mode Bit[5]: level_adj_enable Bit[4]: merge_avg Bit[3]: merge_en Bit[2]: Reserved Bit[1]: sel_abs_diff 0: Using absolute level of current frames 1: Using absolute level difference between current and previous frames Bit[0]: psv_en

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

address	register name	default value	R/W	description
0x2001	PSV CTRL01	0x01	RW	Bit[7]: clk_sel_o Bit[6:0]: frame_interval Minimum number of frames between two PSV interrupt signal
0x2002	PSV CTRL02	0x00	RW	Bit[7:0]: thres_man
0x2003	PSV CTRL03	0x00	RW	Bit[7:0]: thres_man
0x2004	THRESHOLD0 HIGH	0x40	RW	Bit[7:0]: threshold0[15:8]
0x2005	THRESHOLD0 LOW	0x00	RW	Bit[7:0]: threshold0[7:0]
0x2006	THRESHOLD1 HIGH	0x00	RW	Bit[7:0]: threshold1[15:8]
0x2007	THRESHOLD1 LOW	0x00	RW	Bit[7:0]: threshold1[7:0]
0x2008	PSV CTRL08	0x00	RW	Bit[7:1]: Not used Bit[0]: thres_man_en
0x2009	STEP NORM	0x00	RW	Bit[7:0]: step_norm
0x200A	LM SMALL0	0xFF	RW	Bit[7:0]: lm_small[15:8] Small zone light level boundary
0x200B	LM SMALL1	0xFF	RW	Bit[7:0]: lm_small[7:0] Small zone light level boundary
0x200C	LM LARGE0	0xFF	RW	Bit[7:0]: lm_large[15:8] Small zone light level boundary
0x200D	LM LARGE1	0xFF	RW	Bit[7:0]: lm_large[7:0] Small zone light level boundary
0x200E	ANA REG0	0x82	RW	Bit[7:0]: reg_ana[15:8]
0x200F	ANA REG1	0x10	RW	Bit[7:0]: reg_ana[7:0]
0x2010	PSV CTRL10	0x9A	RW	Bit[7]: Interrupt enable Bit[6:4]: wait_reading_en Bit[3]: Interrupt polarity 0: Active low 1: Active high Bit[2]: rdy_mask_auto_only Bit[1]: out_of_range_intr_en Bit[0]: host_clear_frm
0x2011	PSV CTRL11	0x02	RW	Bit[7]: rdy_opt Bit[6:0]: mask_period Mask first N frames after reset

table 6-2 PSV control registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x2012	PSV CTRL12	0x01	RW	Bit[7:0]: intr_num
0x2013	PSV CTRL13	0x00	RW	Bit[7:6]: Interrupt calculation mode select 00: Compare light level count of current frame 01: Compare absolute difference against previous frame 10: Combine mode0 and mode1 11: Compare change percentage against previous frame Bit[5:4]: psv_output_sel 00: Level interrupt (last one frame (65536 cycles)) 01: out_range (debug) 10: Pulse interrupt 11: lm_in1_i (debug) Bit[3:0]: Debug mode
0x2016	PSV INTR CLEAR	–	W	Bit[7:1]: Debug mode Bit[0]: PSV interrupt clear
0x2017	PSV CTRL17	0xC4	RW	Bit[7:4]: level_adj_ht Higher threshold for automatic level adjustment Bit[3:0]: level_adj_lt Lower threshold for automatic level adjustment
0x2018	PSV CTRL18	0x00	RW	Bit[7:4]: Debug mode Bit[3:1]: psv_pulse_length PSV pulse width = Nx16 clock cycles Bit[0]: als_pulse_sel 0: Detect negative edge of als_trigger 1: Detect positive edge of als_trigger
0x2020	RO LEVEL ADJ	–	R	Bit[7]: Not used Bit[6:2]: level_adj Bit[1]: lm_in2 Bit[0]: lm_in1
0x2021	RO LM1 COUNT0	–	R	Bit[7:0]: lm1_count[15:8]
0x2022	RO LM1 COUNT1	–	R	Bit[7:0]: lm1_count[7:0]
0x2022	RO LM2 COUNT0	–	R	Bit[7:0]: lm2_count[15:8]

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

address	register name	default value	R/W	description
0x2023	RO LM2 COUNT1	–	R	Bit[7:0]: lm2_count[7:0]
0x2024	RSVD	–	–	Reserved
0x2025	RO LM1 DIFF0	–	R	Bit[7:0]: lm1_diff[15:8]
0x2026	RO LM1 DIFF1	–	R	Bit[7:0]: lm1_diff[7:0]
0x2027	RO LM2 DIFF0	–	R	Bit[7:0]: lm2_diff[15:8]
0x2028	RO LM2 DIFF1	–	R	Bit[7:0]: lm2_diff[7:0]
0x2029	RO PSV CTRL29	–	R	Bit[7:4]: Debug mode Bit[3]: out_range_i Bit[2]: psv_trig_i Bit[1]: psv_trig2_i Bit[0]: psv_trig1_i

6.3 AEC control [0x3A01 - 0x3A07, 0x3A0F]

table 6-3 AEC control registers

address	register name	default value	R/W	description
0x3A01	AEC CTRL01	0x01	RW	Bit[7]: Debug mode Bit[6:4]: long_short_ration Bit[3:0]: min_expo
0x3A02	AEC CTRL02	0x78	RW	Bit[7:0]: wpt[7:0]
0x3A03	AEC CTRL03	0x68	RW	Bit[7:0]: bpt[7:0]
0x3A04	AEC CTRL04	0x00	RW	Bit[7:5]: Debug mode Bit[4:0]: max_expo[12:8]
0x3A05	AEC CTRL05	0xF0	RW	Bit[7:0]: max_expo[7:0]
0x3A07	AEC CTRL07	0xF8	RW	Bit[7:0]: gain_ceil[7:0]
0x3A0F	AEC CTRL0F	–	R	Bit[7:3]: Debug mode Bit[2]: inc_cur Bit[1]: dec_cur Bit[0]: bal_cur

6.4 system control [0x3000 - 0x300D, 0x3010 - 0x3024, 0x302A, 0x3030 - 0x303F]

table 6-4 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
0x3001	RSVD	–	–	Reserved
0x3002	PAD OEN2	0x20	RW	Bit[7:4]: Reserved Bit[3]: io_strobe_oen Bit[2]: io_sda_oen Bit[1]: Reserved Bit[0]: io_gpio_oen
0x3003	RSVD	–	–	Reserved
0x3004	SCCB ID	0x6C	RW	Bit[7:0]: sccb_id SCCB programmed ID
0x3005	CLKRST5	0xF0	RW	Bit[7:6]: Not used Bit[5]: sclk_src Bit[4]: sclk_syncfifo Bit[3:2]: Not used Bit[1]: rst_src Bit[0]: rst_syncfifo
0x3006	SCCB ID2	0x42	RW	Bit[7:0]: sccb_id2 SCCB ID2
0x3007	R ISPOUT BITSEL	0x00	RW	Bit[7]: pll12_daclk_sel Bit[6]: r_pump_clk_sel Bit[5]: r_fc_bypass Bit[4]: Reserved Bit[3]: r_rst_pll_sleep_dis Bit[2]: r_db_out_en Bit[1:0]: r_vsync_sel 00: mipi_vsync 01: lvds_vsync 10: fmt_vsync 11: tc_vsync
0x3008~ 0x3009	RSVD	–	–	Reserved
0x300A	CHIP ID	0x00	R	Bit[7:0]: chip_id[23:16]
0x300B	CHIP ID	0x27	R	Bit[7:0]: chip_id[15:8]
0x300C	CHIP ID	0x40	R	Bit[7:0]: chip_id[7:0]

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

address	register name	default value	R/W	description
0x300D	PAD OUT2	0x00	RW	Bit[7]: io_vsync_o Bit[6:4]: Reserved 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[7:0]: Not used
0x3012	SCCB R12	0x20	RW	Bit[7:0]: sccb_id_r12
0x3013~ 0x3014	RSVD	–	–	Reserved
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
0x3016~ 0x3017	RSVD	–	–	Reserved

table 6-4 system control registers (sheet 3 of 6)

address	register name	default value	R/W	description
0x3018	MIPI SC CTRL	0x32	RW	Bit[7:5]: mipi_lane_mode N+1 lane Bit[4]: mipi_en 0: LVDS enable 1: MIPI enable Bit[3:2]: r_phy_pd_mipi Bit[1]: phy_rst option 1: Reset PHY when rst_sync 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]: sclk_gt 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	0xD0	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
0x301C	CLKRST2	0xF0	RW	Bit[7]: Not used Bit[6]: sclk_mipi Bit[5]: sclk_psv_ctrl Bit[4]: sclk_otp Bit[3]: Not used Bit[2]: rst_mipi Bit[1]: rst_psv_ctrl Bit[0]: rst_otp
0x301D	CLKRST3	0xD0	RW	Bit[7]: sclk_asram_tst Bit[6]: sclk_grp Bit[5]: sclk_bist Bit[4]: sclk_aec Bit[3]: rst_asram_tst Bit[2]: rst_grp Bit[1]: rst_bist Bit[0]: rst_aec

