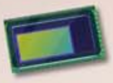
OmniVision.



OV2722/OV2222

datasheet

PRODUCT SPECIFICATION

1/5.8" CMOS 2 megapixel (1932 x 1092) image sensor
with OmniBSI+™ technology

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

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

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

Trademark Information

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

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

color CMOS 2 megapixel (1932 x 1092) image sensor with OmniBSI+™ technology

datasheet (CSP3)
PRODUCT SPECIFICATION

version 2.2
september 2012

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

applications

- ultra books
- smart phones
- notebook cameras
- tablet cameras
- handheld digital video
- cellular phones

ordering information

- **OV02722-A50A** (color, lead-free)
  50-pin CSP3
- **OV02222-A50A** (b&w, lead-free)
  50-pin CSP3

features

- 1.4 µm x 1.4 µm pixel with OmniBSI+ technology for high performance (high sensitivity, low crosstalk, low noise)
- optical size of 1/5.8"
- automatic image control functions: automatic exposure control (AEC), automatic white balance (AWB), automatic band filter (ABF), and automatic black level calibration (ABLC)
- programmable controls for frame rate, AEC/AGC 16-zone size/position/weight control, mirror and flip, cropping, windowing, and panning
- image quality controls: lens correction, 2-D defective pixel canceling
- support for output formats: 8-/10-bit raw RGB data
- support for video or snapshot operations
- support for horizontal and vertical sub-sampling
- standard serial SCCB interface
- digital video port (DVP) parallel output interface
- MIPI interface (two lanes)
- support for second camera interface
- 32 bytes of embedded one-time programmable (OTP) memory
- on-chip phase lock loop (PLL)
- embedded 1.5V regulator for core power
- programmable I/O drive capability, I/O tri-state configurability
- support for black sun cancellation
- suitable for module size of 8.5 x 8.5 x 6 mm

key specifications (typical)

- **active array size:** 1932 x 1092
- **power supply:**
  core: 1.5V ± 5% (with embedded 1.5V regulator)
  analog: 2.6 ~ 3.0V (2.8V typical)
  I/O: 1.7V ~ 3.0V
- **power requirements:**
  active: 74 mA
  standby: 15 µA
- **temperature range:**
  operating: -20°C to 70°C junction temperature (see table 8-2)
  stable image: 0°C to 50°C junction temperature (see table 8-2)
- **output formats:** 8-/10-bit RGB RAW output
- **lens size:** 1/5.8"
- **lens chief ray angle:** 28° (see figure 10-2)
- **input clock frequency:** 6~27 MHz
- **max S/N ratio:** 36 dB
- **dynamic range:** 70.4 dB @ 8x gain
- **maximum image transfer rate:**
  1080p (1920 x 1080): 30 fps
  720p: 60 fps
  VGA (640 x 480): 120 fps
  QVGA (320 x 240): 240 fps
- **sensitivity:** 700 mV/Lux-sec
- **shutter:** rolling shutter
- **maximum exposure interval:** 1118 x t_{ROW}
- **pixel size:** 1.4 µm x 1.4 µm
- **dark current:** 5mV/s @ 60°C junction temperature
- **image area:** 2732.8 µm x 1534.4 µm
- **package dimensions:** 5385 µm x 3285 µm

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 OV2722/OV2222 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	voltage
A1	SCL	input	SCCB interface input clock	–
A2	VH	reference	reference analog circuit	–
A3	DOGND	ground	digital I/O ground	–
A4	DVDD	power	digital logic power	1.5V
A5	AVDD	power	analog power	2.8V
A6	AGND	ground	analog ground	–
A7	DOGND	ground	digital I/O ground	–
A8	D2	I/O	video data output	–
A9	D1	I/O	video data output	–
A10	DOVDD	power	digital pad power	1.8V
B1	AVDD	power	analog power	2.8V
B2	DVDD	power	digital logic power	1.5V
B3	SDA	I/O	SCCB interface data	–
B4	VSYNC	I/O	video output vertical signal or video data input	–
B5	HREF	I/O	video output horizontal signal or video data input	–
B6	VN	reference	reference analog circuit	–
B7	DVDD	power	digital logic power	1.5V
B8	MDN1	output	MIPI data lane1 negative output	–
C1	AGND	ground	analog ground	–
C2	PWDN	input	power down (active high with internal pull down resistor)	–
C3	RESETB	input	hardware reset (active low with internal pull up resistor)	–
C7	D0	I/O	video data output	–
C8	MDP1	output	MIPI data lane1 positive output	–
C9	EVDD	power	MIPI power	1.5V

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

pin number	signal name	pin type	description	voltage
C10	MCN	output	MIPI clock negative output	–
D1	XVCLK	I/O	system input clock	–
D2	TM	input	test mode	–
D3	NC	–	no connect	–
D7	NC	–	no connect	–
D8	MDN2	output	MIPI data lane2 negative output	–
D9	EGND	power	ground for MIPI TX circuit	–
D10	MCP	output	MIPI clock positive output	–
E1	AGND	ground	analog ground	–
E2	AVDD	power	analog power	2.8V
E3	D9	I/O	video data output	–
E4	D7	I/O	video data output	–
E5	D5	I/O	video data output	–
E6	D3	I/O	video data output	–
E7	PVDD	power	PLL analog power	2.8V
E8	MDP2	output	MIPI data lane2 positive output	–
F1	DOVDD	power	digital pad power	1.8V
F2	DOGND	ground	digital I/O ground	–
F3	D8	I/O	video data output	–
F4	D6	I/O	video data output	–
F5	D4	I/O	video data output	–
F6	PCLK	I/O	video output clock	–
F7	DVDD	power	digital logic power	1.5V
F8	DOVDD	power	digital pad power	1.8V
F9	DOGND	ground	digital I/O ground	–
F10	PGND	ground	PLL analog ground	–

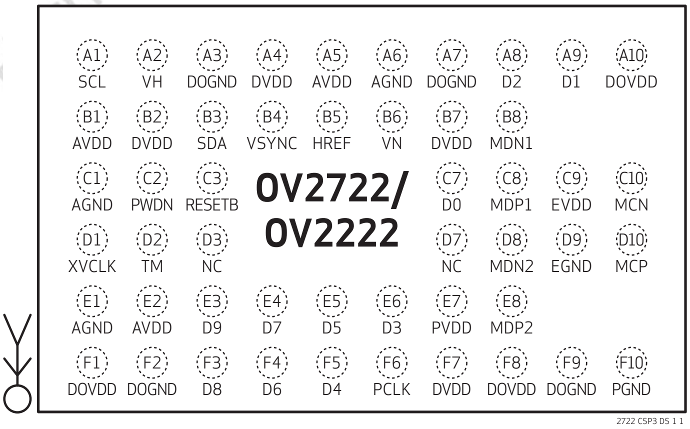
table 1-2 pin configuration under various conditions

signal	RESET[a]	RESET[b]	post-RESET	software sleep	hardware standby (power down pin = 1)
VSYNC	high-z	high-z	input by default (configurable)	high-z by default (configurable)	high-z by default (configurable)
HREF	high-z	high-z	input by default (configurable)	high-z by default (configurable)	high-z by default (configurable)
PCLK	high-z	high-z	input by default (configurable)	high-z by default (configurable)	high-z by default (configurable)
D[9:0]	high-z	high-z	input by default (configurable)	high-z by default (configurable)	high-z by default (configurable)
XVCLK	high-z	input	input	input	high-z
SDA	open drain	open drain	open drain	open drain	open drain
SCL	high-z	input	input	input	high-z
MCP	0	output	output	0	0
MCN	0	output	output	0	0
MDP1	high-z	high-z	output	high-z	high-z
MDN1	high-z	high-z	output	high-z	high-z
MDP2	high-z	high-z	output	high-z	high-z
MDN2	high-z	high-z	output	high-z	high-z

a. PWDN pin = 1 when chip power up

b. PWDN pin = 0 when chip power up

figure 1-1 pin diagram



A1 SCL	A2 VH	A3 DOGND	A4 DVDD	A5 AVDD	A6 AGND	A7 DOGND	A8 D2	A9 D1	A10 DOVDD
B1 AVDD	B2 DVDD	B3 SDA	B4 VSYNC	B5 HREF	B6 VN	B7 DVDD	B8 MDN1		
C1 AGND	C2 PWDN	C3 RESETB				C7 D0	C8 MDP1	C9 EVDD	C10 MCN
D1 XVCLK	D2 TM	D3 NC				D7 NC	D8 MDN2	D9 EGND	D10 MCP
E1 AGND	E2 AVDD	E3 D9	E4 D7	E5 D5	E6 D3	E7 PVDD	E8 MDP2		
F1 DOVDD	F2 DOGND	F3 D8	F4 D6	F5 D4	F6 PCLK	F7 DVDD	F8 DOVDD	F9 DOGND	F10 PGND

2 system level description

2.1 overview

The OV2722 (color) and OV2222 (b&w) are low voltage, high performance, 2 megapixel CMOS image sensors that provide 1920x1080 video output using OmniBSI+™ technology. They provide multiple resolution raw images via the control of the serial camera control bus.

The OV2722/OV2222 has an image array capable of operating up to 30 fps in 1920x1080 resolution with user control of image quality, data transfer, camera functions through the SCCB interface. The OV2722/OV2222 uses innovative OmniBSI+ technology to improve the sensor performance without the physical and optical trade-off.

The OV2722/OV2222 includes a one-time programmable (OTP) memory.

2.2 architecture

The OV2722/OV2222 sensor core generates streaming pixel data at a constant frame rate, indicated by HREF and VSYNC. **figure 2-1** shows the functional block of the OV2722/OV2222 image sensor. **figure 2-2** shows an example application of the OV2722/OV2222 sensor.

figure 2-1 OV2722/OV2222 block diagram

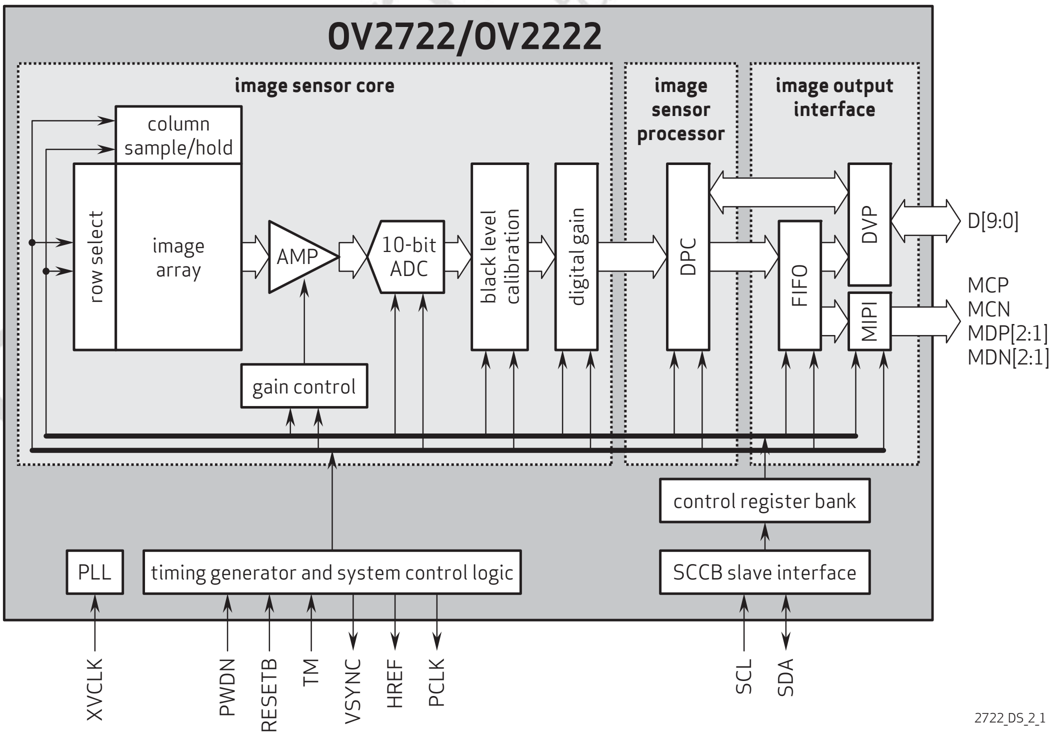
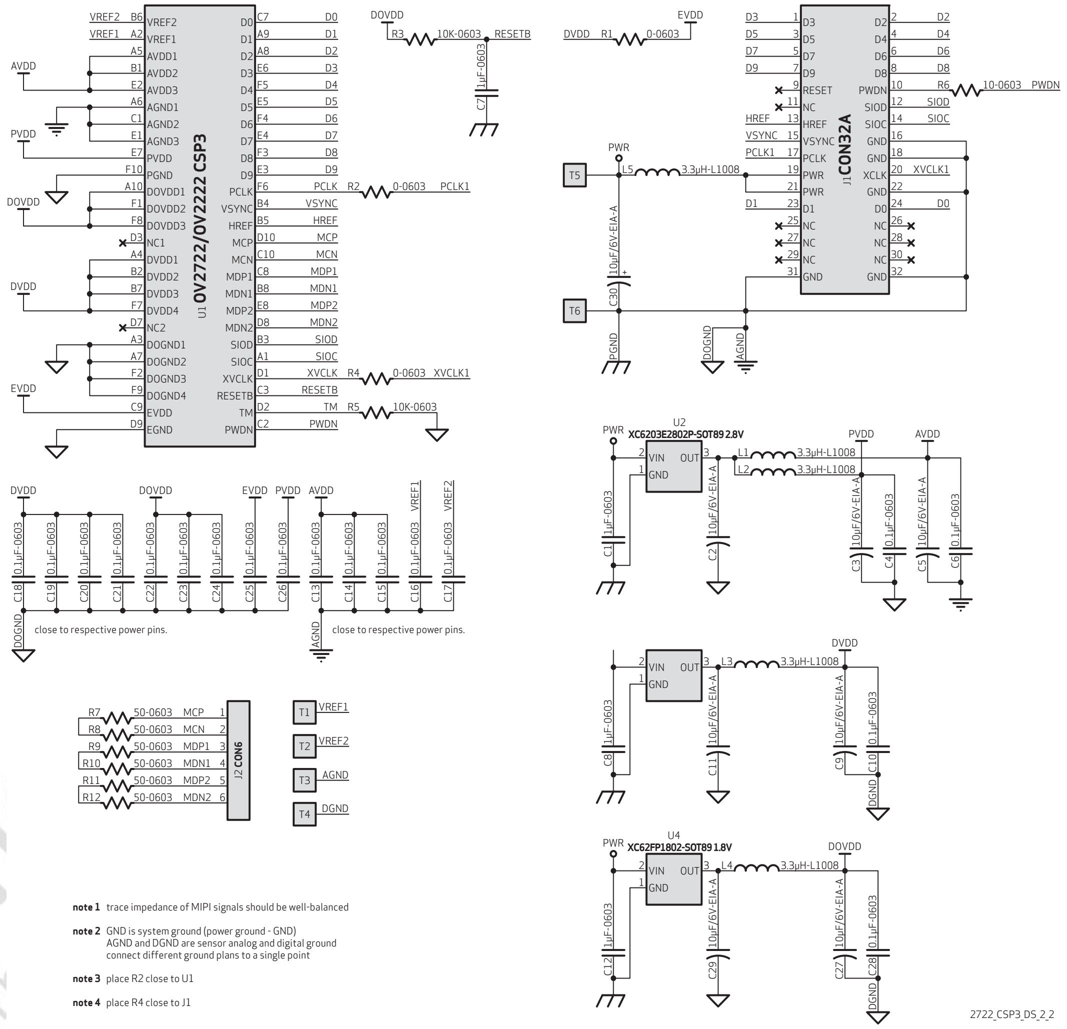


figure 2-2 reference design schematic



note 1 trace impedance of MIPI signals should be well-balanced

note 2 GND is system ground (power ground - GND)
AGND and DGND are sensor analog and digital ground
connect different ground plans to a single point

note 3 place R2 close to U1

note 4 place R4 close to J1

2.3 format and frame rate

table 2-1 format and frame rate

format	resolution	frame rate	methodology	pixel clock
1080p	1920 x 1080	30 fps	cropping	72 MHz
720p	1280 x 720	60 fps	cropping	72 MHz
VGA	640 x 480	90 fps	cropping or binning or skip	48 MHz
VGA	640 x 480	120 fps	cropping or skip	48 MHz
CIF	352 x 288	120 fps	cropping or binning or skip	24 MHz
QVGA	320 x 240	180 fps	cropping or binning or skip	24 MHz
QVGA	320 x 240	240 fps	cropping or skip	24 MHz

2.4 I/O control

2.4.1 system clock control

A PLL inside the chip generates a maximum 72 MHz system clock from a 6~27 MHz input clock. A programmable clock divider is provided to generate different frequencies.

2.5 power up sequence

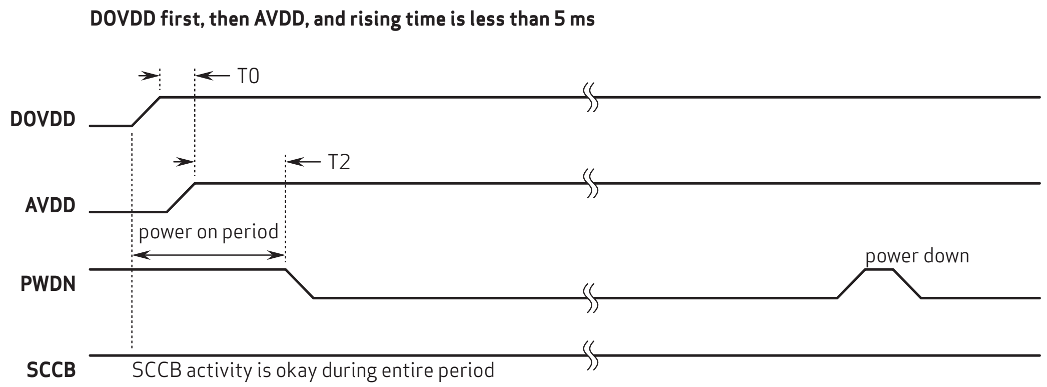
Based on the system power configuration (1.8V or 2.8V for I/O power, using external DVDD or internal DVDD), the power up sequence will differ.

2.5.1 power up with internal DVDD

For powering up with the internal DVDD and SCCB access during the power ON period, the following conditions must occur:

1. when DOVDD and AVDD are turned ON, make sure DOVDD becomes stable before AVDD becomes stable
2. PWDN is active high with an asynchronized design (does not need clock)
3. PWDN must be high during the power up period
4. for PWDN to go low, power must first become stable (AVDD to PWDN ≥ 5 ms)
5. RESETB is active low with an asynchronized design
6. state of RESETB does not matter during power up period once DOVDD is stable
7. master clock XVCLK should be provided at least 1 ms before host accesses the sensor’s registers
8. host can access SCCB bus (if shared) during the entire period. 20 ms after PWDN goes low or 20 ms after RESETB goes high if reset is inserted after PWDN goes low, host can access the sensor’s registers to initialize the sensor

figure 2-3 power up timing with internal DVDD


DOVDD first, then AVDD, and rising time is less than 5 ms
T0
DOVDD
T2
AVDD
power on period
power down
PWDN
SCCB
SCCB activity is okay during entire period

note T0 ≥ 0 ms: delay from DOVDD stable to AVDD stable
T2 ≥ 5 ms: delay from AVDD stable to sensor power up stable

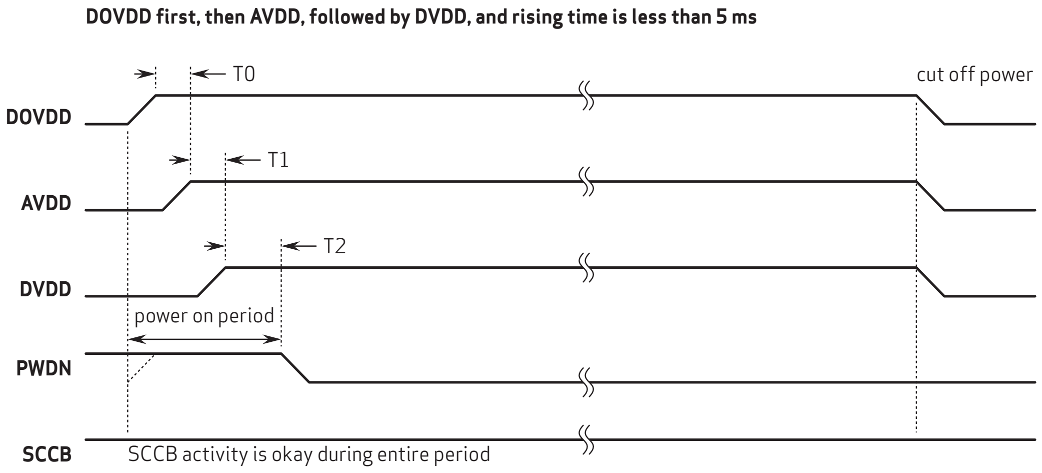
2722_DS_2_3

2.5.2 power up with external DVDD source

For powering up with an external DVDD source and SCCB access during the power ON period, the following conditions must occur:

1. when DOVDD and AVDD are turned ON, make sure DOVDD becomes stable before AVDD becomes stable
2. when AVDD and DVDD are turned ON, make sure AVDD becomes stable before DVDD becomes stable
3. PWDN is active high with an asynchronized design (does not need clock)
4. for PWDN to go low, power must first become stable (DVDD to PWDN ≥ 5 ms)
5. all power supplies are cut off when the camera is not in use (power down mode is not recommended)
6. RESETB is active low with an asynchronized design
7. state of RESETB does not matter during power up period once DOVDD is stable
8. master clock XVCLK should be provided at least 1 ms before host accesses the sensor’s registers
9. host can access SCCB bus (if shared) during entire period. 20 ms after PWDN goes low or 20 ms after RESETB goes high if reset is inserted after PWDN goes high, host can access the sensor’s registers to initialize the sensor

figure 2-4 power up timing with external DVDD source


DOVDD first, then AVDD, followed by DVDD, and rising time is less than 5 ms
T0
cut off power
DOVDD
T1
AVDD
T2
DVDD
power on period
PWDN
SCCB
SCCB activity is okay during entire period

note T0 ≥ 0 ms: delay from DOVDD stable to AVDD stable
T1 ≥ 0 ms: delay from AVDD stable to DVDD stable
T2 ≥ 5 ms: delay from DVDD stable to sensor power up stable

2722_DS_2_4

2.6 reset

Two reset modes are available for the OV2722/OV2222:

- hardware reset
- SCCB software reset

The OV2722/OV2222 sensor includes a **RESETB** pin that forces a complete hardware reset when it is pulled low (GND). The OV2722/OV2222 clears all registers and resets them to their default values when a hardware reset occurs.

The whole chip will be reset during power up. Manually applying a hard reset upon power up is recommended even though the on-chip power up reset is included. The hard reset is active low with an asynchronized design. The reset pulse width should be greater than or equal to 1 ms.

2.7 hardware and software standby

Two suspend modes are available for the OV2722/OV2222:

- hardware standby
- SCCB software standby

To initiate hardware standby mode, the **PWDN** pin must be tied to high. When this occurs, the OV2722/OV2222 internal device clock is halted and all internal counters are reset and registers are maintained. Executing a software standby (register bit 0x0100[0]) through the SCCB interface suspends internal circuit activity but does not halt the device clock. All register content is maintained in standby mode. To save power in MIPI mode, power down MIPI PHY by setting register 0x3018[4:3] to 2'b11 before sensor power down.

3 block level description

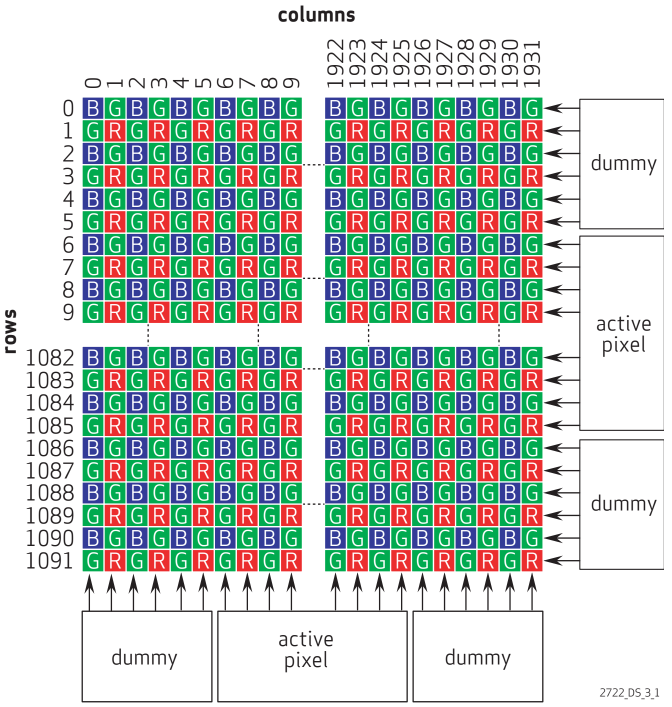
3.1 pixel array structure

The OV2722/OV2222 sensor has an image array of 1932 columns by 1092 rows (2,109,744 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 2,109,744 pixels, 2,073,600 (1920x1080) are active pixels and can be output. The other pixels are used for black level calibration and interpolation. The center 1920x1080 is suggested to be output from the whole active pixel array. The backend processor can use the boundary pixels for additional processing.

