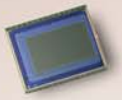
OmniVision



datasheet

PRODUCT SPECIFICATION

1/4" color CMOS WXGA (1 megapixel) HD sensor
with OmniPixel3-HS™ technology

OV9713

color CMOS (1 megapixel) HD sensor with OmniPixel3-HS™ technology

datasheet (CSP3)
PRODUCT SPECIFICATION

version 2.1
april 2012

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

applications

- PC multimedia
- tablets
- security
- entertainment
- cellular phones
- games

ordering information

- **OV09713-A49A** (color, lead-free)
  49-pin CSP3

features

- automatic black level calibration (ABLC)
- programmable controls for frame rate, mirror and flip, cropping and windowing
- image quality controls: lens correction and defective pixel canceling
- supports output formats: 8/10/12-bit RAW RGB (MIPI/LVDS)
- supports horizontal and vertical sub-sampling
- supports images sizes: 1280x800, 640x400, and 320x200
- fast mode switching
- support 2x2 binning
- standard serial SCCB interface
- two-lane MIPI/LVDS serial output interface
- embedded 256 bits one-time programmable (OTP) memory for part identification, etc.
- on-chip phase lock loop (PLL)
- programmable I/O drive capability
- built-in 1.5V regulator for core
- support alternate frame HDR / line HDR

key specifications (typical)

- **active array size:** 1296 x 812
- **power supply:**
  core: 1.5VDC ± 5%
  analog: 2.6 ~ 3.0V
  I/O: 1.7 ~ 3.0V
- **power requirements:**
  active: 95 mA
  standby: 30 µA
  xshutdown: 5 µA
- **temperature range:**
  operating: -30°C to 85°C junction temperature (see **table 8-2**)
  stable image: 0°C to 50°C junction temperature (see **table 8-2**)
- **output formats:** 12-bit RGB RAW
- **lens size:** 1/4"
- **input clock frequency:** 6~27 MHz
- **lens chief ray angle:** 25° non-linear (see **figure 10-2**)
- **max S/N ratio:** 39 dB
- **dynamic range:** 73 dB @ 8x gain
- **maximum image transfer rate:**
  1280x800: 60 fps (see **table 2-1**)
  640x400: 120 fps (see **table 2-1**)
  320x200: 240 fps (see **table 2-1**)
- **sensitivity:** 3300 mV/Lux-sec
- **scan mode:** progressive
- **maximum exposure interval:** 800 x t_{ROW}
- **pixel size:** 3.0 µm x 3.0 µm
- **dark current:** 2.3 mV/s @ 50°C junction temperature
- **image area:** 3936 µm x 2460 µm
- **package dimensions:** 6110 µm x 4930 µm

note
pixel performance shown are target values. These values are subject to change based on real measurements.

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 OV9713 image sensor. The package information is shown in section 9.

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

pin number	signal name	pin type	description
A2	STROBE	I/O	strobe output
A3	AGND	ground	ground for analog circuit
A4	DOVDD	power	power for I/O circuit
A5	DVDD	reference	power for digital circuit
A6	SCL	input	SCCB input clock
A7	VN2	reference	internal analog reference
A8	VH	reference	internal analog reference
A9	PWDNB	input	power down (active low with internal pull up resistor)
A10	DVDD	reference	power for digital circuit
B1	SID	I/O	SCCB ID select input 0: ID=0x20 1: ID=0x6C
B2	DVDD	reference	power for digital circuit
B3	AVDD	power	power for analog circuit
B4	DOGND	ground	ground for I/O and digital circuit
B5	DOVDD	power	power for I/O circuit
B6	SDA	I/O	SCCB data
B7	LED	I/O	LED driver output
B8	VN	reference	internal analog reference
B9	XSHUTDOWN	input	reset and power down (active low with internal pull down resistor)
B10	DOGND	ground	ground for I/O and digital circuit
B11	AVDD	power	power for analog circuit
C1	NC	–	no connect
C2	DOGND	ground	ground for I/O and digital circuit
C11	AGND	ground	ground for analog circuit

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

pin number	signal name	pin type	description
D1	NC	–	no connect
D11	NC	–	no connect
E1	NC	–	no connect
E11	NC	–	no connect
F1	DOGND	ground	ground for I/O and digital circuit
F11	AGND	ground	ground for analog circuit
F6	EGND	ground	ground for MIPI TX circuit
F9	TM	input	test mode (active high with internal pull down resistor)
G1	DOVDD	power	power for I/O circuit
G2	FSIN/VSYNC	I/O	frame sync input/VSYNC output
G3	AGND	ground	ground for analog circuit
G4	MDN0	output	MIPI/LVDS TX data lane negative output
G5	MDP0	output	MIPI/LVDS TX data lane positive output
G6	EVDD	reference	power for MIPI/LVDS TX circuit
G7	MDN1	output	MIPI/LVDS TX second data lane negative output
G8	MDP1	output	MIPI/LVDS TX second data lane positive output
G9	GPIO	I/O	VSYNC (output)/HSYNC (output)/GPIO
G10	DOGND	ground	ground for I/O and digital circuit
G11	AVDD	power	power for analog circuit
H2	DVDD	reference	power for digital circuit
H3	AVDD	power	power for analog circuit
H5	MCN	output	MIPI/LVDS TX clock lane negative output
H6	MCP	output	MIPI/LVDS TX clock lane positive output
H8	XVCLK	input	system input clock/scan clock input
H9	ATEST	output	analog test output
H10	DVDD	reference	power for digital circuit

table 1-2 configuration under various conditions

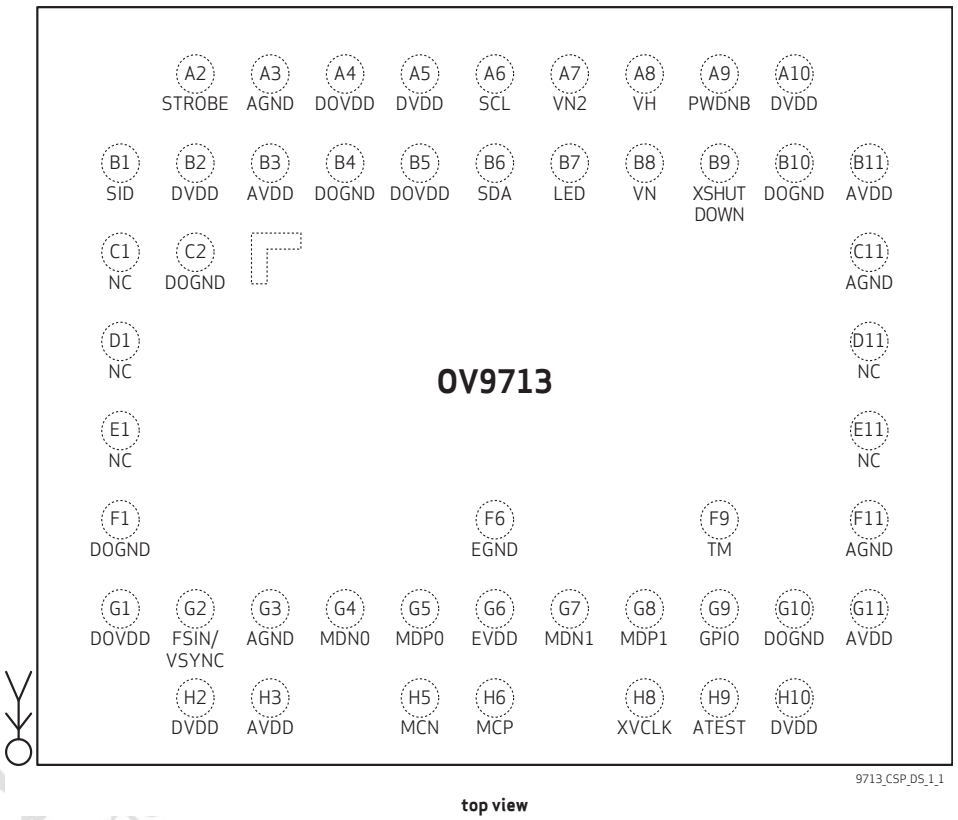
pin	signal name	RESET[a]	after RESET release[b]	software standby	hardware standby[c]
B6	SDA	input	input	input	high-z
G2	FSIN/VSYNC	input	input by default (configurable)	high-z by default (configurable)	high-z by default (configurable)
G4, G5, H6, H5, G7, G8	MDN0, MDP0, MCP, MCN, MDN1, MDP1	high-z	high-z by default (configurable)	high-z by default (configurable)	high-z by default (configurable)
A2	STROBE	input	input by default (configurable)	high-z by default (configurable)	high-z by default (configurable)
A9	PWDNB	input	input	input	input
A10	DVDD	input	input	input	input
A4	DOVDD	input	input	input	input
H8	XVCLK	input	input	input	high-z
A6	SCL	input	input	input	high-z

a. XSHUTDOWN = 0

b. XSHUTDOWN from 0 to 1

c. PWDNB = 0

figure 1-1 pin diagram



top view

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

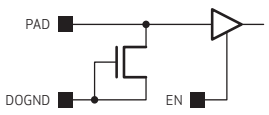
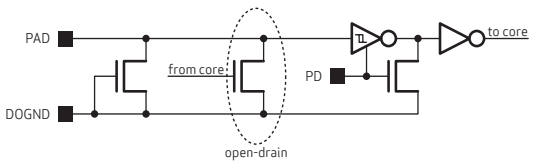
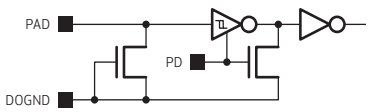
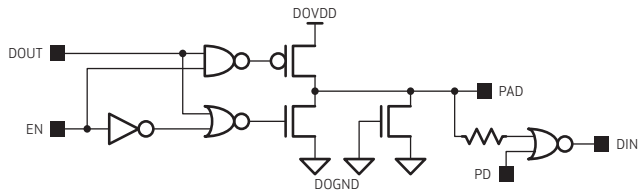
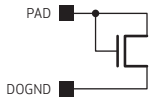
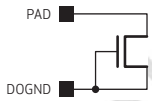
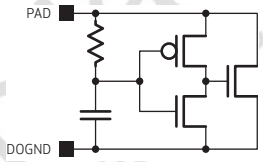
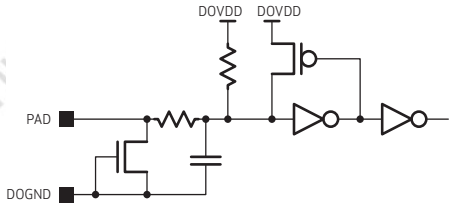
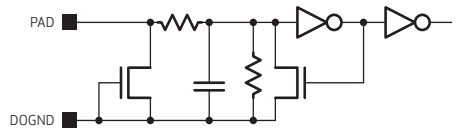
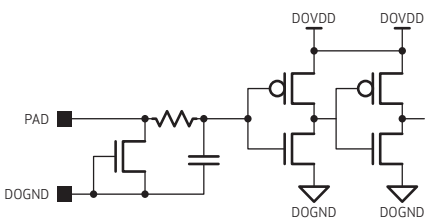
symbol	equivalent circuit
XVCLK	
SDA	
SCL	

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

symbol	equivalent circuit
FSIN/VSYNC, STROBE, LED, GPIO	
VN, VN2	
MDP0, MDN0, MDP1, MDN1, MCP, MCN, EGND, AGND, DOGND, ATEST, VH	
AVDD, EVDD, DVDD, DOVDD	
PWDNB	
TM, XSHUTDOWN	
SID	

2 system level description

2.1 overview

The OV9713 color image sensor is a low voltage, high performance 1/4-inch 1 megapixel CMOS image sensor that provides the functionality of a single WXGA (1280x800) camera using OmniPixel3-HS™ technology. It provides full-frame, sub-sampled, and windowed 12-bit MIPI/LVDS images in various formats via the control of the Serial Camera Control Bus (SCCB) interface.

The OV9713 has an image array capable of operating at up to 60 frames per second (fps) in WXGA resolution with complete user control over image quality, formatting and output data transfer. All required image processing functions, including exposure control, white balance, 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.

The OV9713 includes a one-time programmable (OTP) memory. The OV9713 has up to two lanes of MIPI/LVDS. interface.

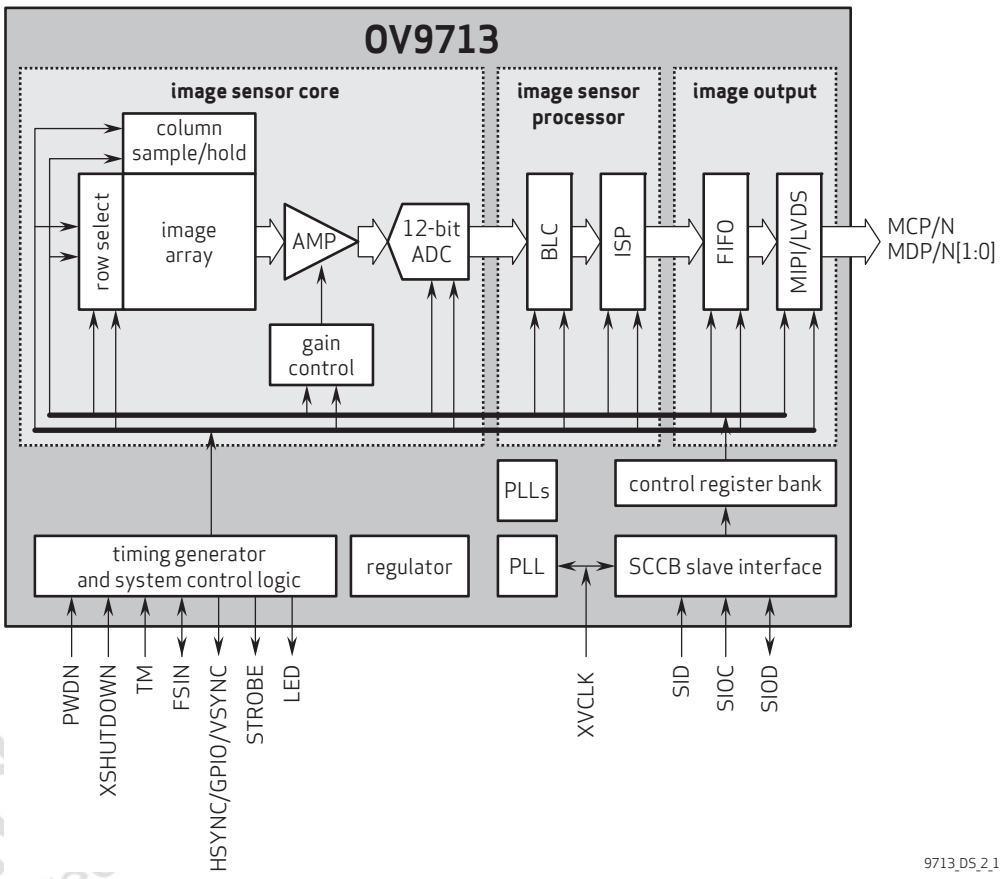
2.2 architecture

The OV9713 sensor core generates streaming pixel data at a constant frame rate. **figure 2-1** shows the functional block diagram of the OV9713 image sensor.

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

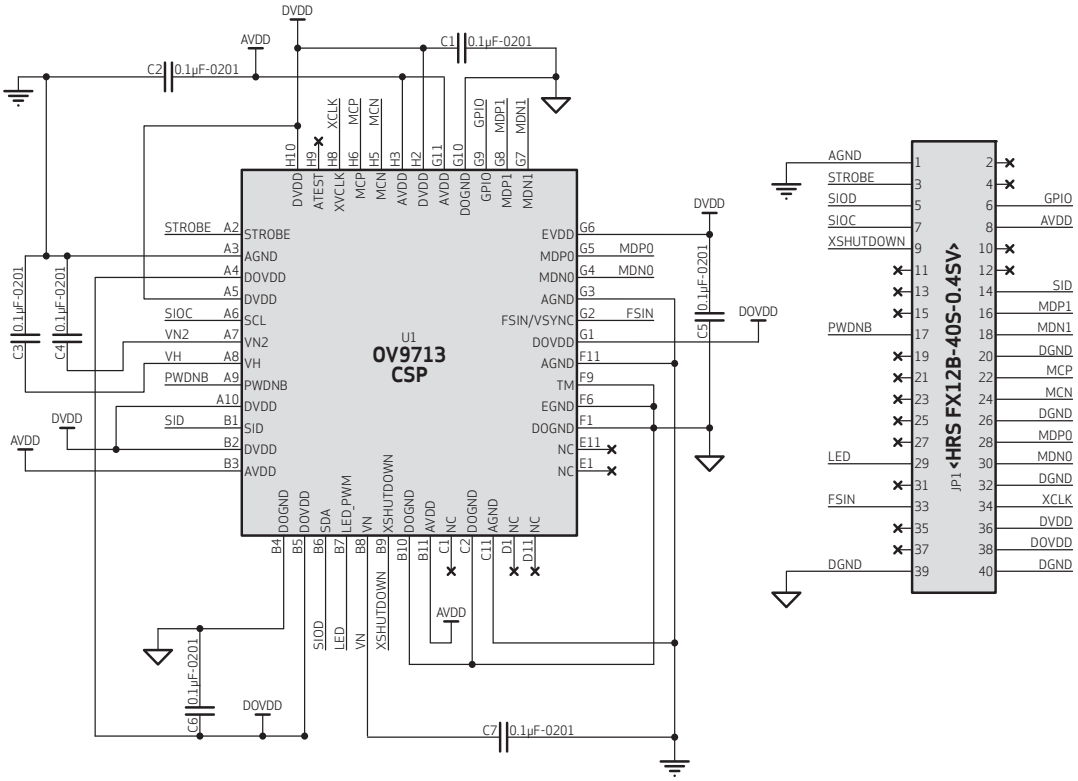
The exposure time is controlled by adjusting the time interval between pre-charging and sampling. After the data of the pixels in the row has been sampled, it is processed through analog circuitry to correct the offset and multiply the data with corresponding gain. Following analog processing is the ADC which outputs 12-bit data for each pixel in the array.

figure 2-1 OV9713 block diagram



9713_DS_2_1

figure 2-2 reference design schematic



note 1 PWDNB is active low and should be pulled high to DOVDD outside of module if unused

note 2 XSHUTDOWN is active low and should be connected to DOVDD outside of the module if unused

note 3 AVDD is 2.6 - 3.0V of sensor analog power (clean)

note 4 DOVDD is 1.7 - 3.0V of sensor digital IO power (clean). 1.8V is recommended.

note 5 EVDD and DVDD are 1.5V±5%. If DOVDD is 1.8V, internal DVDD is recommended. If DOVDD is 2.8V, external DVDD is recommended

note 6 sensor AGND and DGND should be separated inside the module and connected to a single point outside the module (do not connect inside module).

note 7 capacitors should be close to the related sensor pins

note 8 EVDD/EGND are power/ground for MIPI core. MCP and MCN are MIPI clock lane positive and negative output. MDPx and MDNx are MIPI data lane positive and negative output

note 9 traces of MCP, MCNx, MDPx, and MDN should have the same or similar length. The differential impedance of the clock pair and data pair transmission line should be controlled at 100 Ohm.

note 10 SID is I2C slave address selection. When this pin is pulled to ground, I2C slave address is 0x20. When this pin is pulled to DOVDD, I2C slave address is 0x6C.

9713_CSP_DS_2_2

2.3 format and frame

The OV9713 supports RAW RGB output with 2 lane MIPI/LVDS interface.

table 2-1 supported resolution and frame rate

format[a]	resolution	max frame rate with MIPI	methodology
full resolution	1280x800	60 fps	full
640x400	640x400	120 fps	binningx2
320x200	320x200	240 fps	binningx2+skipx2
160x100	160x100	480 fps	binningx2+skipx4

a. all formats with minimum 4 dummy lines and 4 dummy pixels

2.4 I/O control

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

function	register	description
output drive capability control	0x3001	Bit[1:0]: pad I/O drive capability 00: 1x 01: 2x 10: 3x 11: 4x
GPIO I/O control	0x3002	Bit[0]: input/output control for GPIO pin 0: input 1: output
GPIO I/O output select	0x3026	Bit[4]: output selection for GPIO pin 0: normal data path 1: register control value
GPIO output value	0x3008	Bit[4]: GPIO output value
GPIO output data select	0x3667	Bit[4]: pad_gpio0_opt 0: VSYNC 1: HSYNC
FSIN I/O control	0x3004	Bit[7]: input/output control for FSIN pin 0: input 1: output
FSIN output select	0x3024	Bit[7]: output selection for FSIN pin 0: normal data path 1: register control value

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

function	register	description
FSIN output value	0x3006	Bit[7]: FSIN output value
STROBE I/O control	0x3002	Bit[3]: input/output control for STROBE pin 0: input 1: output
STROBE output select	0x3024	Bit[2]: output selection for STROBE pin 0: normal data path 1: register control value
STROBE output value	0x3006	Bit[2]: STROBE output value
LED I/O control	0x3024	Bit[5]: input/output control for LED pin 0: input 1: output
LED output select	0x3907	Bit[7]: output selection for LED pin 0: debug only 1: led_gen

2.5 MIPI interface

The OV9713 supports a 2-lane MIPI transmitter interface with a data transfer rate of up to 640 Mbps per lane.

figure 2-3 MIPI timing

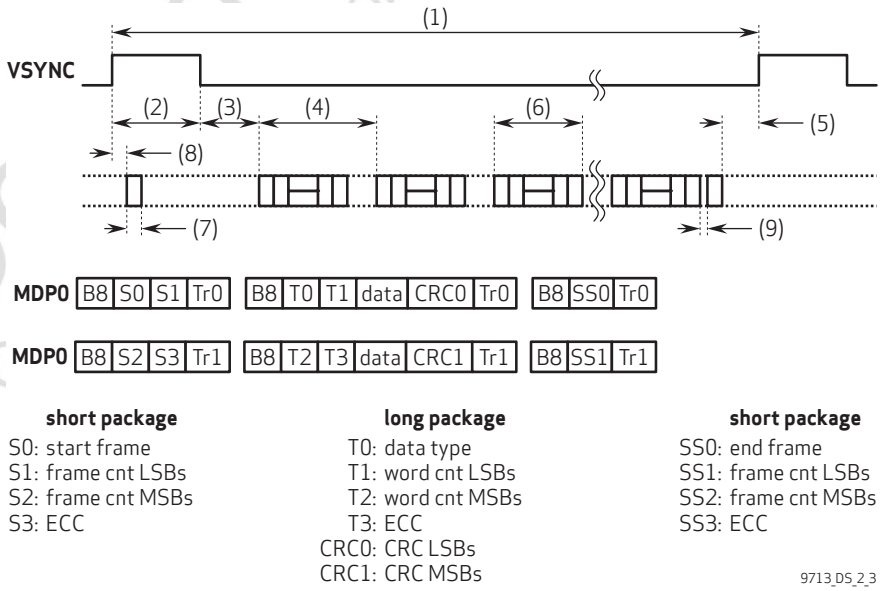


table 2-3 MIPI timing specifications

mode	timing
full 1312x820 => 1280x800 (crop) 60 fps	(1) 1,216,880 tps (2) 1,024 tps (3) 15,782 tpp (4) 1,113 tpp (5) 5,780 tpp (6) 969 tpp (7) 8 tpp (8) 71 tpp (9) 55 tpp where tps = 1 Tsclk, tpp = 1 Tpclk
SUB2 1312x820 => 640x400 (crop + skip) 120 fps	(1) 608,580 tps (2) 1024 tps (3) 10,378 tpp (4) 1,101 tpp (5) 4,832 tpp (6) 489 tpp (7) 8 tpp (8) 72 tpp (9) 55 tpp where tps = 1 Tsclk, tpp = 1 Tpclk
SUB4 1312x820 => 320x200 (crop + skip) 240 fps	(1) 303,880 tps (2) 1,024 tps (3) 9,800 tpp (4) 1,065 tpp (5) 5,086 tpp (6) 249 tpp (7) 8 tpp (8) 72 tpp (9) 55 tpp where tps = 1 Tsclk, tpp = 1 Tpclk
SUB8 1312x820 => 160x100 (crop + skip) 480 fps	(1) 151,620 tps (2) 1,024 tps (3) 9,073 tpp (4) 997 tpp (5) 4,922 tpp (6) 129 tpp (7) 8 tpp (8) 72 tpp (9) 55 tpp where tps = 1 Tsclk, tpp = 1 Tpclk

2.6 VSYNC timing in MIPI mode

2.6.1 VSYNC modes

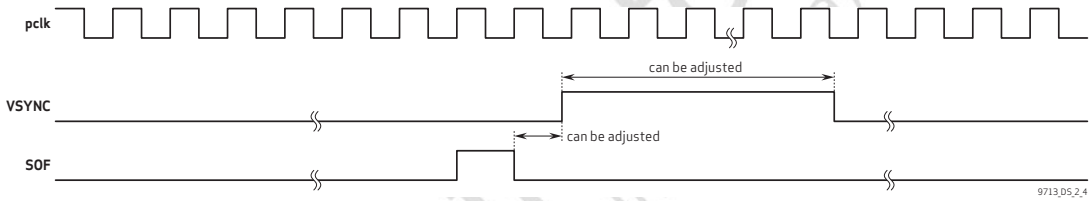
The VSYNC rising edge delay is controlled by register vsync_delay ({0x4314, 0x4315, 0x4316}) in all three VSYNC modes. VSYNC width is controlled by register vsync_width_pixel ({0x4311, 0x4312}) for VSYNC modes 1 and 2. The steps of both registers vsync_delay and vsync_width_pixel are 1 system clock cycle.

Note that VSYNC timing in mode 3 is a long VSYNC mode. The register vsync_width_pixel ({0x4311, 0x4312}) controls VSYNC falling edge differently.

2.6.1.1 VSYNC mode 1

In mode 1, VSYNC is generated by the internal start of frame (SOF) signal (see **figure 2-4**).

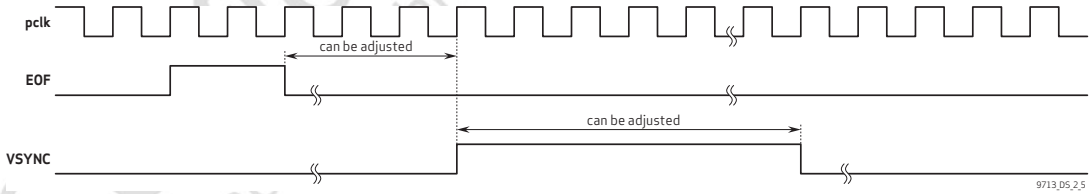
figure 2-4 VSYNC timing in mode 1



2.6.1.2 VSYNC mode 2

In mode 2, VSYNC is generated by the internal end of frame (EOF) signal (see **figure 2-5**).

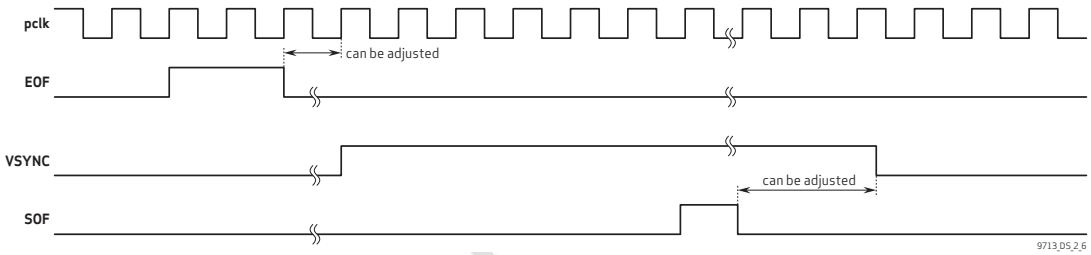
figure 2-5 VSYNC timing in mode 2



2.6.1.3 VSYNC mode 3

In mode 3, VSYNC is generated by EOF and SOF (see **figure 2-6**).

figure 2-6 VSYNC timing in mode 3



2.7 power management

2.7.1 power up sequence

The OV9713 can use either the internal regulator or an external power supply to provide digital core 1.5V DVDD.

