## Omnirzision.



## Copyright ©2015 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. OmniPixel3-GS is a trademark of OmniVision Technologies, Inc.

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

## OS VGA (640 x 480) image sensor with OmniPixel3-GS ${ }^{\text {TM }}$ technology

datasheet (CSP3)
PRODUCT SPECIFICATION
version 2.12
october 2015

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

## applications

- cellular phones
- digital still cameras (DSC)
- digital video camcorders (DVC)
- PC multimedia
- tablets
ordering information
- OV07750-A35A-1F (color, lead-free) 35-pin CSP3
- OV07251-A35A-1F (b\&w, lead-free) 35-pin CSP3


## features

- $3 \mu \mathrm{~m} \times 3 \mu \mathrm{~m}$ pixel with OmniPixel3-GS ${ }^{T M}$ technology
- automatic black level calibration (ABLC)
- programmable controls for frame rate, mirror and flip, cropping and windowing
- support output formats: 8/10-bit RAW
- support for image sizes: $640 \times 480,320 \times 240,160 \times 120$
- fast mode switching
- supports horizontal and vertical 2:1 and 4:1 monochrome subsampling
- supports $2 \times 2$ monochrome binning
- one-lane MIPI serial output interface
- one-lane LVDS serial output interface
- embedded 256 bits of one-time programmable (OTP) memory for part identification
- two on-chip phase lock loops (PLLs)
- built-in 1.5 V regulator for core
- PWM
- built-in strobe control


## key specifications (typical)

- active array size: $640 \times 480$
- power supply:
analog: 2.8 V (nominal)
core: 1.5 V (optional)
I/O: 1.8 V (nominal)
- power requirements:
active: $117 \mathrm{~mW} @ 100 \mathrm{fps}$, VGA output
standby: $15 \mu \mathrm{~A}$ for AVDD, $40 \mu \mathrm{~A}$ for DOVDD
without input clock, $700 \mu \mathrm{~A}$ for DOVDD with input
clock
XSHUTDOWN: $5 \mu \mathrm{~A}$ for AVDD, $5 \mu \mathrm{~A}$ for DOVDD
- temperature range:
operating: $-30^{\circ} \mathrm{C}$ to $70^{\circ} \mathrm{C}$ junction temperature
stable image: $0^{\circ} \mathrm{C}$ to $50^{\circ} \mathrm{C}$ junction temperature
- output interface: 1-lane MIPI/LVDS serial output
- output formats: 10-bit RGB RAW or BW
- lens size: $1 / 7.5^{\prime \prime}$
- input clock frequency: $6 \sim 27 \mathrm{MHz}$
- lens chief ray angle: $29^{\circ}$ non-linear
- max S/N ratio: 39 dB
- dynamic range: 69.6dB @ 8x gain
- maximum image transfer rate:
$640 \times 480$ : 100 fps (see table 2-1)
- OV7750 sensitivity:

2080mV/Lux-sec @ 530nm

- OV7251 sensitivity:
$7190 \mathrm{mV} /\left(\mu \mathrm{W} . \mathrm{cm}^{-2} . \mathrm{sec}\right) @ 850 \mathrm{~nm}$
2800mV/Lux-sec @ 530nm
- scan mode: progressive
- maximum exposure interval: $502 \times t_{\text {ROW }}$
- pixel size: $3 \mu \mathrm{~m} \times 3 \mu \mathrm{~m}$
- dark current: $350 \mathrm{e}^{-} / \mathrm{s}$ @ $50^{\circ} \mathrm{C}$ junction temperature
- image area: $1968 \mu \mathrm{~m} \times 1488 \mu \mathrm{~m}$
- package dimensions: $3910 \mu \mathrm{~m} \times 3410 \mu \mathrm{~m}$


## table of contents

1 signal descriptions ..... 1-1
2 system level description ..... 2-1
2.1 overview ..... 2-1
2.2 architecture ..... 2-1
2.3 format and frame ..... 2-3
2.3.1 MIPI interface ..... 2-4
2.3.2 VSYNC timing in MIPI mode ..... 2-5
2.4 I/O control ..... 2-6
2.5 power management ..... 2-7
2.5.1 power up sequence ..... 2-7
2.5.2 power down sequence ..... 2-10
2.6 reset ..... 2-14
2.6.1 power ON reset generation ..... 2-14
2.7 hardware and software standby ..... 2-14
2.8 system clock control ..... 2-15
2.8.1 PLL configuration ..... 2-15
2.9 serial camera control bus (SCCB) interface ..... 2-18
2.9.1 data transfer protocol ..... 2-18
2.9.2 message format ..... 2-18
2.9.3 read/write operation ..... 2-19
2.9.4 SCCB timing ..... 2-22
2.9.5 group write and fast mode switching ..... 2-23
3 block level description ..... 3-1
3.1 pixel array structure ..... 3-1
3.2 subsampling ..... 3-2
4 image sensor core digital functions ..... 4-1
4.1 mirror and flip ..... 4-1
4.2 image windowing ..... 4-2
4.3 test pattern ..... 4-3
4.3.1 general color bar ..... 4-3
4.3.2 solid color test pattern ..... 4-3
4.4 black level calibration (BLC) ..... 4-4
4.5 one time programmable (OTP) memory ..... 4-7
4.5.1 OTP memory structure ..... 4-7
4.5.2 accessing the OTP memory ..... 4-7
4.5.3 procedure for accessing OTP memory ..... 4-9
4.5.4 procedure to read OTP content ..... 4-9
4.5.5 procedure to program OTP content ..... 4-9
4.5.6 power supply requirement for OTP memory programming ..... 4-9
4.6 pulse width modulation (PWM) ..... 4-10
4.7 strobe ..... 4-11
4.8 low power modes ..... 4-12
4.8.1 low frame rate mode ..... 4-13
4.8.2 snapshot mode ..... 4-13
4.8.3 external trigger snapshot mode ..... 4-14
5 image sensor processor digital functions ..... 5-1
5.1 ISP general controls ..... 5-1
5.2 manual white balance (MWB) ..... 5-1
5.3 manual exposure and gain control ..... 5-2
6 system control ..... 6-1
6.1 mobile industry processor interface (MIPI) ..... 6-1
6.2 low-voltage differential signaling (LVDS) ..... 6-9
6.2.1 output modes ..... 6-10
6.2.2 PHY specification ..... 6-11
6.2.3 LVDS lane configuration and sync ..... 6-11
7 register tables ..... 7-1
7.1 system control [0x0100-0x010A, 0x3001 - 0x301F, 0x3023-0x303B, 0x4501] ..... 7-1
7.2 PLL control [0x3080-0x3083, 0x3098-0x309F, 0x30B0-0x30B6] ..... 7-8
7.3 SCCB and group hold control [0x3100-0x3106, 0x31FF - 0x320F] ..... 7-9
7.4 manual AWB_gain control [0x3400-0x3406] ..... 7-10
7.5 manual AEC/AGC [0x3500-0x350B, 0x5D00-0x5D01, 0x5F00-0x5F05] ..... 7-11
7.6 analog control [0x3600-0x3684] ..... 7-13
7.7 sensor control [0x3700-0x37AF] ..... 7-14
7.8 timing control [0x3800-0x3835, 0x3837] ..... 7-15
7.9 PWM and strobe control [0x3B80-0x3B97] ..... 7-18
7.10 low power mode control [0x3C00-0x3C0F, 0x4A47-0x4A49] ..... 7-20
7.11 OTP control [0x3D80 - 0x3D87, 0x3D8B] ..... 7-21
7.12 BLC control [0×4000-0x4051] ..... 7-22
7.13 frame control [0×4240-0x4244] ..... 7-26
7.14 format control [0x4300-0x4307, 0x4310-0x4316, 0x4320-0x4329] ..... 7-26
7.15 VFIFO control [0x4600-0x4604] ..... 7-28
7.16 MIPI top [ $0 \times 4800-0 \times 4806,0 \times 4810-0 \times 4849,0 \times 4850-0 \times 4854,0 \times 4860-0 \times 4865$ ] ..... 7-29
7.17 LVDS control [0x4A00, 0x4A02 - 0x4AOF] ..... 7-38
7.18 ISP top [0x5000-0x5017, 0x5E00 - 0x5E08] ..... 7-39
7.19 window control [0x5A00-0x5AOC] ..... 7-41
8 operating specifications ..... 8-1
8.1 absolute maximum ratings ..... 8-1
8.2 functional temperature ..... 8-1
8.3 DC characteristics ..... 8-2
8.4 timing characteristics ..... 8-3
9 mechanical specifications ..... 9-1
9.1 physical specifications ..... 9-1
9.2 IR reflow specifications ..... 9-2
10 optical specifications ..... 10-1
10.1 sensor array center ..... 10-1
10.2 lens chief ray angle (CRA) ..... 10-2
10.3 IR cut off wavelength ..... 10-3
10.4 spectrum response ..... 10-3