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

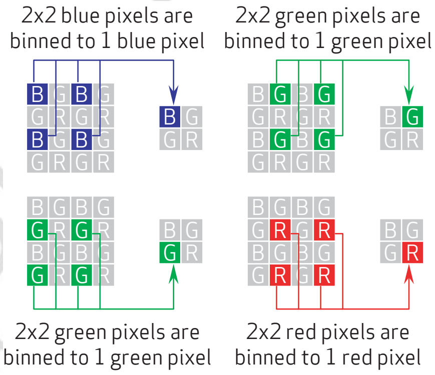
figure 3-1 sensor array region color filter layout



3.2 subsampling

There are two subsampling modes in the OV2722/OV2222: 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 OV2722/OV2222 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


2x2 blue pixels are binned to 1 blue pixel
2x2 green pixels are binned to 1 green pixel
2x2 green pixels are binned to 1 green pixel
2x2 red pixels are binned to 1 red pixel
B G R

2722_DS_3_2

Sensor timing adjustment is necessary after applying binning. Please consult your local OmniVision FAE for details.

table 3-1 horizontal and vertical binning registers

address	register name	default value	R/W	description
0x3820	TIMING_TC_REG20	0x40	RW	Bit[6]: Vertical binning[a] 0: Disable 1: Enable
0x3821	TIMING_TC_REG21	0x00	RW	Bit[0]: Horizontal binning[a] 0: Disable 1: Enable

[a] both vertical and horizontal are binning average

3.3 analog amplifier

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

3.4 10-bit A/D converters

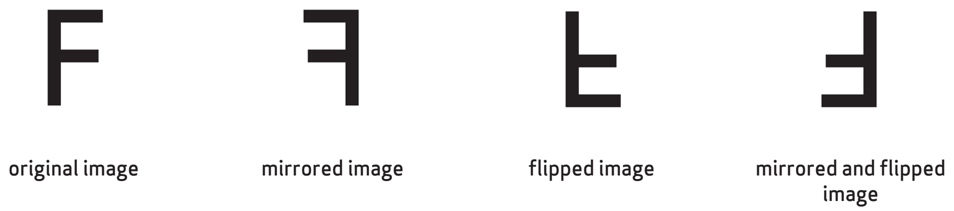
The balanced signal is then digitized by the on-chip 10-bit ADC. It can operate at up to 27 MHz and is fully synchronous to the pixel clock. The actual conversion rate is determined by the frame rate.

4 image sensor core digital functions

4.1 mirror and flip

The OV2722/OV2222 provides mirror and flip read-out modes, which respectively reverse the sensor data read-out order horizontally and vertically (see **figure 4-1**).

figure 4-1 mirror and flip samples



2722_DS_4_1

table 4-1 mirror flip control registers

address	register name	default value	R/W	description
0x3820	TIMING_TC_REG20	0x40	RW	Timing Control Bit[2]: r_vflip_isp Bit[1]: r_vflip_snr
0x3821	TIMING_TC_REG21	0x00	RW	Timing Control Bit[2]: r_mirror_isp Bit[1]: r_mirror_snr

4.2 image windowing

An image windowing area is defined by four parameters, x_addr_start, x_addr_end, y_addr_start, and y_addr_end. By properly setting the parameters, any portion or size within the sensor array can be defined as an visible area. Windowing is achieved by simply masking off the pixels outside of the defined window; thus, the original timing is not affected.

figure 4-2 frame structure

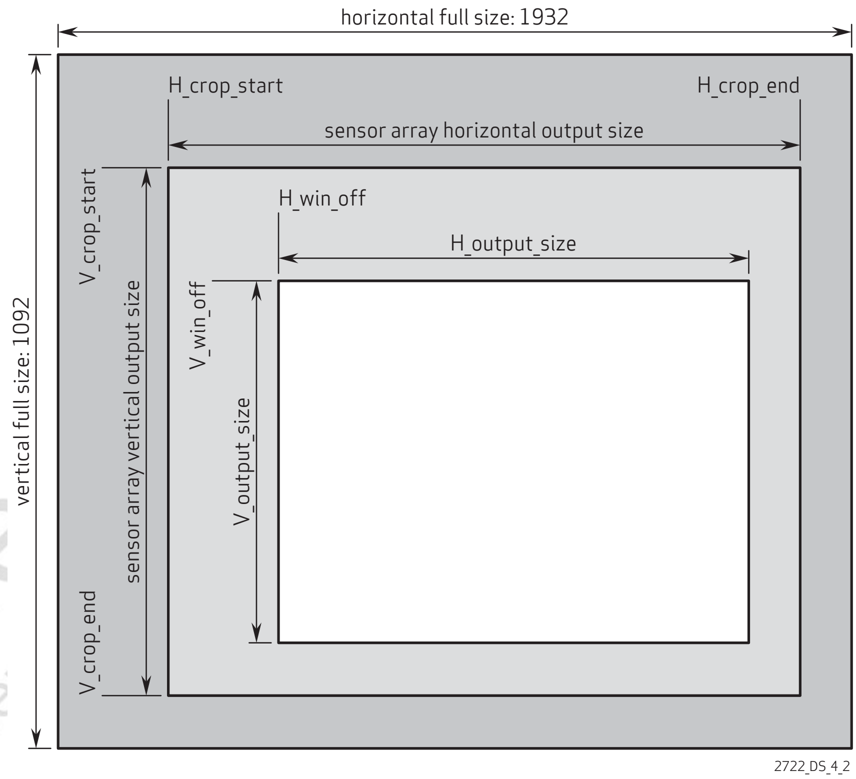


table 4-2 image windowing registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x3800	H_CROP_START	0x00	RW	Bit[3:0]: Horizontal crop start address[11:8]
0x3801	H_CROP_START	0x00	RW	Bit[7:0]: Horizontal crop start address[7:0]
0x3802	V_CROP_START	0x00	RW	Bit[3:0]: Vertical crop start address[11:8]
0x3803	V_CROP_START	0x00	RW	Bit[7:0]: Vertical crop start address[7:0]
0x3804	H_CROP_END	0x07	RW	Bit[3:0]: Horizontal crop end address[11:8]

table 4-2 image windowing registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x3805	H_CROP_END	0x8B	RW	Bit[7:0]: Horizontal crop end address[7:0]
0x3806	V_CROP_END	0x04	RW	Bit[3:0]: Vertical crop end address[11:8]
0x3807	V_CROP_END	0x43	RW	Bit[7:0]: Vertical crop end address[7:0]
0x3808	H_OUTPUT_SIZE	0x07	RW	Bit[3:0]: Horizontal output size[11:8]
0x3809	H_OUTPUT_SIZE	0x80	RW	Bit[7:0]: Horizontal output size[7:0]
0x380A	V_OUTPUT_SIZE	0x04	RW	Bit[3:0]: Vertical output size[11:8]
0x380B	V_OUTPUT_SIZE	0x38	RW	Bit[7:0]: Vertical output size[7:0]
0x380C	TIMING_HTS	0x08	RW	Bit[7:0]: Horizontal total size[15:8]
0x380D	TIMING_HTS	0x5C	RW	Bit[7:0]: Horizontal total size[7:0]
0x380E	TIMING_VTS	0x04	RW	Bit[7:0]: Vertical total size[15:8]
0x380F	TIMING_VTS	0x60	RW	Bit[7:0]: Vertical total size[7:0]
0x3810	H_WIN_OFF	0x00	RW	Bit[3:0]: Horizontal windowing offset[11:8]
0x3811	H_WIN_OFF	0x04	RW	Bit[7:0]: Horizontal windowing offset[7:0]
0x3812	V_WIN_OFF	0x00	RW	Bit[3:0]: Vertical windowing offset[11:8]
0x3813	V_WIN_OFF	0x02	RW	Bit[7:0]: Vertical windowing offset[7:0]
0x3814	H_SUB	0x11	RW	Bit[7:4]: Horizontal subsample odd increase number Bit[3:0]: Horizontal subsample even increase number
0x3815	V_SUB	0x11	RW	Bit[7:4]: Vertical subsample odd increase number Bit[3:0]: Vertical subsample even increase number

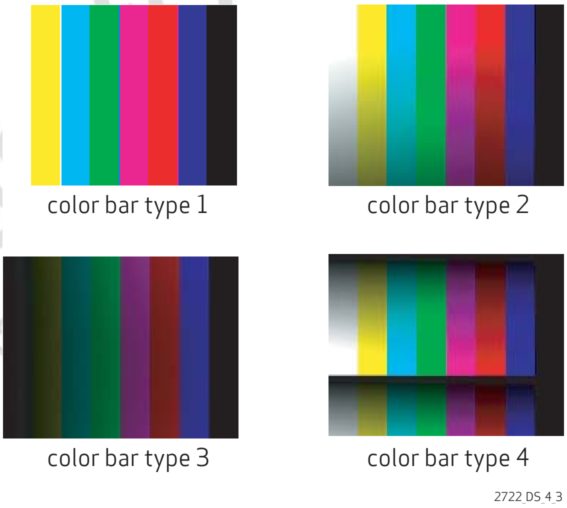
4.3 test pattern

For testing purposes, the OV2722/OV2222 offers one type of analog test pattern and three types of digital test patterns. The analog test pattern is a color bar that is controlled by register 0x370E[3]. The three types of digital test patterns are color bar, square and random data.The OV2722/OV2222 also offers two digital effects: transparent effect and rolling bar effect. The output type of test pattern is controlled by register 0x5040[1:0] (test_pattern_type). The digital test pattern function is controlled by register 0x5040[7].

4.3.1 color bar

There are four types of color bars shown in **figure 4-3**. The output type of color the color bar can be selected by bar style register 0x5040[3:2].

figure 4-3 color bar types


color bar type 1

color bar type 2

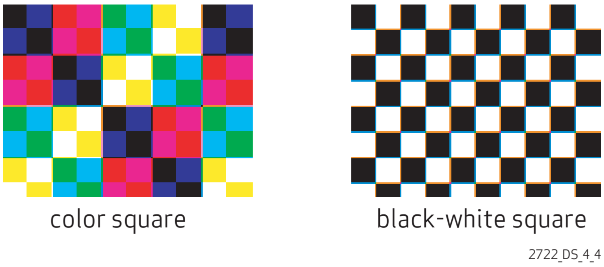
color bar type 3

color bar type 4

4.3.2 square

There are two types of square patterns: color square and black-white square. Register 0x5040[4] (square_mode) determines which type of square will be output (see **figure 4-4**).

figure 4-4 color, black and white square bars


color square

black-white square

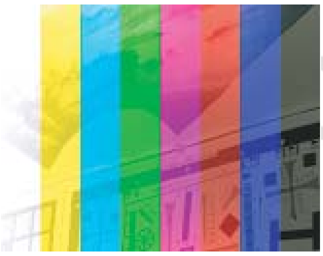
4.3.3 random data

There are two types of random data test patterns: frame-changing and frame-fixed random data. The output type of random data is determined by register 0x5041[4] (seed_same_en). The random seed is set by register 0x5041[3:0]. (rnd_seed).

4.3.4 transparent effect

The transparent effect is enabled by register 0x5040[5] (transparent_en). If this register is set, the transparent test pattern will be used. **figure 4-5** is an example which shows a transparent color bar image.

figure 4-5 transparent effect

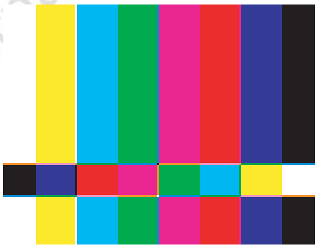


2722_DS_4_5

4.3.5 rolling bar effect

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

figure 4-6 rolling bar effect



2722_DS_4_6

table 4-3 test pattern registers

address	register name	default value	R/W	description
0x5040	PRE CTRL00	0x00	RW	Bit[7]: test_en 0: Disable test function 1: Enable test function Bit[6]: rolling_bar_en 0: Disable rolling bar 1: Enable rolling bar Bit[5]: transparent_en 0: Disable transparent effect 1: Enable transparent effect Bit[4]: square_mode 0: Color square pattern 1: Black-white square pattern 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
0x5041	PRE CTRL01	0x41	RW	Bit[7]: Not used Bit[6]: win_cut_en 0: Do not cut redundant pixels 1: Cut redundant pixels Bit[5]: two_lsb_0_en When this is set, two LSBs of output data are 0 Bit[4]: same_seed_en When this is set, the seed used to generate the random data are same as in seed register Bit[3:0]: Seed This is the seed used in generating random data
0x370E	SENSOR REGE	0x00	RW	Bit[3]: Cbar

4.4 AEC and AGC algorithms

The auto exposure control (AEC) and auto gain control (AGC) allows the image sensor to adjust the image brightness to a desired range by setting the proper exposure time and gain applied to the image. Besides automatic control, exposure time and gain can be set manually from external control. The related registers are listed in **table 4-4**

table 4-4 AEC/AGC control function registers

address	register name	default value	R/W	description
0x3500	AEC PK EXPO	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: Exposure[19:16]
0x3501	AEC PK EXPO	0x02	RW	Bit[7:0]: Exposure[15:8]
0x3502	AEC PK EXPO	0x00	RW	Bit[7:0]: Exposure[7:0]
0x3503	MANUAL CTRL	0x00	RW	Bit[7:6]: Reserved Bit[5]: Gain delay option 0: 1 frame latch 1: Delay 1 frame latch Bit[4]: Choose delay option 0: Delay disable 1: Delay enable Bit[3]: Reserved Bit[2]: VTS manual 0: Auto enable 1: Manual enable Bit[1]: AGC manual 0: Auto enable 1: Manual enable Bit[0]: AEC manual 0: Auto enable 1: Manual enable
0x3508	AGC ADJ	0x00	RW	Bit[1:0]: aec_real_gain[9:8] AEC/AGC real gain output high byte
0x3509	AGC ADJ	0x00	RW	Bit[7:0]: aec_real_gain[7:0] AEC/AGC real gain output low byte
0x350C	VTS DIFF	0x06	RW	Bit[7:0]: vts_diff[15:8] When in manual mode, set to 0x00
0x350D	VTS DIFF	0x18	RW	Bit[7:0]: vts_diff[7:0] When in manual mode, set to 0x00

4.4.1 average-based algorithm

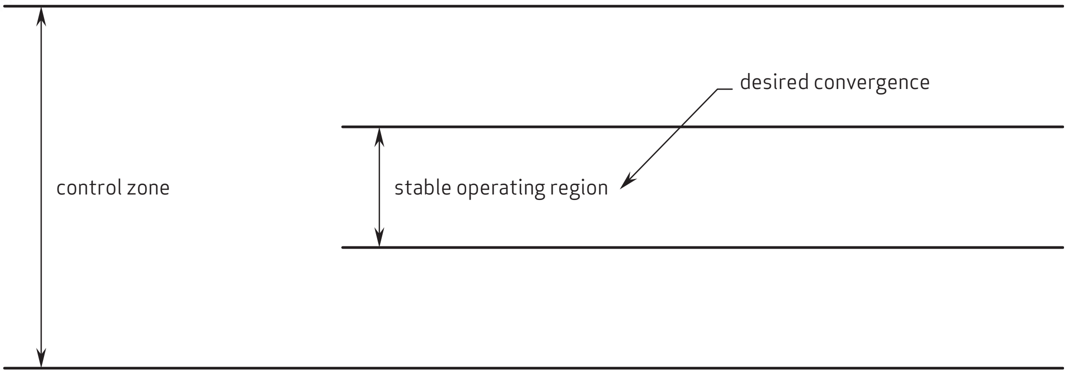
The average-based AEC controls image luminance using registers **WPT** (0x3A0F), **BPT** (0x3A10), **WPT2** (0x3A1B), and **BPT2** (0x3A1E). In average-based mode, the value of register **WPT** (0x3A0F) indicates the high threshold value for image change from unstable to stable state, and the value of register **BPT** (0x3A10) indicates the low threshold value for image change from unstable to stable state. The value of register **WPT2** (0x3A1B) indicates the high threshold value for image change from stable state to unstable state and the value of register **BPT2** (0x3A1E) indicates the low threshold value for image change from stable state to unstable state. When the target image luminance average value AVG (0x5693) is within the range specified by registers **WPT2** (0x3A1B) and **BPT2** (0x3A1E), the AEC keeps the image exposure and gain. When register AVG (0x5693) is greater than the value in register **WPT2** (0x3A1B), the AEC will decrease the image exposure and gain until it falls into the range of {0x3A10, 0x3A0F}. When register AVG (0x5693) is less than the value in register **BPT2** (0x3A1E), the AEC will increase the image exposure and gain until it falls into the range of {0x3A10, 0x3A0F}. Accordingly, the value in register **WPT** (0x3A0F) should be greater than the value in register **BPT** (0x3A10). The gap between the values of registers **WPT2** (0x3A1B) and **BPT2** (0x3A1E) controls the image stability.

The AEC function supports both manual and auto speed selections in order to bring the image exposure into the range set by the values in registers **WPT** (0x3A0F) and **BPT** (0x3A10). For manual mode, the speed supports both normal and fast speed selection. Setting the AEC to normal will allow for the slowest step increment or decrement in image exposure to maintain the specified range. Setting the AEC to fast mode will provide for an approximate ten-step increment or decrement in image exposure to maintain the specified range. For auto speed mode, the step will automatically be adjusted according to the difference between the target and present values. The auto ratio of steps can be set by register bits AEC CTRL05[4:0] (0x3A05); thus, the AEC speed can be adjusted automatically by the image average value or controlled manually.

Register HIGH VPT (0x3A11) and register LOW VPT (0x3A1F) controls the fast AEC range in manual speed mode. If the target image AVG {0x5690[1:0], 0x5691[7:0]} is greater than HIGH VPT (0x3A11), AEC will decrease by half. If register AVG (0x5693) is less than LOW VPT (0x3A1F), AEC will double.

As shown in **figure 4-7**, the AEC/AGC convergence uses two regions, the inner stable operating region and the outer control zone, which defines the convergence step size of fast and slow conditions.

figure 4-7 desired convergence



2722 DS 4 7

As for auto mode, the AEC will automatically calculate the steps needed based on the difference between the target and current values. So, the outer control zone is meaningless for this mode.

table 4-5 AEC/AGC algorithm function registers

address	register name	default value	R/W	description
0x3A0F	WPT	0x78	RW	Bit[7:0]: WPT Stable range high limit (enter)
0x3A10	BPT	0x68	RW	Bit[7:0]: BPT Stable range low limit (enter)
0x3A11	HIGH VPT	0xD0	RW	Bit[7:0]: vpt_high Fast zone high limit when step ratio auto mode is disabled
0x3A1B	WPT2	0x78	RW	Bit[7:0]: wpt2 Stable range high limit (from stable state to unstable state)
0x3A1E	BPT2	0x68	RW	Bit[7:0]: bpt2 Stable range low limit (from stable state to unstable state)
0x3A1F	LOW VPT	0x40	RW	Bit[7:0]: vpt_low Fast zone low limit when step ratio auto mode is disabled

For the average-based AEC/AGC algorithm, the measured window is horizontally and vertically adjustable and divided by sixteen (4x4) zones (see figure 4-8). Each zone (or block) is 1/16th of the image and has a 4-bit weight in calculating the average luminance (YAVG). The 4-bit weight could be n/16 where n is from 0 to 15. The final YAVG is the weighted average of the sixteen zones.

4.4.2 average luminance (YAVG)

Auto exposure time calculation is based on a frame brightness average value. By properly setting x_start, y_start, HW and YH as shown in **figure 4-8**, a 4x4 grid average window is defined. It will automatically divide each zone into 4x4 zones. The average value is the weighted average of the 16 sections. **table 4-6** lists the corresponding registers.

figure 4-8 average-based window definition

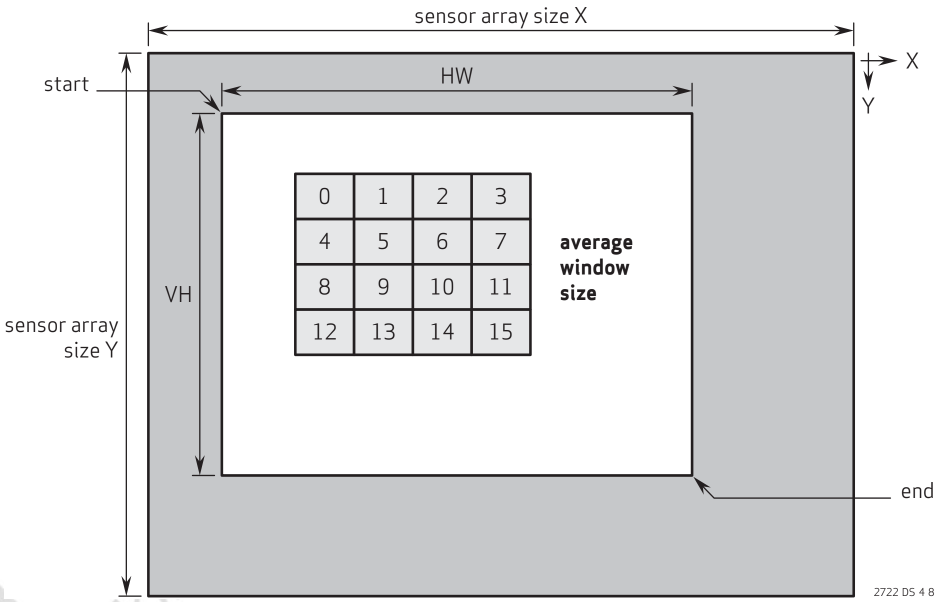


table 4-6 average luminance control function registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5680	X START	0x00	RW	Bit[7:3]: Reserved Bit[2:0]: x_start[10:8] Horizontal start position for average window high byte
0x5681	X START	0x00	RW	Bit[7:0]: x_start[7:0] Horizontal start position for average window low byte
0x5682	Y START	0x00	RW	Bit[7:3]: Reserved Bit[2:0]: y_start[10:8] Vertical start position for average window high byte

table 4-6 average luminance control function registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x5683	Y START	0x00	RW	Bit[7:0]: y_start[7:0] Vertical start position for average window low byte
0x5684	X WINDOW	0x10	RW	Bit[7:5]: Reserved Bit[4:0]: Window X[12:8]
0x5685	X WINDOW	0xA0	RW	Bit[7:0]: Window X[7:0]
0x5686	Y WINDOW	0x0C	RW	Bit[7:4]: Reserved Bit[3:0]: Window Y[11:8]
0x5687	Y WINDOW	0x78	RW	Bit[7:0]: Window Y[7:0]
0x5688	WEIGHT00	0x11	RW	Bit[7:4]: Window1 weight Bit[3:0]: Window0 weight
0x5689	WEIGHT01	0x11	RW	Bit[7:4]: Window3 weight Bit[3:0]: Window2 weight
0x568A	WEIGHT02	0x11	RW	Bit[7:4]: Window5 weight Bit[3:0]: Window4 weight
0x568B	WEIGHT03	0x11	RW	Bit[7:4]: Window7 weight Bit[3:0]: Window6 weight
0x568C	WEIGHT04	0x11	RW	Bit[7:4]: Window9 weight Bit[3:0]: Window8 weight
0x568D	WEIGHT05	0x11	RW	Bit[7:4]: Window11 weight Bit[3:0]: Window10 weight
0x568E	WEIGHT06	0x11	RW	Bit[7:4]: Window13 weight Bit[3:0]: Window12 weight
0x568F	WEIGHT07	0x11	RW	Bit[7:4]: Window15 weight Bit[3:0]: Window14 weight
0x5690	AVG CTRL10	0x02	RW	Bit[7:2]: Reserved 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 0: Auto average window 1: Manual average window
0x5693	AVG READOUT	–	R	Bit[7:0]: Avg value

4.5 AEC/AGC steps

The AEC and AGC work together to obtain optimum exposure/gain based on the current environmental illumination. In order to achieve the best signal-to-noise ratio (SNR), extending the exposure time is always preferred rather than raising the gain when the scene illumination is getting darker. Similarly, under brighter conditions, the action to decrease the gain is always taken prior to shortening the exposure time.

4.5.1 auto exposure control (AEC)

The function of the AEC is to calculate the necessary integration time of the next frame and send the information to the timing control block. In extremely bright situations, the LAEC activates, allowing integration to be less than one row. In extremely dark situations, the night mode activates, allowing the integration time to be larger than one frame.

To avoid image flickering under a periodic light source, the integration time can be adjusted in steps of integer multiples of the period of the light source.

4.5.2 banding mode ON with AEC

In Banding ON mode, the exposure time will fall in steps of integer multiples of the period of light intensity.

For a given light flickering frequency, the band step can be expressed in units of row period.

The band steps for 50Hz and 60Hz light sources can be set in registers {0x3A08[1:0], 0x3A09[7:0]} and {0x3A0A[1:0], 0x3A0B[7:0]}, respectively.

4.5.3 banding mode OFF with AEC

When banding mode is OFF, integration time increases/decreases as normal. It is not necessarily changed in multiples of band steps.

4.5.4 night mode

The OV2722 supports long integration time such as 1 frame, 2 frames, 3 frames, 4 frames, 5 frames, 6 frames, 7 frames, and 8 frames in dark conditions. This is achieved by slowing down the original frame rate and waiting for exposure. Night mode ceiling can be set in register bits 0x3A02[15:8], 0x3A03[7:0]. Night mode can be disabled by setting register bit 0x3A00[2] to 0. Also, when in night mode, the increase and decrease step can be based on band or frames, depending on register 0x3A05[6]. The minimum increase/decrease step can be one band. The step can be based both on bands and frames.

4.5.5 auto gain control (AGC)

Unlike prolonging integration time, increasing gain will amplify both signal and noise. Thus, AGC usually starts after AEC is full. However, in cases where adjacent AEC step changes are too large (>1/16), AGC steps should be inserted in between. The AGC ceiling can be set in {0x3A18[2:0], 0x3A19[7:0]}.