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

address	register name	default value	R/W	description
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	RSVD	–	–	Reserved
0x3020	CLOCK SEL	0x93	RW	Bit[7]: Clock switch output Bit[6]: pclk_ratio_exp 1: Exponential pclk_ratio_i to 2^n Bit[5]: yuv_out_en 0: SOC sensor, output RAW dat,or RAW sensor 1: SOC sensor, output YUV dat Bit[4]: dvp_sclk_en 1: Use pll_sclk_i instead pll_pclk_i for DVP Bit[3]: pclk_div 0: /1 1: /2 Bit[2:1]: Not used Bit[0]: sclk2x_sel
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]: mipi_ctr_en 0: Disable the function 1: Enable MIPI remote reset and suspend control sc Bit[3]: mipi_rst_sel 0: MIPI remote reset all registers 1: MIPI remote reset all digital module Bit[2]: gpio_pclk_en Bit[1]: frex_ef_sel Bit[0]: cen_global_o
0x3022	MIPI SC CTRL	0x01	RW	Bit[7:4]: Not used Bit[3]: lvds_mode_o Bit[2]: Clock lane disable 1 Bit[1]: kc lane disable 0 Bit[0]: pd_mipi enable when rst_sync
0x3023	MIPI LPTX SEL	0x00	RW	Bit[7:0]: Not used

table 6-4 system control registers (sheet 5 of 6)

address	register name	default value	R/W	description
0x3024	REG24	0x10	RW	Bit[7:3]: Not used Bit[2]: rst_ana Bit[1:0]: Not used
0x302A	SUB ID	–	R	Bit[7:4]: Process Bit[3:0]: Version
0x3030	REG30	0x00	RW	Bit[7:5]: Not used 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 0xC: 12-bit mode Others: Not used
0x3032	REG32	0x80	RW	Bit[7]: Not used Bit[6]: asram_clk_sel Bit[5]: Not used Bit[4]: r_rst_otp_sleep_dis Bit[3]: r_rst_ana_sleep_dis Bit[2:0]: Not used
0x3033	REG33	0x24	RW	Bit[7]: Debug mode Bit[6]: emb_ana_gain_sel Bit[5]: r_fmt_eof_sel Bit[4]: sync_point_sel Bit[3]: rip_sof_en Bit[2]: mipi_sel_aslp_dis Bit[1]: Debug mode Bit[0]: lvds_ck_data_sel
0x3034	REG34	0x00	RW	Bit[7]: bit_shift_clip_en Bit[6:4]: bit_shift_mode Bit[3]: sw_stb_rdy_sel Bit[2]: vsync_out_sel Bit[1]: bp_half Bit[0]: mipirx_pclk_sel
0x3035	REG35	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: ramp_vref_high
0x3036	REG36	0x2B	RW	Bit[7:6]: Reserved Bit[5:0]: emb_dt
0x3037	PLL1 CTR1	0x00	RW	Bit[7:1]: Reserved Bit[0]: r_sid
0x3038~ 0x303D	RSVD	–	–	Reserved

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

address	register name	default value	R/W	description
0x303E	ENCLK_SEL	0x03	RW	Bit[7:3]: Not used Bit[2:0]: r_enclk_sel_o
0x303F	CTRL3F	0x00	RW	Bit[7:2]: Not used Bit[1]: sccb_id2_nack Bit[0]: sccb_pgm_id_en

6.5 SCCB control [0x3100 - 0x3107]

table 6-5 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
0x3107	SC_CTRL	0x01	RW	Bit[7]: Reserved Bit[6]: npump_clk_sw Bit[5]: auto_sleep_en Bit[4]: pd_mipi_dis_aslp Bit[3]: pumpclk_cutoff_byp Bit[2]: pclk_cutoff_byp Bit[1]: clk_cutoff_byp Bit[0]: pd_ana_vbk_arb

6.6 group hold [0x3200 - 0x320F]

table 6-6 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	0x08	RW	Group1 Start Address in SRAM, Actual Address is {0x3201[5:0], 4'h0}
0x3202	GROUP ADR2	0x10	RW	Group2 Start Address in SRAM, Actual Address is {0x3202[5:0], 4'h0}
0x3203	GROUP ADR3	0x18	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	0x02	RW	Bit[7:5]: Not used Bit[4]: Group hold SRAM test enable Bit[3:0]: Group hold SRAM RM[3:0]

table 6-6 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

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

table 6-7 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	0x88	RW	Bit[7]: agc_manual_en 0: real_gain auto 1: real_gain manual Bit[6]: Debug mode Bit[5]: Gain change delay option 0: Delay 1 frame 1: Do not delay 1 frame Bit[4]: Gain delay option 0: Delay 1 frame 1: Do not delay 1 frame Bit[3]: aec_manual_en 0: aec_auto 1: aec_manual 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
0x3504	RSVD	–	–	Reserved

table 6-7 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[6:4]: 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
0x3506	RSVD	–	–	Reserved
0x3507	GAIN SHIFT	0x00	RW	Bit[7:3]: Not used Bit[2]: Priority_6 switch_snr_gain_en 0: dac[2:0]_comp[1:0] 1: comp[1:0]_dac[2:0] 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]: Gain[12:8]
0x3509	LONG GAIN	0x80	RW	Bit[7:0]: Gain[7:0] If 0x3503[2] = 0, then Gain[12:0] is real gain format, where low 7 bits are fraction bits. real_gain = Gain[12:0]/128 For example, 0x080 is 1x gain, 0x140 is 2.5x gain. Maximum gain is 15.5x; 0x7C0. If 0x3503[2]=1, then Gain[12:0] is sensor gain format, where: Gain[12:8]: Coarse gain 00000: 1x 00001: 2x 00011: 4x 00111: 8x Gain[7]: 1'b1 Gain[6:3]: Fine gain Gain[2:0]: Always 0 For example, 0x080 is 1x gain, 0x180 is 2x gain, and 0x380 is 4x gain.
0x350A~ 0x350B	RSVD	–	–	Reserved
0x350C	SHORT GAIN	0x00	RW	Bit[7:6]: Not used Bit[5:0]: Short gain[13:8]
0x350D	SHORT GAIN	0x80	RW	Bit[7:0]: Short gain[7:0]

table 6-7 MEC/MGC control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x350E~ 0x350F	RSVD	–	–	Reserved
0x3510	SHORT EXPO	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Short exposure[19:16]
0x3511	SHORT EXPO	0x00	RW	Bit[7:0]: Short exposure[15:8]
0x3512	SHORT EXPO	0x20	RW	Bit[7:0]: Short exposure[7:0] Low 4 bits are fraction bits

6.8 analog control [0x3600 - 0x366F]

table 6-8 analog control registers

address	register name	default value	R/W	description
0x3600~ 0x366F	ANALOG CONTROL	–	–	Analog Control Registers

6.9 sensor control [0x3700 - 0x37CF]

table 6-9 sensor control registers

address	register name	default value	R/W	description
0x3700~ 0x37CF	SENSOR CONTROL	–	–	Sensor Control Registers

6.10 timing control [0x3800 - 0x381B, 0x3820 - 0x3828, 0x382A - 0x382B]

table 6-10 timing control registers (sheet 1 of 3)

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	0x04	RW	Bit[7:0]: y_addr_start[7:0] Array vertical start point low byte
0x3804	X ADDR END	0x0A	RW	Bit[7:4]: Not used Bit[3:0]: x_addr_end[11:8] Array horizontal end point high byte
0x3805	X ADDR END	0x33	RW	Bit[7:0]: x_addr_end[7:0] Array horizontal end point low byte
0x3806	Y ADDR END	0x07	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	0x0A	RW	Bit[7:4]: Not used Bit[3:0]: x_output_size[11:8] ISP horizontal output width high byte
0x3809	X OUTPUT SIZE	0x20	RW	Bit[7:0]: x_output_size[7:0] ISP horizontal output width low byte
0x380A	Y OUTPUT SIZE	0x07	RW	Bit[7:4]: Not used Bit[3:0]: y_output_size[11:8] ISP vertical output height high byte
0x380B	Y OUTPUT SIZE	0x98	RW	Bit[7:0]: y_output_size[7:0] ISP vertical output height low byte
0x380C	HTS	0x05	RW	Bit[7:0]: HTS[15:8] Total horizontal timing size high byte
0x380D	HTS	0xEC	RW	Bit[7:0]: HTS[7:0] Total horizontal timing size low byte
0x380E	VTS	0x08	RW	Bit[7:0]: VTS[15:8] Total vertical timing size high byte

table 6-10 timing control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x380F	VTS	0x40	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 ODD INC	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Horizontal increase number at odd pixel
0x3815	X EVEN INC	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Horizontal increase number at even pixel
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	0x80	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

table 6-10 timing control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x3821	TIMING FORMAT2	0x08	RW	Format2 Bit[7]: dig_hbin4 Bit[6:3]: Debug mode Bit[2]: mirror_dig Bit[1]: mirror_arr Bit[0]: dig_hbin2
0x3822	REG22	0x48	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]: Vertical binning Bit[4]: init_man Bit[3:1]: r_grp_adj Bit[0]: Horizontal binning
0x3824	CS RST FSIN	0x00	RW	Bit[7:0]: cs_rst_fsin[15:8] CS reset value high byte at vs_ext
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 ODD INC	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Vertical increase number at odd row
0x382B	Y EVEN INC	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Vertical increase number at even row

6.11 strobe [0x3B00, 0x3B02 - 0x3B05]

table 6-11 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

6.12 OTP control [0x3D80 - 0x3D91]

table 6-12 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	0xAA	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	0x80	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 6-12 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