## list of figures

figure 1-1 pin diagram ..... 1-3
figure 2-1 OV7750/OV7251 block diagram ..... 2-2
figure 2-2 reference design schematic ..... 2-3
figure 2-3 MIPI timing ..... 2-4
figure 2-4 VSYNC timing in mode 1 ..... 2-5
figure 2-5 VSYNC timing in mode 2 ..... 2-5
figure 2-6 VSYNC timing in mode 3 ..... 2-5
figure 2-7 power up sequence (case 1) ..... 2-8
figure 2-8 power up sequence (case 2) ..... 2-9
figure 2-9 power down sequence (case 1) ..... 2-11
figure 2-10 power down sequence (case 2) ..... 2-12
figure 2-11 standby sequence ..... 2-13
figure 2-12 OV7750/OV7251 PLL1 clock diagram ..... 2-15
figure 2-13 OV7750/OV7251 PLL2 clock diagram ..... 2-15
figure 2-14 clock connection diagram ..... 2-17
figure 2-15 message type ..... 2-18
figure 2-16 SCCB single read from random location ..... 2-19
figure 2-17 SCCB single read from current location ..... 2-19
figure 2-18 SCCB sequential read from random location ..... 2-20
figure 2-19 SCCB sequential read from current location ..... 2-20
figure 2-20 SCCB single write to random location ..... 2-21
figure 2-21 SCCB sequential write to random location ..... 2-21
figure 2-22 SCCB interface timing ..... 2-22
figure 3-1 sensor array region color filter layout ..... 3-1
figure 3-2 example of $2 \times 2$ binning ..... 3-2
figure 3-3 example of $2: 1$ subsampling ..... 3-2
figure 3-4 example of 4:1 subsampling ..... 3-2
figure 4-1 mirror and flip samples ..... 4-1
figure 4-2 image windowing ..... 4-2
figure 4-3 test pattern ..... 4-3
figure 4-4 OTP access ..... 4-8
figure 4-5 PWM output timing ..... 4-10
figure 4-6 low frame rate mode timing ..... 4-13
figure 4-7 snapshot mode timing ..... 4-13
figure 4-8 external snapshot mode timing ..... 4-14
figure 4-9 frame triggered by red pulse diagram ..... 4-14
figure 6-1 LVDS 1-lane mode ..... 6-10
figure 6-2 PHY specification diagram ..... 6-11
figure 6-3 LVDS lane configuration and sync ..... 6-11
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
figure 10-3 OV7251 spectrum response curve ..... 10-3

## list of tables

table 1-1 signal descriptions ..... 1-1
table 1-2 configuration under various conditions ..... 1-2
table 1-3 pad symbol and equivalent circuit ..... 1-3
table 2-1 supported resolution and frame rate ..... 2-3
table 2-2 MIPI timing specifications ..... 2-4
table 2-3 I/O control registers ..... 2-6
table 2-4 power up sequence ..... 2-7
table 2-5 power up sequence timing constraints ..... 2-7
table 2-6 power down sequence ..... 2-10
table 2-7 power down sequence timing constraints ..... 2-10
table 2-8 hardware and software standby description ..... 2-14
table 2-9 PLL control registers ..... 2-15
table 2-10 sample PLL configuration ..... 2-17
table 2-11 PLL speed limitation ..... 2-18
table 2-12 SCCB interface timing specifications ..... 2-22
table 2-13 context switching control ..... 2-23
table 3-1 binning-related registers ..... 3-2
table 4-1 mirror and flip registers ..... 4-1
table 4-2 image windowing control functions ..... 4-2
table 4-3 general color bar selection control ..... 4-3
table 4-4 solid color test pattern control ..... 4-3
table 4-5 BLC control functions ..... 4-4
table 4-6 ALS algorithm control registers ..... 4-5
table 4-7 OTP memory structure ..... 4-7
table 4-8 PWM registers ..... 4-10
table 4-9 strobe control registers ..... 4-11
table 4-10 low power mode control registers ..... 4-12
table 5-1 ISP top registers ..... 5-1
table 5-2 manual AWB_gain registers ..... 5-1
table 5-3 manual exposure and gain control registers ..... 5-2
table 6-1 MIPI top control registers ..... 6-1
table 6-2 LVDS registers ..... 6-9
table 6-3 PHY specifications ..... 6-11
table 7-1 system control registers ..... 7-1
table 7-2 PLL control registers ..... 7-8
table 7-3 SCCB and group hold registers ..... 7-9
table 7-4 manual AWB_gain registers ..... 7-10
table 7-5 manual AEC/AGC registers ..... 7-11
table 7-6 analog control registers ..... 7-13
table 7-7 sensor control registers ..... 7-14
table 7-8 timing control registers ..... 7-15
table 7-9 PWM and strobe control registers ..... 7-18
table 7-10 low power mode control registers ..... 7-20
table 7-11 OTP control registers ..... 7-21
table 7-12 BLC control registers ..... 7-22
table 7-13 frame registers ..... 7-26
table 7-14 format control registers ..... 7-26
table 7-15 VFIFO control registers ..... 7-28
table 7-16 MIPI top control registers ..... 7-29
table 7-17 LVDS registers ..... 7-38
table 7-18 ISP top registers ..... 7-39
table 7-19 window control registers ..... 7-41
table 8-1 absolute maximum ratings ..... 8-1
table 8-2 functional temperature ..... 8-1
table 8-3 DC characteristics $\left(T A=23^{\circ} \mathrm{C} \pm 2^{\circ} \mathrm{C}\right)$ ..... 8-2
table 8-4 timing characteristics ..... 8-3
table 9-1 package dimensions ..... 9-1
table 9-2 reflow conditions ..... 9-2
table 10-1 CRA versus image height plot ..... 10-2

## 1 signal descriptions

table 1-1 lists the signal descriptions and their corresponding pin numbers for the OV7750/OV7251 image sensor. The package information is shown in section 9.
table 1-1 signal descriptions (sheet 1 of 2)

| pin number | signal name | pin <br> type | description |
| :---: | :---: | :---: | :---: |
| A1 | NC | - | no connect |
| A3 | VH | reference | internal analog reference |
| A4 | VN2 | reference | internal analog reference |
| A5 | AGND | ground | ground for analog circuit |
| A6 | NC | - | no connect |
| A7 | NC | - | no connect |
| B2 | VM | reference | internal analog reference |
| B3 | MDP | I/O | MIPI/LVDS data lane positive output |
| B4 | VN1 | reference | internal analog reference |
| B5 | AVDD | power | power for analog circuit |
| B6 | SIOD | 1/O | SCCB data |
| C2 | EVDD | power | power for MIPI circuit (connect to DVDD outside of sensor) |
| C3 | MDN | 1/O | MIPI/LVDS data lane negative output |
| C4 | SIOC | input | SCCB input clock |
| C5 | FSIN/VSYNC | I/O | FSIN input/ VSYNC output |
| C6 | ULPM | output | ULPM open-drain output |
| D1 | NC | - | no connect |
| D2 | EGND | ground | ground for MIPI circuit |
| D3 | MCP | output | MIPI/LVDS clock lane positive output |
| D4 | DGND | ground | ground for I/O and digital circuit |
| D5 | PWM | 1/O | PWM output |
| D6 | XSHUTDOWN | input | reset (active low with internal pull down resistor) |
| D7 | NC | - | no connect |
| E2 | TM | input | test mode (active high with internal pull down resistor) |
| E3 | MCN | output | MIPI/LVDS clock lane negative output |
| E4 | DGND | ground | ground for I/O and digital circuit |

table 1-1 $\quad$ signal descriptions (sheet 2 of 2 )

| pin <br> number | signal name | pin <br> type | description |
| :--- | :--- | :--- | :--- |
| E5 | EXTCLK | input | system input clock/scan clock input |
| E6 | DOVDD | power | power for I/O circuit |
| F1 | NC | - | no connect |
| F2 | NC | - | no connect |
| F3 | DVDD | reference | power for digital circuit |
| F4 | STROBE | I/O | strobe output |
| F5 | DVDD | - | power for digital circuit |
| F6 | NC | - | no connect |
| F7 | NC |  | no connect |

table 1-2 configuration under various conditions

| pin number | signal name | XSHUTDOWN² | after XSHUTDOWN release ${ }^{\text {b }}$ | software standby ${ }^{\text {c }}$ |
| :---: | :---: | :---: | :---: | :---: |
| B3 | MDP | high-z | high-z by default | high-z by default (configurable) |
| B6 | SIOD | high-z | input/open drain | input/open drain |
| C3 | MDN | high-z | high-z by default | high-z by default (configurable) |
| C4 | SIOC | input | input | input |
| C5 | FSIN/VSYNC | high-z | high-z by default | high-z by default (configurable) |
| C6 | ULPM | high-z | input/open drain | input/open drain |
| D3 | MCP | high-z | high-z by default | high-z by default (configurable) |
| D5 | PWM | high-z | high-z by default | high-z by default (configurable) |
| D6 | XSHUTDOWN | input | input | input |
| E2 | TM | input | input | input |
| E3 | MCN | high-z | high-z by default | high-z by default (configurable) |
| E5 | EXTCLK | input | input | input |
| F4 | STROBE | high-z | high-z by default | high-z by default (configurable) |

a. $\mathrm{XSHUTDOWN}=0$
b. $\mathrm{XSHUTDOWN}=1$
c. $\quad X S H U T D O W N=1$
standby initiated by register
figure 1-1 pin diagram

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

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


## 2 system level description

## 2.1 overview

The OV7750 (color) and OV7251 (b\&w) image sensors are low voltage, high performance 1/7.5 inch VGA CMOS image sensors that provide the functionality of a single VGA $(640 \times 480)$ camera using OmniPixel3-GS ${ }^{\text {TM }}$ technology. They provide full-frame, sub-sampled, and windowed 8/10-bit MIPI images via the control of the Serial Camera Control Bus (SCCB) interface.