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

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

table 2-4 power up sequence

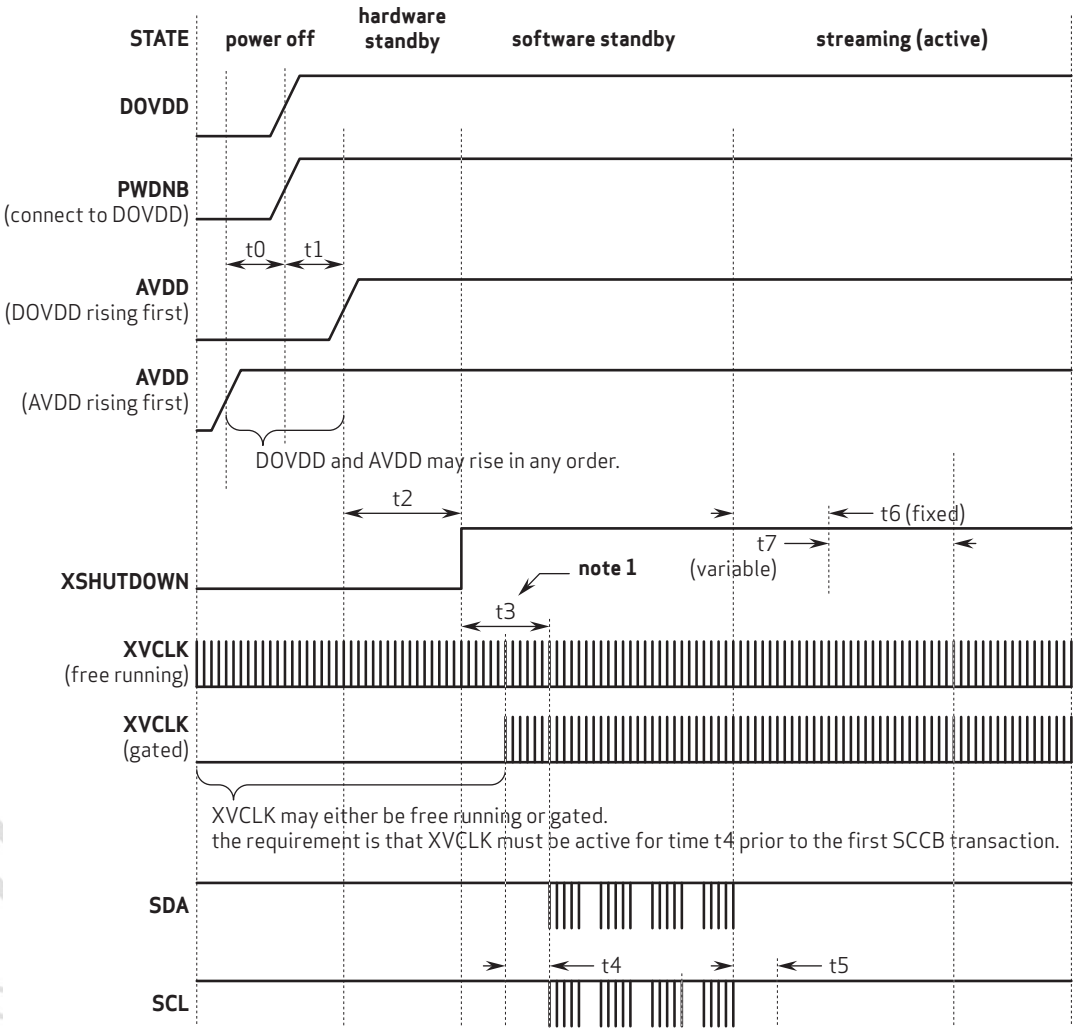
case	DVDD	XSHUTDOWN	PWDNB	power up sequence requirement
1	internal	GPIO	DOVDD	Refer to **figure 2-7** 1. AVDD rising can occur before or after DOVDD rising as long as they are before XSHUTDOWN rising 2. XSHUTDOWN is pulled up after AVDD and DOVDD are stable
2	internal	DOVDD	GPIO	Refer to **figure 2-8** 1. AVDD rising occurs before DOVDD rising 2. PWDNB rising occurs after DOVDD rising
3	external	GPIO	DOVDD	Refer to **figure 2-9** 1. DOVDD rising must occur before external DVDD rising 2. AVDD rising can occur before or after DOVDD rising 3. XSHUTDOWN rising must occur after AVDD, DOVDD and DVDD are stable
4	external	DOVDD	GPIO	Refer to **figure 2-10** 1. AVDD rising occurs before DOVDD rising 2. DOVDD rising occurs before DVDD 3. PWDNB rising occurs after DVDD rising

table 2-5 power up sequence timing constraints

constraint	label	min	max	unit
AVDD rising – DOVDD rising	t0	0	∞	ns
DOVDD rising – AVDD rising	t1			ns
AVDD or DOVDD rising, whichever is last – XSHUTDOWN rising	t2	0.0		ns
XSHUTDOWN rising – first SCCB transaction	t3[a]	8192		XVCLK cycles
minimum number of XVCLK cycles prior to the first SCCB transaction	t4	8192		XVCLK cycles
PLL start up/lock time	t5		0.2	ms
entering streaming mode – first frame start sequence (fixed part)	t6		10	ms
entering streaming mode – first frame start sequence (variable part)	t7	delay is the exposure time value		lines

a. when using the internal DVDD, an additional 1ms must be added to t3 to wait for SCCB to become stable

figure 2-7 power up sequence (case 1)



note 1 when using the internal DVDD, an additional 1ms must be added to t3 to wait for SCCB to become stable

9713 DS 2 7

figure 2-8 power up sequence (case 2)

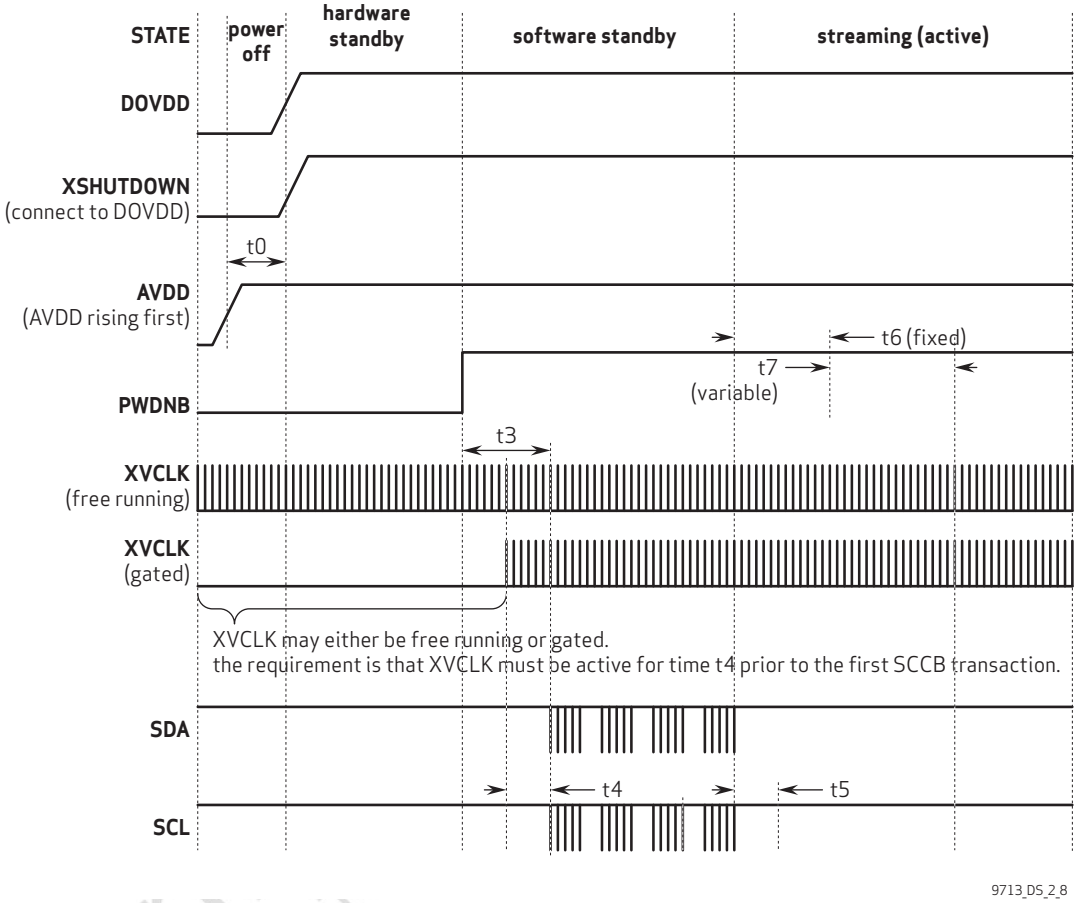


figure 2-9 power up sequence (case 3)

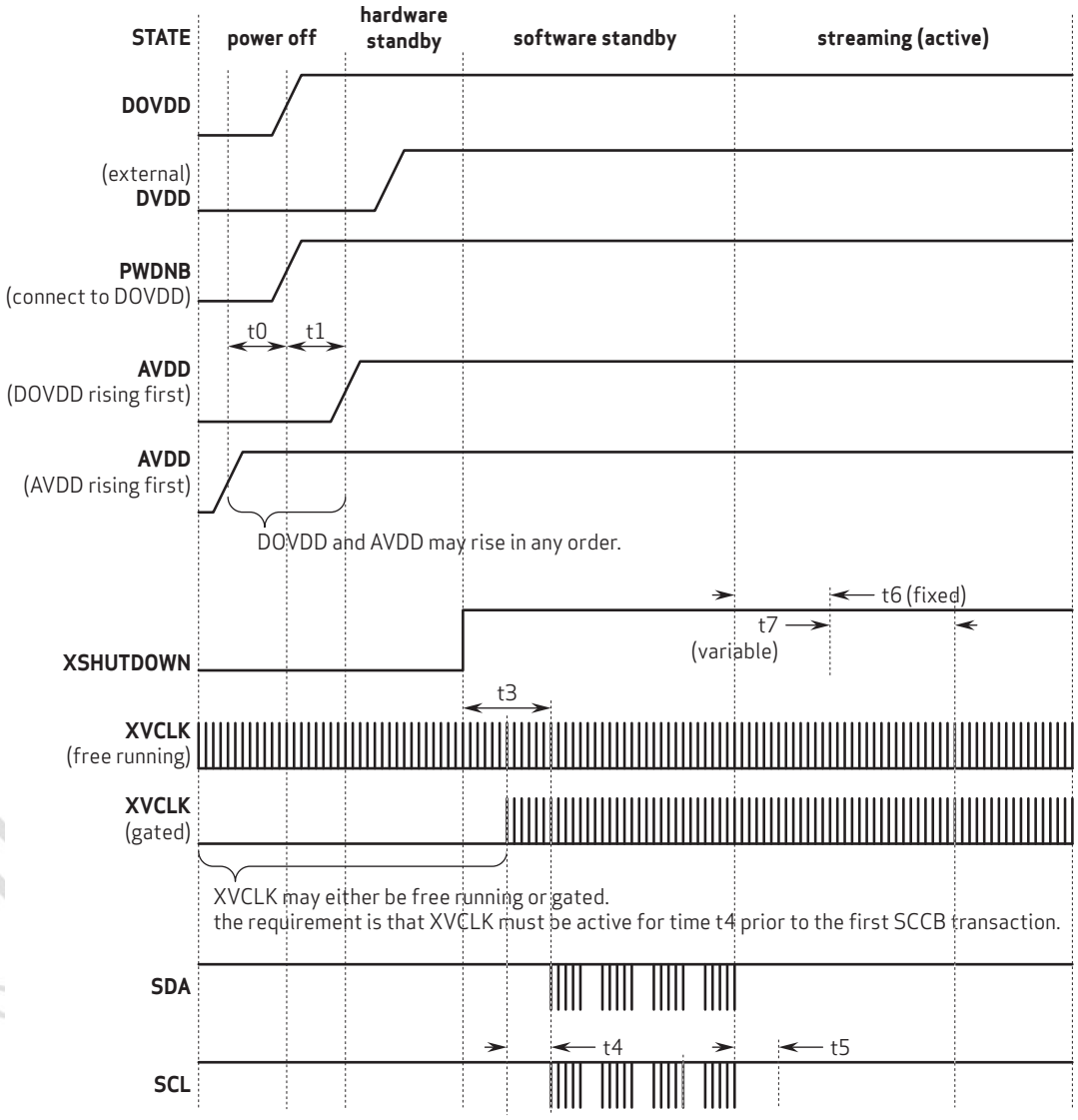
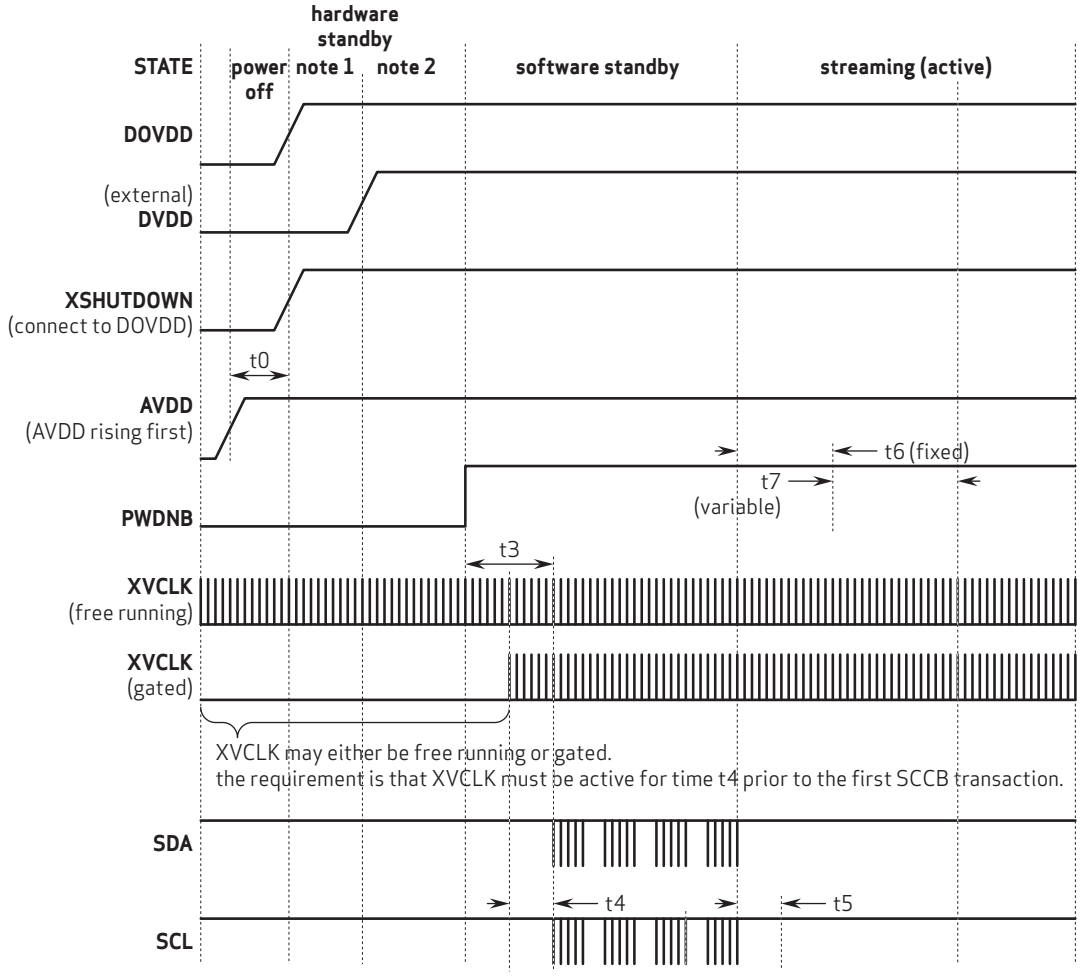


figure 2-10 power up sequence (case 4)



note 1 with minimum power consumption
note 2 with low power consumption

9713_DS_2_10

2.7.2 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). Similarly to the power-up sequence, the XVCLK input clock may be either gated or continuous. If the SCCB command to exit streaming is received while a frame of MIPI data is being output, then the sensor must wait to the MIPI frame end code before entering software standby mode.

If the SCCB command to exit streaming mode is received during the inter frame time, then the sensor must enter software standby mode immediately.

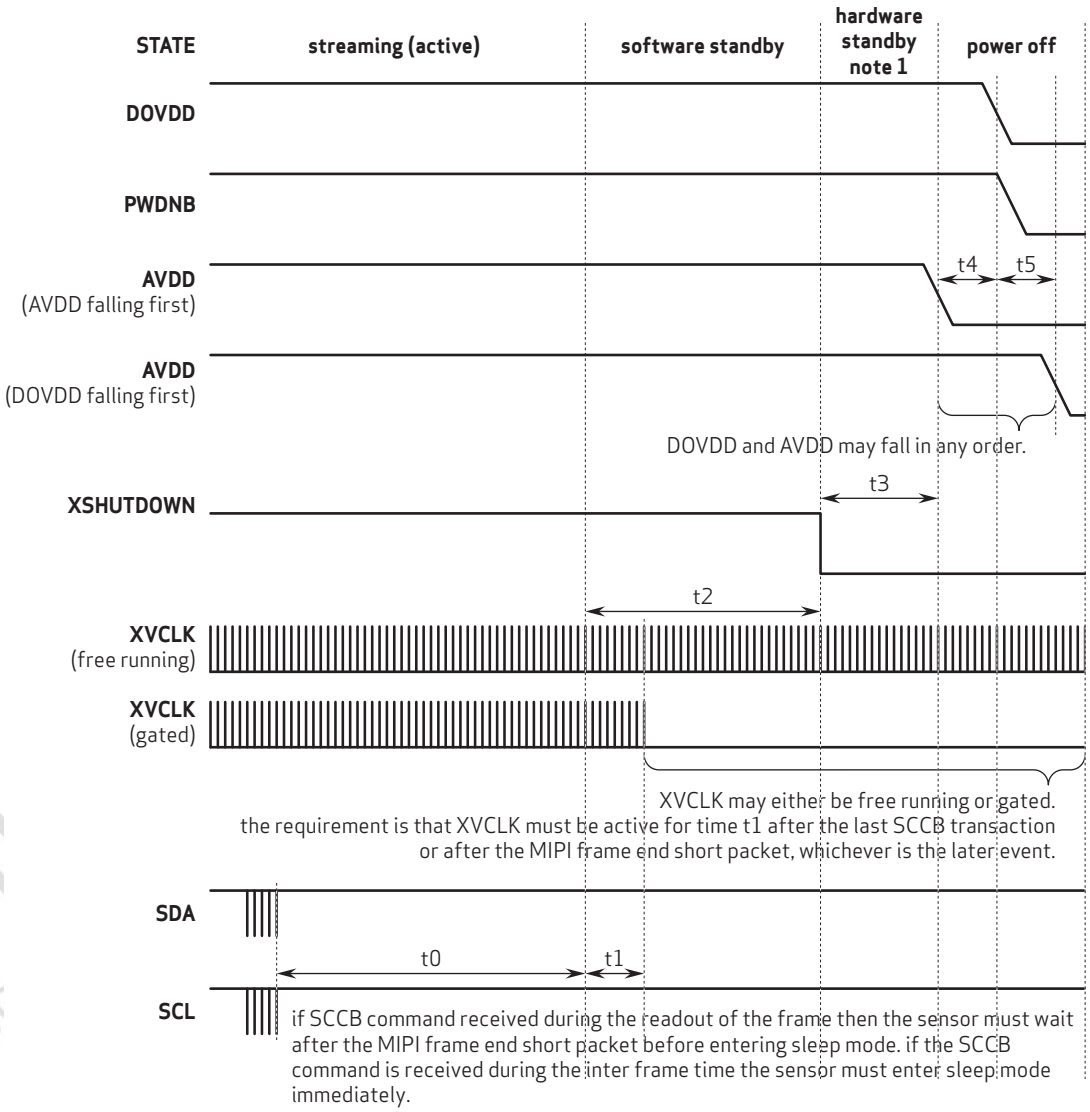
table 2-6 power down sequence

case	DVDD	XSHUTDOWN	PWDNB	power down sequence requirement
1	internal	GPIO	DOVDD	Refer to **figure 2-11** 1. software standby recommended 2. pull XSHUTDOWN low for minimum power consumption 3. AVDD and DOVDD may fall in any order
2	internal	DOVDD	GPIO	Refer to **figure 2-12** 1. software standby recommended 2. pull PWDNB low for low power consumption 3. pull DOVDD low for minimum power consumption or power off (XSHUTDOWN is connected to DOVDD) 4. pull AVDD low
3	external	GPIO	DOVDD	Refer to **figure 2-13** 1. software standby recommended 2. pull XSHUTDOWN low for low power consumption 3. cut off DVDD, then it will be in hardware standby state for minimum power consumption 4. pull AVDD and DOVDD low in any order
4	external	DOVDD	GPIO	Refer to **figure 2-14** 1. software standby recommended 2. pull PWDNB low for low power consumption 3. cut off DVDD, then it will be in hardware standby mode with minimum power consumption 4. pull DOVDD low (XSHUTDOWN connected to DOVDD) 5. pull AVDD low

table 2-7 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, XSHUTDOWN falling	t2	512		XVCLK cycles
XSHUTDOWN 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 infinite		ns
DOVDD falling - AVDD falling	t5			ns
PWDNB falling - DOVDD falling	t6	0.0		ns

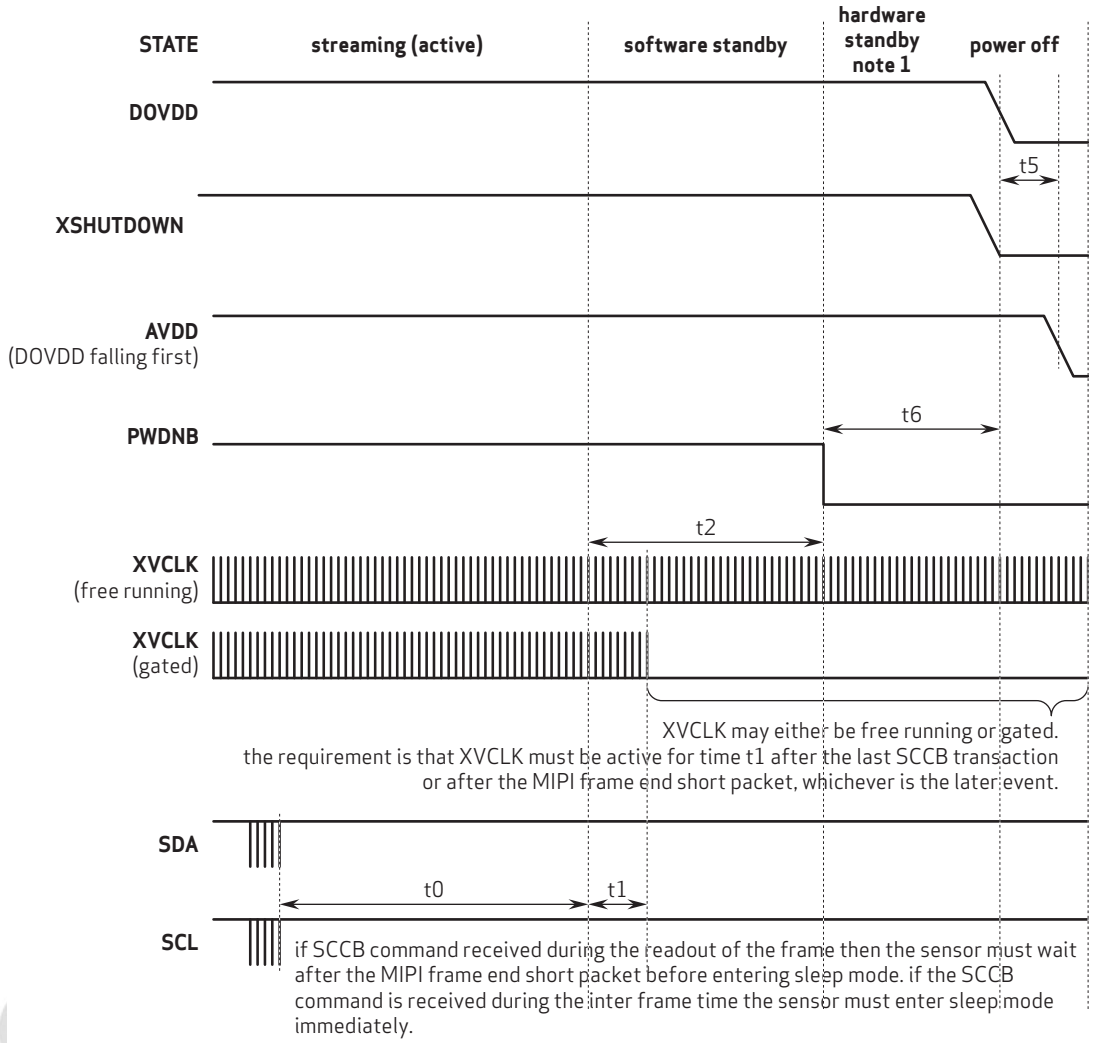
figure 2-11 power down sequence (case 1)



note 1 with minimum power consumption

9713_DS_2_11

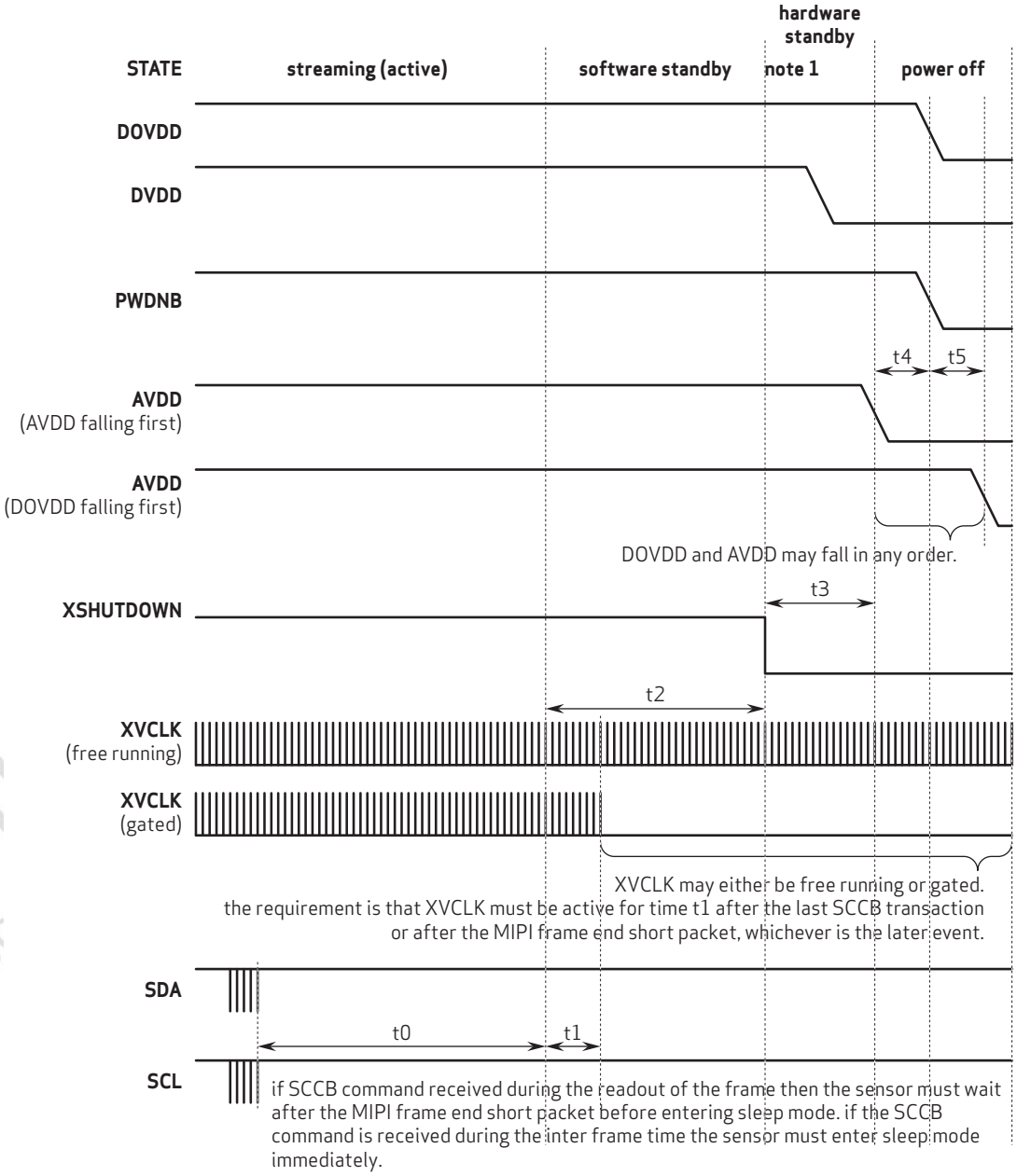
figure 2-12 power down sequence (case 2)