6.13 PSRAM control [0x3F00 - 0x3F0F]

table 6-13 PSRAM control registers

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

6.14 BLC control [0x4000 - 0x4046, 0x4060 - 0x406F]

table 6-14 BLC control registers (sheet 1 of 6)

address	register name	default value	R/W	description
0x4000	BLC CTRL00	0xCF	RW	Bit[7:4]: avg_weight Bit[3]: target_adj_dis Bit[2]: cmp_en Bit[1]: dither_en Bit[0]: mf_en

table 6-14 BLC control registers (sheet 2 of 6)

address	register name	default value	R/W	description
0x4001	BLC CTRL01	0x20	RW	Bit[7:6]: r_hdr_option Bit[5]: kcoef_man_en Bit[4]: off_man_en Bit[3]: zero_line_out_en Bit[2]: blk_line_out_en Bit[1:0]: byp_mode
0x4002	BLC CTRL02	0x00	RW	Bit[7:2]: Debug mode Bit[1:0]: blk_lvl_target[9:8]
0x4003	BLC CTRL03	0x10	RW	Bit[7:0]: blk_lvl_target[7:0]
0x4004	BLC CTRL04	0x00	RW	Bit[7:4]: Debug mode Bit[3:0]: hwin_off[11:8]
0x4005	BLC CTRL05	0x02	RW	Bit[7:0]: hwin_off[7:0]
0x4006	BLC CTRL06	0x00	RW	Bit[7:4]: Debug mode Bit[3:0]: hwin_pad[11:8]
0x4007	BLC CTRL07	0x10	RW	Bit[7:0]: hwin_pad[7:0]
0x4008	BLC CTRL08	0x00	RW	Bit[7:0]: bl_start
0x4009	BLC CTRL09	0x0B	RW	Bit[7:0]: bl_end
0x400A	BLC CTRL0A	0xFF	RW	Bit[7:0]: off_lim_th[15:8]
0x400B	BLC CTRL0B	0xFF	RW	Bit[7:0]: off_lim_th[7:0]
0x400C	BLC CTRL0C	0x00	RW	Bit[7:0]: cvdn_bl_start
0x400D	BLC CTRL0D	0x00	RW	Bit[7:0]: cvdn_bl_end
0x400E	BLC CTRL0E	0x00	RW	Bit[7:2]: Debug mode Bit[1:0]: kcoef_man[9:8]
0x400F	BLC CTRL0F	0x80	RW	Bit[7:0]: kcoef_man[7:0]
0x4010	BLC CTRL10	0x60	RW	Bit[7]: off_trig_en Bit[6]: gain_chg_trig_en Bit[5]: fmt_chg_trig_en Bit[4]: rst_trig_en Bit[3]: man_avg_en Bit[2]: man_trig Bit[1]: off_frz_en Bit[0]: off_always up

table 6-14 BLC control registers (sheet 3 of 6)

address	register name	default value	R/W	description
0x4011	BLC CTRL11	0x70	RW	Bit[7]: Debug mode Bit[6]: off_chg_mf_en Bit[5]: fmt_chg_mf_en Bit[4]: gain_chg_mf_en Bit[3]: rst_mf_mode Bit[2]: off_chg_mf_mode Bit[1]: fmt_chg_mf_mode Bit[0]: gain_chg_mf_mode
0x4012	BLC CTRL12	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: rst_trig_fn
0x4013	BLC CTRL13	0x02	RW	Bit[7:6]: Reserved Bit[5:0]: fmt_trig_fn
0x4014	BLC CTRL14	0x02	RW	Bit[7:6]: Reserved Bit[5:0]: gain_trig_fn
0x4015	BLC CTRL15	0x02	RW	Bit[7:6]: Reserved Bit[5:0]: off_trig_fn
0x4016	BLC CTRL16	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: off_trig_th[9:8]
0x4017	BLC CTRL17	0x00	RW	Bit[7:0]: off_trig_th[7:0]
0x4018~ 0x401F	RSVD	–	–	Reserved
0x4020	BLC CTRL20	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_th000
0x4021	BLC CTRL21	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_k000
0x4022	BLC CTRL22	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_th001
0x4023	BLC CTRL23	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_k001
0x4024	BLC CTRL24	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_th010
0x4025	BLC CTRL25	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_k010
0x4026	BLC CTRL26	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_th011
0x4027	BLC CTRL27	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_k011
0x4028	BLC CTRL28	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_th100

table 6-14 BLC control registers (sheet 4 of 6)

address	register name	default value	R/W	description
0x4029	BLC CTRL29	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_k100
0x402A	BLC CTRL2A	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_th101
0x402B	BLC CTRL2B	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_k101
0x402C	BLC CTRL2C	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_th110
0x402D	BLC CTRL2D	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_k110
0x402E	BLC CTRL2E	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_th111
0x402F	BLC CTRL2F	0x00	RW	Bit[7:6]: Reserved Bit[5:0]: off_cmp_k111
0x4030	BLC CTRL3	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: off_man000[9:8]
0x4031	BLC CTRL31	0x00	RW	Bit[7:0]: off_man000[7:0]
0x4032	BLC CTRL32	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: off_man001[9:8]
0x4033	BLC CTRL33	0x00	RW	Bit[7:0]: off_man001[7:0]
0x4034	BLC CTRL34	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: off_man010[9:8]
0x4035	BLC CTRL35	0x00	RW	Bit[7:0]: off_man010[7:0]
0x4036	BLC CTRL36	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: off_man011[9:8]
0x4037	BLC CTRL37	0x00	RW	Bit[7:0]: off_man01[7:0]
0x4038	BLC CTRL38	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: off_man100[9:8]
0x4039	BLC CTRL39	0x00	RW	Bit[7:0]: off_man100[7:0]
0x403A	BLC CTRL3A	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: off_man101[9:8]
0x403B	BLC CTRL3B	0x00	RW	Bit[7:0]: off_man101[7:0]
0x403C	BLC CTRL3C	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: off_man110[9:8]
0x403D	BLC CTRL3D	0x00	RW	Bit[7:0]: off_man110[7:0]

table 6-14 BLC control registers (sheet 5 of 6)

address	register name	default value	R/W	description
0x403E	BLC CTRL3E	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: off_man111[9:8]
0x403F	BLC CTRL3F	0x00	RW	Bit[7:0]: off_man111[7:0]
0x4040	BLC CTRL40	0x00	RW	Bit[7:5]: Reserved Bit[4:0]: r_mini_row_beta
0x4041	BLC CTRL41	0x00	RW	Bit[7:5]: Reserved Bit[4:0]: r_mini_row_slop
0x4042	BLC CTRL42	0x10	RW	Bit[7]: r_format_trig_beh Bit[6]: r_gain_trig_beh Bit[5]: r_stagger_ls_man Bit[4]: r_blk_col_out_en Bit[3:2]: r_manu_cvdn_out_en Bit[1]: r_lim_off_en Bit[0]: r_cut_blkline_en
0x4043	BLC CTRL43	0x00	RW	Bit[7:6]: Reserved Bit[5]: r_man_ln_en Bit[4]: r_output_sel Bit[3]: r_cvdn_blc_en_man Bit[2]: r_cvdn_blc_en Bit[1]: r_dc_blc_en_man Bit[0]: r_dc_blc_en
0x4044	BLC CTRL44	0x02	RW	Bit[7:2]: Reserved Bit[1:0]: r_rnd_gain_th[9:8]
0x4045	BLC CTRL45	0x00	RW	Bit[7:0]: r_rnd_gain_th[7:0]
0x4046	BLC CTRL46	0x00	RW	Bit[7:5]: r_zln_num Bit[4:0]: r_bln_num
0x4060	BLC CTRL60	–	R	Bit[7:2]: Reserved Bit[1:0]: bline_offset_c000[9:8]
0x4061	BLC CTRL61	–	R	Bit[7:0]: bline_offset_c000[7:0]
0x4062	BLC CTRL62	–	R	Bit[7:2]: Reserved Bit[1:0]: bline_offset_c001[9:8]
0x4063	BLC CTRL63	–	R	Bit[7:0]: bline_offset_c001[7:0]
0x4064	BLC CTRL64	–	R	Bit[7:2]: Reserved Bit[1:0]: bline_offset_c010[9:8]
0x4065	BLC CTRL65	–	R	Bit[7:0]: bline_offset_c010[7:0]
0x4066	BLC CTRL66	–	R	Bit[7:2]: Reserved Bit[1:0]: bline_offset_c011[9:8]
0x4067	BLC CTRL67	–	R	Bit[7:0]: bline_offset_c011[7:0]

table 6-14 BLC control registers (sheet 6 of 6)

address	register name	default value	R/W	description
0x4068	BLC CTRL68	–	R	Bit[7:2]: Reserved Bit[1:0]: bline_offset_c100[9:8]
0x4069	BLC CTRL69	–	R	Bit[7:0]: bline_offset_c100[7:0]
0x406A	BLC CTRL6A	–	R	Bit[7:2]: Reserved Bit[1:0]: bline_offset_c101[9:8]
0x406B	BLC CTRL6B	–	R	Bit[7:0]: bline_offset_c101[7:0]
0x406C	BLC CTRL6C	–	R	Bit[7:2]: Reserved Bit[1:0]: bline_offset_c110[9:8]
0x406D	BLC CTRL6D	–	R	Bit[7:0]: bline_offset_c110[7:0]
0x406E	BLC CTRL6E	–	R	Bit[7:2]: Reserved Bit[1:0]: bline_offset_c111[9:8]
0x406F	BLC CTRL6F	–	R	Bit[7:0]: bline_offset_c111[7:0]

6.15 frame control [0x4200 - 0x4203]

table 6-15 frame control registers (sheet 1 of 2)

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

table 6-15 frame control registers (sheet 2 of 2)

address	register name	default value	R/W	description
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

6.16 format control [0x4300 - 0x4302, 0x4316, 0x4320 - 0x4329]

table 6-16 format control registers (sheet 1 of 2)

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[7:0]: r_seof_vsync_delay[7:0]
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	RSVD	–	–	Reserved
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]

table 6-16 format control registers (sheet 2 of 2)

address	register name	default value	R/W	description
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]