The OV7750/OV7251 has an image array capable of operating at up to 100 frames per second (fps) in 10-bit VGA resolution with complete user control over image quality, formatting and output data transfer.

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 image.

For customized information purposes, the OV7750/OV7251 includes 256 bits of one-time programmable (OTP) memory. The OV7750/OV7251 has one lane MIPI interface.

## 2.2 architecture

The OV7750/OV7251 sensor core generates streaming pixel data at a constant frame rate. figure 2-1 shows the functional block diagram.

The timing generator outputs signals to access the image array. The entire pixel array is reset at the same point of time. After the exposure time has elapsed, the pixels stop gathering light and store the collected charge in a storage node. The charge then reads out row by row.

The exposure time is controlled by adjusting the time interval between reset and transferring the charge to storage node. 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 OV7750/OV7251 block diagram

figure 2-2 reference design schematic


## 2.3 format and frame

The OV7750/OV7251 supports RAW RGB output with a 1-lane MIPI or LVDS interface.
table 2-1 supported resolution and frame rate

| format $^{\text {a }}$ | resolution | max <br> frame rate | methodology | typical MIPI <br> data rate |
| :--- | :--- | :--- | :--- | :--- |
| full resolution | $640 \times 480$ | 100 fps | full | 1-lane @ 800Mbps |
| $320 \times 240$ | $320 \times 240$ | 180 fps | $2 \times 2$ binning, 2:1 sub-sampling | 1-lane @ 800Mbps |
| $160 \times 120$ | $160 \times 120$ | 360 fps | $4: 1$ sub-sampling | 1-lane @ 800Mbps |

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

## CMOS VGA ( $640 \times 480$ ) image sensor with OmniPixel3-GS ${ }^{\text {mw }}$ technology

### 2.3.1 MIPI interface

The OV7750/OV7251 supports a single lane MIPI transmitter interface with a data transfer rate of up to 800 Mbps.
figure 2-3 MIPI timing


table 2-2 MIPI timing specifications

| mode | timing |
| :---: | :---: |
| full resolution $640 \times 480$ | (1) $478,848 \mathrm{tp}$ <br> (2) $1,024 \mathrm{tp}$ <br> (3) $11,486 \mathrm{tp}$ <br> (4) 928 tp <br> (5) $21,251 \mathrm{tp}$ <br> (6) 387 tp <br> (7) 2 tp <br> (8) -178 tp <br> (9) 29 tp <br> where $\mathrm{tp}=$ Tsclk |
| $320 \times 240$ | (1) $234,688 \mathrm{tp}$ <br> (2) $1,024 \mathrm{tp}$ <br> (3) $6,496 \mathrm{tp}$ <br> (4) 772 tp <br> (5) $42,422 \mathrm{tp}$ <br> (6) 216 tp <br> (7) 22 tp <br> (8) -194 tp <br> (9) 9 tp <br> where $\mathrm{tp}=$ Tsclk |

### 2.3.2 VSYNC timing in MIPI mode

The VSYNC rising edge delay is controlled by register vsync_delay ( $\{0 \times 4314,0 \times 4315,0 \times 4316\}$ ) in all three VSYNC modes. VSYNC width is controlled by register vsync_width_pixel ( $\{0 \times 4311,0 \times 4312\}$ ) for VSYNC modes 1 and 2 . The steps of both registers vsync_delay and vsync_width_pixel are 1 system clock cycle.

Note that VSYNC timing in mode 3 is a long VSYNC mode. The register vsync_width_pixel (\{0x4311, 0x4312\}) controls VSYNC falling edge differently.
2.3.2.1 VSYNC mode 1

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


### 2.3.2.2 VSYNC mode 2

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

2.3.2.3 VSYNC mode 3

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


### 2.4 I/O control

table 2-3 I/O control registers

| function | register | description |  |
| :---: | :---: | :---: | :---: |
| output drive capability control | 0x3001 | Bit[6:5]: | I/O pin drive capability <br> 00: 1 x <br> 01: $2 x$ <br> 10: $3 x$ <br> 11: $4 x$ |
| STROBE I/O control | 0x3005 | Bit[3]: | input/output control for STROBE pin <br> 0 : input <br> 1: output |
| STROBE output select | 0x3027 | Bit[3]: | output selection for STROBE pin <br> 0 : normal data path <br> 1: register control value |
| STROBE output value | 0x3009 | Bit[3]: | STROBE output value |
| PWM I/O control | 0x3005 | Bit[2]: | input/output control for PWM pin <br> 0 : input <br> 1: output |
| PWM output select | 0x3027 | Bit[2]: | output selection for PWM pin <br> 0: normal data path <br> 1: register control value |
| PWM output value | 0x3009 | Bit[2]: | PWM output value |
| FSIN/VSYNC I/O control | 0x3005 | Bit[1]: | input/output control for FSIN pin <br> 0 : input <br> 1: output |
| FSIN/VSYNC output select | 0x3027 | Bit[1]: | output selection for FSIN pin <br> 0 : normal data path <br> 1: register control value |
| FSIN/VSYNC output value | 0x3009 | Bit[1]: | FSIN output value |

## 2.5 power management

### 2.5.1 power up sequence

The OV7750/OV7251 can use an internal regulator to provide digital core 1.5 V DVDD for the digital core.
table 2-4
power up sequence

| case | DVDD | XSHUTDOWN | power up sequence requirement |
| :--- | :--- | :--- | :--- | :--- |
| internal | GPIO | Refer to figure 2-7 <br> 1. AVDD rising can occur before or after DOVDD rising as long as they <br> are rising before XSHUTDOWN rising |  |
|  |  |  | 2. XSHUTDOWN is pulled up after AVDD and DOVDD are stable |
|  | external | GPIO | Refer to figure 2-8 <br> 1. AVDD rising can occur before or after DOVDD rising as long as they <br> are rising before XSHUTDOWN rising |
|  |  | 2. XSHUTDOWN is pulled up after AVDD and DOVDD are stable <br> 3. DVDD rises after DOVDD, but before XSHUTDOWN is pulled high |  |

table 2-5
power up sequence timing constraints

| constraint | label | min | max | unit |
| :---: | :---: | :---: | :---: | :---: |
| AVDD rising - DOVDD rising | t0 |  |  | ms |
| DOVDD rising - AVDD rising | t1 |  |  | ms |
| AVDD or DOVDD rising, whichever is last - XSHUTDOWN rising | t2 | 1 |  | ms |
| XSHUTDOWN rising - first SCCB transaction | t3 | 65536 |  | EXTCLK cycles |
| minimum number of EXTCLK cycles prior to the first SCCB transaction | t4 | 65536 |  | EXTCLK cycles |
| PLL start up/lock time | t5 |  | 0.2 | ms |
| entering streaming mode - first frame start sequence (fixed part) | t6 |  | 10 | ms |
| entering streaming mode - first frame start sequence (variable part) | t7 | delay is the exposure time value |  | lines |
| DOVDD to external DVD rising | t8 | 0 |  | ms |
| DOVDD rising to XSHUTDOWN rising | t9 | 0 |  | ms |

figure 2-7 power up sequence (case 1)

ree runn

note 1 with minimum power consumption
figure 2-8 power up sequence (case 2)


### 2.5.2 power down sequence

The digital and analog supply voltages can be powered down in any order (e.g., DOVDD, then AVDD or AVDD, then DOVDD). Similar to the power-up sequence, the EXTCLK input clock may be either gated or continuous. If the SCCB command to exit streaming is received while a frame of MIPI data is being output, then the sensor must wait to the MIPI frame end code before entering software standby mode.

If the SCCB command to exit streaming mode is received during the inter frame time, then the sensor will enter software standby mode immediately.
table 2-6 power down sequence

| case | DVDD | XSHUTDOWN | power down sequence requirement |
| :---: | :---: | :---: | :---: |
| 1 | internal | GPIO | Refer to figure 2-9 <br> 1. software standby recommended <br> 2. pull XSHUTDOWN low for minimum power consumption <br> 3. AVDD and DOVDD may fall in any order |
| 2 | external | GPIO | Refer to figure 2-10 <br> 1. software standby recommended <br> 2. pull XSHUTDOWN low for minimum power consumption <br> 3. pull DVDD low <br> 4. AVDD and DOVDD may fall in any order |

table 2-7 power down sequence timing constraints

| constraint | label | min | max | unit |
| :--- | :--- | :--- | :--- | :--- |
| enter software standby SCCB command <br> device in software standby mode | to | when a frame of MIPI data is output, <br> wait for the MIPI end code before <br> entering the software for standby; <br> otherwise, enter the software standby <br> mode immediately |  |  |
| minimum of EXTCLK cycles after the <br> last SCCB transaction or MIPI frame end | t 1 | 512 | EXTCLK cycles |  |
| last SCCB transaction or MIPI frame <br> end, XSHUTDOWN falling | t 2 | 512 | EXTCLK cycles |  |
| XSHUTDOWN falling - AVDD falling or <br> DOVDD falling whichever is first | t 3 | 0.0 | ms |  |
| AVDD falling - DOVDD falling | t 4 | AVDD and DOVDD may fall in any <br> order, the falling separation can vary <br> from 0 ns to infinity | ms |  |
| DOVDD falling - AVDD falling | t 5 | ms |  |  |
| XSHUTDOWN falling - DVDD falling | t 6 | 0 | ms |  |
| DVDD falling to DOVDD falling | t 7 | 0 | ms |  |

figure 2-9 power down sequence (case 1)

note 1 with minimum power consumption
figure 2-10 power down sequence (case 2)

the requirement is that EXTCLKmust be active for time $t 1$ after the last SCCB transaction or after the MIPI frame end short packet, whichever is the laterevent.

note 1 with low power consumption
figure 2-11 standby sequence


## 2.6 reset

The OV7750/OV7251 sensor includes a XSHUTDOWN pin (pin D6) that forces a complete hardware reset when it is pulled low (GND). The OV7750/OV7251 clears all registers and resets them to their default values when a hardware reset occurs.