note 1 with low power consumption

9713_DS_2_12

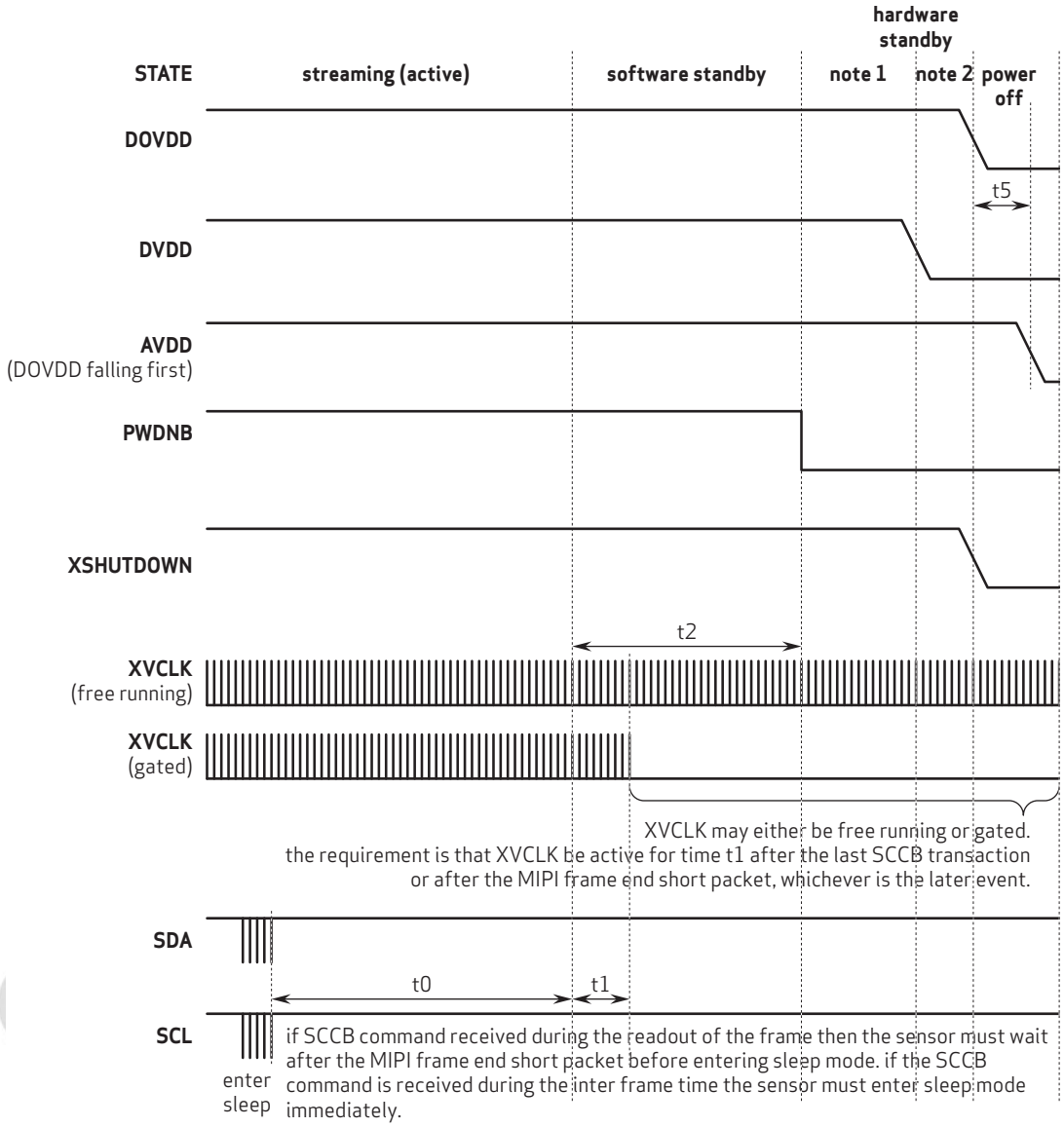
figure 2-13 power down sequence (case 3)



note 1 with low power consumption

9713_DS_2_13

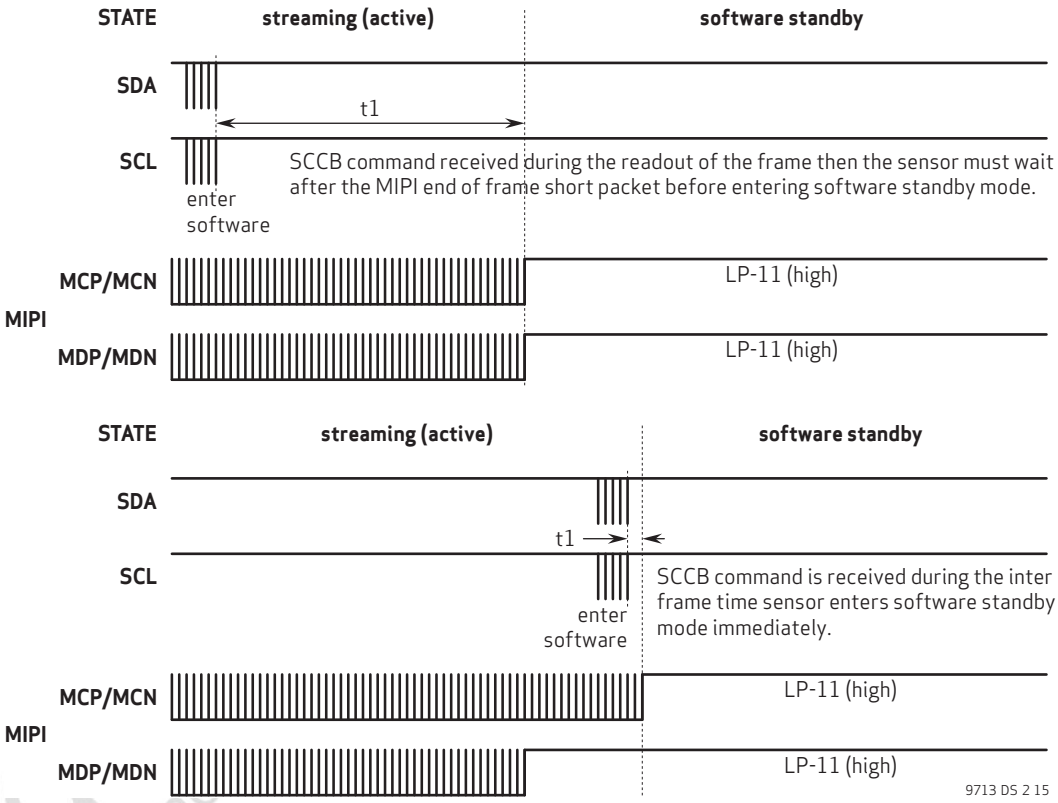
figure 2-14 power down sequence (case 4)



note 1 with low power consumption
note 2 with minimum power consumption

9713_DS_2_14

figure 2-15 standby sequence



2.8 reset

The OV9713 sensor includes a **XSHUTDOWN** pin (pin **B9**) that forces a complete hardware reset when it is pulled low (GND). The OV9713 clears all registers and resets them to their default values when a hardware reset occurs. Reset requires ~2ms settling time.

2.8.1 power ON reset generation

The power on reset can be controlled from external pin. However, inside this chip, a power on reset is generated after core power becomes stable.

2.9 hardware and software standby

Two suspend modes are available for the OV9713:

- hardware standby
- software standby

table 2-8 hardware and standby description

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

2.10 system clock control

The OV9713 has two on-chip PLLs which generate the system clock from a 6~27 MHz input clock. A programmable clock divider is provided to generate different frequencies for the system.

2.10.1 PLL configuration

note
contact your local OmniVision FAE for additional assistance on PLL configuration.

figure 2-16 OV9713 PLL1 clock diagram

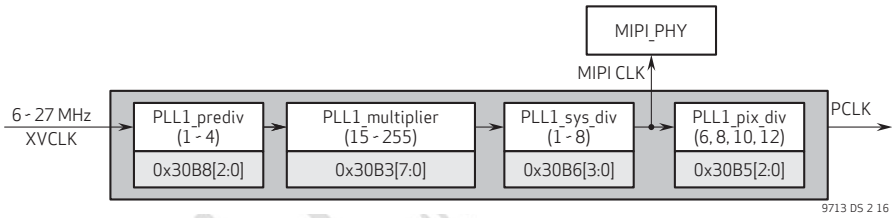


figure 2-17 OV9713 PLL2 clock diagram

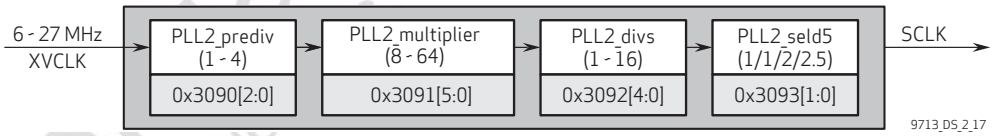


table 2-9 sample PLL configuration[a]

name	address	value
PLL2_prediv	0x3090[2:0]	3'h1
PLL2_multiplier	0x3091[5:0]	6'h0A
PLL2_divs	0x3092[4:0]	4'h3
PLL2_seld5	0x3093[1:0]	2'h1
PLL1_multiplier[8:0]	{0x30B2[0], 0x30B3[7:0]}	9'h28
PLL1_prediv	0x30B8[2:0]	3'h1
PLL1_op_pix_div	0x30B5[2:0]	3'h4
SCLK		80 MHz
MIPI_PCLK		60 MHz
PLL_mipiclk		480 MHz
XVCLK		24 MHz

a. PLL control for 1M @ 60 fps with 2 lane, 12-bit output

figure 2-18 clock connection diagram

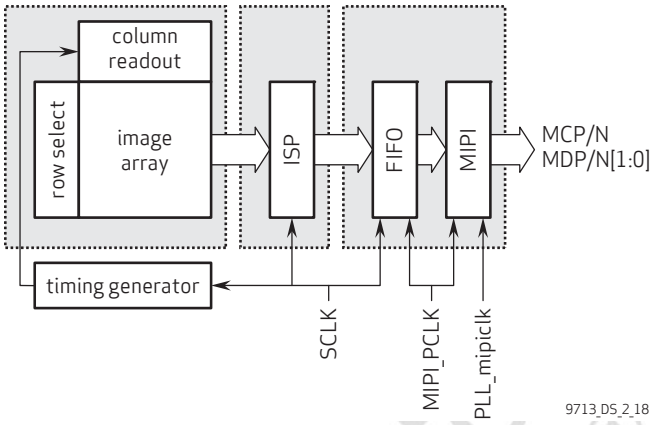


table 2-10 PLL speed limitation

parameter	value
PLL1_input	6~27 MHz
PLL1_VCO	300~1100 MHz
PLL2_input	6~27 MHz
PLL2_VCO	100~400 MHz
SCLK	max 84 MHz

2.11 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 OV9713, the SCCB ID is controlled by the SID pin. If SID is low, the sensor's SCCB address comes from register 0x0107 which has a default value of 0x20. If SID is high, the sensor's SCCB address comes from register 0x0108 which has a default value of 0x6C.

2.11.1 data transfer protocol

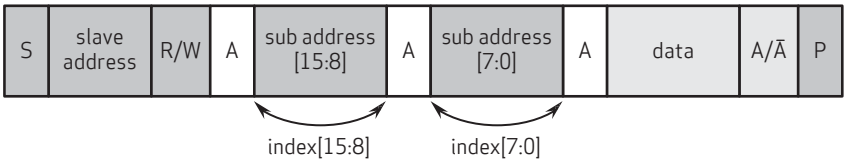
The data transfer of the OV9713 follows the SCCB protocol.

2.11.2 message format

The OV9713 supports the message format shown in **figure 2-19**. The repeated START (Sr) condition is not shown in SCCB single read from current location, but is shown in SCCB single or sequential read from random location.

figure 2-19 message type

message type: 16-bit sub-address, 8-bit data, and 7-bit slave address



from slave to master — S START condition — A acknowledge

from master to slave — P STOP condition — Ā negative acknowledge

direction depends on operation — Sr repeated START condition

9713_DS_2_19

2.11.3 read / write operation

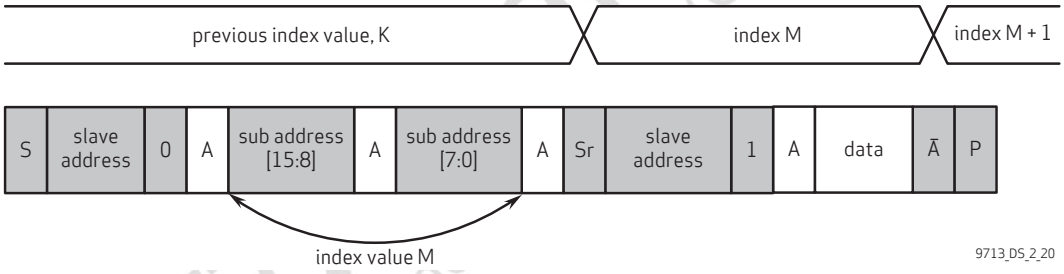
The OV9713 supports four different read operations and two different write operations:

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

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

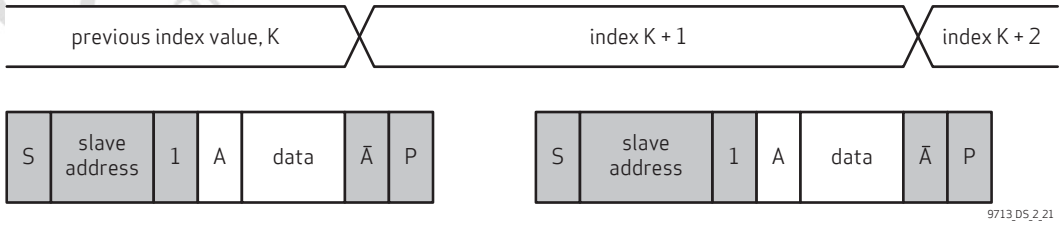
In a single read from random locations, the master does a dummy write operation to desired sub-address, issues a repeated start condition and then addresses the camera again with a read operation. After acknowledging its slave address, the camera starts to output data onto the SIOD line as shown in **figure 2-20**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 2-20 SCCB single read from random location



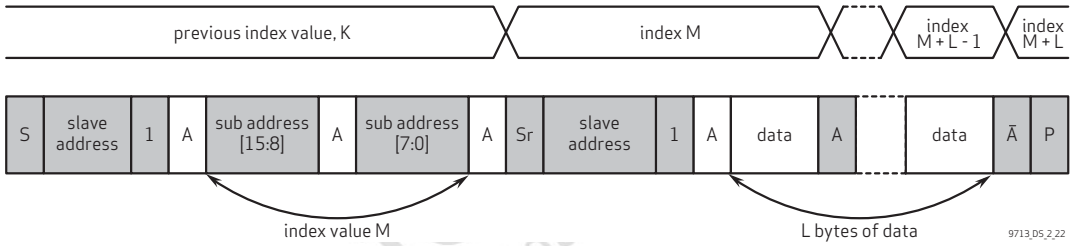
If the host addresses the camera with read operation directly without the dummy write operation, the camera responds by setting the data from last used sub-address to the SIOD line as shown in **figure 2-21**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 2-21 SCCB single read from current location



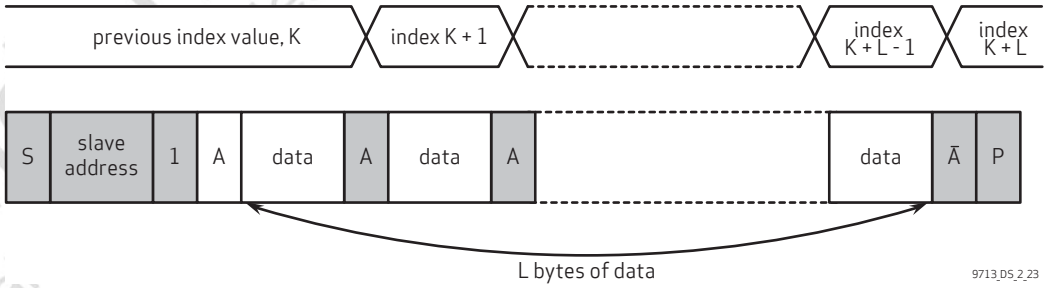
The sequential read from a random location is illustrated in **figure 2-22**. The master does a dummy write to the desired sub-address, issues a repeated start condition after acknowledge from slave and addresses the slave again with read operation. If a master issues an acknowledge after receiving data, 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 2-22 SCCB sequential read from random location



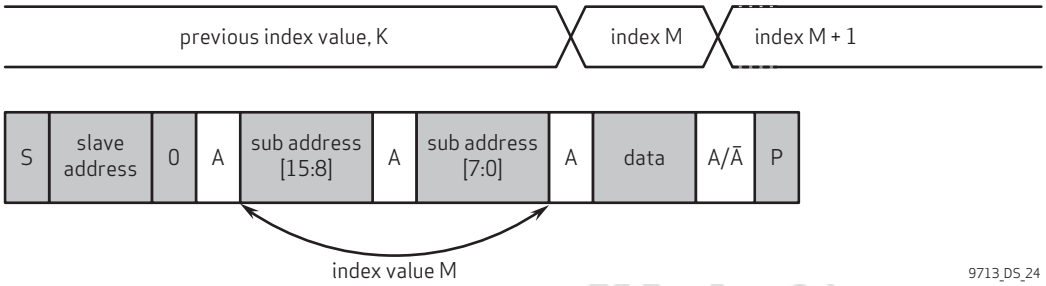
The sequential read from current location is similar to a sequential read from a random location. The only exception is that there is no dummy write operation, as shown in **figure 2-23**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 2-23 SCCB sequential read from current location



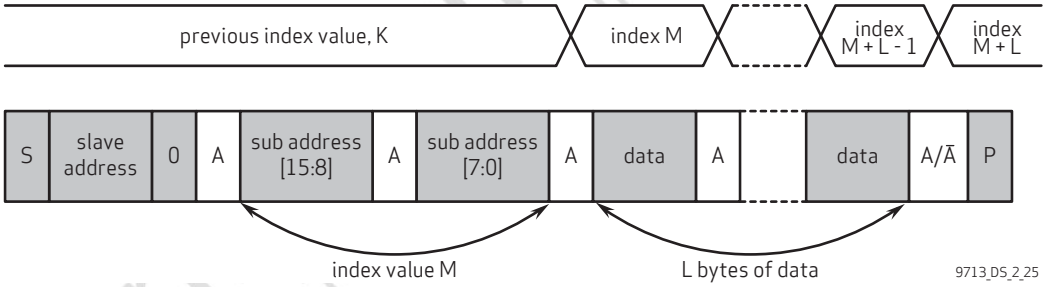
The write operation to a random location is illustrated in **figure 2-24**. The master issues a write operation to the slave, sets the sub-address and data correspondingly after the slave has acknowledged. The write operation is terminated with a stop condition from the master.

figure 2-24 SCCB single write to random location



The sequential write is illustrated in **figure 2-25**. The slave automatically increments the sub-address after each data byte. The sequential write operation is terminated with stop condition from the master.

figure 2-25 SCCB sequential write to random location



2.11.4 SCCB timing

figure 2-26 SCCB interface timing

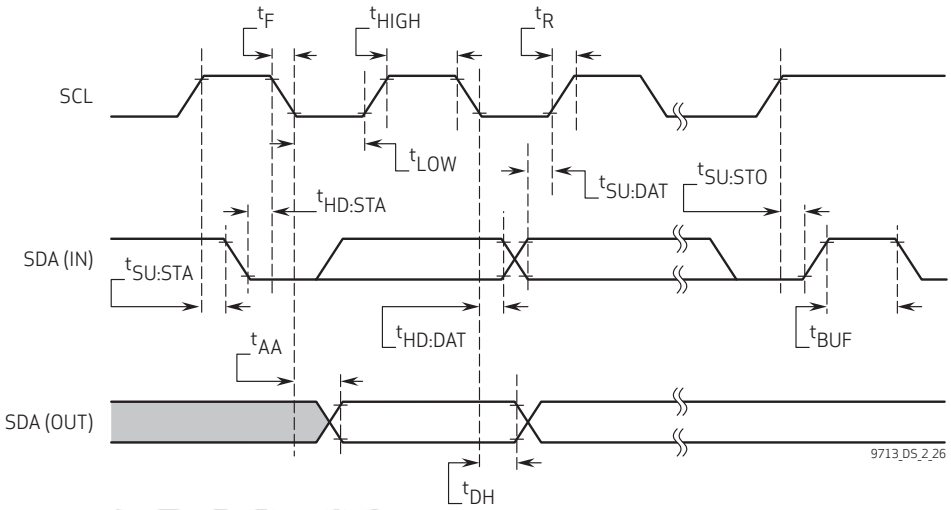


table 2-11 SCCB interface timing specifications[ab]

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 or the end of the falling edge signifies 10%, timing measurement shown in the middle of the rising/falling edge signifies 50%, timing measurement shown at the beginning of the falling edge or the end of the rising edge signifies 90%

2.11.5 group write

Group write is supported in order to update a group of registers in the same frame. These registers are guaranteed to be written prior to the internal latch at the frame boundary.

The OV9713 supports up to four groups. These groups share 512 bytes buffer and the size of each group is programmable by adjusting the start address.

table 2-12 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 launch 1110: Immediate launch others: Reserved 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: Reserved
0x3209	GRP0_PERIOD	0x00	RW	Frames For Staying in Group 0
0x320A	GRP1_PERIOD	0x00	RW	Frames For Staying in Group 1
0x320B	GRP_SWCTRL	0x01	RW	Bit[7]: auto_sw Bit[3]: group_switch_repeat Bit[2]: context_en Bit[1:0]: Second group selection

3 block level description

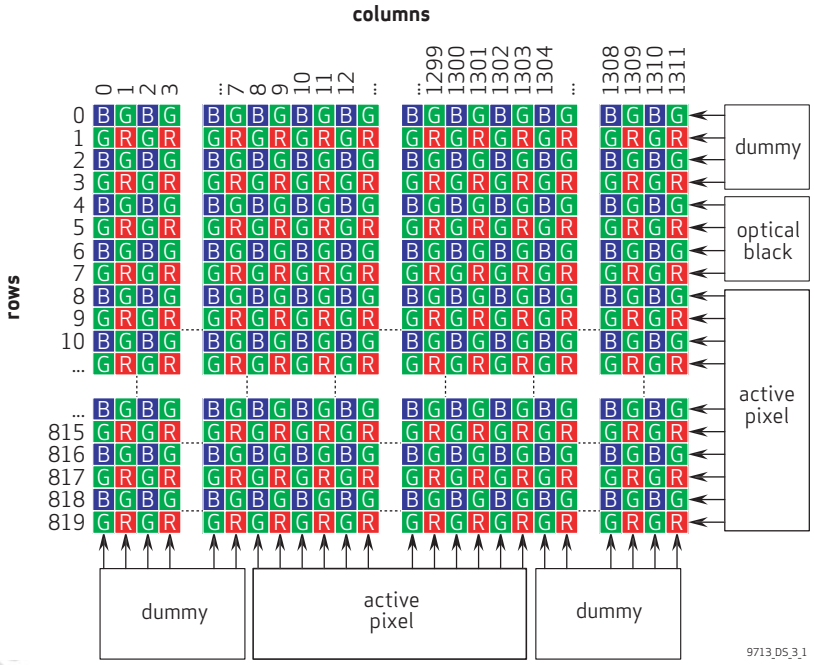
3.1 pixel array structure

The OV9713 sensor has an image array of 1312 columns by 820 rows (1,075,840 pixels). figure 3-1 shows a cross-section of the image sensor array.

The color filters are arranged in a Bayer pattern. The primary color BG/GR array is arranged in line-alternating fashion. Of the 1,075,840 pixels, 1,052,352 (1296x812) are active pixels and can be output. The other pixels are used for black level calibration and DSP.

