

**datasheet**

PRELIMINARY SPECIFICATION

1/4" color CMOS QSXGA (5 megapixel) image sensor  
with OmniBSI™ technology

**OV5640**

Confidential for  
xin jin hui

## Copyright © 2010 OmniVision Technologies, Inc. All rights reserved.

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 QSXGA (5 megapixel) image sensor with OmniBSI™ technology**

datasheet (CSP3)  
PRELIMINARY SPECIFICATION

version 1.0  
february 2010

To learn more about OmniVision Technologies, visit [www.ovt.com](http://www.ovt.com).  
OmniVision Technologies is publicly traded on NASDAQ under the symbol OVTI.

**OV5640**

color CMOS QSXGA (5 megapixel) image sensor with OmniBSI™ technology

Confidential for  
xin jin hui

## applications

- cellular phones
- toys
- PC multimedia
- digital still cameras

## ordering information

- **OV05640-A71A** (color, lead-free)  
71-pin CSP3

## features

- 1.4  $\mu\text{m}$  x 1.4  $\mu\text{m}$  pixel with OmniBSI technology for high performance (high sensitivity, low crosstalk, low noise, improved quantum efficiency)
- optical size of 1/4"
- automatic image control functions: automatic exposure control (AEC), automatic white balance (AWB), automatic band filter (ABF), automatic 50/60 Hz luminance detection, 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: color saturation, hue, gamma, sharpness (edge enhancement), lens correction, defective pixel canceling, and noise canceling
- support for output formats: RAW RGB, RGB565/555/444, CCIR656, YUV422/420, YCbCr422, and compression
- support for video or snapshot operations
- support for LED and flash strobe mode
- support for internal and external frame synchronization for frame exposure mode
- support for horizontal and vertical sub-sampling, binning
- support for minimizing artifacts on binned image
- support for data compression output
- support for anti-shake
- standard serial SCCB interface
- digital video port (DVP) parallel output interface and dual lane MIPI output interface
- embedded 1.5V regulator for core power
- programmable I/O drive capability, I/O tri-state configurability
- support for black sun cancellation
- support for images sizes: 5 megapixel, and any arbitrary size scaling down from 5 megapixel
- support for auto focus control (AFC) with embedded AF VCM driver
- embedded microcontroller
- suitable for module size of 8.5 x 8.5 x <6mm with both CSP and RW packaging

## key specifications

- **active array size:** 2592 x 1944
- **power supply:**
  - core: 1.5V  $\pm$  5% (with embedded 1.5V regulator)
  - analog: 2.6 ~ 3.0V (2.8V typical)
  - I/O: 1.8V / 2.8V
- **power requirements:**
  - active: TBD
  - standby: TBD
- **temperature range:**
  - operating: -30°C to 70°C (see [table 8-1](#))
  - stable image: 0°C to 50°C (see [table 8-1](#))
- **output formats:** 8-/10-bit RGB RAW output
- **lens size:** 1/4"
- **lens chief ray angle:** 24° (see [figure 10-2](#))
- **input clock frequency:** 6~27 MHz
- **S/N ratio:** TBD
- **dynamic range:** TBD
- **maximum image transfer rate:**
  - QSXGA (2592x1944): 15 fps
  - 1080p: 30 fps
  - 1280x960: 45 fps
  - 720p: 60 fps
  - VGA (640x480): 90 fps
  - QVGA (320x240): 120 fps
- **sensitivity:** TBD
- **shutter:** rolling shutter / frame exposure
- **maximum exposure interval:** 1964 x  $t_{\text{ROW}}$
- **pixel size:** 1.4  $\mu\text{m}$  x 1.4  $\mu\text{m}$
- **well capacity:** TBD
- **dark current:** TBD
- **fixed pattern noise (FPN):** TBD
- **image area:** 3673.6  $\mu\text{m}$  x 2738.4  $\mu\text{m}$
- **package dimensions:** 5985  $\mu\text{m}$  x 5835  $\mu\text{m}$

**OV5640**

color CMOS QXGA (5 megapixel) image sensor with OmniBSI™ technology

Confidential for  
xin jin hui

## table of contents

|  |            |
|--|------------|
| <b>1 signal descriptions</b>                 | <b>1-1</b> |
| <b>2 system level description</b>            | <b>2-1</b> |
| 2.1 overview                                 | 2-1        |
| 2.2 architecture                             | 2-1        |
| 2.3 format and frame rate                    | 2-4        |
| 2.4 I/O control                              | 2-5        |
| 2.5 system clock control                     | 2-7        |
| 2.6 SCCB interface                           | 2-7        |
| 2.7 power up sequence                        | 2-10       |
| 2.7.1 power up with internal DVDD            | 2-10       |
| 2.7.2 power up with external DVDD source     | 2-11       |
| 2.8 reset                                    | 2-12       |
| 2.9 standby and sleep                        | 2-12       |
| <b>3 block level description</b>             | <b>3-1</b> |
| 3.1 pixel array structure                    | 3-1        |
| 3.2 binning                                  | 3-2        |
| 3.3 VCM driver                               | 3-3        |
| 3.3.1 output current control mode            | 3-3        |
| <b>4 image sensor core digital functions</b> | <b>4-1</b> |
| 4.1 mirror and flip                          | 4-1        |
| 4.2 image windowing                          | 4-2        |
| 4.3 test pattern                             | 4-5        |
| 4.4 50/60Hz detection                        | 4-5        |
| 4.4.1 overview                               | 4-5        |
| 4.5 AEC/AGC algorithms                       | 4-6        |
| 4.5.1 overview                               | 4-6        |
| 4.5.2 average-based algorithm                | 4-6        |
| 4.6 AEC/AGC steps                            | 4-11       |
| 4.6.1 auto exposure control (AEC)            | 4-11       |
| 4.6.2 manual exposure control                | 4-12       |
| 4.6.3 auto gain control (AGC)                | 4-12       |

|          |  |            |
|----------|--|------------|
| 4.6.4    | manual gain control                                    | 4-12       |
| 4.7      | black level calibration (BLC)                          | 4-13       |
| 4.8      | light frequency selection                              | 4-14       |
| 4.9      | digital gain   | 4-14       |
| 4.10     | strobe flash and frame exposure                        | 4-15       |
| 4.10.1   | strobe flash control                                   | 4-15       |
| 4.10.2   | frame exposure (FREX) mode                             | 4-17       |
| 4.10.3   | FREX strobe flash control                              | 4-18       |
| 4.11     | one time programmable (OTP) memory                     | 4-19       |
| <b>5</b> | <b>image sensor processor digital functions</b>        | <b>5-1</b> |
| 5.1      | ISP general controls                                   | 5-1        |
| 5.2      | lens correction (LENC)                                 | 5-4        |
| 5.3      | auto white balance (AWB)                               | 5-6        |
| 5.4      | raw gamma  | 5-8        |
| 5.5      | defect pixel cancellation (DPC)                        | 5-9        |
| 5.6      | color interpolation (CIP)                              | 5-10       |
| 5.7      | color matrix (CMX)                                     | 5-11       |
| 5.8      | UV average   | 5-12       |
| 5.9      | scaling  | 5-12       |
| 5.10     | UV adjust  | 5-13       |
| 5.10.1   | manual mode  | 5-13       |
| 5.10.2   | auto mode  | 5-13       |
| 5.11     | special digital effects (SDE)                          | 5-14       |
| 5.12     | ISP format   | 5-15       |
| 5.13     | draw window  | 5-15       |
| <b>6</b> | <b>image sensor output interface digital functions</b> | <b>6-1</b> |
| 6.1      | compression engine                                     | 6-1        |
| 6.1.1    | compression mode 1 timing                              | 6-1        |
| 6.1.2    | compression mode 2 timing                              | 6-1        |
| 6.1.3    | compression mode 3 timing                              | 6-2        |
| 6.1.4    | compression mode 4 timing                              | 6-2        |
| 6.1.5    | compression mode 5 timing                              | 6-3        |
| 6.1.6    | compression mode 6 timing                              | 6-3        |



|  |             |
|--|-------------|
| 6.1.7 compression mode control                 | 6-4         |
| 6.2 system control                             | 6-5         |
| 6.3 microcontroller unit (MCU)                 | 6-8         |
| 6.4 frame control (FC)                         | 6-9         |
| 6.5 format description                         | 6-10        |
| 6.6 digital video port (DVP)                   | 6-15        |
| 6.6.1 overview                                 | 6-15        |
| 6.6.2 DVP timing                               | 6-18        |
| 6.7 mobile industry processor interface (MIPI) | 6-20        |
| <b>7 register tables</b>                       | <b>7-1</b>  |
| <b>8 operating specifications</b>              | <b>8-1</b>  |
| 8.1 absolute maximum ratings                   | 8-1         |
| 8.2 functional temperature                     | 8-1         |
| 8.3 DC characteristics                         | 8-2         |
| 8.4 AC characteristics                         | 8-4         |
| <b>9 mechanical specifications</b>             | <b>9-1</b>  |
| 9.1 physical specifications                    | 9-1         |
| 9.2 IR reflow specifications                   | 9-2         |
| <b>10 optical specifications</b>               | <b>10-1</b> |
| 10.1 sensor array center                       | 10-1        |
| 10.2 lens chief ray angle (CRA)                | 10-2        |

**OV5640**

color CMOS QSXGA (5 megapixel) image sensor with OmniBSI™ technology

Confidential for  
xin jin hui

## list of figures

|             |   |      |
|-------------|---|------|
| figure 1-1  | pin diagram   | 1-4  |
| figure 2-1  | OV5640 block diagram  | 2-2  |
| figure 2-2  | reference design schematic  | 2-3  |
| figure 2-3  | power up timing with internal DVDD and I2C access during power up period        | 2-10 |
| figure 2-4  | power up timing with external DVDD source and I2C access during power up period | 2-11 |
| figure 3-1  | sensor array region color filter layout   | 3-1  |
| figure 3-2  | example of 2x2 binning  | 3-2  |
| figure 3-3  | VCM block diagram   | 3-3  |
| figure 3-4  | 1/4 to 3/4 scale settling time (directly jump mode, VDD = 3.0V)                 | 3-6  |
| figure 3-5  | sink current vs. code (VDD = 3.0V, reg 0x30A5 = 0x05, VCM resistance = 23ohms)  | 3-6  |
| figure 4-1  | mirror and flip samples   | 4-1  |
| figure 4-2  | image windowing   | 4-2  |
| figure 4-3  | image windowing configuration   | 4-3  |
| figure 4-4  | test pattern  | 4-5  |
| figure 4-5  | desired convergence   | 4-7  |
| figure 4-6  | average-based window definition   | 4-9  |
| figure 4-7  | xenon flash mode  | 4-15 |
| figure 4-8  | LED 1 & 2 mode - one pulse output   | 4-16 |
| figure 4-9  | LED 1 & 2 mode - multiple pulse output  | 4-16 |
| figure 4-10 | LED 3 mode  | 4-17 |
| figure 4-11 | FREX modes  | 4-17 |
| figure 5-1  | UV adjust graph   | 5-13 |
| figure 6-1  | compression mode 1 timing   | 6-1  |
| figure 6-2  | compression mode 2 timing   | 6-1  |
| figure 6-3  | compression mode 3 timing   | 6-2  |
| figure 6-4  | compression mode 4 timing   | 6-2  |
| figure 6-5  | compression mode 5 timing   | 6-3  |
| figure 6-6  | compression mode 6 timing   | 6-3  |
| figure 6-7  | DVP timing diagram  | 6-18 |
| figure 8-1  | SCCB interface timing   | 8-5  |
| figure 9-1  | package specifications  | 9-1  |
| figure 9-2  | IR reflow ramp rate requirements  | 9-2  |

|             |                       |      |
|-------------|-----------------------|------|
| figure 10-1 | sensor array center   | 10-1 |
| figure 10-2 | chief ray angle (CRA) | 10-2 |

Confidential for  
xin jin hui

## list of tables

|            |   |      |
|------------|---|------|
| table 1-1  | signal descriptions                                   | 1-1  |
| table 2-1  | format and frame rate                                 | 2-4  |
| table 2-2  | driving capability and direction control for I/O pads | 2-5  |
| table 2-3  | group sharing registers                               | 2-7  |
| table 2-4  | group write register                                  | 2-8  |
| table 3-1  | binning-related registers                             | 3-2  |
| table 3-2  | VCM driver control                                    | 3-4  |
| table 3-3  | VCM control registers                                 | 3-4  |
| table 3-4  | single step mode                                      | 3-5  |
| table 3-5  | multi-code step mode                                  | 3-5  |
| table 4-1  | mirror and flip registers                             | 4-1  |
| table 4-2  | image windowing registers                             | 4-3  |
| table 4-3  | test pattern selection control                        | 4-5  |
| table 4-4  | AEC/AGC control functions                             | 4-6  |
| table 4-5  | AEC/AGC control functions                             | 4-8  |
| table 4-6  | timing control functions                              | 4-9  |
| table 4-7  | BLC control functions                                 | 4-13 |
| table 4-8  | light frequency registers                             | 4-14 |
| table 4-9  | flashlight modes                                      | 4-15 |
| table 4-10 | FREX strobe control functions                         | 4-18 |
| table 4-11 | OTP control functions                                 | 4-19 |
| table 5-1  | ISP general control registers                         | 5-1  |
| table 5-2  | LENC control registers                                | 5-4  |
| table 5-3  | AWB control registers                                 | 5-6  |
| table 5-4  | raw gamma control registers                           | 5-8  |
| table 5-5  | DPC control registers                                 | 5-9  |
| table 5-6  | CIP control registers                                 | 5-10 |
| table 5-7  | CMX control registers                                 | 5-11 |
| table 5-8  | UV average register                                   | 5-12 |
| table 5-9  | UV average register                                   | 5-12 |
| table 5-10 | SDE control registers                                 | 5-14 |
| table 5-11 | ISP format control registers                          | 5-15 |

|            |   |      |
|------------|---|------|
| table 5-12 | draw window registers                       | 5-15 |
| table 6-1  | compression control registers               | 6-4  |
| table 6-2  | system control registers                    | 6-5  |
| table 6-3  | MCU control registers                       | 6-8  |
| table 6-4  | FC control registers                        | 6-9  |
| table 6-5  | FORMAT control registers                    | 6-10 |
| table 6-6  | DVP control registers                       | 6-15 |
| table 6-7  | DVP timing specifications                   | 6-18 |
| table 6-8  | MIPI transmitter registers                  | 6-20 |
| table 7-1  | system and IO pad control registers         | 7-1  |
| table 7-2  | SCCB control registers                      | 7-7  |
| table 7-3  | SRB control registers                       | 7-9  |
| table 7-4  | AWB gain control registers                  | 7-10 |
| table 7-5  | AEC/AGC control functions                   | 7-10 |
| table 7-6  | VCM control registers                       | 7-11 |
| table 7-7  | timing control registers                    | 7-12 |
| table 7-8  | AEC/AGC power down domain control registers | 7-13 |
| table 7-9  | STROBE registers                            | 7-16 |
| table 7-10 | 5060HZ detector registers                   | 7-17 |
| table 7-11 | OTP control functions                       | 7-19 |
| table 7-12 | BIST registers                              | 7-20 |
| table 7-13 | MC registers                                | 7-22 |
| table 7-14 | BLC registers                               | 7-24 |
| table 7-15 | frame control registers                     | 7-26 |
| table 7-16 | FORMAT control registers                    | 7-27 |
| table 7-17 | JPEG control registers                      | 7-31 |
| table 7-18 | VFIFO registers                             | 7-34 |
| table 7-19 | DVP control registers                       | 7-35 |
| table 7-20 | MIPI transmitter registers                  | 7-38 |
| table 7-21 | ISP frame control registers                 | 7-41 |
| table 7-22 | ISP top control registers                   | 7-41 |
| table 7-23 | AWB registers                               | 7-46 |
| table 7-24 | CIP control registers                       | 7-48 |
| table 7-25 | CMX control registers                       | 7-49 |
| table 7-26 | gamma control registers                     | 7-50 |

|            |  |      |
|------------|--|------|
| table 7-27 | SDE control registers  | 7-51 |
| table 7-28 | SCALE registers  | 7-53 |
| table 7-29 | AVG registers  | 7-54 |
| table 7-30 | LENC control registers   | 7-56 |
| table 7-31 | AFC control registers  | 7-61 |
| table 8-1  | absolute maximum ratings   | 8-1  |
| table 8-2  | functional temperature   | 8-1  |
| table 8-3  | DC characteristics ( $-20^{\circ}\text{C} < T_A < 70^{\circ}\text{C}$ )      | 8-2  |
| table 8-4  | AC characteristics ( $T_A = 25^{\circ}\text{C}$ , $V_{DD-A} = 2.8\text{V}$ ) | 8-4  |
| table 8-5  | timing characteristics   | 8-4  |
| table 8-6  | SCCB interface timing specifications   | 8-5  |
| table 9-1  | package dimensions   | 9-1  |
| table 9-2  | reflow conditions  | 9-2  |
| table 10-1 | CRA versus image height plot   | 10-2 |

**OV5640**

color CMOS QSXGA (5 megapixel) image sensor with OmniBSI™ technology

Confidential for  
xin jin hui



# 1 signal descriptions

**table 1-1** lists the signal descriptions and their corresponding pin numbers for the OV5640 image sensor. The package information is shown in **section 9**.

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

| pin number | signal name | pin type  | description   |
|------------|-------------|-----------|---|
| A2         | VCMSINK     | I/O       | analog I/O  |
| A3         | AVDD        | power     | power for analog circuit                                  |
| A4         | AGND        | ground    | ground for analog circuit                                 |
| A6         | DGND        | ground    | ground for digital circuit                                |
| A7         | NC          | –         | no connect  |
| A8         | NC          | –         | no connect  |
| A9         | NC          | –         | no connect  |
| A10        | AGND        | ground    | ground for analog circuit                                 |
| A11        | NC          | –         | no connect  |
| B1         | DGND        | ground    | ground for digital circuit                                |
| B2         | VCMSINK     | I/O       | analog I/O  |
| B3         | VCMGND      | I/O       | analog I/O  |
| B4         | VCMGND      | I/O       | analog I/O  |
| B5         | DVDD        | power     | power for digital circuit                                 |
| B6         | NC          | –         | no connect  |
| B7         | NC          | –         | no connect  |
| B8         | NC          | –         | no connect  |
| B9         | DVDD        | power     | power for digital circuit                                 |
| B10        | AVDD        | power     | power for analog circuit                                  |
| B11        | VN          | reference | internal analog reference                                 |
| C1         | DVDD        | power     | power for digital circuit                                 |
| C2         | DGND        | ground    | ground for digital circuit                                |
| C3         | NC          | –         | no connect  |
| C10        | VH          | reference | internal analog reference                                 |
| C11        | AVDD        | power     | power for analog circuit                                  |
| D1         | PWDN        | input     | power down (active high with internal pull-down resistor) |

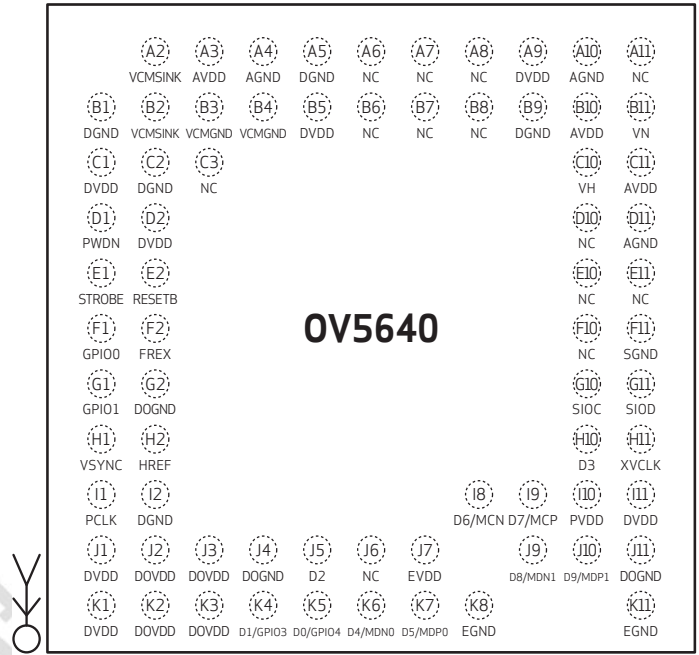
table 1-1 signal descriptions (sheet 2 of 3)

| pin number | signal name | pin type | description  |
|------------|-------------|----------|--|
| D2         | DVDD        | power    | power for digital circuit                                  |
| D10        | NC          | –        | no connect   |
| D11        | AGND        | ground   | ground for analog circuit                                  |
| E1         | STROBE      | I/O      | strobe output  |
| E2         | RESETB      | input    | reset (active low with internal pull-up resistor)          |
| E10        | NC          | –        | no connect   |
| E11        | NC          | –        | no connect   |
| F1         | GPIO0       | I/O      | GPIO port 0  |
| F2         | FREX        | I/O      | frame exposure / mechanical shutter                        |
| F10        | NC          | –        | no connect   |
| F11        | SGND        | ground   | ground for sensor circuit                                  |
| G1         | GPIO1       | I/O      | GPIO port 1  |
| G2         | DOGND       | ground   | ground for I/O circuit                                     |
| G10        | SIOC        | input    | SCCB input clock   |
| G11        | SIOD        | I/O      | SCCB data  |
| H1         | VSYNC       | I/O      | DVP VSYNC output   |
| H2         | HREF        | I/O      | DVP HREF output  |
| H10        | D3          | I/O      | DVP data output port 3                                     |
| H11        | XVCLK       | input    | system input clock   |
| I1         | PCLK        | I/O      | DVP PCLK output  |
| I2         | DGND        | ground   | ground for digital circuit                                 |
| I8         | D6/MCN      | I/O      | DVP data output port 6/ MIPI TX clock lane negative output |
| I9         | D7/MCP      | I/O      | DVP data output port 7/ MIPI TX clock lane positive output |
| I10        | PVDD        | power    | power for PLL circuit                                      |
| I11        | DVDD        | power    | power for digital circuit                                  |
| J1         | DVDD        | power    | power for digital circuit                                  |
| J2         | DOVDD       | power    | power for I/O circuit                                      |
| J3         | DOVDD       | power    | power for I/O circuit                                      |
| J4         | DOGND       | ground   | ground for I/O circuit                                     |
| J5         | D2          | I/O      | DVP data output port 2                                     |

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

| pin number | signal name | pin type  | description  |
|------------|-------------|-----------|--|
| J6         | NC          | –         | no connect   |
| J7         | EVDD        | reference | power for MIPI TX circuit  |
| J9         | D8/MDN1     | I/O       | DVP data output port 8/ MIPI TX second data lane negative output |
| J10        | D9/MDP1     | I/O       | DVP data output port 9/ MIPI TX second data lane positive output |
| J11        | DOGND       | ground    | ground for I/O circuit   |
| K1         | DVDD        | power     | power for digital circuit  |
| K2         | DOVDD       | power     | power for I/O circuit  |
| K3         | DOVDD       | power     | power for I/O circuit  |
| K4         | D1/GPIO3    | I/O       | DVP data output port 1/ GPIO port 3                              |
| K5         | D0/GPIO2    | I/O       | DVP data output port 0/ GPIO port 2                              |
| K6         | D4/MDN0     | I/O       | DVP data output port 4/ MIPI TX first data lane negative output  |
| K7         | D5/MDP0     | I/O       | DVP data output port 5/ MIPI TX first data lane positive output  |
| K8         | EGND        | ground    | ground for MIPI TX circuit                                       |
| K11        | EGND        | ground    | ground for MIPI TX circuit                                       |

figure 1-1 pin diagram



5640 CSP DS 1.1

## 2 system level description

### 2.1 overview

The OV5640 (color) image sensor is a low voltage, high-performance, 1/4-inch 5 megapixel CMOS image sensor that provides the full functionality of a single chip 5 megapixel (2592x1944) camera using OmniBSI™ technology in a small footprint package. It provides full-frame, sub-sampled, windowed or arbitrarily scaled 8-bit/10-bit images in various formats via the control of the Serial Camera Control Bus (SCCB) interface.

The OV5640 has an image array capable of operating at up to 15 frames per second (fps) in 5 megapixel resolution with complete user control over image quality, formatting and output data transfer. All required image processing functions, including exposure control, gamma, white balance, color saturation, hue control, defective pixel canceling, noise canceling, etc., are programmable through the SCCB interface or embedded microcontroller. The OV5640 also includes a compression engine for increased processing power. In addition, Omnivision image sensors use proprietary sensor technology to improve image quality by reducing or eliminating common lighting/electrical sources of image contamination, such as fixed pattern noise, smearing, etc., to produce a clean, fully stable, color image.

The OV5640 has an embedded microcontroller, which can be combined with an internal autofocus engine and programmable general purpose I/O modules (GPIO) for external autofocus control. It also provides an anti-shake function with an internal anti-shake engine. For identification and storage purposes, the OV5640 also includes a one-time programmable (OTP) memory.

The OV5640 supports both a digital video parallel port and a serial MIPI port.

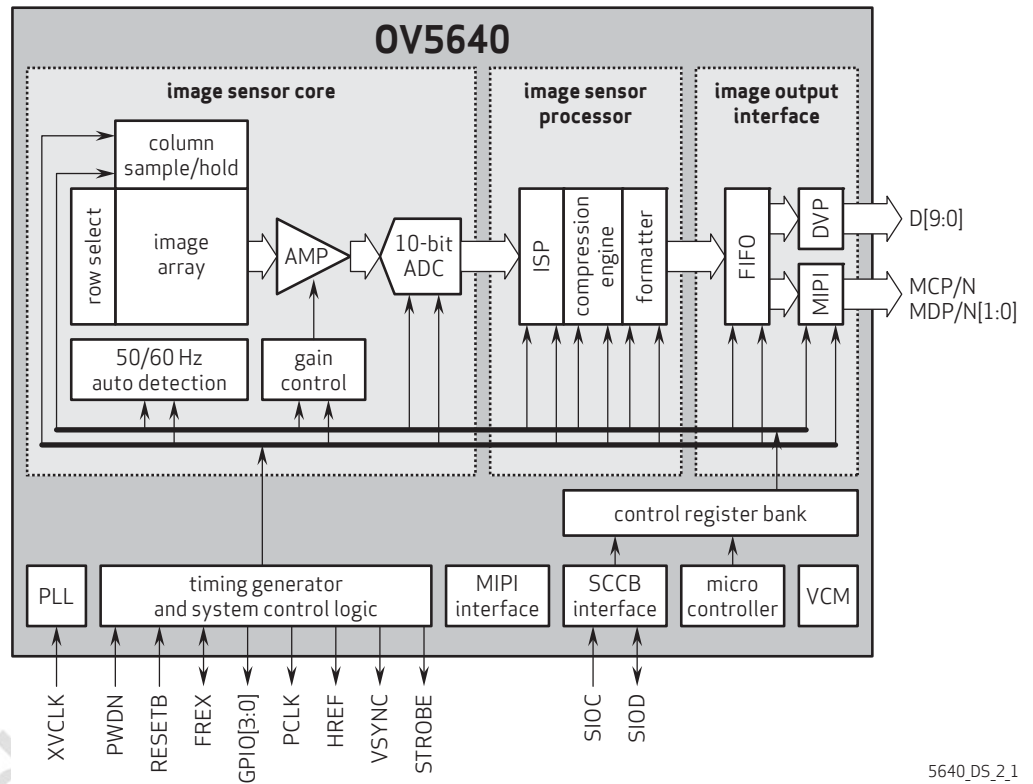
### 2.2 architecture

The OV5640 sensor core generates streaming pixel data at a constant frame rate, indicated by HREF and VSYNC. **figure 2-1** shows the functional block diagram of the OV5640 image sensor.

The timing generator outputs signals to access the rows of the image array, precharging and sampling the rows of the array in series. In the time between pre-charging and sampling a row, the charge in the pixels decreases with the time exposed to the incident light. This is known as exposure time.

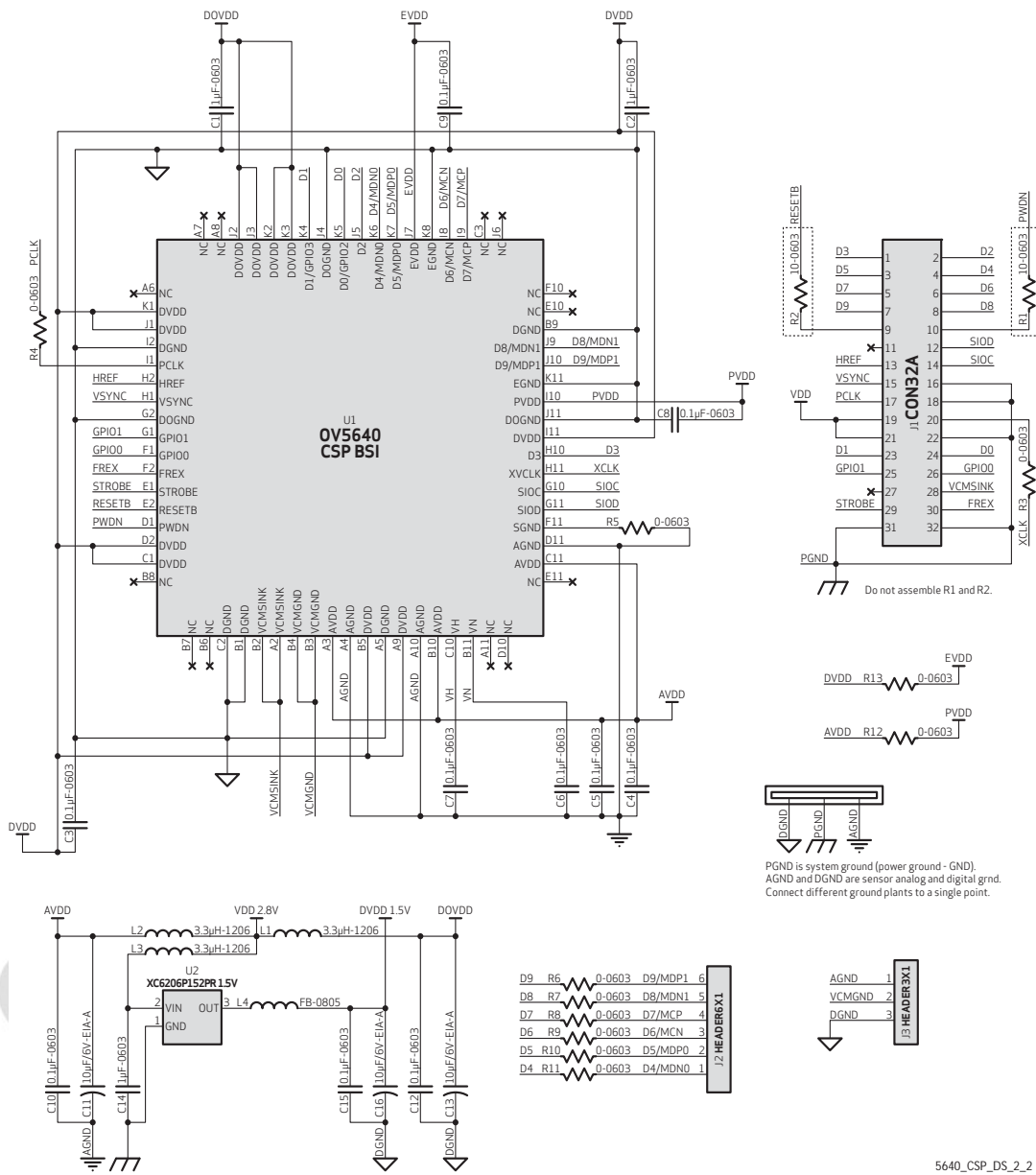
The exposure time is controlled by adjusting the time interval between precharging and sampling. After the data of the pixels in the row has been sampled, it is processed through analog circuitry to correct the offset and multiply the data with corresponding gain. Following analog processing is the ADC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

figure 2-1 OV5640 block diagram



5640\_DS\_2\_1

figure 2-2 reference design schematic



5640\_CSP\_DS\_2\_2

## 2.3 format and frame rate

table 2-1 format and frame rate

| format   | resolution | frame rate | scaling method   | pixel clock |
|----------|------------|------------|--|-------------|
| 5 Mpixel | 2592x1944  | 15 fps     | full resolution (dummy 16 pixel horizontal, 8 lines)<br>2608x1952 with dummy   | 96/192 MHz  |
| 1280x960 | 1280x960   | 45 fps     | subsampling in vertical and horizontal<br>1296x968<br>supports 2x2 binning   | 96/192 MHz  |
| 1080p    | 1920x1080  | 30 fps     | cropping from full resolution<br>1936x1088 with dummy pixels   | 96/192 MHz  |
| 720p     | 1280x720   | 60 fps     | cropping 2592x1944 to 2560x1440<br>subsampling in vertical and horizontal<br>1296x728 with dummy<br>supports 2x2 binning | 96/192 MHz  |
| VGA      | 640x480    | 90 fps     | subsampling from 1280x960<br>648x484 with dummy<br>supports 2x2 binning  | 48/96 MHz   |
| QVGA     | 320x240    | 120 fps    | subsampling from 1280x960<br>324x242 with dummy<br>supports 2x2 binning  | 24/48 MHz   |



## 2.4 I/O control

The OV5640 I/O pad direction and driving capability can be easily adjusted. **table 2-2** lists the driving capability and direction control registers of the I/O pads.

**table 2-2** driving capability and direction control for I/O pads (sheet 1 of 2)

| function                        | register                    | default value | R/W | description   |
|---------------------------------|-----------------------------|---------------|-----|---|
| output drive capability control | 0x302C                      | 0x02          | RW  | Bit[7:6]: output drive capability<br>00: 1x<br>01: 2x<br>10: 3x<br>11: 4x   |
| D[9:0] I/O control              | 0x3017[3:0],<br>0x3018[7:2] | 0x00          | RW  | input/output control for the D[9:0] pins<br>0: input<br>1: output<br><br>D9 share with MD2P pin for MIPI<br>D8 share with MD2N pin for MIPI<br>D7 share with MCP pin for MIPI<br>D6 share with MCN pin for MIPI<br>D5 share with MD1P pin for MIPI<br>D4 share with MD1N pin for MIPI |
| D[9:0] output select            | 0x301D[3:0],<br>0x301E[7:2] | 0x00          | RW  | output selection for the D[9:0] pins<br>0: normal data path<br>1: register-controlled value   |
| D[9:0] output value             | 0x301A[3:0],<br>0x301B[7:2] | 0x00          | RW  | D[9:0] output value   |
| D[9:0] input value              | 0x3051[3:0],<br>0x3052[7:2] | –             | R   | D[9:0] input value  |
| VSYNC I/O control               | 0x3017                      | 0x00          | RW  | Bit[6]: input/output control for the VSYNC pin<br>0: input<br>1: output   |
| VSYNC output select             | 0x301D                      | 0x00          | RW  | Bit[6]: output selection for the VSYNC pin<br>0: normal data path<br>1: register-controlled value   |
| VSYNC output value              | 0x301A                      | 0x00          | RW  | Bit[6]: VSYNC output value  |
| VSYNC input value               | 0x3051                      | –             | R   | Bit[6]: VSYNC input value   |
| HREF I/O control                | 0x3017                      | 0x00          | RW  | Bit[5]: input/output control for the HREF pin<br>0: input<br>1: output  |

table 2-2 driving capability and direction control for I/O pads (sheet 2 of 2)

| function           | register | default value | R/W | description  |
|--------------------|----------|---------------|-----|--|
| HREF output select | 0x301D   | 0x00          | RW  | Bit[5]: output selection for the HREF pin<br>0: normal data path<br>1: register-controlled value |
| HREF output value  | 0x301A   | 0x00          | RW  | Bit[5]: HREF output value  |
| HREF input value   | 0x3051   | –             | R   | Bit[5]: HREF input value   |
| PCLK I/O control   | 0x3017   | 0x00          | RW  | Bit[4]: input/output control for the PCLK pin<br>0: input<br>1: output                           |
| PCLK output select | 0x301D   | 0x00          | RW  | Bit[4]: output selection for the PCLK pin<br>0: normal data path<br>1: register-controlled value |
| PCLK output value  | 0x301A   | 0x00          | RW  | Bit[4]: PCLK output value  |
| PCLK input value   | 0x3051   | –             | R   | Bit[4]: PCLK input value   |

## 2.5 system clock control

The OV5640 PLL allows for an input clock frequency ranging from 6~27 MHz and has a maximum VCO frequency of 800 MHz. MipiClk is for the MIPI and SysClk is for the internal clock of the Image Signal Processing (ISP) block. The PLL can be bypassed by setting register 0x3039[7] to 1.

## 2.6 SCCB interface

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

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

The OV5640 supports up to four groups. These groups share 1 KB RAM and the size of each group is programmable by adjusting the start address. The group hold start addresses range from 0x40 to 0x7F, where the unit is 16 bytes.

**table 2-3** group sharing registers

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x3200  | GROUP ADDR0   | 0x40          | RW  | Start Address for Group0<br>{group_addr0[7:0], 4'h0} |
| 0x3201  | GROUP ADDR1   | 0x4A          | RW  | Start Address for Group1<br>{group_addr1[7:0], 4'h0} |
| 0x3202  | GROUP ADDR2   | 0x54          | RW  | Start Address for Group2<br>{group_addr2[7:0], 4'h0} |
| 0x3203  | GROUP ADDR3   | 0x5E          | RW  | Start Address for Group3<br>{group_addr3[7:0], 4'h0} |

The group write function is controlled by register **0x3212**.

**table 2-4** group write register

| address | register name    | default value | R/W | description  |
|---------|------------------|---------------|-----|--|
| 0x3212  | SRM GROUP ACCESS | –             | W   | SRM Group Access<br>Bit[7]: Group launch enable<br>Bit[6]: Test mode access group<br>Bit[5]: Group launch<br>Bit[4]: Group hold end<br>Bit[3:0]: Group id<br>0xx: Group for register access<br>011: Group to hold register address of embedded line SOF<br>100: Group to hold register address of embedded line EOF<br>101: Test mode for store register value to memory<br>110: Test mode for restore register value from memory<br>111: Group for write mask address |
| 0x3213  | SRM GROUP STATUS | –             | R   | SRM Group Status<br>Bit[7]: Store default<br>Bit[6]: Restore<br>Bit[5]: Group hold<br>Bit[4]: Group launch<br>Bit[3]: Group write<br>Bit[2:0]: Group select  |

The SCCB will enter group write mode after writing to register **0x3212** with a valid group ID. The subsequent registers will be held to the buffer specified by the group\_id instead of writing to the registers. Make sure the number of registers does not exceed the capacity of the group. Setting group\_hold\_end to 1 will exit the group write mode. After that, setting both group\_launch and group\_launch\_en to 1 will write the buffered values to the real registers. Multiple groups of registers can be prepared before writing to the real registers but be sure the correct group\_id is specified when the group write is launched.

The following is an example demonstrating the group write operation:

```

78 3212 00 Enable group0
78 3600 00 Write registers to be held in group0
78 3601 01
78 3212 10 End group0

```

78 3212 01 Enable group1  
78 3602 02 Write registers to be held in group1  
78 3603 03  
78 3212 11 End group1

..... Other direct register access

78 3212 02 Enable group2  
78 3604 04 Write registers to be held in group2  
78 3605 05  
78 3212 12 End group2

78 3212 A0 Launch group0

..... Other direct register access

78 3212 03 Enable group3  
78 3606 06 Write registers to be held in group3  
78 3607 07  
78 3212 13 End group3

78 3212 A1 Launch group1  
78 3212 A2 Launch group2  
78 3212 A3 Launch group3

## 2.7 power up sequence

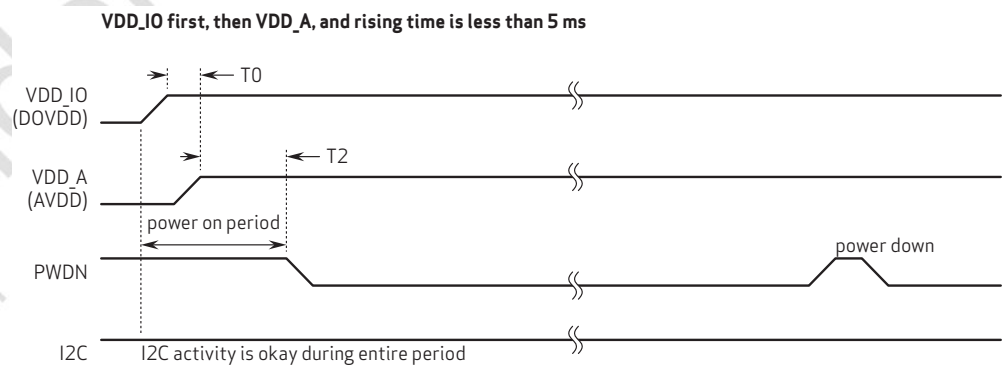
Based on the system power configuration (1.8V or 2.8V for I/O power, using external DVDD or internal DVDD, requiring access to the I2C during power up period or not), the power up sequence will differ. If 1.8V is used for I/O power, using the internal DVDD is preferred. If 2.8V is used for I/O power, due to a high voltage drop at the internal DVDD regulator, there is a potential heat issue. Hence, for a 2.8V power system, OmniVision recommends using an external DVDD source. Due to the higher power down current when using an external DVDD source, OmniVision strongly recommends cutting off all powers, including the external DVDD, when the sensor is not in use in the case of 2.8V I/O and external DVDD.

### 2.7.1 power up with internal DVDD

For powering up with the internal DVDD and I2C access during the power ON period, the following conditions must occur:

1. if  $V_{DD-IO}$  and  $V_{DD-A}$  are turned ON at the same time, make sure  $V_{DD-IO}$  becomes stable before  $V_{DD-A}$  becomes stable
2. PWDN is active high with an asynchronized design (does not need clock)
3. PWDN must go high if I2C is accessed during the power up period
4. for PWDN to go low, power must first become stable ( $AVDD$  to  $PWDN \geq 1$  ms)
5. RESETB is active low with an asynchronized design
6. state of RESETB does not matter during power up period once DOVDD is up

**figure 2-3** power up timing with internal DVDD and I2C access during power up period



**note**  $T0 \geq 0$  ms: delay from  $V_{DD\_IO}$  stable to  $V_{DD\_A}$  stable  
 $T2 \geq 1$  ms: delay from  $V_{DD\_A}$  stable to sensor power up stable

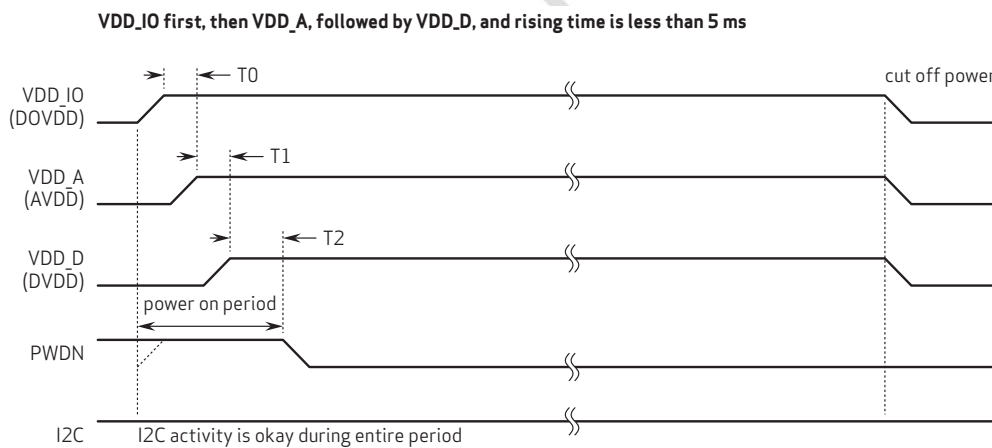
5640\_D5\_2\_3

### 2.7.2 power up with external DVDD source

For powering up with an external DVDD source and I2C access during the power ON period, the following conditions must occur:

1. if  $V_{DD-IO}$  and  $V_{DD-A}$  are turned ON at the same time, make sure  $V_{DD-IO}$  becomes stable before  $V_{DD-A}$  becomes stable
2. if  $V_{DD-A}$  and  $V_{DD-D}$  are turned ON at the same time, make sure  $V_{DD-A}$  becomes stable before  $V_{DD-D}$  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  $\geq 1$  ms)
5. all powers 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 up

**figure 2-4** power up timing with external DVDD source and I2C access during power up period



**note**  $T_0 \geq 0$  ms: delay from VDD\_IO stable to VDD\_A stable  
 $T_1 \geq 0$  ms: delay from VDD\_A stable to VDD\_D stable  
 $T_2 \geq 1$  ms: delay from VDD\_D stable to sensor power up stable

5640\_05\_2\_4

## 2.8 reset

The OV5640 sensor includes a **RESETB** pin that forces a complete hardware reset when it is pulled low (GND). The OV5640 clears all registers and resets them to their default values when a hardware reset occurs. A reset can also be initiated through the SCCB interface by setting register **0x3008[7]** to high.