6.17 MIPI control [0x4800 - 0x483D, 0x484A - 0x4851]

table 6-17 MIPI control registers (sheet 1 of 8)

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

table 6-17 MIPI control registers (sheet 2 of 8)

address	register name	default value	R/W	description
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
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 Bit[2]: r_manu_half2one t_period half to 1 Bit[1]: clk_pre_half Bit[0]: hs_pre_half

table 6-17 MIPI control registers (sheet 3 of 8)

address	register name	default value	R/W	description
0x4804	MIPI CTRL04	0x04	RW	Bit[7:4]: man_lane_num Bit[3]: lane_num_manaul_enable Bit[2]: Debug mode Bit[1]: Vsub select 0: Valid in behind 1: Valid in front Bit[0]: vfifo_8x
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	0x00	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 Power saving T_lpx
0x4808	MIPI CTRL08	0x1A	RW	Bit[7:0]: wkup_dly Mark1 wakeup delay/2^10
0x4809~ 0x480F	RSVD	–	–	Reserved
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
0x4812	RSVD	–	–	Reserved
0x4813	MIPI CTRL13	0x00	RW	Bit[7:3]: Not used Bit[2]: vc_sel Bit[1:0]: VC ID

table 6-17 MIPI control registers (sheet 4 of 8)

address	register name	default value	R/W	description
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
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
0x4817	RSVD	–	–	Reserved
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

table 6-17 MIPI control registers (sheet 5 of 8)

address	register name	default value	R/W	description
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
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

table 6-17 MIPI control registers (sheet 6 of 8)

address	register name	default value	R/W	description
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
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
0x4834~ 0x4836	RSVD	–	–	Reserved
0x4837	PCLK PERIOD	0x1A	RW	Bit[7:0]: pclk_period[7:0] Period of pclk2x, pclk_div=1, and 1 bit decimal

table 6-17 MIPI control registers (sheet 7 of 8)

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~ 0x483B	RSVD	–	–	Reserved
0x483C	MIPI CTRL3C	0x02	RW	Bit[7:4]: Not used Bit[3:0]: t_clk_pre Unit: pclk2x cycle
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

table 6-17 MIPI control registers (sheet 8 of 8)

address	register name	default value	R/W	description
0x484A	SEL MIPI CTRL4A	0x3F	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
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	0x10	RW	Bit[7]: Not used Bit[6:4]: lane1_sel Bit[3]: Not used Bit[2:0]: lane0_sel
0x4851	RSVD	–	–	Reserved

6.18 ISPFC [0x4900 - 0x4903]

table 6-18 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

6.19 ISP control [0x5000 - 0x501F]

table 6-19 ISP control registers (sheet 1 of 4)

address	register name	default value	R/W	description
0x5000	DSP CTRL00	0xBF	RW	Bit[7]: Lens correction (LENC) function enable Bit[6]: var_pix enable Bit[5]: OTP_DPC function enable Bit[4]: Manual AWB gain enable Bit[3]: Black DPC function enable Bit[2]: White DPC function enable Bit[1]: BLC function enable Bit[0]: ISP enable

table 6-19 ISP control registers (sheet 2 of 4)

address	register name	default value	R/W	description
0x5001	DSP CTRL01	0x01	RW	Bit[7]: latch_en for SC_ISP Bit[6:5]: avg_sel Bit[4]: isp_eof_sel Bit[3]: isp_sof_sel Bit[2]: fmt_sel Bit[1]: bypass_sof Bit[0]: avg_size manual enable
0x5002	DSP CTRL02	0x89	RW	Bit[7]: isp_raw_en Bit[6:4]: win_yoff_adj Bit[3]: lenc_bias_on Bit[2]: Sensor bias manual enable Bit[1]: AWB gain bias manual enable Bit[0]: AWB gain bias on
0x5003	AWB_BIAS_MAN	0x10	RW	Bit[7:0]: AWB gain bias manual
0x5004	SENSOR_BIAS_MAN	0x10	RW	Bit[7:0]: Sensor bias manual
0x5005	DSP CTRL05	0x28	RW	Bit[7]: Pre_awb Average enable Bit[6]: Black line rblue signal reverse Bit[5]: DCBLC enable Bit[4]: AWB gain (for AWB) bias manual enable Bit[3]: Zero line rblue Bit[2]: DPC bias manual enable Bit[1]: Null Bit[0]: Long/short reverse for AWB gain
0x5006	DSP CTRL06	0x04	RW	Bit[7]: Bit select manual enable Bit[6:5]: Bit select manual Bit[4]: Bit select bit append Bit[3]: AWB gain use the same long-gain for both long and short Bit[2]: Dummy line auto enable Bit[1]: SRAM test DPC1 Bit[0]: BLC gain change switch for HDR-lite mode
0x5007	DSP CTRL07	0xA4	RW	Bit[7:4]: SRAM RM DPC1 Bit[3]: DPC px_order manual enable Bit[2:1]: DPC px_order manual Bit[0]: BLC real_gain switch for hdr_lite mode
0x5008	FMT_MUX_CTRL	0x00	RW	Bit[7:0]: fmt_mux_ctrl

table 6-19 ISP control registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x5009	DSP CTRL09	0x00	RW	Bit[7]: Average start manual enable Bit[6]: SRAM RME DPC 1 Bit[5]: Null Bit[4]: px_order manual enable Bit[3:2]: px_order manual Bit[1]: Gfirst reverse Bit[0]: Rblue reverse
0x500A	RED AWB_GAIN_L HIGH	0x04	RW	Bit[7:6]: Reserved Bit[5:0]: Red gain for AWB gain long[13:8]
0x500B	RED AWB_GAIN_L LOW	0x00	RW	Bit[7:0]: Red gain for AWB gain long[7:0]
0x500C	GREEN AWB_GAIN_L HIGH	0x04	RW	Bit[7:6]: Reserved Bit[5:0]: Green gain for AWB gain long[13:8]
0x500D	GREEN AWB_GAIN_L LOW	0x00	RW	Bit[7:0]: Green gain for AWB gain long[7:0]
0x500E	BLUE AWB_GAIN_L HIGH	0x04	RW	Bit[7:6]: Reserved Bit[5:0]: Blue gain for AWB gain long[13:8]
0x500F	BLUE AWB_GAIN_L LOW	0x00	RW	Bit[7:0]: Blue gain for AWB gain long[7:0]
0x5010	RED AWB_GAIN_S HIGH	0x04	RW	Bit[7:6]: Reserved Bit[5:0]: Red gain for AWB gain short[13:8]
0x5011	RED AWB_GAIN_S LOW	0x00	RW	Bit[7:0]: Red gain for AWB gain short[7:0]
0x5012	GREEN AWB_GAIN_S HIGH	0x04	RW	Bit[7:6]: Reserved Bit[5:0]: Green gain for AWB gain short[13:8]
0x5013	GREEN AWB_GAIN_S LOW	0x00	RW	Bit[7:0]: Green gain for AWB gain short[7:0]
0x5014	BLUW AWB_GAIN_S HIGH	0x04	RW	Bit[7:6]: Reserved Bit[5:0]: Blue gain for AWB gain short[13:8]

table 6-19 ISP control registers (sheet 4 of 4)

address	register name	default value	R/W	description
0x5015	BLUE AWB_GAIN_S LOW	0x00	RW	Bit[7:0]: Blue gain for AWB gain short[7:0]
0x5016	X_ADDR_END_ ADD	0x00	RW	Bit[7:0]: Additional x_addr_end for output black line
0x5017	Y_ADDR_END_ ADD	0x00	RW	Bit[7:0]: Additional y_addr_end for output black line
0x5018	X_START_MAN HIGH	0x00	RW	Bit[7:1]: Reserved Bit[0]: X_start manual for average[8]
0x5019	X_START_MAN LOW	0x00	RW	Bit[7:0]: X_start manual for average[7:0]
0x501A	Y_START_MAN	0x00	RW	Bit[7:0]: Y_start manual for average
0x501B	AVG_H_MAN HIGH	0x00	RW	Bit[7:1]: Reserved Bit[0]: Average horizontal size manual[8]
0x501C	AVG_H_MAN LOW	0x00	RW	Bit[7:0]: Average horizontal size manual[7:0]
0x501D	AVG_V_MAN	0x00	RW	Bit[7:0]: Average vertical size manual
0x501E	AWB_11_GAIN_ BIAS_MAN	0x10	RW	Bit[7:0]: AWB (for AWB before DPC) gain bias manual
0x501F	DPC_BIAS_MAN	0x10	RW	Bit[7:0]: DPC bias manual

6.20 AVG control [0x5680 - 0x5693]

table 6-20 AVG control registers (sheet 1 of 3)