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

figure 3-1 sensor array region color filter layout



3.2 subsampling

There are two subsampling modes in the OV9713: binning and skipping. Both are acceptable methods of reducing output resolution while maintaining the field of view. Binning is usually preferred as it increases the pixel's signal-to-noise ratio. When the binning function is ON, voltage levels of adjacent pixels are averaged. In skipping mode (binning function is OFF), alternate pixels, which are not output, are merely skipped. The OV9713 supports 2x2 binning. **figure 3-2** illustrates 2x2 binning, where the voltage levels of two horizontal (2x1) adjacent same-color pixels are averaged before entering the ADC. See **table 3-1** for horizontal and vertical binning registers.

figure 3-2 example of 2x2 binning

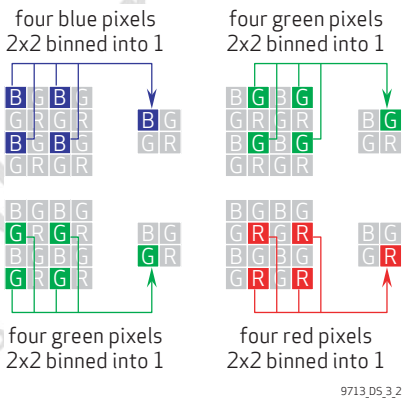


table 3-1 binning-related registers

address	register name	default value	R/W	description
0x370E	REGE	0x00	RW	Bit[3]: Vertical binning
0x3821	FORMAT2	0x00	RW	Bit[0]: Horizontal binning

4 image sensor core digital functions

4.1 mirror and flip

The OV9713 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

9713_DS_4_1

table 4-1 mirror and flip registers

address	register name	default value	R/W	description
0x0101	IMAGE_ORIENTATION	0x00	RW	Timing Control Register Bit[1]: Vertical flip enable 0: Normal 1: Vertical flip
0x0101	IMAGE_ORIENTATION	0x00	RW	Timing Control Register Bit[0]: 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 simply masking off the pixels outside of the window; thus, the original timing is not affected.

figure 4-2 image windowing

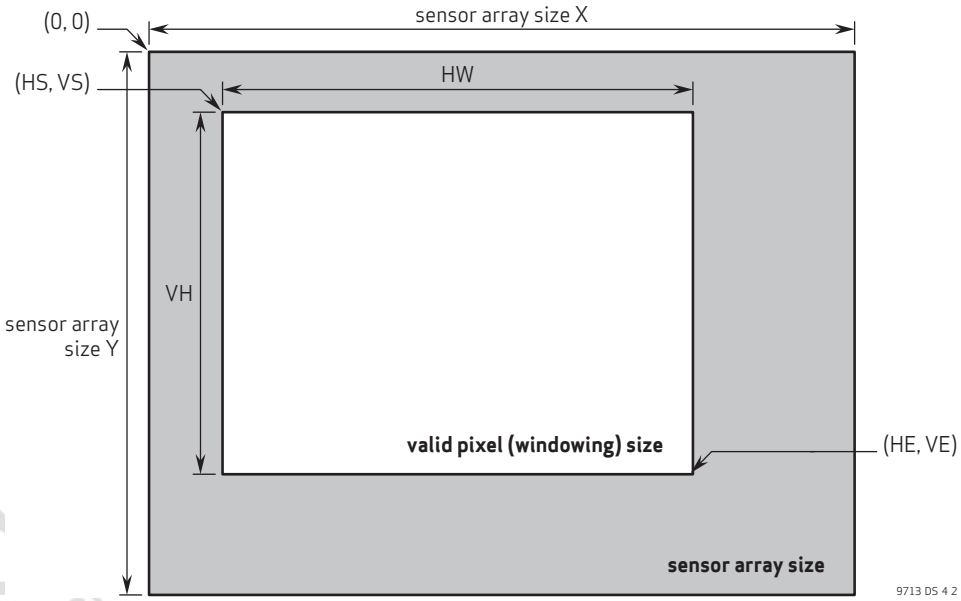


table 4-2 image windowing control functions

function	register	R/W	description
horizontal start	{0x0344, 0x0345}	RW	HS[11:8] = 0x0344 HS[7:0] = 0x0345
vertical start	{0x0346, 0x0347}	RW	VS[11:8] = 0x0346 VS[7:0] = 0x0347
horizontal end	{0x0348, 0x0349}	RW	HW[11:8] = 0x0348 HW[7:0] = 0x0349
vertical end	{0x034A, 0x034B}	RW	VH[11:8] = 0x034A VH[7:0] = 0x034B

4.3 test pattern

For testing purposes, the OV9713 offers three types of test pattern: color bar, square, random data. And the OV9713 also offers two effects: transparent effect and rolling bar effect. The output type of test pattern is controlled by test_pattern_type register (0x5080[1:0]).

4.3.1 color bar

There are four types of color bar which are switched by bar-style register (0x5080[3:2])

figure 4-3 color bar types

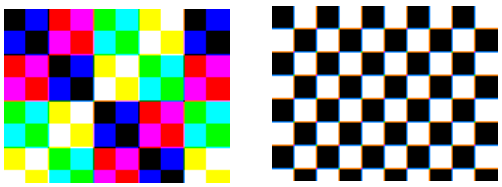
color bar type 1 color bar type 2

color bar type 3 color bar type 4

4.3.2 square

There are two types of squares: color square and black-white square. The square_mode register (0x5080[4]) decides which type of square will be output.

figure 4-4 test patterns



color square black-white square

4.3.3 random data

There are two types of random data test pattern: frame-changing and frame-fixed random data. The output type of random data is decided by same_seed_en register (0x5081[4]). The random seed is set by seed register (0x5081[3:0]).

4.3.4 transparent effect

The transparent effect is enabled by transparent_en register (0x5080[5]). If this register is set, the transparent test pattern will be gotten. **figure 4-5** shows an example of transparent color bar image.

figure 4-5 transparent effect

4.3.5 rolling bar effect

The rolling bar is set by rolling_bar_en register (0x5080[6]). If it is set, a inverted-color rolling bar will roll from up to down. **figure 4-6** shows an example of a rolling bar on color bar image.

figure 4-6 rolling bar effect

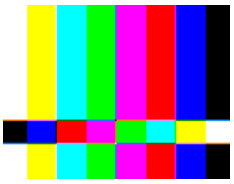


table 4-3 general color bar selection control

function	register	default value	R/W	description
general color bar	0x5080	0x00	RW	Bit[7]: test_enable 0: disable test function 1: enable test function Bit[6]: rolling_enable 0: disable rolling bar function 1: enable rolling bar function Bit[5]: transparent_en 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_mode 00: color bar 01: random data 10: square 11: black image
random data	0x5081	0x00	RW	Bit[4]: same_seed_en when set, the seed used to generate the random data are same which is set Bit[3:0]: seed used in generating random data

4.4 black level calibration (BLC)

The pixel array contains several optically shielded (black) rows. These rows are used as reference for black level calibration.

Black level adjustments can be made with the following registers.

table 4-4 BLC control functions (sheet 1 of 2)

address	register name	default value	R/W	description
0x4000	BLC CTRL0	0x0C	–	Debug Mode
0x4001	START LINE	0x04	–	Debug Mode

table 4-4 BLC control functions (sheet 2 of 2)

address	register name	default value	R/W	description
0x4002	BLC CTRL2	0x45	RW	Bit[7]: format_change_en Format change trigger enable signal When set to 1, BLC will be applied to N frames in bit[5:0] Bit[6]: blc_auto_en BLC auto mode enable signal 0: black levels are set by registers (0x400C~0x401B) 1: Black levels are calculated values Bit[5:0]: reset_frame_num Specify the frame number for reset and/or format change BLC trigger
0x4003	BLC CTRL3	0x01	RW	Bit[7]: blc_trig_man Manual trigger enables signal. When enabled BLC will be applied to N frames in bit[5:0] Bit[6]: Freeze Has lower priority than always_do When set and always_do is set to 0, BLC will freeze Bit[5:0]: manual_frame_num Specify BLC update frame number for manual trigger
0x4004	LINE NUM	0x08	RW	Bit[7:0]: Debug Mode
0x4005	BLC CTRL5	0x40	RW	Bit[7:2]: Debug mode Bit[1]: blc_always_up_en 0: Normal freeze 1: BLC always update
0x4007	BLC CTRL7	0x20	RW	Bit[5]: gain_change_en Gain change trigger enable signal. When set to 1, BLC will be applied in the following frame Bit[4:0]: Debug mode
0x4008	BLC TARGET	0x00	RW	Bit[7:0]: Blackleveltarget0[15:8] Blackleveltarget0 high 8 bits
0x4009	BLC TARGET	0x40	RW	Bit[7:0]: Blackleveltarget0[7:0] Blackleveltarget0 low 8 bits

4.5 one time programmable (OTP) memory

The OV9713 supports a maximum of 256 bits of one-time programmable (OTP) memory to store chip identification and manufacturing information. The first 40 bits of the OTP memory in the first bank of OTP is reserved for OmniVision internal use.The last 32 bits of the OTP memory in the first bank is reserved for PWM function. It is used to store PWM setting parameters.The remaining bits in bank 1 and 128 bits in bank2 are fully user programmable. They can be controlled through the SCCB (see **table 4-5**).

The operating voltage on AVDD for OTP operations is 2.6V~3.0V.

The OTP memory cannot be directly accessed. Instead, it is accessed through its register buffer 0x3D00~0x3D0F. The OV9713 OTP supports two banks, Both banks use the same register buffers and each bank can be selected by register 0x3D84 together with the respective OTP memory start address and end address. Registers 0x3D80[0] and 0x3D81[0] are the "program" and "read" command registers of the OTP memory controller, respectively.

After selecting the OTP bank, when the user writes 0x01 to command register 0x3D81, the OTP memory controller will load the content of all OTP memory bits to its corresponding register buffer. After that, the user can read the OTP content from its register buffer.

When the user writes 0x01 to command register 0x3D80, the OTP memory controller will program the data of registers 0x3D00~0x3D0F to its corresponding OTP memory bits in the pre-set OTP bank, so the user should always program the OTP data to register buffers 0x3D00~0x3D0F before issuing the OTP program command.

When the OTP memory controller is programming data to OTP memory or reading data from OTP memory, the sensor will not respond to any SCCB access. Because OTP programming current is quite big, accessing sensor registers is prohibited in order to prevent any glitch on the power supply. Waiting 10ms after issuing the OTP read and program commands is recommended.

table 4-5 OTP control functions

function	register	description
OTP program	0x3D80	Bit[0]: program OTP
OTP load/dump	0x3D81	Bit[0]: load / dump OTP
OTP bank select	0x3D84	Bit[7]: program_dis 0: enable 1: disable Bit[6]: memory bank enable 0: auto memory bank mode 1: manual memory bank mode Bit[5:0]: memory bank select
OTP start address	0x3D85	Bit[3:0]: start byte index for the 16 bytes in memory bank
OTP end address	0x3D86	Bit[3:0]: end byte index for the 16 bytes in memory bank
OTP load enable	0x3D88	Bit[7]: load enable
dump / program data n	0x3D00~0x3D0F	Bit[7:0]: data dumped or data to be programmed for bits[8*(n+1)-1:8*n]

note
default settings are for typical 24MHz input clock and 80MHz sysclk (please contact your local OmniVision FAE, when using a different clock)

4.6 strobe flash

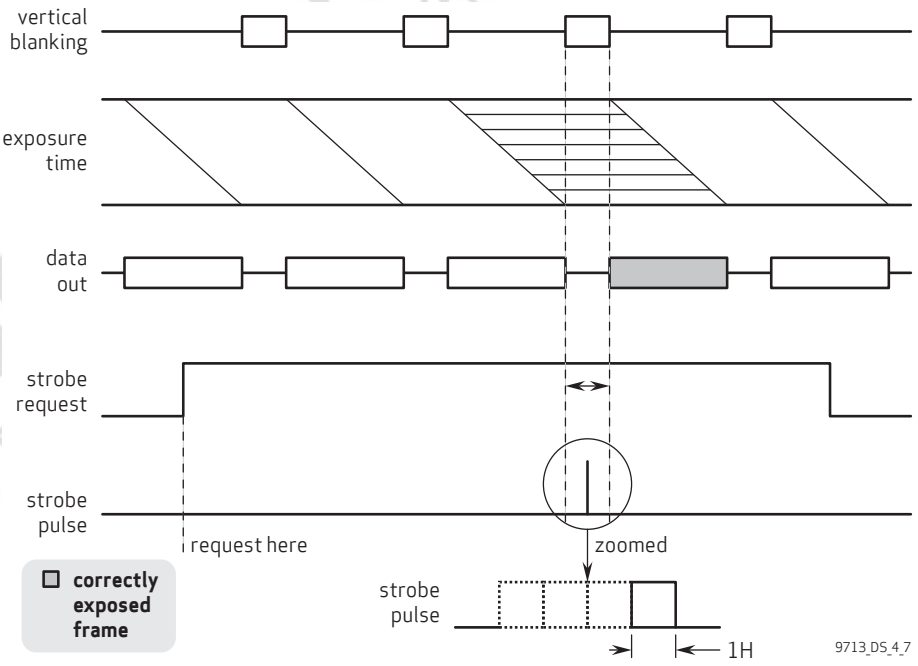
4.6.1 strobe flash control

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 0x3B00[7]. Flash modules are triggered by the rising edge by default or by the falling edge if the signal polarity is changed. OV9713 supports the following flashlight modes.

4.6.1.1 xenon flash control

After a strobe request is submitted, the strobe pulse will be activated at the beginning of the third frame (see figure 4-7). 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 4-7 xenon flash mode



4.6.1.2 LED 1 & 2 mode

In LED 1 & 2 modes, the strobe pulse is active two frames after the strobe request is submitted and the third frame is correctly exposed. The strobe pulse will be activated only one time if the strobe end request is set as shown in **figure 4-8**. If end request has not been sent, the strobe signal is activated intermittently until the strobe end request is set (see **figure 4-9**). For a longer strobe pulse, a number of dummy lines can be inserted by controlling {0x3B02, 0x3B03}. When the strobe request is cleared, the dummy lines will be removed automatically.

figure 4-8 LED 1 & 2 mode - one pulse output

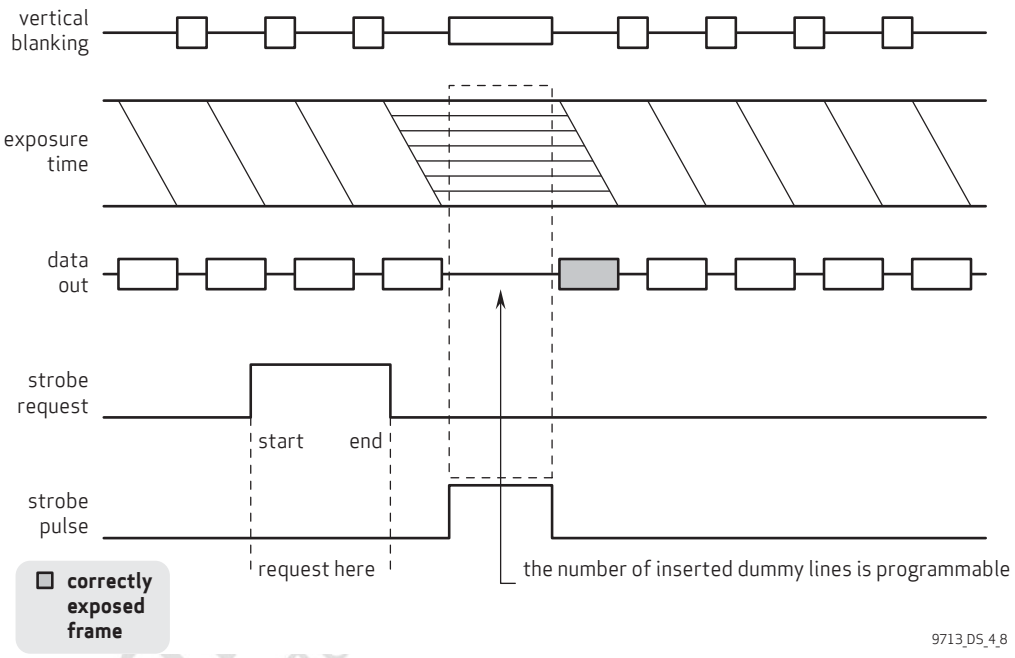
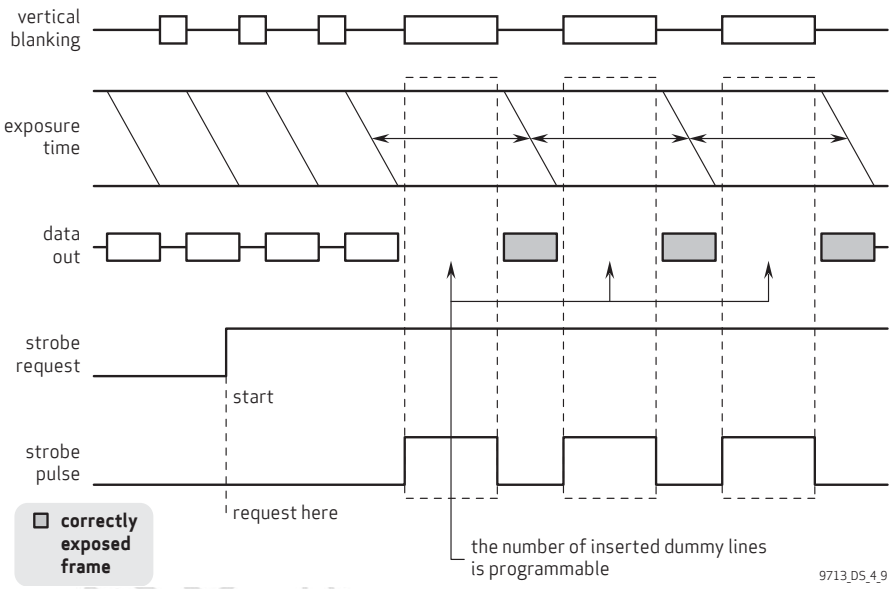


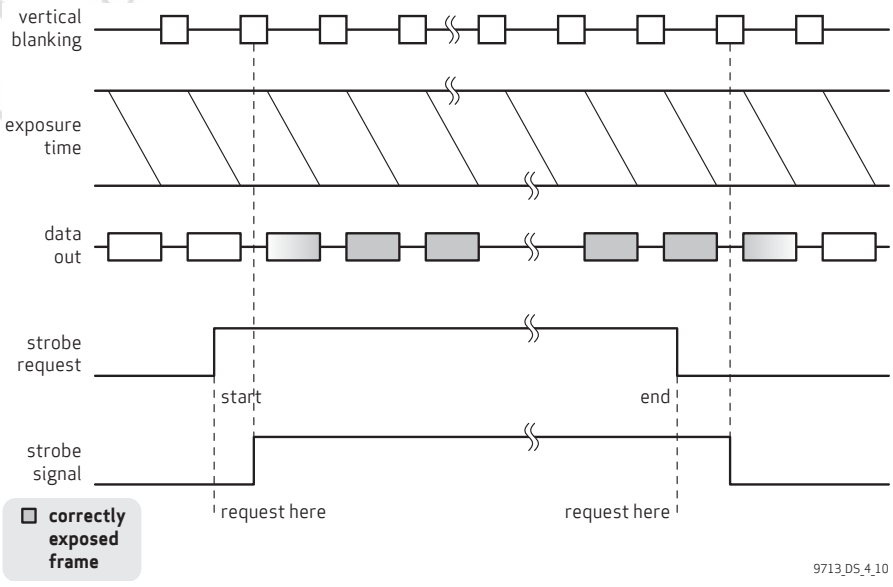
figure 4-9 LED 1 & 2 mode - multiple pulse output



4.6.1.3 LED 3 mode

In LED 3 mode, the strobe signal stays active until the strobe end request is sent (see **figure 4-10**).

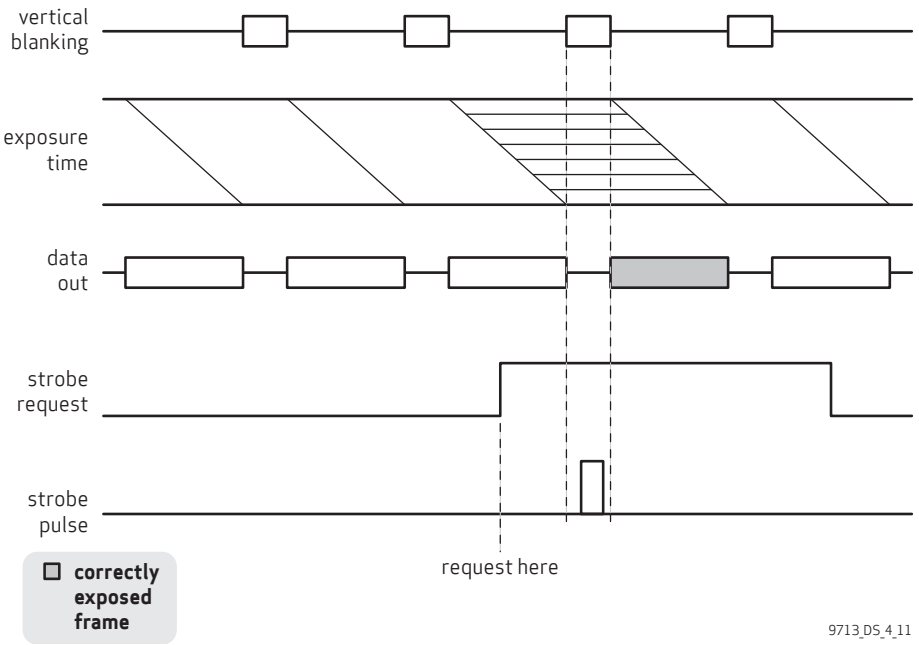
figure 4-10 LED 3 mode



4.6.1.4 LED 4 mode

In LED 4 mode, the strobe signal width is controlled by register 0x3B05 (see **figure 4-11**). Strobe width = 128 × (2^0x3B05[1:0]) × (0x3B05[7:2] + 1) × sclk_period. The maximum value of 0x3B02[7:2] is 6'b111110.

figure 4-11 LED 4 mode



See **table 4-6** for LED strobe control functions.

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

address	register name	default value	R/W	description
0x3B00	STROBE CTRL00	0x00	RW	Bit[7]: Strobe request ON/OFF Bit[6]: Strobe polarity 0: Active high 1: Active low Bit[5:4]: Pulse width in xenon mode Bit[2:0]: Strobe mode select 000: Xenon 001: LED1 010: LED2 011: LED3 100: LED4

table 4-6 LED strobe control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x3B04	STROBE CTRL04	0x00	RW	Bit[3]: start_point_sel Bit[2]: Strobe repeat enable Bit[1:0]: Strobe latency 00: Strobe generated at next frame 01: Delay one frame, strobe generated 2 frames later 10: Delay two frames, strobe generated 3 frames later 11: Delay three frames, strobe generated 4 frames later
0x3B05	STROBE CTRL05	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

5 image sensor processor digital functions

5.1 ISP general controls

The ISP module provides image processor functions, including lens correction, and defect pixel cancellation.

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

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xFF	RW	Bit[7]: Debug mode Bit[6]: dbc_en DBC function enable signal 0: Disable 1: Enable Bit[5]: otp_en OTP function enable signal 0: Disable 1: Enable Bit[4]: awbg_en AWBG function enable signal 0: Disable 1: Enable Bit[3]: bc_en Black DPC function enable signal 0: Disable 1: Enable Bit[2]: wc_en White DPC function enable signal 0: Disable 1: Enable Bit[1]: lenc_en LENC function enable signal 0: Disable 1: Enable Bit[0]: isp_en ISP functions enable signal 0: Disable 1: Enable
0x5001	ISP CTRL01	0x1F	RW	Bit[7:5]: Debug mode Bit[4]: avg_en AVG function enable signal 0: Disable 1: Enable Bit[3]: Debug mode Bit[2]: blc_en BLC function enable signal 0: Disable 1: Enable Bit[1:0]: Debug mode

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

address	register name	default value	R/W	description
0x5002	ISP CTRL02	0x00	RW	Bit[7]: Debug mode Bit[6]: avg_sel Signal select for AVG input 0: Select WINC output as AVG input 1: Select DGC output as AVG input Bit[5:0]: Debug mode
0x5004	RED GAIN	0x04	RW	Bit[3:0]: red_gain[11:8] AWB gain for red channel
0x5005	RED GAIN	0x00	RW	Bit[7:0]: red_gain[7:0] AWB gain for red channel
0x5006	GRN GAIN	0x04	RW	Bit[3:0]: grn_gain[11:8] AWB gain for green channel
0x5007	GRN GAIN	0x00	RW	Bit[7:0]: grn_gain[7:0] AWB gain for green channel
0x5008	BLU GAIN	0x04	RW	Bit[7:4]: Not Used Bit[3:0]: blu_gain[11:8] AWB gain for blue channel
0x5009	BLU GAIN	0x00	RW	Bit[7:0]: blu_gain[7:0] AWB gain for blue channel
0x500A	ISP CTRL0A	0x85	RW	Bit[7:0]: Debug mode
0x500B	ISP CTRL0B	0x10	RW	Bit[7:0]: Debug mode
0x500C	ISP CTRL0C	0x00	RW	Bit[7:0]: Debug mode

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

address	register name	default value	R/W	description
0x5080	CTRL00	0x00	RW	Bit[7]: test_en 0: Disable test function 1: Enable test function Bit[6]: rolling_bar_en 0: Disable rolling bar function 1: Enable rolling bar function Bit[5]: transparent_en 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_mode 00: Color bar 01: Random data 10: Square 11: Black image
0x5081	CTRL01	0x01	RW	Bit[6]: win_cut_en 0: Do not cut the redundant pixels 1: Cut the redundant pixels Bit[5]: two_lsb_0_en When set, two LSB of output data are 0 Bit[4]: same_seed_en When set, the seed used to generate the random data is the same, set by bit[3:0] Bit[3:0]: Seed Seed used in generating random data
0x5082~ 0x5091	PRE REG	–	RW	Debug Mode

5.2 defect pixel cancellation (DPC)

Primarily due to processes anomalies, pixel defects in the sensor array will occur, generating incorrect pixel levels and color values. The purpose of the DPC is to remove the effects caused by defective pixels.

table 5-2 DPC registers

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xFF	RW	Bit[3]: bc_en Black DPC function enable signal 0: Disable 1: Enable Bit[2]: wc_en White DPC function enable signal 0: Disable 1: Enable
0x5400	DPC CTRL00	0x1C	RW	Bit[7:5]: Debug mode Bit[4]: sc_en Enable removing defect pixels in same channel 0: Disable 1: Enable Bit[3]: dc_en Enable removing defect pixels in different channel 0: Disable 1: Enable Bit[2]: smooth_en Enable using average G values of 3 lines for recovery 0: Disable 1: Enable Bit[1:0]: Debug mode
0x5401~ 0x5416	DPC REG	–	RW	Debug Mode

5.3 MWB

The manual white balance (MWB) provides digital gain for R, G, and B channels. Each channel gain is 12-bit. 0x400 is 1x gain.

table 5-3 manual AWB_GAIN registers

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xFF	RW	Bit[4]: awbg_en AWBG function enable signal 0: Disable 1: Enable
0x5004	RED GAIN	0x04	RW	Bit[3:0]: red_gain[11:8] AWB gain for red channel
0x5005	RED GAIN	0x00	RW	Bit[7:0]: red_gain[7:0] AWB gain for red channel
0x5006	GRN GAIN	0x04	RW	Bit[3:0]: grn_gain[11:8] AWB gain for green channel
0x5007	GRN GAIN	0x00	RW	Bit[7:0]: grn_gain[7:0] AWB gain for green channel
0x5008	BLU GAIN	0x04	RW	Bit[3:0]: blu_gain[11:8] AWB gain for blue channel
0x5009	BLU GAIN	0x00	RW	Bit[7:0]: blu_gain[7:0] AWB gain for blue channel

5.4 lens correction (LENC)

The lens correction (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.