4.6 black level calibration (BLC)

The pixel array contains several optically shielded (black) lines. These lines are used as reference for black level calibration. There are three main functions of the BLC:

- adjusting all normal pixel values based on the values of the black levels
- applying multiplication to all pixel values based on digital gain

table 4-7 BLC control functions (sheet 1 of 2)

address	register name	default value	R/W	description
0x4000	BLC CTRL00	0x69	RW	Bit[7:2]: Not used Bit[1]: median_filter_en 0: Disable 1: Enable Bit[0]: BLC enable 0: Disable 1: Enable
0x4001	START LINE	0x00	RW	Bit[7:6]: Not used Bit[5:0]: start_line Specifies the starting black line for statistics
0x4002	BLC CTRL02	0x45	RW	Bit[7]: format_change_en 0: Disable format change BLC trigger 1: Enable format change BLC trigger Bit[6]: offset_auto_en 0: BLC offset from manual register 1: BLC offset from auto statistics Bit[5:0]: reset_frame_num Frame number BLC will redo after reset
0x4003	BLC CTRL03	0x01	RW	Bit[7]: manual_trig BLC will be triggered in manual_frame_num frames, which is counted from its rising edge continuously 0: Disable manual BLC trigger 1: Enable manual BLC trigger Bit[6]: freeze_en 0: BLC will be triggered normally 1: BLC will not be triggered Bit[5:0]: manual_frame_num Defines the frame number used in manual trigger mode and format change update mode

table 4-7 BLC control functions (sheet 2 of 2)

address	register name	default value	R/W	description
0x4004	LINE NUM	0x08	RW	Bit[7:6]: Not used Bit[5:0]: black_line_num Specifies the black line number used in calculating black line level. It should be an even number that is greater than zero
0x4005	BLC CTRL05	0x18	RW	Bit[7:4]: Not used Bit[3]: same_man_offset_en This is in effect when the offset_auto_en is 0. When it is enabled, the manual offsets used in the BLC algorithms are the same. They are all manual_offset00. 0: Disable 1: Enable Bit[2]: Not used Bit[1]: always_do When this is enabled, the BLC will always be updated 0: Disable 1: Enable Bit[0]: Not used
0x4007	BLC CTRL07	0x00	RW	Bit[7:5]: Not used Bit[4:3]: black_line_win_sel 00: Full image 01: Window excludes the first 16 pixels and the last 16 pixels 10: Window excludes the first 1/16 image and the last 1/16 image 11: Window excludes the first 1/8 image and the last 1/8 image Bit[2]: Not used Bit[1:0]: bypass_mode 00: Output limited data 01: Output low bits of input data 1x: Output high bits of input data
0x4009	BLC TARGET	0x10	RW	Bit[7:0]: Target

4.7 one-time programmable (OTP) memory

The OV2722/OV2222 has 256-bit embedded one time programmable (OTP) memory. The OTP memory can be programmed and read back via SCCB bus. This section provides general guidelines for programming and accessing the OTP memory.

4.7.1 OTP memory structure

Forty bits of OTP memory is reserved for OmniVision internal use. These bits are usually used to store the production information or used by the some internal functions. The remaining 216 bits are fully user programmable. The user can store production tracking information, camera module calibration data, etc. to these bits.

table 4-8 OTP memory structure

OTP bits	function
39 - 0	reserved by OmniVision for internal use
255 - 40	user programmable

4.7.2 accessing the OTP memory

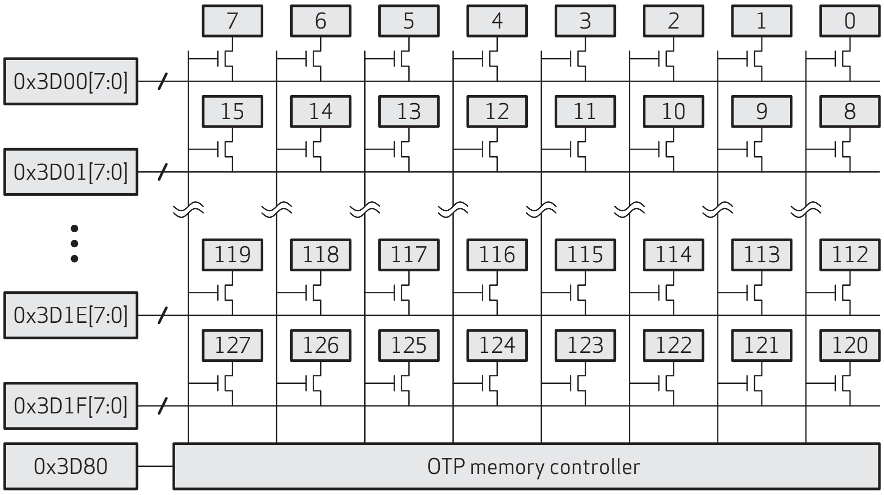
The OTP memory cannot be directly accessed. Instead, it is accessed through its register buffer 0x3D00~0x3D1F as shown in **figure 4-9**. Registers 0x3D80 and 0x3D81 are the command registers to program value to and read value back from OTP memory, respectively.

When 0x01 is written to 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. It is recommended to clear the register buffer to zero before loading the OTP.

When value 0x01 is written to register 0x3D80, the OTP memory controller will program the data of registers 0x3D00~0x3D1F to its corresponding OTP memory bits. Keep in mind, the memory is one time programmable. It cannot be programmed back to 0 once it is programmed to 1. In fact, the OTP memory controller only programs those bits with value 1 in its corresponding memory buffer when programming command is issued. Multi-pass programming is allowed. However, programming 1 to an OTP bit already programmed to 1in previous pass is prohibited. The user should always program a bit from 0 to 1 only in any programming pass.

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 high, accessing sensor register is prohibited in order to prevent any glitch on the power supply. It is recommended to wait 10ms after issuing the OTP read and program command. This delay should be scaled with the system clock period.

figure 4-9 OTP access



2722_DS_4_9

4.7.3 procedure for accessing OTP memory

Since the OTP memory can only be programmed once, the user should be very careful when accessing the OTP. Here is a detailed procedure for OTP access.

4.7.4 procedure to read OTP content

1. Make sure sensor is in active mode and set register 0x100 to 2'b01.
2. Clear software buffer which is to receive the OTP content.
3. Clear register buffer 0x3D00~0x3D1F to 0x00.
4. Set register 0x3D81 to 0x01.
5. Wait 10ms.
6. Read registers 0x3D00~0x3D1F and set to the software buffer.

The OTP read operation is performed to read back the information stored in OTP memory, verify the OTP memory is blank before programming data to it, or to verify OTP contents after programming data to it.

Verifying the OTP content at the last step of camera module testing is highly recommended in case the OTP content is accidently overwritten during the module testing.

4.7.5 procedure to program OTP content

1. Follow **procedure to read OTP content** to make sure the OTP to be programmed is blank.
2. Program the intended OTP content to its corresponding register buffer and clear the unused register buffer to 0.
   a. Registers 0x3D00~0x3D04 must be cleared to 0x00 before initiating the OTP programming command
3. Read back registers 0x3D00~0x3D1F to make sure they are the correct data to program to OTP memory or 0 for all other bits.
4. Write 0x01 to register 0x3D80 to initiate OTP programming.

5. Wait 10ms. Any register access during this period is prohibited.
6. Follow **procedure to read OTP content**.
7. Compare the OTP content read back to the intended OTP content.

4.7.6 power supply requirement for OTP memory programming

The OTP memory is programmed using the analog power. The AVDD voltage for OTP programming must be 2.5V ± 10%; otherwise, there will be reliability issues. The power supply should be able to provide an extra 50mA for OTP programming. There is no such limitation when reading OTP under normal operating conditions.

5 image sensor processor digital functions

5.1 DSP

The main purposes of the DSP includes:

- integrate all sub-modules
- create necessary control signals

table 5-1 DSP registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5000	ISP RW00	0xFF	RW	Bit[7]: avg_en Bit[6]: dpc_en Bit[5]: awb_stat_en Bit[4]: awb_gain_en Bit[3]: Black pixel correction enable Bit[2]: White pixel correction enable Bit[1]: LENC enable Bit[0]: isp_en
0x5001	ISP RW01	0xC1	RW	Bit[7]: v_en Bit[6]: h_en Bit[5]: avg_sel Bit[4]: dpc_cal_start_sel Bit[3]: eof_sel Bit[2]: sof_sel Bit[1]: fmt_sel Bit[0]: blc_en
0x5002	ISP RW02	0x05	RW	Bit[7]: isp_raw_en Bit[6:4]: win_yoff_adj Bit[3]: awb_bias_man_en Bit[2]: awb_bias_on Bit[1]: lenc_bias_man_en Bit[0]: lenc_bias_on
0x5003	ISP RW03	0x10	RW	Bit[7:0]: bias_man
0x5004	ISP RW04	0x10	RW	Bit[7]: gfirst_rvs Bit[6]: rblue_rvs Bit[5]: band50_flag Bit[4]: lenc_bias_plus Bit[3]: sram_test_dpc4 Bit[2]: sram_test_dpc3 Bit[1]: sram_test_dpc2 Bit[0]: sram_test_dpc1
0x5005	ISP RW05	0xAA	RW	Bit[7:4]: sram_rm_dpc2 Bit[3:0]: sram_rm_dpc1
0x5006	ISP RW06	0xAA	RW	Bit[7:4]: sram_rm_dpc4 Bit[3:0]: sram_rm_dpc3

table 5-1 DSP registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x5040	PRE 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
0x5041	PRE CTRL01	0x41	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 it is set, two LSB of output date are 0 Bit[4]: same_seed_en When it is set, the seed used to generate the random data are same which is set in seed register Bit[3:0]: seed It is the seed used in generating random data

5.2 digital gain correction (DGC)

The DGC module uses the input gain to adjust the value of the pixels.

table 5-2 DGC registers

address	register name	default value	R/W	description
0x5A00	DGC CTRL00	0x05	RW	Bit[7:3]: Not used Bit[2]: Debug control Bit[1]: dgc_man_en Bit[0]: dgc_en
0x5A01	NOT USED	–	–	Not Used
0x5A02	DGC CTRL02	0x02	RW	Bit[7:2]: Not used Bit[1:0]: dgc_man[9:8]
0x5A03	DGC CTRL03	0x00	RW	Bit[7:0]: dgc_man[7:0]

5.3 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. Additionally, LENC supports subsampling in both the horizontal and vertical directions. LENC is performed in the RGB domain.

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

address	register name	default value	R/W	description
0x5000	ISP RW00	1'b1	RW	Bit[1]: LENC enable 0: Disable 1: Enable
0x5800	LENC CTRL0	0x03	RW	Bit[7:3]: Not used Bit[2:0]: red_x0[10:8]
0x5801	LENC CTRL1	0xD0	RW	Bit[7:0]: red_x0[7:0]
0x5802	LENC CTRL2	0x02	RW	Bit[7:3]: Not used Bit[2:0]: red_y0[10:8]
0x5803	LENC CTRL3	0x56	RW	Bit[7:0]: red_y0[7:0]
0x5804	LENC CTRL4	0x22	RW	Bit[7]: Not used Bit[6:0]: red_a1

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

address	register name	default value	R/W	description
0x5805	LENC CTRL5	0x07	RW	Bit[7:4]: Not used Bit[3:0]: red_a2
0x5806	LENC CTRL6	0xC2	RW	Bit[7:0]: red_b1
0x5807	LENC CTRL7	0x08	RW	Bit[7:4]: Not used Bit[3:0]: red_b2
0x5808	LENC CTRL8	0x03	RW	Bit[7:3]: Not used Bit[2:0]: grn_x0[10:8]
0x5809	LENC CTRL9	0xD0	RW	Bit[7:0]: grn_x0[7:0]
0x580A	LENC CTRL10	0x02	RW	Bit[7:3]: Not used Bit[2:0]: grn_y0[10:8]
0x580B	LENC CTRL11	0x56	RW	Bit[7:0]: grn_y0[7:0]
0x580C	LENC CTRL12	0x22	RW	Bit[7]: Not used Bit[6:0]: grn_a1
0x580D	LENC CTRL13	0x07	RW	Bit[7:4]: Not used Bit[3:0]: grn_a2
0x580E	LENC CTRL14	0xC2	RW	Bit[7:0]: grn_b1
0x580F	LENC CTRL15	0x08	RW	Bit[7:4]: Not used Bit[3:0]: grn_b2
0x5810	LENC CTRL16	0x03	RW	Bit[7:3]: Not used Bit[2:0]: blu_x0[10:8]
0x5811	LENC CTRL17	0xD0	RW	Bit[7:0]: blu_x0[7:0]
0x5812	LENC CTRL18	0x02	RW	Bit[7:3]: Not used Bit[2:0]: blu_y0[10:8]
0x5813	LENC CTRL19	0x56	RW	Bit[7:0]: blu_y0[7:0]
0x5814	LENC CTRL20	0x22	RW	Bit[7]: Not used Bit[6:0]: blu_a1
0x5815	LENC CTRL21	0x07	RW	Bit[7:4]: Not used Bit[3:0]: blu_a2
0x5816	LENC CTRL22	0xC2	RW	Bit[7:0]: blu_b1
0x5817	LENC CTRL23	0x08	RW	Bit[7:4]: Not used Bit[3:0]: blu_b2

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

address	register name	default value	R/W	description
0x5818	LENC CTRL24	0x04	RW	Bit[7:4]: Not used Bit[3]: seed_reset_en Resets seed when VSYNC occurs Bit[2]: round_en Generates random round bit Bit[1]: coef_man_en Coefficient manual mode enable signal. When this is set, the coefficient used is coef_man Bit[0]: gain_coef_en 0: Disable 1: Enable
0x5819	LENC CTRL28	0x80	RW	Bit[7:0]: min_coefficient Range is [0,128]
0x581A	LENC CTRL29	0x06	RW	Bit[7:0]: low_gain_thresh Low real gain threshold
0x581B	LENC CTRL30	0x0C	RW	Bit[7:0]: high_gain_thresh High real gain threshold
0x581C	LENC CTRL31	0x80	RW	Bit[7:0]: coef_man Manual coefficient Range is [0,128]

5.4 auto white balance (AWB)

The purpose of the auto white balance (AWB) block is to avoid unrealistic colors so that objects that appear white to the human eye are rendered white in the final image or video. This image sensor supports both manual white balance and simple auto white balance. Simple AWB calculates the gain based on gray world assumptions.

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

address	register name	default value	R/W	description
0x5000	ISP RW00	2'b11	RW	Bit[5]: awb_stat_en Bit[4]: awb_gain_en
0x5180	AWB CTRL00	0x01	RW	Bit[7:4]: frame_num AWB will be updated every frame_num frames Bit[3]: Not used Bit[2]: freeze_en 0: Disable 1: Enable Bit[1]: gain_man_en 0: Auto mode: gain is calculated by AWB 1: Manual mode: gain is from AWB register Bit[0]: Not used
0x5181	AWB CTRL01	0xD8	RW	Bit[7:6]: fast_step_mode Fast mode gain step Bit[5:4]: diff_step Bit[3:0]: fast_awb_diff_limitation
0x5182	AWB CTRL02	0x24	RW	Bit[7:4]: stable_range Range to define the stable status Bit[3:0]: stable_wrange Wider range to determine the stable status
0x5183	AWB CTRL03	0xF0	RW	Bit[7:4]: red_gain_top_limit Red gain up limitation Bit[3:0]: red_gain_bot_limit Red gain down limitation
0x5184	AWB CTRL04	0xF0	RW	Bit[7:4]: green_gain_top_limit Green gain up limitation Bit[3:0]: green_gain_bot_limit Green gain down limitation
0x5185	AWB CTRL05	0xF0	RW	Bit[7:4]: blue_gain_top_limit Blue gain up limitation Bit[3:0]: blue_gain_bot_limit Blue gain down limitation

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

address	register name	default value	R/W	description
0x5186	AWB CTRL06	0x04	RW	Bit[7:4]: Not used Bit[3:0]: red_gain_man[11:8] Manual red gain input
0x5187	AWB CTRL07	0x00	RW	Bit[7:0]: red_gain_man[7:0] Manual red gain input
0x5188	AWB CTRL08	0x04	RW	Bit[7:4]: Not used Bit[3:0]: green_gain_man[11:8] Manual green gain input
0x5189	AWB CTRL09	0x00	RW	Bit[7:0]: green_gain_man[7:0] Manual green gain input
0x518A	AWB CTRL0A	0x04	RW	Bit[7:4]: Not used Bit[3:0]: blue_gain_man[11:8] Manual blue gain input
0x518B	AWB CTRL0B	0x00	RW	Bit[7:0]: blue_gain_man[7:0] Manual blue gain input

5.5 defect pixel cancellation (DPC)

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

table 5-5 DPC registers

address	register name	default value	R/W	description
0x5000	ISP RW00	0xFF	RW	Bit[3]: Black pixel correction enable 0: Disable 1: Enable Bit[2]: White pixel correction enable 0: Disable 1: Enable

5.6 windowing (WINC)

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

table 5-6 windowing registers

address	register name	default value	R/W	description
0x5980	WINC CTRL00	0x00	RW	Bit[7:3]: Not used Bit[2:0]: xstart[10:8] Start address in horizontal
0x5981	WINC CTRL01	0x00	RW	Bit[7:0]: xstart[7:0]
0x5982	WINC CTRL02	0x00	RW	Bit[7:3]: Not used Bit[2:0]: ystart[10:8] Start address in vertical
0x5983	WINC CTRL03	0x00	RW	Bit[7:0]: ystart[7:0]
0x5984	WINC CTRL04	0x10	RW	Bit[7:3]: Not used Bit[2:0]: x_win[10:8] Select whole zone width
0x5985	WINC CTRL05	0xA0	RW	Bit[7:0]: x_win[7:0]
0x5986	WINC CTRL06	0x0C	RW	Bit[7:3]: Not used Bit[2:0]: y_win[10:8] Select whole zone height
0x5987	WINC CTRL07	0x78	RW	Bit[7:0]: y_win[7:0]
0x5988	WINC CTRL08	0x00	RW	Bit[7:1]: Not used Bit[0]: win_man_en 0: Window size from window top 1: Window size from registers 0x00 to 0x07

5.7 average (AVG)

The main purpose of the AVG module is to calculate the luminance average. This average is for AEC/AGC algorithms.

table 5-7 AVG registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5680	AVG CTRL00	0x00	RW	Bit[7:3]: Not used Bit[2:0]: xstart_sub[10:8] AVG sub-window horizontal start position
0x5681	AVG CTRL01	0x00	RW	Bit[7:0]: xstart_sub[7:0] AVG sub-window horizontal start position
0x5682	AVG CTRL02	0x00	RW	Bit[7:3]: Not used Bit[2:0]: ystart_sub[10:8] AVG sub-window vertical start position
0x5683	AVG CTRL03	0x00	RW	Bit[7:0]: ystart_sub[7:0] AVG sub-window vertical start position
0x5684	AVG CTRL04	0x10	RW	Bit[7:3]: Not used Bit[2:0]: hsize_sub[10:8] Sub-window width
0x5685	AVG CTRL05	0xA0	RW	Bit[7:0]: hsize_sub[7:0] Sub-window width
0x5686	AVG CTRL06	0x0C	RW	Bit[7:3]: Not used Bit[2:0]: vsize_sub[10:8] Sub-window height
0x5687	AVG CTRL07	0x78	RW	Bit[7:0]: vsize_sub[7:0] Sub-window height
0x5688	AVG CTRL08	0x11	RW	Bit[7:4]: weight01 Weight of zone01 Bit[3:0]: weight00 Weight of zone00
0x5689	AVG CTRL09	0x11	RW	Bit[7:4]: weight03 Weight of zone03 Bit[3:0]: weight02 Weight of zone02
0x568A	AVG CTRL0A	0x11	RW	Bit[7:4]: weight11 Weight of zone11 Bit[3:0]: weight10 Weight of zone10
0x568B	AVG CTRL0B	0x11	RW	Bit[7:4]: weight13 Weight of zone13 Bit[3:0]: weight12 Weight of zone12

table 5-7 AVG registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x568C	AVG CTRL0C	0x11	RW	Bit[7:4]: weight21 Weight of zone 21 Bit[3:0]: weight20 Weight of zone20
0x568D	AVG CTRL0D	0x11	RW	Bit[7:4]: weight23 Weight of zone23 Bit[3:0]: weight22 Weight of zone22
0x568E	AVG CTRL0E	0x11	RW	Bit[7:4]: weight31 Weight of zone31 Bit[3:0]: weight30 Weight of zone30
0x568F	AVG CTRL0F	0x11	RW	Bit[7:4]: weight33 Weight of zone33 Bit[3:0]: weight32 Weight of zone32
0x5690	AVG CTRL10	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
0x5691	AVG ROREG0	–	R	Bit[7:0]: weight_sum Sum of weight
0x5692	AVG ROREG1	–	R	Bit[7:1]: Not used Bit[0]: avg_start AVG calculated indicating signal for SCCB read
0x5693	AVG ROREG2	–	R	Bit[7:0]: AVG High 8 bits of whole image AVG output

6 image sensor output interface digital functions

6.1 system control

System control registers include clock, reset control, and PLL configure. Individual modules can be reset or clock gated by setting the appropriate registers.

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

address	register name	default value	R/W	description
0x3000	SC_CMMN_PAD_OEN0	0x00	RW	Bit[1:0]: io_y_oen[9:8]
0x3001	SC_CMMN_PAD_OEN1	0x00	RW	Bit[7:0]: io_y_oen[7:0]
0x3002	SC_CMMN_PAD_OEN2	0x00	RW	Bit[7]: io_vsync_oen Bit[6]: io_href_oen Bit[5]: io_pclk_oen Bit[4:0]: Reserved
0x3005	SC_CMMN_CLKRST5	0xF0	RW	Bit[7:6]: Reserved Bit[5]: sclk_sram Bit[4]: sclk_syncfifo Bit[3:2]: Reserved Bit[1]: rst_sram Bit[0]: rst_syncfifo
0x3006	RSVD	–	–	Reserved
0x3007	SC_CMMN_PCLK_DIV_CTRL	0x01	RW	Bit[7]: pdiv_auto_mode 0: Use R_DIVM+1 as PCLK divider 1: Use pclk_ratio_i as PCLK divider Bit[6:0]: Reserved
0x3008	SC_CMMN_PAD_OUT0	0x00	RW	Bit[3:0]: io_y_o[9:8]
0x3009	SC_CMMN_PAD_OUT1	0x00	RW	Bit[7:0]: io_y_o[7:0]
0x300A	SC_CMMN_CHIP_ID	0x27	R	Chip ID High
0x300B	SC_CMMN_CHIP_ID	0x22	R	Chip ID Low
0x300C	SC_CMMN_SCCB_ID	0x6C	RW	SCCB ID
0x300D	SC_CMMN_PAD_OUT2	0x00	RW	Bit[7]: io_vsync_o Bit[6]: io_href_o Bit[5]: io_pclk_o Bit[4:0]: Reserved
0x300E	SC_CMMN_PAD_SEL0	0x00	RW	Bit[1:0]: io_y_sel[9:8]
0x300F	SC_CMMN_PAD_SEL1	0x00	RW	Bit[7:0]: io_y_sel[7:0]

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

address	register name	default value	R/W	description
0x3010	SC_CMMN_PAD_SEL2	0x00	RW	Bit[7]: io_vsync_sel Bit[6]: io_href_sel Bit[5]: io_pclk_sel Bit[4:0]: Reserved
0x3011	SC_CMMN_PAD_PK	0x02	RW	Bit[7]: pd_dato_en Bit[6:5]: Drive strength adjustment Bit[4:0]: Reserved
0x3012	RSVD	–	–	Reserved
0x3013	SC_CMMN_A_PWC_PK_O	0x00	RW	Bit[7]: sel_vdd Bit[6]: bp_reg_lp Bit[5]: bp_npump Bit[4]: bp_ppump Bit[3]: bp_regulator Bit[2]: pd_vmrst Bit[1]: pd_other Bit[0]: pd_bias
0x3014	SC_CMMN_A_PWC_PK_O	0x0B	RW	Bit[7]: vsun_sel_c Bit[6:4]: apd[2:0] Bit[3:0]: dio
0x3016	SC_CMMN_MIPI_PHY	0x00	RW	Bit[7:6]: lph Bit[5:4]: Reserved Bit[3]: mipi_pad_enable Bit[2]: pgm_bp_hs_en_lat Bypass the latch of hs_enable Bit[1:0: ictl[1:0] Bias current adjustment
0x3017	SC_CMMN_MIPI_PHY	0x10	RW	Bit[7:6]: pgm_vcm[1:0] High speed common mode voltage Bit[5:4]: pgm_lptx[1:0] 01: Driving strength of low speed transmitter Bit[3]: ihalf Bias current reduction Bit[2]: pgm_vicd CD input low voltage Bit[1]: pgm_vih CD input high voltage dummy Bit[0]: pgm_hs_valid Valid delay dummy