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	RSVD	–	–	Reserved

table 6-20 AVG control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x5683	AVG CTRL03	0x00	RW	Bit[7:0]: y_start_avg[7:0] AVG sub-window vertical start position low byte
0x5684	AVG CTRL04	0x01	RW	Bit[7:1]: Not used Bit[0]: window_width_avg[8] Sub-window width high byte
0x5685	AVG CTRL05	0x40	RW	Bit[7:0]: window_width_avg[7:0] Sub-window width low byte
0x5686	RSVD	–	–	Reserved
0x5687	AVG CTRL07	0xF0	RW	Bit[7:0]: window_height_avg
0x5688	AVG CTRL08	0x11	RW	Bit[7:4]: weight01 Weight of zone01 Bit[3:0]: weight00 Weight of zone00
0x5689	AVG CTRL09	0x11	RW	Bit[7:4]: weight03 Weight of zone03 Bit[3:0]: weight02 Weight of zone02
0x568A	AVG CTRL0A	0x11	RW	Bit[7:4]: weight5 Weight of zone5 Bit[3:0]: weight4 Weight of zone4
0x568B	AVG CTRL0B	0x11	RW	Bit[7:4]: weight7 Weight of zone7 Bit[3:0]: weight6 Weight of zone6
0x568C	AVG CTRL0C	0x11	RW	Bit[7:4]: weight9 Weight of zone9 Bit[3:0]: weight8 Weight of zone8
0x568D	AVG CTRL0D	0x11	RW	Bit[7:4]: weight11 Weight of zone1 Bit[3:0]: weight10 Weight of zone10
0x568E	AVG CTRL0E	0x11	RW	Bit[7:4]: weight13 Weight of zone13 Bit[3:0]: weight12 Weight of zone12
0x568F	AVG CTRL0F	0x11	RW	Bit[7:4]: weight15 Weight of zone15 Bit[3:0]: weight14 Weight of zone14

table 6-20 AVG control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x5690	AVG CTRL10	0x12	RW	Bit[7:5]: Not used Bit[4:3]: px_order manual Bit[2]: px_order manual enable Bit[1]: avg_opt Bit[0]: avg_man
0x5691	AVG RO11	–	R	Bit[7:0]: weight-sumS Sum of weight
0x5692	AVG RO12	–	R	Bit[7:1]: Not used Bit[0]: Average calculated indicating signal for SCCB read
0x5696	AVG RO16	–	R	Bit[7:0]: Whole image average output

6.21 DPC control [0x5780 - 0x579A, 0x57A0 - 0x57A7]

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

address	register name	default value	R/W	description
0x5780	DPC CTRL00	0xD4	RW	Bit[7:6]: Null Bit[5]: Enable tail Bit[4]: Enable saturate crosscluster Bit[3]: Enable 3x3 cluster Bit[2]: Enable cross cluster Bit[1]: Enable general tail Bit[0]: Manual mode enable
0x5781	DPC CTRL01	0x0F	RW	Bit[7:4]: Saturate Bit[3]: Enable diffchannel wpconn Bit[2]: Enable diffchannel bpconn Bit[1]: Enable samechannel wpconn Bit[0]: Enable samechannel bpconn
0x5782	DPC CTRL02	0x44	RW	Bit[7:4]: Status thre step Bit[3:0]: Wthre list0
0x5783	DPC CTRL03	0x02	RW	Bit[7:4]: Debug mode Bit[3:0]: Wthre list1
0x5784	DPC CTRL04	0x01	RW	Bit[7:4]: Debug mode Bit[3:0]: Wthre list2
0x5785	DPC CTRL05	0x01	RW	Bit[7:4]: Debug mode Bit[3:0]: Wthre list3

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

address	register name	default value	R/W	description
0x5786	DPC CTRL06	0x00	RW	Bit[7:4]: Debug mode Bit[3:0]: Adaptive pattern thre
0x5787	DPC CTRL07	0x04	RW	Bit[7:4]: Debug mode Bit[3:0]: Adaptive pattern step
0x5788	DPC CTRL08	0x08	RW	Bit[7:4]: Debug mode Bit[3:0]: More connection case thre
0x5789	DPC CTRL09	0x0F	RW	Bit[7:0]: DPC level list0
0x578A	DPC CTRL10	0xFD	RW	Bit[7:0]: DPC level list1
0x578B	DPC CTRL11	0xF5	RW	Bit[7:0]: DPC level list2
0x578C	DPC CTRL12	0xF5	RW	Bit[7:0]: DPC level list3
0x578D	DPC CTRL13	0x03	RW	Bit[7]: Debug mode Bit[6:0]: Gain list0
0x578E	DPC CTRL14	0x0F	RW	Bit[7]: Debug mode Bit[6:0]: Gain list1
0x578F	DPC CTRL15	0x3F	RW	Bit[7]: Debug mode Bit[6:0]: Gain list2
0x5790	DPC CTRL16	0x08	RW	Bit[7:4]: Debug mode Bit[3:0]: Matching thre
0x5791	DPC CTRL17	0x04	RW	Bit[7:4]: Debug mode Bit[3:0]: Status thre
0x5792	DPC CTRL18	0x04	RW	Bit[7:4]: Debug mode Bit[3:0]: Thre ratio
0x5793	DPC CTRL19	0x52	RW	Bit[7:6]: Vnum list1 Bit[5:4]: Vnum list0 Bit[1]: v153_en Bit[0]: clip_intq_en
0x5794	DPC CTRL20	0xA3	RW	Bit[7:6]: Vnum list3 Bit[5:4]: Vnum list2 Bit[1:0]: Edge option
0x5795	DPC CTRL21	0xFF	RW	Bit[7:0]: awb_gain_apply_max
0x5797	DPC CTRL23	–	R	Bit[7]: Debug mode Bit[6:0]: Bthre
0x5798	DPC CTRL24	–	R	Bit[7:5]: Debug mode Bit[4:0]: Wthre
0x5799	DPC CTRL25	–	R	Bit[7:5]: Debug mode Bit[4:0]: Thre1

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

address	register name	default value	R/W	description
0x579A	DPC CTRL26	–	R	Bit[7:6]: Debug mode Bit[5:0]: Thre2
0x579B	DPC CTRL27	–	R	Bit[7]: Debug mode Bit[6:0]: Thre3
0x579C	DPC RO28	–	R	Bit[7:5]: Debug mode Bit[4:0]: Thre4
0x579D	DPC RO29	–	R	Bit[7:4]: Debug mode Bit[3:0]: Level

6.22 LENC control [0x5800 - 0x58FF]

table 6-22 LENC control registers (sheet 1 of 4)

address	register name	default value	R/W	description
0x5800	LENC G00	0x00	RW	Bit[7:0]: Control point G00 for luminance compensation
0x5801	LENC G01	0x00	RW	Bit[7:0]: Control point G01 for luminance compensation
0x5802	LENC G02	0x00	RW	Bit[7:0]: Control point G02 for luminance compensation
0x5803	LENC G03	0x00	RW	Bit[7:0]: Control point G03 for luminance compensation
0x5804	LENC G04	0x00	RW	Bit[7:0]: Control point G04 for luminance compensation
0x5805	LENC G05	0x00	RW	Bit[7:0]: Control point G05 for luminance compensation
0x5806	LENC G06	0x00	RW	Bit[7:0]: Control point G06 for luminance compensation
0x5807	LENC G07	0x00	RW	Bit[7:0]: Control point G07 for luminance compensation
0x5808	LENC G10	0x00	RW	Bit[7:0]: Control point G10 for luminance compensation
0x5809~ 0x584E	LENC G11~ LENC G96	–	RW	Bit[7:0]: Control point G11~G96 for luminance compensation

table 6-22 LENC control registers (sheet 2 of 4)

address	register name	default value	R/W	description
0x584F	LENC G97	0x80	RW	Bit[7:0]: Control point G97 for luminance compensation
0x5850	LENC B00	0x80	RW	Bit[7:0]: Control point B00 for blue channel compensation
0x5851	LENC B01	0x80	RW	Bit[7:0]: Control point B01 for blue channel compensation
0x5852	LENC B02	0x80	RW	Bit[7:0]: Control point B02 for blue channel compensation
0x5853	LENC B03	0x80	RW	Bit[7:0]: Control point B03 for blue channel compensation
0x5854	LENC B04	0x80	RW	Bit[7:0]: Control point B04 for blue channel compensation
0x5855	LENC B05	0x80	RW	Bit[7:0]: Control point B05 for blue channel compensation
0x5856	LENC B06	0x80	RW	Bit[7:0]: Control point B06 for blue channel compensation
0x5857	LENC B07	0x80	RW	Bit[7:0]: Control point B07 for blue channel compensation
0x5858	LENC B10	0x80	RW	Bit[7:0]: Control point B10 for blue channel compensation
0x5859~ 0x589E	LENC B11~ LENC B96	–	RW	Bit[7:0]: Control point B11~B96 for blue channel compensation
0x589F	LENC B97	0x80	RW	Bit[7:0]: Control point B97 for blue channel compensation
0x58A0	LENC R00	0x80	RW	Bit[7:0]: Control point R00 for red channel compensation
0x58A1	LENC R01	0x80	RW	Bit[7:0]: Control point R01 for red channel compensation
0x58A2	LENC R02	0x80	RW	Bit[7:0]: Control point R02 for red channel compensation
0x58A3	LENC R03	0x80	RW	Bit[7:0]: Control point R03 for red channel compensation
0x58A4	LENC R04	0x80	RW	Bit[7:0]: Control point R04 for red channel compensation
0x58A5	LENC R05	0x80	RW	Bit[7:0]: Control point R05 for red channel compensation
0x58A6	LENC R06	0x80	RW	Bit[7:0]: Control point R06 for red channel compensation

table 6-22 LENC control registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x58A7	LENC R07	0x80	RW	Bit[7:0]: Control point R07 for red channel compensation
0x58A8	LENC R10	0x80	RW	Bit[7:0]: Control point R10 for red channel compensation
0x58A9~ 0x58EE	LENC R11~ LENC R96	–	RW	Bit[7:0]: Control point R11~R96 for red channel compensation
0x58EF	LENC R97	0x80	RW	Bit[7:0]: Control point R97 for red channel compensation
0x58F0	LENC MAXGAIN	0x60	RW	Bit[7:0]: If auto lens switch enable 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
0x58F1	LENC MINGAIN	0x40	RW	Bit[7:0]: If auto lens switch enable 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.
0x58F2	LENC MAXQ	0x40	RW	Bit[6:0]: This value indicates the maximum amplitude which luminance channel compensates when AutoLensSwitchEnable is true. Value should be in the range [0~64]
0x58F3	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]
0x58F4	LENC CTRL	0x36	RW	Bit[5]: Add BLC target after applying compensation Bit[4]: Enable BLC target for LENC 0: Disable BLC target 1: Enable BLC target Bit[3]: br2x mode Bit[2]: autoq_en Bit[1]: dither_en Bit[0]: g2xgain_en