## 2.9 standby and sleep

Two suspend modes are available for the OV5640:

- hardware standby
- SCCB software sleep

To initiate hardware standby mode, the **PWDN** pin must be tied to high. When this occurs, the OV5640 internal device clock is halted and all internal counters are reset and registers are maintained.

Executing a software power down through the SCCB interface suspends internal circuit activity but does not halt the device clock. All register content is maintained in standby mode.



## 3 block level description

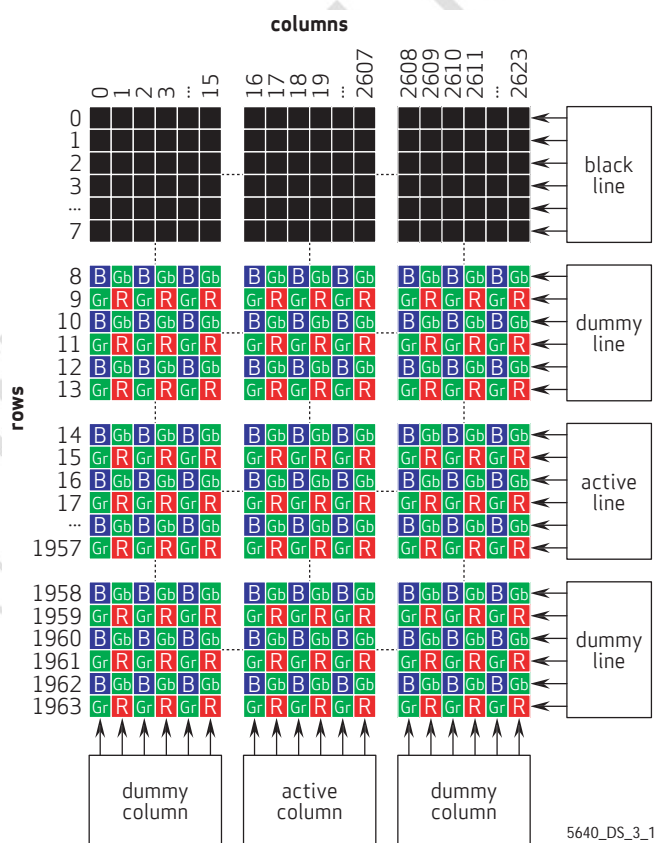
### 3.1 pixel array structure

The OV5640 sensor has an image array of 2624 columns by 1964 rows (5,153,536 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 5,153,536 pixels, 5,038,848 (2592x1944) are active pixels and can be output. The other pixels are used for black level calibration and interpolation.

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

**figure 3-1** sensor array region color filter layout



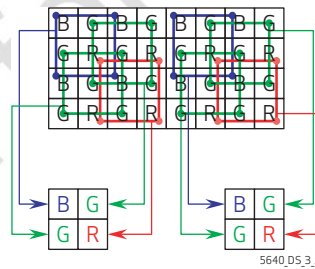
### 3.2 binning

Binning mode is usually used for subsampling. During subsampling, information is periodically dropped when data is output. When the binning function is ON, voltage levels of adjacent pixels are averaged before being sent to the ADC. If the binning function is OFF, the pixels, which are not output, are merely skipped. The OV5640 supports 2x2, 1x2, and 2x1 binning.

**figure 3-2** illustrates 2x2 binning, where the voltage levels of four (2x2) adjacent same-color pixels are averaged before entering the ADC.

In OV5640, vertical binning will automatically turn on when in vertical-subsampled formats.

**figure 3-2** example of 2x2 binning

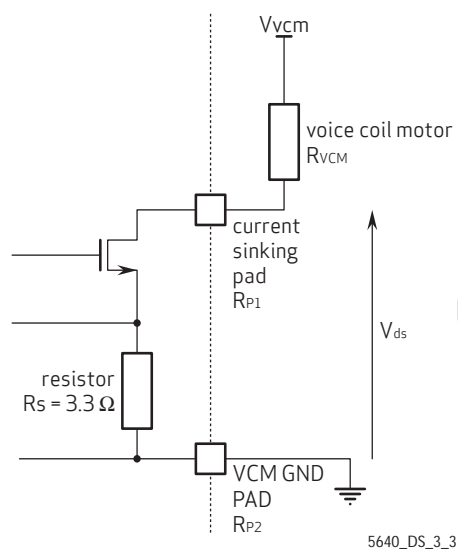


**table 3-1** binning-related registers

| address | register name   | default value | R/W | description                       |
|---------|-----------------|---------------|-----|-----------------------------------|
| 0x3621  | TIMING TC REG21 | 0x00          | RW  | Bit[0]: Horizontal binning enable |

## 3.3 VCM driver

figure 3-3 VCM block diagram



The maximum SINK current can be estimated as:

- $I_{SINK} = (V_{vcm} - V_{ds}) / (R_s + R_{vcm} + R_{p1} + R_{p2})$
- $V_{ds}$  is the transistor headroom
- $R_{p1}$  and  $R_{p2}$  are the resistance in the current path
- $R_{VCM}$  is the resistance of the voice coil motor.

The OV5640 VCM driver is a single 10-bit DAC with 100 mA output current sink capability. It is designed for linear control of the VCM. The DAC is controlled via the SCCB interface with clock rates up to 400 Hz. The OV5640 VCM driver provides three types of output current control modes that allow users to adjust transient response of the sinking current.

### 3.3.1 output current control mode

The OV5640 VCM driver uses 4 bits ( $S_3$ ,  $S_2$ ,  $S_1$ , and  $S_0$ ) to control the output current response.

1.  $S[3:0] = X000$ : Directly jump mode: code directly jumps to target code. Output current transient response time (see [table 3-2](#).)
2.  $S[3:0] = 0001$  to  $0111$ : Single step mode: code increases/decreases by a single step. Single step time durations are  $50\mu s$ ,  $100\mu s$ ,  $200\mu s$ ,  $400\mu s$ ,  $800\mu s$ ,  $1600\mu s$ , and  $3200\mu s$ , which are controlled by  $S_2$ ,  $S_1$ , and  $S_0$  (see [table 3-4](#).)
3.  $S[3:0] = 1001$  to  $1111$ : Multi-code steps mode: Code increases/decreases in multi-code steps. If the target code and the current code have a difference larger than 128, the 64-code step is applied first. When the difference in between target and current codes is no more than 128 but larger than 16, the 16-code step is used. When the difference is less than 16, it will directly jump to the target code. Single step time options are  $50\mu s$ ,  $100\mu s$ ,  $200\mu s$ ,  $400\mu s$ ,  $800\mu s$ ,  $1600\mu s$ , and  $3200\mu s$ , which are controlled by  $S_2$ ,  $S_1$ , and  $S_0$ , (see [table 3-5](#).)

table 3-2 VCM driver control

| function                           | register                    | description   |
|------------------------------------|-----------------------------|---|
| current transient response control | 0x3602                      | Bit[3:0]: Current transient response control<br>x000: mode 0<br>0001~0111: mode 1<br>1001~1111: mode 2                  |
| 10-bit DAC code                    | 0x3603[5:0],<br>0x3602[7:4] | 0x3603[5:0]: D[9:4]<br>0x3602[7:4]: D[3:0]  |
| clock divider                      | 0x3605[3:0],<br>0x3606[7:0] | divide external clock to obtain a 20 KHz clock for VCM control block<br>VCM control clock = external clock / Rdiv[11:0] |

table 3-3 VCM control registers

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x3603  | VCM[15:8]     | 0x01          | RW  | Bit[7]: PD<br>Bit[5:0]: D[9:4]   |
| 0x3602  | VCM[7:0]      | 0x50          | RW  | Bit[7:4]: D[3:0]<br>Bit[3]: S3<br>Bit[2:0]: S[2:0]   |
| 0x3605  | SLEW[11:8]    | 0x46          | RW  | Bit[3:0]: Rdiv[11:8]   |
| 0x3604  | SLEW[7:0]     | 0x05          | RW  | Bit[7:0]: Rdiv[7:0]  |
| 0x3606  | VCM CURRENT   | 0x00          | RW  | Bit[2:0]: VCM output current control<br>000: 0.71 * Id<br>001: 0.77 * Id<br>010: 0.83 * Id<br>011: 0.91 * Id<br>100: 1.00 * Id<br>101: 1.11 * Id<br>110: 1.25 * Id<br>111: 1.43 * Id |

table 3-4 single step mode

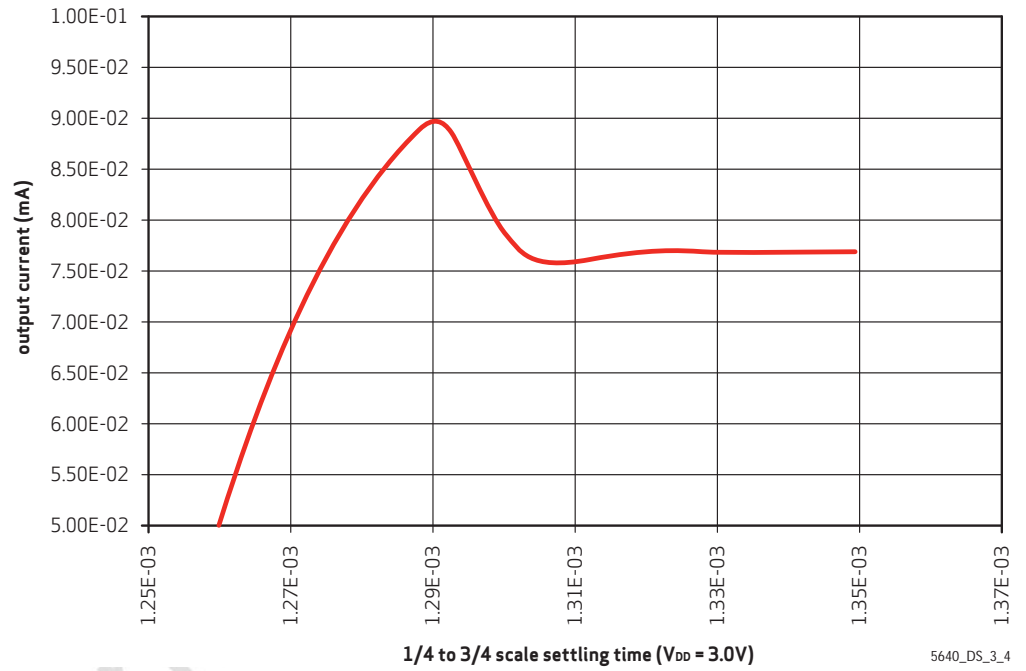
| mode             | S3 | S2 | S1 | S0 | single step transition time | full scale transition time (1023 steps) |
|------------------|----|----|----|----|-----------------------------|---|
| single step mode | 0  | 0  | 0  | 1  | 50 $\mu$ s                  | 51.15ms                                 |
|                  | 0  | 0  | 1  | 0  | 100 $\mu$ s                 | 102.3ms                                 |
|                  | 0  | 0  | 1  | 1  | 200 $\mu$ s                 | 204.6ms                                 |
|                  | 0  | 1  | 0  | 0  | 400 $\mu$ s                 | 409.2ms                                 |
|                  | 0  | 1  | 0  | 1  | 800 $\mu$ s                 | 818.4ms                                 |
|                  | 0  | 1  | 1  | 0  | 1600 $\mu$ s                | 1.637s                                  |
|                  | 0  | 1  | 1  | 1  | 3200 $\mu$ s                | 3.274s                                  |

table 3-5 multi-code step mode

| mode             | S3 | S2 | S1 | S0 | single step transition time | full scale transition time (22 steps) <sup>a</sup> |
|------------------|----|----|----|----|-----------------------------|--|
| single step mode | 1  | 0  | 0  | 1  | 50 $\mu$ s                  | 1.1ms  |
|                  | 1  | 0  | 1  | 0  | 100 $\mu$ s                 | 2.2ms  |
|                  | 1  | 0  | 1  | 1  | 200 $\mu$ s                 | 4.4ms  |
|                  | 1  | 1  | 0  | 0  | 400 $\mu$ s                 | 8.8ms  |
|                  | 1  | 1  | 0  | 1  | 800 $\mu$ s                 | 17.6ms   |
|                  | 1  | 1  | 1  | 0  | 1600 $\mu$ s                | 35.2ms   |
|                  | 1  | 1  | 1  | 1  | 3200 $\mu$ s                | 70.4ms   |

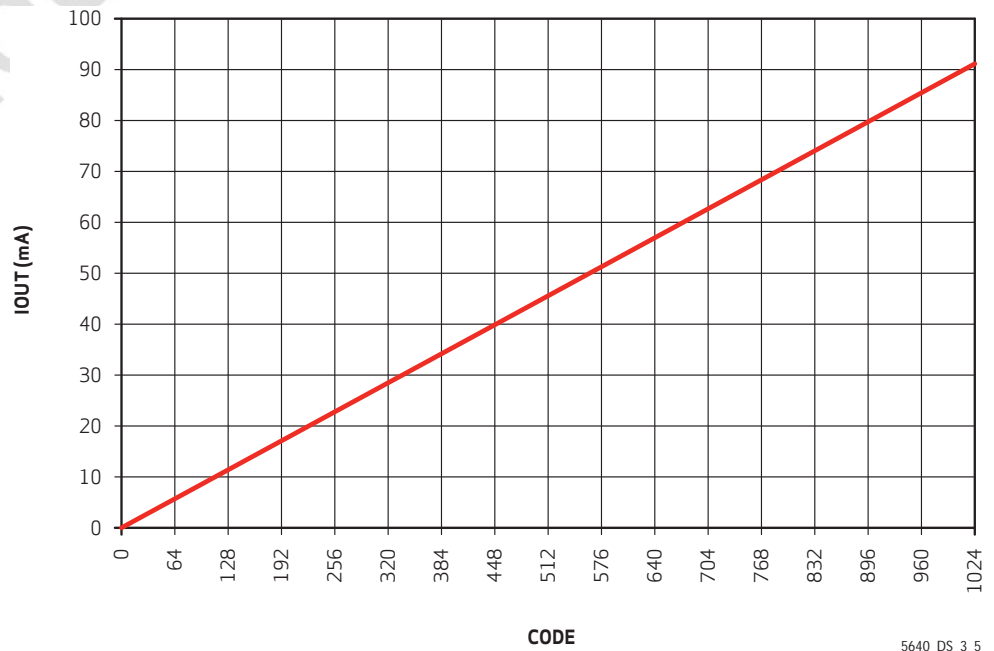
a. a full scale transition includes fourteen 64-code steps, seven 16-code steps and one directly jump step.

figure 3-4 1/4 to 3/4 scale settling time (directly jump mode, VDD = 3.0V)



5640\_DS\_3\_4

figure 3-5 sink current vs. code (VDD = 3.0V, reg 0x30A5 = 0x05, VCM resistance = 23ohms)



5640\_DS\_3\_5

## 4 image sensor core digital functions

### 4.1 mirror and flip

The OV5640 provides Mirror and Flip readout modes, which respectively reverse the sensor data readout order horizontally and vertically (see [figure 4-1](#)). In flip, the OV5640 does not need additional settings because the ISP block will auto-detect whether the pixel is in the red line or blue line and make the necessary adjustments.

**figure 4-1** mirror and flip samples



5640 DS 4 1

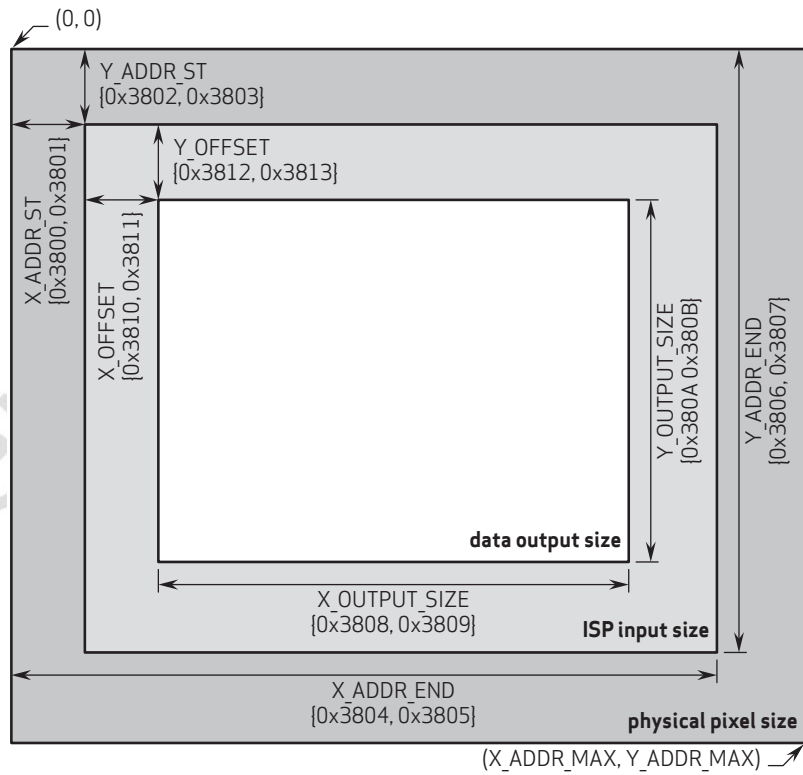
**table 4-1** mirror and flip registers

| address | register name   | default value | R/W | description   |
|---------|-----------------|---------------|-----|---|
| 0x3820  | TIMING TC REG20 | 0x40          | RW  | Timing control<br>Bit[2]: ISP vflip<br>Bit[1]: Sensor vflip   |
| 0x3821  | TIMING TC REG21 | 0x00          | RW  | Timing Control<br>Bit[2]: ISP mirror<br>Bit[1]: Sensor mirror |

### 4.2 image windowing

The OV5640 uses registers 0x3800 ~ 0x3814 for image windowing. **figure 4-2** illustrates how the registers define the windowing size. Physical pixel size is the total pixel array size we have in the sensor. The ISP input size is the total pixel data read from pixel array. Typically, the larger ISP input size is, the less maximum frame rate can be reached. The data output size is the image output size of OV5640. This size is windowed from ISP input size and is defined by x\_offset and y\_offset as **figure 4-2** shows.

**figure 4-2** image windowing



5640\_DS\_4\_2



figure 4-3 shows the windowing configuration when scaling function is enabled. The pre-scaling image size is the ISP input size subtracted by two times of offsets for both horizontal and vertical.

figure 4-3 image windowing configuration

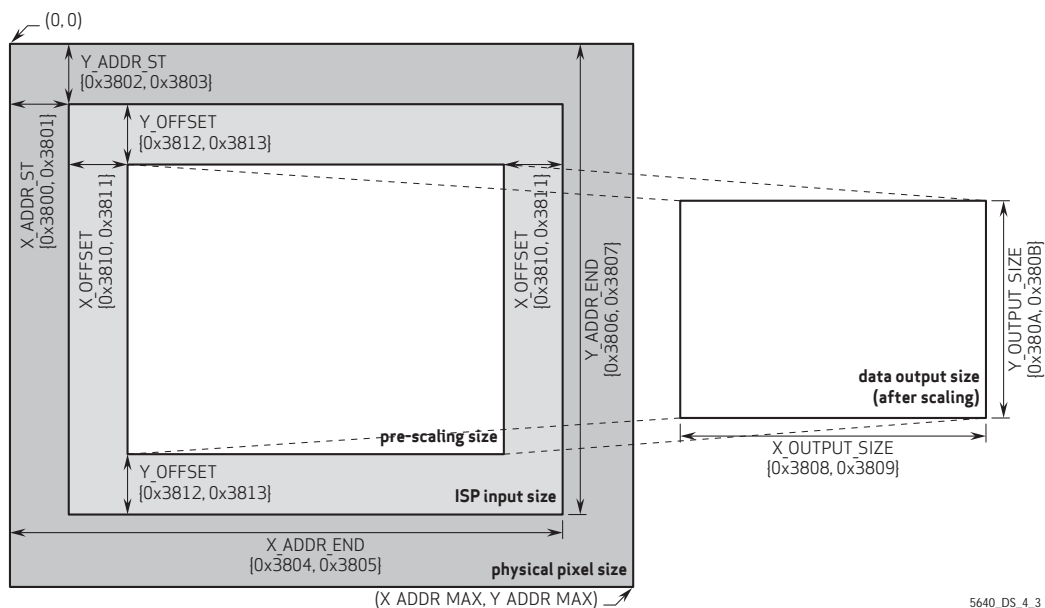


table 4-2 image windowing registers (sheet 1 of 2)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x3800  | TIMING HS     | 0x00          | RW  | Bit[3:0]: X address start high byte[11:8] high byte   |
| 0x3801  | TIMING HS     | 0x00          | RW  | Bit[7:0]: X address start low byte[7:0] low byte      |
| 0x3802  | TIMING VS     | 0x00          | RW  | Bit[2:0]: Y address start high byte[10:8] high byte   |
| 0x3803  | TIMING VS     | 0x00          | RW  | Bit[7:0]: Y address start low byte[7:0] low byte      |
| 0x3804  | TIMING HW     | 0x0A          | RW  | Bit[3:0]: X address end high byte[11:8] high byte     |
| 0x3805  | TIMING HW     | 0x3F          | RW  | Bit[7:0]: X address end low byte[7:0] low byte        |
| 0x3806  | TIMING VH     | 0x07          | RW  | Bit[2:0]: Y address end high byte[10:8] high byte     |
| 0x3807  | TIMING VH     | 0x9F          | RW  | Bit[7:0]: Y address end low byte[7:0] low byte        |
| 0x3808  | TIMING DVPHO  | 0x0A          | RW  | Bit[3:0]: DVP output horizontal width[11:8] high byte |

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

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x3809  | TIMING DVPHO   | 0x20          | RW  | Bit[7:0]: DVP output horizontal width[7:0] low byte  |
| 0x380A  | TIMING DVPVO   | 0x07          | RW  | Bit[2:0]: DVP output vertical height[10:8] high byte |
| 0x380B  | TIMING DVPVO   | 0x98          | RW  | Bit[7:0]: DVP output vertical height[7:0] low byte   |
| 0x380C  | TIMING HTS     | 0x0B          | RW  | Bit[3:0]: Total horizontal size[11:8] high byte      |
| 0x380D  | TIMING HTS     | 0x1C          | RW  | Bit[7:0]: Total horizontal size[7:0] low byte        |
| 0x380E  | TIMING VTS     | 0x07          | RW  | Bit[7:0]: Total vertical size[15:8] high byte        |
| 0x380F  | TIMING VTS     | 0xB0          | RW  | Bit[7:0]: Total vertical size[7:0] low byte          |
| 0x3810  | TIMING HOFFSET | 0x00          | RW  | Bit[3:0]: ISP horizontal offset[11:8] high byte      |
| 0x3811  | TIMING_HOFFSET | 0x10          | RW  | Bit[7:0]: ISP horizontal offset[7:0] low byte        |
| 0x3812  | TIMING VOFFSET | 0x00          | RW  | Bit[2:0]: ISP vertical offset[10:8] high byte        |
| 0x3813  | TIMING VOFFSET | 0x04          | RW  | Bit[7:0]: ISP vertical offset[7:0] low byte          |

## 4.3 test pattern

For testing purposes, the OV5640 offers one type of test pattern, color bar.

figure 4-4 test pattern

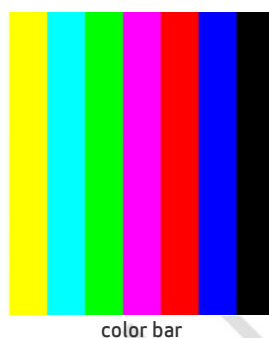


table 4-3 test pattern selection control

| address | register name          | default value | R/W | description   |
|---------|------------------------|---------------|-----|---|
| 0x503D  | PRE ISP TEST SETTING 1 | 0x00          | RW  | Bit[7]: Color bar enable<br>0: Test disable<br>1: Color bar enable<br>Bit[3:2]: Color bar style<br>00: Standard eight color bar<br>01: Gradual change at vertical mode 1<br>10: Gradual change at horizontal<br>11: Gradual change at vertical mode 2 |

## 4.4 50/60Hz detection

### 4.4.1 overview

When the integration time is not an integer multiple of the period of light intensity, the image will flicker. The function of the detector is to detect whether the sensor is under a 50 Hz or 60 Hz light source so that the basic step of integration time can be determined. Contact your local OmniVision FAE for auto detection settings.

## 4.5 AEC/AGC algorithms

### 4.5.1 overview

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 functions

| address | register name    | default value | R/W | description   |
|---------|------------------|---------------|-----|---|
| 0x3500  | AEC PK EXPOSURE  | 0x00          | RW  | Exposure Output<br>Bit[3:0]: Exposure [19:16]   |
| 0x3501  | AEC PK EXPOSURE  | 0x02          | RW  | Exposure Output<br>Bit[7:0]: Exposure [15:8]  |
| 0x3502  | AEC PK EXPOSURE  | 0x00          | RW  | Exposure Output<br>Bit[7:0]: Exposure [7:0]   |
| 0x3503  | AEC PK MANUAL    | 0x00          | RW  | AEC Manual Mode Control<br>Bit[1]: AGC manual<br>0: Auto enable<br>1: Manual enable<br>Bit[0]: AEC manual<br>0: Auto enable<br>1: Manual enable |
| 0x350A  | AEC PK REAL GAIN | 0x00          | RW  | Real Gain<br>Bit[1:0]: Real gain[9:8]   |
| 0x350B  | AEC PK REAL GAIN | 0x10          | RW  | Real Gain<br>Bit[7:0]: Real gain[7:0]   |
| 0x350C  | AEC PK VTS       | 0x00          | RW  | AEC VTS Output<br>Bit[7:0]: VTS[15:8] high bits   |
| 0x350D  | AEC PK VTS       | 0x00          | RW  | AEC VTS Output<br>Bit[7:0]: VTS[7:0] low bits   |

### 4.5.2 average-based algorithm

The average-based AEC controls image luminance using registers (0x3A0F), (0x3A10), (0x3A1B), and (0x3A1E). In average-based mode, the value of register (0x3A0F) indicates the high threshold value, and the value of register (0x3A10) indicates the low threshold value. The value of register (0x3A1B) indicates the high threshold value for image change from stable state to unstable state and the value of register (0x3A1E) indicates the low threshold value for image change from stable state to unstable state. When the target image luminance average value AVG READOUT (0x56A1) is within the range specified by registers (0x3A1B) and (0x3A1E), the AEC keeps the image exposure and gain. When register AVG READOUT (0x56A1) is greater than the value in register (0x3A1B), the AEC will decrease the image exposure and gain until it falls into the range of {0x3A10, 0x3A0F}. When register AVG READOUT (0x56A1) is less than

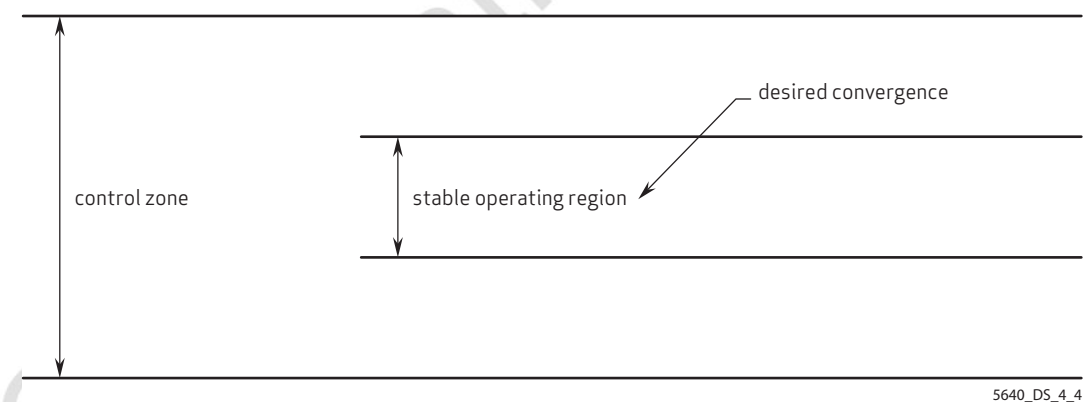
the value in register (0x3A1E), the AEC will increase the image exposure and gain until it falls into the range of {0x3A10, 0x3A0F}. Accordingly, the value in register (0x3A0F) should be greater than the value in register (0x3A10). The gap between the values of registers (0x3A1B) and (0x3A1E) controls 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 (0x3A0F) and (0x3A10). For manual mode, the speed supports both normal and fast speed selection. AEC set to normal mode will allow for the slowest step increment or decrement in the image exposure to maintain the specified range. AEC set to fast mode will provide for an approximate ten-step increment or decrement in the image exposure to maintain the specified range. For auto mode, the speed 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 AVG READOUT (0x56A1); thus, the AEC speed can be adjusted automatically by the image average value or controlled manually.

Register (0x3A11) and register (0x3A1F) controls the fast AEC range in manual speed selection made. If the target image AVG READOUT (0x56A1) is greater than (0x3A11), AEC will decrease by half. If register AVG READOUT (0x56A1) is less than (0x3A1F), AEC will double.

As shown in desired convergence, 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 4-5** desired convergence



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

table 4-5 AEC/AGC control functions

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x3A0F  | AEC CTRL0F    | 0x78          | RW  | Stable Range High Limit (enter)<br>Bit[7:0]: WPT             |
| 0x3A10  | AEC CTRL10    | 0x68          | RW  | Stable Range Low Limit (enter)<br>Bit[7:0]: BPT              |
| 0x3A11  | AEC CTRL11    | 0xD0          | RW  | Step Manual Mode, Fast Zone High Limit<br>Bit[7:0]: vpt_high |
| 0x3A1B  | AEC CTRL1B    | 0x78          | RW  | Stable Range High Limit (go out)<br>Bit[7:0]: WPT2           |
| 0x3A1E  | AEC CTRL1E    | 0x68          | RW  | Stable Range Low Limit (go out)<br>Bit[7:0]: BPT2            |
| 0x3A1F  | AEC CTRL1F    | 0x40          | RW  | Step Manual Mode, Fast Zone Low Limit<br>Bit[7:0]: vpt_low   |

For the average-based AEC/AGC algorithm, the measured window is horizontally and vertically adjustable and divided by sixteen (4x4) zones (see [figure 4-6](#)). 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.

#### 4.5.2.1 average luminance (YAVG)

Auto exposure time calculation is based on a frame brightness average value. By properly setting  $x\_start$ ,  $x\_end$ ,  $y\_start$ , and  $y\_end$  as shown in [figure 4-6](#), 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 4-6 average-based window definition

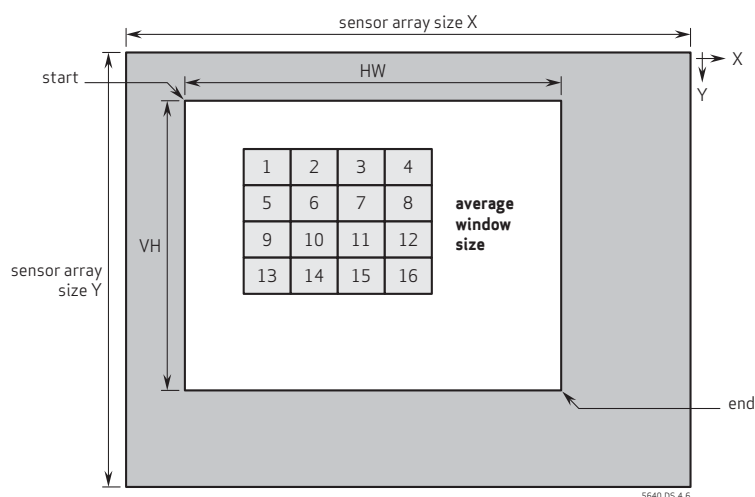


table 4-6 timing control functions (sheet 1 of 2)

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x3810  | TIMING_HOFFSET | 0x00          | RW  | Bit[3:0]: ISP horizontal offset[11:8] high byte   |
| 0x3811  | TIMING_HOFFSET | 0x04          | RW  | Bit[7:0]: ISP Horizontal offset[7:0] low byte   |
| 0x3812  | TIMING_VOFFSET | 0x11          | RW  | Bit[3:0]: ISP vertical offset[11:8] high byte   |
| 0x3813  | TIMING_VOFFSET | 0x11          | RW  | Bit[7:0]: ISP vertical offset[7:0] low byte   |
| 0x3808  | TIMING_DVPHO   | 0x07          | RW  | Bit[3:0]: DVP output horizontal width[11:8] high byte   |
| 0x3809  | TIMING_DVPHO   | 0x98          | RW  | Bit[7:0]: DVP output horizontal width[7:0] low byte   |
| 0x380A  | TIMING_DVPVO   | 0x0B          | RW  | Bit[3:0]: DVP output vertical height[11:8] high byte  |
| 0x380B  | TIMING_DVPVO   | 0x1C          | RW  | Bit[7:0]: DVP output vertical height[7:0] low byte  |
| 0x501D  | ISP_MISC       | 0x00          | RW  | Bit[4]: Average size manual enable  |
| 0x5680  | X_START        | 0x00          | RW  | Bit[3:0]: x start[11:8]<br>Horizontal start position for average window high byte, valid when 0x501D[4]=1 |
| 0x5681  | X_START        | 0x00          | RW  | Bit[7:0]: x start[7:0]<br>Horizontal start position for average window low byte, valid when 0x501D[4]=1   |

table 4-6 timing control functions (sheet 2 of 2)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5682  | Y START       | 0x00          | RW  | Bit[2:0]: y start[10:8]<br>Vertical start position for average window low byte, valid when 0x501D[4]=1    |
| 0x5683  | Y START       | 0x00          | RW  | Bit[7:0]: y start[7:0]<br>Vertical start position for average window low byte, valid when 0x501D[4]=1     |
| 0x5684  | X WINDOW      | 0x10          | RW  | Bit[3:0]: Window X [11:8]<br>Horizontal end position for average window high byte, valid when 0x501D[4]=1 |
| 0x5685  | X WINDOW      | 0xA0          | RW  | Bit[7:0]: Window X [7:0]<br>Horizontal end position for average window low byte, valid when 0x501D[4]=1.  |
| 0x5686  | Y WINDOW      | 0x0C          | RW  | Bit[2:0]: Window Y [10:8]<br>Vertical end position for average window high byte, valid when 0x501D[4]=1   |
| 0x5687  | Y WINDOW      | 0x78          | RW  | Bit[7:0]: Window Y [7:0]<br>Vertical end position for average window low byte, valid when 0x501D[4]=1     |
| 0x5688  | WEIGHT00      | 0x11          | RW  | Bit[7:4]: Window 01 weight<br>Bit[3:0]: Window 00 weight  |
| 0x5689  | WEIGHT01      | 0x11          | RW  | Bit[7:4]: Window 03 weight<br>Bit[3:0]: Window 02 weight  |
| 0x568A  | WEIGHT02      | 0x11          | RW  | Bit[7:4]: Window 11 weight<br>Bit[3:0]: Window 10 weight  |
| 0x568B  | WEIGHT03      | 0x11          | RW  | Bit[7:4]: Window 13 weight<br>Bit[3:0]: Window 12 weight  |
| 0x568C  | WEIGHT04      | 0x11          | RW  | Bit[7:4]: Window 21 weight<br>Bit[3:0]: Window 20 weight  |
| 0x568D  | WEIGHT05      | 0x11          | RW  | Bit[7:4]: Window 23 weight<br>Bit[3:0]: Window 22 weight  |
| 0x568E  | WEIGHT06      | 0x11          | RW  | Bit[7:4]: Window 31 weight<br>Bit[3:0]: Window 30 weight  |
| 0x568F  | WEIGHT07      | 0x11          | RW  | Bit[7:4]: Window 33 weight<br>Bit[3:0]: Window 32 weight  |



## 4.6 AEC/AGC steps

The AEC and AGC work together to obtain adequate exposure/gain based on the current environmental illumination. In order to achieve the best SNR, extending the exposure time is always preferred rather than raising the gain when the current illumination is getting brighter. Vice versa, under dark conditions, the action to decrease the gain is always taken prior to shortening the exposure time.

### 4.6.1 auto exposure control (AEC)

The function of the AEC is to calculate the integration time of the next frame and send the information to the timing control block. Based on the statistics of previous frames, the AEC is able to determine whether the integration time should increase, decrease, fast increase, fast decrease, or remain the same.

In extremely bright situations, the LAEC activates, allowing integration time to be less than one row. In extremely dark situations, the night mode activates, allowing 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. This new AEC step system is called the banding filter, suggesting that the exposure time is not continuous but falls in some steps.

#### 4.6.1.1 LAEC

If the integration time is only one row period but the image is too bright, AEC will enter LAEC mode. Within LAEC, the integration time can be further decreased to the minimum of 1/16 row. LAEC ON/OFF can be set in register bit 0x3A00[6].

#### 4.6.1.2 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. This design is to reject image flickering when the light source is not steady but periodical.

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

Band Step = 'period of light intensity' × 'frame rate' × 'rows per frame'.

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.

When auto-banding is ON, if the next integration time is less than the minimum band step, banding will automatically turn OFF. It will turn ON again when the next integration time becomes larger than the minimum band. If auto banding is disabled, the minimum integration time is one band step. Auto banding can be set in register bit 0x3A00[5].

#### 4.6.1.3 banding mode OFF with AEC

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

#### 4.6.1.4 night mode

The OV5640 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[7:0], 0x3A03[7:0], 0x3A14[7:0], 0x3A15[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.

#### 4.6.2 manual exposure control

To manually change exposure value, you must first set both 0x3503[0], where 0x3503[0] enables manual exposure control. In auto exposure mode, the extra exposure values (larger than 1 frame) in registers 0x350C/0x350D automatically change. In manual exposure mode, these registers will not automatically change. The manually set exposure in registers 0x3500~0x3502 must be less than the maximum exposure value in {0x380E, 0x380F} + {0x350C, 0x350D}. The exposure value in registers 0x3500~0x3502 is in units of line\*16 - the low 4 bits (0x3502[3:0]) is the fraction of line, the maximum value in {0x380E + 0x380F} + {0x350C, 0x350D} is in unit of line. If the manually set exposure value is less than one pre-defined frame period (e.g., 1/15 second in 15fps), there is no need to change 0x380E/0x380F. If the exposure value needs to be set beyond the pre-defined frame period; in other words, if the frame period needs to be extended to extend exposure time, then the maximum frame value in 0x380E/0x380F needs to be set first, then the exposure can be set in registers 0x3500~0x3502 accordingly.

#### 4.6.3 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; otherwise, the integration time will keep switching between two adjacent steps and the image flickers.

##### 4.6.3.1 integration time between 1~16 rows

When integration time is less than 16 rows, the changes between adjacent AEC steps are larger than 1/16, which may possibly make the image oscillate between two AEC levels; thus, some AGC steps are added in between.

##### 4.6.3.2 gain insertion between AEC banding steps

When banding mode is ON, the integration time changes in step of the period of light intensity. For the first 16 band steps, since the exposure time change between adjacent steps is larger than 1/16, AGC steps are inserted to ensure image stability.

##### 4.6.3.3 gain insertion between night mode steps

Between night mode steps (e.g., integration time = 1 frame and 2 frames), AGC steps are inserted to ensure no adjacent step change is larger than 1/16.

##### 4.6.3.4 when AEC reaches maximum

When AEC reaches its maximum step while the image is still too dark, the gain starts to increase until the new frame average falls into the stable range or AGC reaches its maximum step. The AGC ceiling can be set in {0x3A18[9:8], 0x3A19[7:0]}.

#### 4.6.4 manual gain control

To manually change gain, first set register bit 0x3503[1] to enable manual control, then change the values in 0x350A/0x350B for the manual gain. The OV5640 has a maximum of 64x gain.

## 4.7 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:

- Combining two ADC data paths into one data path
- Adjusting all normal pixel values based on the values of the black levels
- Applying multiplication to all pixel values based on digital gain

Black level adjustments can be made with registers 0x4000 through 0x4013.

**table 4-7** BLC control functions

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x4000  | BLC CTRL00    | 0x89          | RW  | BLC Control 00<br>Bit[0]: BLC enable   |
| 0x4002  | BLC CTRL02    | 0x45          | RW  | Bit[7]: Format change enable<br>BLC update when format changes   |
| 0x4003  | BLC CTRL03    | 0x08          | RW  | Bit[7]: BLC redo enable<br>Write 1 into it will trigger a BLC redo N frames begin, N is 0x4003[5:0]<br>Bit[6]: BLC freeze<br>Bit[5:0]: Manual frame number |
| 0x4005  | BLC CTRL05    | 0x18          | RW  | Bit[1]: BLC always update<br>0: Normal freeze<br>1: BLC always update;   |
| 0x4009  | BLACK LEVEL   | 0x10          | RW  | Bit[7:0]: BLC black level target at 10-bit range   |

## 4.8 light frequency selection

The OV5640 can detect the light flickering frequency. When this function is enabled, the sensor can detect the light frequency and select the corresponding banding filter value. To remove banding, the banding filter should be turned on and the banding filter value should be set to the appropriate value.

**table 4-8** light frequency registers

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x3C01  | 5060HZ_CTRL1  | 0x00          | RW  | Bit[7]: Band manual enable<br>0: Auto<br>1: Manual                    |
| 0x3C00  | 5060HZ_CTRL2  | 0x00          | RW  | Bit[2]: Band value manual setting<br>0: 60 Hz light<br>1: 50 Hz light |
| 0x3C0C  | 5060HZ_CTRL3  | –             | R   | Bit[0]: Band50/60<br>0: 60 Hz light<br>1: 50 Hz light                 |

## 4.9 digital gain

The OV5640 supports 1/2/4 digital gain. Normally, the gain is controlled automatically by the automatic gain control (AGC) block.