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

address	register name	default value	R/W	description
0x5200	RED X0	0x02	RW	Bit[2:0]: red_x0[10:8] 3 MSB of horizontal center position of red channel
0x5201	RED X0	0x90	RW	Bit[7:0]: red_x0[7:0] 8 LSB of horizontal center position of red channel

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

address	register name	default value	R/W	description
0x5202	RED Y0	0x01	RW	Bit[1:0]: red_y0[9:8] 2 MSB of vertical center position of red channel
0x5203	RED Y0	0x9A	RW	Bit[7:0]: red_y0[7:0] 8 LSB of vertical center position of red channel
0x5204	RED A1	0x22	RW	Bit[6:0]: red_a1 Gain coefficient for the square of distance for current red pixel position and red center
0x5205	RED A2	0x07	RW	Bit[3:0]: red_a2 Precision of gain coefficient for the square of distance of current red pixel position and red center
0x5206	RED B1	0xC2	RW	Bit[7:0]: red_b1 Gain coefficient for the distance of current red pixel position and red center
0x5207	RED B2	0x08	RW	Bit[3:0]: red_b2 Precision of gain coefficient for the distance of current red pixel position and red center Gain of current red pixel is defined with the distance (r), red_a1, red_a2, red_b1 and red_b2 Gain = [(red_a1* r2)>>red_a2] + [(red_b1*r)>>red_b2]
0x5208	GRN X0	0x02	RW	Bit[2:0]: grn_x0[10:8] 3 MSB of horizontal center position of green channel
0x5209	GRN X0	0x90	RW	Bit[7:0]: grn_x0[7:0] 8 LSB of horizontal center position of green channel
0x520A	GRN Y0	0x01	RW	Bit[1:0]: grn_y0[9:8] 2 MSB of vertical center position of green channel
0x520B	GRN Y0	0x9A	RW	Bit[7:0]: grn_y0[7:0] 8 LSB of vertical center position of green channel
0x520C	GRN A1	0x22	RW	Bit[6:0]: grn_a1 Gain coefficient for the square of distance of current blue pixel position and green center

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

address	register name	default value	R/W	description
0x520D	GRN A2	0x07	RW	Bit[3:0]: grn_a2 Precision of gain coefficient for the square of distance of current green pixel position and green center
0x520E	GRN B1	0xC2	RW	Bit[7:0]: grn_b1 Gain coefficient for the distance of current green pixel position and green center
0x520F	GRN B2	0x08	RW	Bit[3:0]: grn_b2 Precision of gain coefficient for the distance of current green pixel position and green center The gain of current green pixel is defined by the distance (r), grn_a1, grn_a2, grn_b1 and grn_b2. Gain = [(grn_a1* r2)>>grn_a2] + [(grn_b1*r)>>grn_b2]
0x5210	BLU X0	0x02	RW	Bit[2:0]: blu_x0[10:8] 3 MSB of horizontal center position of blue channel
0x5211	BLU X0	0x90	RW	Bit[7:0]: blu_x0[7:0] 8 LSB of horizontal center position of blue channel
0x5212	BLU Y0	0x01	RW	Bit[1:0]: blu_y0[9:8] 2 MSB of vertical center position of blue channel
0x5213	BLU Y0	0x9A	RW	Bit[7:0]: blu_y0[7:0] 8 LSB of vertical center position of blue channel
0x5214	BLU A1	0x22	RW	Bit[6:0]: blu_a1 Gain coefficient for the square of distance of current blue pixel position and blue center
0x5215	BLU A2	0x07	RW	Bit[3:0]: blu_a2 Precision of gain coefficient for the square of distance of current blue pixel position and blue center
0x5216	BLU B1	0xC2	RW	Bit[7:0]: blu_b1 Gain coefficient for the distance of current blue pixel position and blue center

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

address	register name	default value	R/W	description
0x5217	BLU B2	0x08	RW	Bit[3:0]: blu_b2 Precision of gain coefficient for the distance of current blue pixel position and blue center Gain of current blue pixel is defined with the distance (r), blu_a1, blu_a2, blu_b1 and blu_b2. Gain = [(blu_a1* r2)>>blu_a2] + [(blu_b1*r)>>blu_b2]
0x5218	LENC CTRL00	0x04	RW	Bit[2]: rnd_en Round enable generates random round bit Bit[1]: adj_coef_man_en Manual adjust coefficient enable 0: Disable 1: Enable Bit[0]: lenc_gain_adj_coef_en LENC gain adjust coefficient enable signal 0: Disable 1: Enable
0x5219	COEF TH	0x80	RW	Bit[7:0]: coef_th[7:0] Coefficient threshold for higher real gain
0x521A	GAIN TH1	0x06	RW	Bit[6:0]: gain_th1 Lower threshold of real gain. When the current real gain is less than gain_th1, the adjust coefficient of LENC gain is 1
0x521B	GAIN TH2	0x00	RW	Bit[6:0]: gain_th2 Higher threshold of real gain. When the current real gain is larger than gain_th2, the adjust coefficient of LENC gain is adj_coef_th which has 7-bit precision. When current real gain is less than gain_th2 and larger than gain_th1, adjust coefficient is [128-(real_gain-gain_th1)*(128-adj_coef_th) / (gain_th2-gain_th1)]
0x521C	COEF MAN	0x80	RW	Bit[7:0]: coef_man[7:0] Manual adjust coefficient for LENC gain with 7-bit precision Range is [0,128]

5.5 average (AVG)

The main function of AVG is to average the data channel value using special filters.

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

address	register name	default value	R/W	description
0x5900	SUBWIN XSTART	0x00	RW	Bit[4:0]: xstart[12:8] AVG sub-window horizontal start position[12:8]
0x5901	SUBWIN XSTART	0x00	RW	Bit[7:0]: xstart[7:0] AVG sub-window horizontal start position[7:0]
0x5902	SUBWIN YSTART	0x00	RW	Bit[3:0]: ystart_sub[11:8] AVG sub-window vertical start position[11:8]
0x5903	SUBWIN YSTART	0x00	RW	Bit[7:0]: ystart_sub[7:0] AVG sub-window vertical start position[7:0]
0x5904	SUBWIN HSIZE	0x05	RW	Bit[4:0]: hsize_sub Sub-window width[12:8]
0x5905	SUBWIN HSIZE	0x00	RW	Bit[7:0]: hsize_sub Sub-window width[7:0]
0x5906	SUBWIN VSIZE	0x02	RW	Bit[3:0]: vsize_sub Sub-window height[11:8]
0x5907	SUBWIN VSIZE	0xD0	RW	Bit[7:0]: vsize_sub Sub-window height[7:0]
0x5908	WEIGHT001	0x11	RW	Bit[7:4]: Weight01 Weight of zone01 Bit[3:0]: Weight00 Weight of zone00
0x5909	WEIGHT023	0x11	RW	Bit[7:4]: Weight03 Weight of zone03 Bit[3:0]: Weight02 Weight of zone02
0x590A	WEIGHT101	0x11	RW	Bit[7:4]: Weight11 Weight of zone11 Bit[3:0]: Weight10 Weight of zone10
0x590B	WEIGHT123	0x11	RW	Bit[7:4]: Weight13 Weight of zone13 Bit[3:0]: Weight12 Weight of zone12

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

address	register name	default value	R/W	description
0x590C	WEIGHT201	0x11	RW	Bit[7:4]: Weight21 Weight of zone21 Bit[3:0]: Weight20 Weight of zone20
0x590D	WEIGHT223	0x11	RW	Bit[7:4]: Weight23 Weight of zone23 Bit[3:0]: Weight22 Weight of zone22
0x590E	WEIGHT301	0x11	RW	Bit[7:4]: Weight31 Weight of zone31 Bit[3:0]: Weight30 Weight of zone30
0x590F	WEIGHT323	0x11	RW	Bit[7:4]: Weight33 Weight of zone33 Bit[3:0]: Weight32 Weight of zone32
0x5910	AVG CTRL	0x02	RW	Bit[1]: sum_opt 0: Sum = (4 × B + 9 × G × 2 + 10 × R)/8 1: Sum = B + G × 2 + R Bit[0]: sub_win_en Sub-window function enable signal 0: Disable 1: Enable

5.6 window crop (WINC)

The main functions of WINC include:

- adjust output image based on settings
- supply 16-window function

table 5-6 WINC control registers (sheet 1 of 4)

address	register name	default value	R/W	description
0x5A00	XSTART MAN	0x00	RW	Bit[4:0]: Xstart[12:8] 5 MSB of the manual horizontal start for manual output window
0x5A01	XSTART MAN	0x00	RW	Bit[7:0]: Xstart[7:0] 8 LSB of the manual horizontal start for manual output window
0x5A02	YSTART MAN	0x00	RW	Bit[3:0]: Ystart[11:8] 4 MSB of the manual vertical start for manual output window
0x5A03	YSTART MAN	0x00	RW	Bit[7:0]: Ystart[7:0] 8 LSB of the manual vertical start for manual output window
0x5A04	XWIN MAN	0x05	RW	Bit[4:0]: x_win[12:8] 5 MSB of the manual horizontal size for manual output window
0x5A05	XWIN MAN	0x00	RW	Bit[7:0]: x_win[7:0] 8 LSB of the manual horizontal size for manual output window
0x5A06	YWIN MAN	0x03	RW	Bit[3:0]: y_win[11:8] 4 MSB of the manual vertical size for manual output window
0x5A07	YWIN MAN	0x20	RW	Bit[7:0]: y_win[7:0] 8 LSB of the manual vertical size for manual output window

table 5-6 WINC control registers (sheet 2 of 4)

address	register name	default value	R/W	description
0x5A08	WINC CTRL	0x00	RW	Bit[4]: win16_en 16-window function enable signal 0: Disable 1: Enable Bit[3]: sub_line_out 16-window sub-line enable signal 0: Disable 1: Enable Bit[0]: win_man_en Manual window enable signal 0: Output window uses the system parameter 1: Output window uses manual parameters set by registers 0x5A00~0x5A07
0x5A10	WIN16 XSTART0	0x00	RW	Bit[4:0]: win16_xstart0[12:8] 5 MSB of the first horizontal start for 16-window function
0x5A11	WIN16 XSTART0	0x00	RW	Bit[7:0]: win16_xstart0[7:0] 8 LSB of the first horizontal start for 16-window function
0x5A12	WIN16 XSTART1	0x00	RW	Bit[4:0]: win16_xstart1[12:8] 5 MSB of the second horizontal start for 16-window function
0x5A13	WIN16 XSTART1	0x00	RW	Bit[7:0]: win16_xstart1[7:0] 8 LSB of the second horizontal start for 16-window function
0x5A14	WIN16 XSTART2	0x00	RW	Bit[4:0]: win16_xstart2[12:8] 5 MSB of the third horizontal start for 16-window function
0x5A15	WIN16 XSTART2	0x00	RW	Bit[7:0]: win16_xstart2[7:0] 8 LSB of the third horizontal start for 16-window function
0x5A16	WIN16 XSTART3	0x00	RW	Bit[4:0]: win16_xstart3[12:8] 5 MSB of the 4th horizontal start for 16-window function
0x5A17	WIN16 XSTART3	0x00	RW	Bit[7:0]: win16_xstart3[7:0] 8 LSB of the fourth horizontal start for 16-window function
0x5A18	WIN16 YSTART0	0x00	RW	Bit[3:0]: win16_ystart0[11:8] 4 MSB of the first vertical start for 16-window function

table 5-6 WINC control registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x5A19	WIN16 YSTART0	0x00	RW	Bit[7:0]: win16_ystart0[7:0] 8 LSB of the first vertical start for 16-window function
0x5A1A	WIN16 YSTART1	0x00	RW	Bit[3:0]: win16_ystart1[11:8] 4 MSB of the second vertical start for 16-window function
0x5A1B	WIN16 YSTART1	0x00	RW	Bit[7:0]: win16_ystart1[7:0] 8 LSB of the second vertical start for 16-window function
0x5A1C	WIN16 YSTART2	0x00	RW	Bit[3:0]: win16_ystart2[11:8] 4 MSB of the third vertical start for 16-window function
0x5A1D	WIN16 YSTART2	0x00	RW	Bit[7:0]: win16_ystart2[7:0] 8 LSB of the third vertical start for 16-window function
0x5A1E	WIN16 YSTART3	0x00	RW	Bit[3:0]: win16_ystart3[11:8] 4 MSB of the fourth vertical start for 16-window function
0x5A1F	WIN16 YSTART3	0x00	RW	Bit[7:0]: win16_ystart3[7:0] 8 LSB of the fourth vertical start for 16-window function
0x5A20	WIN16 XWIN0	0x00	RW	Bit[4:0]: win16_xsize0[12:8] 5 MSB of the first horizontal size for 16-window function
0x5A21	WIN16 XWIN0	0x00	RW	Bit[7:0]: win16_xsize0[7:0] 8 LSB of the first horizontal size for 16-window function
0x5A22	WIN16 XWIN1	0x00	RW	Bit[4:0]: win16_xsize1[12:8] 5 MSB of the second horizontal size for 16-window function
0x5A23	WIN16 XWIN1	0x00	RW	Bit[7:0]: win16_xsize1[7:0] 8 LSB of the second horizontal size for 16-window function
0x5A24	WIN16 XWIN2	0x00	RW	Bit[4:0]: win16_xsize2[12:8] 5 MSB of the third horizontal size for 16-window function
0x5A25	WIN16 XWIN2	0x00	RW	Bit[7:0]: win16_xsize2[7:0] 8 LSB of the third horizontal size for 16-window function

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

address	register name	default value	R/W	description
0x5A26	WIN16 XWIN3	0x00	RW	Bit[4:0]: win16_xsize3[12:8] 5 MSB of the fourth horizontal size for 16-window function
0x5A27	WIN16 XWIN3	0x00	RW	Bit[7:0]: win16_xsize3[7:0] 8 LSB of the fourth horizontal size for 16-window function
0x5A28	WIN16 YWIN0	0x00	RW	Bit[3:0]: win16_ysize0[11:8] 4 MSB of the first vertical size for 16-window function
0x5A29	WIN16 YWIN0	0x00	RW	Bit[7:0]: win16_ysize0[7:0] 8 LSB of the first vertical size for 16-window function
0x5A2A	WIN16 YWIN1	0x00	RW	Bit[3:0]: win16_ysize1[11:8] 4 MSB of the second vertical size for 16-window function
0x5A2B	WIN16 YWIN1	0x00	RW	Bit[7:0]: win16_ysize1[7:0] 8 LSB of the second vertical size for 16-window function
0x5A2C	WIN16 YWIN2	0x00	RW	Bit[3:0]: win16_ysize2[11:8] 4 MSB of the third vertical size for 16-window function
0x5A2D	WIN16 YWIN2	0x00	RW	Bit[7:0]: win16_ysize2[7:0] 8 LSB of the third vertical size for 16-window function
0x5A2E	WIN16 YWIN3	0x00	RW	Bit[3:0]: win16_ysize3[11:8] 4 MSB of the fourth vertical size for 16-window function
0x5A2F	WIN16 YWIN3	0x00	RW	Bit[7:0]: win16_ysize3[7:0] 8 LSB of the fourth vertical size for 16-window function

6 system control

System control registers include clock, reset control, and PLL configuration. Individual modules can be reset or clock gated by setting the appropriate registers. For system control registers, see **table 7-2**.

6.1 mobile industry processor interface (MIPI)

MIPI provides a single uni-directional clock lane and two uni-directional data lanes to communicate to components in a mobile device. The two data lanes have full support for high speed (HS) uni-directional data transmission. Contact your local OmniVision FAE for more details.

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

address	register name	default value	R/W	description
0x4800	MIPI CTRL 00	0x04	RW	Bit[5]: gate_sc_en 0: Clock lane is free running 1: Gate clock lane when 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
0x4801	MIPI CTRL 01	0x00	RW	Bit[6]: spkt_dt_sel 1: Use dt_spkt as short packet data Bit[5]: first_bit Change clk_lane first bit 0: Output 0x05 1: Output 0xAA Bit[1]: LPX_select for pclk domain 0: Auto calculate t_lpx_p, unit pclk2x cycle 1: Use lpx_p_min[7:0]

table 6-1 MIPI top control registers (sheet 2 of 8)

address	register name	default value	R/W	description
0x4802	MIPI CTRL 02	0x00	RW	MIPI Control 02 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	NOT USED	–	–	Not Used
0x4804	MIPI CTRL 04	0x04	RW	Bit[7:4]: man_lane_num Bit[3]: lane_num_manual_enable Bit[2]: lane4_6b_en 1: Supports 4-, 7-, 8-lane 6-bit
0x4805	MIPI CTRL 05	0x00	RW	Bit[3]: lpda_retim_manu_o Bit[2]: lpda_retim_sel_o 1: Manual Bit[1]: lpck_retim_manu_o Bit[0]: lpck_retim_sel_o 1: Manual
0x4806	NOT USED	–	–	Not Used
0x4807	MIPI CTRL07	0x03	RW	Bit[3:0]: sw_t_lpx ul_tx T_lpx

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

address	register name	default value	R/W	description
0x4808	MIPI CTRL08	0x0A	RW	Bit[7:0]: wkup_dly Mark1 wakeup delay/2^10
0x4810	FCNT MAX	0xFF	RW	Bit[7:0]: fcnt_max[15:8] High byte of maximum frame counter of frame sync short packet
0x4811	FCNT MAX	0xFF	RW	Bit[7:0]: fcnt_max[7:0] Low byte of maximum frame counter of frame sync short packet
0x4813	MIPI CTRL13	0x00	RW	Bit[2]: vc_sel Input vc or reg vc Bit[1:0]: VC Virtual channel of MIPI
0x4814	MIPI CTRL14	0x2A	RW	Bit[6]: lpkt_dt_sel 0: Use mipi_d 1: Use dt_man_o as long packet data Bit[5:0]: dt_man Manual data type
0x4815	MIPI CTRL15	0x00	RW	Bit[6]: pclk_inv 0: Use falling edge of mipi_pclk_o to generate MIPI bus to PHY 1: Use rising edge of mipi_pclk_o to generate MIPI bus to PHY Bit[5:0]: manu_dt_short Manual data type for short packet
0x4816	EMB DT	0x52	RW	Bit[5:0]: emb_dt Manually set embedded data type
0x4817	YUV420 FUN	0x00	RW	Bit[1]: yuv420_2x YUV420 2x in odd line, lcnt[0]=1 Bit[0]: yuv420_en
0x4818	HS ZERO MIN	0x00	RW	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[1:0]: hs_trail_min[9:8] High byte of minimum value of hs_trail, unit ns

table 6-1 MIPI top control registers (sheet 4 of 8)

address	register name	default value	R/W	description
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[1:0]: clk_zero_min[9:8] High byte of minimum value of clk_zero, unit ns
0x481D	CLK ZERO MIN	0x06	RW	Bit[7:0]: clk_zero_min[7:0] Low byte of minimum value of clk_zero clk_zero_real = clk_zero_min_o + Tui*ui_clk_zero_min_o
0x481E	CLK PREPARE MAX	0x5F	RW	Bit[7:0]: clk_prepare_max[7:0] Maximum value of clk_prepare, unit ns
0x481F	CLK PREPARE MIN	0x26	RW	Bit[7:0]: clk_prepare_min[7:0] Minimum value of clk_prepare clk_prepare_real = clk_prepare_min_o + Tui*ui_clk_prepare_min_o
0x4820	CLK POST MIN	0x00	RW	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[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[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	0x28	RW	Bit[7:0]: hs_prepare_min[7:0] Minimum value of hs_prepare, unit ns

table 6-1 MIPI top control registers (sheet 5 of 8)

address	register name	default value	R/W	description
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[1:0]: hs_exit_min[9:8] High byte of minimum value of hs_exit, unit ns
0x4829	HS EXIT MIN	0x64	RW	Bit[7:0]: hs_exit_min[7:0] Low byte of minimum value of hs_exit hs_exit_real = hs_exit_min_o + Tui*ui_hs_exit_min_o
0x482A	UI HS ZERO MIN	0x06	RW	Bit[5:0]: ui_hs_zero_min[5:0] Minimum UI value of hs_zero, unit UI
0x482B	UI HS TRAIL MIN	0x04	RW	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[5:0]: ui_clk_zero_min[5:0] Minimum UI value of clk_zero, unit UI
0x482D	UI CLK PREPARE MIN	0x00	RW	Bit[7:4]: ui_clk_prepare_max Bit[3:0]: ui_clk_prepare_min Value of clk_prepare, unit UI
0x482E	UI CLK POST MIN	0x34	RW	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[5:0]: ui_clk_trail_min[5:0] Minimum UI value of clk_trail, unit UI
0x4830	UI LPX P MIN	0x00	RW	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 Bit[3:0]: ui_hs_prepare_min UI value of hs_prepare, unit UI
0x4832	UI HS EXIT MIN	0x00	RW	Bit[5:0]: ui_hs_exit_min[5:0] Minimum UI value of hs_exit, unit UI
0x4836	NOT USED	–	–	Not Used
0x4837	PCLK PERIOD	0x10	RW	Bit[7:0]: pclk_period[7:0] Period of pclk2x, pclk_div=1, and 1-bit decimal

table 6-1 MIPI top control registers (sheet 6 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
0x4848	MIPI LP GPIO1	0x00	RW	Bit[7]: lp_sel2 0: Auto generate mipi_lp_dir2_o 1: Use lp_dir_man2 to be mipi_lp_dir2_o Bit[6]: lp_dir_man2 0: Input 1: Output Bit[5]: lp_p2_o Bit[4]: lp_n2_o Bit[3]: lp_sel3 0: Auto generate mipi_lp_dir3_o 1: Use lp_dir_man3 to be mipi_lp_dir3_o Bit[2]: lp_dir_man3 0: Input 1: Output Bit[1]: lp_p3_o Bit[0]: lp_n3_o

table 6-1 MIPI top control registers (sheet 7 of 8)

address	register name	default value	R/W	description
0x483A	MIPI LP GPIO2	0x00	RW	Bit[7]: lp_sel4 0: Auto generate mipi_lp_dir4_o 1: Use lp_dir_man4 to be mipi_lp_dir4_o Bit[6]: lp_dir_man4 0: Input 1: Output Bit[5]: lp_p4_o Bit[4]: lp_n4_o Bit[3]: lp_sel5 0: Auto generate mipi_lp_dir5_o 1: Use lp_dir_man5 to be mipi_lp_dir5_o Bit[2]: lp_dir_man5 0: Input 1: Output Bit[1]: lp_p5_o Bit[0]: lp_n5_o
0x483B	MIPI LP GPIO3	0x00	RW	Bit[7]: lp_sel6 0: Auto generate mipi_lp_dir6_o 1: Use lp_dir_man6 to be mipi_lp_dir6_o Bit[6]: lp_dir_man6 0: Input 1: Output Bit[5]: lp_p6_o Bit[4]: lp_n6_o Bit[3]: lp_sel7 0: Auto generate mipi_lp_dir7_o 1: Use lp_dir_man7 to be mipi_lp_dir7_o Bit[2]: lp_dir_man7 0: Input 1: Output Bit[1]: lp_p7_o Bit[0]: lp_n7_o
0x483C	MIPI CTRL3C	0x02	RW	Bit[3:0]: t_clk_pre Unit: pclk2x cycle

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

address	register name	default value	R/W	description
0x483D	MIPI LP GPIO4	0x00	RW	Bit[7]: lp_ck_sel0 0: Auto generate mipi_ck_lp_dir0_o 1: Use lp_ck_dir_man0 to be mipi_ck_lp_dir0_o Bit[6]: lp_ck_dir_man0 0: Input 1: Output Bit[5]: lp_ck_p0_o Bit[4]: lp_ck_n0_o Bit[3]: lp_ck_sel1 0: Auto generate mipi_ck_lp_dir1_o 1: Use lp_ck_dir_man1 to be mipi_ck_lp_dir1_o Bit[2]: lp_ck_dir_man1 0: Input 1: Output Bit[1]: lp_ck_p1_o Bit[0]: lp_ck_n1_o
0x484A	SEL MIPI CTRL4A	0x27	RW	Bit[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	NOT USED	–	–	Not Used
0x484C	SEL MIPI CTRL4C	0x03	RW	Bit[6]: Not used Bit[5]: prbs_enable Bit[4]: hs_test_only MIPI high speed only test mode enable Bit[3]: Not used
0x484D	TEST PATTEN DATA	0xB6	RW	Bit[7:0]: test_patten_data[7:0] Data lane test pattern register
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[7:0]: test_patten_ck_data Clock test pattern register

6.2 low voltage differential signaling (LVDS)

LVDS is a high speed differential serial interface. It provides two data outputs supporting all formats.

table 6-2 LVDS registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x4A00	LVDS R0	0x2A	RW	Bit[7]: Two sync code enable in lane8 mode Bit[6]: Sync code manual mode enable Bit[5]: Sync code enable when only 1 lane Bit[4]: lvds_pclk_inv Bit[3]: Channel ID enable in sync per lane mode Bit[2]: CCIR parameter F Bit[1]: Sav first enable Bit[0]: Sync code mode 0: Split 1: Per lane
0x4A02	LVDS DUMMY DATA0	0x00	RW	Bit[3:0]: lvds_dummy_data0[11:8] Dummy data0 high byte
0x4A03	LVDS DUMMY DATA0	0x80	RW	Bit[7:0]: lvds_dummy_data0[7:0] Dummy data0 low byte
0x4A04	LVDS DUMMY DATA1	0x00	RW	Bit[3:0]: lvds_dummy_data1[11:8] Dummy data1 high byte
0x4A05	LVDS DUMMY DATA1	0x10	RW	Bit[7:0]: lvds_dummy_data1[7:0] Dummy data1 low byte
0x4A06	LVDS R6	0xAA	RW	Bit[7:0]: lvds_r6 Sync code of frame_start in manual sync code mode
0x4A07	LVDS R7	0x55	RW	Bit[7:0]: lvds_r7 Sync code of frame_end in manual sync code mode
0x4A08	LVDS R8	0x99	RW	Bit[7:0]: lvds_r8 Sync code of line_start in manual sync code mode
0x4A09	LVDS R9	0x66	RW	Bit[7:0]: lvds_r9 Sync code of line_end in manual sync code mode
0x4A0A	LVDS RA	0x00	RW	Bit[2]: r_hts_man_en Bit[1]: r_ln2_sel Bit[0]: r_chk_pcnt

table 6-2 LVDS registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x4A0C	LVDS BLK TIMES	0x00	RW	Bit[3:0]: lvds_blk_times[11:8] High byte of r_blk_times
0x4A0D	LVDS BLK TIMES	0x05	RW	Bit[7:0]: lvds_blk_times[7:0] Low byte of r_blk_times
0x4A0E	LVDS HTS MAN	0x00	RW	Bit[7:0]: lvds_hts_man[15:8] High byte of hts_man
0x4A0F	LVDS HTS MAN	0x00	RW	Bit[7:0]: lvds_hts_man[7:0] Low byte of hts_man

7 register tables

The following tables provide descriptions of the device control registers contained in the OV9713. For all register enable/disable bits, ENABLE = 1 and DISABLE = 0. The device slave addresses are 0x20 for write and 0x21 for read (when SID=1, 0x6C for write and 0x6D for read).