table 6-22 LENC control registers (sheet 4 of 4)

address	register name	default value	R/W	description
0x58F5	LENC HSCALE	0x04	RW	Bit[7:5]: Reserved Bit[4:0]: HScale[12:8] For horizontal gain calculation, this value indicates the step between two connected horizontal pixels, where HScale = 4*2^18 / image width
0x58F6	LENC HSCALE	0x3B	RW	Bit[7:0]: HScale[7:0]
0x58F7	LENC VSCALE	0x02	RW	Bit[7:5]: Reserved Bit[4:0]: VScale[12:8] For vertical gain calculation, this value indicates the step between two connected vertical pixels, where VScale = 4*2^17 / image height
0x58F8	LENC VSCALE	0xCD	RW	Bit[7:0]: VScale[7:0]
0x58F9	LENC XOFFSET	–	R	Bit[7:5]: Reserved Bit[4:0]: Input sensor horizontal offset[12:8]
0x58FA	LENC XOFFSET	–	R	Bit[7:0]: Input sensor horizontal offset[7:0]
0x58FB	LENC YOFFSET	–	R	Bit[7:5]: Reserved Bit[4:0]: Input sensor vertical offset[12:8]
0x58FC	LENC YOFFSET	–	R	Bit[7:0]: Input sensor vertical offset[7:0]
0x58FD	LENC INPUT	–	R	Bit[7:6]: Reserved Bit[5]: Input sensor flip Bit[4]: Input sensor mirror Bit[3:2]: Input sensor Y skip Bit[1:0]: Input sensor X skip
0x58FE	LENC OVERFLOW	–	R	Bit[1]: Overflow v Bit[0]: Overflow h
0x58FF	LENC QVALUE	–	R	Bit[6:0]: Real amplitude Q value

6.23 VAP [0x5900 - 0x5901]

table 6-23 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)

6.24 WINC [0x5980 - 0x598D]

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

address	register name	default value	R/W	description
0x5980	WINC CTRL00	0x00	RW	Bit[7:4]: Not used Bit[3:0]: x_start_offset[11:8] Start address in horizontal
0x5981	WINC CTRL01	0x00	RW	Bit[7:0]: x_start_offset[7:0]
0x5982	WINC CTRL02	0x00	RW	Bit[7:4]: Not used Bit[3:0]: y_start_offset[11:8] Start address in vertical
0x5983	WINC CTRL03	0x00	RW	Bit[7:0]: y_start_offset[7:0]
0x5984	WINC CTRL04	0x0C	RW	Bit[7:4]: Not used Bit[3:0]: window_width[11:8] Select whole zone width high byte
0x5985	WINC CTRL05	0xE0	RW	Bit[7:0]: window_width[7:0] Select whole zone width low byte
0x5986	WINC CTRL06	0x09	RW	Bit[7:4]: Not used Bit[3:0]: window_height[11:8] Select whole zone height high byte

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

address	register name	default value	R/W	description
0x5987	WINC CTRL07	0xB0	RW	Bit[7:0]: window_height[7:0] Select whole zone height low byte
0x5988	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]: Debug mode Bit[0]: Manual window enable 0: Window size from window top 1: Window size from 0x5A00 to 0x5A07
0x5989	WINC RO09	–	R	Bit[7:4]: Not used Bit[3:0]: Pixel count[11:8] (for debug)
0x598A	WINC RO0A	–	R	Bit[7:0]: Pixel count[7:0] (for debug)
0x598B	WINC RO0B	–	R	Bit[7:4]: Not used Bit[3:0]: Line count[11:8] (for debug)
0x598C	WINC RO0C	–	R	Bit[7:0]: Line count[7:0] (for debug)
0x598D	WINC NEW STG	–	RW	Bit[7:0]: New stg delay

6.25 OTP DPC control [0x5500 - 0x550D, 0x5510 - 0x5527]

table 6-25 OTP DPC registers (sheet 1 of 4)

address	register name	default value	R/W	description
0x5500	OTP CTRL00	0x00	RW	Bit[7:1]: Not used Bit[0]: Memory start address[8]
0x5501	OTP CTRL01	0x00	RW	Bit[7:0]: Memory start address[7:0]
0x5502	OTP CTRL02	0x01	RW	Bit[7:1]: Not used Bit[0]: Memory end address[8]
0x5503	OTP CTRL03	0xFF	RW	Bit[7:0]: Memory end address[7:0]

table 6-25 OTP DPC registers (sheet 2 of 4)

address	register name	default value	R/W	description
0x5504	OTP CTRL04	0x02	RW	Bit[7:6]: Null 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
0x5505	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
0x5506	OTP CTRL06	0x00	RW	Bit[7]: Not used Bit[6:0]: Constrain exposure threshold[14:8]
0x5507	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)
0x5508	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)
0x5509	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)

table 6-25 OTP DPC registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x550A	OTP CTRL0A	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Manual horizontal even increase step
0x550B	OTP CTRL0B	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Manual horizontal odd increase step
0x550C	OTP CTRL0C	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Manual vertical even increase step
0x550D	OTP CTRL0D	0x01	RW	Bit[7:5]: Not used Bit[4:0]: Manual vertical odd increase step
0x5510	OTP RO10	–	R	Bit[7:4]: Not used Bit[3:0]: Horizontal offset[11:8]
0x5511	OTP RO11	–	R	Bit[7:0]: Horizontal offset[7:0]
0x5512	OTP RO12	–	R	Bit[7:4]: Not used Bit[3:0]: Vertical offset[11:8]
0x5513	OTP RO13	–	R	Bit[7:0]: Vertical offset[7:0]
0x5514	OTP RO14	–	R	Bit[7:5]: Not used Bit[4:0]: Horizontal even increase step
0x5515	OTP RO15	–	R	Bit[7:5]: Not used Bit[4:0]: Horizontal odd increase step
0x5516	OTP RO16	–	R	Bit[7:5]: Not used Bit[4:0]: Vertical even increase step
0x5517	OTP RO17	–	R	Bit[7:5]: Not used Bit[4:0]: Vertical odd increase step
0x5518~ 0x551F	NOT USED	–	–	Not Used
0x5520	OTP CTRL20	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Manual X offset[11:8]
0x5521	OTP CTRL21	0x00	RW	Bit[7:0]: Manual X offset[7:0]
0x5522	OTP CTRL22	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Manual Y offset[11:8]
0x5523	OTP CTRL23	0x00	RW	Bit[7:0]: Manual Y offset[7:0]
0x5524	OTP CTRL06	0x00	RW	Bit[7]: Reserved Bit[6:0]: Constrain exposure short threshold[14:8]
0x5525	OTP CTRL07	0x00	RW	Bit[7:0]: Constrain exposure short threshold[7:0] (disable OTP function when the sensor exposure is smaller than the constrain exposure threshold)

table 6-25 OTP DPC registers (sheet 4 of 4)

address	register name	default value	R/W	description
0x5526	OTP CTRL08	0x07	RW	Bit[7:6]: Reserved Bit[5:0]: Constrain gain short threshold (disable OTP function when the sensor gain is smaller than the constrain gain threshold)
0x5527	OTP CTRL09	0x08	RW	Bit[7:4]: Reserved Bit[3:0]: Recover threshold short (recover when the high 8-bits of the recovered data is bigger than the original one by this threshold)

6.26 pre_DSP control [0x5040 - 0x506E]

table 6-26 pre_DSP control registers (sheet 1 of 4)

address	register name	default value	R/W	description
0x5040	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
0x5041	PRE CTRL01	0x01	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

table 6-26 pre_DSP control registers (sheet 2 of 4)

address	register name	default value	R/W	description
0x5042	PRE CTRL02	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Line number interrupt[11:8]
0x5043	PRE CTRL03	0x01	RW	Bit[7:0]: Line number interrupt[7:0]
0x5044	PRE CTRL04	0x01	RW	Bit[7:0]: Scale X input manual size[15:8]
0x5045	PRE CTRL05	0x00	RW	Bit[7:0]: Scale X input manual size[7:0]
0x5046	PRE CTRL06	0x01	RW	Bit[7:0]: Scale Y input manual size[15:8]
0x5047	PRE CTRL07	0x00	RW	Bit[7:0]: Scale Y input manual size[7:0]
0x5048	PRE CTRL08	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Horizontal manual offset[11:8]
0x5049	PRE CTRL09	0x00	RW	Bit[7:0]: Horizontal manual offset[7:0]
0x504A	PRE CTRL0A	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Vertical manual offset[11:8]
0x504B	PRE CTRL0B	0x00	RW	Bit[7:0]: Vertical manual offset[7:0]
0x504C	PRE RO0C	–	R	Bit[7:4]: Reserved Bit[3:0]: Input image pixel number[11:8]
0x504D	PRE RO0D	–	R	Bit[7:0]: Input image pixel number[7:0]
0x504E	PRE RO0E	–	R	Bit[7:4]: Reserved Bit[3:0]: Input image line number[11:8]
0x504F	PRE RO0F	–	R	Bit[7:0]: Input image line number[7:0]
0x5050	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
0x5051	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