## 4.10 strobe flash and frame exposure

### 4.10.1 strobe flash control

The strobe signal is programmable. It supports both LED and Xenon modes. The polarity of the pulse can be changed. The strobe signal is enabled (turned high/low depending on the pulse's polarity) by requesting the signal via the SCCB interface. Flash modules are triggered by the rising edge by default or by the falling edge if the signal polarity is changed. It supports the following flashlight modes (see [table 4-9](#)).

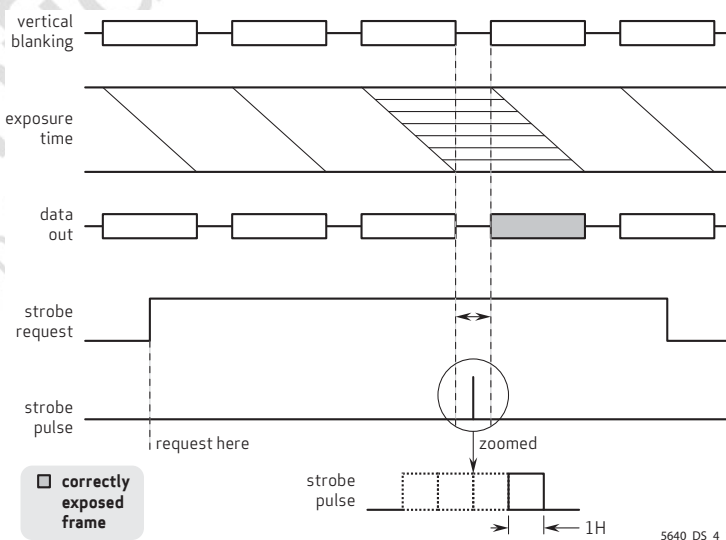
**table 4-9** flashlight modes

| mode  | output     | AEC / AGC | AWB |
|-------|------------|-----------|-----|
| xenon | one-pulse  | no        | no  |
| LED 1 | pulse      | no        | no  |
| LED 2 | pulse      | no        | yes |
| LED 3 | continuous | yes       | yes |

#### 4.10.1.1 xenon flash control

After a strobe request is submitted, the strobe pulse will be activated at the beginning of the third frame (see [figure 4-7](#)). The third frame will be correctly exposed. The pulse width can be changed in Xenon mode between 1H and 4H, depending on register 0x3B00[3:2], where H is one row period.

**figure 4-7** xenon flash mode



4.10.1.2 LED 1 & 2 mode

Two frames after the strobe request is submitted, the third frame is correctly exposed. The strobe pulse will be activated only one time if the strobe end request is set correctly (see figure 4-8). If end request is not sent, the strobe signal is activated intermittently until the strobe end request is set (see figure 4-9). The number of skipped frames is programmable using registers {0x3A1C, 0x3A1D}.

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

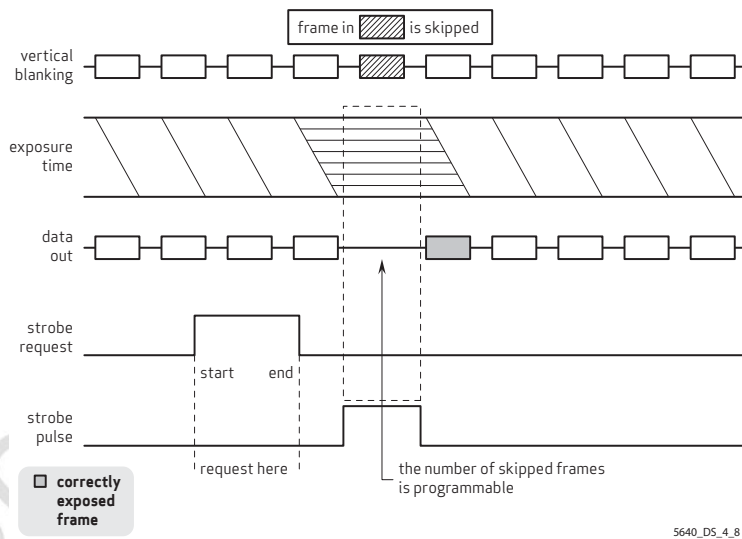
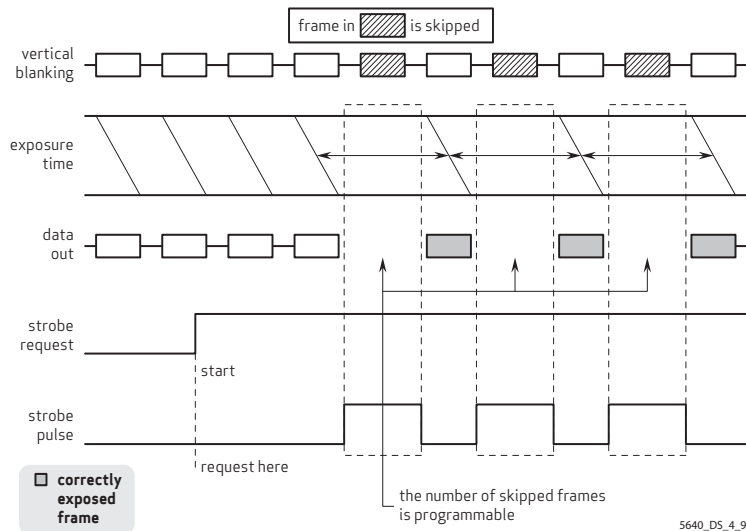


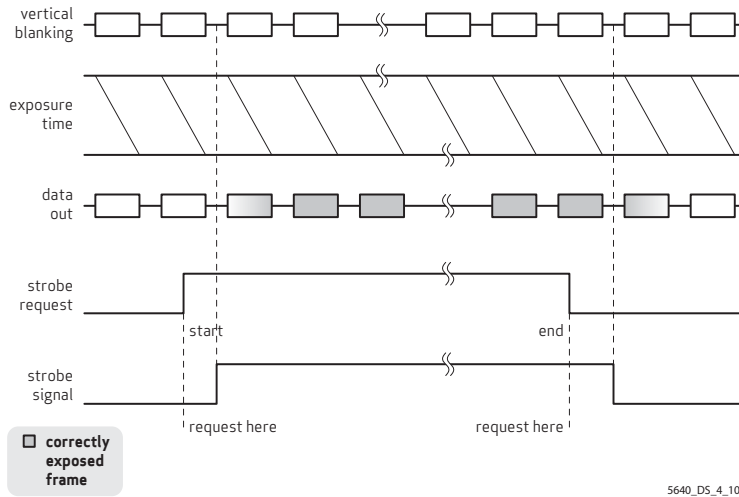
figure 4-9 LED 1 & 2 mode - multiple pulse output



#### 4.10.1.3 LED 3 mode

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

**figure 4-10** LED 3 mode



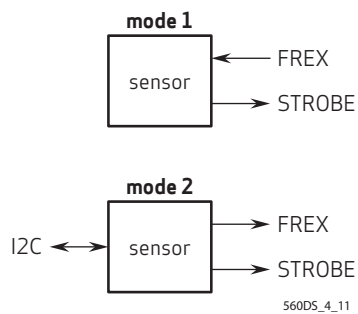
#### 4.10.2 frame exposure (FREX) mode

In FREX mode, whole frame pixels start integration at the same time, rather than integrating row by row. After the user-defined exposure time (registers 0x3B04, 0x3B05), the shutter closes, preventing further integration and the image begins to read out. After the readout finishes, the shutter opens again and the sensor resumes normal mode, waiting for the next FREX request.

The OV5640 supports two modes of FREX (see [figure 4-11](#)):

- mode 1 - frame exposure and shutter control requests come from the external system via the FREX pin. The sensor will send a strobe output signal to control the flash light.
- mode 2 - frame exposure request comes from the external system via the I2C register 0x3B08[0]. The sensor will output two signals, shutter control signal through the FREX pin and strobe signal through the STROBE pin.

**figure 4-11** FREX modes



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

#### 4.10.3 FREX strobe flash control

See [table 4-10](#) for FREX strobe control functions.

**table 4-10** FREX strobe control functions (sheet 1 of 2)

| address | register name         | default value | R/W | description   |
|---------|-----------------------|---------------|-----|---|
| 0x3B00  | STROBE CTRL           | 0x00          | RW  | Strobe Control<br>Bit[7]: Strobe request ON/OFF<br>0: OFF/BLC<br>1: ON<br>Bit[6]: Strobe pulse reverse<br>Bit[3:2]: width_in_xenon<br>Bit[1:0]: Strobe mode<br>00: Xenon<br>01: LED 1<br>10: LED 2<br>11: LED 3 |
| 0x3B01  | FREX EXPOSURE 02      | 0x00          | RW  | Bit[7:0]: FREX exposure time[23:16]   |
| 0x3B02  | FREX SHUTTER DELAY 01 | 0x08          | RW  | Bit[4:0]: Shutter delay time[12:8]  |
| 0x3B03  | FREX SHUTTER DELAY 00 | 0x00          | RW  | Bit[7:0]: Shutter delay time[7:0]<br>Unit: 64x sclk cycle   |
| 0x3B04  | FREX EXPOSURE 01      | 0x04          | RW  | Bit[7:0]: FREX exposure time [15:8]   |
| 0x3B05  | FREX EXPOSURE 00      | 0x00          | RW  | Bit[7:0]: FREX exposure time [7:0]<br>Unit: Tline   |
| 0x3B06  | FREX CTRL 07          | 0x04          | RW  | Bit[7:4]: FREX frame delay<br>Bit[3:0]: FREX STROBE width[3:0]  |
| 0x3B07  | FREX MODE             | 0x08          | RW  | Bit[7:0]: FREX mode selection<br>00: FREX STROBE mode0<br>01: FREX STROBE mode1<br>1x: Rolling STROBE   |
| 0x3B08  | FREX REQUEST          | 0x00          | RW  | FREX Request  |
| 0x3B09  | FREX HREF DELAY       | 0x02          | RW  | FREX HREF Delay   |



table 4-10 FREX strobe control functions (sheet 2 of 2)

| address | register name   | default value | R/W | description  |
|---------|-----------------|---------------|-----|--|
| 0x3B0A  | FREX RST LENGTH | 0x04          | RW  | Bit[2:0]: FREX precharge length<br>000: 1/16 Tline<br>001: 1/8 Tline<br>010: 1/4 Tline<br>011: 1/2 Tline<br>100: 1 Tline<br>101: 2 Tline<br>110: 4 Tline<br>111: 8 Tline |
| 0x3B0B  | STROBE WIDTH    | 0x00          | RW  | Bit[7:0]: STROBE width[19:12]  |
| 0x3B0C  | STROBE WIDTH    | 0x3D          | RW  | Bit[7:0]: STROBE width[11:4]   |

#### 4.11 one time programmable (OTP) memory

The OV5640 supports a maximum of 256 bits of one-time programmable (OTP) memory to store chip identification and manufacturing information.

table 4-11 OTP control functions

| address | register name    | default value | R/W | description   |
|---------|------------------|---------------|-----|---|
| 0x3D20  | OTP PROGRAM CTRL | 0x00          | RW  | Bit[7]: OTP program busy<br>Bit[1]: OTP program speed<br>0: Fast<br>1: Slow<br>Bit[0]: OTP program enable |
| 0x3D21  | OTP READ CTRL    | 0x00          | RW  | Bit[7]: OTP read busy<br>Bit[1]: OTP read speed<br>0: Fast<br>1: Slow<br>Bit[0]: OTP read enable          |
| 0x3D00  | OTP DATA00       | 0x00          | RW  | OTP Dump/Load Data00  |
| 0x3D01  | OTP DATA01       | 0x00          | RW  | OTP Dump/Load Data01  |
| 0x3D02  | OTP DATA02       | 0x00          | RW  | OTP Dump/Load Data02  |
| 0x3D03  | OTP DATA03       | 0x00          | RW  | OTP Dump/Load Data03  |
| 0x3D04  | OTP DATA04       | 0x00          | RW  | OTP Dump/Load Data04  |
| 0x3D05  | OTP DATA05       | 0x00          | RW  | OTP Dump/Load Data05  |
| 0x3D06  | OTP DATA06       | 0x00          | RW  | OTP Dump/Load Data06  |

table 4-11 OTP control functions

| address | register name | default value | R/W | description          |
|---------|---------------|---------------|-----|----------------------|
| 0x3D07  | OTP DATA07    | 0x00          | RW  | OTP Dump/Load Data07 |
| 0x3D08  | OTP DATA08    | 0x00          | RW  | OTP Dump/Load Data08 |
| 0x3D09  | OTP DATA09    | 0x00          | RW  | OTP Dump/Load Data09 |
| 0x3D0A  | OTP DATA0A    | 0x00          | RW  | OTP Dump/Load Data0a |
| 0x3D0B  | OTP DATA0B    | 0x00          | RW  | OTP Dump/Load Data0b |
| 0x3D0C  | OTP DATA0C    | 0x00          | RW  | OTP Dump/Load Data0c |
| 0x3D0D  | OTP DATA0D    | 0x00          | RW  | OTP Dump/Load Data0d |
| 0x3D0E  | OTP DATA0E    | 0x00          | RW  | OTP Dump/Load Data0e |
| 0x3D0F  | OTP DATA0F    | 0x00          | RW  | OTP Dump/Load Data0f |
| 0x3D10  | OTP DATA10    | 0x00          | RW  | OTP Dump/Load Data10 |
| 0x3D11  | OTP DATA11    | 0x00          | RW  | OTP Dump/Load Data11 |
| 0x3D12  | OTP DATA12    | 0x00          | RW  | OTP Dump/Load Data12 |
| 0x3D13  | OTP DATA13    | 0x00          | RW  | OTP Dump/Load Data13 |
| 0x3D14  | OTP DATA14    | 0x00          | RW  | OTP Dump/Load Data14 |
| 0x3D15  | OTP DATA15    | 0x00          | RW  | OTP Dump/Load Data15 |
| 0x3D16  | OTP DATA16    | 0x00          | RW  | OTP Dump/Load Data16 |
| 0x3D17  | OTP DATA17    | 0x00          | RW  | OTP Dump/Load Data17 |
| 0x3D18  | OTP DATA18    | 0x00          | RW  | OTP Dump/Load Data18 |
| 0x3D19  | OTP DATA19    | 0x00          | RW  | OTP Dump/Load Data19 |
| 0x3D1A  | OTP DATA1A    | 0x00          | RW  | OTP Dump/Load Data1a |
| 0x3D1B  | OTP DATA1B    | 0x00          | RW  | OTP Dump/Load Data1b |
| 0x3D1C  | OTP DATA1C    | 0x00          | RW  | OTP Dump/Load Data1c |
| 0x3D1F  | OTP DATA1D    | 0x00          | RW  | OTP Dump/Load Data1d |
| 0x3D1E  | OTP DATA1E    | 0x00          | RW  | OTP Dump/Load Data1e |
| 0x3D1F  | OTP DATA1F    | 0x00          | RW  | OTP Dump/Load Data1f |

## 5 image sensor processor digital functions

### 5.1 ISP general controls

The ISP module provides lens correction, gamma, de-noise, sharpen, auto focus, etc. These functions are enabled by registers 0x5000 ~ 0x5005.

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

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x5000  | ISP CONTROL 00 | 0x06          | RW  | ISP Control 00<br>Bit[7]: LENC correction enable<br>0: Disable<br>1: Enable<br>Bit[5]: RAW gamma enable<br>0: Disable<br>1: Enable<br>Bit[2]: Black pixel cancellation enable<br>0: Disable<br>1: Enable<br>Bit[1]: White pixel cancellation enable<br>0: Disable<br>1: Enable<br>Bit[0]: Color interpolation (CIP) enable<br>0: Disable<br>1: Enable |
| 0x5001  | ISP CONTROL 01 | 0x01          | RW  | ISP Control 01<br>Bit[7]: Special Digital Effects (SDE) enable<br>0: Disable<br>1: Enable<br>Bit[5]: Scaling enable<br>0: Disable<br>1: Enable<br>Bit[2]: UV average enable<br>0: Disable<br>1: Enable<br>Bit[1]: Color matrix enable<br>0: Disable<br>1: Enable<br>Bit[0]: Auto white balance (AWB) enable<br>0: Disable<br>1: Enable                |

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

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x5003  | ISP CONTROL 03 | 0x08          | RW  | ISP Control 03<br>Bit[2]: Bin enable<br>0: Disable<br>1: Enable<br>Bit[1]: Draw window for AFC enable<br>0: Disable<br>1: Enable<br>Bit[0]: Solarize enable<br>0: Disable<br>1: Enable   |
| 0x5005  | ISP CONTROL 05 | 0x36          | RW  | ISP Control 05<br>Bit[6]: AWB bias manual enable<br>0: Disable<br>1: Enable<br>Bit[5]: AWB bias on enable<br>0: Disable<br>1: Enable<br>Bit[4]: AWB bias plus enable<br>0: Disable<br>1: Enable<br>Bit[2]: LENC bias on enable<br>0: Disable<br>1: Enable<br>Bit[1]: GMA bias on enable<br>0: Disable<br>1: Enable<br>Bit[0]: LENC bias manual enable<br>0: Disable<br>1: Enable |
| 0x501E  | ISP MISC       | 0x00          | RW  | Bit[6]: Scale ratio manual enable  |

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

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5020  | DITHER CTRL 0 | 0x00          | RW  | RGB Dither Control<br>Bit[6]: Dither register control selection enable<br>0: From register control<br>1: From system control<br>Bit[5:4]: R channel register control when 0x501E[6] = 0<br>00: Not allowed<br>01: RGB444<br>10: RGB565/555<br>11: Not allowed<br>Bit[3:2]: G channel register control when 0x501E[6] = 0<br>00: Not allowed<br>01: RGB444<br>10: RGB565/555<br>11: Not allowed<br>Bit[1:0]: B channel register control when 0x501E[6] = 0<br>00: Not allowed<br>01: RGB444<br>10: RGB565/555<br>11: Not allowed |

## 5.2 lens correction (LENC)

The main purpose of the LENC is to compensate for lens imperfection. According to the area where each pixel is located, the module calculates a gain for the pixel, correcting each pixel with its gain calculated to compensate for the light distribution due to lens curvature. The LENC correcting curve automatic calculation according sensor gain is also added so that the LENC can adapt with the sensor gain. Also, the LENC supports the subsample function in both horizontal and vertical directions. Contact your local OmniVision FAE for lens correction settings (registers 0x5800–0x5849).

**table 5-2** LENC control registers (sheet 1 of 2)

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x5000  | ISP CONTROL 00 | 0x06          | RW  | Bit[7]: LENC correction enable<br>0: Disable<br>1: Enable   |
| 0x5842  | BR HSCALE      | 0x01          | RW  | Bit[2:0]: br h scale[10:8]<br>Reciprocal of horizontal step for BR channel. BR channel in whole image is divided into 5x5 blocks. The step is used to point to the border of the adjacent block |
| 0x5843  | BR HSCAL       | 0x2B          | RW  | Bit[7:0]: br h scale[7:0]<br>Reciprocal of horizontal step for BR channel. BR channel in whole image is divided into 5x5 blocks. The step is used to point to the border of the adjacent block  |
| 0x5844  | BR VSCALE      | 0x01          | RW  | Bit[2:0]: br v scale[10:8]<br>Reciprocal of vertical step for BR channel. BR channel in whole image is divided into 5x5 blocks. The step is used to point to the border of the adjacent block   |
| 0x5845  | BR VSCALE      | 0x8D          | RW  | Bit[7:0]: br v scale[7:0]<br>Reciprocal of vertical step for BR channel. BR channel in whole image is divided into 5x5 blocks. The step is used to point to the border of the adjacent block    |
| 0x5846  | G HSCALE       | 0x01          | RW  | Bit[2:0]: g h scale[10:8]<br>Reciprocal of horizontal step for G channel. G channel in whole image is divided into 6x6 blocks. The step is used to point to the border of the adjacent block    |

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

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5847  | G HSCAL       | 0x8F          | RW  | Bit[7:0]: g h scale[7:0]<br>Reciprocal of horizontal step for G channel. G channel in whole image is divided into 6x6 blocks. The step is used to point to the border of the adjacent block |
| 0x5848  | G VSCALE      | 0x01          | RW  | Bit[2:0]: g v scale[10:8]<br>Reciprocal of vertical step for G channel. G channel in whole image is divided into 6x6 blocks. The step is used to point to the border of the adjacent block  |
| 0x5849  | G VSCALE      | 0x09          | RW  | Bit[7:0]: g v scale[7:0]<br>Reciprocal of vertical step for G channel. G channel in whole image is divided into 6x6 blocks. The step is used to point to the border of the adjacent block   |

### 5.3 auto white balance (AWB)

The main function of Auto White Balance (AWB) is the process of removing unrealistic color casts so that objects which appear white in person are rendered white in the image or video. Thus, the AWB makes sure that the white color is always a white color in different color temperatures. It supports manual white balance and auto white balance. For auto white balance, simple AWB and advanced AWB methods are supplied. Advance AWB takes into account the *color temperature* of a light source. For advanced AWB settings, contact your local OmniVision FAE.

**table 5-3** AWB control registers (sheet 1 of 2)

| address           | register name  | default value | R/W | description   |
|-------------------|----------------|---------------|-----|---|
| 0x5001            | ISP CONTROL 01 | 0x01          | RW  | Bit[0]: Auto white balance enable<br>0: Disable<br>1: Enable  |
| 0x5181            | AWB CONTROL 01 | 0x58          | RW  | Bit[7:6]: Step local<br>Bit[5:4]: Step fast<br>Bit[3]: Slop 8x<br>Bit[2]: Slop 4x<br>Bit[1]: One zone<br>Bit[0]: AVG all  |
| 0x5182            | AWB CONTROL 02 | 0x11          | RW  | Bit[7:4]: Maximum local counter<br>Bit[3:0]: Maximum fast counter   |
| 0x5183            | AWB CONTROL 03 | 0x90          | RW  | Bit[7]: AWB simple enable<br>0: AWB advance<br>1: AWB simple<br>Bit[6]: YUV enable<br>1: Simple YUV enable<br>Bit[5]: AWB preset<br>Bit[4]: AWB simf<br>Bit[3:2]: AWB win |
| 0x5184            | AWB CONTROL 04 | 0x25          | RW  | Bit[7:6]: Counter area selection<br>Bit[5]: G enable<br>Bit[4:2]: Counter limit control<br>Bit[1:0]: Counter threshold  |
| 0x5185            | AWB CONTROL 05 | 0x24          | RW  | Bit[7:4]: Stable range unstable<br>Threshold for unstable to stable change<br>Bit[3:0]: Stable range stable<br>Threshold for stable to un-stable change                   |
| 0x5186~<br>0x5190 | AWB CONTROL    | –             | –   | Advanced AWB Control Registers  |
| 0x5191            | AWB CONTROL 17 | 0xFF          | RW  | Bit[7:0]: AWB top limit   |
| 0x5192            | AWB CONTROL 18 | 0x00          | RW  | Bit[7:0]: AWB bottom limit  |
| 0x5193            | AWB CONTROL 19 | 0xF0          | RW  | Bit[7:0]: Red limit   |



table 5-3 AWB control registers (sheet 2 of 2)

| address | register name      | default value | R/W | description  |
|---------|--------------------|---------------|-----|--|
| 0x5194  | AWB CONTROL 20     | 0xF0          | RW  | Bit[7:0]: Green limit  |
| 0x5195  | AWB CONTROL 21     | 0xF0          | RW  | Bit[7:0]: Blue limit   |
| 0x5196  | AWB CONTROL 22     | 0x03          | RW  | Bit[5]: AWB freeze<br>Bit[3:2]: AWB simple selection<br>00: AWB simple from after AWB gain<br>01: AWB simple from after RAW GMA<br>10: AWB simple from after AWB gain<br>11: AWB simple from after RAW GMA<br>Bit[1]: Fast enable<br>Bit[0]: AWB bias stat |
| 0x5197  | AWB CONTROL 23     | 0x02          | RW  | Bit[7:0]: Local limit  |
| 0x519E  | AWB CONTROL 30     | 0x00          | RW  | Bit[3]: Local limit select<br>Bit[2]: Simple stable select   |
| 0x519F  | AWB CURRENT R GAIN | –             | R   | Bit[3:0]: Current R setting[11:8]  |
| 0x51A0  | AWB CURRENT R GAIN | –             | R   | Bit[7:0]: Current R setting[7:0]   |
| 0x51A1  | AWB CURRENT G GAIN | –             | R   | Bit[3:0]: Current G setting[11:8]  |
| 0x51A2  | AWB CURRENT G GAIN | –             | R   | Bit[7:0]: Current G setting[7:0]   |
| 0x51A3  | AWB CURRENT B GAIN | –             | R   | Bit[3:0]: Current B setting[11:8]  |
| 0x51A4  | AWB CURRENT B GAIN | –             | R   | Bit[7:0]: Current B setting[7:0]   |
| 0x51A5  | AWB AVERAGE B      | –             | R   | Bit[7:0]: Average r[9:2]   |
| 0x51A6  | AWB AVERAGE B      | –             | R   | Bit[7:0]: Average g[9:2]   |
| 0x51A7  | AWB AVERAGE B      | –             | R   | Bit[7:0]: Average b[9:2]   |
| 0x51D0  | AWB CONTROL74      | –             | R   | Bit[5]: R large<br>Bit[4]: G large<br>Bit[3]: B large<br>Bit[2:1]: Current type  |

## 5.4 raw gamma

The main purpose of the Gamma (GMA) function is to compensate for the non-linear characteristics of the sensor. GMA converts the pixel values according to the Gamma curve to compensate the sensor output under different light strengths. The non-linear gamma curve is approximately constructed with different linear functions. Raw gamma compensates the image in the RAW domain.

**table 5-4** raw gamma control registers

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x5000  | ISP CONTROL 00 | 0x06          | RW  | Bit[5]: Raw gamma enable<br>0: Disable GMA<br>1: Enable GMA      |
| 0x5001  | ISP CONTROL 01 | 0x01          | RW  | Bit[7]: Special digital effect enable<br>0: Disable<br>1: Enable |
| 0x5481  | GAMMA YST00    | 0x26          | RW  | Bit[7:0]: Y yst 00   |
| 0x5482  | GAMMA YST01    | 0x35          | RW  | Bit[7:0]: Y yst 01   |
| 0x5483  | GAMMA YST02    | 0x48          | RW  | Bit[7:0]: Y yst 02   |
| 0x5484  | GAMMA YST03    | 0x57          | RW  | Bit[7:0]: Y yst 03   |
| 0x5485  | GAMMA YST04    | 0x63          | RW  | Bit[7:0]: Y yst 04   |
| 0x5486  | GAMMA YST05    | 0x6E          | RW  | Bit[7:0]: Y yst 05   |
| 0x5487  | GAMMA YST06    | 0x77          | RW  | Bit[7:0]: Y yst 06   |
| 0x5488  | GAMMA YST07    | 0x80          | RW  | Bit[7:0]: Y yst 07   |
| 0x5489  | GAMMA YST08    | 0x88          | RW  | Bit[7:0]: Y yst 08   |
| 0x548A  | GAMMA YST09    | 0x96          | RW  | Bit[7:0]: Y yst 09   |
| 0x548B  | GAMMA YST0A    | 0xA3          | RW  | Bit[7:0]: Y yst 0A   |
| 0x548C  | GAMMA YST0B    | 0xAF          | RW  | Bit[7:0]: Y yst 0B   |
| 0x548D  | GAMMA YST0C    | 0xC5          | RW  | Bit[7:0]: Y yst 0C   |
| 0x548E  | GAMMA YST0D    | 0xD7          | RW  | Bit[7:0]: Y yst 0D   |
| 0x548F  | GAMMA YST0E    | 0xE8          | RW  | Bit[7:0]: Y yst 0E   |
| 0x5490  | GAMMA YST0F    | 0x0F          | RW  | Bit[7:0]: Y yst 0F   |

## 5.5 defect pixel cancellation (DPC)

Due to processes and other reasons, pixel defects in the sensor array will occur. Thus, these bad or wounded pixels will generate wrong color values. The main purpose of Defect Pixel Cancellation (DPC) function is to remove the effect caused by these bad or wounded pixels. Also, some special functions are available for those pixels located at the image boundary. To remove the defect pixel effect correctly, the proper threshold should first be determined.

**table 5-5** DPC control registers

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x5000  | ISP CONTROL 00 | 0x06          | RW  | Bit[2]: Black pixel cancellation enable<br>0: Disable<br>1: Enable<br>Bit[1]: White pixel cancellation enable<br>0: Disable<br>1: Enable |

## 5.6 color interpolation (CIP)

The CIP functions include de-noising of raw images, RAW to RGB interpolation, and edge enhancement. In sensor RAW format, each pixel will be either R, G or B. CIP will calculate the other two color values using the neighboring pixel of the same color. Thus, we can get the full RGB information for each pixel. For edge enhancement, the OV5640 provides both manual and auto modes.

**table 5-6** CIP control registers

| address | register name              | default value | R/W | description  |
|---------|----------------------------|---------------|-----|--|
| 0x5000  | ISP CONTROL 00             | 0x06          | RW  | Bit[0]: Color interpolation enable<br>0: Disable<br>1: Enable  |
| 0x5301  | CIP SHARPENMT THRESHOLD 2  | 0x48          | RW  | Color Interpolation Sharpen MT Threshold 2   |
| 0x5302  | CIP SHARPENMT OFFSET1      | 0x18          | RW  | CIP Sharpen MT Offset1 (Y edge mt manual setting when 0x5308[6]=1)   |
| 0x5303  | CIP SHARPENMT OFFSET2      | 0x0E          | RW  | CIP Sharpen MT Offset2   |
| 0x5304  | CIP DNS THRESHOLD 1        | 0x08          | RW  | CIP DNS Threshold 1  |
| 0x5305  | CIP DNS THRESHOLD 2        | 0x48          | RW  | CIP DNS Threshold 2  |
| 0x5306  | CIP DNS OFFSET1            | 0x09          | RW  | CIP DNS Offset1 (DNS threshold manual setting when 0x5308[4]=1)  |
| 0x5307  | CIP DNS OFFSET2            | 0x16          | RW  | CIP DNS Offset2  |
| 0x5308  | CIP CTRL                   | 0x25          | RW  | Bit[6]: CIP edge MT manual enable<br>Bit[4]: CIP DNS manual enable<br>Bit[2:0]: CIP threshold for BR sharpen |
| 0x5309  | CIP SHARPENTH THRESHOLD 1  | 0x08          | RW  | CIP Sharpen TH Threshold 1   |
| 0x530A  | CIP SHARPENTH THRESHOLD 2  | 0x48          | RW  | CIP Sharpen TH Threshold 2   |
| 0x530B  | CIP SHARPENTH OFFSET1      | 0x04          | RW  | CIP Sharpen TH Offset1 (Sharpen threshold manual setting when 0x5308[6]=1)                                   |
| 0x530C  | CIP SHARPENTH OFFSET2      | 0x06          | RW  | CIP Sharpen TH Offset2   |
| 0x530D  | CIP EDGE MT AUTO           | –             | R   | CIP Edge MT Auto Read  |
| 0x530E  | CIP DNS THRESHOLD AUTO     | –             | R   | CIP DNS Threshold Auto Read  |
| 0x530F  | CIP SHARPEN THRESHOLD AUTO | –             | R   | CIP Sharpen Threshold Auto Read  |

## 5.7 color matrix (CMX)

The main purpose of the Color Matrix (CMX) function is to cancel out crosstalk and convert color space. Given the color correction matrix, CCM, and RGB to YUV conversion matrix, RGB2YUV, the combined matrix is:

$$\text{CMX} = \begin{bmatrix} \text{cmx00} & \text{cmx01} & \text{cmx02} \\ \text{cmx10} & \text{cmx11} & \text{cmx12} \\ \text{cmx20} & \text{cmx21} & \text{cmx22} \end{bmatrix} = \text{RGB2YUV} \times \text{CCM} \times \begin{bmatrix} 1 & -0.25 & 0.75 \\ 1 & -0.25 & -0.25 \\ 1 & 0.75 & -0.25 \end{bmatrix}$$

$$\text{where } \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \text{CCM} \begin{bmatrix} R_0 \\ G_0 \\ B_0 \end{bmatrix}$$

The CMX is then normalized by  $2^{0x5394[3:0]}$ .

**table 5-7** CMX control registers

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x5001  | ISP CONTROL 01 | 0x01          | RW  | Bit[1]: Color matrix enable<br>0: Disable<br>1: Enable  |
| 0x5382  | CMX2           | 0x64          | RW  | Bit[7:0]: CMX2 for Y  |
| 0x5383  | CMX3           | 0x08          | RW  | Bit[7:0]: CMX3 for Y  |
| 0x5384  | CMX4           | 0x30          | RW  | Bit[7:0]: CMX4 for U  |
| 0x5385  | CMX5           | 0x90          | RW  | Bit[7:0]: CMX5 for U  |
| 0x5386  | CMX6           | 0xC0          | RW  | Bit[7:0]: CMX6 for U  |
| 0x5387  | CMX7           | 0xA0          | RW  | Bit[7:0]: CMX7 for V  |
| 0x5388  | CMX8           | 0x98          | RW  | Bit[7:0]: CMX8 for V  |
| 0x5389  | CMX9           | 0x08          | RW  | Bit[7:0]: CMX9 for V  |
| 0x538A  | CMXSIGN        | 0x01          | RW  | Cmxsign<br>Bit[0]: CMX9 sign  |
| 0x538B  | CMXSIGN        | 0x98          | RW  | Cmxsign<br>Bit[7]: CMX8 sign<br>Bit[6]: CMX7 sign<br>Bit[5]: CMX6 sign<br>Bit[4]: CMX5 sign<br>Bit[3]: CMX4 sign<br>Bit[2]: CMX3 sign<br>Bit[1]: CMX2 sign<br>Bit[0]: CMX1 sign |

## 5.8 UV average

The main function of the UV average is to average the U/V channel value using special filters.

**table 5-8** UV average register

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x5001  | ISP CONTROL 01 | 0x4F          | RW  | Bit[2]: UV average enable<br>0: Disable<br>1: Enable |

## 5.9 scaling

The main purpose of the scaling function is to zoom out the image. According to the new width and new height of the new image, the module uses the values of several pixels to generate the values of one pixel. The values of some pixels are divided and used in two or more adjacent pixels. The scaling function supports up to 32x scale.

**table 5-9** UV average register

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x5001  | ISP CONTROL 01 | 0x01          | RW  | Bit[5]: Scale enable<br>0: Disable<br>1: Enable  |
| 0x5601  | SCALE CTRL 1   | 0x00          | RW  | Bit[6:4]: HDIV RW<br>DCW scale times<br>000: DCW 1 time<br>001: DCW 2 time<br>010: DCW 4 time<br>100: DCW 8 time<br>101: DCW 16 time<br>Others: DCW 16 time<br>Bit[2:0]: VDIV RW<br>DCW scale times<br>000: DCW 1 time<br>001: DCW 2 time<br>010: DCW 4 time<br>100: DCW 8 time<br>101: DCW 16 time<br>Others: DCW 16 time |
| 0x5602  | SCALE CTRL 2   | 0x02          | RW  | XSC High Bits  |
| 0x5603  | SCALE CTRL 3   | 0x00          | RW  | XSC Low Bits   |
| 0x5604  | SCALE CTRL 4   | 0x02          | RW  | YSC High Bits  |
| 0x5605  | SCALE CTRL 5   | 0x00          | RW  | YSC Low Bits   |
| 0x5606  | SCALE CTRL 6   | 0x00          | RW  | Bit[3:0]: Voffset  |

## 5.10 UV adjust

The main function of the UV adjust is to adjust the U/V channel value according to sensor gain. It supports both manual and auto modes.

The UV adjust function is integrated in SDE. The main function of the UV adjust is to adjust the U/V channel value according to sensor gain. It supports both manual and auto modes

### 5.10.1 manual mode

By setting SDE CTRL 0 [1] (0x5580) to 1 and SDE CTRL 8 0x5588[6] to 1, UV adjust is controlled only by register SAT U (0x5583[7:0]) and SAT V (0x5584[7:0]) for U and V gains.

### 5.10.2 auto mode

When the UV adjust is set for auto mode(0x5580[1]=1 and 0x5588[6]=0), the UV adjust curve parameters (see **figure 5-1**) should be entered into the corresponding registers. The UV adjust parameters, UV adj th1, UV adj th2, and offset low, offset high should be entered into the registers to set the curve. To get these values, first set the values of UV adj th1, UV adj th2, offset low and offset high. Then, calculate the values of a and k as follows:

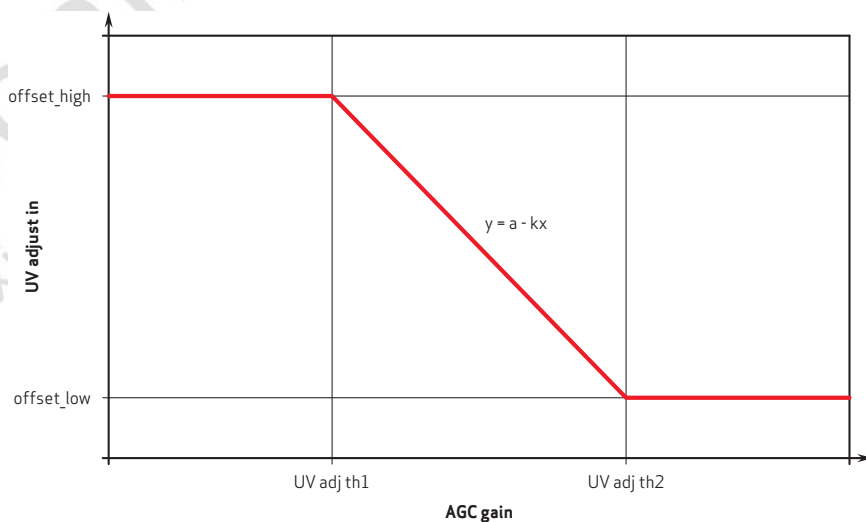
$$k = (\text{offset high} - \text{offset low}) / (\text{UV adj th2} - \text{UV adj th1})$$

$$a = \text{offset high} + (\text{offset high} - \text{offset low}) / (\text{UV adj th2} - \text{UV adj th1})$$

Registers to be changed:

- UV adj th1[8:0] = registers 0x5589[7:0]
- UV adj th2[8:0] = registers {0x558A[0], 0x558B[7:0]}
- offset high = register 0x5583[7:0] (when 0x5580[1]=1 and 0x5588[6]=0)
- offset low = register 0x5584[7:0] (when 0x5580[1]=1 and 0x5588[6]=0)

**figure 5-1** UV adjust graph



## 5.11 special digital effects (SDE)

The Special Digital Effects (SDE) functions include hue/saturation control, brightness, contrast, etc. SDE also supports negative, black/white, sepia, greenish, blueish, redish, solarize and other image effects.

**table 5-10** SDE control registers

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x5001  | ISP CONTROL 01 | 0x01          | RW  | Bit[7]: Special digital effect enable<br>0: Disable<br>1: Enable  |
| 0x5581  | SDE CTRL1      | 0x80          | RW  | Bit[7:0]: Hue cos coefficient   |
| 0x5582  | SDE CTRL2      | 0x00          | RW  | Bit[7:0]: Hue sin coefficient   |
| 0x5583  | SDE CTRL3      | 0x40          | RW  | Bit[7:0]: Saturation U when 0x5580[1]=1 and 0x5588[6]=1, max value for UV adjust when 0x5580[1]=1 and 0x5588[6]=0; or fixed U when 0x5580[3]=1  |
| 0x5584  | SDE CTRL4      | 0x40          | RW  | Bit[7:0]: Saturation V when 0x5580[1]=1 and 0x5588[6]=1, min value for UV adjust when 0x5580[1]=1 and 0x5588[6]=0; or Vreg when 0x5580[4]=1   |
| 0x5585  | SDE CTRL5      | 0x00          | RW  | Bit[7:0]: Yoffset for contrast when 0x5044[3]=1; or fixed Y when 0x5580[7]=1  |
| 0x5586  | SDE CTRL6      | 0x20          | RW  | Bit[7:0]: Y gain for contrast   |
| 0x5587  | SDE CTRL7      | 0x00          | RW  | Bit[7:0]: Y bright for contrast   |
| 0x5588  | SDE CTRL8      | 0x01          | RW  | Bit[6]: UV adjust manual enable<br>Bit[5]: Sign5 for hue V, cos<br>Bit[4]: Sign4 for hue U, cos<br>Bit[3]: Sign3 Y bright sign for contrast<br>0: Keep Y bright sign<br>1: Negative Y bright sign<br>Bit[2]: Sign2<br>Y offset sign for contrast when 0x5044[3]=1<br>0: Keep Y offset sign<br>1: Negative Y offset sign<br>Bit[1]: Sign1 for hue V, sin<br>Bit[0]: Sign0 for hue U, sin |
| 0x5589  | SDE CTRL9      | 0x01          | RW  | Bit[7:0]: UV adjust threshold 1<br>Valid when 0x5580[1]=1   |
| 0x558A  | SDE CTRL10     | 0x01          | RW  | Bit[0]: UV adjust threshold 2[8]<br>Valid when 0x5580[1]=1  |
| 0x558B  | SDE CTRL11     | 0xFF          | RW  | Bit[7:0]: UV adjust threshold 2[7:0]<br>Valid when 0x5580[1]=1  |
| 0x558C  | SDE CTRL12     | –             | R   | Bit[7:0]: UV adjust value read out  |



## 5.12 ISP format

**table 5-11** ISP format control registers

| address | register name      | default value | R/W | description  |
|---------|--------------------|---------------|-----|--|
| 0x501F  | FORMAT MUX CONTROL | 0x00          | RW  | Format MUX Control<br>Bit[2:0]: Format selection<br>000: ISP YUV422<br>001: ISP RGB<br>010: ISP dither<br>011: ISP RAW (DPC)<br>100: SNR RAW<br>101: ISP RAW (CIP) |

## 5.13 draw window

The draw window module is used to display a window on top of live video. It is usually used by autofocus to display a focus window.