7.1 general configuration and image size [0x0000 - 0x0387]

table 7-1 general configuration and image size registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x0100	MODE_SELECT	0x00	RW	Bit[7:1]: Not used Bit[0]: Mode select 0: software_standby 1: Streaming
0x0101	IMAGE ORIENTATION	0x00	RW	Bit[7:2]: Not used Bit[1]: v_flip Bit[0]: h_mirror
0x0103	SOFTWARE RESET	0x00	RW	Bit[7:1]: Not used Bit[0]: software_reset
0x0106	FAST STANDBY CTRL	0x00	RW	Bit[7:1]: Not used Bit[0]: fast_standby 0: Frame completes before mode entry 1: Frame may be truncated before mode entry
0x0107	CCI ADDRESS CONTROL 20	0x20	RW	Bit[7:0]: Debug mode
0x0108	CCI ADDRESS CONTROL 6C	0x6C	RW	Bit[7:0]: Debug mode
0x0340	FRAME LENGTH LINES	0x02	RW	Bit[7:0]: Total vertical size[15:8]
0x0341	FRAME LENGTH LINES	0xE8	RW	Bit[7:0]: Total vertical size[7:0]
0x0342	LINE LENGTH PCK	0x06	RW	Bit[7:0]: Total horizontal size[15:8]
0x0343	LINE LENGTH PCK	0x60	RW	Bit[7:0]: Total horizontal size[7:0]
0x0344	X ADDR START	0x00	RW	Bit[7:4]: Debug mode Bit[3:0]: x_addr_start[11:8]
0x0345	X ADDR START	0x08	RW	Bit[7:0]: x_addr_start[7:0]
0x0346	Y ADDR START	0x00	RW	Bit[7:4]: Debug mode Bit[3:0]: y_addr_start[11:8]

table 7-1 general configuration and image size registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x0347	Y ADDR START	0x04	RW	Bit[7:0]: y_addr_start[7:0]
0x0348	X ADDR END	0x05	RW	Bit[7:4]: Debug mode Bit[3:0]: x_addr_end[11:8]
0x0349	X ADDR END	0x17	RW	Bit[7:0]: x_addr_end_int horizontal width[7:0]
0x034A	Y ADDR END	0x02	RW	Bit[7:4]: Debug mode Bit[3:0]: y_addr_end_int vertical height[11:8]
0x034B	Y ADDR END	0xD7	RW	Bit[7:0]: y_addr_end_int vertical height[7:0]
0x034C	X OUTPUT SIZE	0x05	RW	Bit[7:4]: Debug mode Bit[3:0]: Image output horizontal width[11:8]
0x034D	X OUTPUT SIZE	0x00	RW	Bit[7:0]: Image output horizontal width[7:0]
0x034E	Y OUTPUT SIZE	0x02	RW	Bit[7:4]: Debug mode Bit[3:0]: Image output vertical height[11:8]
0x034F	Y OUTPUT SIZE	0xD0	RW	Bit[7:0]: Image output vertical height[7:0]
0x0381	X EVEN INC	0x01	RW	Bit[7:4]: Debug mode Bit[3:0]: h_even_inc Horizontal subsample even increase number
0x0383	X ODD INC	0x01	RW	Bit[7:4]: Debug mode Bit[3:0]: h_odd_inc Horizontal subsample odd increase number
0x0385	Y EVEN INC	0x01	RW	Bit[7:4]: Debug mode Bit[3:0]: v_even_inc Vertical subsample even increase number
0x0387	Y ODD INC	0x01	RW	Bit[7:4]: Debug mode Bit[3:0]: v_odd_inc Vertical subsample odd increase number

7.2 system control [0x3001 - 0x30B8]

table 7-2 system control registers (sheet 1 of 6)

address	register name	default value	R/W	description
0x3001	PAD CTRL	0x03	RW	Bit[7:5]: Not used Bit[4]: strobe_in NOR opt Bit[3:2]: Not used Bit[1:0]: I/O strength
0x3002	PAD OEN2	0x00	RW	Bit[7:6]: Not used Bit[5]: io_sid_oen Bit[4]: Not used Bit[3]: io_strobe_oen Bit[2]: io_sda_oen Bit[1]: Not used Bit[0]: io_gpio_oen
0x3004	PAD OEN3	0x00	RW	Bit[7]: io_fsin_oen Bit[6:0]: Not used
0x3006	PAD OUT2	0x00	RW	Bit[7]: io_fsin_o Bit[6:5]: Not used Bit[4]: io_sid_o Bit[3]: Not used Bit[2]: io_strobe_o Bit[1]: io_sda_o Bit[0]: Not used
0x3008	PAD OUT1	0x00	RW	Bit[7:5]: Not used Bit[4]: io_gpio_o Bit[3:0]: Not used
0x3009	PAD OUT0	0x00	RW	Bit[7:0]: pad_out0[7:0]
0x300A	CHIP ID	0x97	R	Bit[7:0]: Chip ID high byte
0x300B	CHIP ID	0x13	R	Bit[7:0]: Chip ID low byte
0x3010	SC REG10	0x81	RW	Bit[7]: scale_div_man_en Bit[6:3]: Not used Bit[2:0]: pll_scale_div
0x3011	SC REG11	0x5A	RW	Bit[7:1]: Reserved Bit[0]: mipi_ck_lane_dis_opt Disable MIPI clock lane

table 7-2 system control registers (sheet 2 of 6)

address	register name	default value	R/W	description
0x3012	A MIPI PK	0x00	RW	Bit[7:6]: pgm_hspredrv_iset Bias current control for LVDS Bit[5:4]: Lph Bit[3]: phy_mode 0: MIPI 1: LVDS Bit[2]: predrv_sw Bit[1:0]: Ictl Bias current control for MIPI
0x3013	A MIPI PK	0x12	RW	Bit[7:6]: pgm_vcm MIPI high speed common-mode voltage control Bit[5:4]: pgm_lptx Register for driving strength of low speed transmitter Bit[3]: bp_d_hs_en_lat Bypass high speed data lane enable latch Bit[2]: bp_c_hs_en_lat Bypass high speed clock lane enable latch Bit[1]: mipi_pad MIPI mode pad switch input (default is MIPI mode) Bit[0]: Not used
0x3014	MIPI SC O	0x04	RW	Bit[7:6]: mipi_ck_skew_o Clock lane skew adjustment Bit[5]: lane_dis2_op Second data lane disable Bit[4]: mipi_pd Power down mipi_phy except low speed receiver Bit[3]: Not used Bit[2]: mipi_en MIPI mode enable Bit[1]: Not used Bit[0]: First data lane disable
0x3015	MIPI SC O	–	–	Bit[7:0]: Reserved
0x3016~ 0x301E	CLK RST REG	–	–	Debug Mode

table 7-2 system control registers (sheet 3 of 6)

address	register name	default value	R/W	description
0x301F	MISC CTRL	0x03	RW	Bit[7:6]: mipi_d1_skew_o First data lane skew adjustment Bit[5]: mipi_clk_lane_ctrl 0: Clock lane hold lp00 when pd_mipi 1: Clock lane is high-z when pd_mipi 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]: Not used Bit[1]: lane_num 0: 1-lane 1: 2-lane Bit[0]: cen_global_o
0x3020	MISC CTRL1	0x00	RW	Bit[7:2]: Not used Bit[1:0]: mipi_d2_skew_o Second data lane skew adjustment
0x3022	RSVD	–	–	Reserved
0x3023	LOW PWR CTR	0x07	RW	Bit[7]: Not used Bit[6]: phy_pd_mipi_pwdn_dis mask Bit[5]: phy_pd_lprx_pwdn_dis mask Bit[4]: stb_rst_dis mask 0: Reset all block at software standby mode 1: TC, sensor_control, ISP are reset, others not Bit[3]: pd_ana_dis mask Bit[2]: pd_big_regulator_dis mask Bit[1]: phy_pd_mipi_slppd_dis mask Bit[0]: phy_pd_lprx_slppd_dis mask
0x3024	PAD SEL2	0x00	RW	Bit[7]: io_fsin_sel Bit[6]: io_sid_sel Bit[5]: Debug mode Bit[4:3]: Not used Bit[2]: io_strobe_sel Bit[1]: io_sda_sel Bit[0]: Not used
0x3025	SC REG25	0x00	RW	Bit[7:0]: Not used

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

address	register name	default value	R/W	description
0x3026	PAD SEL1	0x00	RW	Bit[7:5]: Reserved Bit[4]: io_gpio_sel[0] Bit[3:0]: Reserved
0x3027	IO_Y_SEL_L	0x00	RW	Bit[7:0]: io_y_sel[7:0]
0x302B	SCCB ID2	0x90	RW	Bit[7:0]: sccb_id2
0x302C	AUTO SLEEP PERIOD	0x01	RW	Bit[7:0]: auto_sleep_period[31:24]
0x302D	AUTO SLEEP PERIOD	0x00	RW	Bit[7:0]: auto_sleep_period[23:16]
0x302E	AUTO SLEEP PERIOD	0x00	RW	Bit[7:0]: auto_sleep_period[15:8]
0x302F	AUTO SLEEP PERIOD	0x00	RW	Bit[7:0]: auto_sleep_period[7:0]
0x3030	SC LP CTRL0	0x03	RW	Bit[7]: auto_sleep_en Bit[6]: gpio_sel 0: Debug mode 1: Sleep can be read by GPIO (Y9) Bit[5:4]: Debug mode Bit[3:0]: frame_on_num
0x3031	PAD OEN SLEEP 2	0xFF	RW	Bit[7:4]: Reserved Bit[3]: io_strobe_oen_sleep Bit[2]: io_sda_oen_sleep Bit[1]: Not used Bit[0]: io_gpio_oen_sleep
0x3032	PAD OEN SLEEP 1	0xFF	RW	Bit[7]: io_fsin_oen_sleep Bit[6:0]: Reserved
0x3033	RSVD	–	–	Reserved
0x3034	PAD OEN PWDN 2	0xFF	RW	Bit[7:4]: Not used Bit[3]: io_strobe_oen_pwdn Bit[2]: io_sda_oen_pwdn Bit[1]: Not used Bit[0]: io_gpio_oen_pwdn
0x3035	PAD OEN PWDN 1	0xFF	RW	Bit[7]: io_fsin_oen_pwdn Bit[6:0]: Reserved
0x3036	RSVD	–	–	Reserved

table 7-2 system control registers (sheet 5 of 6)

address	register name	default value	R/W	description
0x3037	CLKRST0	0xF0	RW	Bit[7]: Not used Bit[6]: sclk_hdr_frm Bit[5]: sclk_fmt Bit[4]: sclk_tpm Bit[3]: Not used Bit[2]: rst_hdr_frm Bit[1]: rst_fmt Bit[0]: rst_tpm
0x3038	FREX RST MASK0	0x00	RW	Bit[7:3]: Not used Bit[2]: frex_mask_hdr_frm Bit[1]: frex_mask_fmt Bit[0]: frex_mask_tpm
0x303B	SC REG3B	0x01	RW	Bit[7:5]: Debug mode Bit[4]: sccb_pgm_id_en Bit[3:1]: Debug mode Bit[0]: sccb_id2_nack_en
0x303F	CLKRST	0xD2	RW	Bit[7:0]: Debug mode
0x3081~ 0x3084	PLL REG	–	RW	Bit[7:0]: Debug mode
0x3090	PLL10	0x01	RW	Bit[7:3]: Not used Bit[2:0]: pll2_pre_div_o[2:0]
0x3091	PLL11	0x05	RW	Bit[7:6]: Not used Bit[5:0]: pll2_divp_o[5:0]
0x3092	PLL12	0x03	RW	Bit[7:5]: Not used Bit[4:0]: pll2_divs_o[4:0]
0x3093	PLL13	0x01	RW	Bit[7:2]: Not used Bit[1:0]: pll2_seld5_o[1:0] 0x: /1 10: /2 11: /2.5
0x3094	PLL14	0x02	RW	Bit[7:2]: Not used Bit[1:0]: pll2_divr
0x3095~ 0x3099	PLL REG	–	RW	Bit[7:0]: Debug mode
0x309A	PLL1A	0x01	RW	Bit[7:5]: Not used Bit[4:0]: pll3_divs_o[4:0]
0x309B	PLL1B	0x01	RW	Bit[7:2]: Not used Bit[1:0]: pll3_seld5_o[1:0]
0x309C	PLL1C	0x01	RW	Bit[7:2]: Not used Bit[1:0]: pll3_divr[1:0]

table 7-2 system control registers (sheet 6 of 6)

address	register name	default value	R/W	description
0x309D~ 0x30A2	PLL REG	–	RW	Bit[7:0]: Debug mode
0x30A3	PLL23	0x19	RW	Bit[7:6]: Not used Bit[5:0]: pll3_divp_o[5:0]
0x30A4	PLL REG	–	RW	Bit[7:0]: Debug mode
0x30A5	PLL25	0x03	RW	Bit[7:3]: Not used Bit[2:0]: pll3_pre_div_opt[2:0]
0x30B2	PLL MULTIPLIER	0x00	RW	Bit[7:1]: Reserved Bit[0]: pll_multiplier[8]
0x30B3	PLL MULTIPLIER	0x28	RW	Bit[7:0]: pll_multiplier[7:0]
0x30B5	PLL1 OP PIX CLK DIV	0x04	RW	Bit[7:3]: Reserved Bit[2:0]: pll1_op_pix_clk_div_r[2:0] 0x03: 16 0x04: 18 0x05: 110 0x10: 112
0x30B6	PLL1 OP SYS CLK DIV	0x02	RW	Bit[7:4]: Reserved Bit[3:0]: pll1_op_sys_clk_div[3:0]
0x30B7	PLL1 OP 2LANE CLK DIV	0x02	RW	Bit[7:2]: Reserved Bit[1:0]: pll1_op_2lane_clk_div[1:0]
0x30B8	PLL1 R PREDIV	0x01	RW	Bit[7:3]: Reserved Bit[2:0]: pll1_r_prediv[2:0]

7.3 group hold control [0x3200 - 0x320C]

table 7-3 group hold registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x3200	GROUP ADR0	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Start address for group 0 (every bit indicates 16 bytes)
0x3201	GROUP ADR1	0x04	RW	Bit[7:4]: Reserved Bit[3:0]: Start address for group 1 (every bit indicates 16 bytes)
0x3202	GROUP ADR2	0x08	RW	Bit[7:4]: Reserved Bit[3:0]: Start address for group 2 (every bit indicates 16 bytes)

table 7-3 group hold registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x3203	GROUP ADR3	0x0B	RW	Bit[7:4]: Reserved Bit[3:0]: Start address for group 3 (every bit indicates 16 bytes)
0x3204~ 0x3207	GROUP LEN	–	R	Debug Mode
0x3208	GROUP ACCESS	–	W	Bit[7:4]: group_ctrl 0000: Group hold start 0001: Group hold end 1010: Group launch Others: Reserved 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: Reserved
0x3209	GRP0 PERIOD	0x00	RW	Bit[7:0]: Frames for staying in grp0
0x320A	GRP1 PERIOD	0x00	RW	Bit[7:0]: Frames for staying in grp1
0x320B	GRP_SW_CTRL	0x01	RW	Bit[7]: auto_sw Bit[6:4]: Reserved Bit[3]: group_switch_repeat Bit[2]: context_en Bit[1:0]: Second group selection
0x320C	SRAM TEST	0x0A	RW	Bit[7:5]: Not used Bit[4:0]: Debug mode

7.4 manual AEC/AGC [0x3500 - 0x3A07, 0x5900 - 0x5910]

table 7-4 manual AEC/AGC registers (sheet 1 of 5)

address	register name	default value	R/W	description
0x3500	EXPO0	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Expo0[19:16] 4 MSB of exposure which is the exposure in normal mode, the long exposure in HDR mode or the exposure of first frame in frame exposure mode
0x3501	EXPO0	0x02	RW	Bit[7:0]: Expo0[15:8] Exposure in normal mode, the long exposure in HDR mode or the exposure of first frame in frame exposure mode
0x3502	EXPO0	0x00	RW	Bit[7:0]: Expo0[7:0] 8 LSB of exposure which is the exposure in normal mode, the long exposure in HDR mode or the exposer of first frame in frame exposure mode. Bit[3:0] is for fractional exposure, which is smaller than 0x0B.
0x3503	AECPK CTRL	0x03	RW	Bit[7]: dig_gain_man_en Manual digital gain enable signal 0: Disable 1: Enable Bit[6]: Not used Bit[5:4]: dig_gain_man Manual digital gain Bit[3]: agc_no_dly Delay option for AGC comparing with AEC 0: AGC is later 1 frame than AEC 1: No delay Bit[2]: man_gain_as_snr_gain Manual gain format option 0: Manual gains are looked as real gain 1: Manual gains are looked as sensor gain Bit[1]: agc_manual Manual AGC enable signal 0: Disable 1: Enable Bit[0]: aec_manual Manual AEC enable signal 0: Disable 1: Enable

table 7-4 manual AEC/AGC registers (sheet 2 of 5)

address	register name	default value	R/W	description
0x3504	EXPO1	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Expo1[19:16] 4 MSB of exposure which is the short exposure in HDR mode or the exposure of second frame in frame exposure mode
0x3505	EXPO1	0x02	RW	Bit[7:0]: Expo1[15:8] Short exposure in HDR mode or the exposure of second frame in frame exposure mode
0x3506	EXPO1	0x00	RW	Bit[7:0]: Expo1[7:0] 8 LSB of exposure which is the short exposure in HDR mode or the exposure of second frame in frame exposure mode. Bit[3:0] is for fractional exposure
0x3507	EXPO2	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Expo2[19:16] 4 MSB of exposure which is the exposure of third frame in frame exposure mode
0x3508	EXPO2	0x02	RW	Bit[7:0]: Expo2[15:8] Exposure of third frame in frame exposure mode
0x3509	EXPO2	0x00	RW	Bit[7:0]: Expo2[7:0] 8 LSB of exposure which is the exposure of third frame in frame exposure mode. Bit[3:0] is for fractional exposure
0x350A	GAIN0	0x00	RW	Bit[7:2]: Not used Bit[1:0]: Gain0[9:8] 2 MSB of gain which is the gain in normal mode, the long gain in HDR mode or the gain of first frame in frame exposure mode
0x350B	GAIN0	0x10	RW	Bit[7:0]: Gain0[7:0] 8 LSB of gain which is the gain in normal mode, the long gain in HDR mode or the gain of first frame in frame exposure mode. Bit[3:0] is for fractional gain
0x350C	GAIN1	0x00	RW	Bit[7:2]: Not used Bit[1:0]: Gain1[9:8] 2 MSB of gain which is the long gain in HDR mode or the gain of second frame in frame exposure mode

table 7-4 manual AEC/AGC registers (sheet 3 of 5)

address	register name	default value	R/W	description
0x350D	GAIN1	0x10	RW	Bit[7:0]: Gain1[7:0] 8 LSB of gain which is the long gain in HDR mode or the gain of second frame in frame exposure mode. Bit[3:0] is for fractional gain
0x350E	GAIN2	0x00	RW	Bit[7:2]: Not used Bit[1:0]: Gain2[9:8] 2 MSB of gain which is the gain of third frame in frame exposure mode
0x350F	GAIN2	0x10	RW	Bit[7:0]: Gain2[7:0] 8 LSB of gain which is the gain of third frame in frame exposure mode. Bit[3:0] is for fractional gain
0x3A01	AEC CTRL01	0x21	RW	Bit[7]: Debug mode Bit[6:4]: long_short_ratio Bit[3:0]: Minimum exposure
0x3A02	AEC CTRL02	0x78	RW	Bit[7:0]: AEC stable range high limit
0x3A03	AEC CTRL03	0x68	RW	Bit[7:0]: AEC stable range low limit
0x3A04	AEC CTRL04	0x06	RW	Bit[7:0]: Maximum exposure[15:8]
0x3A05	AEC CTRL05	0x14	RW	Bit[7:0]: Maximum exposure[7:0]
0x3A06	AEC CTRL06	0x03	RW	Bit[7:2]: Debug mode Bit[1:0]: Gain ceiling[9:8]
0x3A07	AEC CTRL07	0xE0	RW	Bit[7:0]: Gain ceiling[7:0]
0x5900	SUBWIN XSTART	0x00	RW	Bit[7:5]: Not used Bit[4:0]: Xstart[12:8] AVG sub-window horizontal start position
0x5901	SUBWIN XSTART	0x00	RW	Bit[7:0]: Xstart[7:0] AVG sub-window horizontal start position
0x5902	SUBWIN YSTART	0x00	RW	Bit[7:4]: Not used Bit[3:0]: ystart_sub[11:8] AVG sub-window vertical start position
0x5903	SUBWIN YSTART	0x00	RW	Bit[7:0]: ystart_sub[7:0] AVG sub-window vertical start position
0x5904	SUBWIN HSIZE	0x05	RW	Bit[7:5]: Not used Bit[4:0]: hsize_sub[12:8] Sub-window width
0x5905	SUBWIN HSIZE	0x00	RW	Bit[7:0]: hsize_sub[7:0] Sub-window width

table 7-4 manual AEC/AGC registers (sheet 4 of 5)

address	register name	default value	R/W	description
0x5906	SUBWIN VSIZE	0x02	RW	Bit[7:4]: Not used Bit[3:0]: vsize_sub[11:8] Sub-window height
0x5907	SUBWIN VSIZE	0xD0	RW	Bit[7:0]: vsize_sub Sub-window height[7:0]
0x5908	WEIGHT001	0x11	RW	Bit[7:4]: Weight01 Weight of zone01 Bit[3:0]: Weight00 Weight of zone00
0x5909	WEIGHT023	0x11	RW	Bit[7:4]: Weight03 Weight of zone03 Bit[3:0]: Weight02 Weight of zone02
0x590A	WEIGHT101	0x11	RW	Bit[7:4]: Weight11 Weight of zone11 Bit[3:0]: Weight10 Weight of zone10
0x590B	WEIGHT123	0x11	RW	Bit[7:4]: Weight13 Weight of zone13 Bit[3:0]: Weight12 Weight of zone12
0x590C	WEIGHT201	0x11	RW	Bit[7:4]: Weight21 Weight of zone21 Bit[3:0]: Weight20 Weight of zone20
0x590D	WEIGHT223	0x11	RW	Bit[7:4]: Weight23 Weight of zone23 Bit[3:0]: Weight22 Weight of zone22
0x590E	WEIGHT301	0x11	RW	Bit[7:4]: Weight31 Weight of zone31 Bit[3:0]: Weight30 Weight of zone30
0x590F	WEIGHT323	0x11	RW	Bit[7:4]: Weight33 Weight of zone33 Bit[3:0]: Weight32 Weight of zone32

table 7-4 manual AEC/AGC registers (sheet 5 of 5)

address	register name	default value	R/W	description
0x5910	AVG CTRL	0x02	RW	Bit[7:2]: Not used Bit[1]: sum_opt 0: Sum=(4×B+9×G×2+10×R)/8 1: Sum=B+G×2+R Bit[0]: sub_win_en Sub-window function enable signal 0: Disable 1: Enable

7.5 analog control [0x3600 - 0x3684]

Changing these register values is not recommended.

table 7-5 analog control registers

address	register name	default value	R/W	description
0x3600~ 0x3635	ANALOG REGISTERS	–	RW	Analog Control Registers
0x3636	PWC 6	0x00	RW	Bit[7:4]: Debug mode Bit[3]: bp_regulator 0: Not used 1: Bypass regulator Bit[2:0]: Debug mode
0x3637~ 0x3666	ANALOG REGISTERS	–	RW	Analog Control Registers
0x3667	CORE 7	0x00	RW	Bit[7:5]: Debug mode Bit[4]: pad_gpio_o_opt Bit[3:2]: Debug mode Bit[1:0]: pad_share_strobe_o_opt
0x3668~ 0x3684	ANALOG REGISTERS	–	RW	Analog Control Registers

7.6 sensor control [0x3700 - 0x374B]

Changing these register values is not recommended.

table 7-6 sensor control registers

address	register name	default value	R/W	description
0x3700~ 0x374B	SENSOR CTRL REGISTERS	–	RW	Sensor Control Registers

7.7 timing control [0x3800 - 0x3832]

Changing these register values is not recommended.

table 7-7 timing control registers

address	register name	default value	R/W	description
0x3800~ 0x3832	TIMING CTRL REGISTERS	–	RW	Timing Control Registers
0x3823	REG23	0x00	RW	Bit[7]: Debug mode Bit[6]: ext_vs_re Bit[5]: ext_vs_en Bit[4:0]: Debug mode
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	0x00	RW	Bit[7:0]: r_rst_fsin[7:0] R reset value low byte at vs_ext
0x3828~ 0x3832	TIMING CTRL REGISTERS	–	RW	Timing Control Registers

7.8 strobe control [0x3B00 - 0x3B05]

table 7-8 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]: Reserved Bit[2:0]: 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
0x3B03	STROBE ADD DUMMY	0x00	RW	Bit[7:0]: strobe_add_dummy[7:0] Dummy line number added at strobe
0x3B04	STROBE CTL1	0x00	RW	Bit[7]: strobe_valid (read only) Bit[6:4]: Reserved Bit[3]: start_point_sel Bit[2]: Strobe repeat enable Bit[1:0]: Strobe latency 00: Strobe generated at next frame 01: Delay one frame, strobe generated 2 frames later 10: Delay one frame, strobe generated 3 frames later 11: Delay one frame, 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

7.9 OTP control [0x3D80 - 0x3D87]

table 7-9 OTP control registers

address	register name	default value	R/W	description
0x3D80	OTP PGM CTR	–	RW	Bit[7]: otp_pgenb_o 1: Programing time Bit[6:1]: Not used Bit[0]: Program OTP
0x3D81	OTP LOAD CTR	–	RW	Bit[7]: otp_load_o 1: Loading time Bit[6:1]: Not used Bit[0]: otp_rd Read start signal Write 0 to 1 will start load data
0x3D82	OTP PGM PULSE	0x66	RW	Bit[7:0]: r_pgm_pulse[7:0] Used to control program strobe pulse by 8×Tsclk
0x3D83	OTP LOAD PULSE	0x05	RW	Bit[7:4]: Reserved Bit[3:0]: r_load_pulse[3:0] Control load strobe pulse by Tsclk
0x3D84	OTP MODE CTRL	0x01	RW	Bit[7]: program_dis 0: Enable 1: Disable Bit[6]: mode_select 0: Auto mode 1: Manual mode Bit[5:0]: Memory select Used to select the exact one bank in OTP 0x0: Select memory 1... 0x7: Select memory 8…
0x3D85	OTP START ADDRESS	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: start_address[3:0] Start address for manual mode
0x3D86	OTP END ADDRESS	0x1F	RW	Bit[7:4]: Reserved Bit[3:0]: end_address[3:0] End address for manual mode
0x3D87	OTP PS2CS	0x03	RW	Bit[7:4]: Reserved Bit[3:0]: Ps2cs[3:0] PS to CSB time control by SCLK

7.10 frame control [0x4240 - 0x4243]

table 7-10 frame control registers

address	register name	default value	R/W	description
0x4240	ISP FRAME CTRL0	0x00	RW	Bit[7:0]: Debug mode
0x4241	ISP FRAME ON NUMBER	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: frame_on_number
0x4242	ISP FRAME OFF NUMBER	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: frame_off_number
0x4243	ISP FRAME CTRL1	0x00	RW	Bit[7:0]: Debug mode

7.11 format control [0x4300 - 0x4316]

table 7-11 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] Data max high byte
0x4301	CLIP MIN HI	0x00	RW	Bit[7:0]: clip_min[11:4] Data min high byte
0x4302	CLIP LO	0xF0	RW	Bit[7:4]: clip_max[3:0] Data max low byte Bit[3:0]: clip_min[3:0] Data min low byte
0x4303~ 0x4306	FORMAT CTRL3	–	–	Debug Mode
0x4307	EMBED CTRL	0x31	RW	Bit[7:4]: embed_line_st Bit[3]: embed_start_man Bit[2]: dpc_threshold_opt 0: For white pixel 1: For black pixel Bit[1]: embed_byte_order Bit[0]: embedded_en
0x4308	TST X START HIGH	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: tst_x_start_high[10:7] Test window x start address

table 7-11 format control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x4309	TST X START LOW	0x00	RW	Bit[7:1]: tst_x_start_low[6:0] Test window x start address Bit[0]: Reserved
0x430A	TST Y START HIGH	0x00	RW	Bit[7:5]: Reserved Bit[4:0]: tst_y_start_high[11:7] Test window y start address
0x430B	TST Y START LOW	0x00	RW	Bit[7:1]: tst_y_start_low[6:0] Test window y start address Bit[0]: Not used
0x430C	TST WIDTH HIGH	0x00	RW	Bit[7:3]: Not used Bit[2:0]: tst_width_high[10:8] Test window width
0x430D	TST WIDTH LOW	0x00	RW	Bit[7:0]: tst_width_low[7:0] Test window width
0x430E	TST HIGHT HIGH	0x00	RW	Bit[7:4]: Not used Bit[3:0]: tst_hight_high[11:8] Test window height
0x430F	TST HIGHT LOW	0x00	RW	Bit[7:0]: tst_hight_low[7:0] Test window height
0x4310	RSVD	–	–	Reserved
0x4311	REG11	0x04	RW	Bit[7:0]: hsyvsy_neg_width[15:8] VSYNC width (in term of pixel numbers)
0x4312	REG12	0x00	RW	Bit[7:0]: hsyvsy_neg_width[7:0] VSYNC width (in term of pixel numbers)
0x4313	REG13	0x00	RW	Bit[7:5]: Not used Bit[4]: VSYNC polarity Bit[3:2]: VSYNC output select Bit[1]: VSYNC mode 3 Bit[0]: VSYNC mode 2
0x4314	REG14	0x00	RW	Bit[7:0]: seof_vsync_delay[23:16] VSYNC trigger to VSYNC delay
0x4315	REG15	0x01	RW	Bit[7:0]: seof_vsync_delay[15:8] VSYNC trigger to VSYNC delay
0x4316	REG16	0x00	RW	Bit[7:0]: seof_vsync_delay[7:0] VSYNC trigger to VSYNC delay