### 2.6.1 power ON reset generation

The OV7750/OV7251 has a power on reset that is generated after the core power becomes stable.

## 2.7 hardware and software standby

Two suspend modes are available for the OV7750/OV7251:

- hardware standby
- software standby
table 2-8 hardware and software standby description

| mode | description |
| :--- | :--- |
| hardware standby with XSHUTDOWN | 1. enabled by pulling XSHUTDOWN pad low <br> 2. power down all blocks |
| software standby | 3. register values are reset to default values <br> 4. no SCCB communication |
| 5. minimum power consumption |  |

## 2.8 system clock control

The OV7750/OV7251 has two on-chip PLLs which generate the system clock from a $6 \sim 27 \mathrm{MHz}$ input clock. A programmable clock divider is provided to generate different frequencies for the system.

### 2.8.1 PLL configuration

figure 2-12 OV7750/OV7251 PLL1 clock diagram


Contact your local OmniVision FAE for additional assistance on PLL configuration.
figure 2-13 OV7750/OV7251 PLL2 clock diagram

table 2-9 PLL control registers (sheet 1 of 3)

| function | address | description |
| :---: | :---: | :---: |
| PLL2_pre_divider | 0x3098 | Bit[4:0]: PLL2 pre-divider  <br>  0x2: $/ 1$ <br>  $0 \times 3:$ 11.5 <br>  $0 \times 4:$ 12 <br>  $0 \times 5:$ 12.5 <br>  $0 \times 6:$ 13 <br>  $0 \times 8:$ 14 <br>  $0 \times C:$ 16 <br>  0x10: 18 <br>  Others: $/ 1$  |
| PLL2_multiplier | 0x3099 | $\begin{array}{ll}\text { Bit[7:0]: } & \text { PLL2 multiplier } \\ & \text { Multiplier }=0 \times 3099[7: 0]\end{array}$ |

table 2-9 PLL control registers (sheet 2 of 3 )

| function | address | description |
| :---: | :---: | :---: |
| PLL2_divider | 0x309D | Bit[2]: PLL2 divider <br>  $0: \quad / 1$ <br>  $1: \quad / 1.5$ |
| PLL2_sys_divider | 0x309A | Bit[3:0]: System clock divider  <br>  $0 \times 2:$ $/ 4$ <br>  $0 \times 3:$ 16 <br>  $0 \times 4:$ 18 <br>  $0 \times 5:$ 110 <br>  $0 \times 6:$ 112 <br> $0 \times 7:$ 114  <br> $0 \times 8:$ 116  <br>  $0 \times 9:$ 118 <br>  Others: Not allowed |
| PLL2_ADC_divider | 0x309B | Bit[3:0]: PLL2 ADC clock divider  <br>  $0 \times 2: \quad 11$  <br>  $0 \times 3: \quad 11.5$  <br>  $0 \times 4: \quad 12$  <br>  $0 \times 5: \quad 12.5$  <br>  $0 \times 6:$ 13 <br>  $0 \times 7: \quad 13.5$  <br>  $0 \times 8:$ 14 <br>  $0 \times 9: \quad 14.5$  <br>  Others: Not allowed  |
| PLL1_multiplier | 0x30B3 | $\begin{array}{ll} \text { Bit[7:0]: } & \text { PLL1 multiplier } \\ & \text { Multiplier }=0 \times 30 \mathrm{~B} 3[7: 0] \end{array}$ |
| PLL1_pre_divider | 0x30B4 | Bit[3:0]: PLL1 pre-divider  <br>  $0 \times 0:$ 11 <br>  $0 \times 1:$ 11 <br>  $0 \times 2:$ 12 <br>  $0 \times 3:$ 13 <br>  $0 \times 4:$ 14 <br>  $0 \times 5:$ 11.5 <br>  $0 \times 6:$ 16 <br>  $0 \times 7:$ 12.5 <br>  $0 \times 8: \quad 18$  <br>  Others: Not allowed  |
| PLL1_pix_divider | 0x30B0 | $\begin{array}{ll} \text { Bit[3:0]: } & \text { PLL1 pixel divider } \\ & 0 \times 8: \quad / 8 \\ & \text { 0xA: } / 10 \\ & \text { Others: Not allowed } \end{array}$ |
| PLL1_divider | 0x30B1 | Bit[4:0]: PLL1 divider <br> Divider $=0 \times 30 \mathrm{~B} 1$ [4:0], when $1 \leq 0 \times 30 \mathrm{~B} 1[4: 0] \leq 16$ <br> Others: Not allowed |

table 2-9 PLL control registers (sheet 3 of 3 )

| function | address | description |  |
| :--- | :---: | :---: | :---: |
|  |  | Bit[2:0]: | PLL1 MIPI divider |
| PLL1_MIPI_divider | $0 \times 30 \mathrm{~B} 5$ |  | $0 \times 2:$ |
|  |  | $0 \times 4:$ | 12 |
|  |  | Others: | 11 |

table 2-10 sample PLL configuration ${ }^{\text {a }}$

| name | address | value |
| :--- | :--- | :--- |
| PLL2_pre_divider | $0 \times 3098[4: 0]$ | $0 \times 04$ |
| PLL2_multiplier | $0 \times 3099[7: 0]$ | $0 \times 28$ |
| PLL2_sys_divider | $0 \times 309 \mathrm{~A}[3: 0]$ | $0 \times 05$ |
| PLL2_ADC_divider | $0 \times 309 \mathrm{~B}[3: 0]$ | $0 \times 04$ |
| PLL_PLL1D | $0 \times 309 \mathrm{D}[2]$ | $0 \times 00$ |
| PLL1_multiplier | $0 \times 30 \mathrm{~B} 3[7: 0]$ | $0 \times 32$ |
| PLL1_pre_divider | $0 \times 30 \mathrm{~B} 4[3: 0]$ | $0 \times 02$ |
| PLL1_pix_divider | $0 \times 30 \mathrm{B0}[3: 0]$ | $0 \times 0 \mathrm{~A}$ |
| PLL1_divider | $0 \times 30 \mathrm{~B} 1[4: 0]$ | $0 \times 01$ |
| SYS_CLK | 48 MHz |  |
| MIPI_CLK | 600 Mbps |  |
| EXTCLK | 24 MHz |  |

a. PLL control for VGA @ 100 fps with 1 lane, 10-bit output
figure 2-14 clock connection diagram

table 2-11 PLL speed limitation

| parameter | value |
| :--- | :--- |
| PLL1_multiplier input | $4 \sim 27 \mathrm{MHz}$ |
| PLL1_multiplier output | $300 \sim 800 \mathrm{MHz}$ |
| PLL2_multiplier input | $4 \sim 27 \mathrm{MHz}$ |
| PLL2_multiplier output | $200 \sim 800 \mathrm{MHz}$ |
| SYS_CLK | up to 51 MHz |

## 2.9 serial camera control bus (SCCB) interface

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

The OV7750/OV7251 responds to two SCCB ID set by register SC_SCCB_ID1 (default 0xC0) and SC_SCCB_ID2 (default $0 x E 0$ ). One of them can be used as a broadcasting ID and the other one can be programmed to a unique ID.

### 2.9.1 data transfer protocol

The data transfer of the OV7750/OV7251 follows the SCCB protocol.

### 2.9.2 message format

The OV7750/OV7251 supports the message format shown in figure 2-15. The repeated START (Sr) condition is not shown in SCCB single read from random location, but is shown in SCCB single read from current location and SCCB sequential read from random location.
figure 2-15 message type
message type: 16 -bit sub-address, 8 -bit data, and 7-bit slave address


### 2.9.3 read / write operation

The OV7750/OV7251 supports four different read operations and two different write operations:

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

The sub-address in the sensor automatically increases by one after each read/write operation.
In a single read from random locations, the master does a dummy write operation to desired sub-address, issues a repeated start condition and then addresses the camera again with a read operation. After acknowledging its slave address, the camera starts to output data onto the SIOD line as shown in figure 2-16. The master terminates the read operation by setting a negative acknowledge and stop condition.
figure 2-16 SCCB single read from random location


If the host addresses the camera with read operation directly without the dummy write operation, the camera responds by setting the data from last used sub-address to the SIOD line as shown in figure 2-17. The master terminates the read operation by setting a negative acknowledge and stop condition.
figure 2-17 SCCB single read from current location


The sequential read from a random location is illustrated in figure 2-18. The master does a dummy write to the desired sub-address, issues a repeated start condition after acknowledge from slave and addresses the slave again with read operation. If a master issues an acknowledge after receiving data, it acts as a signal to the slave that the read operation shall continue from the next sub-address. When master has read the last data byte, it issues a negative acknowledge and stop condition.
figure 2-18 SCCB sequential read from random location