**table 5-12** draw window registers (sheet 1 of 2)

| address | register name                      | default value | R/W | description   |
|---------|------------------------------------|---------------|-----|---|
| 0x5003  | ISP CONTROL 03                     | 0x08          | RW  | Bit[1]: Draw window for AFC enable<br>0: Disable<br>1: Enable   |
| 0x501F  | FORMAT MUX CONTROL                 | 0x00          | RW  | Bit[2:0]: Format select<br>000: ISP YUV422<br>001: ISP RGB<br>010: ISP dither<br>011: ISP RAW (DPC)<br>100: SNR RAW<br>101: ISP RAW (CIP) |
| 0x5027  | DRAW WINDOW CONTROL 00             | 0x02          | RW  | Bit[0]: Draw window control<br>0: No fixed Y<br>1: Fixed Y  |
| 0x5028  | DRAW WINDOW LEFT POSITION CONTROL  | 0x04          | RW  | Bit[3:0]: Draw window left[11:8] high byte  |
| 0x5029  | DRAW WINDOW LEFT POSITION CONTROL  | 0x90          | RW  | Bit[7:0]: Draw window left[7:0] low byte  |
| 0x502A  | DRAW WINDOW RIGHT POSITION CONTROL | 0x05          | RW  | Bit[3:0]: Draw window right[11:8] high byte   |
| 0x502B  | DRAW WINDOW RIGHT POSITION CONTROL | 0x90          | RW  | Bit[7:0]: Draw window right[7:0] low byte   |

table 5-12 draw window registers (sheet 2 of 2)

| address | register name                                 | default value | R/W | description   |
|---------|---|---------------|-----|---|
| 0x502C  | DRAW WINDOW TOP POSITION CONTROL              | 0x03          | RW  | Bit[2:0]: Draw window top[10:8] high byte                       |
| 0x502D  | DRAW WINDOW TOP POSITION CONTROL              | 0x6C          | RW  | Bit[7:0]: Draw window top[7:0] low byte                         |
| 0x502E  | DRAW WINDOW BOTTOM POSITION CONTROL           | 0x04          | RW  | Bit[2:0]: Draw window bottom[10:8] high byte                    |
| 0x502F  | DRAW WINDOW BOTTOM POSITION CONTROL           | 0x2C          | RW  | Bit[7:0]: Draw window bottom[7:0] low byte                      |
| 0x5030  | DRAW WINDOW HORIZONTAL BOUNDARY WIDTH CONTROL | 0x00          | RW  | Bit[3:0]: Draw window horizontal boundary width[11:8] high byte |
| 0x5031  | DRAW WINDOW HORIZONTAL BOUNDARY WIDTH CONTROL | 0x14          | RW  | Bit[7:0]: Draw window horizontal boundary width[7:0] low byte   |
| 0x5032  | DRAW WINDOW VERTICAL BOUNDARY WIDTH CONTROL   | 0x00          | RW  | Bit[2:0]: Draw window vertical boundary width[10:8] high byte   |
| 0x5033  | DRAW WINDOW VERTICAL BOUNDARY WIDTH CONTROL   | 0x14          | RW  | Bit[7:0]: Draw window vertical boundary width[7:0] low byte     |
| 0x5034  | DRAW WINDOW Y CONTROL                         | 0x80          | RW  | Bit[7:0]: Fixed Y for draw window                               |
| 0x5035  | DRAW WINDOW U CONTROL                         | 0x2A          | RW  | Bit[7:0]: Fixed U for draw window                               |
| 0x5036  | DRAW WINDOW V CONTROL                         | 0x14          | RW  | Bit[7:0]: Fixed V for draw window                               |

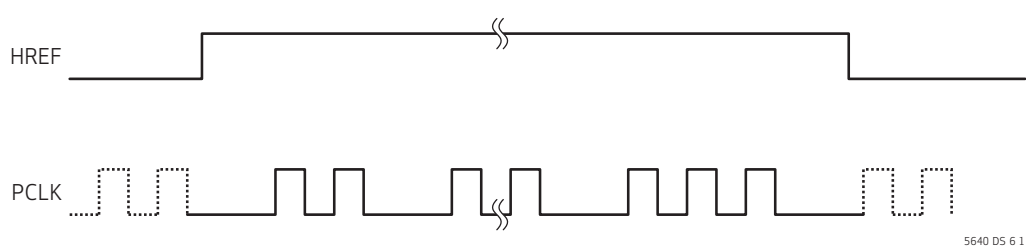
## 6 image sensor output interface digital functions

### 6.1 compression engine

#### 6.1.1 compression mode 1 timing

The whole frame has only one line. PCLK will be gated when there is no valid image data transmitted.

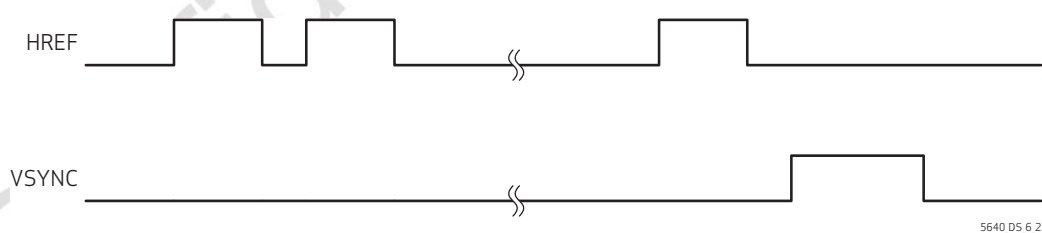
**figure 6-1** compression mode 1 timing



#### 6.1.2 compression mode 2 timing

Compression data is transmitted with programmable line width. PCLK is free running. The last line may contain dummy data to match the width. By default, the line number varies from frame to frame. The user can set register 4600[5] (0x4600) to ensure every frame has a fixed line number (programmable).

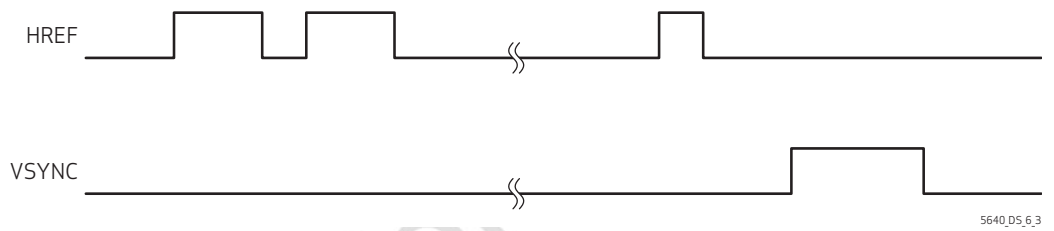
**figure 6-2** compression mode 2 timing



6.1.3 compression mode 3 timing

Compression data is transmitted with programmable width. The last line width maybe different from the other line (there is no dummy data). In each frame, the line number may be different.

figure 6-3 compression mode 3 timing

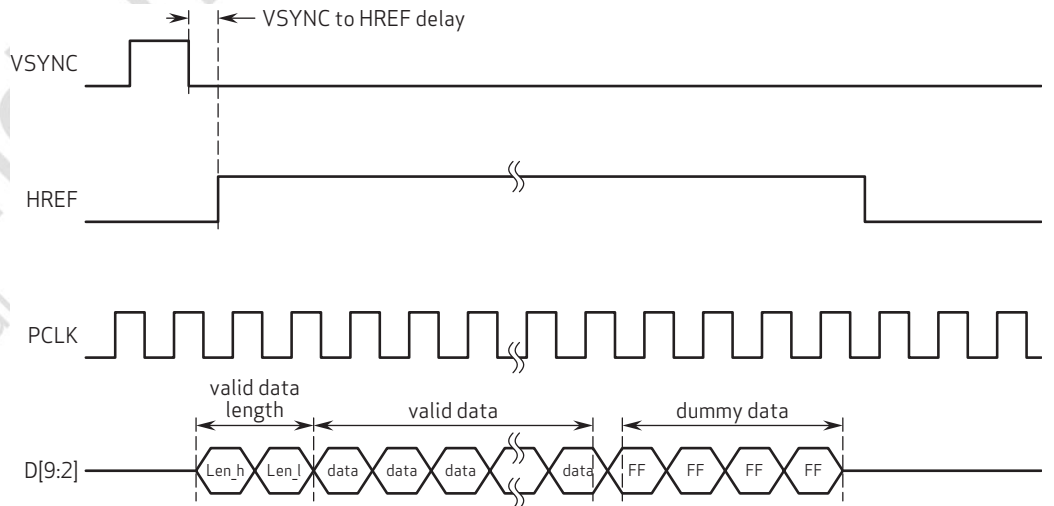


5640\_DS\_6\_3

6.1.4 compression mode 4 timing

The width and height are fixed in each frame. The first two bytes are valid data length in every line, followed by valid image data. Dummy data (0xFF) may be used as padding at each line end if the current valid image data is less than the line width.

figure 6-4 compression mode 4 timing

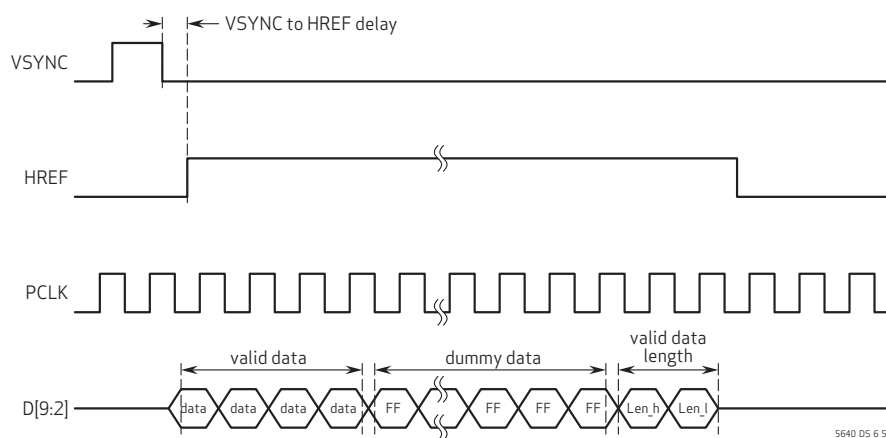


5640\_DS\_6\_4

### 6.1.5 compression mode 5 timing

The width and height are fixed in each frame. Every line begins with valid image data. Dummy data may be used as padding at each line end if the current valid image data is less than the line width. The last two bytes of every line is valid data length.

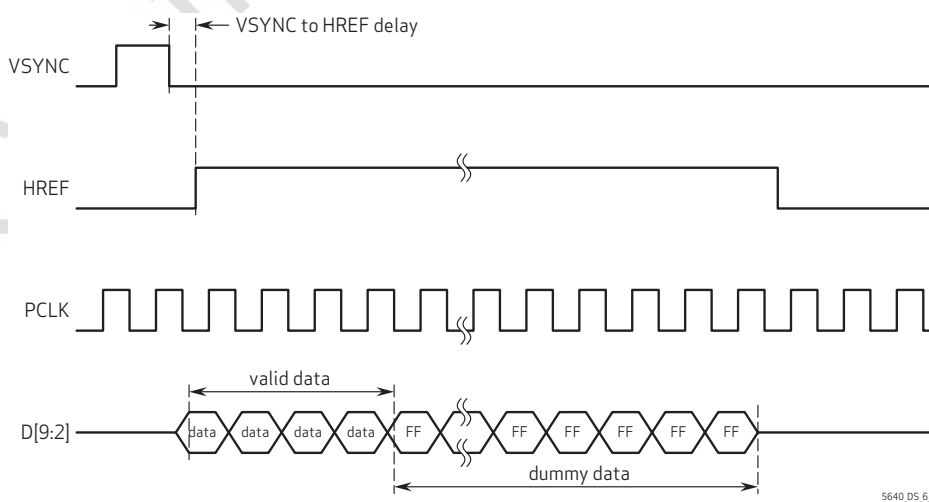
**figure 6-5** compression mode 5 timing



### 6.1.6 compression mode 6 timing

The width and height are fixed in each frame. Every line begins with valid image data. Dummy data may be used as padding at each line end if the current valid image data less than the line width.

**figure 6-6** compression mode 6 timing



## 6.1.7 compression mode control

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

| address | register name      | default value | R/W | description  |
|---------|--------------------|---------------|-----|--|
| 0x3821  | COMPRESSION ENABLE | 0x00          | RW  | Bit[5]: JPEG enable  |
| 0x4600  | VFIFO CTRL00       | 0x80          | RW  | Bit[5]: JPEG output fixed height enable<br>0: In JPEG mode2<br>JPEG height is different in each frame<br>1: In JPEG mode2<br>JPEG height is fixed in each frame  |
| 0x4602  | VFIFO HSIZE        | 0x04          | RW  | JPEG Output Width High Byte  |
| 0x4603  | VFIFO HSIZE        | 0x00          | RW  | JPEG Output Width Low Byte   |
| 0x4604  | VFIFO VSIZE        | 0x03          | RW  | JPEG Output Height High Byte   |
| 0x4605  | VFIFO HSIZE        | 0x00          | RW  | JPEG Output Height Low Byte  |
| 0x460C  | VFIFO CTRL0C       | 0x20          | RW  | Bit[7:4]: JPEG dummy data pad speed  |
| 0x460D  | VFIFO CTRL0D       | 0x00          | RW  | JPEG PAD Dummy Data  |
| 0x4713  | JPG MODE SELECT    | 0x02          | RW  | Bit[2:0]: JPEG mode select<br>001: JPEG mode 1<br>010: JPEG mode 2<br>011: JPEG mode 3<br>100: JPEG mode 4<br>101: JPEG mode 5<br>110: JPEG mode 6   |
| 0x471F  | DVP HREF CTRL      | 0x40          | RW  | HREF Minimum Blanking in JPEG Mode23   |
| 0x4723  | DVP CTRL23         | 0x00          | RW  | DVP JPEG Mode456 Skip Line Number  |
| 0x4400  | JPEG CTRL00        | 0x81          | RW  | Bit[7]: input_format<br>0: YUV420<br>1: YUV422<br>Bit[6:0]: JFIFO read speed control   |
| 0x4401  | JPEG CTRL01        | 0x01          | RW  | Bit[7:4]: SFIFO output buffer speed control<br>Bit[3]: Read SRAM enable when blanking<br>0: Disable<br>1: Enable<br>Bit[2]: Read SRAM at first blanking<br>0: Disable<br>1: Enable<br>Bit[1:0]: SFIFO read speed control |

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

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x4404  | JPEG CTRL04    | 0x24          | RW  | Bit[7]: jfifo_pwrn<br>Bit[6]: SFIFO pwrn<br>Bit[5]: Header output enable<br>Bit[4]: Enable gated clock<br>0: Disable gated clock<br>1: Enable gated clock<br>Bit[3]: Substitute 0xFF to 0xFE in QT<br>Bit[2:0]: Quantization rounding<br>Bias: set value = Bias/8 |
| 0x4417  | JFIFO OVERFLOW | –             | R   | Bit[0]: JFIFO overflow indicator  |

## 6.2 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 6-2 system control registers (sheet 1 of 3)

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x3000  | SYSTEM RESET00 | 0x30          | RW  | Reset for Individual Block<br>(0: enable block; 1: reset block)<br>Bit[7]: Reset BIST<br>Bit[6]: Reset MCU program memory<br>Bit[5]: Reset MCU<br>Bit[4]: Reset OTP<br>Bit[3]: Reset STB<br>Bit[2]: Reset d5060<br>Bit[1]: Reset timing control<br>Bit[0]: Reset array control |
| 0x3001  | SYSTEM RESET01 | 0x08          | RW  | Reset for Individual Block<br>(0: enable block; 1: reset block)<br>Bit[7]: Reset AWB registers<br>Bit[6]: Reset AFC<br>Bit[5]: Reset ISP<br>Bit[4]: Reset FC<br>Bit[3]: Reset S2P<br>Bit[2]: Reset BLC<br>Bit[1]: Reset AEC registers<br>Bit[0]: Reset AEC                     |

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

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x3002  | SYSTEM RESET02 | 0x1C          | RW  | Reset for Individual Block<br>(0: enable block; 1: reset block)<br>Bit[7]: Reset VFIFO<br>Bit[5]: Reset format<br>Bit[4]: Reset JFIFO<br>Bit[3]: Reset SFIFO<br>Bit[2]: Reset JPG<br>Bit[1]: Reset format MUX<br>Bit[0]: Reset average  |
| 0x3003  | SYSTEM RESET03 | 0x00          | RW  | Reset for Individual Block<br>(0: enable block; 1: reset block)<br>Bit[5]: Reset digital gain compensation<br>Bit[4]: Reset SYNC FIFO<br>Bit[3]: Reset PSRAM<br>Bit[2]: Reset ISP FC<br>Bit[1]: Reset MIPI<br>Bit[0]: Reset DVP   |
| 0x3004  | CLOCK ENABLE00 | 0xCF          | RW  | Clock Enable Control<br>(0: disable clock; 1: enable clock)<br>Bit[7]: Enable BIST clock<br>Bit[6]: Enable MCU program memory clock<br>Bit[5]: Enable MCU clock<br>Bit[4]: Enable OTP clock<br>Bit[3]: Enable STROBE clock<br>Bit[2]: Enable D5060 clock<br>Bit[1]: Enable timing control clock<br>Bit[0]: Enable array control clock |
| 0x3005  | CLOCK ENABLE01 | 0xF7          | RW  | Clock Enable Control<br>(0: disable clock; 1: enable clock)<br>Bit[7]: Enable AWB register clock<br>Bit[6]: Enable AFC clock<br>Bit[5]: Enable ISP clock<br>Bit[4]: Enable FC clock<br>Bit[3]: Enable S2P clock<br>Bit[2]: Enable BLC clock<br>Bit[1]: Enable AEC register clock<br>Bit[0]: Enable AEC clock                          |
| 0x3006  | CLOCK ENABLE02 | 0xE3          | RW  | Clock Enable Control<br>(0: disable clock; 1: enable clock)<br>Bit[7]: Enable PSRAM clock<br>Bit[6]: Enable FMT clock<br>Bit[5]: Enable JPEG 2x clock<br>Bit[3]: Enable JPEG clock<br>Bit[1]: Enable format MUX clock<br>Bit[0]: Enable average clock   |



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

| address | register name   | default value | R/W | description  |
|---------|-----------------|---------------|-----|--|
| 0x3007  | CLOCK ENABLE03  | 0xFF          | RW  | Clock Enable Control<br>(0: disable clock; 1: enable clock)<br>Bit[7]: Enable digital gain compensation clock<br>Bit[6]: Enable SYNC FIFO clock<br>Bit[5]: Enable ISPFC SCLK clock<br>Bit[4]: Enable MIPI PCLK clock<br>Bit[3]: Enable MIPI clock<br>Bit[2]: Enable DVP PCLK clock<br>Bit[1]: Enable VFIFO PCLK clock<br>Bit[0]: Enable VFIFO SCLK clock |
| 0x3008  | SYSTEM CTROL0   | 0x02          | RW  | System Control<br>Bit[7]: Software reset<br>Bit[6]: Software power down  |
| 0x3034  | SC PLL CONTRL0  | 0x1A          | RW  | Bit[6:4]: PLL charge pump control<br>Bit[3:0]: MIPI bit mode<br>0x8: 8-bit mode<br>0xA: 10-bit mode  |
| 0x3035  | SC PLL CONTRL1  | 0x11          | RW  | Bit[7:4]: System clock divider<br>Slow down all clocks<br>Bit[3:0]: Scale divider for MIPI<br>MIPI PCLK/SERCLK can be slowed down  |
| 0x3036  | SC PLL CONTRL2  | 0x69          | RW  | Bit[7:0]: PLL multiplier (4~252)<br>Can be any integer for 4~127 and only even integer for 128~252   |
| 0x3037  | SC PLL CONTRL3  | 0x03          | RW  | Bit[4]: PLL root divider<br>0: Bypass<br>1: Divided by 2<br>Bit[3:0]: PLL pre-divider<br>1, 2, 3, 4, 6, 8  |
| 0x3039  | SC PLL CONTRL 5 | 0x00          | RW  | Bit[7]: PLL bypass   |

## 6.3 microcontroller unit (MCU)

The MCU firmware can be downloaded by writing to registers starting from 0x8000. A total of 6 KB of program memory can be used for program storage. Before downloading the firmware, the user must enable the MCU clock by setting register 0x3000[5] to 1'b1. After downloading the firmware, set register 0x3000[5] to 1'b0 to enable the MCU. The MCU interrupts are triggered by several internal signals for firmware development.

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

| address | register name              | default value | R/W | description   |
|---------|----------------------------|---------------|-----|---|
| 0x3F00  | MC CTRL 00                 | 0x00          | RW  | Bit[0]: MCU soft reset<br>1: Reset MCU  |
| 0x3F01  | MC INTERRUPT MASK0         | 0x00          | RW  | Mask0 for Interrupt<br>(0: disable interrupt bit; 1: enable interrupt bit)<br>Bit[7]: JFIFO over flow<br>Bit[6]: JFIFO end of image<br>Bit[5]: ISP end of frame<br>Bit[4]: ISP start of frame<br>Bit[3]: AFC done<br>Bit[2]: AWB done<br>Bit[1]: VFIFO full<br>Bit[0]: VFIFO empty                        |
| 0x3F02  | MC INTERRUPT MASK1         | 0x00          | RW  | Mask1 for Interrupt<br>(0: disable interrupt bit; 1: enable interrupt bit)<br>Bit[7]: AEC done<br>Bit[6]: ISP average done<br>Bit[5]: AEC trigger<br>Bit[4]: JPG over size<br>Bit[3]: SRM operation start<br>Bit[2]: SRM operation done<br>Bit[1]: DVP frame counter change<br>Bit[0]: BLC start of frame |
| 0x3F03  | MC READ INTERRUPT ADDRESS  | 0x70          | RW  | Bit[7:0]: Set high byte for SCCB address that will trigger interrupt when read  |
| 0x3F04  | MC READ INTERRUPT ADDRESS  | 0x00          | RW  | Bit[7:0]: Set low byte for SCCB address that will trigger interrupt when read   |
| 0x3F05  | MC WRITE INTERRUPT ADDRESS | 0x70          | RW  | Bit[7:0]: Set high byte for SCCB address that will trigger interrupt when written   |
| 0x3F06  | MC WRITE INTERRUPT ADDRESS | 0x04          | RW  | Bit[7:0]: Set low byte for SCCB address that will trigger interrupt when written  |

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

| address | register name        | default value | R/W | description   |
|---------|----------------------|---------------|-----|---|
| 0x3F0C  | MC INTERRUPT0 STATUS | 0x00          | RW  | Interrupt0 Status Indicator<br>Bit[5]: ISP EOF<br>Bit[4]: ISP SOF<br>Bit[2]: AWB done<br>Bit[1]: VFIFO full<br>Bit[0]: VFIFO empty  |
| 0x3F0D  | MC INTERRUPT1 STATUS | 0x00          | RW  | Interrupt1 Status Indicator<br>Bit[7]: AEC done<br>Bit[6]: Average done<br>Bit[5]: AEC trigger<br>Bit[3]: MIPI turn around<br>Bit[2]: MIPI low power contention detect<br>Bit[0]: BLC SOF |

## 6.4 frame control (FC)

Frame control (FC) is used to mask some specified frame by setting the appropriate registers.

table 6-4 FC control registers

| address | register name    | default value | R/W | description  |
|---------|------------------|---------------|-----|--|
| 0x4201  | FRAME CONTROL 00 | 0x00          | RW  | Control Passed Frame Number<br>When both ON and OFF numbers are set to 0x00, frame control is in bypass mode<br>Bit[3:0]: Frame ON number  |
| 0x4202  | FRAME CONTROL 01 | 0x00          | RW  | Control Masked Frame Number<br>When both ON and OFF numbers are set to 0x00, frame control is in bypass mode<br>Bit[3:0]: Frame OFF number |

## 6.5 format description

Format control converts the internal data format into the desired output format including YUV, RGB, or RAW.

**table 6-5**      **FORMAT control registers (sheet 1 of 5)**

| address | register name     | default value | R/W | description  |
|---------|-------------------|---------------|-----|--|
| 0x4300  | FORMAT CONTROL 00 | 0xF8          | RW  | Format Control 00<br>Bit[7:4]: Output format of formatter module<br>0x0: RAW<br>Bit[3:0]: Output sequence<br>0x0: BGBG... / GRGR...<br>0x1: GBGB... / RGRG...<br>0x2: GRGR... / BGBG...<br>0x3: RGRG... / GBGB...<br>0x4~0xF: Not allowed<br>0x1: Y8<br>Bit[3:0]: Does not matter<br>0x2: YUV444/RGB888 (not available for full resolution)<br>Bit[3:0]: Output sequence<br>0x0: YUVYUV..., or<br>GBRBR...<br>0x1: YVUYVU..., or<br>GRBGRB...<br>0x2: UYVUYV..., or<br>BGRBGR...<br>0x3: VYUVYU..., or<br>RGBRGR...<br>0x4: UVYUVY..., or<br>BRBGRB...<br>0x5: VUYVUY..., or<br>RBGRBG...<br>0x6~0xE: Not allowed<br>0xF: UYVUYV..., or<br>BGRBGR...<br>0x3: YUV422<br>Bit[3:0]: Output sequence<br>0x0: YUYV...<br>0x1: YVYU...<br>0x2: UYVY...<br>0x3: VYUY...<br>0x4~0xE: Not allowed<br>0xF: UYVY...<br>0x4: YUV420<br>Bit[3:0]: Output sequence<br>0x0: YYYY... / YUYV...<br>0x1: YYYY... / YVYU...<br>0x2: YYYY... / UYVY...<br>0x3: YYYY... / VYUY...<br>0x4: YUYV... / YYYY... |

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

| address | register name | default value | R/W | description                                |
|---------|---------------|---------------|-----|--|
|         |               |               |     | 0x5: YVYU... / YYYY...                     |
|         |               |               |     | 0x6: UYVY... / YYYY...                     |
|         |               |               |     | 0x7: VYUY... / YYYY...                     |
|         |               |               |     | 0x8~0xE: Not allowed                       |
|         |               |               |     | 0xF: YYYY... / UYVY...                     |
|         |               |               |     | 0x5: YUV420 (for MIPI only)                |
|         |               |               |     | Bit[3:0]: Output sequence                  |
|         |               |               |     | 0x0~0xD: Not allowed                       |
|         |               |               |     | 0xE: VYVYVY... / UYUUYU...                 |
|         |               |               |     | 0xF: UYUUYU... / VYVYVY...                 |
|         |               |               |     | 0x6: RGB565                                |
|         |               |               |     | Bit[3:0]: Output sequence                  |
|         |               |               |     | 0x0: {b[4:0],g[5:3]}, {g[2:0],r[4:0]}      |
|         |               |               |     | 0x1: {r[4:0],g[5:3]}, {g[2:0],b[4:0]}      |
|         |               |               |     | 0x2: {g[4:0],r[5:3]}, {r[2:0],b[4:0]}      |
|         |               |               |     | 0x3: {b[4:0],r[5:3]}, {r[2:0],g[4:0]}      |
|         |               |               |     | 0x4: {g[4:0],b[5:3]}, {b[2:0],r[4:0]}      |
|         |               |               |     | 0x5: {r[4:0],b[5:3]}, {b[2:0],g[4:0]}      |
|         |               |               |     | 0x6~0xE: Not allowed                       |
|         |               |               |     | 0xF: {g[2:0],b[4:0]}, {r[4:0],g[5:3]}      |
|         |               |               |     | 0x7: RGB555 format 1                       |
|         |               |               |     | Bit[3:0]: Output sequence                  |
|         |               |               |     | 0x0: {b[4:0],g[4:2]}, {g[1:0],1'b0,r[4:0]} |
|         |               |               |     | 0x1: {r[4:0],g[4:2]}, {g[1:0],1'b0,b[4:0]} |
|         |               |               |     | 0x2: {g[4:0],r[4:2]}, {r[1:0],1'b0,b[4:0]} |
|         |               |               |     | 0x3: {b[4:0],r[4:2]}, {r[1:0],1'b0,g[4:0]} |
|         |               |               |     | 0x4: {r[4:0],b[4:2]}, {b[1:0],1'b0,g[4:0]} |
|         |               |               |     | 0x5: {g[4:0],b[4:2]}, {b[1:0],1'b0,r[4:0]} |
|         |               |               |     | 0x6~0xE: Not allowed                       |
|         |               |               |     | 0xF: {g[1:0],1'b0,b[4:0]}, {r[4:0],g[4:2]} |
|         |               |               |     | 0x8: RGB555 format 2                       |
|         |               |               |     | Bit[3:0]: Output sequence                  |
|         |               |               |     | 0x0: {1'b0,b[4:0],g[4:3]}, {g[2:0],r[4:0]} |

table 6-5 FORMAT control registers (sheet 3 of 5)

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
|         |               |               |     | 0x1: {1'b0,r[4:0],g[4:2],<br>{g[2:0],b[4:0]}         |
|         |               |               |     | 0x2: {1'b0,g[4:0],r[4:2],<br>{r[2:0],b[4:0]}         |
|         |               |               |     | 0x3: {1'b0,b[4:0],r[4:2],<br>{r[2:0],g[4:0]}         |
|         |               |               |     | 0x4: {1'b0,r[4:0],b[4:2],<br>{b[2:0],g[4:0]}         |
|         |               |               |     | 0x5: {1'b0,g[4:0],b[4:2],<br>{b[2:0],r[4:0]}         |
|         |               |               |     | 0x6: {b[4:0],1'b0,g[4:3],<br>{g[2:0],r[4:0]}         |
|         |               |               |     | 0x7: {r[4:0],1'b0,g[4:2],<br>{g[2:0],b[4:0]}         |
|         |               |               |     | 0x8: {g[4:0],1'b0,r[4:2],<br>{r[2:0],b[4:0]}         |
|         |               |               |     | 0x9: {b[4:0],1'b0,r[4:2],<br>{r[2:0],g[4:0]}         |
|         |               |               |     | 0xA: {r[4:0],1'b0,b[4:2],<br>{b[2:0],g[4:0]}         |
|         |               |               |     | 0xB: {g[4:0],1'b0,b[4:2],<br>{b[2:0],r[4:0]}         |
|         |               |               |     | 0xC~0xF: Not allowed                                 |
|         |               |               |     | 0x9: RGB444 format 1                                 |
|         |               |               |     | Bit[3:0]: Output sequence                            |
|         |               |               |     | 0x0: {1'b0,b[3:0],2'h0,g[3],<br>{g[2:0],1'b0,r[3:0]} |
|         |               |               |     | 0x1: {1'b0,r[3:0],2'h0,g[3],<br>{g[2:0],1'b0,b[3:0]} |
|         |               |               |     | 0x2: {1'b0,g[3:0],2'h0,r[3],<br>{r[2:0],1'b0,b[3:0]} |
|         |               |               |     | 0x3: {1'b0,b[3:0],2'h0,r[3],<br>{r[2:0],1'b0,g[3:0]} |
|         |               |               |     | 0x4: {1'b0,r[3:0],2'h0,b[3],<br>{b[2:0],1'b0,g[3:0]} |
|         |               |               |     | 0x5: {1'b0,g[3:0],2'h0,b[3],<br>{b[2:0],1'h0,r[3:0]} |
|         |               |               |     | 0x6: {b[3:0],1'b0,g[3:1],<br>{g[0],2'h0,r[3:0],1'b0} |
|         |               |               |     | 0x7: {r[3:0],1'b0,g[3:1],<br>{g[0],2'h0,b[3:0],1'b0} |
|         |               |               |     | 0x8: {g[3:0],1'b0,r[3:1],<br>{r[0],2'h0,b[3:0],1'b0} |
|         |               |               |     | 0x9: {b[3:0],1'b0,r[3:1],<br>{r[0],2'h0,g[3:0],1'b0} |
|         |               |               |     | 0xA: {r[3:0],1'b0,b[3:1],<br>{b[0],2'h0,g[3:0],1'b0} |
|         |               |               |     | 0xB: {g[3:0],1'b0,b[3:1],<br>{b[0],2'h0,r[3:0],1'b0} |
|         |               |               |     | 0xC~0xE: Not allowed                                 |

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

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
|         |               |               |     | 0xF: {g[0],2'h2,b[3:0],1'b1},<br>{r[3:0],1'b1,g[3:1]}                 |
|         |               |               |     | 0xA: RGB444 format 2  |
|         |               |               |     | Bit[3:0]: Output sequence   |
|         |               |               |     | 0x0: {4'b0,b[3:0]},<br>{g[3:0],r[3:0]}                                |
|         |               |               |     | 0x1: {4'b0,r[3:0]},<br>{g[3:0],b[3:0]}                                |
|         |               |               |     | 0x2: {4'b0,b[3:0]},<br>{r[3:0],g[3:0]}                                |
|         |               |               |     | 0x3: {4'b0,r[3:0]},<br>{b[3:0],g[3:0]}                                |
|         |               |               |     | 0x4: {4'b0,g[3:0]},<br>{b[3:0],r[3:0]}                                |
|         |               |               |     | 0x5: {4'b0,g[3:0]},<br>{r[3:0],b[3:0]}                                |
|         |               |               |     | 0x6: {b[3:0],g[3:0],2'h0},<br>{r[3:0],b[3:0],2'h0,g[3:0],r[3:0],2'h0} |
|         |               |               |     | 0x7: {r[3:0],g[3:0],2'h0},<br>{b[3:0],r[3:0],2'h0,g[3:0],b[3:0],2'h0} |
|         |               |               |     | 0x8: {b[3:0],r[3:0],2'h0},<br>{g[3:0],b[3:0],2'h0,r[3:0],g[3:0],2'h0} |
|         |               |               |     | 0x9: {r[3:0],b[3:0],2'h0},<br>{g[3:0],r[3:0],2'h0,b[3:0],g[3:0],2'h0} |
|         |               |               |     | 0xA: {g[3:0],b[3:0],2'h0},<br>{r[3:0],g[3:0],2'h0,b[3:0],r[3:0],2'h0} |
|         |               |               |     | 0xB: {g[3:0],r[3:0],2'h0},<br>{b[3:0],g[3:0],2'h0,r[3:0],b[3:0],2'h0} |
|         |               |               |     | 0xC~0xF: Not allowed  |
|         |               |               |     | 0xB~0xE: Not allowed  |
|         |               |               |     | 0xF: Bypass formatter module (not recommended)                        |
|         |               |               |     | Bit[3:0]: Output format   |
|         |               |               |     | 0x8: Raw  |
|         |               |               |     | 0x9: YUV422   |
|         |               |               |     | 0xA: YUV444   |
|         |               |               |     | 0xE: VYYVYY.../UYUYUY   |
|         |               |               |     | 0xF: UYYUY.../VYYVYY  |

table 6-5 FORMAT control registers (sheet 5 of 5)

| address | register name     | default value | R/W | description  |
|---------|-------------------|---------------|-----|--|
| 0x4301  | FORMAT CONTROL 01 | 0x00          | RW  | Format Control 01<br>Bit[1:0]: YUV422 UV control<br>00: U/V generated from average<br>01: U/V generated from first pixel<br>10: Not valid<br>11: U/V generated from second pixel |
| 0x4302  | YMAX VALUE        | 0x03          | RW  | Bit[1:0]: Y max clip value[9:8]  |
| 0x4303  | YMAX VALUE        | 0xFF          | RW  | Bit[7:0]: Y max clip value[7:0]  |
| 0x4304  | YMIN VALUE        | 0x00          | RW  | Bit[1:0]: Y min clip value[9:8]  |
| 0x4305  | YMIN VALUE        | 0x00          | RW  | Bit[7:0]: Y min clip value[7:0]  |
| 0x4306  | UMAX VALUE        | 0x03          | RW  | Bit[1:0]: U max clip value[9:8]  |
| 0x4307  | UMAX VALUE        | 0xFF          | RW  | Bit[7:0]: U max clip value[7:0]  |
| 0x4308  | UMIN VALUE        | 0x00          | RW  | Bit[1:0]: U min clip value[9:8]  |
| 0x4309  | UMIN VALUE        | 0x00          | RW  | Bit[7:0]: U min clip value[7:0]  |
| 0x430A  | VMAX VALUE        | 0x03          | RW  | Bit[1:0]: V max clip value[9:8]  |
| 0x430B  | VMAX VALUE        | 0xFF          | RW  | Bit[7:0]: V max clip value[7:0]  |
| 0x430C  | VMIN VALUE        | 0x00          | RW  | Bit[1:0]: V min clip value[9:8]  |
| 0x430D  | VMIN VALUE        | 0x00          | RW  | Bit[7:0]: V min clip value[7:0]  |



## 6.6 digital video port (DVP)

### 6.6.1 overview

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 will be sent out through the OV5640 MIPI interface.

**table 6-6** DVP control registers (sheet 1 of 3)

| address | register name    | default value | R/W | description  |
|---------|------------------|---------------|-----|--|
| 0x4709  | DVP VYSNC WIDTH0 | 0x02          | RW  | VSYNC Width Line Unit  |
| 0x470A  | DVP VYSNC WIDTH1 | 0x00          | RW  | Bit[7:0]: VSYNC width PCLK unit[15:8]  |
| 0x470B  | DVP VYSNC WIDTH2 | 0x01          | RW  | Bit[7:0]: VSYNC width PCLK unit[7:0]   |
| 0x4711  | PAD LEFT CTRL    | 0x00          | RW  | HSYNC Mode Left Padding Pixel Counter<br>Add padding data at start of a line   |
| 0x4712  | PAD RIGHT CTRL   | 0x00          | RW  | HSYNC Mode Right Padding Pixel Counter<br>Add padding data at end of a line  |
| 0x4713  | JPG MODE SELECT  | 0x02          | RW  | Bit[2:0]: JPEG mode select<br>001: JPEG mode 1<br>010: JPEG mode 2<br>011: JPEG mode 3<br>100: JPEG mode 4<br>101: JPEG mode 5<br>110: JPEG mode 6 |
| 0x4715  | 656 DUMMY LINE   | 0x00          | RW  | Bit[3:0]: CCIR656 dummy line number<br>Control dummy line number at beginning of the frame   |
| 0x4719  | CCIR656 CTRL     | 0x00          | RW  | Bit[1:0]: CCIR656 EAV/SAV option   |
| 0x471B  | HSYNC CTRL00     | 0x02          | RW  | Bit[0]: HSYNC mode enable  |

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

| address | register name         | default value | R/W | description  |
|---------|-----------------------|---------------|-----|--|
| 0x471D  | DVP VSYNC CTRL        | 0x01          | RW  | Bit[1:0]: vsync_mode<br>00: VSYNC positive edge trigger by end of field, negative edge trigger by start of frame<br>01: VSYNC positive edge trigger by end of frame, the width define by register<br>10: VSYNC positive edge trigger by start of field, the width define by register   |
| 0x471F  | DVP HREF CTRL         | 0x40          | RW  | HREF Minimum Blanking in JPEG Mode23   |
| 0x4721  | VERTICAL START OFFSET | 0x00          | RW  | Bit[3:0]: Vertical start delay between video output and video input  |
| 0x4722  | VERTICAL END OFFSET   | 0x00          | RW  | Bit[3:0]: Vertical end delay between video output and video input  |
| 0x4723  | DVP CTRL23            | 0x00          | RW  | DVP JPEG Mode456 Skip Line Number  |
| 0x4730  | CCIR656 CTRL00        | 0x00          | RW  | Bit[7]: SYNC code selection<br>0: Automatically generate SYNC code<br>1: SYNC code from register setting 0x4732~4735<br>Bit[6]: f value in CCIR656 SYNC code when fixed f value<br>Bit[5]: Fixed f value<br>Bit[4:3]: Blank toggle data options<br>00: Toggle data is 1'h040/1'h200<br>01: Use register setting 0x4736~0x4738<br>10: Blanking data always keep 0<br>Bit[1]: Clip data disable<br>Bit[0]: CCIR656 mode enable |
| 0x4731  | CCIR656 CTRL01        | 0x00          | RW  | Bit[0]: Blanking toggle data order option  |
| 0x4732  | CCIR656 FS            | 0x01          | RW  | CCIR656 Sync Code Frame Start  |
| 0x4733  | CCIR656 FE            | 0x0F          | RW  | CCIR656 Sync Code Frame End  |
| 0x4734  | CCIR656 LS            | 0x00          | RW  | CCIR656 Sync Code Line Start   |

table 6-6 DVP control registers (sheet 3 of 3)

| address | register name   | default value | R/W | description  |
|---------|-----------------|---------------|-----|--|
| 0x4735  | CCIR656 LE      | 0x00          | RW  | CCIR656 Sync Code line End   |
| 0x4736  | CCIR656 CTRL6   | 0x00          | RW  | Bit[3:2]: Toggle data0[9:8]<br>Bit[1:0]: Toggle data1[9:8]   |
| 0x4737  | CCIR656 CTRL7   | 0x00          | RW  | Bit[7:0]: Toggle data0[7:0]  |
| 0x4738  | CCIR656 CTRL8   | 0x00          | RW  | Bit[7:0]: Toggle data1[7:0]  |
| 0x4740  | POLARITY CTRL00 | 0x20          | RW  | Bit[5]: PCLK polarity<br>0: Active low<br>1: Active high<br>Bit[3]: Gate PCLK under VSYNC<br>Bit[2]: Gate PCLK under HREF<br>Bit[1]: HREF polarity<br>0: Active low<br>1: Active high<br>Bit[0]: VSYNC polarity<br>0: Active low<br>1: Active high |
| 0x4741  | TEST PATTERN    | 0x00          | RW  | Bit[2]: Test pattern enable<br>Bit[1]: Test pattern select<br>0: Output test pattern 0<br>1: Output test pattern 1<br>Bit[0]: Test pattern 8-bit/10-bit<br>0: 10-bit test pattern<br>1: 8-bit test pattern   |
| 0x4745  | DATA ORDER      | 0x00          | RW  | Bit[2:1]: DVP order option for debug<br>00: D[9:0]<br>10: {D[7:0], D[9:8]}<br>x1: {D[1:0], D[9:2]}<br>Bit[0]: Output data order<br>0: Normal output<br>1: Reverse output data bit order  |

6.6.2 DVP timing

figure 6-7 DVP timing diagram

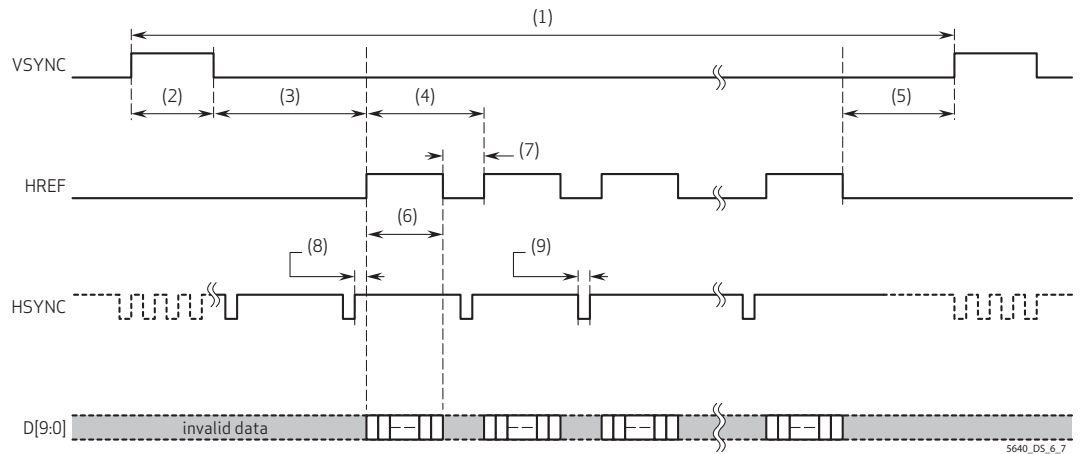


table 6-7 DVP timing specifications (sheet 1 of 2)

| mode                     | timing  |
|--------------------------|---|
| 5 megapixel<br>2592x1944 | (1) 5596992 tp<br>(2) 5688 tp<br>(3) 48276 tp<br>(4) 2844 tp<br>(5) 14544 tp<br>(6) 2592 tp<br>(7) 252 tp<br>(8) 0 tp<br>(9) 252 tp     |
| 1080p<br>1920x1080       | (1) 5596992 tp<br>(2) 5688 tp<br>(3) 2506164 tp<br>(4) 2844 tp<br>(5) 14544 tp<br>(6) 1920 tp<br>(7) 924 tp<br>(8) 0 tp<br>(9) 924 tp   |
| UXGA<br>1600x1200        | (1) 5596992 tp<br>(2) 5688 tp<br>(3) 2165204 tp<br>(4) 2844 tp<br>(5) 14544 tp<br>(6) 1600 tp<br>(7) 1244 tp<br>(8) 0 tp<br>(9) 1244 tp |



note

The timing values shown in table 6-7 may vary depending upon register settings.

table 6-7 DVP timing specifications (sheet 2 of 2)

| mode             | timing  |
|------------------|---|
| 720p<br>1280x720 | (1) 5596992 tp<br>(2) 5688 tp<br>(3) 3530644 tp<br>(4) 2844 tp<br>(5) 14544 tp<br>(6) 1280 tp<br>(7) 1564 tp<br>(8) 0 tp<br>(9) 1564 tp |
| XGA<br>1024x768  | (1) 5596992 tp<br>(2) 5688tp<br>(3) 2825588 tp<br>(4) 2844 tp<br>(5) 583344tp<br>(6) 1024 tp<br>(7) 1820 tp<br>(8) 0 tp<br>(9) 1820 tp  |
| VGA<br>640x480   | (1) 5596992 tp<br>(2) 5688 tp<br>(3) 4213844 tp<br>(4) 2844 tp<br>(5) 14544 tp<br>(6) 640 tp<br>(7) 2204 tp<br>(8) 0 tp<br>(9) 2204 tp  |

## 6.7 mobile industry processor interface (MIPI)

MIPI provides a single uni-directional clock lane and two bi-directional data lane solution for communication links between components inside a mobile device. The two data lanes have full support for HS (uni-direction) and LP (bi-direction) data transfer mode. Contact your local OmniVision FAE for more details.