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

address	register name	default value	R/W	description
0x3018	SC_CMMN_MIPI_SC_CTRL	0x58	RW	Bit[7:5]: Chip debug Bit[4]: r_phy_pd_mipi 1: Power down PHY HS TX Bit[3]: r_phy_pd_lprx 1: Power down PHY LP RX module Bit[2]: mipi_en 0: DVP enable 1: MIPI enable Bit[1]: mipi_susp_reg MIPI system suspend register 1: Suspend Bit[0]: lane_dis_op 0: Use ~mipi_release1/2 and lane_disable1/2 to disable two data lanes 1: Use lane_disable1/2 to disable two data lanes
0x3019	SC_CMMN_MIPI_SC_CTRL	0x10	RW	Bit[7:0]: MIPI ULPS resume mark1 detect length
0x301A	SC_CMMN_CLKRST0	0xF0	RW	Bit[7]: sclk_gt Bit[6]: Reserved Bit[5]: sclk_aec Bit[4]: sclk_tc Bit[3]: mipi_phy_rst_o Bit[2]: Reserved Bit[1]: rst_aec Bit[0]: rst_tc
0x301B	SC_CMMN_CLKRST1	0xF0	RW	Bit[7]: sclk_blc Bit[6]: sclk_isp Bit[5]: sclk_avg Bit[4]: sclk_vfifo Bit[3]: rst_blc Bit[2]: rst_isp Bit[1]: rst_avg Bit[0]: rst_vfifo
0x301C	SC_CMMN_CLKRST2	0xF0	RW	Bit[7]: pclk_dvp Bit[6]: sclk_mipi Bit[5]: sclk_cif Bit[4]: sclk_OTP Bit[3]: rst_dvp Bit[2]: rst_mipi Bit[1]: rst_cif Bit[0]: rst_OTP

table 6-1 system control registers (sheet 4 of 6)

address	register name	default value	R/W	description
0x301D	SC_CMMN_CLKRST3	0xF4	RW	Bit[7]: Reserved Bit[6]: sclk_grp Bit[5]: sclk_bist Bit[4]: sclk_ac Bit[3]: Reserved Bit[2]: rst_grp Bit[1]: rst_bist Bit[0]: rst_ac
0x301E	SC_CMMN_CLKRST4	0xF0	RW	Bit[7]: Reserved Bit[6]: padclk_mipi_sc Bit[5]: pclk_vfifo Bit[4]: pclk_mipi Bit[3]: Reserved Bit[2]: rst_mipi_sc Bit[1]: rst_gt Bit[0]: Reserved
0x301F	RSVD	–	–	Reserved
0x3020	SC_CMMN_CLOCK_SEL	0x13	RW	Bit[6]: pclk_ratio_exp 1: Exponential pclk_ratio_i to 2^n Bit[5]: yuv_out_en 0: SOC sensor, output RAW data, or RAW sensor 1: SOC sensor, output YUV data Bit[4]: dvp_sclk_en 1: Use pll_sclk_i instead pll_pclk_i for DVP Bit[3]: pclk_sel Bit[2:1]: Reserved Bit[0]: sclk2x_sel

table 6-1 system control registers (sheet 5 of 6)

address	register name	default value	R/W	description
0x3021	SC_CMMN_MISC_CTRL	0x23	RW	Bit[5]: fst_stby_ctr 0: Software standby enter at v_blk 1: Software standby enter at l_blk Bit[4]: mipi_ctr_en 0: Disable the function 1: Enable MIPI remote reset and suspend control SC Bit[3]: mipi_rst_sel 0: MIPI remote reset all registers 1: MIPI remote reset all digital modules Bit[2]: gpio_pclk_en Bit[1]: frex_ef_sel Bit[0]: cen_global_o
0x3022	SC_CMMN_MIPI_SC_CTRL	0x00	RW	Bit[3]: lptx_ck_opt Bit[2]: pull_down_clk_lane Bit[1]: pull_down_data_lane2 Bit[0]: pull_down_data_lane1
0x302A	SC_CMMN_SUB_ID	–	R	Bit[7:4]: Process Bit[3:0]: Version
0x3034	SC_CMMN_PLL_CTRL0	0x1A	RW	Bit[6:4]: pll_charge_pump Bit[3:0]: mipi_bit_mode 0x8: 8-bit mode 0xA: 10-bit mode Others: Reserved for future use
0x3035	SC_CMMN_PLL_CTRL1	0x11	RW	Bit[7:4]: system_clk_div Slows down all clocks Bit[3:0]: scale_divider_mipi MIPI PCLK/SERCLK can be slowed down when image is scaled down
0x3036	SC_CMMN_PLL_MULTIPLIER	0x69	RW	Bit[6:0]: PLL_multiplier (4~129)
0x3037	SC_CMMN_PLL_CTR13	0x03	RW	Bit[7]: pll_root_div 0: Bypass 1: /2 Bit[6:4]: pll_prediv 1, 2, 3, 4, 6, 8 Bit[3]: Reserved Bit[2:0]: pll1_cp
0x3038	SC_CMMN_PLL_DEBUG_OPT	0x00	RW	Bit[1:0]: pll1_divb

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

address	register name	default value	R/W	description
0x3039	SC_CMMN_PLL_CTRL_R	0x00	RW	Bit[7]: pll_bypass Bit[6:0]: Reserved
0x303A	SC_CMMN_PLLS_CTRL0	0x00	RW	Bit[7]: plls_bypass Bit[6:0]: Reserved
0x303B	SC_CMMN_PLLS_CTRL1	0x19	RW	Bit[4:0]: plls_multiplier
0x303C	SC_CMMN_PLLS_CTRL2	0x11	RW	Bit[6:4]: plls_cp Bit[3:0]: plls_sys_div
0x303D	SC_CMMN_PLLS_CTRL3	0x30	RW	Bit[5:4]: plls_pre_div 00: /1 01: /1.5 10: /2 11: /3 Bit[2]: plls_div_r 0: /1 1: /2 Bit[1:0]: plls_seld5 00: /1 01: /1 10: /2 11: /2.5
0x3040	SC_SOC_CLKRST5	0xF0	RW	Bit[7]: sclk_isp_fc Bit[6]: sclk_fc Bit[5]: Reserved Bit[4]: sclk_fmt Bit[3]: rst_isp_fc Bit[2]: rst_fc Bit[1]: Reserved Bit[0]: rst_fmt
0x3041~ 0x3044	CHIP_DEBUG	–	–	Chip Debug

6.2 SCCB

table 6-2 SCCB registers

address	register name	default value	R/W	description
0x3100	SCCB CTRL	0x00	RW	Bit[3]: r_sda_dly_en Bit[2:0]: r_sda_dly
0x3101	SCCB OPT	0x12	RW	Bit[4]: en_ss_addr_inc Bit[3]: r_sda_byp_sync 0: Two clock stage sync for sda_i 1: No sync for sda_i Bit[2]: r_scl_byp_sync 0: Two clock stage sync for scl_i 1: No sync for scl_i Bit[1]: r_msk_glitch Bit[0]: r_msk_stop
0x3102	SCCB FILTER	0x00	RW	Bit[7:4]: r_sda_num Bit[3:0]: r_scl_num
0x3103	SCCB SYSREG	0x00	RW	Bit[6]: ctrl_rst_mipisc Bit[5]: ctrl_rst_srb Bit[4]: ctrl_rst_sccb_s Bit[3]: ctrl_rst_pon_sccb_s Bit[2]: ctrl_rst_clkmod Bit[1]: ctrl_rst_mipi_phy_rst_o Bit[0]: ctrl_pll_rst_o
0x3104	PWUP DIS	0x01	RW	Bit[4]: r_srb_clk_syn_en Bit[3]: pwup_dis2 Bit[2]: pwup_dis1 Bit[1]: pll_clk_sel Bit[0]: pwup_dis0
0x3105	PADCLK DIV	0x11	RW	Bit[5]: sclk use p_clk_i Bit[4]: Sleep enable Bit[3:0]: Padclk divider for SCCB
0x3106	SRB CTRL	0xF9	RW	Bit[7:4]: Reserved Bit[3:2]: PLL clock divider 00: pll_sclk 01: pll_sclk/2 10: pll_sclk/4 11: pll_sclk Bit[1]: rst_arb 1: Reset arbiter Bit[0]: sclk_arb 1: Enable sclk to arbiter

6.2.1 group hold

The OV2722/OV2222 supports group hold with up to four groups. Each group can have up to 16 registers.

Example settings:

```
6C 0x3208 0x00; Record Group 0 begin

6C 0x3503 0x03; register 1

6C 0x3501 0x7A; register 2

6C 0x3502 0xA0; register 3

6C 0x3208 0x10; Record Group 0 end

6C 0x3208 0xA0; write register group 0
```

table 6-3 group hold control registers

address	register name	default value	R/W	description
0x3200	GROUP ADR0	0x00	RW	Group0 Start Address in SRAM, actual address is {0x3200[3:0], 4'h0}
0x3201	GROUP ADR1	0x04	RW	Group1 Start Address in SRAM, actual address is {0x3201[3:0], 4'h0}
0x3202	GROUP ADR2	0x08	RW	Group2 Start Address in SRAM, actual address is {0x3202[3:0], 4'h0}
0x3203	GROUP ADR3	0x0B	RW	Group3 Start Address in SRAM, actual address is {0x3203[3:0], 4'h0}
0x3204	GROUP LEN0	–	R	Length of Group0
0x3205	GROUP LEN1	–	R	Length of Group1
0x3206	GROUP LEN2	–	R	Length of Group2
0x3207	GROUP LEN3	–	R	Length of Group3
0x3208	GROUP ACCESS	–	W	Bit[7:4]: Group_ctrl 0000: Group hold start 0001: Group hold end 1010: Group hold launch Others: Reserved Bit[3:0]: Group ID 0000: Group bank 1t 0001: Group bank 2 1010: Group bank 3 Others: Reserved
0x3209	SRAM TEST	0x0A	RW	Bit[4]: Group hold SRAM test enable Bit[3:0]: Group hold SRAM RM[3:0]

6.3 timing control

table 6-4 timing control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x3800	H_CROP_START	0x00	RW	Bit[3:0]: Horizontal crop start address[11:8]
0x3801	H_CROP_START	0x00	RW	Bit[7:0]: Horizontal crop start address[7:0]
0x3802	V_CROP_START	0x00	RW	Bit[3:0]: Vertical crop start address[11:8]
0x3803	V_CROP_START	0x00	RW	Bit[7:0]: Vertical crop start address[7:0]
0x3804	H_CROP_END	0x07	RW	Bit[3:0]: Horizontal crop end address[11:8]
0x3805	H_CROP_END	0x8B	RW	Bit[7:0]: Horizontal crop end address[7:0]
0x3806	V_CROP_END	0x04	RW	Bit[3:0]: Vertical crop end address[11:8]
0x3807	V_CROP_END	0x43	RW	Bit[7:0]: Vertical crop end address[7:0]
0x3808	H_OUTPUT_SIZE	0x07	RW	Bit[3:0]: Horizontal output size[11:8]
0x3809	H_OUTPUT_SIZE	0x80	RW	Bit[7:0]: Horizontal output size[7:0]
0x380A	V_OUTPUT_SIZE	0x04	RW	Bit[3:0]: Vertical output size[11:8]
0x380B	V_OUTPUT_SIZE	0x38	RW	Bit[7:0]: Vertical output size[7:0]
0x380C	TIMING_HTS	0x08	RW	Bit[7:0]: Horizontal total size[15:8]
0x380D	TIMING_HTS	0x5C	RW	Bit[7:0]: Horizontal total size[7:0]
0x380E	TIMING_VTS	0x04	RW	Bit[7:0]: Vertical total size[15:8]
0x380F	TIMING_VTS	0x60	RW	Bit[7:0]: Vertical total size[7:0]
0x3810	H_WIN_OFF	0x00	RW	Bit[3:0]: Horizontal windowing offset[11:8]
0x3811	H_WIN_OFF	0x04	RW	Bit[7:0]: Horizontal windowing offset[7:0]
0x3812	V_WIN_OFF	0x00	RW	Bit[3:0]: Vertical windowing offset[11:8]
0x3813	V_WIN_OFF	0x02	RW	Bit[7:0]: Vertical windowing offset[7:0]
0x3814	H_SUB	0x11	RW	Bit[7:4]: Horizontal subsample odd increase number Bit[3:0]: Horizontal subsample even increase number
0x3815	V_SUB	0x11	RW	Bit[7:4]: Vertical subsample odd increase number Bit[3:0]: Vertical subsample even increase number
0x3816	TIMING_HSYNCST	0x00	RW	Bit[3:0]: HSYNC start point[11:8]
0x3817	TIMING_HSYNCST	0x00	RW	Bit[7:0]: HSYNC start point[7:0]
0x3818	TIMING_HSYNCW	0x00	RW	Bit[3:0]: HSYNC window[11:8]

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

address	register name	default value	R/W	description
0x3819	TIMING_HSYNCW	0x00	RW	Bit[7:0]: HSYNC window[7:0]
0x3820	TIMING_TC_REG20	0x40	RW	Bit[7]: Reserved Bit[6]: Vbin enable Bit[5]: r_vsub48_alt Bit[4]: r_vflip_blc Bit[3]: Reserved Bit[2]: r_vflip_isp Bit[1]: r_vflip_snr Bit[0]: Chip debug
0x3821	TIMING_TC_REG21	0x00	RW	Bit[7]: r_hdr_en Bit[6]: r_thn_en Bit[5]: r_jpg_en Bit[4]: Reserved Bit[3]: r_arr_hskip Bit[2]: r_mirror_isp Bit[1]: r_mirror_snr Bit[0]: r_hbin
0x3822	TIMING_TC_REG22	0x10	RW	Bit[4:0]: r_ablc
0x3823	TIMING_TC_REG23	0x00	RW	Bit[0]: r_isp_bypass
0x3824	TIMING_TC_REG24	0x01	RW	Bit[4:0]: r_pclk_ratio_o
0x3825	TIMING_TC_REG25	0x00	RW	Bit[3]: r_fx1_o Bit[2:0]: r_expstb_sel_o
0x3826	TIMING_TC_REG26	0x03	RW	Bit[5:4]: r_vts_ctrl Bit[3]: r_dvpshp_en Bit[2]: r_rip_sof_en_o Bit[1:0]: r_grp_adj
0x3827	TIMING_TC_REG27	0x80	RW	Bit[7:0]: r_shpdly
0x3828	TIMING_TC_REG28	0x08	RW	Bit[3:0]: r_shldly
0x3829	TIMING_TC_REG29	0x00	RW	Bit[7:4]: r_vs_int Bit[3:0]: emb_line_adj
0x382A	TIMING_TC_REG2A	0x00	RW	Bit[1]: r_vsize_opt Bit[0]: r_hsize_opt
0x382B	TIMING_TC_REG2B	0x0B	RW	Bit[7:0]: hrefst_f
0x382C	TIMING_TC_REG2C	0x00	RW	Bit[7:0]: x_size_opt
0x3831	TIMING_REG31	0x0F	RW	Bit[7:4]: Reserved Bit[3:2]: r_vts_double Bit[1:0]: r_hts_half

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

address	register name	default value	R/W	description
0x3832	TIMING_REG32	0x01	RW	Bit[7:5]: Reserved Bit[4]: r_blc_tcsnr_opt Bit[3:2]: Reserved Bit[1:0]: r_blcnum_opt
0x3833	TIMING_REG33	0x00	RW	Bit[7:0]: href_size_opt
0x3834	TIMING_REG34	0x00	RW	Bit[7:0]: vref_size_opt

6.4 camera interface (CIF)

The camera interface (CIF) is used to receive external sensor image data through the DVP.

table 6-5 CIF control registers

address	register name	default value	R/W	description
0x4100	CIF CTRL 00	0x00	RW	Reverse External Sensor Input Polarity Bit[7]: Source select 0: From external DVP input 1: From MIPI receiver Bit[6]: End-of-frame (EOF) signal select Bit[5]: Reverse VSYNC polarity Bit[4]: Reverse HREF polarity Bit[3]: Reverse PCLK polarity Bit[2]: Reverse RBLUE polarity Bit[1]: Reverse ptn_RBLUE polarity Bit[0]: Reverse CIF RBLUE polarity

6.5 frame control (FC)

6.5.1 frame control for ISP input

Frame control for ISP input is used to mask some specified frame by setting the appropriate registers.

table 6-6 ISP input frame control registers

address	register name	default value	R/W	description
0x4200	ISP INPUT FRAME CTRL00	0x00	RW	Bit[2]: fcnt_eof_sel Bit[1]: fcnt_mask_dis Bit[0]: Frame counter reset
0x4201	ISP INPUT FRAME CTRL01	0x00	RW	Control Passed Frame Number Bit[3:0]: Frame ON number When both ON and OFF numbers are set to 0x00, frame control is in bypass mode
0x4202	ISP INPUT FRAME CTRL02	0x00	RW	Control Masked Frame Number Bit[3:0]: Frame OFF number When both ON and OFF numbers are set to 0x00, frame control is in bypass mode
0x4203	ISP INPUT FRAME CTRL03	0x00	RW	Bit[5]: data_mask_dis Bit[4]: valid_mask_dis Bit[3]: href_mask_dis Bit[2]: eof_mask_dis Bit[1]: sof_mask_dis Bit[0]: all_mask_dis

6.5.2 frame control for ISP output

Frame control after ISP output is used to mask some specified frame after ISP by setting the appropriate registers.

table 6-7 ISP output frame control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x4900	ISP OUTPUT FRAME CTRL01	0x00	RW	Bit[2]: fcnt_eof_sel Bit[1]: fcnt_mask_dis Bit[0]: Frame counter reset
0x4901	ISP OUTPUT FRAME CTRL01	0x00	RW	Control Passed Frame Number Bit[3:0]: Frame ON number When both ON and OFF numbers are set to 0x00, frame control is in bypass mode

table 6-7 ISP output frame control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x4902	ISP OUTPUT FRAME CTRL02	0x00	RW	Control Masked Frame Number Bit[3:0]: Frame OFF number When both ON and OFF numbers are set to 0x00, frame control is in bypass mode
0x4903	ISP OUTPUT FRAME CTRL03	0x00	RW	Bit[5]: data_mask_dis Bit[4]: valid_mask_dis Bit[3]: href_mask_dis Bit[2]: eof_mask_dis Bit[1]: sof_mask_dis Bit[0]: all_mask_dis

6.6 digital video port (DVP)

The digital video port (DVP) provides 10-bit parallel data output in all formats supported and extended features including compression mode, HSYNC mode, CCIR656 mode, and test pattern output. The DVP is also used to receive the video data from an external camera, which is sent out through the MIPI interface.

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

address	register name	default value	R/W	description
0x4302	FORMAT MAX	0x00	RW	Bit[1:0]: Clipping max[9:8]
0x4303	FORMAT MAX	0xFF	RW	Bit[7:0]: Clipping max[7:0]
0x4304	FORMAT MIN	0x00	RW	Bit[1:0]: Clipping min[1:0]
0x4305	FORMAT MIN	0x00	RW	Bit[7:0]: Clipping min[7:0]
0x4700	DVP MODE SELECT	0x04	RW	Bit[3]: CCIR v select Bit[2]: CCIR f select Bit[1]: CCIR656 mode enable Bit[0]: HSYNC mode enable
0x4701	DVP VSYNC WIDTH CONTRL	0x01	RW	Bit[7:0]: VSYNC width (in terms of number of lines)
0x4702	DVP_HSYVSY_NEG_WIDTH	0x01	RW	Bit[7:0]: VSYNC length[15:8] (in terms of pixel count)
0x4703	DVP_HSYVSY_NEG_WIDTH	0x00	RW	Bit[7:0]: VSYNC length[7:0] (in terms of pixel count)
0x4704	DVP VSYNC MODE	0x00	RW	Bit[3:2]: r_vsyncount_sel Bit[1]: r_vsync3_mod Bit[0]: r_vsync2_mod

table 6-8 DVP control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x4705	DVP_EOF_VSYNC DELAY	0x00	RW	Bit[7:0]: eof_vsync_delay[23:16] SOF/EOF negative edge to VSYNC positive edge delay
0x4706	DVP_EOF_VSYNC DELAY	0x00	RW	Bit[7:0]: eof_vsync_delay[15:8] SOF/EOF negative edge to VSYNC positive edge delay
0x4707	DVP_EOF_VSYNC DELAY	0x00	RW	Bit[7:0]: eof_vsync_delay[7:0] SOF/EOF negative edge to VSYNC positive edge delay
0x4708	DVP_POL_CTRL	0x01	RW	Bit[7]: Clock DDR mode enable Bit[5]: VSYNC gated clock enable Bit[4]: HREF gated clock enable Bit[3]: No first for FIFO Bit[2]: HREF polarity reverse Bit[1]: VSYNC polarity reverse Bit[0]: PCLK polarity reverse
0x4709	BIT_TEST_PATTERN	0x00	RW	Bit[7]: FIFO bypass mode Bit[6:4]: Data bit swap Bit[3]: Bit test mode Bit[2]: 10-bit bit test Bit[1]: 8-bit bit test Bit[0]: Bit test enable
0x470A	DVP_BYP_CTRL	0x00	RW	Bit[7:0]: Bypass control[15:8]
0x470B	DVP_BYP_CTRL	0x00	RW	Bit[7:0]: Bypass control[7:0]
0x470C	DVP_BYP_SEL	0x00	RW	Bit[4]: HREF select Bit[3:0]: Bypass select

6.6.1 DVP timing

figure 6-1 DVP timing diagram

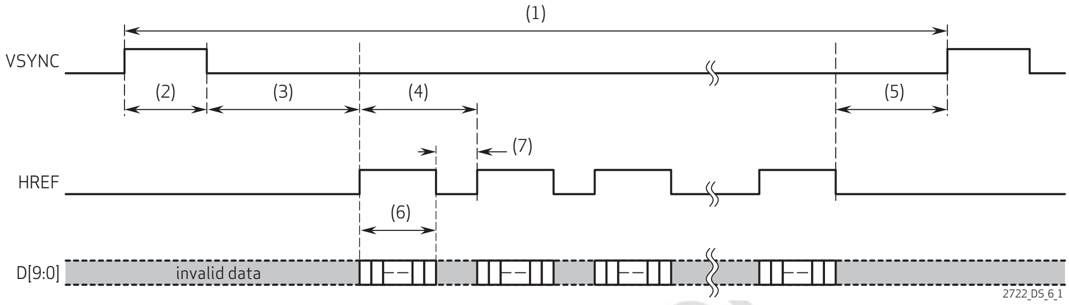


table 6-9 DVP timing specifications

mode	timing
1080p 1920x1080	(1) 2396800 tp (2140x1120) (2) 2394 tp (3) 23318 tp (4) 2140 tp (5) 59819 tp (6) 1920 tp (7) 220 tp
720p 1280x720	(1) 1200192 tp (1596x752) (2) 2398 tp (3) 22084 tp (4) 1536 tp (5) 833126 tp (6) 1280 tp (7) 876 tp
VGA 640x480	(1) 1533504 tp (1042x512) (2) 1298 tp (3) 7512 tp (4) 1042 tp (5) 26234 tp (6) 640 tp (7) 402 tp
QVGA 320x240	(1) 133000 tp (532x250) (2) 788 tp (3) 2677 tp (4) 532 tp (5) 2067 tp (6) 320 tp (7) 212 tp
CIF 352x288	(1) 199980 tp (606x330) (2) 862 tp (3) 4924 tp (4) 606 tp (5) 19922 tp (6) 352 tp (7) 354 tp

note
Timing values shown in **table 6-9** may vary depending upon register settings.

6.7 mobile industry processor interface (MIPI)

The OV2722/OV2222 MIPI interface provides a single uni-directional clock lane and two uni-directional data lanes to communicate to components in a mobile device. The data lanes have full support for high speed (HS) and low power (LP) data transfer modes. Contact your local OmniVision FAE for more details.

table 6-10 MIPI transmitter registers (sheet 1 of 8)

address	register name	default value	R/W	description
0x4800	MIPI CTRL 00	0x04	RW	MIPI Control 00 Bit[7]: mipi_hs_only 0: MIPI can support CD and ESCAPE mode 1: MIPI always in high speed mode Bit[6]: ck_mark1_en 1: Enable clock lane mark1 when resume Bit[5]: Clock lane gate enable 0: Clock lane is free running 1: Gate clock lane when no packet to transmit Bit[4]: Line sync enable 0: Do not send line short packet for each line 1: Send line short packet for each line Bit[3]: Lane select 0: Use lane1 as default data lane 1: Use lane2 as default data lane Bit[2]: Idle status 0: MIPI bus will be LP00 when no packet to transmit 1: MIPI bus will be LP11 when no packet to transmit Bit[1]: Clock lane first bits 0: Output 0x55 1: Output 0xAA Bit[0]: Clock lane disable 1: Manually set clock lane to low power mode

table 6-10 MIPI transmitter registers (sheet 2 of 8)

address	register name	default value	R/W	description
0x4801	MIPI CTRL 01	0x0F	RW	MIPI Control 01 Bit[7]: Long packet data type manual enable 0: Use mipi_dt 1: Use dt_man_o as long packet data (see register **0x4814**[5:0]) Bit[6]: Short packet data type manual enable 1: Use dt_spkt as short packet data (see register **0x4815**[5:0]) Bit[5]: Short packet word counter manual enable 0: Use frame counter or line counter 1: Select spkt_wc_reg_o (see {0x4812, 0x4813}) Bit[4]: PH bit order for ECC 0: {DI[7:0],WC[7:0],WC[15:8]} 1: {DI[0:7],WC[0:7],WC[8:15]} Bit[3]: PH byte order for ECC 0: {DI,WC_l,WC_h} 1: {DI,WC_h,WC_l} Bit[2]: PH byte order2 for ECC 0: {DI,WC} 1: {WC,DI} Bit[1]: mark1_en1 1: After each reset release, lane 1 should send mark1 for wkup_dly_o when mipi_sys_susp = 1 Bit[0]: mark1_en2 1: After each reset release, lane 2 should send mark1 for wkup_dly_o when mipi_sys_susp = 1

table 6-10 MIPI transmitter registers (sheet 3 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	MIPI CTRL 03	0x50	RW	MIPI Control 03 Bit[7:6]: lp_glitch_nu 0: Use 2d of lp_in 1: Mask one sclk cycle glitch of lp_in Bit[5:4]: cd_glitch_nu 0: Use 2d of lp_cd_in 1: Mask one sclk cycle glitch of lp_cd_in Bit[3]: Enable CD plus of data lane1 0: Disable 1: Enable Bit[2]: Enable CD plus of data lane2 0: Disable 1: Enable Bit[1]: Enable CD of data_lane1 from PHY 0: Disable 1: Enable Bit[0]: Enable CD of data_lane2 from PHY 0: Disable 1: Enable

table 6-10 MIPI transmitter registers (sheet 4 of 8)

address	register name	default value	R/W	description
0x4804	MIPI CTRL 04	0x8D	RW	MIPI Control 04 Bit[7]: wait_pkt_end 1: Wait HS packet end when send UL command Bit[6]: tx_lsb_first 0: lp_tx and lp_rx high bit first 1: Low power transmit low bit first Bit[5]: dir_recover_sel 0: Auto change to output only when turnaround command 1: Auto change to output when LP11 and GPIO is output Bit[4]: mipi_reg_en 0: Disable MIPI_REG_P to access registers, LP data will write to VFIFO 1: Enable MIPI_REG_P to access registers Bit[3]: Address read/write register will auto add 1 0: Disable 1: Enable Bit[2]: LP TX lane select 0: Select lane1 to transmit LP data 1: Select lane2 to transmit LP data Bit[1]: wr_first_byte 1: lp_rx will write first byte (command byte) to RAM Bit[0]: rd_ta_en 1: Send turnaround command after sending register read data

table 6-10 MIPI transmitter registers (sheet 5 of 8)

address	register name	default value	R/W	description
0x4805	MIPI CTRL 05	0x10	RW	MIPI Control 05 Bit[7]: MIPI lane1 disable 1: Disable MIPI data lane1, lane1 will be LP00 Bit[6]: MIPI lane2 disable 1: Disable MIPI data lane2, lane2 will be LP00 Bit[5]: lpx_p_sel 0: Auto calculate t_lpx_o in pclk2x domain, unit pclk2x 1: Use lp_p_min[7:0] Bit[4]: lp_rx_intr_sel 0: Send lp_rx_intr_o at the first byte 1: Send lp_rx_intr_o at the end of receiving Bit[3]: cd_tst_sel 1: Select PHY test pins Bit[2]: mipi_reg_mask 1: Disable MIPI access SRB Bit[1]: clip enable Bit[0]: hd_sk_en 0: Disable MIPI and MCU handshake registers 1: Disable MIPI and MCU handshake registers
0x4806	MIPI REG RW CTRL	0x28	RW	Bit[7]: prbs_en Test mode Bit[6]: mipi_test Bit[5]: mipi_lp_op 0: Use new option to reduce mipi_lptx_p Bit[4]: two_lane_man_en 1: Use two_lane_man to manually control two_lane_mode Bit[3]: two_lane_man Bit[2]: rst_rtn_en 1: Change to input to allow host RW register after reset Bit[1]: frame_end_en 1: After frame end packet, change to input to allow host RW register Bit[0]: line_end_en 1: After line end packet, change to input to allow host RW register