7.12 VFIFO control [0x4500 - 0x460F]

table 7-12 VFIFO control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x4500~ 0x4502	DEBUG MODE	–	–	Debug Mode
0x4600	R VFIFO READ START	0x00	RW	Bit[7:0]: vfifo_read_start[15:8] High byte of read start size
0x4601	R VFIFO READ START	–	R	Bit[7:0]: vfifo_read_start[7:0] Low byte of read start size
0x4602	R2	0xF0	RW	Bit[7:4]: Rm Bit[3]: Test1 Bit[2]: Not used Bit[1]: Frame rst enable If set to 1'b1, reset FIFO when sof_i comes Bit[0]: RAM bypass enable If set to 1'b1, SRAM is bypassed completely
0x4513	PSYNC_CTRL_13	0x20	RW	Bit[7:6]: Debug mode Bit[5]: r_vbin_cvdn_avg_en Bit[4]: r_href_mode Bit[3]: r_rblue_re Bit[2]: r_href_in_sel Bit[1:0]: r_channel_sel
0x4514~ 0x4607	DEBUG MODE	–	–	Debug Mode
0x4608	READ_START_H	0x00	RW	Bit[7:0]: Read start point[15:8]
0x4609	READ_START_L	0x04	RW	Bit[7:0]: Read start point[7:0]
0x460A	HSYNC_START_H	0x00	RW	Bit[7:0]: HSYNC start point[15:8]
0x460B	HSYNC_START_L	0xBF	RW	Bit[7:0]: HSYNC start point[7:0]
0x460C	HSYNC_CTRL	0x00	RW	Bit[7:4]: HSYNC_header Bit[3:0]: HSYNC trailer
0x460D	SRAM_TEST	0x01	RW	Bit[7:2]: Reserved Bit[1:0]: SRAM TSEL
0x460E	MIPI_MODE2	0x00	RW	Bit[7]: mipi_mode2_en Bit[6:4]: md2_start_offset Bit[3:2]: Reserved Bit[1:0]: io_spd_ratio 0: Output speed (= input speed) 1: Output speed (= 2 times input speed)

table 7-12 VFIFO control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x460F	IR_DETC_CTRL	0x65	RW	Bit[7:4]: r_ovf_flag_dat Bit[3:0]: r_no_ovf_dat

7.13 MIPI control [0x4800 - 0x484F]

table 7-13 MIPI registers (sheet 1 of 8)

address	register name	default value	R/W	description
0x4800	MIPI CTRL 00	0x04	RW	Bit[7:6]: Not used 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[3:0]: Not used
0x4801	MIPI CTRL 01	0x00	RW	Bit[7]: Reserved Bit[6]: spkt_dt_sel 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]: Reserved Bit[1]: LPX_select for PCLK domain 0: Auto calculate t_lpx_p, unit pclk2x cycle 1: Use lpx_p_min[7:0] Bit[0]: Not used

table 7-13 MIPI registers (sheet 2 of 8)

address	register name	default value	R/W	description
0x4802	MIPI CTRL 02	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	NOT USED	–	–	Not Used
0x4804	MIPI CTRL 04	0x04	RW	Bit[7:4]: man_lane_num Bit[3]: lane_num_manual_enable Bit[2]: lane4_6b_en 0: Not used 1: Supports 4-, 7-, 8-lane 6-bit Bit[1:0]: Not used
0x4805	MIPI CTRL 05	0x00	RW	Bit[7:4]: Reserved 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	NOT USED	–	–	Not Used
0x4807	MIPI CTRL07	0x03	RW	Bit[7:4]: Not used Bit[3:0]: sw_t_lpx ul_tx T_lpx

table 7-13 MIPI registers (sheet 3 of 8)

address	register name	default value	R/W	description
0x4808	MIPI CTRL08	0x0A	RW	Bit[7:0]: wkup_dly mark1 wakeup delay/2^10
0x4810	FCNT MAX	0xFF	RW	Bit[7:0]: fcnt_max[15:8] Max frame counter of frame sync short packet
0x4811	FCNT MAX	0xFF	RW	Bit[7:0]: fcnt_max[7:0] Max frame counter of frame sync short packet
0x4813	MIPI CTRL13	0x00	RW	Bit[7:3]: Reserved Bit[2]: vc_sel Input vc or reg vc Bit[1:0]: Vc Virtual channel of MIPI
0x4814	MIPI CTRL14	0x2A	RW	Bit[7]: Reserved Bit[6]: lpkt_dt_sel 0: Use mipi_d 1: Use dt_man_o as long packet data Bit[5:0]: dt_man Manual data type
0x4815	MIPI CTRL15	0x00	RW	Bit[7]: Reserved 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 data type for short packet
0x4816	EMB DT	0x52	RW	Bit[7:6]: Reserved Bit[5:0]: emb_dt Manually set embedded data type
0x4817	YUV420 FUN	0x00	RW	Bit[7:2]: Reserved Bit[1]: yuv420_2x YUV420 2x in odd line, lcnt[0]=1 Bit[0]: yuv420_en
0x4818	HS ZERO MIN	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: hs_zero_min[9:8] 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] Minimum value of hs_trail, unit ns

table 7-13 MIPI registers (sheet 4 of 8)

address	register name	default value	R/W	description
0x481B	HS TRAIL MIN	0x3C	RW	Bit[7:0]: hs_trail_min[7:0] 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]: Reserved Bit[1:0]: clk_zero_min[9:8] Minimum value of clk_zero, unit ns
0x481D	CLK ZERO MIN	0x06	RW	Bit[7:0]: clk_zero_min[7:0] Minimum value of clk_zero clk_zero_real = clk_zero_min_o + Tui*ui_clk_zero_min_o
0x481E	CLK PREPARE MAX	0x5F	RW	Bit[7:0]: clk_prepare_max[7:0] Max 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]: Reserved Bit[1:0]: clk_post_min[9:8] Minimum value of clk_post, unit ns
0x4821	CLK POST MIN	0x3C	RW	Bit[7:0]: clk_post_min[7:0] 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]: Reserved Bit[1:0]: clk_trail_min[9:8] Minimum value of clk_trail, unit ns
0x4823	CLK TRAIL MIN	0x3C	RW	Bit[7:0]: clk_trail_min[7:0] 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]: Reserved Bit[1:0]: lpx_p_min[9:8] Minimum value of lpx_p, unit ns
0x4825	LPX P MIN	0x32	RW	Bit[7:0]: lpx_p_min[7:0] Minimum value of lpx_p lpx_p_real = lpx_p_min_o + Tui*ui_lpx_p_min_o
0x4826	HS PREPARE MIN	0x28	RW	Bit[7:0]: hs_prepare_min[7:0] Minimum value of hs_prepare, unit ns

table 7-13 MIPI registers (sheet 5 of 8)

address	register name	default value	R/W	description
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]: Reserved Bit[1:0]: hs_exit_min[9:8] Minimum value of hs_exit, unit ns
0x4829	HS EXIT MIN	0x64	RW	Bit[7:0]: hs_exit_min[7:0] 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]: Reserved 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]: Reserved 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]: Reserved 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 Bit[3:0]: ui_clk_prepare_min Value of clk_prepare, unit UI
0x482E	UI CLK POST MIN	0x34	RW	Bit[7:6]: Reserved 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]: Reserved 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[7:4]: Reserved 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 Bit[3:0]: ui_hs_prepare_min UI value of hs_prepare, unit UI
0x4832	UI HS EXIT MIN	0x00	RW	Bit[7:4]: Reserved Bit[5:0]: ui_hs_exit_min[5:0] Minimum UI value of hs_exit, unit UI
0x4836	NOT USED	–	–	Not Used

table 7-13 MIPI registers (sheet 6 of 8)

address	register name	default value	R/W	description
0x4837	PCLK PERIOD	0x10	RW	Bit[7:0]: pclk_period[7:0] Period of pclk2x, pclk_div=1 and 1 bit decimal
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
0x4848	MIPI LP GPIO1	0x00	RW	Bit[7]: lp_sel2 0: Auto generate mipi_lp_dir2_o 1: Use lp_dir_man2 to be mipi_lp_dir2_o Bit[6]: lp_dir_man2 0: Input 1: Output Bit[5]: lp_p2_o Bit[4]: lp_n2_o Bit[3]: lp_sel3 0: Auto generate mipi_lp_dir3_o 1: Use lp_dir_man3 to be mipi_lp_dir3_o Bit[2]: lp_dir_man3 0: Input 1: Output Bit[1]: lp_p3_o Bit[0]: lp_n3_o

table 7-13 MIPI registers (sheet 7 of 8)

address	register name	default value	R/W	description
0x483A	MIPI LP GPIO2	0x00	RW	Bit[7]: lp_sel4 0: Auto generate mipi_lp_dir4_o 1: Use lp_dir_man4 to be mipi_lp_dir4_o Bit[6]: lp_dir_man4 0: Input 1: Output Bit[5]: lp_p4_o Bit[4]: lp_n4_o Bit[3]: lp_sel5 0: Auto generate mipi_lp_dir5_o 1: Use lp_dir_man5 to be mipi_lp_dir5_o Bit[2]: lp_dir_man5 0: Input 1: Output Bit[1]: lp_p5_o Bit[0]: lp_n5_o
0x483B	MIPI LP GPIO3	0x00	RW	Bit[7]: lp_sel6 0: Auto generate mipi_lp_dir6_o 1: Use lp_dir_man6 to be mipi_lp_dir6_o Bit[6]: lp_dir_man6 0: Input 1: Output Bit[5]: lp_p6_o Bit[4]: lp_n6_o Bit[3]: lp_sel7 0: Auto generate mipi_lp_dir7_o 1: Use lp_dir_man7 to be mipi_lp_dir7_o Bit[2]: lp_dir_man7 0: Input 1: Output Bit[1]: lp_p7_o Bit[0]: lp_n7_o
0x483C	MIPI CTRL3C	0x02	RW	Bit[7:4]: Reserved Bit[3:0]: t_clk_pre Unit: pclk2x cycle

table 7-13 MIPI registers (sheet 8 of 8)

address	register name	default value	R/W	description
0x483D	MIPI LP GPIO4	0x00	RW	Bit[7]: lp_ck_sel0 0: Auto generate mipi_ck_lp_dir0_o 1: Use lp_ck_dir_man0 to be mipi_ck_lp_dir0_o Bit[6]: lp_ck_dir_man0 0: Input 1: Output Bit[5]: lp_ck_p0_o Bit[4]: lp_ck_n0_o Bit[3]: lp_ck_sel1 0: Auto generate mipi_ck_lp_dir1_o 1: Use lp_ck_dir_man1 to be mipi_ck_lp_dir1_o Bit[2]: lp_ck_dir_man1 0: Input 1: Output Bit[1]: lp_ck_p1_o Bit[0]: lp_ck_n1_o
0x484A	SEL MIPI CTRL4A	0x27	RW	Bit[7:6]: Reserved 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	NOT USED	–	–	Not Used
0x484C	SEL MIPI CTRL4C	0x03	RW	Bit[7]: Reserved Bit[6]: Not used Bit[5]: prbs_enable Bit[4]: hs_test_only MIPI high speed only test mode enable Bit[3]: Not used Bit[2:0]: Reserved
0x484D	TEST PATTEN DATA	0xB6	RW	Bit[7:0]: test_patten_data[7:0] Data lane test pattern register
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[7:0]: test_patten_ck_data Clock test pattern register

7.14 LVDS top control [0x4A00 - 0x4A0F]

table 7-14 LVDS top control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x4A00	LVDS R0	0x2A	RW	Bit[7]: Two sync code enable in lane8 mode Bit[6]: Sync code manual mode enable Bit[5]: Sync code enable when only 1 lane Bit[4]: lvds_pclk_inv Bit[3]: Channel id enable in sync per lane mode Bit[2]: CCIR parameter F Bit[1]: Sav first enable Bit[0]: Sync code mode 0: Split 1: Per lane
0x4A02	LVDS DUMMY DATA0	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: lvds_dummy_data0[11:8] Dummy data0
0x4A03	LVDS DUMMY DATA0	0x80	RW	Bit[7:0]: lvds_dummy_data0[7:0] Dummy data0
0x4A04	LVDS DUMMY DATA1	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: lvds_dummy_data1[11:8] Dummy data1
0x4A05	LVDS DUMMY DATA1	0x10	RW	Bit[7:0]: lvds_dummy_data1[7:0] Dummy data1
0x4A06	LVDS R6	0xAA	RW	Bit[7:0]: lvds_r6 Sync code of frame_start in manual sync code mode
0x4A07	LVDS R7	0x55	RW	Bit[7:0]: lvds_r7 Sync code of frame_end in manual sync code mode
0x4A08	LVDS R8	0x99	RW	Bit[7:0]: lvds_r8 Sync code of line_start in manual sync code mode
0x4A09	LVDS R9	0x66	RW	Bit[7:0]: lvds_r9 Sync code of line_end in manual sync code mode
0x4A0A	LVDS RA	0x00	RW	Bit[7:3]: Reserved Bit[2]: r_hts_man_en Bit[1]: r_ln2_sel Bit[0]: r_chk_pcnt
0x4A0C	LVDS BLK TIMES	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: lvds_blk_times[11:8] High byte of r_blk_times

table 7-14 LVDS top control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x4A0D	LVDS BLK TIMES	0x05	RW	Bit[7:0]: lvds_blk_times[7:0] Low byte of r_blk_times
0x4A0E	LVDS HTS MAN	0x00	RW	Bit[7:0]: lvds_hts_man[15:8] High byte of hts_man
0x4A0F	LVDS HTS MAN	0x00	RW	Bit[7:0]: lvds_hts_man[7:0] Low byte of hts_man

7.15 frame HDR control [0x4580-0x4587]

table 7-15 frame HDR registers

address	register name	default value	R/W	description
0x4580	HDR FRAME REG0	0x00	RW	Bit[7:4]: Reserved Bit[3]: hdrfm_bypass Bit[2]: Reserved Bit[1:0]: hdrfm_mode 0: non_HDRfm 1: Row HDR, n(n>=2): n frame HDR
0x4581	HDR FRAME REG1	0x30	RW	Bit[7:6]: Reserved Bit[5]: gain_chg_en Bit[4]: exp_chg_en Bit[3]: rst_lat_pre Bit[2]: exp_gain_lat_span2 Bit[1]: exp_gain_lat_span Bit[0]: frm_cnt_next_interrupt
0x4582	HDR FRAME REG2	0x80	RW	Bit[7]: tc_r_current Bit[6:0]: tc_r_ofst
0x4583	HDR FRAME REG3	0x81	RW	Bit[7]: tc_cs_current Bit[6:0]: tc_cs_ofst
0x4584	HDR FRAME REG4	0x0F	RW	Bit[7:4]: Reserved Bit[3:2]: Vtsdble Bit[1:0]: Htshlf
0x4585	HDR FRAME REG5	0x00	RW	Bit[7:0]: Reserved
0x4586	HDR FRAME REG6	0x00	RW	Bit[7:0]: Reserved
0x4587	HDR FRAME REG7	0x00	RW	Bit[7:0]: Reserved

7.16 BLC control [0x4000 - 0x401E]

table 7-16 BLC registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x4000	BLC CTRL0	0x0C	RW	Bit[7:6]: Debug mode Bit[5:4]: Not used Bit[3:2]: Debug mode Bit[1]: mid_filter_en Median filter function enable signal 0: Disable 1: Enable Bit[0]: Not used
0x4001	START LINE	0x04	RW	Bit[7:6]: Not used Bit[5:0]: Debug mode
0x4002	BLC CTRL2	0x45	RW	Bit[7]: format_change_en Format change trigger enable signal When set to 1, BLC will be applied in following N frames. N is set by bit[5:0] Bit[6]: blc_auto_en BLC auto mode enable signal 0: Used black levels are set by registers 0x400C ~ 0x401B 1: Used black levels are calculated values Bit[5:0]: reset_frame_num Specify the frame number for reset and/or format change BLC trigger
0x4003	BLC CTRL3	0x01	RW	Bit[7]: blc_trig_man Manual trigger enable signal When enabled, BLC will be applied in following N frames. N is set in bit[5:0] Bit[6]: Freeze It has lower priority than always_do. When set and always_do is 0, the BLC will freeze Bit[5:0]: manual_frame_num Specify BLC update frame number for manual trigger
0x4004	LINE NUM	0x08	RW	Bit[7:6]: Not used Bit[5:0]: Debug mode
0x4005	BLC CTRL5	0x40	RW	Bit[7:2]: Debug mode Bit[1]: blc_always_up_en 0: Normal freeze 1: BLC always update Bit[0]: Debug mode

table 7-16 BLC registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x4007	BLC CTRL7	0x20	RW	Bit[7:6]: Not used Bit[5]: gain_change_en Gain change trigger enable signal When set to1, BLC will be applied in the following frame Bit[4:0]: Debug mode
0x4008	BLC TARGET	0x00	RW	Bit[7:0]: Blackleveltarget0[15:8]
0x4009	BLC TARGET	0x40	RW	Bit[7:0]: Blackleveltarget0[7:0]
0x400C~ 0x401C	BLC REG	–	RW	Debug Mode
0x401D	STABLE RANGE	0x7F	RW	Bit[7:0]: stable_range Specify the threshold which can trigger BLC update If offset difference of previous and current black level is out of stable_range, BLC will be updated
0x401E	DEBUG MODE	0x80	RW	Debug Mode

7.17 window control [0x5A00 - 0x5A2F]

table 7-17 window control registers (sheet 1 of 5)

address	register name	default value	R/W	description
0x5A00	XSTART MAN	0x00	RW	Bit[7:5]: Not Used Bit[4:0]: Xstart[12:8] 5 MSB of the manual horizontal start for manual output window
0x5A01	XSTART MAN	0x00	RW	Bit[7:0]: Xstart[7:0] 8 LSB of the manual horizontal start for manual output window
0x5A02	YSTART MAN	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Ystart[11:8] 4 MSB of the manual vertical start for manual output window
0x5A03	YSTART MAN	0x00	RW	Bit[7:0]: Ystart[7:0] 8 LSB of the manual vertical start for manual output window

table 7-17 window control registers (sheet 2 of 5)

address	register name	default value	R/W	description
0x5A04	XWIN MAN	0x05	RW	Bit[7:5]: Not used Bit[4:0]: x_win[12:8] 5 MSB of the manual horizontal size for manual output window
0x5A05	XWIN MAN	0x00	RW	Bit[7:0]: x_win[7:0] 8 LSB of the manual horizontal size for manual output window
0x5A06	YWIN MAN	0x03	RW	Bit[7:4]: Not used Bit[3:0]: y_win[11:8] 4 MSB of the manual vertical size for manual output window
0x5A07	YWIN MAN	0x20	RW	Bit[7:0]: y_win[7:0] 8 LSB of the manual vertical size for manual output window
0x5A08	WINC CTRL	0x00	RW	Bit[7:5]: Not used Bit[4]: win16_en 16-window function enable signal 0: Disable 1: Enable Bit[3]: sub_line_out 16-window sub-line enable signal 0: Disable 1: Enable Bit[2:1]: Not used Bit[0]: win_man_en Manual window enable signal 0: Output window uses the system parameters 1: Output window uses manual parameters which are set in registers 0x5A00~0x5A07
0x5A10	WIN16 XSTART0	0x00	RW	Bit[7:5]: Not used Bit[4:0]: win16_xstart0[12:8] 5 MSB of the first horizontal start for 16-window function
0x5A11	WIN16 XSTART0	0x00	RW	Bit[7:0]: win16_xstart0[7:0] 8 LSB of the first horizontal start for 16- window function
0x5A12	WIN16 XSTART1	0x00	RW	Bit[7:5]: Not used Bit[4:0]: win16_xstart1[12:8] 5 MSB of the second horizontal start for 16-window function

table 7-17 window control registers (sheet 3 of 5)

address	register name	default value	R/W	description
0x5A13	WIN16 XSTART1	0x00	RW	Bit[7:0]: win16_xstart1[7:0] 8 LSB of the second horizontal start for 16-window function
0x5A14	WIN16 XSTART2	0x00	RW	Bit[7:5]: Not used Bit[4:0]: win16_xstart2[12:8] 5 MSB of the third horizontal start for 16-window function
0x5A15	WIN16 XSTART2	0x00	RW	Bit[7:0]: win16_xstart2[7:0] 8 LSB of the third horizontal start for 16-window function
0x5A16	WIN16 XSTART3	0x00	RW	Bit[7:5]: Not used Bit[4:0]: win16_xstart3[12:8] 5 MSB of the fourth horizontal start for 16-window function
0x5A17	WIN16 XSTART3	0x00	RW	Bit[7:0]: win16_xstart3[7:0] 8 LSB of the fourth horizontal start for 16-window function
0x5A18	WIN16 YSTART0	0x00	RW	Bit[7:4]: Not used Bit[3:0]: win16_ystart0[11:8] 4 MSB of the first vertical start for 16-window function
0x5A19	WIN16 YSTART0	0x00	RW	Bit[7:0]: win16_ystart0[7:0] 8 LSB of the first vertical start for 16-window function
0x5A1A	WIN16 YSTART1	0x00	RW	Bit[7:4]: Not used Bit[3:0]: win16_ystart1[11:8] 4 MSB of the second vertical start for 16-window function
0x5A1B	WIN16 YSTART1	0x00	RW	Bit[7:0]: win16_ystart1[7:0] 8 LSB of the second vertical start for 16-window function
0x5A1C	WIN16 YSTART2	0x00	RW	Bit[7:4]: Not used Bit[3:0]: win16_ystart2[11:8] 4 MSB of the third vertical start for 16-window function
0x5A1D	WIN16 YSTART2	0x00	RW	Bit[7:0]: win16_ystart2[7:0] 8 LSB of the third vertical start for 16-window function
0x5A1E	WIN16 YSTART3	0x00	RW	Bit[7:4]: Not used Bit[3:0]: win16_ystart3[11:8] 4 MSB of the fourth vertical start for 16-window function

table 7-17 window control registers (sheet 4 of 5)

address	register name	default value	R/W	description
0x5A1F	WIN16 YSTART3	0x00	RW	Bit[7:0]: win16_ystart3[7:0] 8 LSB of the fourth vertical start for 16-window function
0x5A20	WIN16 XWIN0	0x00	RW	Bit[7:5]: Not used Bit[4:0]: win16_xsize0[12:8] 5 MSB of the first horizontal size for 16-window function
0x5A21	WIN16 XWIN0	0x00	RW	Bit[7:0]: win16_xsize0[7:0] 8 LSB of the first horizontal size for 16-window function
0x5A22	WIN16 XWIN1	0x00	RW	Bit[7:5]: Not Used Bit[4:0]: win16_xsize1[12:8] 5 MSB of the second horizontal size for 16-window function
0x5A23	WIN16 XWIN1	0x00	RW	Bit[7:0]: win16_xsize1[7:0] 8 LSB of the second horizontal size for 16-window function
0x5A24	WIN16 XWIN2	0x00	RW	Bit[7:5]: Not used Bit[4:0]: win16_xsize2[12:8] 5 MSB of the third horizontal size for 16-window function
0x5A25	WIN16 XWIN2	0x00	RW	Bit[7:0]: win16_xsize2[7:0] 8 LSB of the third horizontal size for 16-window function
0x5A26	WIN16 XWIN3	0x00	RW	Bit[7:5]: Not used Bit[4:0]: win16_xsize3[12:8] 5 MSB of the fourth horizontal size for 16-window function
0x5A27	WIN16 XWIN3	0x00	RW	Bit[7:0]: win16_xsize3[7:0] 8 LSB of the fourth horizontal size for 16-window function
0x5A28	WIN16 YWIN0	0x00	RW	Bit[7:4]: Not used Bit[3:0]: win16_ysize0[11:8] 4 MSB of the first vertical size for 16-window function
0x5A29	WIN16 YWIN0	0x00	RW	Bit[7:0]: win16_ysize0[7:0] 8 LSB of the first vertical size for 16-window function
0x5A2A	WIN16 YWIN1	0x00	RW	Bit[7:4]: Not used Bit[3:0]: win16_ysize1[11:8] 4 MSB of the second vertical size for 16-window function

table 7-17 window control registers (sheet 5 of 5)

address	register name	default value	R/W	description
0x5A2B	WIN16 YWIN1	0x00	RW	Bit[7:0]: win16_ysize1[7:0] 8 LSB of the second vertical size for 16-window function
0x5A2C	WIN16 YWIN2	0x00	RW	Bit[7:4]: Not used Bit[3:0]: win16_ysize2[11:8] 4 MSB of the third vertical size for 16-window function
0x5A2D	WIN16 YWIN2	0x00	RW	Bit[7:0]: win16_ysize2[7:0] 8 LSB of the third vertical size for 16-window function
0x5A2E	WIN16 YWIN3	0x00	RW	Bit[7:4]: Not used Bit[3:0]: win16_ysize3[11:8] 4 MSB of the fourth vertical size for 16-window function
0x5A2F	WIN16 YWIN3	0x00	RW	Bit[7:0]: win16_ysize3[7:0] 8 LSB of the fourth vertical size for 16-window function

7.18 DPC control [0x5400 - 0x5426]

table 7-18 DPC control registers

address	register name	default value	R/W	description
0x5400~ 0x5426	DPC CTRL REGISTERS	–	RW	DPC Control Registers