**table 6-8** MIPI transmitter registers (sheet 1 of 3)

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x4800  | MIPI CTRL 00  | 0x04          | RW  | MIPI Control 00<br>Bit[5]: Clock lane gate enable<br>0: Clock lane is free running<br>1: Gate clock lane when no packet to transmit<br>Bit[4]: Line sync enable<br>0: Do not send line short packet for each line<br>1: Send line short packet for each line<br>Bit[3]: Lane select<br>0: Use lane1 as default data lane<br>1: Use lane2 as default data lane<br>Bit[2]: Idle status<br>0: MIPI bus will be LP00 when no packet to transmit<br>1: MIPI bus will be LP11 when no packet to transmit |
| 0x4801  | MIPI CTRL 01  | 0x04          | RW  | MIPI Control 01<br>Bit[4]: PH bit order for ECC<br>0: {DI[7:0],WC[7:0],WC[15:8]}<br>1: {DI[0:7],WC[0:7],WC[8:15]}<br>Bit[3]: PH byte order for ECC<br>0: {DI,WC_l,WC_h}<br>1: {DI,WC_h,WC_l}<br>Bit[2]: PH byte order2 for ECC<br>0: {DI,WC}<br>1: {WC,DI}   |
| 0x4805  | MIPI CTRL 05  | 0x10          | RW  | Bit[7]: MIPI lane1 disable<br>1: Disable MIPI data lane1<br>Lane1 will be LP00<br>Bit[6]: MIPI lane1 disable<br>1: Disable MIPI data lane1<br>Lane1 will be LP00<br>Bit[5]: LPX Global timing select<br>0: Auto calculate t_lpx_o in pclk2x domain,<br>unit clk2x<br>1: Use lp_p_min[7:0]  |

table 6-8 MIPI transmitter registers (sheet 2 of 3)

| address | register name          | default value | R/W | description  |
|---------|------------------------|---------------|-----|--|
| 0x480A  | MIPI DATA ORDER        | 0x00          | RW  | Bit[2]: Bit order reverse<br>Bit[1:0]: Bit position adjust<br>01: {data[7:0],data[9:8]}<br>10: {data[1:0],data[9:2]}   |
| 0x4818  | MIN HS ZERO H          | 0x00          | RW  | High Byte of Minimum Value of hs_zero<br>Unit ns   |
| 0x4819  | MIN HS ZERO L          | 0x96          | RW  | Low Byte of Minimum Value of hs_zero<br>$hs\_zero\_real = hs\_zero\_min\_o + Tui*ui\_hs\_zero\_min\_o$                 |
| 0x481A  | MIN MIPI HS TRAIL H    | 0x00          | RW  | High Byte of Minimum Value of hs_trail<br>Unit ns  |
| 0x481B  | MIN MIPI HS TRAIL L    | 0x3C          | RW  | Low Byte of Minimum Value of hs_trail<br>$hs\_trail\_real = hs\_trail\_min\_o + Tui*ui\_hs\_trail\_min\_o$             |
| 0x481C  | MIN MIPI CLK ZERO H    | 0x01          | RW  | High Byte of Minimum Value of clk_zero   |
| 0x481D  | MIN MIPI CLK ZERO L    | 0x86          | RW  | Low Byte of Minimum Value of clk_zero<br>$clk\_zero\_real = clk\_zero\_min\_o + Tui*ui\_clk\_zero\_min\_o$             |
| 0x481E  | MIN MIPI CLK PREPARE H | 0x00          | RW  | High Byte of Minimum Value of clk_prepare<br>Unit ns   |
| 0x481F  | MIN MIPI CLK PREPARE L | 0x3C          | RW  | Low Byte of Minimum Value of clk_prepare<br>$clk\_prepare\_real = clk\_prepare\_min\_o + Tui*ui\_clk\_prepare\_min\_o$ |
| 0x4820  | MIN CLK POST H         | 0x00          | RW  | High Byte of Minimum Value of clk_post<br>Unit ns  |
| 0x4821  | MIN CLK POST L         | 0x56          | RW  | Low Byte of Minimum Value of clk_post<br>$clk\_post\_real = clk\_post\_min\_o + Tui*ui\_clk\_post\_min\_o$             |
| 0x4822  | MIN CLK TRAIL H        | 0x00          | RW  | High Byte of Minimum Value of clk_trail<br>Unit ns   |
| 0x4823  | MIN CLK TRAIL L        | 0x3C          | RW  | Low Byte of Minimum Value of clk_trail<br>$clk\_trail\_real = clk\_trail\_min\_o + Tui*ui\_clk\_trail\_min\_o$         |
| 0x4824  | MIN LPX PCLK H         | 0x00          | RW  | High Byte of Minimum Value of lpx_p<br>Unit ns   |
| 0x4825  | MIN LPX PCLK L         | 0x32          | RW  | Low Byte of Minimum Value of lpx_p<br>$lpx\_p\_real = lpx\_p\_min\_o + Tui*ui\_lpx\_p\_min\_o$                         |
| 0x4826  | MIN HS PREPARE H       | 0x00          | RW  | High Byte of Minimum Value of hs_prepare<br>Unit ns  |
| 0x4827  | MIN HS PREPARE L       | 0x32          | RW  | Low Byte of Minimum Value of hs_prepare<br>$hs\_prepare\_real = hs\_prepare\_min\_o + Tui*ui\_hs\_prepare\_min\_o$     |

table 6-8 MIPI transmitter registers (sheet 3 of 3)

| address | register name      | default value | R/W | description  |
|---------|--------------------|---------------|-----|--|
| 0x4828  | MIN HS EXIT H      | 0x00          | RW  | High Byte of Minimum Value of hs_exit<br>Unit ns   |
| 0x4829  | MIN HS EXIT L      | 0x64          | RW  | Low Byte of Minimum Value of hs_exit<br>$hs\_exit\_real = hs\_exit\_min\_o + Tui * ui\_hs\_exit\_min\_o$ |
| 0x482A  | MIN HS ZERO/UI     | 0x05          | RW  | Minimum UI Value of hs_zero<br>Unit UI   |
| 0x482B  | MIN HS TRAIL/UI    | 0x04          | RW  | Minimum UI Value of hs_trail<br>Unit UI  |
| 0x482C  | MIN CLK ZERO/UI    | 0x00          | RW  | Minimum UI Value of clk_zero<br>Unit UI  |
| 0x482D  | MIN CLK PREPARE/UI | 0x00          | RW  | Minimum UI Value of clk_prepare<br>Unit UI   |
| 0x482E  | MIN CLK POST/UI    | 0x34          | RW  | Minimum UI Value of clk_post<br>Unit UI  |
| 0x482F  | MIN CLK TRAIL/UI   | 0x00          | RW  | Minimum UI Value of clk_trail<br>Unit UI   |
| 0x4830  | MIN LPX PCLK/UI    | 0x00          | RW  | Minimum UI Value of lpx_p(pclk2x domain)<br>Unit UI  |
| 0x4831  | MIN HS PREPARE/UI  | 0x04          | RW  | Minimum UI Value of hs_prepare<br>Unit UI  |
| 0x4832  | MIN HS EXIT/UI     | 0x00          | RW  | Minimum UI Value of hs_exit<br>Unit UI   |
| 0x4837  | PCLK PERIOD        | 0x10          | RW  | The period of pixel clock, pclk_div=1, and 1 bit decimal   |



## 7 register tables

The following tables provide descriptions of the device control registers contained in the OV5640. For all registers enable/disable bits, ENABLE = 1 and DISABLE = 0.

**table 7-1** system and IO pad control registers (sheet 1 of 7)

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x3000  | SYSTEM RESET00 | 0x30          | RW  | Reset for Individual Block<br>(0: enable block; 1: reset block)<br>Bit[7]: Reset BIST<br>Bit[6]: Reset MCU program memory<br>Bit[5]: Reset MCU<br>Bit[4]: Reset OTP<br>Bit[3]: Reset STB<br>Bit[2]: Reset d5060<br>Bit[1]: Reset timing control<br>Bit[0]: Reset array control |
| 0x3001  | SYSTEM RESET01 | 0x08          | RW  | Reset for Individual Block<br>(0: enable block; 1: reset block)<br>Bit[7]: Reset AWB registers<br>Bit[6]: Reset AFC<br>Bit[5]: Reset ISP<br>Bit[4]: Reset FC<br>Bit[3]: Reset S2P<br>Bit[2]: Reset BLC<br>Bit[1]: Reset AEC registers<br>Bit[0]: Reset AEC                     |
| 0x3002  | SYSTEM RESET02 | 0x1C          | RW  | Reset for Individual Block<br>(0: enable block; 1: reset block)<br>Bit[7]: Reset VFIFO<br>Bit[6]: Debug mode<br>Bit[5]: Reset format<br>Bit[4]: Reset JFIFO<br>Bit[3]: Reset SFIFO<br>Bit[2]: Reset JPG<br>Bit[1]: Reset format MUX<br>Bit[0]: Reset average                   |
| 0x3003  | SYSTEM RESET03 | 0x00          | RW  | Reset for Individual Block<br>(0: enable block; 1: reset block)<br>Bit[7:6]: Debug mode<br>Bit[5]: Reset digital gain compensation<br>Bit[4]: Reset SYNC FIFO<br>Bit[3]: Reset PSRAM<br>Bit[2]: Reset ISP FC<br>Bit[1]: Reset MIPI<br>Bit[0]: Reset DVP                        |

table 7-1 system and IO pad control registers (sheet 2 of 7)

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x3004  | CLOCK ENABLE00 | 0xCF          | RW  | Clock Enable Control<br>(0: disable clock; 1: enable clock)<br>Bit[7]: Enable BIST clock<br>Bit[6]: Enable MCU program memory clock<br>Bit[5]: Enable MCU clock<br>Bit[4]: Enable OTP clock<br>Bit[3]: Enable STROBE clock<br>Bit[2]: Enable D5060 clock<br>Bit[1]: Enable timing control clock<br>Bit[0]: Enable array control clock                    |
| 0x3005  | CLOCK ENABLE01 | 0xF7          | RW  | Clock Enable Control<br>(0: disable clock; 1: enable clock)<br>Bit[7]: Enable AWB register clock<br>Bit[6]: Enable AFC clock<br>Bit[5]: Enable ISP clock<br>Bit[4]: Enable FC clock<br>Bit[3]: Enable S2P clock<br>Bit[2]: Enable BLC clock<br>Bit[1]: Enable AEC register clock<br>Bit[0]: Enable AEC clock   |
| 0x3006  | CLOCK ENABLE02 | 0xE3          | RW  | Clock Enable Control<br>(0: disable clock; 1: enable clock)<br>Bit[7]: Enable PSRAM clock<br>Bit[6]: Enable FMT clock<br>Bit[5]: Enable JPEG 2x clock<br>Bit[3]: Enable JPEG clock<br>Bit[2]: Debug mode<br>Bit[1]: Enable format MUX clock<br>Bit[0]: Enable average clock  |
| 0x3007  | CLOCK ENABLE03 | 0xFF          | RW  | Clock Enable Control<br>(0: disable clock; 1: enable clock)<br>Bit[7]: Enable digital gain compensation clock<br>Bit[6]: Enable SYNC FIFO clock<br>Bit[5]: Enable ISPFC SCLK clock<br>Bit[4]: Enable MIPI PCLK clock<br>Bit[3]: Enable MIPI clock<br>Bit[2]: Enable DVP PCLK clock<br>Bit[1]: Enable VFIFO PCLK clock<br>Bit[0]: Enable VFIFO SCLK clock |
| 0x3008  | SYSTEM CTROL0  | 0x02          | RW  | System Control<br>Bit[7]: Software reset<br>Bit[6]: Software power down<br>Bit[5:0]: Debug mode  |
| 0x3009  | DEBUG MODE     | –             | –   | Debug Mode   |

table 7-1 system and IO pad control registers (sheet 3 of 7)

| address           | register name        | default value | R/W | description  |
|-------------------|----------------------|---------------|-----|--|
| 0x300A            | CHIP ID HIGH BYTE    | 0x56          | R   | Chip ID High Byte  |
| 0x300B            | CHIP ID LOW BYTE     | 0x40          | R   | Chip ID Low Byte   |
| 0x300C~<br>0x300D | DEBUG MODE           | –             | –   | Debug Mode   |
| 0x300E            | MIPI CONTROL 00      | 0x58          | RW  | MIPI Control 00<br>Bit[7:5]: mipi_lane_mode<br>000: One lane mode<br>001: Two lane mode<br>Others: Debug mode<br>Bit[4]: MIPI TX PHY power down<br>0: Debug mode<br>1: Power down PHY HS TX<br>Bit[3]: MIPI RX PHY power down<br>0: Debug mode<br>1: Power down PHY LP RX module<br>Bit[2]: mipi_en<br>0: DVP enable<br>1: MIPI enable<br>Bit[1:0]: Debug mode |
| 0x300F~<br>0x3015 | DEBUG MODE           | –             | –   | Debug Mode   |
| 0x3016            | PAD OUTPUT ENABLE 00 | 0x00          | RW  | Input/Output Control (0: input; 1: output)<br>Bit[7:2]: Debug mode<br>Bit[1]: Strobe output enable<br>Bit[0]: SIOD output enable   |
| 0x3017            | PAD OUTPUT ENABLE 01 | 0x00          | RW  | Input/Output Control (0: input; 1: output)<br>Bit[7]: FREX output enable<br>Bit[6]: VSYNC output enable<br>Bit[5]: HREF output enable<br>Bit[4]: PCLK output enable<br>Bit[3:0]: D[9:6] output enable  |
| 0x3018            | PAD OUTPUT ENABLE 02 | 0x00          | RW  | Input/Output Control (0: input; 1: output)<br>Bit[7:2]: D[5:0] output enable<br>Bit[1]: GPIO1 output enable<br>Bit[0]: GPIO0 output enable   |

table 7-1 system and IO pad control registers (sheet 4 of 7)

| address | register name       | default value | R/W | description  |
|---------|---------------------|---------------|-----|--|
| 0x3019  | PAD OUTPUT VALUE 00 | 0x00          | RW  | PAD Output Value<br>Bit[7]: MIPI data lane option<br>Bypass latch in MIPI PHY<br>Bit[6]: MIPI lane2 state in sleep mode<br>0: LP00<br>1: LP11<br>Bit[5]: MIPI lane1 state in sleep mode<br>0: LP00<br>1: LP11<br>Bit[4]: MIPI CLK lane state in sleep mode<br>0: LP00<br>1: LP11<br>Bit[3:2]: Debug mode<br>Bit[1]: Strobe<br>Bit[0]: SIOD |
| 0x301A  | PAD OUTPUT VALUE 01 | 0x00          | RW  | GPIO Output Value 01<br>Bit[7]: FREX<br>Bit[6]: VSYNC<br>Bit[5]: HREF<br>Bit[4]: PCLK<br>Bit[3:0]: D[9:6]  |
| 0x301B  | PAD OUTPUT VALUE 02 | 0x00          | RW  | GPIO Output Value 02<br>Bit[7:2]: D[5:0]<br>Bit[1]: GPIO1<br>Bit[0]: GPIO0   |
| 0x301C  | PAD SELECT 00       | 0x00          | RW  | Output Selection for GPIO<br>Bit[7:2]: Debug mode<br>Bit[1]: IO strobe select<br>Bit[0]: IO SIOD select  |
| 0x301D  | PAD SELECT 01       | 0x00          | RW  | Output Selection for GPIO<br>Bit[7]: FREX select<br>Bit[6]: VSYNC select<br>Bit[5]: HREF select<br>Bit[4]: PCLK select<br>Bit[3:0]: IO D[9:6] select   |
| 0x301E  | PAD SELECT 02       | 0x00          | RW  | Output Selection for GPIO<br>Bit[7:2]: IO D[5:0] select<br>Bit[1]: IO GPIO1 select<br>Bit[0]: IO GPIO0 select  |
| 0x301F  | DEBUG MODE          | –             | –   | Debug Mode   |
| 0x302A  | CHIP REVISION       | 0xB0          | R   | Bit[7:4]: Process<br>0xA: FSI<br>0xB: BSI<br>Bit[3:0]: Chip revision   |

table 7-1 system and IO pad control registers (sheet 5 of 7)

| address           | register name  | default value | R/W | description   |
|-------------------|----------------|---------------|-----|---|
| 0x302B            | DEBUG MODE     | –             | –   | Debug Mode  |
| 0x302C            | PAD CONTROL 00 | 0x02          | RW  | Pad Control<br>Bit[7:6]: Output drive capability<br>00: 1x<br>01: 2x<br>10: 3x<br>11: 4x<br>Bit[5:2]: Debug mode<br>Changing this value is not allowed<br>Bit[1]: FREX enable<br>Bit[0]: Debug mode<br>Changing this value is not allowed |
| 0x302D~<br>0x302F | SYSTEM CONTROL | –             | RW  | System Control Registers<br>Changing these values is not recommended  |
| 0x3030            | SC A PWC PK O  | 0x0B          | RW  | PWC Control<br>Bit[7:6]: Debug mode<br>Changing this value is not allowed<br>Bit[5]: bp_regulator<br>Bit[4:0]: Debug mode<br>Changing this value is not allowed   |
| 0x3031~<br>0x3033 | SYSTEM CONTROL | –             | RW  | System Control Registers<br>Changing these values is not recommended  |
| 0x3034            | SC PLL CONTRL0 | 0x1A          | RW  | Bit[7]: Debug mode<br>Bit[6:4]: PLL charge pump control<br>Bit[3:0]: MIPI bit mode<br>0x8: 8-bit mode<br>0xA: 10-bit mode<br>Others: Debug mode   |
| 0x3035            | SC PLL CONTRL1 | 0x11          | RW  | Bit[7:4]: System clock divider<br>Slow down all clocks<br>Bit[3:0]: Scale divider for MIPI<br>MIPI PCLK/SERCLK can be slow down   |
| 0x3036            | SC PLL CONTRL2 | 0x69          | RW  | Bit[7:0]: PLL multiplier (4~252)<br>Can be any integer from 4~127<br>and only even integers from 128~252  |

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

| address           | register name   | default value | R/W | description   |
|-------------------|-----------------|---------------|-----|---|
| 0x3037            | SC PLL CONTRL3  | 0x03          | RW  | Bit[7:5]: Debug mode<br>Bit[4]: PLL root divider<br>0: Bypass<br>1: Divided by 2<br>Bit[3:0]: PLL pre-divider<br>1,2,3,4,6,8  |
| 0x3038            | SYSTEM CONTROL  | –             | RW  | System Control Registers<br>Changing these values is not recommended  |
| 0x3039            | SC PLL CONTRL 5 | 0x00          | RW  | Bit[7]: PLL bypass<br>Bit[6:0]: Debug mode  |
| 0x303A            | SC PLLS CTRL0   | 0x00          | RW  | Bit[7]: PLLS bypass<br>Bit[6:0]: Debug mode   |
| 0x303B            | SC PLLS CTRL1   | 0x19          | RW  | Bit[7:5]: Debug mode<br>Bit[4:0]: PLLS multiplier   |
| 0x303C            | SC PLLS CTRL2   | 0x11          | RW  | Bit[7]: Debug mode<br>Bit[6:4]: PLLS charge pump control<br>Bit[3:0]: PLLS system divider   |
| 0x303D            | SC PLLS CTRL3   | 0x30          | RW  | Bit[7:6]: Debug mode<br>Bit[5:4]: PLLS pre-divider<br>00: /1<br>01: /1.5<br>10: /2<br>11: /3<br>Bit[3]: Debug mode<br>Bit[2]: PLLS root divider<br>0: /1<br>1: /2<br>Bit[1:0]: PLLS seld5<br>00: /1<br>01: /1<br>10: /2<br>11: /2.5 |
| 0x303E~<br>0x304F | SYSTEM CONTROL  | –             | RW  | System Control Registers<br>Changing these values is not recommended  |
| 0x3050            | IO PAD VALUE    | –             | R   | Read Pad Value<br>Bit[7:5]: Debug mode<br>Bit[4]: FREX<br>Bit[3]: PWDN<br>Bit[2]: Debug mode<br>Bit[1]: SIOC<br>Bit[0]: Debug mode  |

table 7-1 system and IO pad control registers (sheet 7 of 7)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x3051  | IO PAD VALUE  | –             | R   | Read Pad Value<br>Bit[7]: OTP memory out<br>Bit[6]: VSYNC<br>Bit[5]: HREF<br>Bit[4]: PCLK<br>Bit[3:0]: D[9:6] |
| 0x3052  | IO PAD VALUE  | –             | R   | Pad Input Status<br>Bit[7:2]: D[5:0]<br>Bit[1]: GPIO1<br>Bit[0]: GPIO0  |

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

| address           | register name     | default value | R/W | description   |
|-------------------|-------------------|---------------|-----|---|
| 0x3100            | SCCB_ID           | 0x78          | RW  | SCCB Slave ID   |
| 0x3101            | SCCB SYSTEM CTRL0 | –             | RW  | SCCB Control Registers<br>Changing these values is not allowed  |
| 0x3102            | SCCB SYSTEM CTRL0 | 0x00          | RW  | Bit[7]: Debug mode<br>Bit[6]: MIPI SC reset<br>Bit[5]: SRB reset<br>Bit[4]: SCCB slave reset<br>Bit[3]: rst_pon_sccb<br>Bit[2]: Debug mode<br>Bit[1]: MIPI PHY reset<br>Bit[0]: PLL reset                           |
| 0x3103            | SCCB SYSTEM CTRL1 | 0x00          | RW  | PLL Clock Select<br>Bit[7:2]: Debug mode<br>Changing this value is not allowed<br>Bit[1]: Select system input clock<br>0: From pad clock<br>1: From PLL<br>Bit[0]: Debug mode<br>Changing this value is not allowed |
| 0x3104–<br>0x3107 | SCCB SYSTEM CTRL0 | –             | RW  | SCCB Control Registers<br>Changing these values is not allowed  |

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

| address | register name       | default value | R/W | description   |
|---------|---------------------|---------------|-----|---|
| 0x3108  | SYSTEM ROOT DIVIDER | 0x16          | RW  | Pad Clock Divider for SCCB Clock<br>Bit[7:6]: Debug mode<br>Bit[5:4]: PCLK root divider<br>00: PCLK = pll_clki<br>01: PCLK = pll_clki/2<br>10: PCLK = pll_clki/4<br>11: PCLK = pll_clki/8<br>Bit[3:2]: sclk2x root divider<br>00: SCLK2x = pll_clki<br>01: SCLK2x = pll_clki/2<br>10: SCLK2x = pll_clki/4<br>11: SCLK2x = pll_clki/8<br>Bit[1:0]: SCLK root divider<br>00: SCLK = pll_clki<br>01: SCLK = pll_clki/2<br>10: SCLK = pll_clki/4<br>11: sclk = pll_clki/8 |

Confidential for  
xin jin hui



table 7-3 SRB control registers

| address           | register name                    | default value | R/W | description  |
|-------------------|----------------------------------|---------------|-----|--|
| 0x3200            | GROUP ADDR0                      | 0x40          | RW  | SRAM Group Address0  |
| 0x3201            | GROUP ADDR1                      | 0x4A          | RW  | SRAM Group Address1  |
| 0x3202            | GROUP ADDR2                      | 0x54          | RW  | SRAM Group Address2  |
| 0x3203            | GROUP ADDR3                      | 0x5E          | RW  | SRAM Group Address3  |
| 0x3204~<br>0x3211 | GROUP WRITE<br>CONTROL REGISTERS | –             | RW  | Group Write Registers<br>Changing these values is not recommended  |
| 0x3212            | SRM GROUP ACCESS                 | –             | W   | SRM Group Access<br>Bit[7]: Group launch enable<br>Bit[6]: Test mode access group<br>Bit[5]: Group launch<br>Bit[4]: Group hold end<br>Bit[3:0]: Group ID<br>00x: Group for register access<br>011: Group to hold register address<br>of embedded line SOF<br>100: Group to hold register address<br>of embedded line EOF<br>101: Test mode for store register<br>value to memory<br>110: Test mode for restore register<br>value from memory<br>111: Group for write mask address |
| 0x3213            | SRM GROUP STATUS                 | –             | R   | SRM Group Status<br>Bit[7]: Store<br>Bit[6]: Restore<br>Bit[5]: Group hold<br>Bit[4]: Group launch<br>Bit[3]: Group write<br>Bit[2:0]: Group select  |

table 7-4 AWB gain control registers

| address | register name      | default value | R/W | description  |
|---------|--------------------|---------------|-----|--|
| 0x3400  | AWB R GAIN         | 0x04          | RW  | Bit[3:0]: AWB R gain[11:8]   |
| 0x3401  | AWB R GAIN         | 0x00          | RW  | Bit[7:0]: AWB R gain[7:0]  |
| 0x3402  | AWB G GAIN         | 0x04          | RW  | Bit[3:0]: AWB G gain[11:8]   |
| 0x3403  | AWB G GAIN         | 0x00          | RW  | Bit[7:0]: AWB G gain[7:0]  |
| 0x3404  | AWB B GAIN         | 0x04          | RW  | Bit[3:0]: AWB B gain[11:8]   |
| 0x3405  | AWB B GAIN         | 0x00          | RW  | Bit[7:0]: AWB B gain[7:0]  |
| 0x3406  | AWB MANUAL CONTROL | 0x00          | RW  | Bit[7:1]: Debug mode<br>Bit[0]: AWB gain manual enable<br>0: Auto<br>1: Manual |

table 7-5 AEC/AGC control functions

| address | register name    | default value | R/W | description   |
|---------|------------------|---------------|-----|---|
| 0x3500  | AEC PK EXPOSURE  | 0x00          | RW  | Exposure Output<br>Bit[7:4]: Debug mode<br>Bit[3:0]: Exposure [19:16]   |
| 0x3501  | AEC PK EXPOSURE  | 0x02          | RW  | Exposure Output<br>Bit[7:0]: Exposure [15:8]  |
| 0x3502  | AEC PK EXPOSURE  | 0x00          | RW  | Exposure Output<br>Bit[7:0]: Exposure [7:0]   |
| 0x3503  | AEC PK MANUAL    | 0x00          | RW  | AEC Manual Mode Control<br>Bit[7:2]: Debug mode<br>Bit[1]: AGC manual<br>0: Auto enable<br>1: Manual enable<br>Bit[0]: AEC manual<br>0: Auto enable<br>1: Manual enable |
| 0x350A  | AEC PK REAL GAIN | 0x00          | RW  | Real Gain<br>Bit[7:2]: Debug mode<br>Bit[1:0]: Real gain[9:8]   |
| 0x350B  | AEC PK REAL GAIN | 0x10          | RW  | Real Gain<br>Bit[7:0]: Real gain[7:0]   |
| 0x350C  | AEC PK VTS       | 0x00          | RW  | AEC VTS Output<br>Bit[7:0]: VTS[15:8] high bits   |
| 0x350D  | AEC PK VTS       | 0x00          | RW  | AEC VTS Output<br>Bit[7:0]: VTS[7:0] low bits   |

table 7-6 VCM control registers

| address           | register name | default value | R/W | description  |
|-------------------|---------------|---------------|-----|--|
| 0x3600~<br>0x3601 | DEBUG MODE    | –             | –   | Debug Mode   |
| 0x3602            | VCM CONTROL 0 | 0x50          | RW  | VCM Control<br>Bit[7:4]: VCM target value[3:0]<br>Bit[3]: Slew rate control s3<br>Bit[2:0]: Slew rate control s2,s1,s0 |
| 0x3603            | VCM CONTROL 1 | 0x01          | RW  | VCM Control<br>Bit[7]: PWDN VCM<br>Bit[6]: Debug mode<br>Bit[5:0]: VCM target value[9:4]                               |
| 0x3604            | VCM CONTROL 2 | 0x05          | RW  | VCM Control<br>Bit[7:0]: vcm_rdiv[7:0]   |
| 0x3605            | VCM CONTROL 3 | 0x46          | RW  | VCM Control<br>Bit[7:5]: Debug mode<br>Bit[4]: VCM RIH<br>Bit[3:0]: vcm_rdiv[11:8]                                     |
| 0x3606            | VCM CONTROL 4 | 0x00          | RW  | VCM Control<br>Bit[7:3]: Debug mode<br>Bit[2:0]: ib_vcm  |

table 7-7 timing control registers (sheet 1 of 2)

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x3800  | TIMING HS      | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: X address start[11:8] high byte                             |
| 0x3801  | TIMING HS      | 0x00          | RW  | Bit[7:0]: X address start[7:0] low byte   |
| 0x3802  | TIMING VS      | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: Y address start[11:8] high byte                             |
| 0x3803  | TIMING VS      | 0x00          | RW  | Bit[7:0]: Y address start[7:0] low byte   |
| 0x3804  | TIMING HW      | 0x0A          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: X address end[11:8] high byte                               |
| 0x3805  | TIMING HW      | 0x3F          | RW  | Bit[7:0]: X address end[7:0] low byte   |
| 0x3806  | TIMING VH      | 0x07          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: Y address end[10:8] high byte                               |
| 0x3807  | TIMING VH      | 0x9F          | RW  | Bit[7:0]: Y address end[7:0] low byte   |
| 0x3808  | TIMING DVPHO   | 0x0A          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: DVP output horizontal width[11:8] high byte                 |
| 0x3809  | TIMING DVPHO   | 0x20          | RW  | Bit[7:0]: DVP output horizontal width[7:0] low byte   |
| 0x380A  | TIMING DVPVO   | 0x07          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: DVP output vertical height[10:8] high byte                  |
| 0x380B  | TIMING DVPVO   | 0x98          | RW  | Bit[7:0]: DVP output vertical height[7:0] low byte  |
| 0x380C  | TIMING HTS     | 0x0B          | RW  | Bit[7:5]: Debug mode<br>Bit[4:0]: Total horizontal size[12:8] high byte                       |
| 0x380D  | TIMING HTS     | 0x1C          | RW  | Bit[7:0]: Total horizontal size[7:0] low byte   |
| 0x380E  | TIMING VTS     | 0x07          | RW  | Bit[7:0]: Total vertical size[15:8] high byte   |
| 0x380F  | TIMING VTS     | 0xB0          | RW  | Bit[7:0]: Total vertical size[7:0] low byte   |
| 0x3810  | TIMING HOFFSET | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: ISP horizontal offset[11:8] high byte                       |
| 0x3811  | TIMING_HOFFSET | 0x10          | RW  | Bit[7:0]: Horizontal offset[7:0] low byte   |
| 0x3812  | TIMING VOFFSET | 0x00          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: Vertical offset[10:8] high byte                             |
| 0x3813  | TIMING VOFFSET | 0x04          | RW  | Bit[7:0]: Vertical offset[7:0] low byte   |
| 0x3814  | TIMING X INC   | 0x11          | RW  | Bit[7:4]: Horizontal odd subsample increment<br>Bit[3:0]: Horizontal even subsample increment |

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

| address | register name   | default value | R/W | description   |
|---------|-----------------|---------------|-----|---|
| 0x3815  | TIMING Y INC    | 0x11          | RW  | Bit[7:4]: Vertical odd subsample increment<br>Bit[3:0]: Vertical even subsample increment   |
| 0x3816  | HSYNC START     | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: HSYNC start point[11:8]   |
| 0x3817  | HSYNC START     | 0x00          | RW  | Bit[7:0]: HSYNC start point[7:0]  |
| 0x3818  | HSYNC WIDTH     | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: HSYNC width[11:8]   |
| 0x3819  | HSYNC WIDTH     | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: HSYNC width[7:8]  |
| 0x3820  | TIMING TC REG20 | 0x40          | RW  | Timing Control<br>Bit[7:3]: Debug mode<br>Bit[2]: ISP vflip<br>Bit[1]: Sensor vflip   |
| 0x3821  | TIMING TC REG21 | 0x00          | RW  | Timing Control<br>Bit[7:6]: Debug mode<br>Bit[5]: JPEG enable<br>Bit[4:3]: Debug mode<br>Bit[2]: ISP mirror<br>Bit[1]: Sensor mirror<br>Bit[0]: Horizontal binning enable |

table 7-8 AEC/AGC power down domain control registers (sheet 1 of 4)

| address | register name       | default value | R/W | description   |
|---------|---------------------|---------------|-----|---|
| 0x3A00  | AEC_CTRL00          | 0x78          | RW  | AEC System Control (0: disable; 1: enable)<br>Bit[7]: Debug mode<br>Bit[6]: Less one line enable<br>Bit[5]: Band function enable<br>Bit[4]: Less 1 band enable<br>Bit[3]: Start selection<br>Bit[2]: Night mode<br>Bit[1]: New balance function<br>Bit[0]: Freeze |
| 0x3A01  | AEC_MIN_EXPOSURE    | 0x01          | RW  | Minimum Exposure Output Limit<br>Bit[7:0]: Minimum exposure   |
| 0x3A02  | AEC_MAX_EXPO (60HZ) | 0x3D          | RW  | 60Hz Maximum Exposure Output Limit<br>Bit[7:0]: Maximum exposure[15:8]  |
| 0x3A03  | AEC_MAX_EXPO (60HZ) | 0x80          | RW  | 60Hz Maximum Exposure Output Limit<br>Bit[7:0]: Maximum exposure[7:0]   |

table 7-8 AEC/AGC power down domain control registers (sheet 2 of 4)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x3A04  | DEBUG MODE    | –             | –   | Debug Mode  |
| 0x3A05  | AEC CTRL05    | 0x30          | RW  | AEC System Control 2<br>Bit[7]: Debug mode<br>Bit[6]: frame insert<br>0: In night mode, insert frame disable<br>1: In night mode, insert frame enable<br>Bit[5]: Step auto enable<br>0: Step manual mode<br>1: Step auto mode<br>Bit[4:0]: Step auto ratio<br>In step auto mode, step ratio setting to adjust speed |
| 0x3A06  | AEC CTRL06    | 0x10          | RW  | AEC System Control 3<br>Bit[7:5]: Debug mode<br>Bit[4:0]: Step manual setting 1<br>Step manual<br>Increase mode fast step   |
| 0x3A07  | AEC CTRL07    | 0x18          | RW  | AEC Manual Step Register<br>Bit[7:4]: Step manual setting 2<br>Step manual, slow step<br>Bit[3:0]: Step manual setting 3<br>Step manual, decrease mode fast step  |
| 0x3A08  | AEC B50 STEP  | 0x01          | RW  | 50Hz Band Width<br>Bit[7:2]: Debug mode<br>Bit[1:0]: B50 step[9:8]  |
| 0x3A09  | AEC B50 STEP  | 0x27          | RW  | 50Hz Band Width<br>Bit[7:0]: B50 step[7:0]  |
| 0x3A0A  | AEC B60 STEP  | 0x00          | RW  | 60Hz Band Width<br>Bit[7:2]: Debug mode<br>Bit[1:0]: B60 step[13:8]   |
| 0x3A0B  | AEC B60 STEP  | 0xF6          | RW  | 60Hz Band Width<br>Bit[7:0]: B60 step[7:0]  |
| 0x3A0C  | AEC CTRL0C    | 0xE4          | RW  | Bit[7:4]: E1 max<br>Decimal line high limit zone<br>Bit[3:0]: E1 min<br>Decimal line low limit zone   |
| 0x3A0D  | AEC CTRL0D    | 0x08          | RW  | 60Hz Max Bands in One Frame<br>Bit[7:6]: Debug mode<br>Bit[5:0]: B60 max  |

table 7-8 AEC/AGC power down domain control registers (sheet 3 of 4)

| address | register name       | default value | R/W | description  |
|---------|---------------------|---------------|-----|--|
| 0x3A0E  | AEC CTRL0E          | 0x06          | RW  | 50Hz Max Bands in One Frame<br>Bit[7:6]: Debug mode<br>Bit[5:0]: B50 max   |
| 0x3A0F  | AEC CTRL0F          | 0x78          | RW  | Stable Range High Limit (Enter)<br>Bit[7:0]: WPT   |
| 0x3A10  | AEC CTRL10          | 0x68          | RW  | Stable Range Low Limit (Enter)<br>Bit[7:0]: BPT  |
| 0x3A11  | AEC CTRL11          | 0xD0          | RW  | Step Manual Mode, Fast Zone High Limit<br>Bit[7:0]: VPT high   |
| 0x3A12  | DEBUG MODE          | –             | –   | Debug Mode   |
| 0x3A13  | AEC CTRL13          | 0x40          | RW  | Bit[7]: Debug mode<br>Bit[6]: Pre-gain enable<br>Bit[5:0]: Pre-gain value<br>0x40 = 1x   |
| 0x3A14  | AEC MAX EXPO (50HZ) | 0x0E          | RW  | 50Hz Maximum Exposure Output Limit<br>Bit[7:4]: Debug mode<br>Bit[3:0]: Max exposure[15:8]                                     |
| 0x3A15  | AEC MAX EXPO (50HZ) | 0x40          | RW  | 50Hz Maximum Exposure Output Limit<br>Bit[7:0]: Max exposure[7:0]  |
| 0x3A16  | DEBUG MODE          | –             | –   | Debug Mode   |
| 0x3A17  | AEC CTRL17          | 0x01          | RW  | Gain Base When in Night Mode<br>Bit[7:2]: Debug mode<br>Bit[1:0]: Gain night threshold<br>00: 00<br>01: 10<br>10: 30<br>11: 70 |
| 0x3A18  | AEC GAIN CEILING    | 0x03          | RW  | Gain Output Top Limit<br>Bit[7:2]: Debug mode<br>Bit[1:0]: AEC gain ceiling[9:8]<br>Real gain format                           |
| 0x3A19  | AEC GAIN CEILING    | 0xE0          | RW  | Gain Output Top Limit<br>Bit[7:0]: AEC gain ceiling[7:0]<br>Real gain format   |
| 0x3A1A  | AEC DIFF MIN        | 0x04          | RW  | Reserved Default Value for This Register<br>Bit[7:0]: Difference minimal   |
| 0x3A1B  | AEC CTRL1B          | 0x78          | RW  | Stable Range High Limit (Go Out)<br>Bit[7:0]: WPT2   |
| 0x3A1C  | LED ADD ROW         | 0x06          | RW  | Exposure Values Added When Strobe is On<br>Bit[7:0]: AEC LED add row[15:8]   |

table 7-8 AEC/AGC power down domain control registers (sheet 4 of 4)

| address           | register name | default value | R/W | description   |
|-------------------|---------------|---------------|-----|---|
| 0x3A1D            | LED ADD ROW   | 0x18          | RW  | Exposure Values Added When Strobe is On<br>Bit[7:0]: AEC LED add row[7:0]   |
| 0x3A1E            | AEC CTRL1E    | 0x68          | RW  | Stable Range Low Limit (Go Out)<br>Bit[7:0]: BPT2                           |
| 0x3A1F            | AEC CTRL1F    | 0x40          | RW  | Step Manual Mode, Fast Zone Low Limit<br>Bit[7:0]: VPT low                  |
| 0x3A20            | AEC CTRL20    | 0x20          | RW  | Bit[7:3]: Debug mode<br>Bit[2]: Strobe option<br>Bit[1:0]: Debug mode       |
| 0x3A21            | AEC CTRL21    | 0x78          | RW  | Bit[7]: Debug mode<br>Bit[6:4]: Insert frame number<br>Bit[3:0]: Debug mode |
| 0x3A22~<br>0x3A24 | DEBUG MODE    | –             | –   | Debug Mode  |
| 0x3A25            | AEC CTRL25    | 0x00          | RW  | Bit[7:5]: Debug mode<br>Bit[4:2]: Freeze counter<br>Bit[1:0]: Debug mode    |

table 7-9 STROBE registers (sheet 1 of 2)

| address | register name            | default value | R/W | description  |
|---------|--------------------------|---------------|-----|--|
| 0x3B00  | STROBE CTRL              | 0x00          | RW  | Bit[7]: Strobe request on/off<br>0: Off<br>1: On<br>Bit[6]: Strobe pulse reverse<br>Bit[3:2]: width_in_xenon<br>Bit[1:0]: Strobe mode<br>00: Xenon<br>01: LED1<br>10: LED2<br>11: LED3 |
| 0x3B01  | FREX EXPOSURE 02         | 0x00          | RW  | Bit[7:0]: FREX exposure time[23:16]  |
| 0x3B02  | FREX SHUTTER<br>DELAY 01 | 0x08          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Shutter delay time[12:8]   |
| 0x3B03  | FREX SHUTTER<br>DELAY 00 | 0x00          | RW  | Bit[7:0]: Shutter delay time[7:0]<br>Unit: 64x SCLK cycle  |
| 0x3B04  | FREX EXPOSURE 01         | 0x04          | RW  | Bit[7:0]: FREX exposure time[15:8]   |
| 0x3B05  | FREX EXPOSURE 00         | 0x00          | RW  | Bit[7:0]: FREX exposure time[7:0]<br>Unit: Tline   |



table 7-9 STROBE registers (sheet 2 of 2)

| address | register name   | default value | R/W | description  |
|---------|-----------------|---------------|-----|--|
| 0x3B06  | FREX CTRL 07    | 0x04          | RW  | Bit[7:4]: FREX frame delay<br>Bit[3:0]: Strobe width[3:0]  |
| 0x3B07  | FREX MODE       | 0x08          | RW  | FREX Mode Selection<br>Bit[7:0]: FREX mode<br>00: FREX strobe mode0<br>01: FREX strobe mode1<br>1x: Rolling strobe   |
| 0x3B08  | FREX REQUEST    | 0x00          | RW  | FREX Request   |
| 0x3B09  | FREX HREF DELAY | 0x02          | RW  | FREX HREF Delay  |
| 0x3B0A  | FREX RST LENGTH | 0x04          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: FREX precharge length<br>000: 1/16 Tline<br>001: 1/8 Tline<br>010: 1/4 Tline<br>011: 1/2 Tline<br>100: 1 Tline<br>101: 2 Tline<br>110: 4 Tline<br>111: 8 Tline |
| 0x3B0B  | STROBE WIDTH    | 0x00          | RW  | Bit[7:0]: Strobe width[19:12]  |
| 0x3B0C  | STROBE WIDTH    | 0x3D          | RW  | Bit[7:0]: Strobe width[11:4]   |