The sequential read from current location is similar to a sequential read from a random location. The only exception is that there is no dummy write operation, as shown in figure 2-19. The master terminates the read operation by setting a negative acknowledge and stop condition.
figure 2-19 SCCB sequential read from current location


The write operation to a random location is illustrated in figure 2-20. The master issues a write operation to the slave, sets the sub-address and data correspondingly after the slave has acknowledged. The write operation is terminated with a stop condition from the master.
figure 2-20 SCCB single write to random location


The sequential write is illustrated in figure 2-21. The slave automatically increments the sub-address after each data byte. The sequential write operation is terminated with stop condition from the master.
figure 2-21 SCCB sequential write to random location


### 2.9.4 SCCB timing

figure 2-22 SCCB interface timing

table 2-12 SCCB interface timing specifications ${ }^{\text {ab }}$

| symbol | parameter | min | typ | max | unit |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\mathrm{f}_{\text {SIOC }}$ | clock frequency |  |  | 400 | kHz |
| tow | clock low period | 1.3 |  |  | $\mu \mathrm{s}$ |
| $\mathrm{t}_{\text {HIGH }}$ | clock high period | 0.6 |  |  | $\mu \mathrm{s}$ |
| $\mathrm{t}_{\mathrm{AA}}$ | SIOC low to data out valid | 0.1 |  | 0.9 | $\mu \mathrm{s}$ |
| $\mathrm{t}_{\text {BUF }}$ | bus free time before new start | 1.3 |  |  | $\mu \mathrm{s}$ |
| $t_{\text {HD: }}$ STA | start condition hold time | 0.6 |  |  | $\mu \mathrm{s}$ |
| $\mathrm{t}_{\text {SU:STA }}$ | start condition setup time | 0.6 |  |  | $\mu \mathrm{s}$ |
| $t_{\text {HD: DAT }}$ | data in hold time | 0 |  |  | $\mu \mathrm{s}$ |
| $\mathrm{t}_{\text {SU:DAT }}$ | data in setup time | 0.1 |  |  | $\mu \mathrm{s}$ |
| $\mathrm{t}_{\text {SU: }}$ Sto | stop condition setup time | 0.6 |  |  | $\mu \mathrm{s}$ |
| $t_{R}, t_{F}$ | SCCB rise/fall times |  |  | 0.3 | $\mu \mathrm{s}$ |
| $t_{\text {DH }}$ | data out hold time | 0.05 |  |  | $\mu \mathrm{s}$ |

a. SCCB timing is based on 400 kHz mode
b. timing measurement shown at the beginning of the rising edge or the end of the falling edge signifies $30 \%$, timing measurement shown in the middle of the rising/falling edge signifies $50 \%$, timing measurement shown at the beginning of the falling edge or the end of the rising edge signifies $70 \%$

### 2.9.5 group write and fast mode switching

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. Group write can be used to switch modes quickly.
table 2-13 context switching control

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3208 | GROUP ACCESS | - | W | ```Group Access Bit[7:4]: group_ctrl 0000: Group hold start 0001: Group hold end 1010: Group launch 1110: Immediate launch others: Reserved Bit[3:0]: group_id 0000: Group bank 0, default start from address 0x00 0001: Group bank 1, default start from address 0x40 others: Reserved``` |
| 0x3209 | GRP0_PERIOD | $0 \times 00$ | RW | Frames For Staying in Group 0 |
| 0x320A | GRP1_PERIOD | 0x00 | RW | Frames For Staying in Group 1 |
| 0x320B | GRP_SWCTRL | $0 \times 01$ | RW | $\operatorname{Bit}[3]:$ group_switch_repeat <br> $\operatorname{Bit}[2]:$ Group switch enable <br> $\operatorname{Bit}[1: 0]:$ Second group selection |
| 0x320D | GRP_ACT | - | R | Indicates Which Group is Active |
| 0x320E | FRAME_CNT_GRPO | - | R | frame_cnt_grp0 |
| 0x320F | FRAME_CNT_GRP1 | - | R | frame_cnt_grp1 |

## 3 block level description

## 3.1 pixel array structure

The OV7750/OV7251 sensor has an image array of 656 columns by 496 rows ( 325,376 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 325,376 pixels, $316,224(648 \times 488)$ are active pixels and can be output. The other pixels are used for black level calibration and interpolation.
figure 3-1 sensor array region color filter layout


## 3.2 subsampling

There are two subsampling modes in the OV7750/OV7251: binning and skipping. Both are acceptable methods of reducing output resolution while maintaining the field of view. Binning is usually preferred as it increases the pixel's signal-to-noise ratio. When the binning function is ON, voltage levels of adjacent pixels are averaged. In skipping mode (binning function is OFF), alternate pixels, which are not output, are merely skipped. The OV7750/OV7251 supports $2 \times 2$ binning. figure 3-2 illustrates $2 \times 2$ binning, where the voltage levels of two horizontal ( $2 \times 1$ ) adjacent same-color pixels are averaged before entering the ADC.
figure 3-2 example of $2 \times 2$ binning

figure 3-3 example of $2: 1$ subsampling

figure 3-4 example of 4:1 subsampling

table 3-1 binning-related registers

| address | register name | default <br> value | R/W | description |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $0 \times 3820$ | TIMING_FORMAT1 | $0 \times 00$ | RW | Bit[1]: | Vertical binning |
| $0 \times 3821$ | TIMING_FORMAT2 | $0 \times 00$ | RW | Bit[0]: | Horizontal binning |

## 4 image sensor core digital functions

## 4.1 mirror and flip

The OV7750/OV7251 provides mirror and flip readout modes, which respectively reverse the sensor data readout order horizontally and vertically (see figure 4-1).
figure 4-1 mirror and flip samples

F
original image

mirrored image

flipped image

mirrored and flipped image
7750_DS_4_1
table 4-1 mirror and flip registers

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3820 | IMAGE_ ORIENTATION | 0x00 | RW | Timing Control Register Bit[2]: Vertical flip enable <br> 0: Normal <br> 1: Vertical flip |
| 0x3821 | IMAGE_ ORIENTATION | 0x00 | RW | Timing Control Register <br> Bit[2]: Horizontal mirror enable <br> 0: Normal <br> 1: Horizontal mirror |

## 4.2 image windowing

An image windowing area is defined by four parameters, horizontal start (HS), horizontal end (HE), vertical start (VS), and vertical end (VE). By properly setting the parameters, any portion within the sensor array size can output as a visible area. Windowing is achieved by simply masking off the pixels outside of the window; thus, the timing is not affected.
figure 4-2 image windowing

table 4-2 image windowing control functions

| function | register | R/W | description |
| :---: | :---: | :---: | :---: |
| horizontal start | \{0x3800, 0x3801\} | RW | $\begin{aligned} \mathrm{HS}[9: 8] & =0 \times 3800 \\ \mathrm{HS}[7: 0] & =0 \times 3801 \end{aligned}$ |
| vertical start | \{0x3802, 0x3803\} | RW | $\begin{aligned} \mathrm{VS}[9: 8] & =0 \times 3802 \\ \mathrm{VS}[7: 0] & =0 \times 3803 \end{aligned}$ |
| horizontal end | \{0x3804, 0x3805\} | RW | $\begin{aligned} \mathrm{HE}[9: 8] & =0 \times 3804 \\ \mathrm{HE}[7: 0] & =0 \times 3805 \end{aligned}$ |
| vertical end | \{0x3806, 0x3807\} | RW | $\begin{aligned} \mathrm{VE}[9: 8] & =0 \times 3806 \\ \mathrm{VE}[7: 0] & =0 \times 3807 \end{aligned}$ |

## 4.3 test pattern

For testing purposes, the OV7750/OV7251 offers three test patterns:

### 4.3.1 general color bar

figure 4-3 test pattern

table 4-3 general color bar selection control

| function | register | default value | R/W | description |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| general color bar | $0 \times 5 \mathrm{E} 00$ | $0 \times 0 \mathrm{C}$ | RW | Bit[7]: | Color bar enable |

### 4.3.2 solid color test pattern

table 4-4 solid color test pattern control (sheet 1 of 2)

| function | register | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| solid color test pattern | 0x4320 | 0x80 | RW | Bit[7:6]: Pixel order <br>  $00:$ GR/BG <br>  $01:$ RG/GB <br>  $10: \quad$ BG/GR <br>  $11:$ GB/RG <br> Bit[1]: Solid color test enable <br>  $0: \quad$ Solid color test OFF <br>  $1: \quad$ Solid color test enable <br> Bit[0]: Debug control |
| solid color B | 0x4322 | $0 \times 00$ | RW | Bit[1:0]: solid_color_b[9:8] |
| solid color B | $0 \times 4323$ | $0 \times 00$ | RW | Bit[7:0]: solid_color_b[7:0] |
| solid color Gb | 0x4324 | $0 \times 00$ | RW | Bit[1:0]: solid_color_Gb[9:8] |
| solid color Gb | 0x4325 | $0 \times 00$ | RW | Bit[7:0]: solid_color_Gb[7:0] |

table 4-4 solid color test pattern control (sheet 2 of 2 )

| function | register | default value | RMW | description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| solid color R | $0 \times 4326$ | $0 \times 00$ | RW | Bit[1:0]: solid_color_r[9:8] |
| solid color R | $0 \times 4327$ | $0 \times 00$ | RW | Bit[7:0]: solid_color_r[7:0] |
| solid color Gr | $0 \times 4328$ | $0 \times 00$ | RW | Bit[1:0]: solid_color_Gr[9:8] |
| solid color Gr | $0 \times 4329$ | $0 \times 00$ | RW | Bit[7:0]: solid_color_Gr[7:0] |