0x480A	MIPI BIT ORDER	0x00	RW	Bit[2]: Bit order reverse
				Bit[1:0]: Bit position adjustment
				01: {D[7:0],D[9:8]}
				10: {D[1:0],D[9:2]}

table 6-10 MIPI transmitter registers (sheet 6 of 8)

address	register name	default value	R/W	description
0x4810	MIPI MAX FRAME COUNT	0xFF	RW	High Byte of Maximum Frame Count of Frame Sync Short Packet
0x4811	MIPI MAX FRAME COUNT	0xFF	RW	Low Byte of Maximum Frame Count of Frame Sync Short Packet
0x4814	MIPI CTRL14	0x2A	RW	MIPI Control 14 Bit[7:6]: Virtual channel of MIPI Bit[5:0]: Data type in manual mode
0x4815	MIPI_DT_SPKT	0x00	RW	Bit[6]: pclk_div 0: Use rising edge of mipi_pclk_o to generate MIPI bus to PHY 1: Use falling edge of mipi_pclk_o to generate MIPI bus to PHY Bit[5:0]: Manual data type for short packet
0x4818	HS_ZERO_MIN	0x00	RW	High Byte of Minimum Value for hs_zero Unit ns
0x4819	HS_ZERO_MIN	0x96	RW	Low Byte of Minimum Value for hs_zero, unit ns hs_zero_real = hs_zero_min_o + Tui*ui_hs_zero_min_o
0x481A	HS_TRAIL_MIN	0x00	RW	High Byte of Minimum Value for hs_trail, unit ns
0x481B	HS_TRAIL_MIN	0x3C	RW	Low Byte of Minimum Value for hs_trail, hs_trail_real = hs_trail_min_o + Tui*ui_hs_trail_min_o
0x481C	CLK_ZERO_MIN	0x01	RW	High Byte of Minimum Value for clk_zero, unit ns
0x481D	CLK_ZERO_MIN	0x86	RW	Low Byte of Minimum Value for clk_zero, clk_zero_real = clk_zero_min_o + Tui*ui_clk_zero_min_o
0x481E	CLK_PREPARE_MIN	0x00	RW	High Byte of Minimum Value for clk_prepare, unit ns Bit[1:0]: clk_prepare_min[9:8]
0x481F	CLK_PREPARE_MIN	0x3C	RW	Low Byte of Minimum Value for clk_prepare clk_prepare_real = clk_prepare_min_o + Tui*ui_clk_prepare_min_o
0x4820	CLK_POST_MIN	0x00	RW	High Byte of Minimum Value for clk_post, unit ns Bit[1:0]: clk_post_min[9:8]
0x4821	CLK_POST_MIN	0x56	RW	Low Byte of Minimum Value for clk_post clk_post_real = clk_post_min_o + Tui*ui_clk_post_min_o
0x4822	CLK_TRAIL_MIN	0x00	RW	High Byte of Minimum Value for clk_trail, unit ns Bit[1:0]: clk_trail_min[9:8]
0x4823	CLK_TRAIL_MIN	0x3C	RW	Low Byte of Minimum Value for clk_trail clk_trail_real = clk_trail_min_o + Tui*ui_clk_trail_min_o

table 6-10 MIPI transmitter registers (sheet 7 of 8)

address	register name	default value	R/W	description
0x4824	LPX_P_MIN	0x00	RW	High Byte of Minimum Value for lpx_p, unit ns Bit[1:0]: lpx_p_min[9:8]
0x4825	LPX_P_MIN	0x32	RW	Low Byte of Minimum Value for lpx_p lpx_p_real = lpx_p_min_o + Tui*ui_lpx_p_min_o
0x4826	HS_PREPARE_MIN	0x00	RW	High Byte of Minimum Value for hs_prepare, unit ns Bit[1:0]: hs_prepare_min[9:8]
0x4827	HS_PREPARE_MIN	0x32	RW	Low Byte of Minimum Value for hs_prepare hs_prepare_real = hs_prepare_min_o + Tui*ui_hs_prepare_min_o
0x4827	HS_PREPARE_MIN	0x32	RW	Low Byte of Minimum Value for hs_prepare hs_prepare_real = hs_prepare_min_o + Tui*ui_hs_prepare_min_o
0x4828	HS_EXIT_MIN	0x00	RW	High Byte of Minimum Value for hs_exit, unit ns Bit[1:0]: hs_exit_min[9:8]
0x4829	HS_EXIT_MIN	0x64	RW	Low Byte of Minimum Value for hs_exit hs_exit_real = hs_exit_min_o + Tui*ui_hs_exit_min_o
0x482A	UI_HS_ZERO_MIN	0x05	RW	Minimum UI Value of hs_zero, unit UI
0x482B	UI_HS_TRAIL_MIN	0x04	RW	Minimum UI Value of hs_trail, unit UI
0x482C	UI_CLK_ZERO_MIN	0x00	RW	Minimum UI Value of clk_zero, unit UI
0x482D	UI_CLK_PREPARE_MIN	0x00	RW	Minimum UI Value of clk_prepare, unit UI
0x482E	UI_CLK_POST_MIN	0x34	RW	Minimum UI Value of clk_post, unit UI
0x482F	UI_CLK_TRAIL_MIN	0x00	RW	Minimum UI Value of clk_trail, unit UI
0x4830	UI_LPX_P_MIN	0x00	RW	Minimum UI Value of lpx_p, unit UI
0x4831	UI_HS_PREPARE_MIN	0x04	RW	Minimum UI Value of hs_prepare, unit UI
0x4832	UI_HS_EXIT_MIN	0x00	RW	Minimum UI Value of hs_exit, unit UI
0x4833	MIPI_REG_MIN	0x00	RW	MIPI Register Address, Lower Bound (high byte) Address range of MIPI RW registers is from mipi_reg_min to mipi_reg_max
0x4834	MIPI_REG_MIN	0x00	RW	MIPI Register Address, Lower Bound (low byte)
0x4835	MIPI_REG_MAX	0xFF	RW	MIPI Register Address, Upper Bound (high byte)
0x4836	MIPI_REG_MAX	0xFF	RW	MIPI Register Address, Upper Bound (low byte)
0x4837	PCLK_PERIOD	0x15	RW	Period of pclk2x, pclk_div = 1 and 1-bit decimal
0x4838	WKUP_DLY	0x02	RW	Wakeup Delay for MIPI

table 6-10 MIPI transmitter registers (sheet 8 of 8)

address	register name	default value	R/W	description
0x483A	DIR_DLY	0x08	RW	Change LP Direction Delay/2 After LP11
0x483C	MIPI CTRL 33	0x4F	RW	Bit[7:4]: t_lpx Unit: sclk cycles Bit[3:0]: t_clk_pre Unit: sclk cycles
0x483D	MIPI_T_TA_GO	0x10	RW	t_ta_go Unit: sclk cycles
0x483E	MIPI_T_TA_SURE	0x06	RW	t_ta_sure Unit: sclk cycles
0x483F	MIPI_T_TA_GET	0x14	RW	t_ta_get Unit: sclk cycles
0x4843	SNR_PCLK_DIV	0x00	RW	Bit[0]: PCLK divider 0: PCLK/sclk = 2 and pclk_div = 1 1: PCLK/sclk = 1 and pclk_div = 1
0x4860	MIPI CTRL 60	–	R	MIPI Read/Write Only Bit[0]: mipi_dis_me 0: Enable MIPI read/write registers 1: Disable MIPI read/write registers
0x4861	HD_SK_REG0	–	R	MIPI Read/Write, SCCB and MCU Read Only
0x4862	HD_SK_REG1	–	R	MIPI Read/Write, SCCB and MCU Read Only
0x4863	HD_SK_REG2	–	R	MIPI Read/Write, SCCB and MCU Read Only
0x4864	HD_SK_REG3	–	R	MIPI Read/Write, SCCB and MCU Read Only
0x4865	MIPI_ST	–	R	Bit[5]: lp_rx_sel_i 1: MIPI_LP_RX receives LP data Bit[4]: tx_busy_i 1: MIPI_TX_LP_TX is busy to send LP data Bit[3]: mipi_lp_p1_i MIPI low power input for lane 1p Bit[2]: mipi_lp_n1_i MIPI low power input for lane 1n Bit[1]: mipi_lp_p2_i MIPI low power input for lane 2p Bit[0]: mipi_lp_n2_i MIPI low power input for lane 2n
0x4866	T_GLB_TIM_H	–	R	Bit[7]: VHREF ahead of flag Must delay VHREF Bit[6:0]: vhref_delay_h
0x4867	T_GLB_TIM_L	–	R	vhref_delay_l

7 register tables

The following tables provide descriptions of the device control registers contained in the OV2722. For all registers enable/disable bits, ENABLE = 1 and DISABLE = 0. The device slave addresses are 0x6C for write and 0x6D for read.

7.1 system control [0x3000 - 0x3044]

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

address	register name	default value	R/W	description
0x3000	SC_CMMN_PAD_OEN0	0x00	RW	Bit[7:2]: Not used Bit[1:0]: io_y_oen[9:8]
0x3001	SC_CMMN_PAD_OEN1	0x00	RW	Bit[7:0]: io_y_oen[7:0]
0x3002	SC_CMMN_PAD_OEN2	0x00	RW	Bit[7]: io_vsync_oen Bit[6]: io_href_oen Bit[5]: io_pclk_oen Bit[4:0]: Reserved
0x3003~0x3004	NOT USED	–	–	Not Used
0x3005	SC_CMMN_CLKRST5	0x00	RW	Bit[7:6]: Reserved Bit[5]: sclk_sram Bit[4]: sclk_syncfifo Bit[3:2]: Reserved Bit[1]: rst_sram Bit[0]: rst_syncfifo
0x3006	RSVD	–	–	Reserved
0x3007	SC_CMMN_PCLK_DIV_CTRL	0x01	RW	Bit[7]: pdiv_auto_mod 0: Use R_DIVM+1 as PCLK divider 1: Use pclk_ratio_i as PCLK divider Bit[6:0]: Reserved
0x3008	SC_CMMN_PAD_OUT0	0x00	RW	Bit[7:2]: Not used Bit[1:0]: io_y_o[9:8]
0x3009	SC_CMMN_PAD_OUT1	0x00	RW	Bit[7:0]: io_y_o[7:0]
0x300A	SC_CMMN_CHIP_ID	0x27	R	Chip ID High Byte
0x300B	SC_CMMN_CHIP_ID	0x22	R	Chip ID Low Byte
0x300C	SC_CMMN_SCCB_ID	0x6C	RW	SCCB ID

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

address	register name	default value	R/W	description
0x300D	SC_CMMN_PAD_OUT2	0x00	RW	Bit[7]: io_vsync_o Bit[6]: io_href_o Bit[5]: io_pclk_o Bit[4:0]: Reserved
0x300E	SC_CMMN_PAD_SEL0	0x00	RW	Bit[7:2]: Not used Bit[1:0]: io_y_sel[9:8]
0x300F	SC_CMMN_PAD_SEL1	0x00	RW	Bit[7:0]: io_y_sel[7:0]
0x3010	SC_CMMN_PAD_SEL2	0x00	RW	Bit[7]: io_vsync_sel Bit[6]: io_href_sel Bit[5]: io_pclk_sel Bit[4:0]: Reserved
0x3011	SC_CMMN_PAD_PK	0x02	RW	Bit[7]: pd_dato_en Bit[6:5]: Drive strength adjustment Bit[4:0]: Reserved
0x3012	RSVD	–	–	Reserved
0x3013	SC_CMMN_A_PWC_PK_O	0x00	RW	Bit[7]: sel_vdd Bit[6]: bp_reg_lp Bit[5]: bp_npump Bit[4]: bp_ppump Bit[3]: bp_regulator Bit[2]: pd_vmrst Bit[1]: pd_other Bit[0]: pd_bias
0x3014	SC_CMMN_A_PWC_PK_O	0x0B	RW	Bit[7]: vsun_sel_c Bit[6:4]: apd[2:0] Bit[3:0]: dio
0x3016	SC_CMMN_MIPI_PHY	0x00	RW	Bit[7:6]: lph Bit[5:4]: Reserved Bit[3]: mipi_pad_enable Bit[2]: pgm_bp_hs_en_lat Bypass the latch of hs_enable Bit[1:0]: ictl[1:0] Bias current adjustment

table 7-1 system control registers (sheet 3 of 6)

address	register name	default value	R/W	description
0x3017	SC_CMMN_MIPI_PHY	0x10	RW	Bit[7:6]: pgm_vcm[1:0] High speed common mode voltage Bit[5:4]: pgm_lptx[1:0] 01: Driving strength of low speed transmitter Bit[3]: ihalf Biascurrent reduction Bit[2]: pgm_vicd CD input low voltage Bit[1]: pgm_vih CD input high voltage dummy Bit[0]: pgm_hs_valid Valid delay dummy
0x3018	SC_CMMN_MIPI_SC_CTRL	0x58	RW	Bit[7:5]: Chip debug Bit[4]: r_phy_pd_mipi 1: Power down PHY HS TX Bit[3]: r_phy_pd_lprx 1: Power down PHY LP RX module Bit[2]: mipi_en 0: DVP enable 1: MIPI enable Bit[1]: mipi_susp_reg MIPI system suspend register 1: suspend Bit[0]: lane_dis_op 0: Use ~mipi_release1/2 and lane_disable1/2 to disable two-data lane 1: Use lane_disable1/2 to disable two-data lane
0x3019	SC_CMMN_MIPI_SC_CTRL	0x10	RW	Bit[7:0]: MIPI ULPS resume mark1 detect length
0x301A	SC_CMMN_CLKRST0	0xF0	RW	Bit[7]: sclk_gt Bit[6]: Reserved Bit[5]: sclk_aec Bit[4]: sclk_tc Bit[3]: mipi_phy_rst_o Bit[2]: Reserved Bit[1]: rst_aec Bit[0]: rst_tc

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

address	register name	default value	R/W	description
0x301B	SC_CMMN_CLKRST1	0xF0	RW	Bit[7]: sclk_blc Bit[6]: sclk_isp Bit[5]: sclk_avg Bit[4]: sclk_vfifo Bit[3]: rst_blc Bit[2]: rst_isp Bit[1]: rst_avg Bit[0]: rst_vfifo
0x301C	SC_CMMN_CLKRST2	0xF0	RW	Bit[7]: pclk_dvp Bit[6]: sclk_mipi Bit[5]: sclk_cif Bit[4]: sclk_OTP Bit[3]: rst_dvp Bit[2]: rst_mipi Bit[1]: rst_cif Bit[0]: rst_OTP
0x301D	SC_CMMN_CLKRST3	0xF4	RW	Bit[7]: Reserved Bit[6]: sclk_grp Bit[5]: sclk_bist Bit[4]: sclk_ac Bit[3]: Reserved Bit[2]: rst_grp Bit[1]: rst_bist Bit[0]: rst_ac
0x301E	SC_CMMN_CLKRST4	0xF0	RW	Bit[7]: Reserved Bit[6]: padclk_mipi_sc Bit[5]: pclk_vfifo Bit[4]: pclk_mipi Bit[3]: Reserved Bit[2]: rst_mipi_sc Bit[1]: rst_gt Bit[0]: Reserved
0x301F	RSVD	–	–	Reserved
0x3020	SC_CMMN_CLOCK_SEL	0x13	RW	Bit[7]: Not used Bit[6]: pclk_ratio_exp 1: Exponential pclk_ratio_i to 2^n Bit[5]: yuv_out_en 0: SOC sensor, output RAW data, or RAW sensor 1: SOC sensor, output YUV data Bit[4]: dvp_sclk_en 1: Use pll_sclk_i instead pll_pclk_i for DVP Bit[3]: pclk_sel Bit[2:1]: Reserved Bit[0]: sclk2x_sel

table 7-1 system control registers (sheet 5 of 6)

address	register name	default value	R/W	description
0x3021	SC_CMMN_MISC_CTRL	0x23	RW	Bit[7:6]: Not used Bit[5]: fst_stby_ctr 0: Software standby enter at v_blk 1: Software standby enter at l_blk Bit[4]: mipi_ctr_en 0: Disable 1: Enable MIPI remote reset and suspend control SC Bit[3]: mipi_rst_sel 0: MIPI remote reset all registers 1: MIPI remote reset all digital module Bit[2]: gpio_pclk_en Bit[1]: frex_ef_sel Bit[0]: cen_global_o
0x3022	SC_CMMN_MIPI_SC_CTRL	–	R	Bit[7:4]: Not used Bit[3]: lptx_ck_opt Bit[2]: pull_down_clk_lane Bit[1]: pull_down_data_lane2 Bit[0]: pull_down_data_lane1
0x302A	SC_CMMN_SUB_ID	–	R	Bit[7:4]: Process Bit[3:0]: Version
0x3034	SC_CMMN_PLL_CTRL0	0x1A	RW	Bit[7]: Not used Bit[6:4]: pll_charge_pump Bit[3:0]: mipi_bit_mode 0x8: 8-bit mode 0xA: 10-bit mode Others: Reserved for future use
0x3035	SC_CMMN_PLL_CTRL1	0x11	RW	Bit[7:4]: system_clk_div Slows down all clocks Bit[3:0]: scale_divider_mipi MIPI PCLK/SERCLK can be slowed down when image is scaled down
0x3036	SC_CMMN_PLL_MULTIPLIER	0x69	RW	Bit[7]: Not used Bit[6:0]: PLL_multiplier (4~129)
0x3037	SC_CMMN_PLL_CTR13	0x03	RW	Bit[7]: pll_root_div 0: Bypass 1: /2 Bit[6:4]: pll_prediv 1, 2, 3, 4, 6, 8 Bit[3]: Reserved Bit[2:0]: pll1_cp

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

address	register name	default value	R/W	description
0x3038	SC_CMMN_PLL_DEBUG_OPT	0x00	RW	Bit[7:2]: Not used Bit[1:0]: pll1_divb
0x3039	SC_CMMN_PLL_CTRL_R	0x00	RW	Bit[7]: pll_bypass Bit[6:0]: Reserved
0x303A	SC_CMMN_PLLS_CTRL0	0x00	RW	Bit[7]: plls_bypass Bit[6:0]: Reserved
0x303B	SC_CMMN_PLLS_CTRL1	0x19	RW	Bit[7:5]: Not used Bit[4:0]: plls_multiplier
0x303C	SC_CMMN_PLLS_CTRL2	0x11	RW	Bit[7]: Not used Bit[6:4]: plls_cp Bit[3:0]: plls_sys_div
0x303D	SC_CMMN_PLLS_CTRL3	0x30	RW	Bit[7:6]: Not used Bit[5:4]: plls_pre_div 00: /1 01: /1.5 10: /2 11: /3 Bit[2]: plls_div_r 0: /1 1: /2 Bit[1:0]: plls_seld5 00: /1 01: /1 10: /2 11: /2.5
0x3040	SC_SOC_CLKRST5	0xF0	RW	Bit[7]: sclk_isp_fc Bit[6]: sclk_fc Bit[5]: Reserved Bit[4]: sclk_fmt Bit[3]: rst_isp_fc Bit[2]: rst_fc Bit[1]: Reserved Bit[0]: rst_fmt
0x3041~ 0x3044	CHIP_DEBUG	–	RW	Debug Mode

7.2 SCCB control [0x3100 - 0x3209]

table 7-2 SCCB registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x3100	SCCB CTRL	0x00	RW	Bit[7:4]: Not used Bit[3]: r_sda_dly_en Bit[2:0]: r_sda_dly
0x3101	SCCB OPT	0x12	RW	Bit[7:5]: Not used Bit[4]: en_ss_addr_inc Bit[3]: r_sda_byp_sync 0: Two clock stage sync for sda_i 1: No sync for sda_i Bit[2]: r_scl_byp_sync 0: Two clock stage sync for scl_i 1: No sync for scl_i Bit[1]: r_msk_glitch Bit[0]: r_msk_stop
0x3102	SCCB FILTER	0x00	RW	Bit[7:4]: r_sda_num Bit[3:0]: r_scl_num
0x3103	SCCB SYSREG	0x00	RW	Bit[7]: Not used Bit[6]: ctrl_rst_mipisc Bit[5]: ctrl_rst_srb Bit[4]: ctrl_rst_sccb_s Bit[3]: ctrl_rst_pon_sccb_s Bit[2]: ctrl_rst_clkmod Bit[1]: ctrl_rst_mipi_phy_rst_o Bit[0]: ctrl_pll_rst_o
0x3104	PWUP DIS	0x01	RW	Bit[7:5]: Not used Bit[4]: r_srb_clk_syn_en Bit[3]: pwup_dis2 Bit[2]: pwup_dis1 Bit[1]: pll_clk_sel Bit[0]: pwup_dis0
0x3105	PADCLK DIV	0x11	RW	Bit[7:6]: Not used Bit[5]: sclk use p_clk_i Bit[4]: Sleep enable Bit[3:0]: Padclk divider for SCCB

table 7-2 SCCB registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x3106	SRB CTRL	0xF9	RW	Bit[7:4]: Reserved Bit[3:2]: PLL clock divider 00: pll_sclk 01: pll_sclk/2 10: pll_sclk/4 11: pll_sclk Bit[1]: rst_arb 1: Reset arbiter Bit[0]: sclk_arb 1: Enable sclk to arbiter
0x3200	GROUP ADR0	0x00	RW	Group0 Start Address in SRAM Actual address is {0x3200[3:0], 4'h0}
0x3201	GROUP ADR1	0x04	RW	Group1 Start Address in SRAM Actual address is {0x3201[3:0], 4'h0}
0x3202	GROUP ADR2	0x08	RW	Group2 Start Address in SRAM Actual address is {0x3202[3:0], 4'h0}
0x3203	GROUP ADR3	0x0B	RW	Group3 Start Address in SRAM Actual address is {0x3203[3:0], 4'h0}
0x3204	GROUP LEN0	–	R	Length of Group0
0x3205	GROUP LEN1	–	R	Length of Group1
0x3206	GROUP LEN2	–	R	Length of Group2
0x3207	GROUP LEN3	–	R	Length of Group3
0x3208	GROUP ACCESS	–	W	Bit[7:4]: group_ctrl 0000: Group hold start 0001: Group hold end 1010: Group launch Others: Reserved Bit[3:0]: Group ID 0000: Group bank 00 0001: Group bank 1 0010: Group bank 2 0011: Group bank 3 Others: Reserved
0x3209	SRAM TEST	0x0A	RW	Bit[7:5]: Not used Bit[4]: Group hold SRAM test enable Bit[3:0]: Group hold SRAM RM[3:0]