table 6-26 pre_DSP control registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x5052	PRE RO12	–	R	Bit[7:0]: HREF blank length for dummy line[15:8]
0x5053	PRE RO13	–	R	Bit[7:0]: HREF blank length for dummy line[7:0]
0x5054	PRE RO14	–	R	Bit[7:0]: HREF length for dummy line[15:8]
0x5055	PRE RO15	–	R	Bit[7:0]: HREF length for dummy line[7:0]
0x5056	PRE RO16	–	R	Bit[7:5]: Reserved Bit[4]: Dummy error indicating signal Bit[3]: Reserved Bit[2:0]: Dummy line clock ratio output
0x5057	PRE RO17	–	R	Bit[7:4]: Horizontal odd increase step Bit[3:0]: Vertical odd increase step
0x5058	PRE RO18	–	R	Bit[7:4]: Reserved Bit[3:0]: Horizontal sensor offset[11:8]
0x5059	PRE RO19	–	R	Bit[7:0]: Horizontal sensor offset[7:0]
0x505A	PRE RO1A	–	R	Bit[7:4]: Reserved Bit[3:0]: Vertical sensor offset[11:8]
0x505B	PRE RO1B	–	R	Bit[7:0]: Vertical sensor offset[7:0]
0x505C	PRE RO1C	–	R	Bit[7:4]: Reserved Bit[3:0]: Horizontal window offset[11:8]
0x505D	PRE RO1D	–	R	Bit[7:0]: Horizontal window offset[7:0]
0x505E	PRE RO1E	–	R	Bit[7:4]: Reserved Bit[3:0]: Vertical window offset[11:8]
0x505F	PRE RO1F	–	R	Bit[7:0]: Vertical window offset[7:0]
0x5060	PRE RO20	–	R	Bit[7:5]: Reserved Bit[4:0]: Horizontal window output size[12:8]
0x5061	PRE RO21	–	R	Bit[7:0]: Horizontal window output size[7:0]
0x5062	PRE RO22	–	R	Bit[7:4]: Reserved Bit[3:0]: Vertical window output size[11:8]
0x5063	PRE RO23	–	R	Bit[7:0]: Vertical window output size[7:0]
0x5064	PRE RO24	–	R	Bit[7:6]: Reserved Bit[5:4]: Horizontal skip Bit[3:2]: Reserved Bit[1:0]: Vertical skip
0x5065	PRE RO25	–	R	Bit[7:4]: Horizontal even increase step Bit[3:0]: Vertical even increase step
0x5066	NOT USED	–	–	Not Used

table 6-26 pre_DSP control registers (sheet 4 of 4)

address	register name	default value	R/W	description
0x5067	PRE RO27	–	R	Bit[7:4]: Reserved Bit[3:0]: Cut top offset for bi-linear BLC[11:8]
0x5068	PRE RO28	–	R	Bit[7:0]: Cut top offset for bi-linear BLC[7:0]
0x5069	PRE RO29	–	R	Bit[7:4]: Reserved Bit[3:0]: Cut bottom offset for bi-linear BLC[11:8]
0x506A	PRE RO2A	–	R	Bit[7:0]: Cut bottom offset for bi-linear BLC[7:0]
0x506B	PRE CTRL2B	0x09	RW	Bit[7:4]: Reserved Bit[3:0]: Array height for bi-linear BLC[11:8]
0x506C	PRE CTRL2C	0xB0	RW	Bit[7:0]: Array height for bi-linear BLC[7:0]
0x506D	PRE CTRL2D	0x00	RW	Bit[7:6]: Reserved Bit[5]: Manual horizontal skip enable Bit[4:0]: Reserved
0x506E	PRE CTRL2E	0x00	RW	Bit[7:6]: Reserved Bit[5]: Manual vertical skip enable Bit[4:0]: Reserved

6.27 AWB control [0x5200 - 0x5214]

table 6-27 AWB control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5200	AWBM CTRL00	0x04	RW	Bit[7:4]: Debug mode Bit[3:0]: Stable range
0x5201	AWBM CTRL01	0x08	RW	Bit[7:4]: Debug mode Bit[3:0]: Stable rangew
0x5202	AWBM CTRL02	0x00	RW	Bit[7:1]: Debug mode Bit[0]: gain_man_en
0x5203	AWBM CTRL03	0x04	RW	Bit[7:4]: Debug mode Bit[3:0]: Blue gain manua[11:8]
0x5204	AWBM CTRL04	0x00	RW	Bit[7:0]: Blue gain manual[7:0]
0x5205	AWBM CTRL05	0x04	RW	Bit[7:4]: Debug mode Bit[3:0]: Green gain manual[11:8]
0x5206	AWBM CTRL06	0x00	RW	Bit[7:0]: Green gain manual[7:0]
0x5207	AWBM CTRL07	0x04	RW	Bit[7:4]: Debug mode Bit[3:0]: Red gain manual[11:8]

table 6-27 AWB control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x5208	AWBM CTRL08	0x00	RW	Bit[7:0]: Red gain manual[7:0]
0x5209	AWBM CTRL09	–	R	Bit[7:0]: Blue gain[15:8]
0x520A	AWBM CTRL0a	–	R	Bit[7:0]: Blue gain[7:0]
0x520B	AWBM CTRL0b	–	R	Bit[7:0]: Green gain[15:8]
0x520C	AWBM CTRL0c	–	R	Bit[7:0]: Green gain[7:0]
0x520D	AWBM CTRL0d	–	R	Bit[7:0]: Red gain[15:8]
0x520E	AWBM CTRL0e	–	R	Bit[7:0]: Red gain[7:0]
0x520F	AWBM CTRL0f	–	R	Bit[7:0]: Blue after gain average[15:8]
0x5210	AWBM CTRL10	–	R	Bit[7:0]: Blue after gain average[7:0]
0x5211	AWBM CTRL11	–	R	Bit[7:0]: Green after gain average[15:8]
0x5212	AWBM CTRL12	–	R	Bit[7:0]: Green after gain average[7:0]
0x5213	AWBM CTRL13	–	R	Bit[7:0]: Red after gain average[15:8]
0x5214	AWBM CTRL14	–	R	Bit[7:0]: Red after gain average[7:0]

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
peak solder temperature (10 second dwell time)		245°C

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-A}	supply voltage (analog)	2.6	2.8	3.0	V
V_{DD-D}	supply voltage (digital core for 2-lane MIPI up to 1000 Mbps/lane)	1.1	1.2	1.3	V
V_{DD-IO}	supply voltage (digital I/O)	1.7	1.8	3.0	V
I_{DD-A}	active (operating) current[b]		17.2	TBD	mA
I_{DD-IO}			1.1	TBD	mA
I_{DD-D}			40.2	TBD	mA
$I_{DDS-SCCB}$	standby current[c]		590	TBD	µA
$I_{DDS-PWDN}$			250	TBD	µA
$I_{DDS-XSHUTDN}$			0.15	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}[d]	SCL and SDA	-0.5	0	0.54	V
V_{IH}	SCL and SDA	1.28	1.8	3.0	V

a. maximum active current is measured under typical supply voltage

b. DVDD is provided by external regulator for lower power consumption. DOVDD = 1.8V

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

d. 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 physical specifications

figure 8-1 package specifications

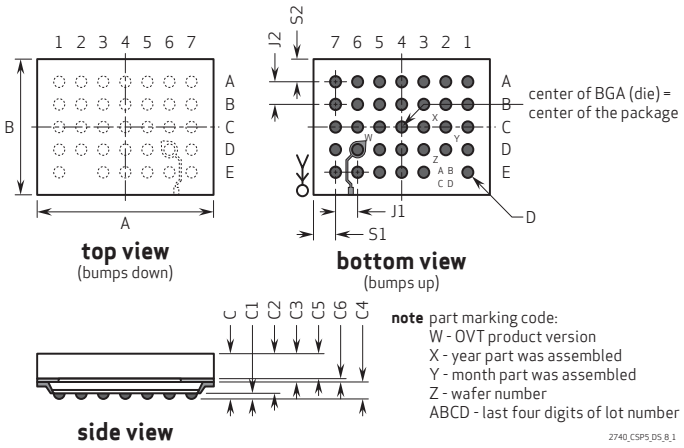


table 8-1 package dimensions

parameter	symbol	min	typ	max	unit
package body dimension x	A	3830	3855	3880	µm
package body dimension y	B	2894	2919	2944	µm
package height	C	660	720	780	µm
ball height	C1	80	110	140	µm
package body thickness	C2	575	610	645	µm
thickness from top glass surface to die	C3	425	445	465	µm
image plane height	C4	250	295	340	µm
glass thickness	C5	385	400	415	µm
air gap between sensor and glass	C6	41	45	49	µm
ball diameter	D	170	200	230	µm
total pin count	N		34		
pins pitch x-axis	J1		530		µm
pins pitch y-axis	J2		480		µm
edge-to-pin center distance along x	S1	308	337.5	368	µm
edge-to-pin center distance along y	S2	470	499.5	530	µm

8.2 IR reflow specifications

note The OV2740 uses a lead-free package.

note To reduce image artifacts from infrared light and to provide the best image quality. OmniVision recommends an IR cut filter.