## 4.4 black level calibration (BLC)

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

Black level adjustments can be made with registers $0 \times 4000,0 \times 4001,0 \times 4002,0 \times 4003,0 \times 4004$ and $0 \times 4009$.
table 4-5 BLC control functions (sheet 1 of 2 )

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x5000 | ISP CTRLOO | 0x85 | RW | Bit[0]: BLC enable  <br>  $0:$ Disable <br>  $1:$ Enable |
| 0x4001 | BLC CTRL 01 | 0xC2 | RW | Bit[7]: Slope apply enable <br>  <br>  <br> $0:$ <br>  <br> 1: $\quad$ Disable <br> Bit[5:0]le <br> BLC start line number  |
| 0x4002 | BLC_AUTO | $0 \times 45$ | RW |  |

table 4-5 BLC control functions (sheet 2 of 2 )

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4003 | BLC_FREEZE | $0 \times 08$ | RW |  |
| 0x4004 | BLC NUM | 0x04 | RW | Bit[7:6]: Reserved <br> Bit[5:0]: Number of black lines used |
| 0x4009 | BLC_TARGET | 0x10 | RW | Bit[7:0]: Black target level[7:0] |

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

| address | register name | default <br> value | R/W | Rescription |
| :--- | :--- | :--- | :--- | :--- | :--- |

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

|  |  | default |  |  |
| :---: | :---: | :---: | :---: | :---: |
| address | register name | value | R/W | description |
| 0x4E07 | ALS CTRL 07 | 0x00 | RW | Bit[7:0]: Large zone manual threshold[7:0] |
| 0x4E08 | ALS CTRL 08 | 0x00 | RW | Bit[0]: Threshold manual enable <br> 0: Disable <br> 1: Enable |
| 0x4E09 | ALS CTRL 09 | 0x00 | RW | Bit[7:0]: Percentage step |
| 0x4E0A | ALS CTRL OA | 0x00 | RW | Bit[7:0]: Light meter $\mathrm{x} 0[15: 8]$ |
| 0x4E0B | ALS CTRL OB | 0x00 | RW | Bit[7:0]: Light meter x0[7:0] |
| 0x4E0C | ALS CTRL OC | 0xFF | RW | Bit[7:0]: Light meter $\times 1$ [15:8] |
| 0x4E0D | ALS CTRL OD | 0xFF | RW | Bit[7:0]: Light meter x1[7:0] |
| 0x4E0E | ALS CTRL OE | $0 \times 00$ | RW | Bit[7:0]: Analog register control[15:8] |
| 0x4E0F | ALS CTRL OF | 0x23 | RW | Bit[7:0]: Analog register control[7:0] |
| 0x4E10 | ALS CTRL 10 | 0x0A | RW | Bit[7]: Interrupt trigger enable <br>  <br>  <br> $0:$ <br> 0: Disable <br> 1: Enable  <br> Bit[6]: Reserved <br> Bit[5]: Interrupt_sel <br> Bit[4]: ready_intr_pol <br> Bit[3]: threshold_intr_pol <br> $\operatorname{Bit}[2]:$ Reserved <br> $\operatorname{Bit}[1]:$ Out range interrupt enable <br> $\operatorname{Bit[0]:~}$ Host clear format |
| 0x4E11 | ALS CTRL 11 | 0x02 | RW | Bit[7]: Ready_opt <br> Bit[6:0]: Mask period number |
| 0x4E12 | ALS CTRL 12 | $0 \times 01$ | RW | Bit[7:0]: Interrupt number control interrupt width |
| 0x4E20 | ALS CTRL 20 | - | R | Bit[6:2]: Range adjust <br> Bit[1]: Light meter zone2 <br> Bit[0]: Light meter zone1 |
| 0x4E21 | ALS CTRL 21 | - | R | Zone 1 Light Meter Statistics High Byte |
| 0x4E22 | ALS CTRL 22 | - | R | Zone 1 Light Meter Statistics Low Byte |
| 0x4E23 | ALS CTRL 23 | - | R | Zone 2 Light Meter Statistics High Byte |
| 0x4E24 | ALS CTRL 24 | - | R | Zone 2 Light Meter Statistics Low Byte |
| 0x4E25 | ALS CTRL 25 | - | R | Zone 1 Light Meter Statistics High Byte |
| 0x4E26 | ALS CTRL 26 | - | R | Zone 1 light Meter Statistics Low Byte |
| 0x4E27 | ALS CTRL 27 | - | R | Zone 2 Light Meter Statistics High Byte |
| 0x4E28 | ALS CTRL 28 | - | R | Zone 2 Light Meter Statistics Low Byte |

table 4-6 ALS algorithm control registers (sheet 3 of 3 )

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |

## 4.5 one time programmable (OTP) memory

The OV7750/OV7251 has 256-bit embedded one time programmable (OTP) memory. The OTP memory can be programmed and read back via SCCB bus. This document provides general guidelines for programming and accessing the OTP memory.

### 4.5.1 OTP memory structure

128 bits of OTP memory are reserved for OmniVision internal use. These bits are usually used to store the production information or used by the some internal functions. The remaining 128 bits are fully user programmable. The user can store the production tracking information, the camera module calibration data, etc. to these bits.
table 4-7 OTP memory structure

| OTP bits | function |
| :--- | :--- |
| $[127: 0]$ | reserved by OmniVision for internal use |
| $[255: 128]$ | user programmable |

### 4.5.2 accessing the OTP memory

The OTP memory cannot be directly accessed. Instead, it is accessed through its register buffer 0x3D00~0x3D1F as shown in figure 4-4. Register $0 \times 3$ D80 and $0 \times 3$ D81 is the command register to program value to and read value back from OTP memory.

When $0 \times 01$ is programmed to register 0x3D81, the OTP memory controller will load the content of all OTP memory bits to its corresponding register buffer. After that, the user can read the OTP content from its register buffer. It is recommended to clear the register buffer to zero before loading the OTP.

When value $0 \times 01$ is programmed to register $0 \times 3$ D80, the OTP memory controller will program the data of register 0x3D00~0x3D1F to its corresponding OTP memory bits. Keep in mind, the memory is one time programmable. It cannot be programmed back to 0 once it is programmed to 1 . In fact, the OTP memory controller only programs those bits with value 1 in its corresponding memory buffer when programming command is issued. Multi-pass programming is allowed. However, programming 1 to an OTP bit already programmed to 1 in previous pass is prohibited. The user should always program a bit from 0 to 1 only in any programming pass.

OTP access is in system clock domain, so register $0 \times 100$ has to be set to 1 to enable system clock PLL in order to access OTP. The OTP programming pulse width is controlled by register 0x3D82 and the unit is 8 system clock periods. The default value of $0 \times 65$ is for 48 MHz system clock and the programming pulse width is $16.8 \mu \mathrm{~s}$. The OTP read pulse width is set by register $0 \times 3$ D83 and the default value of $0 \times 05$ gives 104 ns at 48 MHz system clock. When the system clock frequency is different, the programming pulse should set to the closest value to $10 \mu \mathrm{~s}$ and should be greater than $9 \mu \mathrm{~s}$. The system clock frequency is dependent on the input clock and PLL configuration. Refer to section 2.8 for details.

When the OTP memory controller is programming data to OTP memory or reading data from OTP memory, the sensor will not respond to any SCCB access. Because OTP programming current is quite high, accessing sensor register is prohibited in order to prevent any glitch on the power supply. It is recommended to wait 15 ms after issuing the OTP read and program command. This delay should scaled with the system clock period.
figure 4-4 OTP access


### 4.5.3 procedure for accessing OTP memory

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

### 4.5.4 procedure to read OTP content

1. Clear software buffer which is to receive the OTP content.
2. Configure PLL and set register $0 \times 100$ to 1 if not yet set.
3. Clear register buffer 0x3D00~0x3D1F to $0 \times 00$.
4. Set register $0 \times 3 D 81$ to $0 \times 01$.
5. Wait 15 ms .
6. Read register $0 \times 3 \mathrm{D} 00 \sim 0 \times 3 \mathrm{D} 1 \mathrm{~F}$ and set to the software buffer.

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

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

### 4.5.5 procedure to program OTP content

1. Follow procedure to read OTP content to make sure the OTP to be programmed is blank.
2. Program the intended OTP content to its corresponding register buffer, and clear unused register buffer to 0 .
a. For customer - registers $0 \times 3$ D00 $\sim 0 \times 3$ D0F must be cleared to $0 \times 00$ before initiating the OTP programming command
3. Read back registers $0 \times 3 \mathrm{D} 00 \sim 0 \times 3 \mathrm{D} 1 \mathrm{~F}$ to make sure they are the correct data to program to OTP memory or 0 for all other bits.
4. Write $0 \times 01$ to register $0 \times 3 \mathrm{D} 80$ to initiate OTP programming.
5. Wait 15 ms , any register access during this period is prohibited.
6. Follow procedure to read OTP content to read back the OTP content.
7. Compare the OTP content read back to the intended OTP content.