7.3 AEC control [0x3500 - 0x350D, 0x3A00 - 0x3A62]

table 7-3 AEC registers (sheet 1 of 5)

address	register name	default value	R/W	description
0x3500	AEC PK EXPO	0x00	RW	Bit[7:4]: Reserved Bit[3:0]: aec_11_pk_expo[19:16]
0x3501	AEC PK EXPO	0x02	RW	Bit[7:0]: aec_11_pk_expo[15:8]
0x3502	AEC PK EXPO	0x00	RW	Bit[7:0]: aec_11_pk_expo[7:0]
0x3503	AEC PK CTRL0	0x00	RW	Bit[7:6]: Reserved Bit[5]: Gain delay option 0: 1 frame latch 1: Delay 1 frame latch Bit[4]: Choose delay option 0: Delay disable 1: Delay enable Bit[3:2]: Reserved Bit[1]: AGC manual 0: Auto enable 1: Manual enable Bit[0]: AEC manual 0: Auto enable 1: Manual enable
0x3508	AEC PK GAIN	0x00	RW	Bit[7:2]: Reserved Bit[1:0]: AEC real gain[9:8]
0x3509	AEC PK GAIN	0x00	RW	Bit[7:0]: AEC real gain[7:0]
0x350C	AEC PK VTS	0x00	RW	Bit[7:0]: AEC VTS[15:8]
0x350D	AEC PK VTS	0x00	RW	Bit[7:0]: AEC VTS[7:0]

table 7-3 AEC registers (sheet 2 of 5)

address	register name	default value	R/W	description
0x3A00	AEC CTRL00	0x78	RW	Bit[7]: Not used Bit[6]: less_1line_en Enable the function that exposure can be less than one line 0: Disable 1: Enable Bit[5]: band_en Enable banding effect removal 0: Disable 1: Enable Bit[4]: less_1band_en Enable the function that exposure can be less than 1 band 0: Disable 1: Enable Bit[3]: start_sel 0: Select tc_eof_blc_i as start signal 1: Select avg_done_i as start signal Bit[2]: night_mode_en 0: Disable 1: Enable Bit[1]: new_bal_mode_sel Select new balance method Bit[0]: freeze_en Enable freeze mode 0: Disable 1: Enable
0x3A01	AEC CTRL01	0x04	RW	Bit[7:0]: min_expo Minimum exposure (set 1, high 4 bits represent integer lines, low 4 bits represent fraction lines)
0x3A02	AEC CTRL02	0x0F	RW	Bit[7:5]: Not used Bit[4:0]: max_expo_60[12:8] Maximum exposure for 60Hz (set 1)
0x3A03	AEC CTRL03	0xE0	RW	Bit[7:0]: max_expo_60[12:8] Maximum exposure for 60Hz (set 1, no fraction lines)
0x3A04	NOT USED	–	–	Not Used
0x3A05	AEC CTRL05	0x30	RW	Bit[7]: d5060_rvs Reverse flag signal of 50/60 Hz Bit[6]: night_mode_insert_frame_en Enable frame insertion in night mode 0: Disable 1: Enable Bit[5]: auto_step_en Enable automatic step calculation Bit[4:0]: step_ratio Step ratio used in automatic step calculation

table 7-3 AEC registers (sheet 3 of 5)

address	register name	default value	R/W	description
0x3A06	AEC CTRL06	0x10	RW	Bit[7:5]: Not used Bit[4:0]: manual_step1 Step 1 in manual step calculation
0x3A07	AEC CTRL07	0x18	RW	Bit[7:4]: manual_step2 Step 2 in manual step calculation Bit[3:0]: manual_step3 Step 3 in manual step calculation
0x3A08	AEC CTRL08	0x00	RW	Bit[7:2]: Not used Bit[1:0]: band_step50[9:8] Band step for 50 Hz light
0x3A09	AEC CTRL09	0x98	RW	Bit[7:0]: band_step50[7:0] Band step for 50 Hz light
0x3A0A	AEC CTRL0A	0x00	RW	Bit[7:2]: Not used Bit[1:0]: band_step60[9:8] Band step for 60 Hz light
0x3A0B	AEC CTRL0B	0x7F	RW	Bit[7:0]: band_step60[7:0] Band step for 60 Hz light
0x3A0C	AEC CTRL0C	0xE4	RW	Bit[7:4]: max_fraction_expo Maximum fraction exposure Bit[3:0]: min_fraction_expo Minimum fraction exposure
0x3A0D	AEC CTRL0D	0x04	RW	Bit[7:6]: Not used Bit[5:0]: band_num60 Maximum band number in one frame for 60 Hz light source
0x3A0E	AEC CTRL0E	0x03	RW	Bit[7:6]: Not used Bit[5:0]: band_num50 Maximum band number in one frame for 50 Hz light source
0x3A0F	AEC CTRL0F	0x78	RW	Bit[7:0]: stable_range_top_limit Upper bound of stable range (set 1)
0x3A10	AEC CTRL10	0x68	RW	Bit[7:0]: stable_range_bot_limit Lower bound of stable range (set 1)
0x3A11	AEC CTRL11	0xD0	RW	Bit[7:0]: manual_step_top_limit Upper bound of the range to determine step value in manual step calculation
0x3A12	AEC CTRL12	0x00	RW	Bit[7:0]: average_man Manual average value

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

address	register name	default value	R/W	description
0x3A13	AEC CTRL13	0x90	RW	Bit[7]: pre_gain_en Enable pre sensor gain Bit[6:0]: pre_gain Value of pre sensor gain where pre-gain means the output real gain must be larger than pre-gain
0x3A14	AEC CTRL14	0x0E	RW	Bit[7:5]: Not used Bit[4:0]: max_expo50[12:8] Maximum exposure for 50Hz (set 1)
0x3A15	AEC CTRL15	0x40	RW	Bit[7:0]: max_expo50[7:0] Maximum exposure for 50Hz (set 1, no fraction lines)
0x3A16	NOT USED	–	–	Not Used
0x3A17	AEC CTRL16	0x01	RW	Bit[7:2]: Not used Bit[1:0]: gain_night_thresh Threshold value of gain_night
0x3A18	AEC CTRL17	0x07	RW	Bit[7:3]: Not used Bit[2:0]: snr_gain_top_limit[10:8] Upper bound of output sensor gain
0x3A19	AEC CTRL18	0xC0	RW	Bit[7:0]: snr_gain_top_limit[7:0] Upper bound of output sensor gain
0x3A1A	AEC CTRL19	0x00	RW	Bit[7:0]: min_vts_diff Minimum difference between VTS and maximum one frame exposure
0x3A1B	AEC CTRL1A	0x78	RW	Bit[7:0]: stable_range_top_limit Upper bound of stable range (set 2)
0x3A1C	AEC CTRL1B	0x06	RW	Bit[7:0]: led_add_row[15:8] Added rows number for LED mode
0x3A1D	AEC CTRL1C	0x18	RW	Bit[7:0]: led_add_row[7:0] Added rows number for LED mode
0x3A1E	AEC CTRL1D	0x68	RW	Bit[7:0]: stable_range_bot_limit Lower bound of stable range (set 2)
0x3A1F	AEC CTRL1E	0x40	RW	Bit[7:0]: manual_step_bot_limit Lower bound of the range to determine step value in manual step calculation

table 7-3 AEC registers (sheet 5 of 5)

address	register name	default value	R/W	description
0x3A20	AEC CTRL1F	0x20	RW	Bit[7]: blc_on Enable the function of subtracting the black level Bit[6:3]: BLC Black level Bit[2]: strobe_en Enable strobe option Bit[1]: average_man_en Enable manual average Bit[0]: expo_nochg_cal_en Enable the calculation of the case current exposure doesn't change
0x3A21	AEC CTRL20	0x70	RW	Bit[7]: Not used Bit[6:4]: night_mode_max_inset_frame Maximum inserted frame number in frame insertion method of night mode Bit[3:1]: Not used Bit[0]: gain_adj_opt 0: Add 1 to calculated gain if exposure and gain remains unchanged and AEC increases. Subtract 1 to calculated gain if exposure and gain remains unchanged and AEC decreases. 1: No operations of adding 1 or subtracting 1
0x3A22~ 0x3A24	NOT USED	–	–	Not Used
0x3A25	AEC CTRL21	0x00	RW	Bit[7:5]: Not used Bit[4:2]: freeze_cnt AEC update once per freeze_cnt frames Bit[1]: less_1line_fraction_lim_en Enable fraction constraint in less than one line state Bit[0]: auto_step_same_speed_en Enable same auto_step speed in automatic step mode
0x3A26	AEC CTRL25	0x02	RW	Bit[7:0]: expo_line If exposure is less than expo_line, speed of auto_step is step_man2; otherwise, speed is r_step_auto
0x3A27~ 0x3A4F	NOT USED	–	–	Not Used
0x3A50~ 0x3A62	AEC DEBUG	–	R	AEC Debug Information

7.4 analog control [0x3600 - 0x373A]

table 7-4 analog control registers

address	register name	default value	R/W	description
0x3600~ 0x3637	DEBUG	–	–	Debug Mode
0x3700~ 0x373A	DEBUG	–	–	Debug Mode

7.5 system timing [0x3800 - 0x3834]

table 7-5 system timing registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x3800	H_CROP_START	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Horizontal crop start address[11:8]
0x3801	H_CROP_START	0x00	RW	Bit[7:0]: Horizontal crop start address[7:0]
0x3802	V_CROP_START	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Vertical crop start address[11:8]
0x3803	V_CROP_START	0x00	RW	Bit[7:0]: Vertical crop start address[7:0]
0x3804	H_CROP_END	0x07	RW	Bit[7:4]: Not used Bit[3:0]: Horizontal crop end address[11:8]
0x3805	H_CROP_END	0x8B	RW	Bit[7:0]: Horizontal crop end address[7:0]
0x3806	V_CROP_END	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Vertical crop end address[11:8]
0x3807	V_CROP_END	0x43	RW	Bit[7:0]: Vertical crop end address[7:0]
0x3808	H_OURPUT_SIZE	0x07	RW	Bit[7:4]: Not used Bit[3:0]: Horizontal output size[11:8]
0x3809	H_OUTPUT_SIZE	0x80	RW	Bit[7:0]: Horizontal output size[7:0]
0x380A	V_OUTPUT_SIZE	0x04	RW	Bit[7:4]: Not used Bit[3:0]: Vertical output size[11:8]
0x380B	V_OUTPUT_SIZE	0x38	RW	Bit[7:0]: Vertical output size[7:0]
0x380C	TIMING_HTS	0x08	RW	Bit[7:0]: Horizontal total size[15:8]
0x380D	TIMING_HTS	0x5C	RW	Bit[7:0]: Horizontal total size[7:0]

table 7-5 system timing registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x380E	TIMING_VTS	0x04	RW	Bit[7:0]: Vertical total size[15:8]
0x380F	TIMING_VTS	0x60	RW	Bit[7:0]: Vertical total size[7:0]
0x3810	H_WIN_OFF	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Horizontal windowing offset[11:8]
0x3811	H_WIN_OFF	0x04	RW	Bit[7:0]: Horizontal windowing offset[7:0]
0x3812	V_WIN_OFF	0x00	RW	Bit[7:4]: Not used Bit[3:0]: Vertical windowing offset[11:8]
0x3813	V_WIN_OFF	0x02	RW	Bit[7:0]: Vertical windowing offset[7:0]
0x3814	H_SUB	0x11	RW	Bit[7:4]: Horizontal sub-sample odd increase number Bit[3:0]: Horizontal sub-sample even increase number
0x3815	V_SUB	0x11	RW	Bit[7:4]: Vertical sub-sample odd increase number Bit[3:0]: Vertical sub-sample even increase number
0x3816	TIMING_HSYNCST	0x00	RW	Bit[7:4]: Not used Bit[3:0]: HSYNC start point[11:8]
0x3817	TIMING_HSYNCST	0x00	RW	Bit[7:0]: HSYNC start point[7:0]
0x3818	TIMING_HSYNCW	0x00	RW	Bit[7:4]: Not used Bit[3:0]: HSYNC window[11:8]
0x3819	TIMING_HSYNCW	0x00	RW	Bit[7:0]: HSYNC window[7:0]
0x3820	TIMING_TC_REG20	0x40	RW	Bit[7]: Reserved Bit[6]: Vertical binning Bit[5]: r_vsub48_alt Bit[4]: r_vflip_blc Bit[3]: Sensor select 0: Internal sensor mode 1: External sensor mode Bit[2]: r_vflip_isp Bit[1]: r_vflip_snr Bit[0]: Chip debug
0x3821	TIMING_TC_REG21	0x00	RW	Bit[7:5]: Not used Bit[4]: Array hskip/hbin disable Bit[3]: Reserved Bit[2]: r_mirror_isp Bit[1]: r_mirror_snr Bit[0]: Horizontal binning
0x3822	TIMING_TC_REG22	0x10	RW	Bit[7:5]: Not used Bit[4:0]: r_ablc

table 7-5 system timing registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x3823	TIMING_TC_REG23	0x00	RW	Bit[7:1]: Not used Bit[0]: r_isp_bypass
0x3824	TIMING_TC_REG24	0x01	RW	Bit[7:5]: Not used Bit[4:0]: r_pclk_ratio_o
0x3825	RSVD	–	–	Reserved
0x3826	TIMING_TC_REG26	0x03	RW	Bit[7:6]: Not used Bit[5:4]: r_vts_ctrl Bit[3]: r_dvpshp_en Bit[2]: r_rip_sof_en_o Bit[1:0]: r_grp_adj
0x3827	TIMING_TC_REG27	0x80	RW	Bit[7:0]: r_shpdly
0x3828	TIMING_TC_REG28	0x08	RW	Bit[7:4]: Not used Bit[3:0]: r_shldly
0x3829	TIMING_TC_REG29	0x00	RW	Bit[7:4]: r_vs_int Bit[3:0]: emb_line_adj
0x382A	TIMING_TC_REG2A	0x00	RW	Bit[7:3]: Not used Bit[2]: Drop frame enable Bit[1]: r_vsize_opt Bit[0]: r_hsize_opt
0x382B	TIMING_TC_REG2B	0x0B	RW	Bit[7:0]: hrefst_f
0x382C	TIMING_TC_REG2C	0x00	RW	Bit[7:0]: x_size_opt
0x3831	TIMING_REG31	0x0F	RW	Bit[7:4]: Not used Bit[3:2]: r_vts_double Bit[1:0]: r_hts_half
0x3832	TIMING_REG32	0x01	RW	Bit[7:5]: Not used Bit[4]: r_blc_tcsnr_opt Bit[3:2]: Reserved Bit[1:0]: r_blcnum_opt
0x3833	TIMING_REG33	0x00	RW	Bit[7:0]: href_size_opt
0x3834	TIMING_REG34	0x00	RW	Bit[7:0]: vref_size_opt

7.6 OTP control [0x3D00 - 0x3D87]

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

address	register name	default value	R/W	description
0x3D00	OTP_DATA_00	0x00	RW	OTP Dump/Load Data_00
0x3D01	OTP_DATA_1	0x00	RW	OTP Dump/Load Data_1
0x3D02	OTP_DATA_2	0x00	RW	OTP Dump/Load Data_2
0x3D03	OTP_DATA_3	0x00	RW	OTP Dump/Load Data_3
0x3D04	OTP_DATA_4	0x00	RW	OTP Dump/Load Data_4
0x3D05	OTP_DATA_5	0x00	RW	OTP Dump/Load Data_5
0x3D06	OTP_DATA_6	0x00	RW	OTP Dump/Load Data_6
0x3D07	OTP_DATA_7	0x00	RW	OTP Dump/Load Data_7
0x3D08	OTP_DATA_8	0x00	RW	OTP Dump/Load Data_8
0x3D09	OTP_DATA_9	0x00	RW	OTP Dump/Load Data_9
0x3D0A	OTP_DATA_A	0x00	RW	OTP Dump/Load Data_a
0x3D0B	OTP_DATA_B	0x00	RW	OTP Dump/Load Data_b
0x3D0C	OTP_DATA_C	0x00	RW	OTP Dump/Load Data_c
0x3D0D	OTP_DATA_D	0x00	RW	OTP Dump/Load Data_d
0x3D0E	OTP_DATA_E	0x00	RW	OTP Dump/Load Data_e
0x3D0F	OTP_DATA_F	0x00	RW	OTP Dump/Load Data_f
0x3D10	OTP_DATA_16	0x00	RW	OTP Dump/Load Data_10
0x3D11	OTP_DATA_17	0x00	RW	OTP Dump/Load Data_11
0x3D12	OTP_DATA_18	0x00	RW	OTP Dump/Load Data_12
0x3D13	OTP_DATA_19	0x00	RW	OTP Dump/Load Data_13
0x3D14	OTP_DATA_20	0x00	RW	OTP Dump/Load Data_14
0x3D15	OTP_DATA_21	0x00	RW	OTP Dump/Load Data_15
0x3D16	OTP_DATA_22	0x00	RW	OTP Dump/Load Data_16
0x3D17	OTP_DATA_23	0x00	RW	OTP Dump/Load Data_17
0x3D18	OTP_DATA_24	0x00	RW	OTP Dump/Load Data_18
0x3D19	OTP_DATA_25	0x00	RW	OTP Dump/Load Data_19

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

address	register name	default value	R/W	description
0x3D1A	OTP_DATA_26	0x00	RW	OTP Dump/Load Data_1a
0x3D1B	OTP_DATA_27	0x00	RW	OTP Dump/Load Data_1b
0x3D1C	OTP_DATA_28	0x00	RW	OTP Dump/Load Data_1c
0x3D1D	OTP_DATA_29	0x00	RW	OTP Dump/Load Data_1d
0x3D1E	OTP_DATA_30	0x00	RW	OTP Dump/Load Data_1e
0x3D1F	OTP_DATA_31	0x00	RW	OTP Dump/Load Data_1f
0x3D80	OTP_PROGRAM_CTRL	0x00	RW	Bit[7]: OTP_wr_busy Bit[6:1]: Not used Bit[0]: OTP_program_enable
0x3D81	OTP_LOAD_CTRL	0x00	RW	Bit[7]: OTP_rd_busy Bit[6:1]: Not used Bit[0]: OTP_load_enable
0x3D82	OTP_PGM_PULSE	0x40	RW	Program Strobe Pulse Width Unit: 8 × system clock period
0x3D83	OTP_LOAD_PULSE	0x03	RW	Load Strobe Pulse Width Unit: System clock period
0x3D84	OTP_MODE_CTRL	0x00	RW	Bit[7:5]: Not used Bit[4]: Program disable 1: Disable Bit[3]: Mode select 0: Auto mode (program/load 32 bytes from address 0) 1: Manual mode (program/load n bytes from address start {**0x3D85**, **0x3D86**) Bit[2:0]: Reserved
0x3D85	OTP_START_ADDRESS	0x00	RW	OTP Start Address for Manual Mode (see **0x3D84**[3])
0x3D86	OTP_END_ADDRESS	0x0F	RW	OTP End Address for Manual Mode (see **0x3D84**[3])
0x3D87	OTP_PS2CS	0x03	RW	OTP PS to CSB Delay Unit: System clock period

7.7 BLC control [0x4000 ~ 0x4051]

table 7-7 BLC registers (sheet 1 of 5)

address	register name	default value	R/W	description
0x4000	BLC CTRL00	0x69	RW	Bit[7:2]: Not used Bit[1]: median_filter_en 0: Disable 1: Enable Bit[0]: offset_compensation_en 0: Disable 1: Enable
0x4001	START LINE	0x00	RW	Bit[7:6]: Not used Bit[5:0]: start_line Specify the start black line for statistics
0x4002	BLC CTRL02	0x45	RW	Bit[7]: format_chg_en 0: Disable format change BLC trigger 1: Enable format change BLC trigger Bit[6]: offset_auto_en 0: BLC offset from manual register 1: BLC offset from auto statistics Bit[5:0]: reset_frame_num Frame number BLC will redo after reset
0x4003	BLC CTRL03	0x01	RW	Bit[7]: manual_trig BLC will be triggered in manual_frame_num frames, which is counted from its rising edge, continuously. 0: Disable BLC manual trigger 1: Enable BLC manual trigger Bit[6]: freeze_en Its priority is lower than always_do option 0: BLC will be triggered normally 1: BLC will not be triggered Bit[5:0]: manual_frame_num Defines the frame number used in manual trigger mode and format change update mode
0x4004	LINE NUM	0x08	RW	Bit[7:6]: Not used Bit[5:0]: black_line_num Specify the black line number used in calculating black line level. It should be an even number greater than 0

table 7-7 BLC registers (sheet 2 of 5)

address	register name	default value	R/W	description
0x4005	BLC CTRL05	0x18	RW	Bit[7:4]: Not used Bit[3]: same_man_offset_en When this bit is enabled, the manual offsets used in BLC algorithms are same. They are all manual_offset00 (see registers {**0x400C**, **0x400D**}). This bit is effective when register **0x4002**[6] is 0. 0: Disable 1: Enable Bit[2]: Not used Bit[1]: always_do When this is enabled, the BLC will be always updated 0: Disable 1: Enable Bit[0]: Not used
0x4007	BLC CTRL07	0x00	RW	Bit[7:5]: Not used Bit[4:3]: black_line_win_sel 00: Full image 01: Windows exclude the first 16 pixels and the last 16 pixels 10: Windows exclude the first 1/16 image and the last 1/16 image 11: Windows exclude the first 1/8 image and the last 1/8 image Bit[2]: Not used Bit[1:0]: bypass_mode 00: Output limited data 01: Output low bits of input data 1x: Output high bits of input data
0x4009	BLC TARGET	0x10	RW	Bit[7:0]: Target
0x400C	MANUAL OFFSET00	0x00	RW	Bit[7:3]: Not used Bit[2:0]: manual_offset00[10:8] BLC manual offset MSB for B channel
0x400D	MANUAL OFFSET00	0x00	RW	Bit[7:0]: manual_offset00[7:0] BLC manual offset LSB for B channel
0x400E	MANUAL OFFSET01	0x00	RW	Bit[7:3]: Not used Bit[2:0]: manual_offset01[10:8] BLC manual offset MSB for Gb channel
0x400F	MANUAL OFFSET01	0x00	RW	Bit[7:0]: manual_offset01[7:0] BLC manual offset LSB for Gb channel
0x4010	MANUAL OFFSET10	0x00	RW	Bit[7:3]: Not used Bit[2:0]: manual_offset10[10:8] BLC manual offset MSB for Gr channel

table 7-7 BLC registers (sheet 3 of 5)

address	register name	default value	R/W	description
0x4011	MANUAL OFFSET10	0x00	RW	Bit[7:0]: manual_offset10[7:0] BLC manual offset LSB for Gr channel
0x4012	MANUAL OFFSET11	0x00	RW	Bit[7:3]: Not used Bit[2:0]: manual_offset11[10:8] BLC manual offset MSB for R channel
0x4013	MANUAL OFFSET11	0x00	RW	Bit[7:0]: manual_offset11[7:0] BLC manual offset LSB for R channel
0x4014	GB THRESH0	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gb_thre0[4:0] Complement number Range is [-15,15]
0x4015	GB THRESH1	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gb_thre1[4:0] Complement number Range is [-15,15]
0x4016	GB THRESH2	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gb_thre2[4:0] Complement number Range is [-15,15]
0x4017	GB THRESH3	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gb_thre3[4:0] Complement number Range is [-15,15]
0x4018	GB THRESH4	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gb_thre4[4:0] Complement number Range is [-15,15]
0x4019	GB THRESH5	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gb_thre5[4:0] Complement number Range is [-15,15]
0x401A	GR THRESH0	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gr_thre0[4:0] Complement number Range is [-15,15]
0x401B	GR THRESH1	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gr_thre1[4:0] Complement number Range is [-15,15]

table 7-7 BLC registers (sheet 4 of 5)

address	register name	default value	R/W	description
0x401C	GR THRESH2	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gr_thre2[4:0] Complement number Range is [-15,15]
0x401D	GR THRESH3	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gr_thre3[4:0] Complement number Range is [-15,15]
0x401E	GR THRESH4	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gr_thre4[4:0] Complement number Range is [-15,15]
0x401F	GR THRESH5	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gr_thre5[4:0] Complement number Range is [-15,15]
0x4020	B THRESH0	0x00	RW	Bit[7:5]: Not used Bit[4:0]: gb_thre0[4:0] Complement number Range is [-15,15]
0x4021	B THRESH1	0x00	RW	Bit[7:5]: Not used Bit[4:0]: b_thre1[4:0] Complement number Range is [-15,15]
0x4022	B THRESH2	0x00	RW	Bit[7:5]: Not used Bit[4:0]: b_thre2[4:0] Complement number Range is [-15,15]
0x4023	B THRESH3	0x00	RW	Bit[7:5]: Not used Bit[4:0]: b_thre3[4:0] Complement number Range is [-15,15]
0x4024	B THRESH4	0x00	RW	Bit[7:5]: Not used Bit[4:0]: b_thre4[4:0] Complement number Range is [-15,15]
0x4025	B THRESH5	0x00	RW	Bit[7:5]: Not used Bit[4:0]: b_thre5[4:0] Complement number Range is [-15,15]

table 7-7 BLC registers (sheet 5 of 5)

address	register name	default value	R/W	description
0x4026	R THRESH0	0x00	RW	Bit[7:5]: Not used Bit[4:0]: r_thre0[4:0] Complement number Range is [-15,15]
0x4027	R THRESH1	0x00	RW	Bit[7:5]: Not used Bit[4:0]: r_thre1[4:0] Complement number. Range is [-15,15]
0x4028	R THRESH2	0x00	RW	Bit[7:5]: Not used Bit[4:0]: r_thre2[4:0] Complement number Range is [-15,15]
0x4029	R THRESH3	0x00	RW	Bit[7:5]: Not used Bit[4:0]: r_thre3[4:0] Complement number Range is [-15,15]
0x402A	R THRESH4	0x00	RW	Bit[7:5]: Not used Bit[4:0]: r_thre4[4:0] Complement number Range is [-15,15]
0x402B	GB THRESH5	0x00	RW	Bit[7:5]: Not used Bit[4:0]: r_thre5[4:0] Complement number Range is [-15,15]
0x402C~ 0x4033	BLC DEBUG	–	R	BLC Debug Information
0x404F	BLC CTRL4F	0x7F	RW	Bit[7]: stable_range_en Stable range function enable signal 0: Disable 1: Enable Bit[6:0]: stable_range When the difference between the current black level and the previous is greater than the stable range and the stable_range_en is 1, the BLC will be triggered.
0x4051	BLC CTRL51	0x00	RW	Bit[7]: Not used Bit[6]: black_line_bypass_en 0: Do not output black lines 1: Output black lines Bit[5:0]: removed_black_line_num For debug purposes only

7.8 CIF control [0x4100]

table 7-8 CIF registers

address	register name	default value	R/W	description
0x4100	CIF CTRL 00	0x00	RW	Reverse External Sensor Input Polarity Bit[7]: Source select 0: From external DVP input 1: From MIPI receiver Bit[6]: End-of-frame signal select Bit[5]: Reverse VSYNC polarity Bit[4]: Reverse HREF polarity Bit[3]: Reverse PCLK polarity Bit[2]: Reverse RBLUE polarity Bit[1]: Reverse ptn_RBLUE polarity Bit[0]: Reverse CIF RBLUE polarity

7.9 ISP input frame control [0x4200 - 0x4203]

table 7-9 ISP input frame control registers

address	register name	default value	R/W	description
0x4200	ISP INPUT FRAME CTRL00	0x00	RW	Bit[7:3]: Not used Bit[2]: fcnt_eof_sel Bit[1]: fcnt_mask_dis Bit[0]: Frame counter reset
0x4201	ISP INPUT FRAME CTRL01	0x00	RW	Control Passed Frame Number Bit[7:4]: Not used Bit[3:0]: Frame ON number When both ON and OFF numbers are set to 0x00, frame control is in bypass mode
0x4202	ISP INPUT FRAME CTRL02	0x00	RW	Control Masked Frame Number Bit[7:4]: Not used Bit[3:0]: Frame OFF number When both ON and OFF numbers are set to 0x00, frame control is in bypass mode
0x4203	ISP INPUT FRAME CTRL03	0x00	RW	Bit[7:6]: Not used Bit[5]: data_mask_dis Bit[4]: valid_mask_dis Bit[3]: href_mask_dis Bit[2]: eof_mask_dis Bit[1]: sof_mask_dis Bit[0]: all_mask_dis