table 7-10 5060HZ detector registers (sheet 1 of 3)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x3C00  | 5060HZ CTRL00 | 0x00          | RW  | Bit[7:6]: Debug mode<br>Bit[5]: Time counter threshold divisor enable<br>Bit[4]: Low limit enable<br>Bit[3]: Debug mode<br>Bit[2]: Band50 default value<br>Bit[1:0]: Time counter threshold<br>00: 1s<br>01: 2s<br>10: 4s<br>11: 8s |
| 0x3C01  | 5060HZ CTRL01 | 0x00          | RW  | Bit[7]: Band manual enable<br>Bit[6]: Band begin reset enable<br>Bit[5]: Sum auto mode enable<br>Bit[4]: Band counter enable<br>Bit[3:0]: Band counter<br>Counter threshold for band change   |

table 7-10 5060HZ detector registers (sheet 2 of 3)

| address | register name          | default value | R/W | description   |
|---------|------------------------|---------------|-----|---|
| 0x3C02  | 5060HZ CTRL02          | 0x00          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Low light threshold<br>No detection under low light<br>Bit[7:6]: Low light limit mode |
| 0x3C03  | 5060HZ CTRL03          | 0x00          | RW  | Bit[7:0]: Counter threshold for low light   |
| 0x3C04  | 5060HZ CTRL04          | 0x20          | RW  | Bit[7:0]: Threshold for low sum   |
| 0x3C05  | 5060HZ CTRL05          | 0x70          | RW  | Bit[7:0]: Threshold for high sum  |
| 0x3C06  | LIGHT METER1 THRESHOLD | 0x00          | RW  | Bit[7:0]: Lightmeter1 threshold[15:8]   |
| 0x3C07  | LIGHT METER1 THRESHOLD | 0x00          | RW  | Bit[7:0]: Lightmeter1 threshold[7:0]  |
| 0x3C08  | LIGHT METER2 THRESHOLD | 0x01          | RW  | Bit[7:0]: Lightmeter2 threshold[15:8]   |
| 0x3C09  | LIGHT METER2 THRESHOLD | 0x2C          | RW  | Bit[7:0]: Lightmeter2 threshold[7:0]  |
| 0x3C0A  | SAMPLE NUMBER          | 0x4E          | RW  | Bit[7:0]: Sample number[15:8]   |
| 0x3C0B  | SAMPLE NUMBER          | 0x1F          | RW  | Bit[7:0]: Sample number[7:0]  |
| 0x3C0C  | SIGMADELTA CTRL0C      | –             | R   | Bit[7:4]: Band counter<br>Bit[1]: Sign bit of sum50/60<br>Bit[0]: Band50/60<br>0: 60Hz light<br>1: 50Hz light           |
| 0x3C0D  | SUM 50                 | –             | R   | Bit[7:5]: Debug mode<br>Bit[4:0]: Sum50[28:24]  |
| 0x3C0E  | SUM 50                 | –             | R   | Bit[7:0]: Sum50[23:16]  |
| 0x3C0F  | SUM 50                 | –             | R   | Bit[7:0]: Sum50[15:8]   |
| 0x3C10  | SUM 50                 | –             | R   | Bit[7:0]: Sum50[7:0]  |
| 0x3C11  | SUM 60                 | –             | R   | Bit[7:5]: Debug mode<br>Bit[4:0]: Sum60[28:24]  |
| 0x3C12  | SUM 60                 | –             | R   | Bit[7:0]: Sum60[23:16]  |
| 0x3C13  | SUM 60                 | –             | R   | Bit[7:0]: Sum60[15:8]   |
| 0x3C14  | SUM 60                 | –             | R   | Bit[7:0]: Sum60[7:0]  |
| 0x3C15  | SUM 50 60              | –             | R   | Bit[7:0]: Sum50/60[15:8]  |
| 0x3C16  | SUM 50 60              | –             | R   | Bit[7:0]: Sum50/60[7:0]   |
| 0x3C17  | BLOCK COUNTER          | –             | R   | Bit[7:0]: Block counter[15:8]   |

table 7-10 5060HZ detector registers (sheet 3 of 3)

| address | register name     | default value | R/W | description   |
|---------|-------------------|---------------|-----|---|
| 0x3C18  | BLOCK COUNTER     | –             | R   | Bit[7:0]: Block counter[7:0]                                |
| 0x3C19  | B6                | –             | R   | Bit[7:0]: B6[15:8]  |
| 0x3C1A  | B6                | –             | R   | Bit[7:0]: B6[7:0]   |
| 0x3C1B  | LIGHTMETER OUTPUT | –             | R   | Bit[7:4]: Debug mode<br>Bit[3:0]: Light meter output[19:16] |
| 0x3C1C  | LIGHTMETER OUTPUT | –             | R   | Bit[7:0]: Light meter output[15:8]                          |
| 0x3C1D  | LIGHTMETER OUTPUT | –             | R   | Bit[7:0]: Light meter output[7:0]                           |
| 0x3C1E  | SUM THRESHOLD     | –             | R   | Sum Threshold   |

table 7-11 OTP control functions (sheet 1 of 2)

| address | register name | default value | R/W | description          |
|---------|---------------|---------------|-----|----------------------|
| 0x3D00  | OTP DATA00    | 0x00          | RW  | OTP Dump/Load Data00 |
| 0x3D01  | OTP DATA01    | 0x00          | RW  | OTP Dump/Load Data01 |
| 0x3D02  | OTP DATA02    | 0x00          | RW  | OTP Dump/Load Data02 |
| 0x3D03  | OTP DATA03    | 0x00          | RW  | OTP Dump/Load Data03 |
| 0x3D04  | OTP DATA04    | 0x00          | RW  | OTP Dump/Load Data04 |
| 0x3D05  | OTP DATA05    | 0x00          | RW  | OTP Dump/Load Data05 |
| 0x3D06  | OTP DATA06    | 0x00          | RW  | OTP Dump/Load Data06 |
| 0x3D07  | OTP DATA07    | 0x00          | RW  | OTP Dump/Load Data07 |
| 0x3D08  | OTP DATA08    | 0x00          | RW  | OTP Dump/Load Data08 |
| 0x3D09  | OTP DATA09    | 0x00          | RW  | OTP Dump/Load Data09 |
| 0x3D0A  | OTP DATA0A    | 0x00          | RW  | OTP Dump/Load Data0a |
| 0x3D0B  | OTP DATA0B    | 0x00          | RW  | OTP Dump/Load Data0b |
| 0x3D0C  | OTP DATA0C    | 0x00          | RW  | OTP Dump/Load Data0c |
| 0x3D0D  | OTP DATA0D    | 0x00          | RW  | OTP Dump/Load Data0d |
| 0x3D0E  | OTP DATA0E    | 0x00          | RW  | OTP Dump/Load Data0e |
| 0x3D0F  | OTP DATA0F    | 0x00          | RW  | OTP Dump/Load Data0f |
| 0x3D10  | OTP DATA10    | 0x00          | RW  | OTP Dump/Load Data10 |

table 7-11 OTP control functions (sheet 2 of 2)

| address | register name    | default value | R/W | description   |
|---------|------------------|---------------|-----|---|
| 0x3D11  | OTP DATA11       | 0x00          | RW  | OTP Dump/Load Data11  |
| 0x3D12  | OTP DATA12       | 0x00          | RW  | OTP Dump/Load Data12  |
| 0x3D13  | OTP DATA13       | 0x00          | RW  | OTP Dump/Load Data13  |
| 0x3D14  | OTP DATA14       | 0x00          | RW  | OTP Dump/Load Data14  |
| 0x3D15  | OTP DATA15       | 0x00          | RW  | OTP Dump/Load Data15  |
| 0x3D16  | OTP DATA16       | 0x00          | RW  | OTP Dump/Load Data16  |
| 0x3D17  | OTP DATA17       | 0x00          | RW  | OTP Dump/Load Data17  |
| 0x3D18  | OTP DATA18       | 0x00          | RW  | OTP Dump/Load Data18  |
| 0x3D19  | OTP DATA19       | 0x00          | RW  | OTP Dump/Load Data19  |
| 0x3D1A  | OTP DATA1A       | 0x00          | RW  | OTP Dump/Load Data1A  |
| 0x3D1B  | OTP DATA1B       | 0x00          | RW  | OTP Dump/Load Data1B  |
| 0x3D1C  | OTP DATA1C       | 0x00          | RW  | OTP Dump/Load Data1C  |
| 0x3D1D  | OTP DATA1D       | 0x00          | RW  | OTP Dump/Load Data1D  |
| 0x3D1E  | OTP DATA1E       | 0x00          | RW  | OTP Dump/Load Data1E  |
| 0x3D1F  | OTP DATA1F       | 0x00          | RW  | OTP Dump/Load Data1F  |
| 0x3D20  | OTP PROGRAM CTRL | 0x00          | RW  | Bit[7]: OTP program busy<br>Bit[6:2]: Debug mode<br>Bit[1]: OTP program speed<br>0: Fast<br>1: Slow<br>Bit[0]: OTP program enable |
| 0x3D21  | OTP READ CTRL    | 0x00          | RW  | Bit[7]: OTP read busy<br>Bit[1]: OTP read speed<br>0: Fast<br>1: Slow<br>Bit[0]: OTP read enable                                  |

table 7-12 BIST registers (sheet 1 of 3)

| address | register name      | default value | R/W | description                                |
|---------|--------------------|---------------|-----|--|
| 0x3E00  | BIST_START_ADDRESS | 0x00          | RW  | Bit[7:0]: Specify BIST start address[15:8] |
| 0x3E01  | BIST_START_ADDRESS | 0x00          | RW  | Bit[7:0]: Specify BIST start address[7:0]  |
| 0x3E02  | BIST_END_ADDRESS   | 0x0F          | RW  | Bit[7:0]: Specify BIST end address[15:8]   |

table 7-12 BIST registers (sheet 2 of 3)

| address | register name    | default value | R/W | description   |
|---------|------------------|---------------|-----|---|
| 0x3v03  | BIST_END_ADDRESS | 0xDB          | RW  | Bit[7:0]: Specify BIST end address[7:0]   |
| 0x3E04  | BIST_R4          | 0x14          | RW  | BIST Option Counter<br>Bit[7:5]: Debug mode<br>Bit[4:0]: BIST OP counter<br>Specify BIST operation number<br>It can be settled from 0x02 to 0x14  |
| 0x3E05  | BIST_R5          | 0x00          | RW  | BIST Special Data<br>Bit[7:0]: Specify data written into memory in special mode   |
| 0x3E06  | BIST_R6          | 0x03          | RW  | SRAM Select<br>Specify SRAM Select  |
| 0x3E07  | BIST_CTRL07      | 0x20          | RW  | Bit[7]: BIST start<br>Specify BIST start<br>It should first be high, then low<br>Bit[6]: Specify BIST restart<br>Bit[5]: BIST mode<br>Specifies BIST test mode<br>0: Manual mode<br>1: Auto mode<br>Bit[4]: Special BIST special mode<br>Bit[3]: Specify read only mode<br>Bit[2:0]: Debug mode |
| 0x3E0C  | BIST_CTRL12      | 0x00          | RW  | Bit[7:5]: Debug mode<br>Bit[4]: BIST busy<br>Specify BIST busy status<br>Bit[2:0]: BIST error byte<br>Specify BIST error type   |
| 0x3E0D  | BIST_RESULT00    | 0x00          | RW  | BIST Test Result00 (0: BIST pass; 1: BIST error)<br>Bit[7:2]: Debug mode<br>Bit[1]: MCPRAM<br>Bit[0]: SRM   |
| 0x3E0E  | BIST_RESULT01    | 0x00          | RW  | BIST Test Result01 (0: BIST pass; 1: BIST error)<br>Bit[7]: PFIFO2<br>Bit[6]: PFIFO1<br>Bit[5]: VFIFO<br>Bit[4]: JFIFO<br>Bit[3]: SFIFO<br>Bit[2]: SCALE2<br>Bit[1]: SCALE1<br>Bit[0]: DPC4   |

table 7-12 BIST registers (sheet 3 of 3)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x3E0F  | BIST RESULT02 | 0x00          | RW  | BIST Test Result02 (0: BIST pass; 1: BIST error)<br>Bit[7]: DPC3<br>Bit[6]: DPC2<br>Bit[5]: DPC1<br>Bit[4]: CIP3<br>Bit[3]: CIP2<br>Bit[2]: CIP1<br>Bit[1]: AWB<br>Bit[0]: MCU IRAM |

table 7-13 MC registers (sheet 1 of 3)

| address | register name              | default value | R/W | description   |
|---------|----------------------------|---------------|-----|---|
| 0x3F00  | MC CTRL00                  | 0x00          | RW  | Bit[7:5]: Debug mode<br>Bit[4]: DW8051 manual reset enable<br>Bit[3]: DW8051 manual reset<br>Bit[2]: IRAM manual reset enable<br>Bit[1]: IRAM manual reset<br>Bit[0]: Soft reset MCU<br>0: Debug mode<br>1: Reset MCU |
| 0x3F01  | MC INTERRUPT MASK0         | 0x00          | RW  | Bit[7:0]: Mask0 for interrupt<br>0: Disable interrupt bit<br>1: Enable interrupt bit  |
| 0x3F02  | MC INTERRUPT MASK1         | 0x00          | RW  | Bit[7:0]: Mask1 for interrupt<br>0: Disable interrupt bit<br>1: Enable interrupt bit  |
| 0x3F03  | MC READ INTERRUPT ADDRESS  | 0x70          | RW  | Bit[7:0]: Set high byte for SCCB address that will trigger interrupt when read  |
| 0x3F04  | MC READ INTERRUPT ADDRESS  | 0x00          | RW  | Bit[7:0]: Set low byte for SCCB address that will trigger interrupt when read   |
| 0x3F05  | MC WRITE INTERRUPT ADDRESS | 0x70          | RW  | Bit[7:0]: Set high byte for SCCB address that will trigger interrupt when written   |
| 0x3F06  | MC WRITE INTERRUPT ADDRESS | 0x04          | RW  | Bit[7:0]: Set low byte for SCCB address that will trigger interrupt when written  |

table 7-13 MC registers (sheet 2 of 3)

| address | register name                  | default value | R/W | description   |
|---------|--------------------------------|---------------|-----|---|
| 0x3F08  | MC INTERRUPT SOURCE SELECTION1 | 0x00          | RW  | Bit[7:6]: Interrupt1[7] source selection<br>0x: AEC done<br>10: Debug mode<br>11: DVP HREF falling edge<br>Bit[5:4]: Interrupt1[6] source selection<br>00: ISP average done<br>01: FREX rising edge<br>10: Debug mode<br>11: DVP VSYNC falling edge<br>Bit[3:2]: Interrupt1[5] source selection<br>00: AEC trigger<br>01: FREX falling edge<br>10: MIPI frame end<br>11: DVP VSYNC rising edge<br>Bit[1:0]: Interrupt1[4] source selection<br>0x: JPEG over size<br>10: MIPI line end<br>11: DVP HREF rising edge   |
| 0x3F09  | MC INTERRUPT SOURCE SELECTION2 | 0x00          | RW  | Bit[7:6]: Interrupt1[3] source selection<br>0x: Debug mode<br>11: SRM operation start<br>Bit[5:4]: Interrupt1[2] source selection<br>0x: Debug mode<br>11: SRM operation done<br>Bit[3:2]: Interrupt1[1] source selection<br>0x: Debug mode<br>11: DVP frame counter change<br>Bit[1:0]: Interrupt1[0] source selection<br>0x: BLC SOF<br>11: Debug mode  |
| 0x3F0A  | MC INTERRUPT SOURCE SELECTION3 | 0x00          | RW  | Bit[7:6]: Interrupt0[7] source selection<br>00: JFIFO over flow<br>01: ISP end frame<br>10: SFIFO end of frame<br>11: JFIFO end frame<br>Bit[5:4]: Interrupt0[6] source selection<br>0x: JFIFO end of image<br>10: F1 FIFO write<br>11: VFIFO ready<br>Bit[3:2]: Interrupt0[5] source selection<br>00: ISP end of frame<br>01: SFIFO end of frame<br>10: JFIFO end of frame<br>11: VFIFO end of frame<br>Bit[1:0]: Interrupt0[4] source selection<br>00: ISP start of frame<br>01: SFIFO start of frame<br>10: JFIFO start of frame<br>11: VFIFO start of frame |

table 7-13 MC registers (sheet 3 of 3)

| address | register name                  | default value | R/W | description  |
|---------|--------------------------------|---------------|-----|--|
| 0x3F0B  | MC INTERRUPT SOURCE SELECTION4 | 0x00          | RW  | Bit[7:6]: Interrupt0[3] source selection<br>0x: AFC done<br>10: HREF falling edge<br>11: BIST done<br>Bit[5:4]: Interrupt0[2] source selection<br>00: AWB done<br>01: ISP start of frame<br>10: SFIFO start of frame<br>11: JFIFO start of frame<br>Bit[3:2]: Interrupt0[1] source selection<br>00: VFIFO full<br>01: ISP FC SOF<br>10: f2_fifo_wr<br>11: Write a specific address<br>Bit[1:0]: Interrupt0[0] source selection<br>00: VFIFO empty<br>01: ISP line<br>10: ISP end of frame<br>11: Read a specific address |
| 0x3F0C  | MC INTERRUPT0 STATUS           | –             | R   | Bit[7:0]: Interrupt0 status indication   |
| 0x3F0D  | MC INTERRUPT1 STATUS           | –             | R   | Bit[7:0]: Interrupt1 status indication   |

table 7-14 BLC registers (sheet 1 of 3)

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x4000  | BLC CTRL00    | 0x89          | RW  | BLC Control 00 (0: disable; 1: enable)<br>Bit[7:3]: Debug mode<br>Bit[2]: Apply2blackline<br>Bit[1]: Black line average frame<br>Bit[0]: BLC enable                                  |
| 0x4001  | BLC CTRL01    | 0x00          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: BLC start line   |
| 0x4002  | BLC CTRL02    | 0x45          | RW  | Bit[7]: Format change enable<br>BLC update when format changes<br>Bit[6]: BLC auto enable<br>0: Manual<br>1: Auto<br>Bit[5:0]: Reset frame number<br>Frame number BLC do after reset |



table 7-14 BLC registers (sheet 2 of 3)

| address           | register name | default value | R/W | description   |
|-------------------|---------------|---------------|-----|---|
| 0x4003            | BLC CTRL03    | 0x08          | RW  | Bit[7]: BLC redo enable<br>Write 1 to thisbit will trigger a BLC redo N frames begin, where N is 0x4003[5:0]<br>Bit[6]: BLC freeze<br>Bit[5:0]: Manual frame number   |
| 0x4004            | BLC CTRL04    | 0x08          | RW  | Bit[7:0]: BLC line number<br>Specify the line number BLC process  |
| 0x4005            | BLC CTRL05    | 0x18          | RW  | Bit[7:2]: Debug mode<br>Bit[1]: BLC always update<br>0: Normal freeze<br>1: BLC always update<br>Bit[0]: Debug mode   |
| 0x4006            | DEBUG MODE    | --            | --  | Debug Mode  |
| 0x4007            | BLC CTRL07    | 0x00          | RW  | Bit[7:5]: Debug mode<br>Bit[4:3]: Window selection<br>00: Full image<br>01: A windows not contain the first 16 pixels and the end 16 pixels<br>10: A windows not contain the first 1/16 image and the end 1/16 image<br>11: A windows not contain the first 1/8 image and the end 1/8 image<br>Bit[2:0]: Debug mode |
| 0x4008            | DEBUG MODE    | --            | --  | Debug Mode  |
| 0x4009            | BLACK LEVEL   | 0x10          | RW  | Bit[7:0]: BLC black level target at 10-bit range  |
| 0x400C~<br>0x402B | DEBUG MODE    | --            | --  | Debug Mode  |
| 0x402C            | BLACK LEVEL00 | 0x00          | RW  | Bit[7:0]: Blacklevel00[15:8]<br>With 3 decimal  |
| 0x402D            | BLACK LEVEL00 | 0x00          | RW  | Bit[7:0]: Blacklevel00[7:0]<br>With 3 decimal   |
| 0x402E            | BLACK LEVEL01 | 0x00          | RW  | Bit[7:0]: Blacklevel01[15:8]<br>With 3 decimal  |
| 0x402F            | BLACK LEVEL01 | 0x00          | RW  | Bit[7:0]: Blacklevel01[7:0]<br>With 3 decimal   |
| 0x4030            | BLACK LEVEL10 | 0x00          | RW  | Bit[7:0]: Blacklevel10[15:8]<br>With 3 decimal  |

table 7-14 BLC registers (sheet 3 of 3)

| address | register name | default value | R/W | description                                    |
|---------|---------------|---------------|-----|--|
| 0x4031  | BLACK LEVEL10 | 0x00          | RW  | Bit[7:0]: Blacklevel10[7:0]<br>With 3 decimal  |
| 0x4032  | BLACK LEVEL11 | 0x00          | RW  | Bit[7:0]: Blacklevel11[15:8]<br>With 3 decimal |
| 0x4033  | BLACK LEVEL11 | 0x00          | RW  | Bit[7:0]: Blacklevel11[7:0]<br>With 3 decimal  |

table 7-15 frame control registers

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x4201  | FRAME CTRL01  | 0x00          | R/W | Control Passed Frame Number<br>When both ON and OFF number set to 0x00,<br>frame control is in bypass mode<br>Bit[7:4]: Debug mode<br>Bit[3:0]: Frame ON number  |
| 0x4202  | FRAME CTRL02  | 0x00          | R/W | Control Masked Frame Number<br>When both ON and OFF number set to 0x00,<br>frame control is in bypass mode<br>Bit[7:4]: Debug mode<br>BIT[3:0]: Frame OFF number |

table 7-16 FORMAT control registers (sheet 1 of 5)

| address | register name     | default value | R/W | description   |
|---------|-------------------|---------------|-----|---|
| 0x4300  | FORMAT CONTROL 00 | 0xF8          | RW  | <p>Format Control 00</p> <p>Bit[7:4]: Output format of formatter module</p> <p>0x0: RAW</p> <p>Bit[3:0]: Output sequence</p> <p>0x0: BGBG... / GRGR...</p> <p>0x1: GBGB... / RGRG...</p> <p>0x2: GRGR... / BGBG...</p> <p>0x3: RGRG... / GBGB...</p> <p>0x4~0xF: Not allowed</p> <p>0x1: Y8</p> <p>Bit[3:0]: Does not matter</p> <p>0x2: YUV444/RGB888 (not available for full resolution)</p> <p>Bit[3:0]: Output sequence</p> <p>0x0: YUVYUV..., or<br/>GBRGRB...</p> <p>0x1: YVUYVU..., or<br/>GRBGRB...</p> <p>0x2: UYVUYV..., or<br/>BGRBGR...</p> <p>0x3: VYUVYU..., or<br/>RBRGRB...</p> <p>0x4: UYVUYV..., or<br/>BRBGRG...</p> <p>0x5: VUYVUY..., or<br/>RBGRBG...</p> <p>0x6~0xE: Not allowed</p> <p>0xF: UYVUYV..., or<br/>BGRBGR...</p> <p>0x3: YUV422</p> <p>Bit[3:0]: Output sequence</p> <p>0x0: YUYV...</p> <p>0x1: YVYU...</p> <p>0x2: UYVY...</p> <p>0x3: VYUY...</p> <p>0x4~0xE: Not allowed</p> <p>0xF: UYVY...</p> <p>0x4: YUV420</p> <p>Bit[3:0]: Output sequence</p> <p>0x0: YYYY... / YUYV...</p> <p>0x1: YYYY... / YVYU...</p> <p>0x2: YYYY... / UYVY...</p> <p>0x3: YYYY... / VYUY...</p> <p>0x4: YUYV... / YYYY...</p> |

table 7-16 FORMAT control registers (sheet 2 of 5)

| address | register name             | default value | R/W | description                                |
|---------|---------------------------|---------------|-----|--|
|         |                           |               |     | 0x5: YVYU... / YYYY...                     |
|         |                           |               |     | 0x6: UYVY... / YYYY...                     |
|         |                           |               |     | 0x7: VYUY... / YYYY...                     |
|         |                           |               |     | 0x8~0xE: Not allowed                       |
|         |                           |               |     | 0xF: YYYY... / UYVY...                     |
| 0x5:    | YUV420 (for MIPI only)    |               |     |  |
|         | Bit[3:0]: Output sequence |               |     |  |
|         |                           |               |     | 0x0~0xD: Not allowed                       |
|         |                           |               |     | 0xE: VYVY... / UYUY...                     |
|         |                           |               |     | 0xF: UYUY... / VYVY...                     |
| 0x6:    | RGB565                    |               |     |  |
|         | Bit[3:0]: Output sequence |               |     |  |
|         |                           |               |     | 0x0: {b[4:0],g[5:3]}, {g[2:0],r[4:0]}      |
|         |                           |               |     | 0x1: {r[4:0],g[5:3]}, {g[2:0],b[4:0]}      |
|         |                           |               |     | 0x2: {g[4:0],r[5:3]}, {r[2:0],b[4:0]}      |
|         |                           |               |     | 0x3: {b[4:0],r[5:3]}, {r[2:0],g[4:0]}      |
|         |                           |               |     | 0x4: {g[4:0],b[5:3]}, {b[2:0],r[4:0]}      |
|         |                           |               |     | 0x5: {r[4:0],b[5:3]}, {b[2:0],g[4:0]}      |
|         |                           |               |     | 0x6~0xE: Not allowed                       |
|         |                           |               |     | 0xF: {g[2:0],b[4:0]}, {r[4:0],g[5:3]}      |
| 0x7:    | RGB555 format 1           |               |     |  |
|         | Bit[3:0]: Output sequence |               |     |  |
|         |                           |               |     | 0x0: {b[4:0],g[4:2]}, {g[1:0],1'b0,r[4:0]} |
|         |                           |               |     | 0x1: {r[4:0],g[4:2]}, {g[1:0],1'b0,b[4:0]} |
|         |                           |               |     | 0x2: {g[4:0],r[4:2]}, {r[1:0],1'b0,b[4:0]} |
|         |                           |               |     | 0x3: {b[4:0],r[4:2]}, {r[1:0],1'b0,g[4:0]} |
|         |                           |               |     | 0x4: {r[4:0],b[4:2]}, {b[1:0],1'b0,g[4:0]} |
|         |                           |               |     | 0x5: {g[4:0],b[4:2]}, {b[1:0],1'b0,r[4:0]} |
|         |                           |               |     | 0x6~0xE: Not allowed                       |
|         |                           |               |     | 0xF: {g[1:0],1'b0,b[4:0]}, {r[4:0],g[4:2]} |
| 0x8:    | RGB555 format 2           |               |     |  |
|         | Bit[3:0]: Output sequence |               |     |  |
|         |                           |               |     | 0x0: {1'b0,b[4:0],g[4:3]}, {g[2:0],r[4:0]} |

table 7-16 FORMAT control registers (sheet 3 of 5)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
|         |               |               |     | 0x1: {1'b0,r[4:0],g[4:2]},<br>{g[2:0],b[4:0]}         |
|         |               |               |     | 0x2: {1'b0,g[4:0],r[4:2]},<br>{r[2:0],b[4:0]}         |
|         |               |               |     | 0x3: {1'b0,b[4:0],r[4:2]},<br>{r[2:0],g[4:0]}         |
|         |               |               |     | 0x4: {1'b0,r[4:0],b[4:2]},<br>{b[2:0],g[4:0]}         |
|         |               |               |     | 0x5: {1'b0,g[4:0],b[4:2]},<br>{b[2:0],r[4:0]}         |
|         |               |               |     | 0x6: {b[4:0],1'b0,g[4:3]},<br>{g[2:0],r[4:0]}         |
|         |               |               |     | 0x7: {r[4:0],1'b0,g[4:2]},<br>{g[2:0],b[4:0]}         |
|         |               |               |     | 0x8: {g[4:0],1'b0,r[4:2]},<br>{r[2:0],b[4:0]}         |
|         |               |               |     | 0x9: {b[4:0],1'b0,r[4:2]},<br>{r[2:0],g[4:0]}         |
|         |               |               |     | 0xA: {r[4:0],1'b0,b[4:2]},<br>{b[2:0],g[4:0]}         |
|         |               |               |     | 0xB: {g[4:0],1'b0,b[4:2]},<br>{b[2:0],r[4:0]}         |
|         |               |               |     | 0xC~0xF: Not allowed                                  |
|         |               |               |     | 0x9: RGB444 format 1                                  |
|         |               |               |     | Bit[3:0]: Output sequence                             |
|         |               |               |     | 0x0: {1'b0,b[3:0],2'h0,g[3]},<br>{g[2:0],1'b0,r[3:0]} |
|         |               |               |     | 0x1: {1'b0,r[3:0],2'h0,g[3]},<br>{g[2:0],1'b0,b[3:0]} |
|         |               |               |     | 0x2: {1'b0,g[3:0],2'h0,r[3]},<br>{r[2:0],1'b0,b[3:0]} |
|         |               |               |     | 0x3: {1'b0,b[3:0],2'h0,r[3]},<br>{r[2:0],1'b0,g[3:0]} |
|         |               |               |     | 0x4: {1'b0,r[3:0],2'h0,b[3]},<br>{b[2:0],1'b0,g[3:0]} |
|         |               |               |     | 0x5: {1'b0,g[3:0],2'h0,b[3]},<br>{b[2:0],1'h0,r[3:0]} |
|         |               |               |     | 0x6: {b[3:0],1'b0,g[3:1]},<br>{g[0],2'h0,r[3:0],1'b0} |
|         |               |               |     | 0x7: {r[3:0],1'b0,g[3:1]},<br>{g[0],2'h0,b[3:0],1'b0} |
|         |               |               |     | 0x8: {g[3:0],1'b0,r[3:1]},<br>{r[0],2'h0,b[3:0],1'b0} |
|         |               |               |     | 0x9: {b[3:0],1'b0,r[3:1]},<br>{r[0],2'h0,g[3:0],1'b0} |
|         |               |               |     | 0xA: {r[3:0],1'b0,b[3:1]},<br>{b[0],2'h0,g[3:0],1'b0} |
|         |               |               |     | 0xB: {g[3:0],1'b0,b[3:1]},<br>{b[0],2'h0,r[3:0],1'b0} |
|         |               |               |     | 0xC~0xE: Not allowed                                  |

table 7-16 FORMAT control registers (sheet 4 of 5)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
|         |               |               |     | 0xF: {g[0],2'h2,b[3:0],1'b1},<br>{r[3:0],1'b1,g[3:1]}                       |
|         |               |               |     | 0xA: RGB444 format 2  |
|         |               |               |     | Bit[3:0]: Output sequence   |
|         |               |               |     | 0x0: {4'b0,b[3:0]},<br>{g[3:0],r[3:0]}                                      |
|         |               |               |     | 0x1: {4'b0,r[3:0]},<br>{g[3:0],b[3:0]}                                      |
|         |               |               |     | 0x2: {4'b0,b[3:0]},<br>{r[3:0],g[3:0]}                                      |
|         |               |               |     | 0x3: {4'b0,r[3:0]},<br>{b[3:0],g[3:0]}                                      |
|         |               |               |     | 0x4: {4'b0,g[3:0]},<br>{b[3:0],r[3:0]}                                      |
|         |               |               |     | 0x5: {4'b0,r[3:0]},<br>{r[3:0],b[3:0]}                                      |
|         |               |               |     | 0x6: {b[3:0],g[3:0],2'h0},<br>{r[3:0],b[3:0],2'h0,g[3:0]},<br>{r[3:0],2'h0} |
|         |               |               |     | 0x7: {r[3:0],g[3:0],2'h0},<br>{b[3:0],r[3:0],2'h0,g[3:0]},<br>{b[3:0],2'h0} |
|         |               |               |     | 0x8: {b[3:0],r[3:0],2'h0},<br>{g[3:0],b[3:0],2'h0,r[3:0]},<br>{g[3:0],2'h0} |
|         |               |               |     | 0x9: {r[3:0],b[3:0],2'h0},<br>{g[3:0],r[3:0],2'h0,b[3:0]},<br>{g[3:0],2'h0} |
|         |               |               |     | 0xA: {g[3:0],b[3:0],2'h0},<br>{r[3:0],g[3:0],2'h0,b[3:0]},<br>{r[3:0],2'h0} |
|         |               |               |     | 0xB: {g[3:0],r[3:0],2'h0},<br>{b[3:0],g[3:0],2'h0,r[3:0]},<br>{b[3:0],2'h0} |
|         |               |               |     | 0xC~0xF: Not allowed  |
|         |               |               |     | 0xB~0xE: Not allowed  |
|         |               |               |     | 0xF: Bypass formatter module (not recommended)                              |
|         |               |               |     | Bit[3:0]: Output format   |
|         |               |               |     | 0x8: Raw  |
|         |               |               |     | 0x9: YUV422   |
|         |               |               |     | 0xA: YUV444   |
|         |               |               |     | 0xE: VYYVYY.../UYVYY  |
|         |               |               |     | 0xF: UYYUY.../VYYVYY  |

table 7-16 FORMAT control registers (sheet 5 of 5)

| address | register name     | default value | R/W | description  |
|---------|-------------------|---------------|-----|--|
| 0x4301  | FORMAT CONTROL 01 | 0x00          | RW  | Format Control 01<br>Bit[1:0]: YUV422 UV control<br>00: U/V generated from average<br>01: U/V generated from first pixel<br>10: Not valid<br>11: U/V generated from second pixel |
| 0x4302  | YMAX VALUE        | 0x03          | RW  | Bit[7:2]: Debug mode<br>Bit[1:0]: Y max clip value[9:8]  |
| 0x4303  | YMAX VALUE        | 0xFF          | RW  | Bit[7:0]: Y max clip value[7:0]  |
| 0x4304  | YMIN VALUE        | 0x00          | RW  | Bit[7:2]: Debug mode<br>Bit[1:0]: Y min clip value[9:8]  |
| 0x4305  | YMIN VALUE        | 0x00          | RW  | Bit[7:0]: Y min clip value[7:0]  |
| 0x4306  | UMAX VALUE        | 0x03          | RW  | Bit[7:2]: Debug mode<br>Bit[1:0]: U max clip value[9:8]  |
| 0x4307  | UMAX VALUE        | 0xFF          | RW  | Bit[7:0]: U max clip value[7:0]  |
| 0x4308  | UMIN VALUE        | 0x00          | RW  | Bit[7:2]: Debug mode<br>Bit[1:0]: U min clip value[9:8]  |
| 0x4309  | UMIN VALUE        | 0x00          | RW  | Bit[7:0]: U min clip value[7:0]  |
| 0x430A  | VMAX VALUE        | 0x03          | RW  | Bit[7:2]: Debug mode<br>Bit[1:0]: V max clip value[9:8]  |
| 0x430B  | VMAX VALUE        | 0xFF          | RW  | Bit[7:0]: V max clip value[7:0]  |
| 0x430C  | VMIN VALUE        | 0x00          | RW  | Bit[7:2]: Debug mode<br>Bit[1:0]: V min clip value[9:8]  |
| 0x430D  | VMIN VALUE        | 0x00          | RW  | Bit[7:0]: V min clip value[7:0]  |

table 7-17 JPEG control registers (sheet 1 of 3)

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x4400  | JPEG CTRL00   | 0x81          | RW  | Bit[7]: input_format<br>0: YUV420<br>1: YUV422<br>Bit[6:0]: JFIFO read speed control |

table 7-17 JPEG control registers (sheet 2 of 3)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x4401  | JPEG CTRL01   | 0x01          | RW  | Bit[7:4]: SFIFO output buffer speed control<br>Bit[3]: Read SRAM enable when blanking<br>0: Disable<br>1: Enable<br>Bit[2]: Read SRAM at first blanking<br>0: Disable<br>1: Enable<br>Bit[1:0]: SFIFO read speed control  |
| 0x4402  | JPEG_CTRL02   | 0x10          | RW  | Bit[7]: SFIFO output control mode<br>0: Control by HREF and valid which before scale down<br>1: Control by input HREF and valid<br>Bit[6:4]: SOF control<br>001: Start at the first valid HREF<br>010: Start at the eighth valid HREF<br>Bit[3:0]: SFIFO output buffer speed control at last stripe         |
| 0x4403  | JPEG CTRL03   | 0x33          | RW  | Bit[7]: Memory select<br>0: Select ROM QT<br>1: Select SRAM QT<br>Bit[6]: MPEG enable<br>Bit[5]: Enable zero stuff<br>Bit[4]: Enable Huffman table output<br>Bit[3]: Rounding enable for C<br>Bit[2]: Rounding enable for Y<br>Bit[1]: Input shift 128 select for C<br>Bit[0]: Input shift 128 select for Y |
| 0x4404  | JPEG CTRL04   | 0x24          | RW  | Bit[7]: jfifo_pwrn<br>Bit[6]: SFIFO PWRDN<br>Bit[5]: Header output enable<br>Bit[4]: Enable gated clock<br>0: Disable gated clock<br>1: Enable gated clock<br>Bit[3]: Substitute 0xFF to 0xFE in QT<br>Bit[2:0]: Quantization rounding bias: Set value = Bias/8   |
| 0x4405  | JPEG CTRL05   | 0x40          | RW  | Bit[7:0]: QZ out truncate for Y   |
| 0x4406  | JPEG CTRL06   | 0x40          | RW  | Bit[7:0]: QZ out truncate for C   |
| 0x4407  | JPEG CTRL07   | 0x0C          | RW  | Bit[7]: Enable read QTA auto increment<br>Bit[5:0]: QS<br>Quantization scale  |



table 7-17 JPEG control registers (sheet 3 of 3)

| address           | register name  | default value | R/W | description  |
|-------------------|----------------|---------------|-----|--|
| 0x4408            | JPEG ISI CTRL  | 0x00          | RW  | Bit[7]: Scalado mode enable<br>0: Normal<br>1: Insert 0xFF after EOB<br>Bit[6]: JPEG size manual enable<br>Bit[5:4]: Debug mode<br>Bit[3]: Replace 0xFF to 0xFE in comment data<br>Bit[2]: Cut 0xD9 at the end of frame<br>Bit[1]: EOI generation enable<br>Bit[0]: ISI insert |
| 0x4409            | JPEG CTRL09    | 0x00          | RW  | Bit[7:0]: D9 data  |
| 0x440A            | JPEG CTRL0A    | 0x4E          | RW  | Bit[7:0]: JFIFO output delay   |
| 0x440B            | JPEG CTRL0B    | 0x16          | RW  | Bit[5]: Dummy read speed manual mode<br>Bit[4:0]: SFIFO SOF delay  |
| 0x440C            | JPEG CTRL0C    | 0x00          | RW  | Bit[7:0]: Dummy read speed   |
| 0x440D~<br>0x440F | DEBUG MODE     | –             | –   | Debug Mode   |
| 0x4410            | JPEG QT DATA   | 0x00          | RW  | Bit[7:0]: QT data  |
| 0x4411            | JPEG QT ADDR   | 0x00          | RW  | Bit[7:0]: QT address   |
| 0x4412            | JPEG ISI DATA  | 0x00          | RW  | Bit[7:0]: ISI data   |
| 0x4413            | JPEG ISI CTRL  | –             | RO  | Bit[7:4]: Debug mode<br>Bit[3]: D9 odd (read only)<br>Bit[2]: Reset counter write<br>0: Debug mode<br>1: Reset counter<br>Bit[1]: ISO EOF  |
| 0x4414            | JPEG LENGTH    | –             | RO  | Bit[7:0]: JPEG length[23:16]   |
| 0x4415            | JPEG LENGTH    | –             | RO  | Bit[7:0]: JPEG length[15:8]  |
| 0x4416            | JPEG LENGTH    | –             | RO  | Bit[7:0]: JPEG length[7:0]   |
| 0x4417            | JFIFO OVERFLOW | –             | RO  | Bit[7:1]: Debug mode<br>Bit[0]: JFIFO overflow indicator   |
| 0x4418~<br>0x4419 | DEBUG MODE     | –             | –   | Debug Mode   |
| 0x4420~<br>0x442F | JPEG COMMENT   | 0x00          | RW  | JPEG Comment Data Embedded in JPEG Data  |
| 0x4430            | JPEG COMMENT   | 0x00          | RW  | Comment Length<br>Two bytes align  |
| 0x4431            | JPEG COMMENT   | 0xFE          | RW  | Comment Data Marker  |

table 7-18 VFIFO registers

| address           | register name | default value | R/W | description   |
|-------------------|---------------|---------------|-----|---|
| 0x4600            | VFIFO CTRL00  | 0x80          | RW  | VFIFO Control 00<br>Bit[7:6]: Reserved<br>Bit[5]: Compression output fixed height enable<br>0: In compression mode2, compression height is different in each frame<br>1: In compression mode2, compression height is fixed in each frame  |
| 0x4601            | DEBUG MODE    | –             | –   | Debug Mode  |
| 0x4602            | VFIFO HSIZE   | 0x04          | RW  | Compression Output Width High Byte  |
| 0x4603            | VFIFO HSIZE   | 0x00          | RW  | Compression Output Width Low Byte   |
| 0x4604            | VFIFO VSIZE   | 0x03          | RW  | Compression Output Height High Byte   |
| 0x4605            | VFIFO HSIZE   | 0x00          | RW  | Compression Output Height Low Byte  |
| 0x4606~<br>0x460B | DEBUG MODE    | –             | –   | Debug Mode  |
| 0x460C            | VFIFO CTRL0C  | 0x20          | RW  | Bit[7:4]: JPEG dummy data pad speed<br>Bit[2]: Footer disable<br>JPEG footer disable<br>0: In jpg_mode2 footer will be added in the last six bytes of each frame<br>1: Disable footer<br>Bit[1]: PCLK manual enable<br>0: DVP PCLK divider control by auto mode<br>1: DVP PCLK divider control by 0x3824[4:0] |
| 0x460D            | VFIFO CTRL0D  | 0x00          | RW  | Dummy Data  |

table 7-19 DVP control registers (sheet 1 of 3)

| address           | register name    | default value | R/W | description  |
|-------------------|------------------|---------------|-----|--|
| 0x4709            | DVP VYSNC WIDTH0 | 0x02          | RW  | VSYNC Width Line Unit  |
| 0x470A            | DVP VYSNC WIDTH1 | 0x00          | RW  | Bit[7:0]: VSYNC width PCLK unit[15:8]  |
| 0x470B            | DVP VYSNC WIDTH2 | 0x01          | RW  | Bit[7:0]: VSYNC width PCLK unit[7:0]   |
| 0x470C~<br>0x4710 | DEBUG MODE       | –             | –   | Debug Mode   |
| 0x4711            | PAD LEFT CTRL    | 0x00          | RW  | HSYNC Mode Left Padding Pixel Counter<br>Add padding data at start of a line   |
| 0x4712            | PAD RIGHT CTRL   | 0x00          | RW  | HSYNC Mode Right Padding Pixel Counter<br>Add padding data at end of a line  |
| 0x4713            | JPG MODE SELECT  | 0x02          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: JPEG mode select<br>001: JPEG mode 1<br>010: JPEG mode 2<br>011: JPEG mode 3<br>100: JPEG mode 4<br>101: JPEG mode 5<br>110: JPEG mode 6   |
| 0x4715            | 656 DUMMY LINE   | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: CCIR656 dummy line number<br>Control dummy line number at beginning of the frame   |
| 0x4719            | CCIR656 CTRL     | 0x00          | RW  | Bit[7:2]: Debug mode<br>Bit[1:0]: CCIR656 EAV/SAV option   |
| 0x471B            | HSYNC CTRL00     | 0x02          | RW  | Bit[7:1]: Debug mode<br>Bit[0]: HSYNC mode enable  |
| 0x471D            | DVP VSYNC CTRL   | 0x01          | RW  | Bit[7:2]: Debug mode<br>Bit[1:0]: vsync_mode<br>00: VSYNC positive edge trigger by end of field, negative edge trigger by start of frame<br>01: VSYNC positive edge trigger by end of frame, the width define by register<br>10: VSYNC positive edge trigger by start of field, the width define by register |