### 4.5.6 power supply requirement for OTP memory programming

The OTP memory is programmed using the analog power. The AVDD voltage for OTP programming must be $2.6 \mathrm{~V} \sim$ 3.0 V . The power supply should be able to provide extra 50 mA for OTP programming.

## 4.6 pulse width modulation (PWM)

The PWM uses the pad clock input of $6 \sim 27 \mathrm{MHz}$ and generates a waveform with programmable frequency and duty cycle.
Frequency: frequency of pad clock divided by (1~65535)
Duty cycle: 0~100\%
The PWM output is calculated as follows:
frequency $=$ input_pad_clk_freq / pwm_freq_div_cycle_reg
duty cycle $=\left(\right.$ pwm_duty_cycle_reg $\left./ p w m \_f r e q \_d i v \_c y c l e \_r e g\right) ~ \times 100 \%$
figure 4-5 PWM output timing

table 4-8 PWM registers

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- |
| 0x3B82 | LED_PWM_REG02 | $0 \times 10$ | RW | Bit[7:0]: pwm_freq_div_cycle_reg[15:8] |
| $0 \times 3 B 83$ | LED_PWM_REG03 | $0 \times 00$ | RW | Bit[7:0]: pwm_freq_div_cycle_reg[7:0] |
| $0 \times 3 B 84$ | LED_PWM_REG04 | $0 \times 08$ | RW | Bit[7:0]: pwm_duty_cycle_reg[15:8] |
| $0 \times 3 B 85$ | LED_PWM_REG05 | $0 \times 00$ | RW | Bit[7:0]: pwm_duty_cycle_reg[7:0] |

## 4.7 strobe

Strobe facilitates implementation of a flashlight. Strobe generates a pulse with a reference starting point at the time when the pixel array starts integration. Following a delay after the reference starting point, which is controlled by strobe_frame_shift_direction, strobe_frame_shift[30:0], a pulse with a width of strobe_frame_span[31:0] is generated. The step width of shift and span is programmable under system clock domain.
table 4-9 strobe control registers

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3B88 | LED_PWM_REG08 | 0x00 | RW | Bit[7]: Shift direction <br> Bit[6:0]: strobe_frame_shift[30:24] |
| 0x3B89 | LED_PWM_REG09 | $0 \times 00$ | RW | Bit[7:0]: strobe_frame_shift[23:16] |
| 0x3B8A | LED_PWM_REG0A | 0x00 | RW | Bit[7:0]: strobe_frame_shift[15:8] |
| 0x3B8B | LED_PWM_REGOB | 0x05 | RW | Bit[7:0]: strobe_frame_shift[7:0] |
| 0x3B8C | LED_PWM_REG0C | $0 \times 00$ | RW | Bit[7:0]: strobe_frame_span[31:24] |
| 0x3B8D | LED_PWM_REG0D | $0 \times 00$ | RW | Bit[7:0]: strobe_frame_span[23:16] |
| 0x3B8E | LED_PWM_REG0E | $0 \times 00$ | RW | Bit[7:0]: strobe_frame_span[15:8] |
| 0x3B8F | LED_PWM_REG0F | $0 \times 1 \mathrm{~A}$ | RW | Bit[7:0]: strobe_frame_span[7:0] |
| 0x3B90 | LED_PWM_REG10 | $0 \times 01$ | RW | Bit[7:0]: r_strobe_row_st[15:8] |
| 0x3B91 | LED_PWM_REG11 | 0xB4 | RW | Bit[7:0]: r_strobe_row_st[7:0] |
| 0x3B92 | LED_PWM_REG12 | $0 \times 00$ | RW | Bit[7:0]: r_strobe_cs_st[15:8] |
| 0x3B93 | LED_PWM_REG13 | $0 \times 10$ | RW | $\operatorname{Bit[7:0]:~r\_ strobe+cs\_ st[7:0]~}$ |
| 0x3B94 | LED_PWM_REG14 | 0x05 | RW | Bit[7:0]: step_onerow_man[15:8] |
| 0x3B95 | LED_PWM_REG15 | 0xF2 | RW | Bit[7:0]: step_onerow_man[7:0] |
| 0x3B96 | LED_PWM_REG16 | $0 \times 40$ | RW | Bit[7]: r_strobe_frm_pwen <br> Bit[6]: r_strobe_frm_pwst <br> Bit[5]: r_strobe_pol <br> Bit[4]: r_strobe_step_pix <br> Bit[3]: r_step_onerow_precision_man <br> Bit[2:0]: r_strobe_st_opt |

## 4.8 low power modes

The OV7750/OV7251 sensor supports three low power modes:

- low frame rate streaming mode
- internal trigger snapshot mode
- external trigger snapshot mode
table 4-10 low power mode control registers

| register | description |
| :---: | :---: |
| 0x3C00 | debug control for low power mode, maintains default value of 0x89 all the time. |
| 0x3C01 | Power Control Options 0x63: low power mode 0xAB: normal mode |
| 0x3C02 | Bit[1]: idle phase enable <br> Bit[0]: streaming phase enable <br> 00: not allowed <br> 01: normal streaming mode <br> 10: not allowed <br> 11: enable low power streaming mode |
| 0x3C03 | Low Power Mode Control <br> Bit[7]: Trigger for internal trigger snapshot mode <br> A rising edge on $0 \times 3 \mathrm{C} 03[7]$ wakes the sensor up and streams out $\{0 \times 3404,0 \times 3405\}$ frames <br> Bit[6:0]: Mode control <br> $0 \times 00$ : Low frame rate streaming mode (i.e., repeating the sequence of streaming $\{0 \times 3404,0 \times 3405\}$ frames and then sleeping $\{0 \times 3$ C06, $0 \times 3 \mathrm{C} 07\}$ frame) <br> $0 \times 17$ : External trigger snapshot mode <br> A rising edge on FSIN pin wakes the sensor up and streams out $\{0 \times 3404,0 \times 3405\}$ frames <br> 0x03: Internal trigger snapshot mode <br> A rising edge on $0 \times 3 \mathrm{C} 03[7]$ wakes the sensor up and streams out $\{0 \times 3404,0 \times 3405\}$ frames <br> Others: For debug only |
| \{0x3C04, 0x3C05\} | number of active frames |
| \{0x3C06, 0x3C07\} | number of idle frames |
| 0x3C08~0x3C0B | not used |
| \{0×3C0C, 0x3C0D $\}$ | row period in units of input clock period |
| \{0x3C0E, 0x3C0F\} | number of rows per base frame, usually set to the same value as $\{0 \times 380 \mathrm{E}, 0 \times 380 \mathrm{~F}\}$ |
| 0x3023 | Bit[1]: MIPI power down enable during sleep period <br> 0 : disable for low power streaming mode |

### 4.8.1 low frame rate mode

In low frame rate mode, the OV7750/OV7251 sensor streams N frames, idles for M frames, and then repeats. The power consumption of the OV7750/OV7251 sensor is close to $N /(N+M)$ of the current in full speed streaming mode but the maximum integration time is limited to about $t_{\text {Frame }}-40 t_{\text {Row }}$.
figure 4-6 low frame rate mode timing


### 4.8.2 snapshot mode

In snapshot mode, the OV7750/OV7251 streams N frames upon request through the SCCB and then stays idle until the next request (see figure 4-7).
figure 4-7 snapshot mode timing


### 4.8.3 external trigger snapshot mode

Upon the rising edge of FSIN pulse, the sensor wakes up from sleep mode, starts integration, reads out and sends out number (set by $0 \times 3 C 04,0 \times 3 C 05$ ) of frames. The sensor then returns back to sleep mode (see figure 4-8).
figure 4-8 external snapshot mode timing


FSIN pulse width, $\mathrm{t}_{\text {FSIN_High }}$, should be no shorter than 5 input clock cycles. The wake up sequence takes 16384 input clock cycles, and then the pixel array is reset. The integration starts when the pixel reset finishes. The interval from FSIN rising to integration, $\mathrm{t}_{\text {Exp_Dly }}$, equals to $16388 \times \mathrm{t}_{\text {XVCLK }}+11 \mathrm{t}_{\text {Row }}$. The frame start short packet is sent out about 8 row periods after integration finishes.