7.10 ISP output frame control [0x4900 - 0x4903]

table 7-10 ISP output frame control registers

address	register name	default value	R/W	description
0x4900	ISP OUTPUT FRAME CTRL00	0x00	RW	Bit[7:3]: Not used Bit[2]: fcnt_eof_sel Bit[1]: fcnt_mask_dis Bit[0]: Frame counter reset
0x4901	ISP OUTPUT FRAME CTRL01	0x00	RW	Control Passed Frame Number Bit[7:4]: Not used Bit[3:0]: Frame ON number When both ON and OFF numbers are set to 0x00, frame control is in bypass mode
0x4902	ISP OUTPUT FRAME CTRL02	0x00	RW	Control Masked Frame Number Bit[7:4]: Not used Bit[3:0]: Frame OFF number When both ON and OFF numbers are set to 0x00, frame control is in bypass mode
0x4903	ISP OUTPUT FRAME CTRL03	0x00	RW	Bit[7:6]: Not used Bit[5]: data_mask_dis Bit[4]: valid_mask_dis Bit[3]: href_mask_dis Bit[2]: eof_mask_dis Bit[1]: sof_mask_dis Bit[0]: all_mask_dis

7.11 DVP control [0x4302 - 0x4305, 0x4700 - 0x470C]

table 7-11 DVP registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x4302	FORMAT MAX	0x00	RW	Bit[7:2]: Not used Bit[1:0]: clipping max[9:8]
0x4303	FORMAT MAX	0xFF	RW	Bit[7:0]: clipping max[7:0]
0x4304	FORMAT MIN	0x00	RW	Bit[7:2]: Not used Bit[1:0]: clipping min[9:8]
0x4305	FORMAT MIN	0x00	RW	Bit[7:0]: clipping min[7:0]

table 7-11 DVP registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x4700	DVP MODE SELECT	0x04	RW	Bit[7:4]: Not used Bit[3]: CCIR V select Bit[2]: CCIR F select Bit[1]: CCIR656 mode enable Bit[0]: HSYNC mode enable
0x4701	DVP VSYNC WIDTH CTRL	0x01	RW	Bit[7:0]: VSYNC width (in terms of number of lines)
0x4702	DVP_HSYVSY_NEG_WIDTH	0x01	RW	Bit[7:0]: VSYNC width[15:8] (in terms of pixel count)
0x4703	DVP_HSYVSY_NEG_WIDTH	0x00	RW	Bit[7:0]: VSYNC width[7:0] (in terms of pixel count)
0x4704	DVP VSYNC MODE	0x00	RW	Bit[7:4]: Not used Bit[3:2]: r_vsyncount_sel Bit[1]: r_vsync3_mod Bit[0]: r_vsync2_mod
0x4705	DVP_EOF_VSYNC DELAY	0x00	RW	Bit[7:0]: SOF/EOF negative edge to VSYNC positive edge delay[23:16]
0x4706	DVP_EOF_VSYNC DELAY	0x00	RW	Bit[7:0]: SOF/EOF negative edge to VSYNC positive edge delay[15:8]
0x4707	DVP_EOF_VSYNC DELAY	0x00	RW	Bit[7:0]: SOF/EOF negative edge to VSYNC positive edge delay[7:0]
0x4708	DVP_POL_CTRL	0x01	RW	Bit[7]: Clock DDR mode enable Bit[6]: Reserved Bit[5]: VSYNC gated clock enable Bit[4]: HREF gated clock enable Bit[3]: No first for FIFO Bit[2]: HREF polarity reverse Bit[1]: VSYNC polarity reverse Bit[0]: PCLK polarity reverse
0x4709	BIT_TEST_PATTERN	0x00	RW	Bit[7]: FIFO bypass mode Bit[6:4]: Data bit swap Bit[3]: Bit test mode Bit[2]: 10-bit bit test Bit[1]: 8-bit bit test Bit[0]: Bit test enable
0x470A	DVP_BYP_CTRL	0x00	RW	Bypass Control High Byte
0x470B	DVP_BYP_CTRL	0x00	RW	Bypass Control Low Byte
0x470C	DVP_BYP_SEL	0x00	RW	Bit[7:5]: Not used Bit[4]: HREF select Bit[3:0]: Bypass select

7.12 sync FIFO control [0x4500 - 0x4521]

table 7-12 sync FIFO registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x4500~ 0x4507	PSYNC DEBUG	0x00	RW	Debug Control
0x4510	PSYNC_CTRL_03	0x03	RW	Bit[7:0]: r_href_st
0x4511	PSYNC_CTRL_04	0x22	RW	psync_href_end
0x4512	PSYNC_CTRL_00	0x05	RW	psync_href_end
0x4513	PSYNC_CTRL_01	0x00	RW	Bit[7:5]: Reserved Bit[4]: r_href_mode Bit[3]: r_rblue_re Bit[2]: r_href_in_sel Bit[1:0]: r_channel_sel
0x4514	PSYNC_CTRL_02	0x00	RW	Bit[7:5]: Reserved Bit[4]: r_data_order_man_en Bit[3:0]: r_data_order 0x0: binning, f1w0 + f2w0, f1w1 + f2w1, f1w2 + f2w2, f1w3 + f2w3 0x1: binning, f1w1 + f2w1, f1w0 + f2w0, f1w3 + f2w3, f1w2 + f2w2 0x2: binning, (f1w0 + f2w0) / 2, (f1w1 + f2w1) / 2, (f1w2+f2w2) / 2, (f1w3 + f2w3) / 2 0x3: binning, (f1w1 + f2w1) / 2, (f1w0 + f2w0) / 2, (f1w3 + f2w3) / 2, (f1w2 + f2w2) / 2 0x0: full, f1w0, f2w0, f1w1, f2w1, f1w2, f2w2, f1w3, f2w3 0x1: full, f2w0, f1w0, f2w1, f1w1, f2w2, f1w2, f2w3, f1w3 (flip) 0x2: full, f1w1, f2w1, f1w0, f2w0, f1w3, f2w3, f1w2, f2w2 (mirror+flip) 0x3: full, f2w1, f1w1, f2w0, f1w0, f2w3, f1w3, f2w2, f1w2 (mirror)
0x4520	PSYNC_CTRL_03	0x00	RW	Bit[7]: sync_data_swap_en Bit[6:4]: sync_data_swap_en_mode Bit[3]: sram_data_swap_en Bit[2:0]: sram_data_swap_en_mode

table 7-12 sync FIFO registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x4521	PSYNC_CTRL_04	0x00	RW	Bit[7]: r_gray_data_sel Bit[6]: r_fifo_mux_man Bit[5]: r_fifo_mux_sel 0: f1 on 1: f2 on Bit[4]: wclk_re Bit[3:2]: f2_fifo_wr_dly Bit[1:0]: f1_fifo_wr_dly

7.13 MIPI control [0x4800 - 0x4867]

table 7-13 MIPI registers (sheet 1 of 9)

address	register name	default value	R/W	description
0x4800	MIPI CTRL 00	0x04	RW	MIPI Control 00 Bit[7]: mipi_hs_only 0: MIPI can support CD and ESCAPE mode 1: MIPI always in high speed mode Bit[6]: ck_mark1_en 1: Enable clock lane mark1 when resume Bit[5]: Clock lane gate enable 0: Clock lane is free running 1: Gate clock lane when no packet to transmit Bit[4]: Line sync enable 0: Do not send line short packet for each line 1: Send line short packet for each line Bit[3]: Lane select 0: Use lane1 as default data lane 1: Use lane2 as default data lane Bit[2]: Idle status 0: MIPI bus will be LP00 when no packet to transmit 1: MIPI bus will be LP11 when no packet to transmit Bit[1]: Clock lane first bits 0: Output 0x55 1: Output 0xAA Bit[0]: Clock lane disable 1: Manually set clock lane to low power mode

table 7-13 MIPI registers (sheet 2 of 9)

address	register name	default value	R/W	description
0x4801	MIPI CTRL 01	0x0F	RW	MIPI Control 01 Bit[7]: Long packet data type manual enable 0: Use mipi_dt 1: Use dt_man_o as long packet data (see register **0x4814**[5:0]) Bit[6]: Short packet data type manual enable 1: Use dt_spkt as short packet data (see register **0x4815**[5:0]) Bit[5]: Short packet word counter manual enable 0: Use frame counter or line counter 1: Select spkt_wc_reg_o (see {0x4812, 0x4813}) Bit[4]: PH bit order for ECC 0: {DI[7:0],WC[7:0],WC[15:8]} 1: {DI[0:7],WC[0:7],WC[8:15]} Bit[3]: PH byte order for ECC 0: {DI,WC_l,WC_h} 1: {DI,WC_h,WC_l} Bit[2]: PH byte order2 for ECC 0: {DI,WC} 1: {WC,DI} Bit[1]: mark1_en1 1: After each reset release, lane 1 should send mark1 for wkup_dly_o when mipi_sys_susp = 1 Bit[0]: mark1_en2 1: After each reset release, lane 2 should send mark1 for wkup_dly_o when mipi_sys_susp = 1

table 7-13 MIPI registers (sheet 3 of 9)

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	MIPI CTRL 03	0x50	RW	MIPI Control 03 Bit[7:6]: lp_glitch_nu 0: Use 2d of lp_in 1: Mask one sclk cycle glitch of lp_in Bit[5:4]: cd_glitch_nu 0: Use 2d of lp_cd_in 1: Mask one sclk cycle glitch o flp_cd_in Bit[3]: Enable CD plus of data lane1 0: Disable 1: Enable Bit[2]: Enable CD plus of data lane2 0: Disable 1: Enable Bit[1]: Enable CD of data_lane1 from PHY 0: Disable 1: Enable Bit[0]: Enable CD of data_lane2 from PHY 0: Disable 1: Enable

table 7-13 MIPI registers (sheet 4 of 9)

address	register name	default value	R/W	description
0x4804	MIPI CTRL 04	0x8D	RW	MIPI Control 04 Bit[7]: wait_pkt_end 1: Wait HS packet end when sending UL command Bit[6]: tx_lsb_first 0: lp_tx and lp_rx high bit first 1: Low power transmit low bit first Bit[5]: dir_recover_sel 0: Auto change to output only when turnaround command 1: Auto change to output when LP11 and GPIO is output Bit[4]: mipi_reg_en 0: Disable MIPI_REG_P to access registers, LP data will write to VFIFO 1: Enable MIPI_REG_P to access registers Bit[3]: Address read/write register will auto add 1 1: Enable Bit[2]: LP TX lane select 0: Select lane1 to transmit LP data 1: Select lane2 to transmit LP data Bit[1]: wr_first_byte 1: lp_rx will write first byte (command byte) to RAM Bit[0]: rd_ta_en 1: Send turnaround command after sending register read data

table 7-13 MIPI registers (sheet 5 of 9)

address	register name	default value	R/W	description
0x4805	MIPI CTRL 05	0x10	RW	MIPI Control 05 Bit[7]: MIPI lane1 disable 1: Disable MIPI data lane1, lane1 will be LP00 Bit[6]: MIPI lane2 disable 1: Disable MIPI data lane2, lane2 will be LP00 Bit[5]: lpx_p_sel 0: Auto calculate t_lpx_o in pclk2x domain, unit pclk2x 1: Use lp_p_min[7:0] Bit[4]: lp_rx_intr_sel 0: Send lp_rx_intr_o at the first byte 1: Send lp_rx_intr_o at the end of receiving Bit[3]: cd_tst_sel 1: Select PHY test pins Bit[2]: mipi_reg_mask 1: Disable MIPI access SRB Bit[1]: clip enable Bit[0]: hd_sk_en 0: Disable MIPI and MCU hand shake registers 1: Disable MIPI and MCU had shake registers
0x4806	MIPI REG RW CTRL	0x28	RW	Bit[7]: prbs_en Test mode Bit[6]: mipi_test Bit[5]: mipi_lp_op 0: Use new option to reduce mipi_lptx_p Bit[4]: two_lane_man_en 1: Use two_lane_man to manually control two_lane_mode Bit[3]: two_lane_man Bit[2]: rst_rtn_en 1: Change to input to allow host RW register after reset Bit[1]: frame_end_en 1: After frame end packet, change to input to allow host RW register Bit[0]: line_end_en 1: After line end packet, change to input to allow host RW register
0x480A	MIPI BIT ORDER	0x00	RW	Bit[7:3]: Not used Bit[2]: Bit order reverse Bit[1:0]: Bit position adjustment 01: {data[7:0],data[9:8]} 10: {data[1:0],data[9:2]}

table 7-13 MIPI registers (sheet 6 of 9)

address	register name	default value	R/W	description
0x4810	MIPI MAX FRAME COUNT	0xFF	RW	High Byte of Maximum Frame Count of Frame Sync Short Packet
0x4811	MIPI MAX FRAME COUNT	0xFF	RW	Low Byte of Maximum Frame Count of Frame Sync Short Packet
0x4814	MIPI CTRL14	0x2A	RW	MIPI Control 14 Bit[7:6]: Virtual channel of MIPI Bit[5:0]: Data type in manual mode
0x4815	MIPI_DT_SPKT	0x00	RW	Bit[7]: Not used Bit[6]: pclk_div 0: Use rising edge of mipi_pclk_o to generate MIPI bus to PHY 1: Use falling edge of mipi_pclk_o to generate MIPI bus to PHY Bit[5:0]: Manual data type for short packet
0x4818	HS_ZERO_MIN	0x00	RW	High Byte of Minimum Value for hs_zero, unit ns
0x4819	HS_ZERO_MIN	0x96	RW	Low Byte of Minimum Value for hs_zero, unit ns hs_zero_real = hs_zero_min_o + Tui*ui_hs_zero_min_o
0x481A	HS_TRAIL_MIN	0x00	RW	High Byte of Minimum Value for hs_trail, unit ns
0x481B	HS_TRAIL_MIN	0x3C	RW	Low Byte of Minimum Value for hs_trail, unit ns hs_trail_real = hs_trail_min_o + Tui*ui_hs_trail_min_o
0x481C	CLK_ZERO_MIN	0x01	RW	High Byte of Minimum Value for clk_zero, unit ns
0x481D	CLK_ZERO_MIN	0x86	RW	Low Byte of Minimum Value for clk_zero, unit ns clk_zero_real = clk_zero_min_o + Tui*ui_clk_zero_min_o
0x481E	CLK_PREPARE_MIN	0x00	RW	High Byte of Minimum Value for clk_prepare, unit ns Bit[1:0]: clk_prepare_min[9:8]
0x481F	CLK_PREPARE_MIN	0x3C	RW	Low Byte of Minimum Value for clk_prepare, unit ns clk_prepare_real = clk_prepare_min_o + Tui*ui_clk_prepare_min_o
0x4820	CLK_POST_MIN	0x00	RW	High Byte of Minimum Value for clk_post, unit ns Bit[1:0]: clk_post_min[9:8]
0x4821	CLK_POST_MIN	0x56	RW	Low Byte of Minimum Value for clk_post clk_post_real = clk_post_min_o + Tui*ui_clk_post_min_o
0x4822	CLK_TRAIL_MIN	0x00	RW	High Byte of Minimum Value for clk_trail, unit ns Bit[1:0]: clk_trail_min[9:8]
0x4823	CLK_TRAIL_MIN	0x3C	RW	Low Byte of Minimum Value for clk_trail clk_trail_real = clk_trail_min_o + Tui*ui_clk_trail_min_o

table 7-13 MIPI registers (sheet 7 of 9)

address	register name	default value	R/W	description
0x4824	LPX_P_MIN	0x00	RW	High Byte of Minimum Value for lpx_p, unit ns Bit[1:0]: lpx_p_min[9:8]
0x4825	LPX_P_MIN	0x32	RW	Low Byte of Minimum Value for lpx_p lpx_p_real = lpx_p_min_o + Tui*ui_lpx_p_min_o
0x4826	HS_PREPARE_MIN	0x00	RW	High Byte of Minimum Value for hs_prepare, unit ns Bit[1:0]: hs_prepare_min[9:8]
0x4827	HS_PREPARE_MIN	0x32	RW	Low Byte of Minimum Value for hs_prepare hs_prepare_real = hs_prepare_min_o + Tui*ui_hs_prepare_min_o
0x4828	HS_EXIT_MIN	0x00	RW	High Byte of Minimum Value for hs_exit, unit ns Bit[1:0]: hs_exit_min[9:8]
0x4829	HS_EXIT_MIN	0x64	RW	Low Byte of Minimum Value for hs_exit hs_exit_real = hs_exit_min_o + Tui*ui_hs_exit_min_o
0x482A	UI_HS_ZERO_MIN	0x05	RW	Minimum UI Value of hs_zero, unit UI
0x482B	UI_HS_TRAIL_MIN	0x04	RW	Minimum UI Value of hs_trail, unit UI
0x482C	UI_CLK_ZERO_MIN	0x00	RW	Minimum UI Value of clk_zero, unit UI
0x482D	UI_CLK_PREPARE_MIN	0x00	RW	Minimum UI Value of clk_prepare, unit UI
0x482E	UI_CLK_POST_MIN	0x34	RW	Minimum UI Value of clk_post, unit UI
0x482F	UI_CLK_TRAIL_MIN	0x00	RW	Minimum UI Value of clk_trail, unit UI
0x4830	UI_LPX_P_MIN	0x00	RW	Minimum UI Value of lpx_p, unit UI
0x4831	UI_HS_PREPARE_MIN	0x04	RW	Minimum UI Value of hs_prepare, unit UI
0x4832	UI_HS_EXIT_MIN	0x00	RW	Minimum UI Value of hs_exit, unit UI
0x4833	MIPI_REG_MIN	0x00	RW	MIPI Register Address, Lower Bound (high byte) Address range of MIPI RW registers is from mipi_reg_min to mipi_reg_max
0x4834	MIPI_REG_MIN	0x00	RW	MIPI Register Address, Lower Bound (low byte)
0x4835	MIPI_REG_MAX	0xFF	RW	MIPI Register Address, Upper Bound (high byte)
0x4836	MIPI_REG_MAX	0xFF	RW	MIPI Register Address, Upper Bound (low byte)
0x4837	PCLK_PERIOD	0x15	RW	Period of pclk2x, pclk_div = 1 and 1-bit decimal
0x4838	WKUP_DLY	0x02	RW	Wakeup Delay for MIPI
0x483A	DIR_DLY	0x08	RW	Change LP Direction Delay/2 After LP11

table 7-13 MIPI registers (sheet 8 of 9)

address	register name	default value	R/W	description
0x483B	MIPI_LP_GPIO	0x33	RW	Bit[7]: lp_sel1 0: Auto generate mipi_lp_dir1_o 1: Use lp_dir_man1 as mipi_lp_dir1_o Bit[6]: lp_dir_man1 0: Input 1: Output Bit[5]: lp_p1_o Bit[4]: lp_n1_o Bit[3]: lp_sel2 0: Auto generate mipi_lp_dir2_o 1: Use lp_dir_man2 as mipi_lp_dir2_o Bit[2]: lp_dir_man2 0: Input 1: Output Bit[1]: lp_p2_o Bit[0]: lp_n2_o
0x483C	MIPI CTRL 33	0x4F	RW	Bit[7:4]: t_lpx Unit: sclk cycles Bit[3:0]: t_clk_pre Unit: sclk cycles
0x483D	MIPI_T_TA_GO	0x10	RW	t_ta_go Unit: sclk cycles
0x483E	MIPI_T_TA_SURE	0x06	RW	t_ta_sure Unit: sclk cycles
0x483F	MIPI_T_TA_GET	0x14	RW	t_ta_get Unit: sclk cycles
0x4843	SNR_PCLK_DIV	0x00	RW	Bit[7:1]: Not used Bit[0]: PCLK divider 0: PCLK/sclk = 2 and pclk_div = 1 1: PCLK/sclk = 1 and pclk_div = 1
0x4860	MIPI CTRL 60	–	R	MIPI Read/Write Only Bit[7:1]: Not used Bit[0]: mipi_dis_me 0: Enable MIPI read/write registers 1: Disable MIPI read/write registers
0x4861	HD_SK_REG0	–	R	MIPI Read/Write, SCCB and MCU Read Only
0x4862	HD_SK_REG1	–	R	MIPI Read/Write, SCCB and MCU Read Only
0x4863	HD_SK_REG2	–	R	MIPI Read/Write, SCCB and MCU Read Only
0x4864	HD_SK_REG3	–	R	MIPI Read/Write, SCCB and MCU Read Only

table 7-13 MIPI registers (sheet 9 of 9)

address	register name	default value	R/W	description
0x4865	MIPI_ST	–	R	Bit[7:6]: Not used Bit[5]: lp_rx_sel_i 1: MIPI_LP_RX receives LP data Bit[4]: tx_busy_i 1: MIPI_TX_LP_TX is busy to send LP data Bit[3]: mipi_lp_p1_i MIPI low power input for lane 1p Bit[2]: mipi_lp_n1_i MIPI low power input for lane 1n Bit[1]: mipi_lp_p2_i MIPI low power input for lane 2p Bit[0]: mipi_lp_n2_i MIPI low power input for lane 2n
0x4866	T_GLB_TIM_H	–	R	Bit[7]: VHREF ahead of flag Must delay VHREF Bit[6:0]: vhref_delay_h
0x4867	T_GLB_TIM_L	–	R	vhref_delay_l

7.14 ISP control [0x5000 ~ 0x5064]

table 7-14 ISP registers (sheet 1 of 4)

address	register name	default value	R/W	description
0x5000	ISP RW00	0xFF	RW	Bit[7]: avg_en Bit[6]: dpc_en Bit[5]: awb_stat_en Bit[4]: awb_gain_en Bit[3]: Black pixel correction enable Bit[2]: White pixel correction enable Bit[1]: LENC enable Bit[0]: isp_en
0x5001	ISP RW01	0xC1	RW	Bit[7]: v_en Bit[6]: h_en Bit[5]: avg_sel Bit[4]: dpc_cal_start_sel Bit[3]: eof_sel Bit[2]: sof_sel Bit[1]: fmt_sel Bit[0]: blc_en

table 7-14 ISP registers (sheet 2 of 4)

address	register name	default value	R/W	description
0x5002	ISP RW02	0x05	RW	Bit[7]: isp_raw_en Bit[6:4]: win_yoff_adj Bit[3]: awb_bias_man_en Bit[2]: awb_bias_on Bit[1]: lenc_bias_man_en Bit[0]: lenc_bias_on
0x5003	ISP RW03	0x10	RW	Bit[7:0]: bias_man
0x5004	ISP RW04	0x10	RW	Bit[7]: gfirst_rvs Bit[6]: rblue_rvs Bit[5]: band50_flag Bit[4]: lenc_bias_plus Bit[3:0]: Debug control
0x5005	ISP RW05	0xAA	RW	Bit[7:0]: Debug control
0x5006	ISP RW06	0xAA	RW	Bit[7:0]: Debug control
0x5040	PRE 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

table 7-14 ISP registers (sheet 3 of 4)

address	register name	default value	R/W	description
0x5041	PRE CTRL01	0x41	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 this is set, two LSBs of output date are 0 Bit[4]: same_seed_en When this is set, the seed used to generate the random data are same which is set in seed register Bit[3:0]: seed Seed used in generating random data
0x5042	PRE CTRL02	0x00	RW	Bit[7:3]: Not used Bit[2:0]: intp_ln_num[10:8] Line number interrupt high bits
0x5043	PRE CTRL03	0x00	RW	Bit[7:0]: intp_ln_num[7:0] Line number interrupt low bits
0x5044	PRE CTRL04	0x02	RW	Bit[7:3]: Not used Bit[2:0]: scale_hsize_man[10:8] High bits of scale manual horizontal output size
0x5045	PRE CTRL05	0x80	RW	Bit[7:0]: scale_hsize_man[7:0] Low bits of scale manual horizontal output size
0x5046	PRE CTRL06	0x01	RW	Bit[7:3]: Not used Bit[2:0]: scale_vsize_man[10:8] High bits of scale manual vertical output size
0x5047	PRE CTRL07	0xE0	RW	Bit[7:0]: scale_vsize_man[7:0] Low bits of scale manual vertical output size
0x5048	PRE CTRL08	0x00	RW	Bit[7:3]: Not used Bit[2:0]: hoff_man[10:8] High bits of horizontal manual offset
0x5049	PRE CTRL09	0x00	RW	Bit[7:0]: hoff_man[7:0] Low bits of horizontal manual offset
0x504A	PRE CTRL0A	0x00	RW	Bit[7:3]: Not used Bit[2:0]: voff_man[10:8] High bits of vertical manual offset

table 7-14 ISP registers (sheet 4 of 4)

address	register name	default value	R/W	description
0x504B	PRE CTRL0B	0x00	RW	Bit[7:0]: voff_man[7:0] Low bits of vertical manual offset
0x504C~ 0x504F	ISP DEBUG	–	R	ISP Debug Information
0x5050	PRE CTRL10	0x30	RW	Bit[7:6]: Not used Bit[5]: mirror_opt Take the first pixel in the same position with no mirror image enable Bit[4]: flip_opt Take the first pixel in the same position with no flip image enable Bit[3]: first_px_out_sel 0: First pixel is Gb or R with window output 1: First pixel is B or Gr with window output Bit[2]: first_ln_out_sel 0: First line is GR with window output 1: First line is BG with window output Bit[1]: offset_man_en 0: Disable 1: Enable Bit[0]: scale_size_man_en 0: Disable 1: Enable
0x5051	PRE CTRL11	0x00	RW	Bit[7]: dummy_man Dummy line manual mode enable signal 0: Disable 1: Enable Bit[6:4]: dummy_line Dummy line number Bit[3]: dummy_half Reduce HREF low length by half Bit[2:0]: dummy_ratio Manual clock/valid ratio
0x5052~ 0x5064	ISP DEBUG	–	R	ISP Debug Information