table 7-19 DVP control registers (sheet 2 of 3)

| address | register name         | default value | R/W | description   |
|---------|-----------------------|---------------|-----|---|
| 0x471F  | DVP HREF CTRL         | 0x40          | RW  | HREF Minimum Blanking in JPEG Mode23  |
| 0x4721  | VERTICAL START OFFSET | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: Vertical start delay between video output and video input   |
| 0x4722  | VERTICAL END OFFSET   | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: Vertical end delay between video output and video input   |
| 0x4723  | DVP CTRL23            | 0x00          | RW  | DVP JPEG Mode456 Skip Line Number   |
| 0x4730  | CCIR656 CTRL00        | 0x00          | RW  | Bit[7]: SYNC code selection<br>0: Auto generate sync code<br>1: Sync code from register setting 0x4732~4735<br>Bit[6]: f value in CCIR656 SYNC code when fixed f value<br>Bit[5]: Fixed f value<br>Bit[4:3]: Blank toggle data options<br>00: Toggle data is 1'h040/1'h200<br>01: Use register setting 0x4736~0x4738<br>10: Blanking data always keep 0<br>Bit[1]: Clip data disable<br>Bit[0]: CCIR656 mode enable |
| 0x4731  | CCIR656 CTRL01        | 0x00          | RW  | Bit[7:1]: Debug mode<br>Bit[0]: Blanking toggle data order option   |
| 0x4732  | CCIR656 FS            | 0x01          | RW  | CCIR656 SYNC Code Frame Start   |
| 0x4733  | CCIR656 FE            | 0x0F          | RW  | CCIR656 SYNC Code Frame End   |
| 0x4734  | CCIR656 LS            | 0x00          | RW  | CCIR656 SYNC Code Line Start  |
| 0x4735  | CCIR656 LE            | 0x00          | RW  | CCIR656 SYNC Code Line End  |
| 0x4736  | CCIR656 CTRL6         | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:2]: Toggle data0[9:8]<br>Bit[1:0]: Toggle data1[9:8]  |
| 0x4737  | CCIR656 CTRL7         | 0x00          | RW  | Bit[7:0]: Toggle data0[7:0]   |
| 0x4738  | CCIR656 CTRL8         | 0x00          | RW  | Bit[7:0]: Toggle data1[7:0]   |

table 7-19 DVP control registers (sheet 3 of 3)

| address           | register name   | default value | R/W | description  |
|-------------------|-----------------|---------------|-----|--|
| 0x4740            | POLARITY CTRL00 | 0x20          | RW  | Bit[7:6]: Debug mode<br>Bit[5]: PCLK polarity<br>0: Active low<br>1: Active high<br>Bit[3]: Gate PCLK under VSYNC<br>Bit[2]: Gate PCLK under HREF<br>Bit[1]: HREF polarity<br>0: Active low<br>1: Active high<br>Bit[0]: VSYNC polarity<br>0: Active low<br>1: Active high |
| 0x4741            | TEST PATTERN    | 0x00          | RW  | Bit[7:3]: Debug mode<br>Bit[2]: Test pattern enable<br>Bit[1]: Test pattern select<br>0: Output test pattern 0<br>1: Output test pattern 1<br>Bit[0]: Test pattern 8-bit/10-bit<br>0: 10-bit test pattern<br>1: 8-bit test pattern   |
| 0x4742~<br>0x4744 | DEBUG MODE      | –             | –   | Debug Mode   |
| 0x4745            | DATA ORDER      | 0x00          | RW  | Bit[7:3]: Debug mode<br>Bit[2:1]: DVP order option for debug<br>00: Data[9:0]<br>10: {data[7:0],data[9:8]}<br>x1: {data[1:0],data[9:2]}<br>Bit[0]: Output data order<br>0: Normal output<br>1: Reverse output data bit order   |

table 7-20 MIPI transmitter registers (sheet 1 of 3)

| address       | register name | default value | R/W | description   |
|---------------|---------------|---------------|-----|---|
| 0x4800        | MIPI CTRL 00  | 0x04          | RW  | <p>MIPI Control 00</p> <p>Bit[7:6]: Debug mode</p> <p>Bit[5]: Clock lane gate enable<br/>                     0: Clock lane is free running<br/>                     1: Gate clock lane when no packet to transmit</p> <p>Bit[4]: Line sync enable<br/>                     0: Do not send line short packet for each line<br/>                     1: Send line short packet for each line</p> <p>Bit[3]: Lane select<br/>                     0: Use lane1 as default data lane<br/>                     1: Use lane2 as default data lane</p> <p>Bit[2]: Idle status<br/>                     0: MIPI bus will be LP00 when no packet to transmit<br/>                     1: MIPI bus will be LP11 when no packet to transmit</p> <p>Bit[1:0]: Debug mode</p> |
| 0x4801        | MIPI CTRL 01  | 0x04          | RW  | <p>MIPI Control 01</p> <p>Bit[7:5]: Debug mode</p> <p>Bit[4]: PH bit order for ECC<br/>                     0: {DI[7:0],WC[7:0],WC[15:8]}<br/>                     1: {DI[0:7],WC[0:7],WC[8:15]}</p> <p>Bit[3]: PH byte order for ECC<br/>                     0: {DI,WC_l,WC_h}<br/>                     1: {DI,WC_h,WC_l}</p> <p>Bit[2]: PH byte order2 for ECC<br/>                     0: {DI,WC}<br/>                     1: {WC,DI}</p> <p>Bit[1:0]: Debug mode</p>   |
| 0x4802~0x4804 | DEBUG MODE    | -             | -   | Debug Mode  |
| 0x4805        | MIPI CTRL 05  | 0x10          | RW  | <p>Bit[7]: MIPI lane1 disable<br/>                     1: Disable MIPI data lane1<br/>                     Lane1 will be LP00</p> <p>Bit[6]: MIPI lane1 disable<br/>                     1: Disable MIPI data lane1<br/>                     Lane1 will be LP00</p> <p>Bit[5]: LPX Global timing select<br/>                     0: Auto calculate t_lpx_o in pclk2x domain, unit clk2x<br/>                     1: Use lp_p_min[7:0]</p> <p>Bit[6:0]: Debug mode</p>   |
| 0x4806~0x4809 | DEBUG MODE    | -             | -   | Debug Mode  |

table 7-20 MIPI transmitter registers (sheet 2 of 3)

| address           | register name          | default value | R/W | description  |
|-------------------|------------------------|---------------|-----|--|
| 0x480A            | MIPI DATA ORDER        | 0x00          | RW  | Bit[7:3]: Debug mode<br>Bit[2]: Bit order reverse<br>Bit[1:0]: Bit position adjust<br>01: {data[7:0],data[9:8]}<br>10: {data[1:0],data[9:2]} |
| 0x480B~<br>0x4817 | DEBUG MODE             | –             | –   | Debug Mode   |
| 0x4818            | MIN HS ZERO H          | 0x00          | RW  | High Byte of Minimum Value of hs_zero<br>Unit ns   |
| 0x4819            | MIN HS ZERO L          | 0x96          | RW  | Low Byte of Minimum Value of hs_zero<br>$hs\_zero\_real = hs\_zero\_min\_o + Tui*ui\_hs\_zero\_min\_o$                                       |
| 0x481A            | MIN MIPI HS TRAIL H    | 0x00          | RW  | High Byte of Minimum Value of hs_trail<br>Unit ns  |
| 0x481B            | MIN MIPI HS TRAIL L    | 0x3C          | RW  | Low Byte of Minimum Value of hs_trail<br>$hs\_trail\_real = hs\_trail\_min\_o + Tui*ui\_hs\_trail\_min\_o$                                   |
| 0x481C            | MIN MIPI CLK ZERO H    | 0x01          | RW  | High Byte of Minimum Value of clk_zero   |
| 0x481D            | MIN MIPI CLK ZERO L    | 0x86          | RW  | Low Byte of Minimum Value of clk_zero<br>$clk\_zero\_real = clk\_zero\_min\_o + Tui*ui\_clk\_zero\_min\_o$                                   |
| 0x481E            | MIN MIPI CLK PREPARE H | 0x00          | RW  | High Byte of Minimum Value of clk_prepare<br>Unit ns   |
| 0x481F            | MIN MIPI CLK PREPARE L | 0x3C          | RW  | Low Byte of Minimum Value of clk_prepare<br>$clk\_prepare\_real = clk\_prepare\_min\_o + Tui*ui\_clk\_prepare\_min\_o$                       |
| 0x4820            | MIN CLK POST H         | 0x00          | RW  | High Byte of Minimum Value of clk_post<br>Unit ns  |
| 0x4821            | MIN CLK POST L         | 0x56          | RW  | Low Byte of Minimum Value of clk_post<br>$clk\_post\_real = clk\_post\_min\_o + Tui*ui\_clk\_post\_min\_o$                                   |
| 0x4822            | MIN CLK TRAIL H        | 0x00          | RW  | High Byte of Minimum Value of clk_trail<br>Unit ns   |
| 0x4823            | MIN CLK TRAIL L        | 0x3C          | RW  | Low Byte of Minimum Value of clk_trail<br>$clk\_trail\_real = clk\_trail\_min\_o + Tui*ui\_clk\_trail\_min\_o$                               |
| 0x4824            | MIN LPX PCLK H         | 0x00          | RW  | High Byte of Minimum Value of lpx_p<br>Unit ns   |
| 0x4825            | MIN LPX PCLK L         | 0x32          | RW  | Low Byte of Minimum Value of lpx_p<br>$lpx\_p\_real = lpx\_p\_min\_o + Tui*ui\_lpx\_p\_min\_o$   |
| 0x4826            | MIN HS PREPARE H       | 0x00          | RW  | High Byte of Minimum Value of hs_prepare<br>Unit ns  |

table 7-20 MIPI transmitter registers (sheet 3 of 3)

| address           | register name      | default value | R/W | description  |
|-------------------|--------------------|---------------|-----|--|
| 0x4827            | MIN HS PREPARE L   | 0x32          | RW  | Low Byte of Minimum Value of hs_prepare<br>$hs\_prepare\_real = hs\_prepare\_min\_o + Tui * ui\_hs\_prepare\_min\_o$ |
| 0x4828            | MIN HS EXIT H      | 0x00          | RW  | High Byte of Minimum Value of hs_exit<br>Unit ns   |
| 0x4829            | MIN HS EXIT L      | 0x64          | RW  | Low Byte of Minimum Value of hs_exit<br>$hs\_exit\_real = hs\_exit\_min\_o + Tui * ui\_hs\_exit\_min\_o$             |
| 0x482A            | MIN HS ZERO/UI     | 0x05          | RW  | Minimum UI Value of hs_zero<br>Unit UI   |
| 0x482B            | MIN HS TRAIL/UI    | 0x04          | RW  | Minimum UI Value of hs_trail<br>Unit UI  |
| 0x482C            | MIN CLK ZERO/UI    | 0x00          | RW  | Minimum UI Value of clk_zero<br>Unit UI  |
| 0x482D            | MIN CLK PREPARE/UI | 0x00          | RW  | Minimum UI Value of clk_prepare<br>Unit UI   |
| 0x482E            | MIN CLK POST/UI    | 0x34          | RW  | Minimum UI Value of clk_post<br>Unit UI  |
| 0x482F            | MIN CLK TRAIL/UI   | 0x00          | RW  | Minimum UI Value of clk_trail<br>Unit UI   |
| 0x4830            | MIN LPX PCLK/UI    | 0x00          | RW  | Minimum UI Value of lpx_p(pclk2x domain)<br>Unit UI  |
| 0x4831            | MIN HS PREPARE/UI  | 0x04          | RW  | Minimum UI Value of hs_prepare<br>Unit UI  |
| 0x4832            | MIN HS EXIT/UI     | 0x00          | RW  | Minimum UI Value of hs_exit<br>Unit UI   |
| 0x4833~<br>0x4836 | DEBUG MODE         | –             | –   | Debug Mode   |
| 0x4837            | PCLK PERIOD        | 0x10          | RW  | Period of pixel clock, pclk_div=1, and 1-bit decimal   |



table 7-21 ISP frame control registers

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x4901  | FRAME CTRL01  | 0x00          | RW  | Control Passed Frame Number<br>When both ON and OFF number set to 0x00, frame control is in bypass mode<br>Bit[7:4]: Debug mode<br>Bit[3:0]: Frame ON number  |
| 0x4902  | FRAME CTRL02  | 0x00          | RW  | Control Masked Frame Number<br>When both ON and OFF number set to 0x00, frame control is in bypass mode<br>Bit[7:4]: Debug mode<br>Bit[3:0]: Frame OFF number |

table 7-22 ISP top control registers (sheet 1 of 5)

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x5000  | ISP CONTROL 00 | 0x06          | RW  | Bit[7]: LENC correction enable<br>0: Disable<br>1: Enable<br>Bit[6]: Debug mode<br>Bit[5]: RAW GMA enable<br>0: Disable<br>1: Enable<br>Bit[4:3]: Debug mode<br>Bit[2]: Black pixel cancellation enable<br>0: Disable<br>1: Enable<br>Bit[1]: White pixel cancellation enable<br>0: Disable<br>1: Enable<br>Bit[0]: Color interpolation enable<br>0: Disable<br>1: Enable |

table 7-22 ISP top control registers (sheet 2 of 5)

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x5001  | ISP CONTROL 01 | 01            | RW  | Bit[7]: Special digital effect enable<br>0: Disable<br>1: Enable<br>Bit[6]: Debug mode<br>Bit[5]: Scale enable<br>0: Disable<br>1: Enable<br>Bit[4:3]: Debug mode<br>Bit[2]: UV average enable<br>0: Disable<br>1: Enable<br>Bit[1]: Color matrix enable<br>0: Disable<br>1: Enable<br>Bit[0]: Auto white balance enable<br>0: Disable<br>1: Enable |
| 0x5002  | DEBUG MODE     | –             | –   | Debug Mode  |
| 0x5003  | ISP CONTROL 03 | 08            | RW  | Bit[7:3]: Debug mode<br>Bit[2]: Bin enable<br>0: Disable<br>1: Enable<br>Bit[1]: Draw window for AFC enable<br>0: Disable<br>1: Enable<br>Bit[0]: Solarize enable<br>0: Disable<br>1: Enable  |
| 0x5004  | DEBUG MODE     | –             | –   | Debug Mode  |

table 7-22 ISP top control registers (sheet 3 of 5)

| address           | register name      | default value | R/W | description  |
|-------------------|--------------------|---------------|-----|--|
| 0x5005            | ISP CONTROL 05     | 0x36          | RW  | Bit[7]: Debug mode<br>Bit[6]: AWB bias manual enable<br>0: Disable<br>1: Enable<br>Bit[5]: AWB bias ON enable<br>0: Disable<br>1: Enable<br>Bit[4]: AWB bias plus enable<br>0: Disable<br>1: Enable<br>Bit[3]: Debug mode<br>Bit[2]: LENC bias ON enable<br>0: Disable<br>1: Enable<br>Bit[1]: GMA bias ON enable<br>0: Disable<br>1: Enable<br>Bit[0]: LENC bias manual enable<br>0: Disable<br>1: Enable |
| 0x5006~<br>0x501C | DEBUG MODE         | –             | –   | Debug Mode   |
| 0x501D            | ISP MISC           | 0x00          | RW  | Bit[7]: Debug mode<br>Bit[6]: SDE AVG manual enable<br>Bit[5]: AWB YUV2CBCR enable<br>Bit[4]: Average size manual enable<br>Bit[3:0]: Debug mode   |
| 0x501E            | ISP MISC           | 0x00          | RW  | Bit[7]: Debug mode<br>Bit[6]: Scale ratio manual enable<br>Bit[5:0]: Debug mode  |
| 0x501F            | FORMAT MUX CONTROL | 0x00          | RW  | Format MUX Control<br>Bit[7:4]: Debug mode<br>Bit[3]: Fmt vfirst<br>Bit[2:0]: Format select<br>000: ISP YUV422<br>001: ISP RGB<br>010: ISP dither<br>011: ISP RAW (DPC)<br>100: SNR RAW<br>101: ISP RAW (CIP)  |
| 0x5020            | DITHER CTRL 0      | 0x00          | RW  | Bit[7]: Debug mode<br>Bit[6]: Dither MUX<br>Bit[5:4]: R dithering<br>Bit[3:2]: G dithering<br>Bit[1:0]: B dithering  |

table 7-22 ISP top control registers (sheet 4 of 5)

| address           | register name                                 | default value | R/W | description  |
|-------------------|---|---------------|-----|--|
| 0x5021~<br>0x5026 | DEBUG MODE                                    | –             | –   | Debug Mode   |
| 0x5027            | DRAW WINDOW CONTROL 00                        | 0x02          | RW  | Bit[7:1]: Debug mode<br>Bit[0]: Draw window control<br>0: No fixed Y<br>1: Fixed Y |
| 0x5028            | DRAW WINDOW LEFT POSITION CONTROL             | 0x04          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: Draw window left[11:8]                           |
| 0x5029            | DRAW WINDOW LEFT POSITION CONTROL             | 0x90          | RW  | Bit[7:0]: Draw window left[7:0]  |
| 0x502A            | DRAW WINDOW RIGHT POSITION CONTROL            | 0x05          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: Draw window right[11:8]                          |
| 0x502B            | DRAW WINDOW RIGHT POSITION CONTROL            | 0x90          | RW  | Bit[7:0]: Draw window right[7:0]   |
| 0x502C            | DRAW WINDOW TOP POSITION CONTROL              | 0x03          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: Draw window top[10:8]                            |
| 0x502D            | DRAW WINDOW TOP POSITION CONTROL              | 0x6C          | RW  | Bit[7:0]: Draw window top[7:0]   |
| 0x502E            | DRAW WINDOW BOTTOM POSITION CONTROL           | 0x04          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: Draw window bottom[10:8]                         |
| 0x502F            | DRAW WINDOW BOTTOM POSITION CONTROL           | 0x2C          | RW  | Bit[7:0]: Draw window bottom[7:0]  |
| 0x5030            | DRAW WINDOW HORIZONTAL BOUNDARY WIDTH CONTROL | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: Draw window horizontal boundary width[11:8]      |
| 0x5031            | DRAW WINDOW HORIZONTAL BOUNDARY WIDTH CONTROL | 0x14          | RW  | Bit[7:0]: Draw window horizontal boundary width[7:0]                               |
| 0x5032            | DRAW WINDOW VERTICAL BOUNDARY WIDTH CONTROL   | 0x00          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: Draw window vertical boundary width[10:8]        |

table 7-22 ISP top control registers (sheet 5 of 5)

| address           | register name                               | default value | R/W | description  |
|-------------------|---|---------------|-----|--|
| 0x5033            | DRAW WINDOW VERTICAL BOUNDARY WIDTH CONTROL | 0x14          | RW  | Bit[7:0]: Draw window vertical boundary width[7:0]   |
| 0x5034            | DRAW WINDOW Y CONTROL                       | 0x80          | RW  | Bit[7:0]: Fixed Y for draw window  |
| 0x5035            | DRAW WINDOW U CONTROL                       | 0x2A          | RW  | Bit[7:0]: Fixed U for draw window  |
| 0x5036            | DRAW WINDOW V CONTROL                       | 0x14          | RW  | Bit[7:0]: Fixed V for draw window  |
| 0x5037~<br>0x503C | DEBUG MODE                                  | –             | –   | Debug Mode   |
| 0x503D            | PRE ISP TEST SETTING 1                      | 0x00          | RW  | Bit[7]: Pre ISP test enable<br>0: Test disable<br>1: Color bar enable<br>Bit[6]: Rolling<br>Bit[5]: Transparent<br>Bit[4]: Square BW<br>Bit[3:2]: Pre ISP bar style<br>00: Standard 8 color bar<br>01: Gradual change at vertical mode 1<br>10: Gradual change at horizontal<br>11: Gradual change at vertical mode 2<br>Bit[1:0]: Test select<br>00: Color bar<br>01: Random data<br>10: Square data<br>11: Black image |
| 0x503E~<br>0x5060 | DEBUG MODE                                  | –             | –   | Debug Mode   |
| 0x5061            | ISP SENSOR BIAS I                           | –             | R   | ISP Sensor Bias I  |
| 0x5062            | ISP SENSOR GAIN I                           | –             | R   | ISP Sensor Gain I  |
| 0x5063            | ISP SENSOR GAIN I                           | –             | R   | ISP Sensor Gain I  |

table 7-23 AWB registers (sheet 1 of 2)

| address           | register name  | default value | R/W | description   |
|-------------------|----------------|---------------|-----|---|
| 0x5180            | AWB CONTROL 00 | 0xFF          | RW  | Bit[7:0]: AWB B block   |
| 0x5181            | AWB CONTROL 01 | 0x58          | RW  | Bit[7:6]: Step local<br>Bit[5:4]: Step fast<br>Bit[3]: Slop 8x<br>Bit[2]: Slop 4x<br>Bit[1]: One zone<br>Bit[0]: AVG all  |
| 0x5182            | AWB CONTROL 02 | 0x11          | RW  | Bit[7:4]: Max local counter<br>Bit[3:0]: Max fast counter   |
| 0x5183            | AWB CONTROL 03 | 0x90          | RW  | Bit[7]: AWB simple enable<br>0: AWB advance<br>1: AWB simple<br>Bit[6]: AWB advance<br>0: YUV enable<br>1: Simple YUV enable<br>Bit[5]: AWB preset<br>Bit[4]: AWB SIMF<br>Bit[3:2]: AWB win<br>Bit[0]: Debug mode |
| 0x5184            | AWB CONTROL 04 | 0x25          | RW  | Bit[7:6]: Count area selection<br>Bit[5]: G enable<br>Bit[4:2]: Count limit control<br>Bit[1:0]: Counter threshold  |
| 0x5185            | AWB CONTROL 05 | 0x24          | RW  | Bit[7:4]: Stable range unstable<br>Threshold for unstable to stable change<br>Bit[3:0]: Stable range stable<br>Threshold for stable to unstable change  |
| 0x5186~<br>0x5190 | AWB CONTROL    | –             | –   | Advanced AWB Control Registers  |
| 0x5191            | AWB CONTROL 17 | 0xFF          | RW  | Bit[7:0]: AWB top limit   |
| 0x5192            | AWB CONTROL 18 | 0x00          | RW  | Bit[7:0]: AWB bottom limit  |
| 0x5193            | AWB CONTROL 19 | 0xF0          | RW  | Bit[7:0]: Red limit   |
| 0x5194            | AWB CONTROL 20 | 0xF0          | RW  | Bit[7:0]: Green limit   |
| 0x5195            | AWB CONTROL 21 | 0xF0          | RW  | Bit[7:0]: Blue limit  |

table 7-23 AWB registers (sheet 2 of 2)

| address           | register name         | default value | R/W | description  |
|-------------------|-----------------------|---------------|-----|--|
| 0x5196            | AWB CONTROL 22        | 0x03          | RW  | Bit[7:6]: Debug mode<br>Bit[5]: AWB freeze<br>Bit[4]: Debug mode<br>Bit[3:2]: AWB simple selection<br>00: AWB simple from after AWB gain<br>01: AWB simple from after RAW GMA<br>10: AWB simple from after RAW GMA<br>11: AWB simple from after AWB gain<br>Bit[1]: Fast enable<br>Bit[0]: AWB bias stat |
| 0x5197            | AWB CONTROL 23        | 0x02          | RW  | Bit[7:0]: Local limit  |
| 0x5198~<br>0x519D | DEBUG MODE            | –             | –   | Debug Mode   |
| 0x519E            | AWB CONTROL 30        | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3]: Local limit select<br>Bit[2]: Simple stable select<br>Bit[1:0]: Debug mode   |
| 0x519F            | AWB CURRENT<br>R GAIN | –             | R   | Bit[7:4]: Debug mode<br>Bit[3:0]: Current R setting[11:8]  |
| 0x51A0            | AWB CURRENT<br>R GAIN | –             | R   | Bit[7:0]: Current R setting[7:0]   |
| 0x51A1            | AWB CURRENT<br>G GAIN | –             | R   | Bit[7:4]: Debug mode<br>Bit[3:0]: Current G setting[11:8]  |
| 0x51A2            | AWB CURRENT<br>G GAIN | –             | R   | Bit[7:0]: Current G setting[7:0]   |
| 0x51A3            | AWB CURRENT<br>B GAIN | –             | R   | Bit[7:4]: Debug mode<br>Bit[3:0]: Current B setting[11:8]  |
| 0x51A4            | AWB CURRENT<br>B GAIN | –             | R   | Bit[7:0]: Current B setting[7:0]   |
| 0x51A5            | AWB AVERAGE B         | –             | R   | Bit[7:0]: Average R[9:2]   |
| 0x51A6            | AWB AVERAGE B         | –             | R   | Bit[7:0]: Average G[9:2]   |
| 0x51A7            | AWB AVERAGE B         | –             | R   | Bit[7:0]: Average B[9:2]   |
| 0x51D0            | AWB CONTROL74         | –             | R   | Bit[7:6]: Debug mode<br>Bit[5]: R large<br>Bit[4]: G large<br>Bit[3]: B large<br>Bit[2:1]: Current type<br>Bit[0]: Debug mode  |

table 7-24 CIP control registers

| address | register name              | default value | R/W | description  |
|---------|----------------------------|---------------|-----|--|
| 0x5300  | CIP SHARPENMT THRESHOLD 1  | 0x08          | RW  | Color Interpolation Sharpen MT Threshold 1   |
| 0x5301  | CIP SHARPENMT THRESHOLD 2  | 0x48          | RW  | Color Interpolation Sharpen MT Threshold 2   |
| 0x5302  | CIP SHARPENMT OFFSET1      | 0x18          | RW  | CIP Sharpen MT Offset1 (Y edge mt manual setting when 0x5308[6]=1)   |
| 0x5303  | CIP SHARPENMT OFFSET2      | 0x0E          | RW  | CIP Sharpen MT Offset2   |
| 0x5304  | CIP DNS THRESHOLD 1        | 0x08          | RW  | CIP DNS Threshold 1  |
| 0x5305  | CIP DNS THRESHOLD 2        | 0x48          | RW  | CIP DNS Threshold 2  |
| 0x5306  | CIP DNS OFFSET1            | 0x09          | RW  | CIP DNS Offset1 (DNS threshold manual setting when 0x5308[4]=1)  |
| 0x5307  | CIP DNS OFFSET2            | 0x16          | RW  | CIP DNS Offset2  |
| 0x5308  | CIP CTRL                   | 0x25          | RW  | Bit[7]: Debug mode<br>Bit[6]: CIP edge MT manual enable<br>Bit[4]: CIP DNS manual enable<br>Bit[2:0]: CIP threshold for BR sharpen |
| 0x5309  | CIP SHARPENTH THRESHOLD 1  | 0x08          | RW  | CIP Sharpen TH Threshold 1   |
| 0x530A  | CIP SHARPENTH THRESHOLD 2  | 0x48          | RW  | CIP Sharpen TH Threshold 2   |
| 0x530B  | CIP SHARPENTH OFFSET1      | 0x04          | RW  | CIP Sharpen TH Offset1 (Sharpen threshold manual setting when 0x5308[6]=1)   |
| 0x530C  | CIP SHARPENTH OFFSET2      | 0x06          | RW  | CIP Sharpen TH Offset2   |
| 0x530D  | CIP EDGE MT AUTO           | –             | R   | CIP Edge MT Auto Read  |
| 0x530E  | CIP DNS THRESHOLD AUTO     | –             | R   | CIP DNS Threshold Auto Read  |
| 0x530F  | CIP SHARPEN THRESHOLD AUTO | –             | R   | CIP Sharpen Threshold Auto Read  |



table 7-25 CMX control registers

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5381  | CMX1          | 0x20          | RW  | Bit[7:2]: Debug mode<br>Bit[1]: Color matrix enable<br>0: Disable<br>1: Enable<br>Bit[0]: Debug mode  |
| 0x5382  | CMX2          | 0x64          | RW  | Bit[7:0]: CMX2 for Y  |
| 0x5383  | CMX3          | 0x08          | RW  | Bit[7:0]: CMX3 for Y  |
| 0x5384  | CMX4          | 0x30          | RW  | Bit[7:0]: CMX4 for U  |
| 0x5385  | CMX5          | 0x90          | RW  | Bit[7:0]: CMX5 for U  |
| 0x5386  | CMX6          | 0xC0          | RW  | Bit[7:0]: CMX6 for U  |
| 0x5387  | CMX7          | 0xA0          | RW  | Bit[7:0]: CMX7 for V  |
| 0x5388  | CMX8          | 0x98          | RW  | Bit[7:0]: CMX8 for V  |
| 0x5389  | CMX9          | 0x08          | RW  | Bit[7:0]: CMX9 for V  |
| 0x538A  | CMXSIGN       | 0x01          | RW  | Cmxsign<br>Bit[7:1]: Debug mode<br>Bit[0]: CMX9 sign  |
| 0x538B  | CMXSIGN       | 0x98          | RW  | Cmxsign<br>Bit[7]: CMX8 sign<br>Bit[6]: CMX7 sign<br>Bit[5]: CMX6 sign<br>Bit[4]: CMX5 sign<br>Bit[3]: CMX4 sign<br>Bit[2]: CMX3 sign<br>Bit[1]: CMX2 sign<br>Bit[0]: CMX1 sign |

table 7-26 gamma control registers

| address | register name   | default value | R/W | description  |
|---------|-----------------|---------------|-----|--|
| 0x5480  | GAMMA CONTROL00 | 0x00          | RW  | Bit[7:2]: Debug mode<br>Bit[1]: YSLP15 manual enable<br>Bit[0]: BIAS plus on |
| 0x5481  | GAMMA YST00     | 0x26          | RW  | Bit[7:0]: Y yst 00   |
| 0x5482  | GAMMA YST01     | 0x35          | RW  | Bit[7:0]: Y yst 01   |
| 0x5483  | GAMMA YST02     | 0x48          | RW  | Bit[7:0]: Y yst 02   |
| 0x5484  | GAMMA YST03     | 0x57          | RW  | Bit[7:0]: Y yst 03   |
| 0x5485  | GAMMA YST04     | 0x63          | RW  | Bit[7:0]: Y yst 04   |
| 0x5486  | GAMMA YST05     | 0x6E          | RW  | Bit[7:0]: Y yst 05   |
| 0x5487  | GAMMA YST06     | 0x77          | RW  | Bit[7:0]: Y yst 06   |
| 0x5488  | GAMMA YST07     | 0x80          | RW  | Bit[7:0]: Y yst 07   |
| 0x5489  | GAMMA YST08     | 0x88          | RW  | Bit[7:0]: Y yst 08   |
| 0x548A  | GAMMA YST09     | 0x96          | RW  | Bit[7:0]: Y yst 09   |
| 0x548B  | GAMMA YST0A     | 0xA3          | RW  | Bit[7:0]: Y yst 0A   |
| 0x548C  | GAMMA YST0B     | 0xAF          | RW  | Bit[7:0]: Y yst 0B   |
| 0x548D  | GAMMA YST0C     | 0xC5          | RW  | Bit[7:0]: Y yst 0C   |
| 0x548E  | GAMMA YST0D     | 0xD7          | RW  | Bit[7:0]: Y yst 0D   |
| 0x548F  | GAMMA YST0E     | 0xE8          | RW  | Bit[7:0]: Y yst 0E   |
| 0x5490  | GAMMA YST0F     | 0x0F          | RW  | Bit[7:0]: Y yst 0F   |

table 7-27 SDE control registers (sheet 1 of 2)

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x5580  | SDE CTRL0     | 0x00          | RW  | Bit[7]: Fixed Y enable<br>0: Disable<br>1: Enable<br>Bit[6]: Negative enable<br>0: Disable<br>1: Enable<br>Bit[5]: Gray enable<br>0: Disable<br>1: Enable<br>Bit[4]: Fixed V enable<br>0: Disable<br>1: Enable<br>Bit[3]: Fixed U enable<br>0: Disable<br>1: Enable<br>Bit[2]: Contrast enable<br>0: Disable<br>1: Enable<br>Bit[1]: Saturation enable<br>0: Disable<br>1: Enable<br>Bit[0]: Hue enable<br>0: Disable<br>1: Enable |
| 0x5581  | SDE CTRL1     | 0x80          | RW  | Bit[7:0]: Hue cos coefficient  |
| 0x5582  | SDE CTRL2     | 0x00          | RW  | Bit[7:0]: Hue sin coefficient  |
| 0x5583  | SDE CTRL3     | 0x40          | RW  | Bit[7:0]: Saturation U when 0x5580[1]=1 and 0x5588[6]=1, max value for UV adjust when 0x5580[1]=1 and 0x5588[6]=0 or fixed U when 0x5580[3]=1  |
| 0x5584  | SDE CTRL4     | 0x40          | RW  | Bit[7:0]: Saturation V when 0x5580[1]=1 and 0x5588[6]=1, min value for UV adjust when 0x5580[1]=1 and 0x5588[6]=0 or Vreg when 0x5580[4]=1   |
| 0x5585  | SDE CTRL5     | 0x00          | RW  | Bit[7:0]: Y offset for contrast when 0x5044[3]=1; or fixed Y when 0x5580[7]=1  |
| 0x5586  | SDE CTRL6     | 0x20          | RW  | Bit[7:0]: Y gain for contrast  |
| 0x5587  | SDE CTRL7     | 0x00          | RW  | Bit[7:0]: Y bright for contrast  |

table 7-27 SDE control registers (sheet 2 of 2)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5588  | SDE CTRL8     | 0x01          | RW  | Bit[7]: Debug mode<br>Bit[5]: Sign5 for hue V, cos<br>Bit[4]: Sign4 for hue U, cos<br>Bit[3]: Sign3 Y bright sign for contrast<br>0: Keep Y bright sign<br>1: Negative Y bright sign<br>Bit[2]: Sign2<br>Y offset sign for contrast when<br>0x5044[3]=1<br>0: Keep Y offset sign<br>1: Negative Y offset sign<br>Bit[1]: Sign1 for hue V, sin<br>Bit[0]: Sign0 for hue U, sin |
| 0x5589  | SDE CTRL9     | 0x01          | RW  | Bit[7:0]: UV adjust threshold 1<br>Valid when 0x5580[1]=1   |
| 0x558A  | SDE CTRL10    | 0x01          | RW  | Bit[7:1]: Debug mode<br>Bit[0]: UV adjust threshold 2[8]<br>Valid when 0x5580[1]=1  |
| 0x558B  | SDE CTRL11    | 0xFF          | RW  | Bit[7:0]: UV adjust threshold 2[7:0]<br>Valid when 0x5580[1]=1  |
| 0x558C  | SDE CTRL12    | –             | R   | Bit[7:0]: UV adjust value read out  |

table 7-28 SCALE registers

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5600  | SCALE CTRL 0  | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5]: UV drop<br>YUV444 to 422 drop mode versus AVG mode selection<br>0: AVG mode<br>1: Drop mode<br>Bit[4]: Debug mode<br>Bit[3]: Hround<br>DCW hrounding<br>0: No horizontal rounding<br>1: Horizontal rounding<br>Bit[2]: Hdrop<br>DCW drop mode<br>0: Horizontal average mode<br>1: Horizontal drop mode<br>Bit[1]: Vround<br>DCW vrounding<br>0: No vertical rounding<br>1: Vertical rounding<br>Bit[0]: Vdrop<br>DCW drop mode<br>0: Vertical average mode<br>1: Vertical drop mode |
| 0x5601  | SCALE CTRL 1  | 0x00          | RW  | Bit[7]: Debug mode<br>Bit[6:4]: HDIV RW<br>DCW scale times<br>000: DCW 1 time<br>001: DCW 2 time<br>010: DCW 4 time<br>011: DCW 8 time<br>1xx: DCW 16 time<br>Bit[2:0]: VDIV RW<br>DCW scale times<br>000: DCW 1 time<br>001: DCW 2 time<br>010: DCW 4 time<br>011: DCW 8 time<br>1xx: DCW 16 time  |
| 0x5602  | SCALE CTRL 2  | 0x02          | RW  | Bit[7:0]: XSC[15:8]   |
| 0x5603  | SCALE CTRL 3  | 0x00          | RW  | Bit[7:0]: XSC[7:0]  |
| 0x5604  | SCALE CTRL 4  | 0x02          | RW  | Bit[7:0]: YSC[15:8]   |
| 0x5605  | SCALE CTRL 5  | 0x00          | RW  | Bit[7:0]: YSC[7:0]  |
| 0x5606  | SCALE CTRL 6  | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: Voffset   |

table 7-29 AVG registers (sheet 1 of 2)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5680  | X START       | 0x00          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: x start[11:8]<br>Horizontal start position for average window high byte, valid when 0x501D[4]=1 |
| 0x5681  | X START       | 0x00          | RW  | Bit[7:0]: x start[7:0]<br>Horizontal start position for average window low byte, valid when 0x501D[4]=1                           |
| 0x5682  | Y START       | 0x00          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: y start[10:8]<br>Vertical start position for average window low byte, valid when 0x501D[4]=1    |
| 0x5683  | Y START       | 0x00          | RW  | Bit[7:0]: y start[7:0]<br>Vertical start position for average window low byte, valid when 0x501D[4]=1                             |
| 0x5684  | X WINDOW      | 0x10          | RW  | Bit[7:4]: Debug mode<br>Bit[3:0]: Window X[11:8]<br>Horizontal end position for average window high byte, valid when 0x501D[4]=1  |
| 0x5685  | X WINDOW      | 0xA0          | RW  | Bit[7:0]: Window X[7:0]<br>Horizontal end position for average window low byte, valid when 0x501D[4]=1                            |
| 0x5686  | Y WINDOW      | 0x0C          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: Window Y[10:8]<br>Vertical end position for average window high byte, valid when 0x501D[4]=1    |
| 0x5687  | Y WINDOW      | 0x78          | RW  | Bit[7:0]: Window Y[7:0]<br>Vertical end position for average window low byte, valid when 0x501D[4]=1                              |
| 0x5688  | WEIGHT00      | 0x11          | RW  | Bit[7:4]: Window 01 weight<br>Bit[3:0]: Window 00 weight  |
| 0x5689  | WEIGHT01      | 0x11          | RW  | Bit[7:4]: Window 03 weight<br>Bit[3:0]: Window 02 weight  |
| 0x568A  | WEIGHT02      | 0x11          | RW  | Bit[7:4]: Window 11 weight<br>Bit[3:0]: Window 10 weight  |
| 0x568B  | WEIGHT03      | 0x11          | RW  | Bit[7:4]: Window 13 weight<br>Bit[3:0]: Window 12 weight  |
| 0x568C  | WEIGHT04      | 0x11          | RW  | Bit[7:4]: Window 21 weight<br>Bit[3:0]: Window 20 weight  |
| 0x568D  | WEIGHT05      | 0x11          | RW  | Bit[7:4]: Window 23 weight<br>Bit[3:0]: Window 22 weight  |

table 7-29 AVG registers (sheet 2 of 2)

| address | register name  | default value | R/W | description   |
|---------|----------------|---------------|-----|---|
| 0x568E  | WEIGHT06       | 0x11          | RW  | Bit[7:4]: Window 31 weight<br>Bit[3:0]: Window 30 weight  |
| 0x568F  | WEIGHT07       | 0x11          | RW  | Bit[7:4]: Window 33 weight<br>Bit[3:0]: Window 32 weight  |
| 0x5690  | AVG CTRL10     | 0x01          | RW  | Bit[7:1]: Debug mode<br>Bit[0]: AVG option<br>0: $\text{Sum}=(4*B+9*G*2+10*R)/8$<br>1: $\text{Sum}=(B+G*2+R)$ |
| 0x5691  | AVG WIN 00     | –             | R   | Bit[7:0]: Average of win 00   |
| 0x5692  | AVG WIN 01     | –             | R   | Bit[7:0]: Average of win 01   |
| 0x5693  | AVG WIN 02     | –             | R   | Bit[7:0]: Average of win 02   |
| 0x5694  | AVG WIN 03     | –             | R   | Bit[7:0]: Average of win 03   |
| 0x5695  | AVG WIN 10     | –             | R   | Bit[7:0]: Average of win 10   |
| 0x5696  | AVG WIN 11     | –             | R   | Bit[7:0]: Average of win 11   |
| 0x5697  | AVG WIN 12     | –             | R   | Bit[7:0]: Average of win 12   |
| 0x5698  | AVG WIN 13     | –             | R   | Bit[7:0]: Average of win 13   |
| 0x5699  | AVG WIN 20     | –             | R   | Bit[7:0]: Average of win 20   |
| 0x569A  | AVG WIN 21     | –             | R   | Bit[7:0]: Average of win 21   |
| 0x569B  | AVG WIN 22     | –             | R   | Bit[7:0]: Average of win 22   |
| 0x569C  | AVG WIN 23     | –             | R   | Bit[7:0]: Average of win 23   |
| 0x569D  | AVG WIN 30     | –             | R   | Bit[7:0]: Average of win 30   |
| 0x569E  | AVG WIN 31     | –             | R   | Bit[7:0]: Average of win 31   |
| 0x569F  | AVG WIN 32     | –             | R   | Bit[7:0]: Average of win 32   |
| 0x56A0  | AVG WIN 33     | –             | R   | Bit[7:0]: Average of win 33   |
| 0x56A1  | AVG READOUT    | –             | R   | Bit[7:0]: High 8 bits of average value  |
| 0x56A2  | AVG WEIGHT SUM | –             | R   | Bit[7:0]: Average weight sum  |

table 7-30 LENC control registers (sheet 1 of 5)

| address | register name | default value | R/W | description                                       |
|---------|---------------|---------------|-----|---|
| 0x5800  | GMTRX00       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 00 |
| 0x5801  | GMTRX01       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 01 |
| 0x5802  | GMTRX02       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 02 |
| 0x5803  | GMTRX03       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 03 |
| 0x5804  | GMTRX04       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 04 |
| 0x5805  | GMTRX05       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 05 |
| 0x5806  | GMTRX10       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 06 |
| 0x5807  | GMTRX11       | 0x08          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 07 |
| 0x5808  | GMTRX12       | 0x08          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 08 |
| 0x5809  | GMTRX13       | 0x08          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 09 |
| 0x580A  | GMTRX14       | 0x08          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 0A |
| 0x580B  | GMTRX15       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 0B |
| 0x580C  | GMTRX20       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 0C |
| 0x580D  | GMTRX21       | 0x08          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 0D |
| 0x580E  | GMTRX22       | 0x00          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 0E |
| 0x580F  | GMTRX23       | 0x00          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 0F |
| 0x5810  | GMTRX24       | 0x08          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 10 |
| 0x5811  | GMTRX25       | 0x10          | RW  | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 11 |