figure 8-2 IR reflow ramp rate requirements

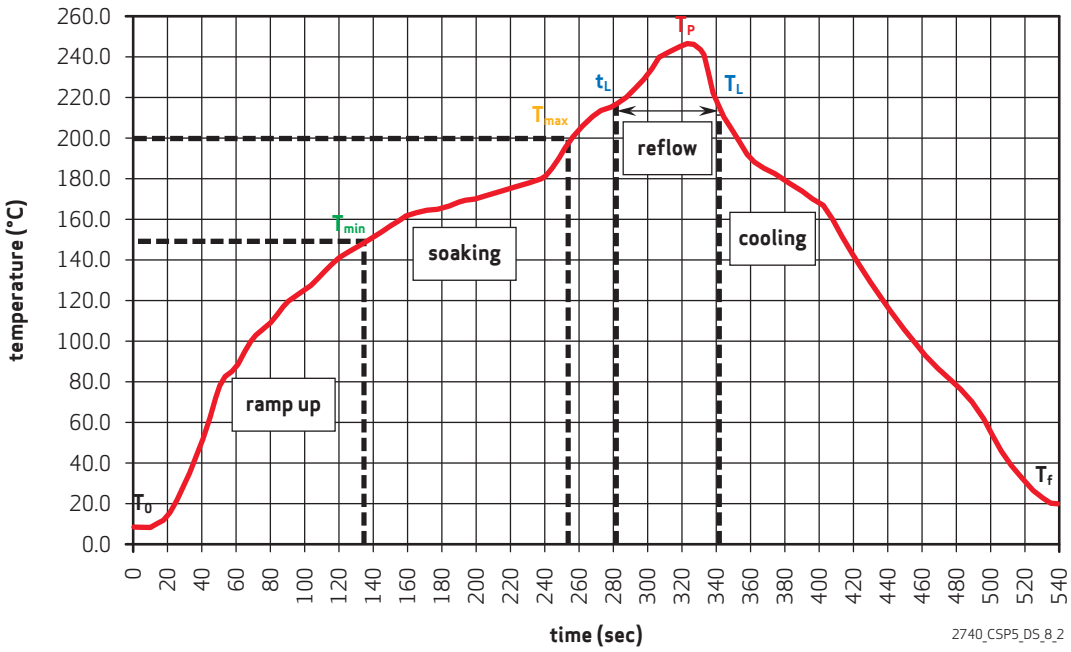


table 8-2 reflow conditions[ab]

zone	description	exposure
ramp up A (T_0 to T_{min})	heating from room temperature to 150°C	temperature slope ≤ 3°C per second
soaking	heating from 150°C to 200°C	90 ~ 150 seconds
ramp up B (t_L to T_P)	heating from 217°C to 245°C	temperature slope ≤ 3°C per second
peak temperature	maximum temperature in SMT	245°C +0/-5° (duration max 30 sec)
reflow (t_L to T_L)	temperature higher than 217°C	30 ~ 120 seconds
ramp down A (T_P to T_L)	cooling down from 245°C to 217°C	temperature slope ≤ 3°C per second
ramp down B (T_L to T_f)	cooling down from 217°C to room temperature	temperature slope ≤ 2°C per second
T_0 to T_P	room temperature to peak temperature	≤ 8 minutes

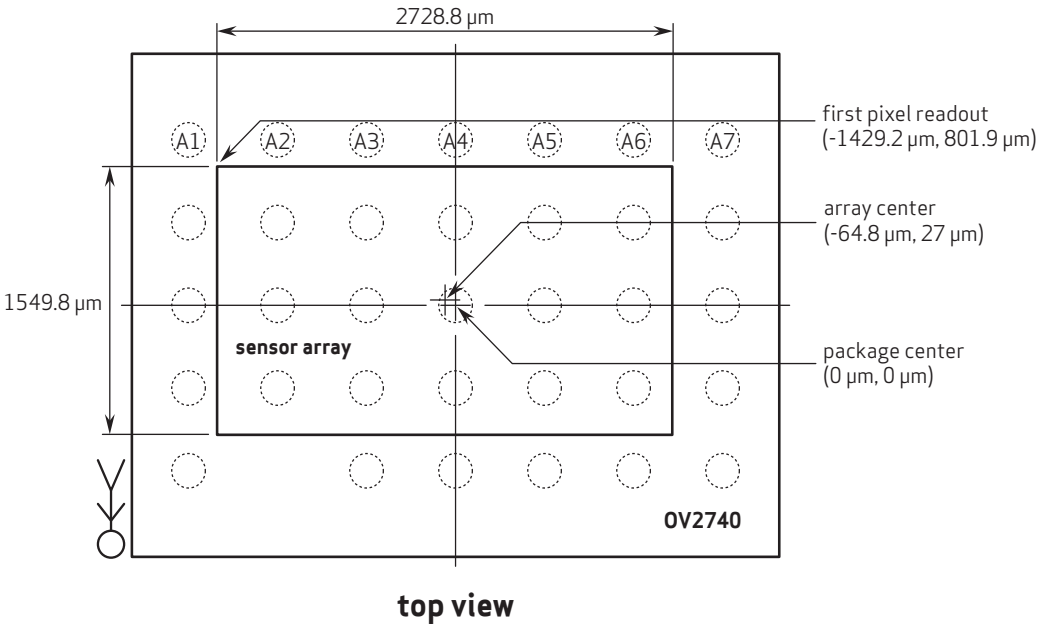
a. maximum number of reflow cycles = 3

b. N2 gas reflow or control O2 gas PPM <500 as recommendation

9 optical specifications

9.1 sensor array center

figure 9-1 sensor array center



top view

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 pins A1 to A7 oriented down on the PCB.

2740_CSP5_DS_9_1

9.2 lens chief ray angle (CRA)

figure 9-2 chief ray angle (CRA)

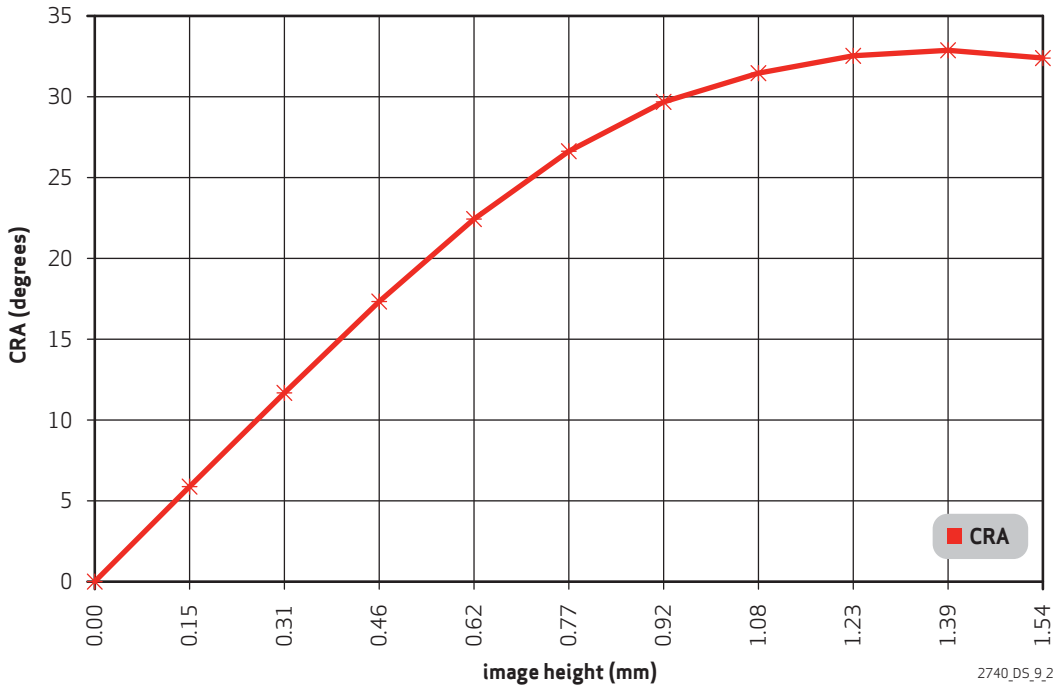


table 9-1 CRA versus image height plot

field (%)	image height (mm)	CRA (degrees)
0.00	0.00	0.0
0.10	0.15	5.9
0.20	0.31	11.8
0.30	0.46	17.4
0.40	0.62	22.5
0.50	0.77	26.7
0.60	0.92	29.7
0.70	1.08	31.5
0.80	1.23	32.6
0.90	1.39	33.0
1.00	1.54	32.5

revision history

version 1.0 02.21.2014

- initial release

version 1.01 03.03.2014

- in figure 2-2, added note 10

version 1.02 04.01.2014

- in table 1-3, changed removed "VH" from symbol in fifth row and added "VN" to symbol in the sixth row
- in table 1-3, changed "DOGND" to "DOVDD" in equivalent schematic of last row
- in table 5-7 and table 6-7, changed description of register 0x3508 to "Bit[4:0]: Gain[12:8]" and completely changed description of register 0x3509

version 1.1 04.18.2014

- in key specifications, changed active power requirements from 153 mW to 98.4 mW and standby power requirements from 16 µA to 250 µA
- in section 4.9, changed section title from "ultra low power mode (ULPM)" to "power saving (PSV) mode" and changed all references of "ULPM" to "PSV" throughout entire datasheet
- in table 5-7, changed bit description for register 0x3509[7:0] by removing "0x100 is 2x gain" and adding "0x140 is 2.5x gain. Maximum gain is 15.5x; 0x7C0."
- in table 6-7, changed bit description for register 0x3509[7:0] by removing "0x100 is 2x gain" and adding "0x140 is 2.5x gain. Maximum gain is 15.5x; 0x7C0."
- in table 7-3, changed active (operating) current typical value for I_{DD-A} from 21.5 mA to 17.2 mA, I_{DD-IO} from 3.1 mA to 1.1 mA, and I_{DD-D} from 72.6 mA to 40.2 mA
- in table 7-3, changed standby current typical value for $I_{DDS-SCCB}$ from 116.1 µA to 590 µA and $I_{DDS-PWDN}$ from 120.5 µA to 250 µA

version 1.11 05.30.2014

- in section 4.8, removed first sentence from second paragraph of section description
- in table 4-8, removed register 0x5A08

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: sales@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 6933 1933

TAIWAN

Taipei +886 2 2657 9800
Hsinchu +886 3 6110933

website: **www.ovt.com**