7.15 AWB control [0x5180 - 0x519B]

table 7-15 AWB registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5180	AWB CTRL00	0x01	RW	Bit[7:4]: frame_num AWB will be updated every frame_num frames Bit[3]: Not used Bit[2]: freeze_en 0: Disable 1: Enable Bit[1]: gain_man_en 0: Auto mode: gain is calculated by AWB 1: Manual mode: gain is from AWB register Bit[0]: Not used
0x5181	AWB CTRL01	0xD8	RW	Bit[7:6]: fast_step_mode Fast mode gain step Bit[5:4]: diff_step Bit[3:0]: fast_awb_diff_limitation
0x5182	AWB CTRL02	0x24	RW	Bit[7:4]: stable_range Range to define the stable status Bit[3:0]: stable_wrange Wider range to determine the stable status
0x5183	AWB CTRL03	0xF0	RW	Bit[7:4]: red_gain_top_limit Red gain up limitation Bit[3:0]: red_gain_bot_limit Red gain down limitation
0x5184	AWB CTRL04	0xF0	RW	Bit[7:4]: green_gain_top_limit Green gain up limitation Bit[3:0]: green_gain_bot_limit Green gain down limitation
0x5185	AWB CTRL05	0xF0	RW	Bit[7:4]: blue_gain_top_limit Blue gain up limitation Bit[3:0]: blue_gain_bot_limit Blue gain down limitation
0x5186	AWB CTRL06	0x04	RW	Bit[7:4]: Not used Bit[3:0]: red_gain_man[11:8] Manual red gain input
0x5187	AWB CTRL07	0x00	RW	Bit[7:0]: red_gain_man[7:0] Manual red gain input
0x5188	AWB CTRL08	0x04	RW	Bit[7:4]: Not used Bit[3:0]: green_gain_man[11:8] Manual green gain input
0x5189	AWB CTRL09	0x00	RW	Bit[7:0]: green_gain_man[7:0] Manual green gain input

table 7-15 AWB registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x518A	AWB CTRL0A	0x04	RW	Bit[7:4]: Not used Bit[3:0]: blue_gain_man[11:8] Manual blue gain input
0x518B	AWB CTRL0B	0x00	RW	Bit[7:0]: blue_gain_man[7:0] Manual blue gain input
0x518C~ 0x519B	AWB DEBUG	–	R	AWB Debug Information

7.16 AVG control [0x5680 ~ 0x5693]

table 7-16 average (AVG) registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5680	AVG CTRL00	0x00	RW	Bit[7:3]: Not used Bit[2:0]: xstart_sub[10:8] AVG sub-window horizontal start position
0x5681	AVG CTRL01	0x00	RW	Bit[7:0]: xstart_sub[7:0] AVG sub-window horizontal start position
0x5682	AVG CTRL02	0x00	RW	Bit[7:3]: Not used Bit[2:0]: ystart_sub[10:8] AVG sub-window vertical start position
0x5683	AVG CTRL03	0x00	RW	Bit[7:0]: ystart_sub[7:0] AVG sub-window vertical start position
0x5684	AVG CTRL04	0x10	RW	Bit[7:3]: Not used Bit[2:0]: hsize_sub[10:8] Sub-window width
0x5685	AVG CTRL05	0xA0	RW	Bit[7:0]: hsize_sub[7:0] Sub-window width
0x5686	AVG CTRL06	0x0C	RW	Bit[7:3]: Not used Bit[2:0]: vsize_sub[10:8] Sub-window height
0x5687	AVG CTRL07	0x78	RW	Bit[7:0]: vsize_sub[7:0] Sub-window height
0x5688	AVG CTRL08	0x11	RW	Bit[7:4]: weight01 Weight of zone01 Bit[3:0]: weight00 Weight of zone00

table 7-16 average (AVG) registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x5689	AVG CTRL09	0x11	RW	Bit[7:4]: weight03 Weight of zone03 Bit[3:0]: weight02 Weight of zone02
0x568A	AVG CTRL0A	0x11	RW	Bit[7:4]: weight11 Weight of zone11 Bit[3:0]: weight10 Weight of zone10
0x568B	AVG CTRL0B	0x11	RW	Bit[7:4]: weight13 Weight of zone13 Bit[3:0]: weight12 Weight of zone12
0x568C	AVG CTRL0C	0x11	RW	Bit[7:4]: weight21 Weight of zone 21 Bit[3:0]: weight20 Weight of zone20
0x568D	AVG CTRL0D	0x11	RW	Bit[7:4]: weight23 Weight of zone23 Bit[3:0]: weight22 Weight of zone22
0x568E	AVG CTRL0E	0x11	RW	Bit[7:4]: weight31 Weight of zone31 Bit[3:0]: weight30 Weight of zone30
0x568F	AVG CTRL0F	0x11	RW	Bit[7:4]: weight33 Weight of zone33 Bit[3:0]: weight32 Weight of zone32
0x5690	AVG CTRL10	0x02	RW	Bit[7:2]: Reserved 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
0x5691~ 0x5693	AVG DEBUG	–	R	AVG Debug Information

7.17 DPC control [0x5780 - 0x57A6]

table 7-17 DPC registers

address	register name	default value	R/W	description
0x5780~ 0x57A6	DPC CTRL	–	–	Debug Only

7.18 LENC control [0x5800 - 0x581C]

table 7-18 LENC registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5800	LENC CTRL0	0x03	RW	Bit[7:3]: Not used Bit[2:0]: red_x0[10:8]
0x5801	LENC CTRL1	0xD0	RW	Bit[7:0]: red_x0[7:0]
0x5802	LENC CTRL2	0x02	RW	Bit[7:3]: Not used Bit[2:0]: red_y0[10:8]
0x5803	LENC CTRL3	0x56	RW	Bit[7:0]: red_y0[7:0]
0x5804	LENC CTRL4	0x22	RW	Bit[7]: Not used Bit[6:0]: red_a1
0x5805	LENC CTRL5	0x07	RW	Bit[7:4]: Not used Bit[3:0]: red_a2
0x5806	LENC CTRL6	0xC2	RW	Bit[7:0]: red_b1
0x5807	LENC CTRL7	0x08	RW	Bit[7:4]: Not used Bit[3:0]: red_b2
0x5808	LENC CTRL8	0x03	RW	Bit[7:3]: Not used Bit[2:0]: grn_x0[10:8]
0x5809	LENC CTRL9	0xD0	RW	Bit[7:0]: grn_x0[7:0]
0x580A	LENC CTRL10	0x02	RW	Bit[7:3]: Not used Bit[2:0]: grn_y0[10:8]
0x580B	LENC CTRL11	0x56	RW	Bit[7:0]: grn_y0[7:0]
0x580C	LENC CTRL12	0x22	RW	Bit[7]: Not used Bit[6:0]: grn_a1

table 7-18 LENC registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x580D	LENC CTRL13	0x07	RW	Bit[7:4]: Not used Bit[3:0]: grn_a2
0x580E	LENC CTRL14	0xC2	RW	Bit[7:0]: grn_b1
0x580F	LENC CTRL15	0x08	RW	Bit[7:4]: Not used Bit[3:0]: grn_b2
0x5810	LENC CTRL16	0x03	RW	Bit[7:3]: Not used Bit[2:0]: blu_x0[10:8]
0x5811	LENC CTRL17	0xD0	RW	Bit[7:0]: blu_x0[7:0]
0x5812	LENC CTRL18	0x02	RW	Bit[7:3]: Not used Bit[2:0]: blu_y0[10:8]
0x5813	LENC CTRL19	0x56	RW	Bit[7:0]: blu_y0[7:0]
0x5814	LENC CTRL20	0x22	RW	Bit[7]: Not used Bit[6:0]: blu_a1
0x5815	LENC CTRL21	0x07	RW	Bit[7:4]: Not used Bit[3:0]: blu_a2
0x5816	LENC CTRL22	0xC2	RW	Bit[7:0]: blu_b1
0x5817	LENC CTRL23	0x08	RW	Bit[7:4]: Not used Bit[3:0]: blu_b2
0x5818	LENC CTRL24	0x04	RW	Bit[7:4]: Not used Bit[3]: seed_reset_en Reset seed when VSYNC occurs Bit[2]: round_en Generates random round bit Bit[1]: coef_man_en Coefficient manual mode enable signal. When this is set, the used coefficient is coef_man Bit[0]: gain_coef_en 0: Disable 1: Enable
0x5819	LENC CTRL28	0x80	RW	Bit[7:0]: min_coefficient Range is [0,128]
0x581A	LENC CTRL29	0x06	RW	Bit[7:0]: low_gain_thresh Low real gain threshold
0x581B	LENC CTRL30	0x0C	RW	Bit[7:0]: high_gain_thresh High real gain threshold
0x581C	LENC CTRL31	0x80	RW	Bit[7:0]: coef_man Manual coefficient Range is [0,128]

7.19 DGC control [0x5A00 - 0x5A03]

table 7-19 digital gain correction (DGC) registers

address	register name	default value	R/W	description
0x5A00	DGC CTRL00	0x05	RW	Bit[7:3]: Not used Bit[2]: Debug control Bit[1]: dgc_man_en Bit[0]: dgc_en
0x5A01	NOT USED	–	–	Not Used
0x5A02	DGC CTRL02	0x02	RW	Bit[7:2]: Not used Bit[1:0]: dgc_man[9:8]
0x5A03	DGC CTRL03	0x00	RW	Bit[7:0]: dgc_man[7:0]

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 +125°C
supply voltage (with respect to ground)	V_{DD-A}	4.5V
	V_{DD-D}	3V
	V_{DD-IO}	4.5V
electro-static discharge (ESD)	human body model	2000V
	machine model	200V
all input/output voltages (with respect to ground)		-0.3V to V_{DD-IO} + 1V
I/O current on any input or output pin		±200 mA
peak solder temperature (10 second dwell time)		245°C

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

8.2 functional temperature

table 8-2 functional temperature

parameter	range
operating temperature[a]	-20°C to +70°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 (-20°C < T_J < 70°C)[a b]

symbol	parameter	min	typ	max	unit
supply					
V_{DD-A}	supply voltage (analog)	2.6	2.8	3.0	V
V_{DD-DO}	supply voltage (digital I/O)	1.7	1.8	3.0	V
V_{DD-D}	supply voltage (digital core)[c]	1.425	1.5	1.575	V
V_{DD-E}	supply voltage (MIPI)	1.425	1.5	1.575	V
I_{DD-A}	active (operating) current 1920 x 1080 @ 30 fps		28	50	mA
I_{DD-DO}			46	70	mA
I_{DD-A}	active (operating) current 1280 x 720 @ 30 fps		28		mA
I_{DD-DO}			34		mA
I_{DD-A}	active (operating) current VGA_bin @ 90 fps		32		mA
I_{DD-DO}			38		mA
I_{DD-A}	active (operating) current VGA_skip @ 90 fps		28		mA
I_{DD-DO}			36		mA
$I_{DDS-SCCB}$	standby current[d,e]		15	75	µA
$I_{DDS-PWDN}$			15	75	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}[f]	SCL and SDA	-0.5	0	0.54	V
V_{IH}[f]	SCL and SDA	1.26	1.8	2.3	V

a. active current based on DVP output

b. with MIPI function, active current needs an additional maximum 20 mA on DVDD (MIPI power, EVDD is from DVDD)

c. when internal regulator is bypassed

d. external clock is stopped during measurement

e. max standby current based on V_{DD-A} = 2.8V, V_{DD-DO} = 1.8V and is measured at room temperature

f. based on DOVDD = 1.8V

8.4 AC characteristics

table 8-4 AC characteristics (T_A = 25°C, V_{DD-A} = 2.8V)

symbol	parameter	min	typ	max	unit
ADC parameters					
B	analog bandwidth		72		MHz
DLE	DC differential linearity error		0.5		LSB
ILE	DC integral linearity error		1		LSB
	settling time for hardware reset			<1	ms
	settling time for software reset			<1	ms
	settling time for resolution mode change			<1	ms
	settling time for register setting			<300	ms

table 8-5 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 the internal PLL

9 mechanical specifications

9.1 physical specifications

figure 9-1 package specifications

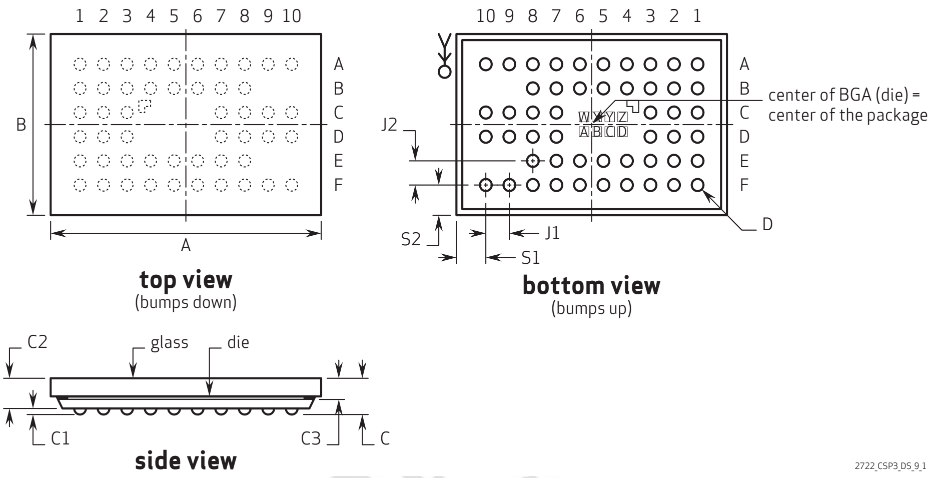
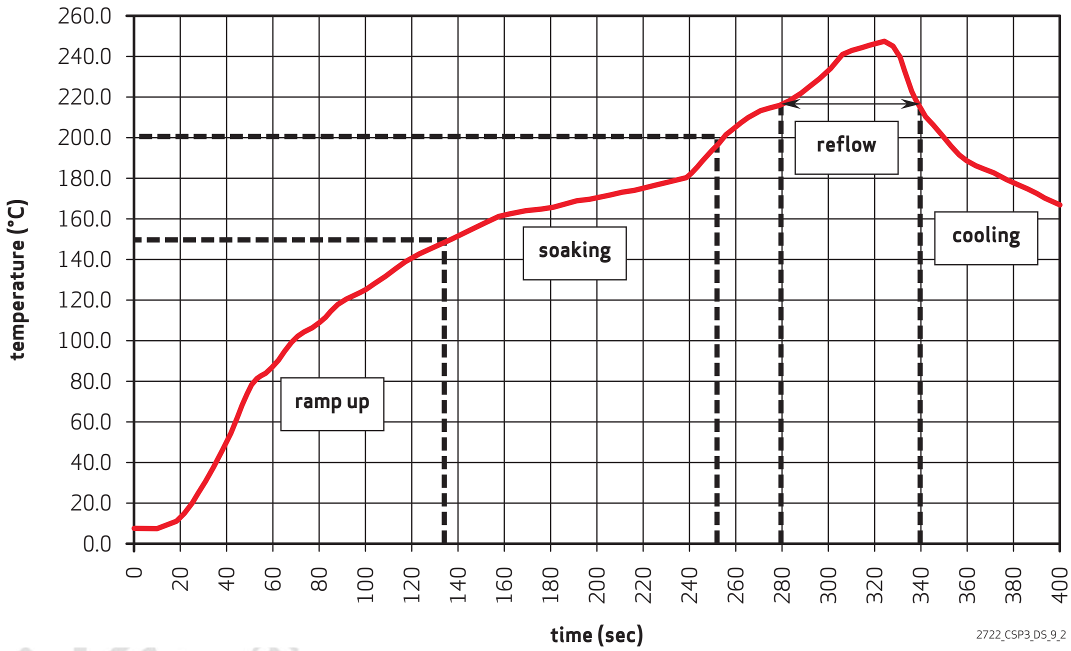


table 9-1 package dimensions

parameter	symbol	min	typ	max	unit
package body dimension x	A	5360	5385	5410	µm
package body dimension y	B	3260	3285	3310	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
ball diameter	D	220	250	280	µm
total pin count	N		50 (2 NC)		
pin count x-axis	N1		10		
pin count y-axis	N2		6		
pins pitch x-axis	J1		500		µm
pins pitch y-axis	J2		500		µm
edge-to-pin center distance analog x	S1	413	443	473	µm
edge-to-pin center distance analog y	S2	363	393	423	µm

9.2 IR reflow specifications

figure 9-2 IR reflow ramp rate requirements



note
The OV2722/OV2222 uses a lead-free package.

table 9-2 reflow conditions[a][b]

zone	description	exposure
ramp up	heating from room temperature to 150°C	temperature slope ≤ 3°C per second
soaking	heating from 150°C to 200°C	90 ~ 150 seconds
reflow	temperature higher than 217°C	30 ~ 120 seconds
peak	maximum temperature in SMT	245°C
cooling	cooling from 217°C to room temperature	temperature slope ≤ 6°C per second

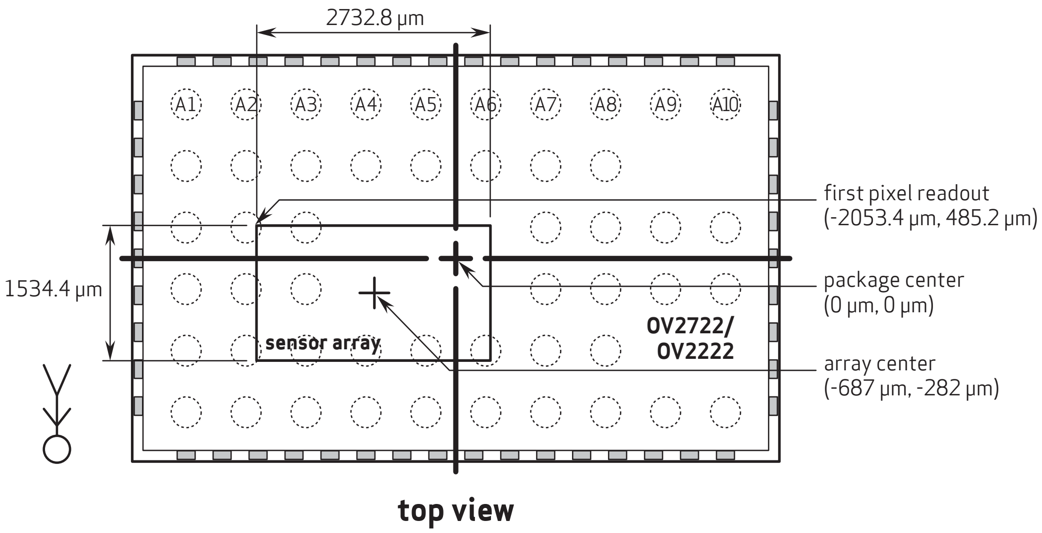
[a] maximum number of reflow cycles = 3

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

10 optical specifications

10.1 sensor array center

figure 10-1 sensor array center



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

note 2 as most optical assemblies invert and mirror the image, the chip is typically mounted with pin A1 to A10 oriented down on the PCB.

2722_CSP3_DS_10_1

10.2 lens chief ray angle (CRA)

figure 10-2 chief ray angle (CRA)

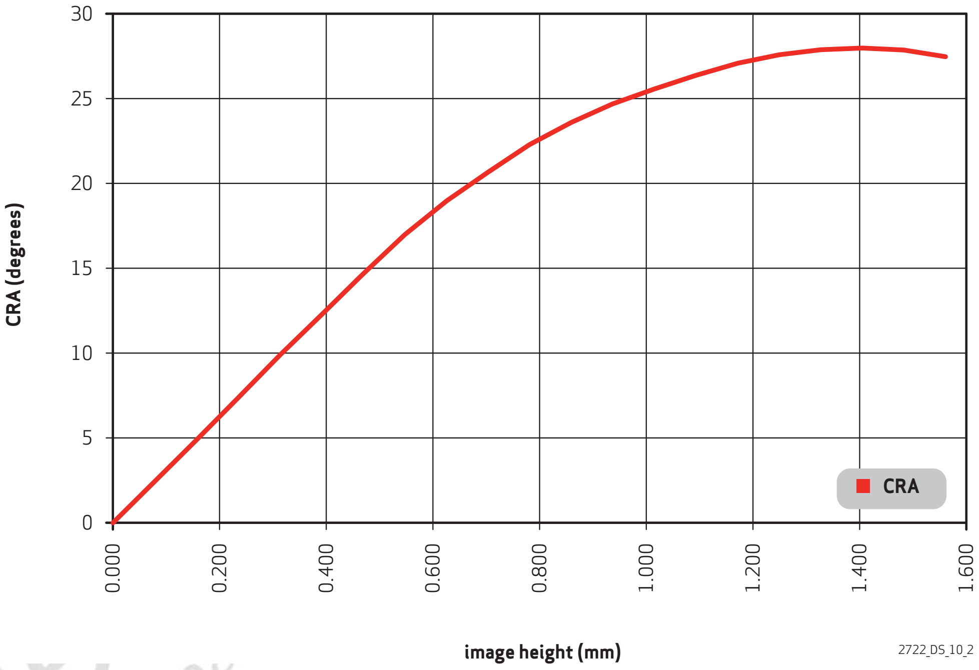


table 10-1 CRA versus image height plot (sheet 1 of 2)

field (%)	image height (mm)	CRA (degrees)
0.00	0.000	0.0
0.05	0.077	2.5
0.10	0.154	4.9
0.15	0.231	7.4
0.20	0.308	9.9
0.25	0.386	12.3
0.30	0.463	14.7
0.35	0.540	17.0
0.40	0.617	19.0
0.45	0.694	20.7
0.50	0.771	22.3

table 10-1 CRA versus image height plot (sheet 2 of 2)

field (%)	image height (mm)	CRA (degrees)
0.55	0.848	23.6
0.60	0.925	24.7
0.65	1.002	25.6
0.70	1.079	26.4
0.75	1.157	27.1
0.80	1.234	27.6
0.85	1.311	27.9
0.90	1.388	28.0
0.95	1.465	27.9
1.00	1.542	27.5

revision history

version 1.0 12.09.2011

- initial release

version 2.0 05.31.2012

- updated to datasheet from Preliminary Specification to Product Specification
- changed "OmniBSI" to "OmniBSI+" throughout entire datasheet
- on page i, under applications, added "ultra books" and "smart phones" and removed "toys"
- on page i, under key specifications, replaced TBDs
- in table 1-1, removed "or video data input" from description of D1, D0, and D4 pins, and "3.3" voltage from PVDD, AVDD pins
- in section 2, updated figure 2-1 and changed table 2-1 table header from "scaling method" to "methodology"
- in sub-section, 2.4.1, changed first sentence to "A PLL inside the chip generates a maximum 72 MHz system clock from a 6~27 MHz input clock."
- in sub-section 2.5.1, changed list number 3 description from "...must go high..." to "...must be high...", list number 7 description from "...should provide..." to "...should be provided..." and list number 8 description from "...during entire period..." to "during the entire period..."
- in sub-section 2.5.2, changed list number 5 description from "...powers are..." to "...power supplies are..." list number 7 description from "...DOVDD is up" to "...DOVDD is stable", and list number 8 description from "...should provide..." to "...should be provided..."
- in section 3.2, changed section title from "binning" to "subsampling" and replaced section description with new description
- in section 4.2, rewrote last sentence in section description
- in section 4, removed sub-section 4.4.1 title
- in section 5, replaced section descriptions with new descriptions for sections 5.3 and 5.4
- in section 6.7, replaced section description with new description
- in table 8-3, replaced TBDs

version 2.1 07.13.2012

- added B&W part number
- in section 2.7, added sentence, "To save power in MIPI mode, power down MIPI PHY by setting register 0x3018[4:3] to 2'b11 before sensor power down."
- in table 3-1, added table footnote a, "both vertical and horizontal are binning average"
- in table 4-4, changed description for register bits 0x3503[5:4] to match the description for the same register bits shown in table 7-3
- in section 4.5, changed second and third sentences to "...when the scene illumination is getting darker. Similarly, under brighter conditions, the action ..."
- removed section 4.5.1, LAEC

- completely re-wrote section 4.7, one-time programmable (OTP) memory

version 2.11 08.10.2012

- in table 4-8, changed OTP bits in second row from "256 - 40" to "255 - 40"

version 2.2 09.26.2012

- changed figure 2-2
- in table 4-6, changed description of register 0x5680 to Bit[7:3]: Reserved, Bit[2:0]: x_start[10:8]
- in table 4-6, changed default values of registers 0x5681, 0x5682, 0x5683, 0x5684, 0x5685, 0x5686, and 0x5687 to 0x00, 0x00, 0x00, 0x10, 0xA0, 0x0C, and 0x78, respectively
- in table 4-6, changed description of register 0x5684 to Bit[7:3]: Reserved, Bit[2:0]: Window X[10:8]
- in table 4-6, changed description of register 0x5686 to Bit[7:3]: Reserved, Bit[2:0]: Window Y[10:8]
- in table 5-7, changed default value for register 0x5690 to 0x02

defining the future of digital imaging™

OmniVision Technologies, Inc.

UNITED STATES

4275 Burton Drive
Santa Clara, CA 95054

tel: + 1 408 567 3000

fax: + 1 408 567 3001

email: salesamerican@ovt.com

UNITED KINGDOM

Hampshire + 44 1256 744 610

GERMANY

Munich +49 89 63 81 99 88

INDIA

Bangalore +91 988 008 0140

CHINA

Beijing + 86 10 6580 1690
Shanghai + 86 21 6175 9888
Shenzhen + 86 755 8384 9733
Hong Kong + 852 2403 4011

JAPAN

Yokohama +81 45 478 7977
Osaka +81 6 4964 2606

KOREA

Seoul + 82 2 3478 2812

SINGAPORE +65 6220 1335

TAIWAN

Taipei +886 2 2657 9800
Hsinchu +886 3 6110933

website: **www.ovt.com**