7.19 LENC [0x5200 - 0x521C]

table 7-19 LENC registers (sheet 1 of 5)

address	register name	default value	R/W	description
0x5200	RED X0	0x02	RW	Bit[7:3]: Not used Bit[2:0]: red_x0[10:8] 3 MSB of horizontal center position of red channel

table 7-19 LENC registers (sheet 2 of 5)

address	register name	default value	R/W	description
0x5201	RED X0	0x90	RW	Bit[7:0]: red_x0[7:0] 8 LSB of horizontal center position of red channel
0x5202	RED Y0	0x01	RW	Bit[7:2]: Not used Bit[1:0]: red_y0[9:8] 2 MSB of vertical center position of red channel
0x5203	RED Y0	0x9A	RW	Bit[7:0]: red_y0[7:0] 8 LSB of vertical center position of red channel
0x5204	RED A1	0x22	RW	Bit[7]: Not used Bit[6:0]: red_a1 Gain coefficient for the square distance of current red pixel position and red center
0x5205	RED A2	0x07	RW	Bit[7:4]: Not used Bit[3:0]: red_a2 Precision of gain coefficient for the square distance of current red pixel position and red center
0x5206	RED B1	0xC2	RW	Bit[7:0]: red_b1 Gain coefficient for the distance of current red pixel position and red center
0x5207	RED B2	0x08	RW	Bit[7:4]: Not used Bit[3:0]: red_b2 Precision of gain coefficient for the distance of current red pixel position and red center Gain of current red pixel is defined with the distance (r), red_a1, red_a2, red_b1 and red_b2 Gain = [(red_a1* r2)>>red_a2] + [(red_b1*r)>>red_b2]
0x5208	GRN X0	0x02	RW	Bit[7:3]: Not used Bit[2:0]: grn_x0[10:8] 3 MSB of horizontal center position of green channel
0x5209	GRN X0	0x90	RW	Bit[7:0]: grn_x0[7:0] 8 LSB of horizontal center position of green channel

table 7-19 LENC registers (sheet 3 of 5)

address	register name	default value	R/W	description
0x520A	GRN Y0	0x01	RW	Bit[7:2]: Not used Bit[1:0]: grn_y0[9:8] 2 MSB of vertical center position of green channel
0x520B	GRN Y0	0x9A	RW	Bit[7:0]: grn_y0[7:0] 8 LSB of vertical center position of green channel
0x520C	GRN A1	0x22	RW	Bit[7]: Not used Bit[6:0]: grn_a1 Gain coefficient for the square distance of current green pixel position and green center
0x520D	GRN A2	0x07	RW	Bit[7:4]: Not used Bit[3:0]: grn_a2 Precision of gain coefficient for the square distance of current green pixel position and green center
0x520E	GRN B1	0xC2	RW	Bit[7:0]: grn_b1 Gain coefficient for the distance of current green pixel position and green center
0x520F	GRN B2	0x08	RW	Bit[7:4]: Not used Bit[3:0]: grn_b2 Precision of gain coefficient for the distance of current green pixel position and green center Gain of current green pixel is defined with the distance (r), grn_a1, grn_a2, grn_b1 and grn_b2 Gain = [(grn_a1* r2)>>grn_a2] + [(grn_b1*r)>>grn_b2]
0x5210	BLU X0	0x02	RW	Bit[7:3]: Not used Bit[2:0]: blu_x0[10:8] 3 MSB of horizontal center position of blue channel
0x5211	BLU X0	0x90	RW	Bit[7:0]: blu_x0[7:0] 8 LSB of horizontal center position of blue channel
0x5212	BLU Y0	0x01	RW	Bit[7:2]: Not used Bit[1:0]: blu_y0[9:8] 2 MSB of vertical center position of blue channel

table 7-19 LENC registers (sheet 4 of 5)

address	register name	default value	R/W	description
0x5213	BLU Y0	0x9A	RW	Bit[7:0]: blu_y0[7:0] 8 LSB of vertical center position of blue channel
0x5214	BLU A1	0x22	RW	Bit[7]: Not used Bit[6:0]: blu_a1 Gain coefficient for the square distance of current blue pixel position and blue center
0x5215	BLU A2	0x07	RW	Bit[7:4]: Not used Bit[3:0]: blu_a2 Precision of gain coefficient for the square distance of current blue pixel position and blue center
0x5216	BLU B1	0xC2	RW	Bit[7:0]: blu_b1 Gain coefficient for the distance of current blue pixel position and blue center
0x5217	BLU B2	0x08	RW	Bit[7:4]: Not used Bit[3:0]: blu_b2 Precision of gain coefficient for the distance of current blue pixel position and blue center Gain of current blue pixel is defined with the distance (r), blu_a1, blu_a2, blu_b1 and blu_b2 Gain = [(blu_a1* r2)>>blu_a2] + [(blu_b1*r)>>blu_b2]
0x5218	LENC CTRL00	0x04	RW	Bit[7:3]: Not used Bit[2]: rnd_en Round enable to generate random round bit Bit[1]: adj_coef_man_en Manual adjust coefficient enable 0: Disable 1: Enable Bit[0]: lenc_gain_adj_coef_en LENC gain adjust coefficient enable signal 0: Disable 1: Enable
0x5219	COEF TH	0x80	RW	Bit[7:0]: coef_th[7:0] Coefficient threshold for higher real gain

table 7-19 LENC registers (sheet 5 of 5)

address	register name	default value	R/W	description
0x521A	GAIN TH1	0x06	RW	Bit[7]: Not used Bit[6:0]: gain_th1 Lower threshold of real gain When the current real gain is less than gain_th1, the adjust coefficient of LENC gain is 1
0x521B	GAIN TH2	0x00	RW	Bit[7]: Not used Bit[6:0]: gain_th2 Higher threshold of real gain. When the current real gain is larger than gain_th2, the adjust coefficient of LENC gain is adj_coef_th with 7-bit precision. When current real gain is less than gain_th2 and larger than gain_th1, the adjust coefficient is [128-(real_gain-gain_th1)*(128-adj_coef_th) / (gain_th2-gain_th1)]
0x521C	COEF MAN	0x80	RW	Bit[7:0]: coef_man[7:0] Manual adjust coefficient for LENC gain with 7-bit precision. Its range is [0,128]

7.20 ISP general control [0x5000 - 0x500F, 0x5080 - 0x5091]

table 7-20 ISP general registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x5000	ISP CTRL00	0xFF	RW	Bit[7]: Debug mode Bit[6]: dbc_en DBC function enable signal 0: Disable 1: Enable Bit[5]: otp_en OTP function enable signal 0: Disable 1: Enable Bit[4]: awbg_en AWBG function enable signal 0: Disable 1: Enable Bit[3]: bc_en Black DPC function enable signal 0: Disable 1: Enable Bit[2]: wc_en White DPC function enable signal 0: Disable 1: Enable Bit[1]: lenc_en LENC function enable signal 0: Disable 1: Enable Bit[0]: isp_en ISP functions enable signal 0: Disable 1: Enable
0x5001	ISP CTRL01	0x1F	RW	Bit[7:5]: Debug mode Bit[4]: avg_en AVG function enable signal 0: Enable 1: Disable Bit[3]: Debug mode Bit[2]: blc_en BLC function enable signal 0: Enable 1: Disable Bit[1:0]: Debug mode

table 7-20 ISP general registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x5002	ISP CTRL02	0x00	RW	Bit[7]: Debug mode Bit[6]: avg_sel Select signal for AVG input 0: Select WINC output as AVG input 1: Select DGC output as AVG input Bit[5:0]: Debug mode
0x5003	ISP CTRL03	0x00	RW	Bit[7:0]: Debug mode
0x5004	RED GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: red_gain[11:8] AWB gain for red channel
0x5005	RED GAIN	0x00	RW	Bit[7:0]: red_gain[7:0] AWB gain for red channel
0x5006	GRN GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: grn_gain[11:8] AWB gain for green channel
0x5007	GRN GAIN	0x00	RW	Bit[7:0]: grn_gain[7:0] AWB gain for green channel
0x5008	BLU GAIN	0x04	RW	Bit[7:4]: Not used Bit[3:0]: blu_gain[11:8] AWB gain for blue channel
0x5009	BLU GAIN	0x00	RW	Bit[7:0]: blu_gain[7:0] AWB gain for blue channel
0x500A	ISP CTRL0A	0x85	RW	Bit[7:0]: Debug mode
0x500B	ISP CTRL0B	0x10	RW	Bit[7:0]: Debug mode
0x500C	ISP CTRL0C	0x00	RW	Bit[7:0]: Debug mode
0x500D	ISP CTRL0D	0xFF	RW	Bit[7:0]: Debug mode
0x500E	ISP CTRL0E	0xFF	RW	Bit[7:0]: Debug mode
0x500F	ISP CTRL0F	0xFF	RW	Bit[7:0]: Debug mode

table 7-20 ISP general registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x5080	CTRL00	0x00	RW	Bit[7]: test_en 0: Disable test function 1: Enable test function Bit[6]: rolling_bar_en 0: Disable rolling bar function 1: Enable rolling bar function Bit[5]: transparent_en 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_mode 00: Color bar 01: Random data 10: Square 11: Black image
0x5081	CTRL01	0x01	RW	Bit[7]: Not used Bit[6]: win_cut_en 0: Do not cut the redundant pixels 1: Cut the redundant pixels Bit[5]: two_lsb_0_en When set, two LSB of output date are 0 Bit[4]: same_seed_en When set, the seed used to generate the random data are the same which is set by bit[3:0] Bit[3:0]: Seed Seed used in generating random data
0x5082~ 0x5091	PRE REG	–	RW	Debug Mode

7.21 LED control [0x3900 - 0x391C]

table 7-21 LED registers

address	register name	default value	R/W	description
0x3900	REG0	0x10	RW	Bit[7:0]: div_reg[15:8]
0x3901	REG1	0x00	RW	Bit[7:0]: div_reg[7:0]
0x3903	REG3	0x00	RW	Bit[7:0]: duty_reg[7:0] Duty cycle 0 ~ 65535: Clock number
0x3904	REG4	0x00	RW	Bit[7:0]: low_limit[15:8]
0x3905	REG5	0x01	RW	Bit[7:0]: low_limit[7:0]
0x3906	REG6	0x00	RW	Bit[7:1]: Not used Bit[0]: led_gen_en_manual 0: led_gen enable at the first frame begin 1: Always enable led_gen
0x3907	REG7	0x00	RW	Bit[7]: led_gen_sel Bit[6:0]: Reserved
0x3910	REG10	0xFF	RW	Bit[7:0]: div_reg[15:8]
0x3911	REG11	0xFF	RW	Bit[7:0]: div_reg[7:0]
0x3912	REG12	0x08	RW	Bit[7:0]: duty_reg[15:8] 0~65535: 0% ~ 100%
0x3913	REG13	0x00	RW	Bit[7:0]: duty_reg[7:0] 0~65535: 0% ~ 100%
0x3914	REG14	0x00	RW	Bit[7:0]: led_duty_cycle_low_reg[15:8]
0x3915	REG15	0x00	RW	Bit[7:0]: led_duty_cycle_low_reg[7:0]
0x3916	REG16	0x00	RW	Bit[7:0]: led_average_l_low_reg[9:8]
0x3917	REG17	0x00	RW	Bit[7:0]: led_average_l_low_reg[7:0]
0x3918	REG18	0x00	RW	Bit[7:0]: average_low_reg[9:8]
0x3919	REG19	0x00	RW	Bit[7:0]: average_low_reg[7:0]
0x391A	REG1A	0x00	RW	Bit[7:0]: led_duty_cycle_slope_reg
0x391B	REG1B	0x74	RW	Bit[7:0]: led_pwm_ctrl1
0x391C	REG1C	0x00	RW	Bit[7:5]: Not used Bit[4]: Duty cycle calculation option Bit[3:1]: Not used Bit[0]: Divider start calculate option

8 operating specifications

8.1 absolute maximum ratings

table 8-1 absolute maximum ratings

parameter		absolute maximum rating[a]
ambient storage temperature		-40°C to +95°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.

8.2 functional temperature

table 8-2 functional temperature

parameter	range
operating temperature[a]	-30°C to +85°C junction temperature
stable image temperature[b]	0°C to 50°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

8.3 DC characteristics

table 8-3 DC characteristics (T_A = 23°C ± 2°C)

symbol	parameter	min	typ	max	unit
supply					
V_{DD-A}	supply voltage (analog)	2.6	2.8	3.0	V
V_{DD-D}[a]	supply voltage (digital core for 2-lane MIPI up to 1 Gbps/lane)	1.425	1.5	1.575	V
V_{DD-IO}	supply voltage (digital I/O)	1.7	1.8	3.0	V
I_{DD-A}	active (operating) current		25	35	mA
I_{DD-IO}			75	90	mA
$I_{DDS-SCCB}$	standby current		30	50	µA
$I_{DDS-PWDN}$			30	50	µA
$I_{DDS-XSHUTDOWN}$			5	10	µA
digital inputs (typical conditions: AVDD = 2.8V, DVDD = 1.5V, 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}[b]	SIOC and SIOD	-0.5	0	0.54	V
V_{IH}	SIOC and SIOD	1.28	1.8	3.0	V

a. using the internal regulator is strongly recommended for minimum power down currents

b. based on DOVDD = 1.8V

8.4 timing characteristics

table 8-4 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			5 (10[a])	ns

a. if using internal PLL

9 mechanical specifications

9.1 physical specifications

figure 9-1 package specifications

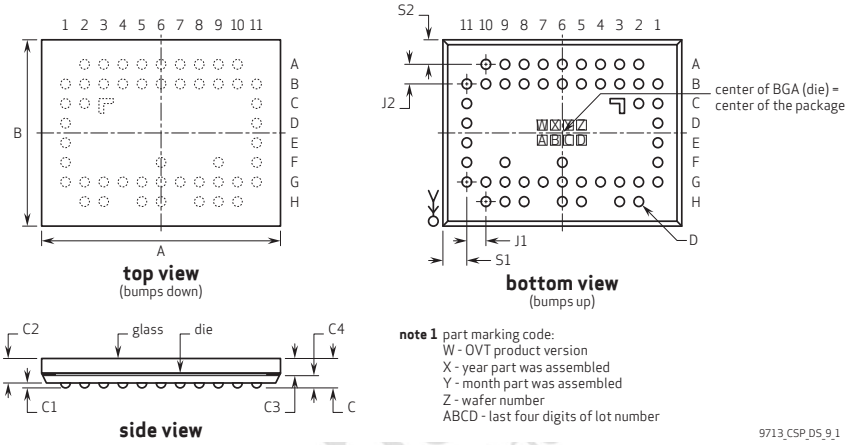


table 9-1 package dimensions

parameter	symbol	min	typ	max	unit
package body dimension x	A	6085	6110	6135	µm
package body dimension y	B	4905	4930	4955	µm
package height	C	690	750	810	µm
ball height	C1	100	130	160	µm
package body thickness	C2	575	620	665	µm
thickness of glass surface to wafer	C3	425	445	465	µm
image plane height	C4	250	305	360	µm
ball diameter	D	220	250	280	µm
total pin count	N		49 (5 NC)		
pin count x-axis	N1		11		
pin count y-axis	N2		8		
pins pitch x-axis	J1		500		µm
pins pitch y-axis	J2		580		µm
edge-to-pin center distance analog x	S1	525	555	585	µm
edge-to-pin center distance analog y	S2	405	435	465	µm

9.2 IR reflow specifications

note
the OV9713 uses a lead free package.

figure 9-2 IR reflow ramp rate requirements

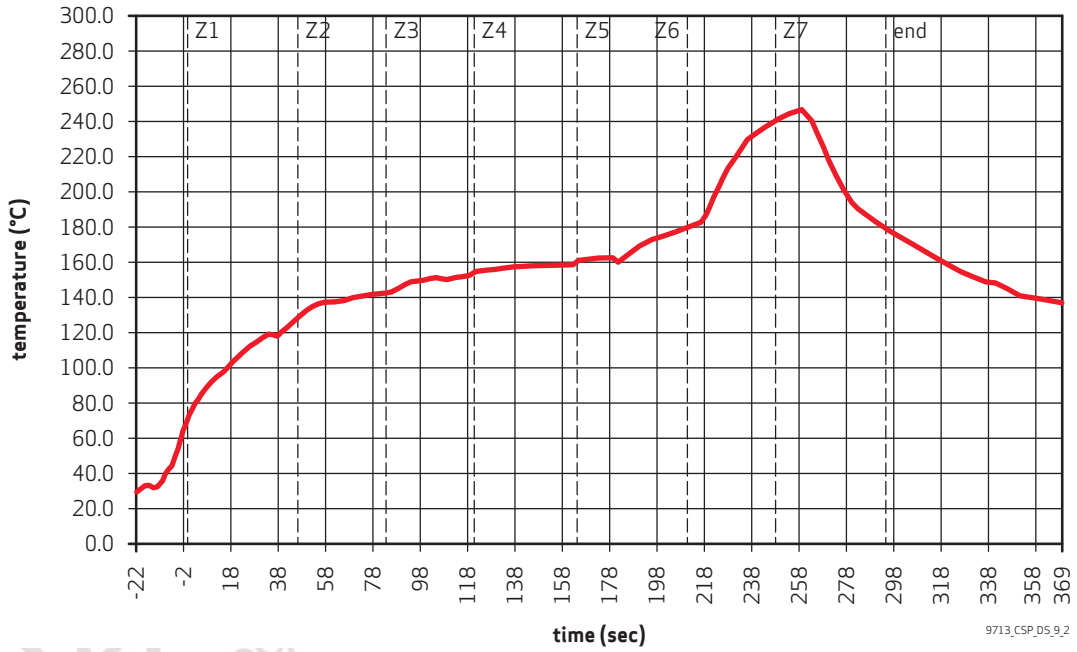


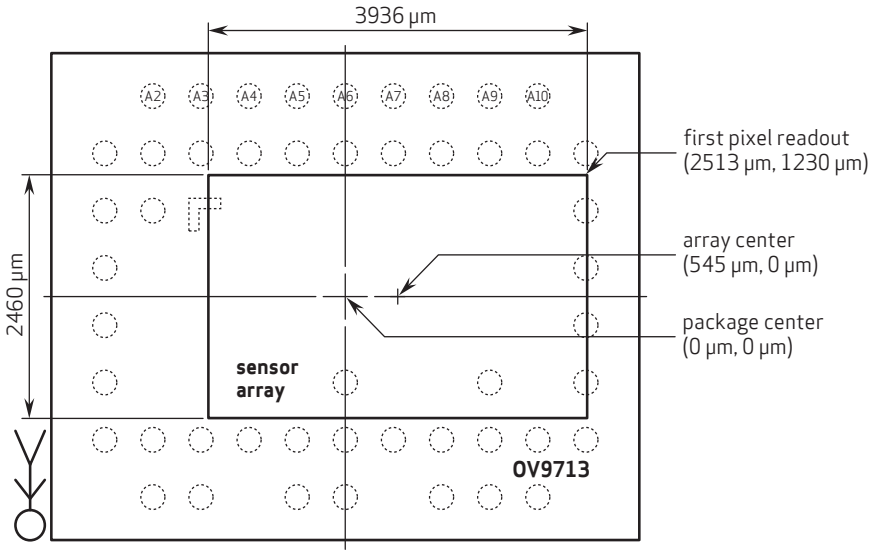
table 9-2 reflow conditions

condition	exposure
average ramp-up rate (30°C to 217°C)	less than 3°C per second
> 100°C	between 330 - 600 seconds
> 150°C	at least 210 seconds
> 217°C	at least 30 seconds (30 ~ 120 seconds)
peak temperature	245°C
cool-down rate (peak to 50°C)	less than 6°C per second
time from 30°C to 245°C	no greater than 390 seconds

10 optical specifications

10.1 sensor array center

figure 10-1 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 A2 to A10 oriented down on the PCB.

9713_CSP_DS_10_1

10.2 lens chief ray angle (CRA)

figure 10-2 chief ray angle (CRA)

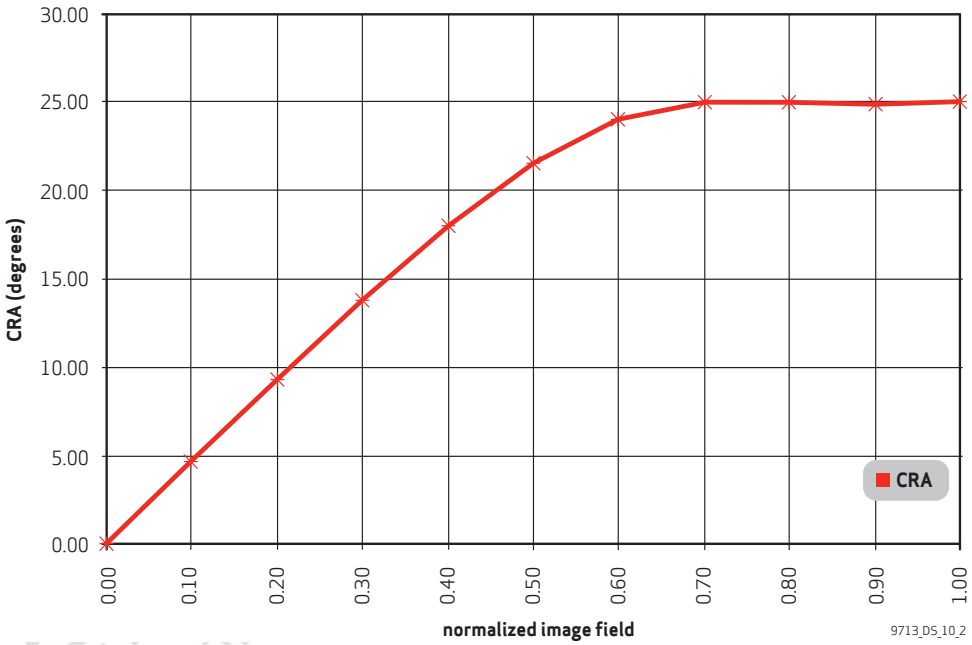
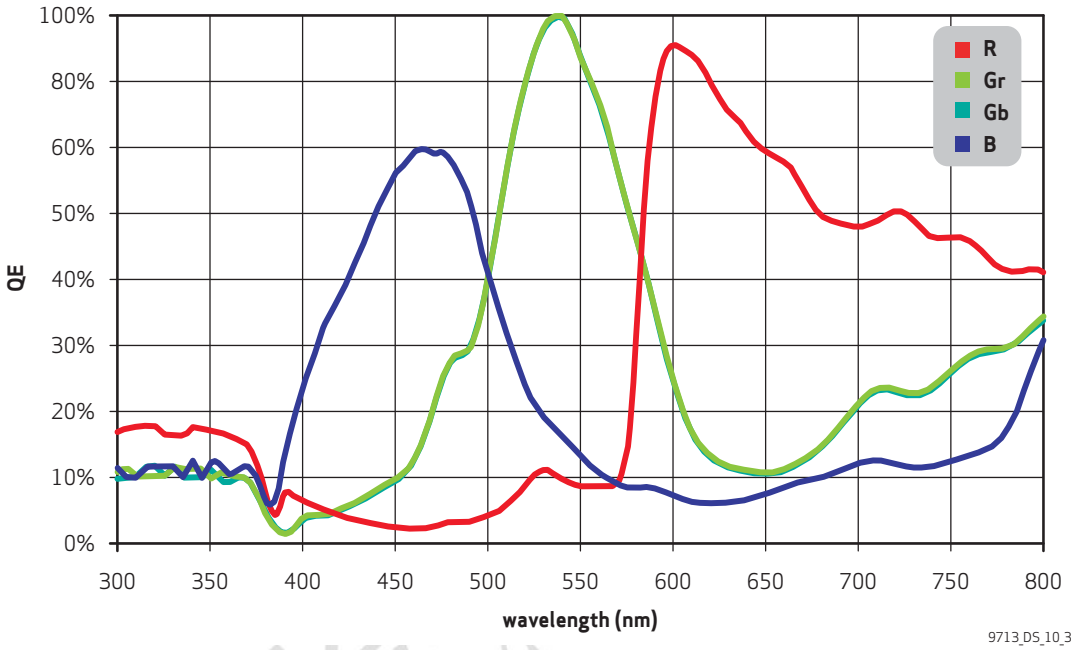


table 10-1 CRA versus image height plot

field (%)	image height (mm)	CRA (degrees)
0.00	0.000	0.0
0.10	0.226	4.7
0.20	0.453	9.3
0.30	0.679	13.8
0.40	0.906	18.0
0.50	1.132	21.6
0.60	1.358	24.0
0.70	1.585	25.0
0.80	1.811	25.0
0.90	2.038	24.9
1.00	2.264	25.0

10.3 spectrum response curve

figure 10-3 spectrum response curve diagram



revision history

version 1.0 10.20.2011

- initial release

version 1.1 12.09.2011

- on page i, under key specifications, changed lens chief ray angle from "28.1" to "25"
- in section 10.2, updated figure 10-2 and table 10-1

version 2.0 03.29.2012

- changed datasheet to Product Specification
- on page i, under features, removed "and 160x100" from supports images sizes, under key specifications, replaced all TBDs, changed image area from "3876" to "3936" and added xshutdown to power requirements
- in table 1-1, changed description for DOGND from "...I/O or digital..." to "...I/O and digital..."
- in section 2.1, removed parentheses from "color"
- in section 2.2, changed "signals" to "clocks" and added "sequentially. In the time between..." to end of second paragraph
- in section 2, updated figure 2-1
- in table 2-1, changed header description from "technology" to "methodology"
- in table 2-2, changed bit descriptions for 0x3008 from "Bit[3]" to "Bit[4]" and 0x3006 from"Bit[1]" to "Bit[2]"
- in table 2-5, added footnote a
- in section 2, updated figures 2-7, 2-8, and 2-9
- in section 2.10, changed section description from "...which generates the system clock with 6..." to "...which generate the system clock from a 6..." and "frequency" to "frequencies"
- in table 2-10, changed PLL speed limitation for PLL1_VCO from "400~1000 MHz" to "300~110 MHz" and PLL2_VCO from "200~400 MHz" to "100~400 MHz"
- in section 3, in first paragraph changed "832 rows" to "820 rows", "(1,091,584 pixels)" to "(1,075,840 pixels)", "1,091,584 pixels" to "1,075,840 pixels" in second paragraph and updated figure 3-1
- in section 3.2, changed "binning" to "subsampling" and updated section description
- in section 4.5, updated section description
- in section 4, removed section 4.6
- in table 4-7, removed rows for registers 0x3204, 0x3205, 0x3206, and 0x3207
- in sub-section 4.6.1, changed "It" to "OV9713" in last sentence
- in sub-section 4.6.1.2, updated section description to "...For a longer strobe pulse, a number of..."
- in section 5, updated section descriptions to sections 5.1, 5.2, and 5.4

- in section 6.1, changed "HS" to "high speed (HS)" and added "data transmission" to second sentence in section description
- in table 6-1, changed bit descriptions for registers 0x4803, 0x4806, 0x4836, 0x484B, and 0x484C[6][3] to "not used"
- in table 6-2, renamed bit address for registers "0x4A01~0x4A07" to "0x4A02~0x4A08"
- in table 7-2, updated bit descriptions for registers 0x3006 and 0x3008 and added registers 0x309A~0x30A5, and removed "DVP" from register descriptions of 0x3013[1]
- in table 7-3, added "(every bit indicate..." to register descriptions of 0x3200~3203 and changed bit descriptions for 0x3204~0x3207 to "Debug Mode"
- in table 7-4, added "...which is smaller than 0x0B" to bit description of 0x3502
- in section 7, added "It is not recommended to change these register values" to section descriptions of 7.5, 7.6, 7.7
- in table 7-5, added registers 0x3636~0x3684
- in table 7-7, added registers 0x3823~0x3827
- in table 7-10, updated register names
- in table 7-13, changed bit descriptions for registers 0x4803, 0x4806, 0x4836, 0x484B, 0x484C[6][3] to "not used"
- in table 7-14, renamed bit address for registers "0x4A01~0x4A07" to "0x4A02~0x4A08"
- in table 8-3, replaced TBDs and changed digital inputs from "1.2" to "1.5" in header
- in section 10, updated figure 10-1 and added section 10.3

version 2.1 04.18.2012

- on page i, under applications, removed "and mobile"
- on page i, under key specifications changed sensitivity value from "2875" to "3300"
- in table 4-5, changed dump / program data n registers from "0x3D00-0x3D1F" to "0x3D00-0x3D0F" and added sidebar note

the clear advantage™

OmniVision Technologies, Inc.

UNITED STATES

4275 Burton Drive
Santa Clara, CA 95054

tel: + 1 408 567 3000

fax: + 1 408 567 3001

email: salesamerican@ovt.com

UNITED KINGDOM

Hampshire + 44 1256 744 610

FINLAND

Mouhijärvi + 358 3 341 1898

GERMANY

Munich +49 89 63 81 99 88

CHINA

Beijing + 86 10 6580 1690

Shanghai + 86 21 6175 9888

Shenzhen + 86 755 8384 9733

Hong Kong + 852 2403 4011

JAPAN

Tokyo + 81 3 5765 6321

KOREA

Seoul + 82 2 3478 2812

SINGAPORE + 65 6562 8250

TAIWAN

Taipei + 886 2 2657 9800 - ext.#100

website: **www.ovt.com**