table 7-30 LENC control registers (sheet 2 of 5)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5812  | GMTRX30       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 12   |
| 0x5813  | GMTRX31       | 0x08          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 13   |
| 0x5814  | GMTRX32       | 0x00          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 14   |
| 0x5815  | GMTRX33       | 0x00          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 15   |
| 0x5816  | GMTRX34       | 0x08          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 16   |
| 0x5817  | GMTRX35       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 17   |
| 0x5818  | GMTRX40       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 18   |
| 0x5819  | GMTRX41       | 0x08          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 19   |
| 0x581A  | GMTRX42       | 0x08          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 1A   |
| 0x581B  | GMTRX43       | 0x08          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 1B   |
| 0x581C  | GMTRX44       | 0x08          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 1C   |
| 0x581E  | GMTRX45       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 1D   |
| 0x581E  | GMTRX50       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 1E   |
| 0x581F  | GMTRX51       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 1F   |
| 0x5820  | GMTRX52       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 20   |
| 0x5821  | GMTRX53       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 21   |
| 0x5822  | GMTRX54       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 22   |
| 0x5823  | GMTRX55       | 0x10          | R/W | Bit[7:6]: Debug mode<br>Bit[5:0]: Green matrix 23   |
| 0x5824  | BRMATRIX00    | 0xAA          | R/W | Bit[7:4]: Blue matrix 00<br>Bit[3:0]: Red matrix 00 |

table 7-30 LENC control registers (sheet 3 of 5)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5825  | BRMATRX01     | 0xAA          | RW  | Bit[7:4]: Blue matrix 01<br>Bit[3:0]: Red matrix 01 |
| 0x5826  | BRMATRX02     | 0xAA          | RW  | Bit[7:4]: Blue matrix 02<br>Bit[3:0]: Red matrix 02 |
| 0x5827  | BRMATRX03     | 0xAA          | RW  | Bit[7:4]: Blue matrix 03<br>Bit[3:0]: Red matrix 03 |
| 0x5828  | BRMATRX04     | 0xAA          | RW  | Bit[7:4]: Blue matrix 04<br>Bit[3:0]: Red matrix 04 |
| 0x5829  | BRMATRX05     | 0xAA          | RW  | Bit[7:4]: Blue matrix 05<br>Bit[3:0]: Red matrix 05 |
| 0x582A  | BRMATRX06     | 0x99          | RW  | Bit[7:4]: Blue matrix 06<br>Bit[3:0]: Red matrix 06 |
| 0x582B  | BRMATRX07     | 0x99          | RW  | Bit[7:4]: Blue matrix 07<br>Bit[3:0]: Red matrix 07 |
| 0x582C  | BRMATRX08     | 0x99          | RW  | Bit[7:4]: Blue matrix 08<br>Bit[3:0]: Red matrix 08 |
| 0x582D  | BRMATRX09     | 0xAA          | RW  | Bit[7:4]: Blue matrix 09<br>Bit[3:0]: Red matrix 09 |
| 0x582E  | BRMATRX20     | 0xAA          | RW  | Bit[7:4]: Blue matrix 20<br>Bit[3:0]: Red matrix 20 |
| 0x582F  | BRMATRX21     | 0x99          | RW  | Bit[7:4]: Blue matrix 21<br>Bit[3:0]: Red matrix 21 |
| 0x5830  | BRMATRX22     | 0x88          | RW  | Bit[7:4]: Blue matrix 22<br>Bit[3:0]: Red matrix 22 |
| 0x5831  | BRMATRX23     | 0x99          | RW  | Bit[7:4]: Blue matrix 23<br>Bit[3:0]: Red matrix 23 |
| 0x5832  | BRMATRX24     | 0xAA          | RW  | Bit[7:4]: Blue matrix 24<br>Bit[3:0]: Red matrix 24 |
| 0x5833  | BRMATRX30     | 0xAA          | RW  | Bit[7:4]: Blue matrix 30<br>Bit[3:0]: Red matrix 30 |
| 0x5834  | BRMATRX31     | 0x99          | RW  | Bit[7:4]: Blue matrix 31<br>Bit[3:0]: Red matrix 31 |
| 0x5835  | BRMATRX32     | 0x99          | RW  | Bit[7:4]: Blue matrix 32<br>Bit[3:0]: Red matrix 32 |
| 0x5836  | BRMATRX33     | 0x99          | RW  | Bit[7:4]: Blue matrix 33<br>Bit[3:0]: Red matrix 33 |
| 0x5837  | BRMATRX34     | 0xAA          | RW  | Bit[7:4]: Blue matrix 34<br>Bit[3:0]: Red matrix 34 |

table 7-30 LENC control registers (sheet 4 of 5)

| address | register name  | default value | R/W | description  |
|---------|----------------|---------------|-----|--|
| 0x5838  | BRMATRIX40     | 0xAA          | R/W | Bit[7:4]: Blue matrix 40<br>Bit[3:0]: Red matrix 40  |
| 0x5839  | BRMATRIX41     | 0xAA          | R/W | Bit[7:4]: Blue matrix 41<br>Bit[3:0]: Red matrix 41  |
| 0x583A  | BRMATRIX42     | 0xAA          | R/W | Bit[7:4]: Blue matrix 42<br>Bit[3:0]: Red matrix 42  |
| 0x583B  | BRMATRIX43     | 0xAA          | R/W | Bit[7:4]: Blue matrix 43<br>Bit[3:0]: Red matrix 43  |
| 0x583C  | BRMATRIX44     | 0xAA          | R/W | Bit[7:4]: Blue matrix 44<br>Bit[3:0]: Red matrix 44  |
| 0x583D  | LENC BR OFFSET | 0x88          | R/W | Bit[7:4]: LENC b offset<br>Bit[3:0]: LENC r offset   |
| 0x583E  | MAX GAIN       | 0x40          | R/W | Bit[7:0]: Maximum gain   |
| 0x583F  | MIN GAIN       | 0x20          | R/W | Bit[7:0]: Minimum gain   |
| 0x5840  | MIN Q          | 0x18          | R/W | Bit[7]: Debug mode<br>Bit[6:0]: Minimum Q  |
| 0x5841  | LENC CTRL59    | 0x0D          | R/W | Bit[7:4]: Debug mode<br>Bit[3]: Add BLC enable<br>0: Disable BLC add back function<br>1: Enable BLC add back function<br>Bit[2]: BLC enable<br>0: Disable BLC function<br>1: Enable BLC function<br>Bit[1]: Gain manual enable<br>Bit[0]: Auto Q enable<br>0: Used constant Q (0x40)<br>1: Used calculated Q |
| 0x5842  | BR HSCALE      | 0x01          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: br h scale[10:8]<br>Reciprocal of horizontal step for BR channel. BR channel in whole image is divided into 5x5 blocks. The step is used to point to the border of the adjacent block  |
| 0x5843  | BR HSCAL       | 0x2B          | RW  | Bit[7:0]: br h scale[7:0]<br>Reciprocal of horizontal step for BR channel. BR channel in whole image is divided into 5x5 blocks. The step is used to point to the border of the adjacent block   |

table 7-30 LENC control registers (sheet 5 of 5)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x5844  | BR VSCALE     | 0x01          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: br v scale[10:8]<br>Reciprocal of vertical step for BR channel. BR channel in whole image is divided into 5x5 blocks. The step is used to point to the border of the adjacent block |
| 0x5845  | BR VSCALE     | 0x8D          | RW  | Bit[7:0]: br v scale[7:0]<br>Reciprocal of vertical step for BR channel. BR channel in whole image is divided into 5x5 blocks. The step is used to point to the border of the adjacent block                          |
| 0x5846  | G HSCALE      | 0x01          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: g h scale[10:8]<br>Reciprocal of horizontal step for G channel. G channel in whole image is divided into 6x6 blocks. The step is used to point to the border of the adjacent block  |
| 0x5847  | G HSCAL       | 0x8F          | RW  | Bit[7:0]: g h scale[7:0]<br>Reciprocal of horizontal step for G channel. G channel in whole image is divided into 6x6 blocks. The step is used to point to the border of the adjacent block                           |
| 0x5848  | G VSCALE      | 0x01          | RW  | Bit[7:3]: Debug mode<br>Bit[2:0]: g v scale[10:8]<br>Reciprocal of vertical step for G channel. G channel in whole image is divided into 6x6 blocks. The step is used to point to the border of the adjacent block    |
| 0x5849  | G VSCALE      | 0x09          | RW  | Bit[7:0]: g v scale[7:0]<br>Reciprocal of vertical step for G channel. G channel in whole image is divided into 6x6 blocks. The step is used to point to the border of the adjacent block                             |

table 7-31 AFC control registers (sheet 1 of 5)

| address | register name | default value | R/W | description  |
|---------|---------------|---------------|-----|--|
| 0x6000  | AFC CTRL00    | 0xFF          | RW  | AFC Control 00<br>Bit[7:5]: Debug mode<br>Bit[4]: Edge filter enable<br>0: afc_edge module will not update<br>1: afc_edge module will update<br>Bit[3]: Edge filter b select<br>Edge b filters select<br>0: Edge selects b2<br>1: Edge selects b1<br>Bit[2:0]: Edge filter a select<br>Edge a filters select<br>001: Edge selects a1<br>010: Edge selects a2<br>100: Edge selects a4<br>101: Edge selects a5 |
| 0x6001  | AFC CTRL01    | 0xFF          | RW  | AFC Control 01<br>Bit[7:0]: Edge window0 left<br>This window coordinate<br>Others: Edge selects a1   |
| 0x6002  | AFC CTRL02    | 0xFF          | RW  | AFC Control 02<br>Bit[7:0]: Edge window0 top<br>This window coordinate   |
| 0x6003  | AFC CTRL03    | 0xFF          | RW  | AFC Control 03<br>Bit[7:0]: Edge window0 right<br>This window coordinate   |
| 0x6004  | AFC CTRL04    | 0xFF          | RW  | AFC Control 04<br>Bit[7:0]: Edge window0 bottom<br>This window coordinate<br>This bottom must be larger than any other   |
| 0x6005  | AFC CTRL05    | 0xFF          | RW  | AFC Control 05<br>Bit[7:0]: Edge window1 left<br>This window coordinate  |
| 0x6006  | AFC CTRL06    | 0xFF          | RW  | AFC Control 06<br>Bit[7:0]: Edge window1 top<br>This window coordinate   |
| 0x6007  | AFC CTRL07    | 0xFF          | RW  | AFC Control 07<br>Bit[7:0]: Edge window1 right<br>This window coordinate   |
| 0x6008  | AFC CTRL08    | 0xFF          | RW  | AFC Control 08<br>Bit[7:0]: Edge window1 bottom<br>This window coordinate  |

table 7-31 AFC control registers (sheet 2 of 5)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x6009  | AFC CTRL09    | 0xFF          | RW  | AFC Control 09<br>Bit[7:0]: Edge window2 left<br>This window coordinate     |
| 0x600A  | AFC CTRL10    | 0xFF          | RW  | AFC Control 10<br>Bit[7:0]: Edge window2 top<br>This window coordinate      |
| 0x600B  | AFC CTRL11    | 0xFF          | RW  | AFC Control 11<br>Bit[7:0]: Edge window2 right<br>This window coordinate    |
| 0x600C  | AFC CTRL12    | 0xFF          | RW  | AFC Control 12<br>Bit[7:0]: Edge window2 bottom<br>This window coordinate   |
| 0x600D  | AFC CTRL13    | 0xFF          | RW  | AFC Control 13<br>Bit[7:0]: Edge window3 left<br>This window coordinate     |
| 0x600E  | AFC CTRL14    | 0xFF          | RW  | AFC Control 14<br>Bit[7:0]: edge window3 top<br>This window coordinate      |
| 0x600F  | AFC CTRL15    | 0xFF          | RW  | AFC Control 15<br>Bit[7:0]: Edge window3 right<br>This window coordinate    |
| 0x6010  | AFC CTRL16    | 0xFF          | RW  | AFC Control 16<br>Bit[7:0]: Edge window3 bottom<br>This window coordinate   |
| 0x6011  | AFC CTRL17    | 0xFF          | RW  | AFC Control 17<br>Bit[7:0]: Edge window4 left<br>This window coordinate     |
| 0x6012  | AFC CTRL18    | 0xFF          | RW  | AFC Control 18<br>Bit[7:0]: Edge window4 top<br>This window coordinate      |
| 0x6013  | AFC CTRL19    | 0xFF          | RW  | AFC Control 19<br>Bit[7:0]: Edge window4 right<br>This window coordinate    |
| 0x6014  | AFC CTRL20    | 0xFF          | RW  | AFC Control 20<br>Bit[7:0]: Edge window4 bottom<br>This window coordinate   |
| 0x6015  | AFC CTRL21    | –             | R   | AFC Control 21<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window0 filter a[29:24] |
| 0x6016  | AFC CTRL22    | –             | R   | AFC Control 22<br>Bit[7:0]: Window0 filter a[23:16]                         |

table 7-31 AFC control registers (sheet 3 of 5)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x6017  | AFC CTRL23    | –             | R   | AFC Control 23<br>Bit[7:0]: Window0 filter a[15:8]                          |
| 0x6018  | AFC CTRL24    | –             | R   | AFC Control 24<br>Bit[7:0]: Window0 filter a[7:0]                           |
| 0x6019  | AFC CTRL25    | –             | R   | AFC Control 25<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window0 filter b[29:24] |
| 0x601A  | AFC CTRL26    | –             | R   | AFC Control 26<br>Bit[7:0]: Window0 filter b[23:16]                         |
| 0x601B  | AFC CTRL27    | –             | R   | AFC Control 27<br>Bit[7:0]: Window0 filter b[15:8]                          |
| 0x601C  | AFC CTRL28    | –             | R   | AFC Control 28<br>Bit[7:0]: Window0 filter b[7:0]                           |
| 0x601D  | AFC CTRL29    | –             | R   | AFC Control 29<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window1 filter a[29:24] |
| 0x601E  | AFC CTRL30    | –             | R   | AFC Control 30<br>Bit[7:0]: Window1 filter a[23:16]                         |
| 0x601F  | AFC CTRL31    | –             | R   | AFC Control 31<br>Bit[7:0]: Window1 filter a[15:8]                          |
| 0x6020  | AFC CTRL32    | –             | R   | AFC Control 32<br>Bit[7:0]: Window1 filter a[7:0]                           |
| 0x6021  | AFC CTRL33    | –             | R   | AFC Control 33<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window1 filter b[29:24] |
| 0x6022  | AFC CTRL34    | –             | R   | AFC Control 34<br>Bit[7:0]: Window1 filter b[23:16]                         |
| 0x6023  | AFC CTRL35    | –             | R   | AFC Control 35<br>Bit[7:0]: Window1 filter b[15:8]                          |
| 0x6024  | AFC CTRL36    | –             | R   | AFC Control 36<br>Bit[7:0]: Window1 filter b[7:0]                           |
| 0x6025  | AFC CTRL37    | –             | R   | AFC Control 37<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window2 filter a[29:24] |
| 0x6026  | AFC CTRL38    | –             | R   | AFC Control 38<br>Bit[7:0]: Window2 filter a[23:16]                         |
| 0x6027  | AFC CTRL39    | –             | R   | AFC Control 39<br>Bit[7:0]: Window2 filter a[15:8]                          |

table 7-31 AFC control registers (sheet 4 of 5)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x6028  | AFC CTRL40    | –             | R   | AFC Control 40<br>Bit[7:0]: Window2 filter a[7:0]                           |
| 0x6029  | AFC CTRL41    | –             | R   | AFC Control 41<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window2 filter b[29:24] |
| 0x602A  | AFC CTRL42    | –             | R   | AFC Control 42<br>Bit[7:0]: Window2 filter b[23:16]                         |
| 0x602B  | AFC CTRL43    | –             | R   | AFC Control 43<br>Bit[7:0]: Window2 filter b[15:8]                          |
| 0x602C  | AFC CTRL44    | –             | R   | AFC Control 44<br>Bit[7:0]: Window2 filter b[7:0]                           |
| 0x602D  | AFC CTRL45    | –             | R   | AFC Control 45<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window3 filter a[29:24] |
| 0x602E  | AFC CTRL46    | –             | R   | AFC Control 46<br>Bit[7:0]: Window3 filter a[23:16]                         |
| 0x602F  | AFC CTRL47    | –             | R   | AFC Control 47<br>Bit[7:0]: Window3 filter a[15:8]                          |
| 0x6030  | AFC CTRL48    | –             | R   | AFC Control 48<br>Bit[7:0]: Window3 filter a[7:0]                           |
| 0x6031  | AFC CTRL49    | –             | R   | AFC Control 49<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window3 filter b[29:24] |
| 0x6032  | AFC CTRL50    | –             | R   | AFC Control 50<br>Bit[7:0]: Window3 filter b[23:16]                         |
| 0x6033  | AFC CTRL51    | –             | R   | AFC Control 51<br>Bit[7:0]: Window3 filter b[15:8]                          |
| 0x6034  | AFC CTRL52    | –             | R   | AFC Control 52<br>Bit[7:0]: Window3 filter b[7:0]                           |
| 0x6035  | AFC CTRL53    | –             | R   | AFC Control 53<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window4 filter a[29:24] |
| 0x6036  | AFC CTRL54    | –             | R   | AFC Control 54<br>Bit[7:0]: Window4 filter a[23:16]                         |
| 0x6037  | AFC CTRL55    | –             | R   | AFC Control 55<br>Bit[7:0]: Window4 filter a[15:8]                          |
| 0x6038  | AFC CTRL56    | –             | R   | AFC Control 56<br>Bit[7:0]: Window4 filter a[7:0]                           |



table 7-31 AFC control registers (sheet 5 of 5)

| address | register name | default value | R/W | description   |
|---------|---------------|---------------|-----|---|
| 0x6039  | AFC CTRL57    | –             | R   | AFC Control 57<br>Bit[7:6]: Debug mode<br>Bit[5:0]: Window4 filter b[29:24] |
| 0x603A  | AFC CTRL58    | –             | R   | AFC Control 58<br>Bit[7:0]: Window4 filter b[23:16]                         |
| 0x603B  | AFC CTRL59    | –             | R   | AFC Control 59<br>Bit[7:0]: Window4 filter b[15:8]                          |
| 0x603C  | AFC CTRL60    | –             | R   | AFC Control 60<br>Bit[7:0]: Window4 filter b[7:0]                           |
| 0x603D  | AFC READ58    | –             | R   | AFC READ 58<br>Bit[7:0]: Window4 filter b[23:16]                            |
| 0x603R  | AFC READ59    | –             | R   | AFC READ 59<br>Bit[7:0]: Window4 filter b[15:8]                             |
| 0x603F  | AFC READ60    | –             | R   | AFC READ 60<br>Bit[7:0]: Window4 filter b[7:0]                              |

**OV5640**

color CMOS QSXGA (5 megapixel) image sensor with OmniBSI™ technology

Confidential for  
xin jin hui

## 8 operating specifications

### 8.1 absolute maximum ratings

**table 8-1** absolute maximum ratings

| parameter  | absolute maximum rating <sup>a</sup> |
|--|--------------------------------------|
| ambient storage temperature                          | -40°C to +95°C                       |
| supply voltage (with respect to ground) <sup>b</sup> | $V_{DD-A}$ 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.
- b. for negative voltage with respect to ground,  $V_{DD-A}$  (-4.5V),  $V_{DD-C}$  (-3V),  $V_{DD-IO}$  (-4.5V)

### 8.2 functional temperature

**table 8-2** functional temperature

| parameter                                   | range          |
|---|----------------|
| operating temperature range <sup>a</sup>    | -20°C to +70°C |
| stable image temperature range <sup>b</sup> | 0°C to +50°C   |

- 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 ( $-20^{\circ}\text{C} < T_A < 70^{\circ}\text{C}$ ) (sheet 1 of 2)

| symbol                    | parameter                            | min   | typ | max   | unit          |
|---------------------------|--------------------------------------|-------|-----|-------|---------------|
| power supply              |                                      |       |     |       |               |
| $V_{DD-A}$                | supply voltage (analog)              | 2.6   | 2.8 | 3.0   | V             |
| $V_{DD-D}^a$              | supply voltage (digital core)        | 1.425 | 1.5 | 1.575 | V             |
| $V_{DD-IO}$               | supply voltage (digital I/O)         | 1.71  | 1.8 | 3.0   | V             |
| internal DVDD, DOVDD=1.8V |                                      |       |     |       |               |
| $I_{DD-A}$                | active (operating) current           | TBD   | TBD | TBD   | mA            |
| $I_{DD-IO}^{b,c}$         |                                      | TBD   | TBD | TBD   | mA            |
| $I_{DDS-SCCB}^d$          | standby current                      | TBD   | TBD | TBD   | $\mu\text{A}$ |
| $I_{DDS-PWDN}^d$          |                                      | TBD   | TBD | TBD   | $\mu\text{A}$ |
| $P_O$                     | active (operating) power consumption | TBD   | TBD | TBD   | mW            |
| $P_{DDS-SCCB}$            | standby power consumption            | TBD   | TBD | TBD   | $\mu\text{W}$ |
| $P_{DDS-PWDN}$            |                                      | TBD   | TBD | TBD   | $\mu\text{W}$ |
| external DVDD, DOVDD=2.8V |                                      |       |     |       |               |
| $I_{DD-A}$                | active (operating) current           | TBD   | TBD | TBD   | mA            |
| $I_{DD-D}^{b,c}$          |                                      | TBD   | TBD | TBD   | mA            |
| $I_{DD-IO}$               |                                      | TBD   | TBD | TBD   | mA            |
| $I_{DDS-SCCB}$            | standby current                      | TBD   | TBD | TBD   | $\mu\text{A}$ |
| $I_{DDS-PWDN}$            |                                      | TBD   | TBD | TBD   | $\mu\text{A}$ |
| $P_O$                     | active (operating) power consumption | TBD   | TBD | TBD   | mW            |
| $P_{DDS-SCCB}$            | standby power consumption            | TBD   | TBD | TBD   | $\mu\text{W}$ |
| $P_{DDS-PWDN}$            |                                      | TBD   | TBD | TBD   | $\mu\text{W}$ |
| external DVDD, DOVDD=1.8V |                                      |       |     |       |               |
| $I_{DD-A}$                | active (operating) current           | TBD   | TBD | TBD   | mA            |
| $I_{DD-D}^{b,c}$          |                                      | TBD   | TBD | TBD   | mA            |
| $I_{DD-IO}$               |                                      | TBD   | TBD | TBD   | mA            |
| $I_{DDS-SCCB}$            | standby current                      | TBD   | TBD | TBD   | $\mu\text{A}$ |
| $I_{DDS-PWDN}$            |                                      | TBD   | TBD | TBD   | $\mu\text{A}$ |
| $P_O$                     | active (operating) power consumption | TBD   | TBD | TBD   | mW            |
| $P_{DDS-SCCB}$            | standby power consumption            | TBD   | TBD | TBD   | $\mu\text{W}$ |
| $P_{DDS-PWDN}$            |                                      | TBD   | TBD | TBD   | $\mu\text{W}$ |

**table 8-3** DC characteristics ( $-20^{\circ}\text{C} < T_A < 70^{\circ}\text{C}$ ) (sheet 2 of 2)

| symbol  | parameter           | min  | typ | max  | unit |
|---|---------------------|------|-----|------|------|
| digital inputs (typical conditions: AVDD = 2.8V, DVDD = 1.5V, DOVDD = 1.8V) |                     |      |     |      |      |
| V <sub>IL</sub>   | input voltage LOW   |      |     | 0.54 | V    |
| V <sub>IH</sub>   | input voltage HIGH  | 1.26 |     |      | V    |
| C <sub>IN</sub>   | input capacitor     |      |     | 10   | pF   |
| digital outputs (standard loading 25 pF)                                    |                     |      |     |      |      |
| V <sub>OH</sub>   | output voltage HIGH | 1.62 |     |      | V    |
| V <sub>OL</sub>   | output voltage LOW  |      |     | 0.18 | V    |
| serial interface inputs <sup>e</sup>  |                     |      |     |      |      |
| V <sub>IL</sub>   | SIOC and SIOD       | -0.5 | 0   | 0.54 | V    |
| V <sub>IH</sub>   | SIOC and SIOD       | 1.26 | 1.8 | 3.0  | V    |

- using the internal DVDD regulator is strongly recommended for minimum power down current
- active current is based on sensor resolution at full size and at full speed in compression format. For smaller sizes such as 720p or below preview, the total active current will be about half.
- DOVDD active current is based on loading of 10pF and typical compression format output PCLK (48MHz). For YUV output with higher PCLK, or higher loading, DOVDD current can go up.
- at room temperature and typical supply voltages
- based on DOVDD = 1.8V.

## 8.4 AC characteristics

**table 8-4** AC characteristics ( $T_A = 25^\circ\text{C}$ ,  $V_{DD-A} = 2.8\text{V}$ )

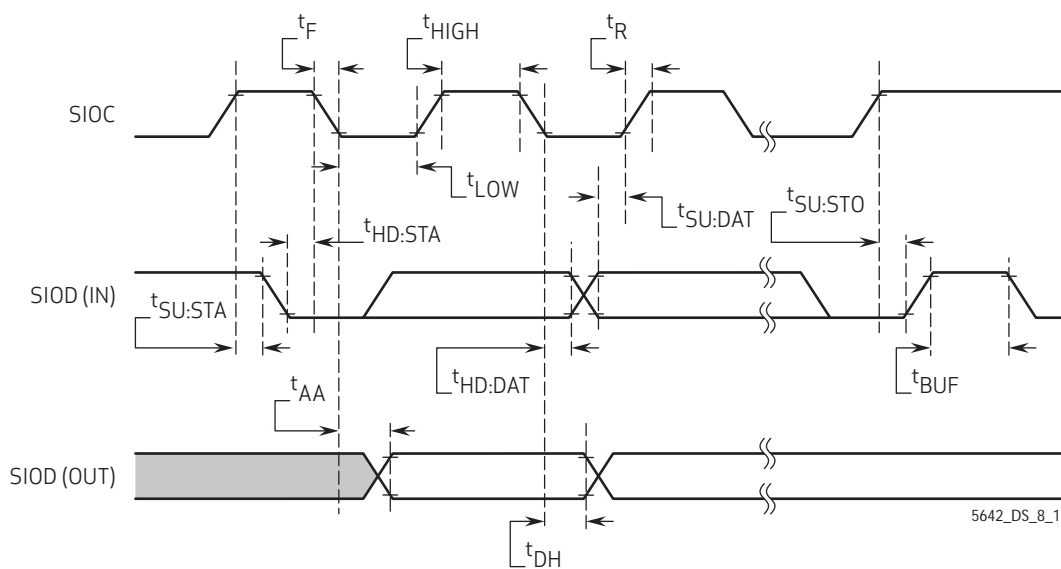
| symbol         | parameter                                | min | typ | max  | unit |
|----------------|--|-----|-----|------|------|
| ADC parameters |  |     |     |      |      |
| B              | analog bandwidth                         |     | 30  |      | MHz  |
| DLE            | DC differential linearity error          |     | 0.5 |      | LSB  |
| IIE            | 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 |   |     |                 |                      |      |
| $f_{\text{OSC}}$           | frequency (XVCLK) <sup>a</sup>          | 6   | 24              | 54                   | MHz  |
| $t_r, t_f$                 | clock input rise/fall time <sup>b</sup> |     |                 | 5 (10 <sup>c</sup> ) | ns   |
| $f_{\text{PCLK}}$          | parallel port output pixel clock        |     | 48 <sup>d</sup> | 96 <sup>e</sup>      | MHz  |

- for input clock range 6~27MHz, the OV5640 can tolerate input clock jitter up to 1ns, for input clock range to 54MHz, the OV5640 can tolerate input clock jitter up to 500ps
- if the PLL is bypassed, the delay from input clock to output clock is approximately 4~5ns
- if using the internal PLL
- typical PCLK is 48 MHz when sensor output is smaller size (VGA YUV or below) or full size compression
- 96 MHz is for sensor RAW data output at 15fps or YUV output at 7.5fps. For higher speeds such as 5 megapixel YUV @ 15fps, OmniVision recommends using the MIPI two-lane interface.

figure 8-1 SCCB interface timing

table 8-6 SCCB interface timing specifications<sup>a</sup>

| symbol       | parameter                      | min  | typ | max              | unit    |
|--------------|--------------------------------|------|-----|------------------|---------|
| $f_{SIOC}$   | clock frequency                |      |     | 400 <sup>b</sup> | KHz     |
| $t_{LOW}$    | clock low period               | 1.3  |     |                  | $\mu$ s |
| $t_{HIGH}$   | clock high period              | 0.6  |     |                  | $\mu$ s |
| $t_{AA}$     | SIOC low to data out valid     | 0.1  |     | 0.9              | $\mu$ s |
| $t_{BUF}$    | bus free time before new start | 1.3  |     |                  | $\mu$ s |
| $t_{HD:STA}$ | start condition hold time      | 0.6  |     |                  | $\mu$ s |
| $t_{SU:STA}$ | start condition setup time     | 1.85 |     |                  | $\mu$ s |
| $t_{HD:DAT}$ | data in hold time              | 0    |     |                  | $\mu$ s |
| $t_{SU:DAT}$ | data in setup time             | 0.1  |     |                  | $\mu$ s |
| $t_{SU:STO}$ | stop condition setup time      | 0.6  |     |                  | $\mu$ s |
| $t_R, t_F$   | SCCB rise/fall times           |      |     | 0.3              | $\mu$ s |
| $t_{DH}$     | data out hold time             | 0.05 |     |                  | $\mu$ s |

a. SCCB timing is based on 400KHz mode

b. SCCB maximum speed is 400KHz when sensor master input clock (XVCLK) is greater than or equal to 13MHz. When XVCLK is less than 13MHz, the maximum SCCB speed is less than 400KHz (approximately  $XVCLK/32.5$ )

**OV5640**

color CMOS QSXGA (5 megapixel) image sensor with OmniBSI™ technology

Confidential for  
xin jin hui



## 9 mechanical specifications

### 9.1 physical specifications

figure 9-1 package specifications

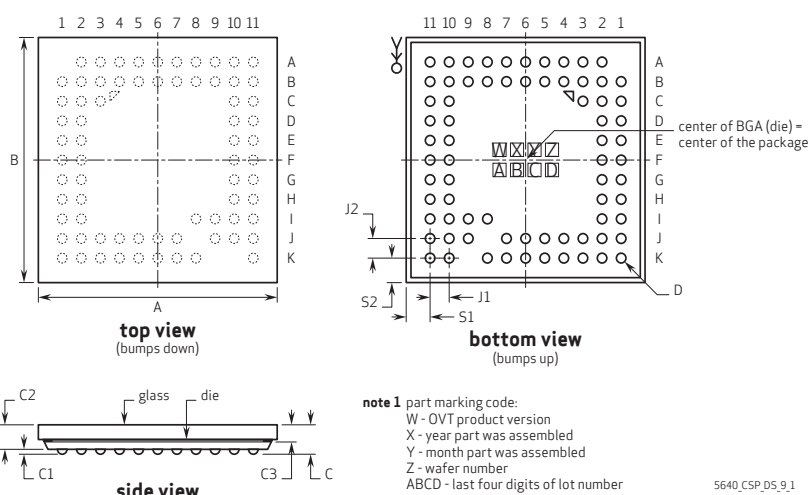
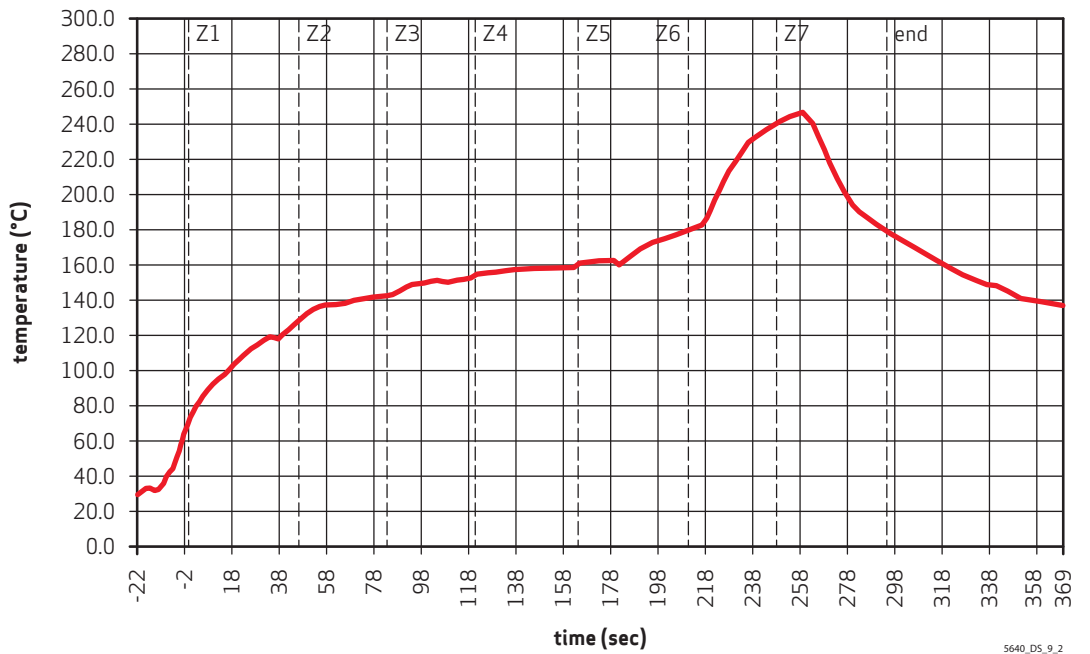


table 9-1 package dimensions

| parameter                            | symbol | min  | typ        | max  | unit          |
|--------------------------------------|--------|------|------------|------|---------------|
| package body dimension x             | A      | 5960 | 5985       | 6010 | $\mu\text{m}$ |
| package body dimension y             | B      | 5810 | 5835       | 5860 | $\mu\text{m}$ |
| package height                       | C      | 690  | 750        | 810  | $\mu\text{m}$ |
| ball height                          | C1     | 100  | 130        | 160  | $\mu\text{m}$ |
| package body thickness               | C2     | 575  | 620        | 665  | $\mu\text{m}$ |
| cover glass thickness                | C3     | 425  | 445        | 465  | $\mu\text{m}$ |
| ball diameter                        | D      | 220  | 250        | 280  | $\mu\text{m}$ |
| total pin count                      | N      |      | 71 (13 NC) |      |               |
| pin count x-axis                     | N1     |      | 11         |      |               |
| pin count y-axis                     | N2     |      | 11         |      |               |
| pins pitch x-axis                    | J1     |      | 500        |      | $\mu\text{m}$ |
| pins pitch y-axis                    | J2     |      | 500        |      | $\mu\text{m}$ |
| edge-to-pin center distance analog x | S1     | 463  | 493        | 523  | $\mu\text{m}$ |
| edge-to-pin center distance analog y | S2     | 388  | 418        | 448  | $\mu\text{m}$ |

## 9.2 IR reflow specifications

figure 9-2 IR reflow ramp rate requirements



**note**

The OV5640 uses a lead-free package.

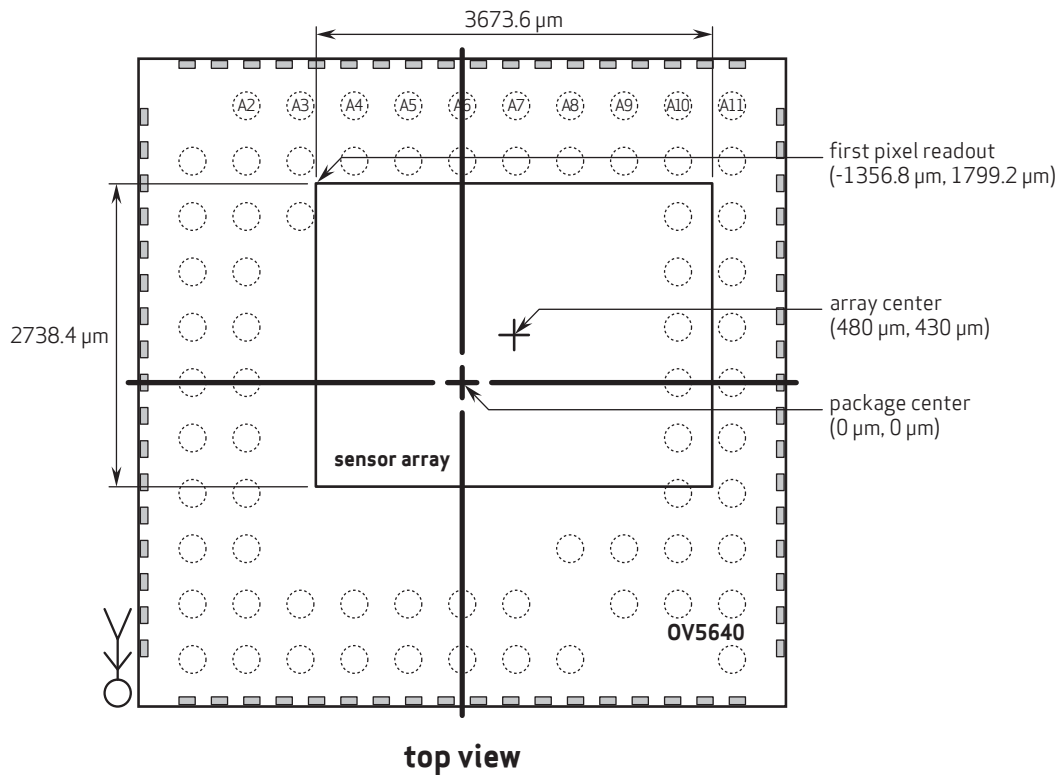
table 9-2 reflow conditions

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

## 10 optical specifications

### 10.1 sensor array center

figure 10-1 sensor array center



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

**note 2** as most optical assemblies invert and mirror the image, the chip is typically mounted with pins A2 to A11 oriented down on the PCB.

5640\_CSP\_DS\_10\_1

### 10.2 lens chief ray angle (CRA)

figure 10-2 chief ray angle (CRA)

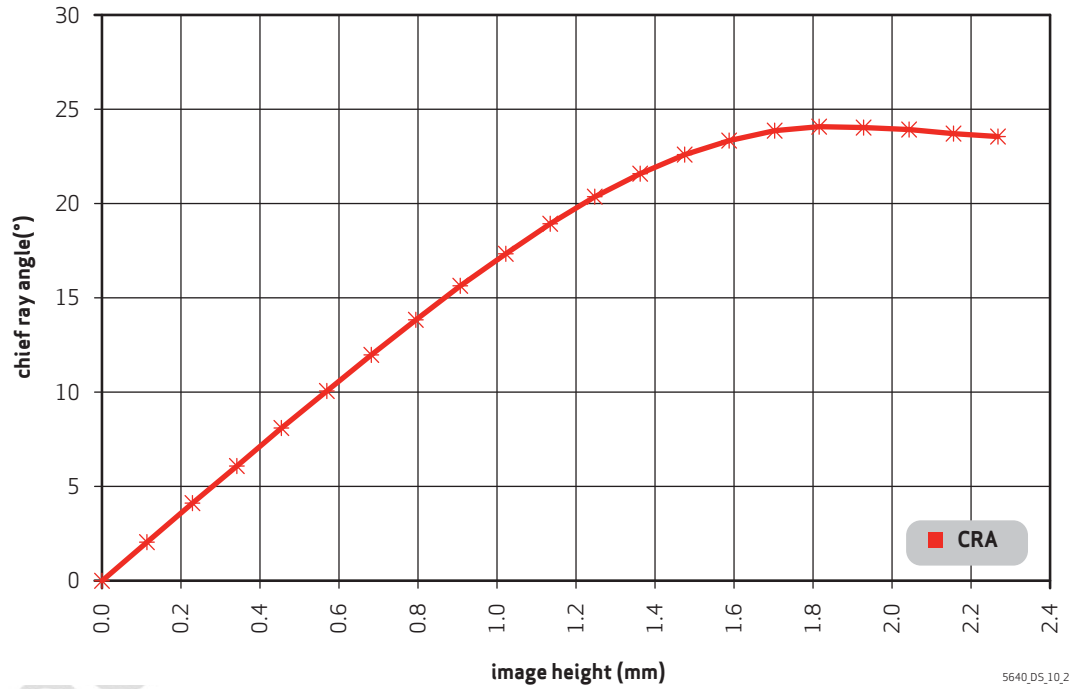


table 10-1 CRA versus image height plot (sheet 1 of 2)

| field (%) | image height (mm) | CRA (degrees) |
|-----------|-------------------|---------------|
| 0         | 0                 | 0             |
| 0.05      | 0.114             | 2.0           |
| 0.1       | 0.227             | 4.1           |
| 0.15      | 0.341             | 6.1           |
| 0.2       | 0.454             | 8.1           |
| 0.25      | 0.568             | 10.1          |
| 0.3       | 0.681             | 12.0          |
| 0.35      | 0.795             | 13.8          |
| 0.4       | 0.908             | 15.6          |
| 0.45      | 1.022             | 17.3          |

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

| field (%) | image height (mm) | CRA (degrees) |
|-----------|-------------------|---------------|
| 0.5       | 1.135             | 18.9          |
| 0.55      | 1.249             | 20.4          |
| 0.6       | 1.362             | 21.6          |
| 0.65      | 1.476             | 22.6          |
| 0.7       | 1.589             | 23.4          |
| 0.75      | 1.703             | 23.9          |
| 0.8       | 1.816             | 24.1          |
| 0.85      | 1.930             | 24.1          |
| 0.9       | 2.043             | 23.9          |
| 0.95      | 2.157             | 23.7          |
| 1         | 2.270             | 23.6          |

Confidential for  
xin jin hui

**OV5640**

color CMOS QXGA (5 megapixel) image sensor with OmniBSI™ technology

Confidential for  
xin jin hui

Confidential for  
xin jin hui

**OV5640**

color CMOS QSXGA (5 megapixel) image sensor with OmniBSI™ technology

Confidential for  
xin jin hui



## revision history

- |             |            |
|-------------|------------|
| version 1.0 | 02.02.2010 |
|-------------|------------|
- initial release

Confidential for  
xin jin hui

**OV5640**

color CMOS QXGA (5 megapixel) image sensor with OmniBSI™ technology

Confidential for  
xin jin hui

Confidential for  
xin jin hui

the clear advantage™

## OmniVision Technologies, Inc.

### UNITED STATES

4275 Burton Drive  
Santa Clara, CA 95054

tel: +1 408 567 3000

fax: +1 408 567 3001

email: [salesamerican@ovt.com](mailto:salesamerican@ovt.com)

### UNITED KINGDOM

Hampshire +44 1256 744 610

### FINLAND

Mouhijärvi +358 3 341 1898

### GERMANY

Munich +49 89 63 81 99 88

### CHINA

Beijing +86 10 6580 1690

Shanghai +86 21 6105 5100

Shenzhen +86 755 8384 9733

Hong Kong +852 2403 4011

### JAPAN

Tokyo +81 3 5765 6321

### KOREA

Seoul +82 2 3478 2812

### SINGAPORE +65 6562 8250

### TAIWAN

Taipei +886 2 2657 9800 -  
ext.#100

website: [www.ovt.com](http://www.ovt.com)