The reference voltage of VN2 is critical for image quality in this mode. It is recommended to have a $1 \mu \mathrm{~F}$ cap on this pin to keep the voltage after the sensor goes to sleep mode between two adjacent triggers in a burst. If the sleep period is too long between bursts (e.g., more than 100 ms ), please discard the first triggered frame (e.g., the frame triggered by the red pulse) as shown in figure 4-9. The second frame can be triggered as early as the first frame finishes. It is recommended to keep the frame rate within the burst no less than 10 fps to prevent VN2 discharging too much.
figure 4-9 frame triggered by red pulse diagram
FSIN


7750_DS_4_10

## 5 image sensor processor digital functions

### 5.1 ISP general controls

table 5-1 ISP top registers

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x5000 | ISP CTRLOO | 0x85 | RW | Bit[2]: window_enable <br>  $0:$ Disable <br>  $1: \quad$ Enable <br> Bit[1]: awb_gain_en <br>  $0:$ Disable <br> 1: Enable  <br> Bit[0]: blc_enable <br>  $0:$ Disable <br>  $1: \quad$ Enable |
| 0x5001 | ISP CTRL 01 | 0x00 | RW | Bit[1]: Bypass ISP option 1 <br> $1: \quad$ When bypass ISP option 0 is <br> disabled, will output data after <br>  awb_gain <br> Bit[0]: Bypass ISP option 0 <br> $1: \quad$ Output data directly from ISP input <br>   |

## 5.2 manual white balance (MWB)

The MWB provides digital gain for $R, G$, and $B$ channels. Each channel gain is 12 -bit. $0 \times 400$ is $1 \times$ gain.
table 5-2 manual AWB_gain registers

|  | register name | default <br> value | R/W | description |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $0 \times 3400$ | AWB RED GAIN | $0 \times 04$ | RW | Bit[3:0]: | AWB red gain[11:8] |
| $0 \times 3401$ | AWB RED GAIN | $0 \times 00$ | RW | Bit[7:0]: | AWB red gain[7:0] |
| $0 \times 3402$ | AWB GRN GAIN | $0 \times 04$ | RW | Bit[3:0]: | AWB green gain[11:8] |
| $0 \times 3403$ | AWB GRN GAIN | $0 \times 00$ | RW | Bit[7:0]: | AWB green gain[7:0] |
| $0 \times 3404$ | AWB BLU GAIN | $0 \times 04$ | RW | Bit[3:0]: | AWB blue gain[11:8] |
| $0 \times 3405$ | AWB BLU GAIN | $0 \times 00$ | RW | Bit[7:0]: | AWB blue gain[7:0] |
| $0 \times 3406$ | AWB MAN CTRL | $0 \times 01$ | RW | Bit[0]: | AWB manual control |

## 5.3 manual exposure and gain control

table 5-3 manual exposure and gain control registers

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| $0 \times 3500$ | AEC EXPO | 0x00 | RW | ```Exposure Bit[7:4]: Not used Bit[3:0]: Exposure[15:12]``` |
| 0x3501 | AEC EXPO | 0x00 | RW | Exposure <br> Bit[7:0]: Exposure[11:4] <br> Minimum exposure time is 1 row period. Maximum exposure time is frame length - 20 row periods, where frame length is set by registers $\{0 \times 380 \mathrm{E}$, $0 \times 380 \mathrm{~F}\}$. |
| 0x3502 | AEC EXPO | 0x00 | RW | Exposure <br> Bit[7:4]: Exposure[3:0] <br> Minimum exposure time is 1 row period. Maximum exposure time is frame length - 20 row periods, where frame length is set by registers $\{0 \times 380 \mathrm{E}$, 0x380F\}. <br> Bit[3:0]: Debug control |
| $0 \times 3503$ | MANUAL CONTROL | 0x00 | RW | Bit[1]: Gain manual enable <br> Bit[0]: Exposure manual enable |
| 0x350B | GAIN | 0×10 | RW | Bit[7:0]: Gain[7:0] |

## 6 system control

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

## 6.1 mobile industry processor interface (MIPI)

The OV7750/OV7251 MIPI interface supports a single uni-directional clock lane and a single uni-directional data lane. The data lane has full support for high speed (HS) data transfer. Contact your local OmniVision FAE for more details.
table 6-1 MIPI top control registers (sheet 1 of 9)

| address | register name | default value | R/W | description |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  | MIPI Control 00 |  |
|  |  |  |  | Bit[6]: ck_mark1_en |  |
|  |  |  |  | 1: Enable clock lane mark1 when resume |  |
|  |  |  |  | Bit[5]: $\begin{array}{ll}\text { C } \\ & 0 \\ & 1\end{array}$ | Clock lane gate enable |
|  |  |  |  |  | 0 : Clock lane is free running |
|  |  |  |  |  | 1: Gate clock lane when no packet to transmit |
|  |  |  |  | Bit[4]: | Line sync enable |
| 0x4800 | MIPI CTRL 00 | $0 \times 44$ | RW |  | 0 : Do not send line short packet for each line |
|  |  |  |  | Bit[2]: | 1: Send line short packet for each line Idle status |
|  |  |  |  |  | 0 : MIPI bus will be LPOO when no packet to transmit |
|  |  |  |  |  | 1: MIPI bus will be LP11 when no packet to transmit |
|  |  |  |  | Bit[0]: | clk_lane_dis |
|  |  |  |  |  | 1: Set clock lane to LP mode manually |

table 6-1 MIPI top control registers (sheet 2 of 9)


MIPI Control 01
Bit[7]: Long packet data type manual enable
0 : Use mipi_dt
1: Use dt_man_o as long packet data (see register 0x4814[5:0])
Bit[6]: Short packet data type manual enable
1: Use dt_spkt as short packet data (see register 0x4815[5:0])
Bit[5]: Short packet wc select
0 : Use frame counter or line counter
1: Select spkt_wc_reg_o (\{0×4812,0×4813\})
Bit[4]: PH bit order for ECC
0: \{DI[7:0],WC[7:0],WC[15:8]\}
1: \{DI[0:7],WC[0:7],WC[8:15]\}
Bit[3]: PH byte order for ECC
0: \{DI,WC_I,WC_h\}
1: \{DI,WC_h,WC_I\}
Bit[2]: PH byte order2 for ECC
0 : $\{\mathrm{DI}, \mathrm{WC}\}$
1: $\{W C, D 1\}$
Bit[1]: mark1_en
1: When mipi_sys_susp $=1$, lane 1 sends mark1 fro wkup_dly_o after each reset release
Bit[0]: mark2_en
1: When mipi_sys_susp $=1$, lane2 sends mark1 fro wkup_dly_o after each reset release
table 6-1 MIPI top control registers (sheet 3 of 9)

| address | register name | default value | R/W | description |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  | MIPI Control 02 |  |
|  |  |  |  | Bit[7]: hs_prepare_sel |  |
|  |  |  |  |  | 0 : Auto calculate T_hs_prepare, unit: pclk2x |
|  |  |  |  |  | 1: Use hs_prepare_min_o[7:0] |
|  |  |  |  | Bit[6]: | clk_prepare_sel - |
|  |  |  |  |  | 0: Auto calculate T_clk_prepare, unit: pclk2x |
|  |  |  |  |  | 1: Use clk_prepare_min_o[7:0] |
|  |  |  |  | Bit[5]: | clk_post_sel - |
|  |  |  |  |  | 0: Auto calculate T_clk_post, unit: pclk2x |
|  |  |  |  |  | 1: Use clk_post_min_o[7:0] |
|  |  |  |  | Bit[4]: | clk_trail_sel |
| 0x4802 | MIPI CTRL 02 | 0x00 | RW |  | 0 : Auto calculate T_clk_trail, unit: pclk2x <br> 1: Use clk_trail_min_o[7:0] |
|  |  |  |  | Bit[3]: | hs_exit_sel |
|  |  |  |  |  | 0 : Auto calculate T_hs_exit, unit: pclk2x <br> 1: Use hs_exit_min_o[7:0] |
|  |  |  |  | Bit[2]: | hs_zero_sel ${ }^{\text {- }}$ |
|  |  |  |  |  | 0 : Auto calculate T_hs_zero, unit: pclk2x <br> 1: Use hs_zero_min_o[7:0] |
|  |  |  |  | Bit[1]: | hs_trail_sel |
|  |  |  |  |  | 0: Auto calculate T_hs_trail, unit: pclk2x |
|  |  |  |  |  | 1: Use hs_trail.min_o[7:0] |
|  |  |  |  | Bit[0]: | clk_zero_sel |
|  |  |  |  |  | 0: Auto calculate T_clk_zero, unit: pclk2x |
|  |  |  |  |  | 1: Use clk_zero_min_o[7:0] |


table 6-1 MIPI top control registers (sheet 4 of 9)

table 6-1 MIPI top control registers (sheet 5 of 9)

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4805 | MIPI CTRL 05 | $0 \times 10$ | RW |  |
| 0x4806 | MIPI CTRL 06 | 0x0F | RW | Bit[7]: mipi_test <br> Bit[6]: prbs_en test mode <br> Bit[3]: mipi_slp_man_st MIPI bus status manual control enable in sleep mode <br> Bit[2]: clk_lane_state <br> Bit[1]: data_lane2_state <br> Bit[0]: data_lane1_state |
| $0 \times 4810$ | MIPI MAX FRAME COUNT | 0xFF | RW | High Byte of Maximum Frame Count of Frame Sync Short Packet |
| $0 \times 4811$ | MIPI MAX FRAME COUNT | 0xFF | RW | Low Byte of Maximum Frame Count of Frame Sync Short Packet |
| $0 \times 4812$ | MIPI SHORT PKT COUNTER | $0 \times 00$ | RW | High Byte of Manual Short Packet Word Counter |
| $0 \times 4813$ | MIPI SHORT PKT COUNTER | $0 \times 00$ | RW | Low Byte of Manual Short Packet Word Counter |
| $0 \times 4814$ | MIPI CTRL14 | $0 \times 2 \mathrm{~A}$ | RW | MIPI Control 14 <br> Bit[7:6]: Virtual channel of MIPI <br> Bit[5:0]: Data type in manual mode |

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

|  |  | default |  |  |
| :---: | :---: | :---: | :---: | :---: |
| address | register name | value | R/W | description |
| 0x4815 | MIPI_DT_SPKT | 0x40 | RW | Bit[6]: pclk_inv <br> 0: Use mipi_pclk_o rising edge <br> 1: Use mipi_pclk_o falling edge |
| 0x4818 | HS_ZERO_MIN | 0x00 | RW | High Byte of Minimum Value for hs_zero, unit: ns |
| 0x4819 | HS_ZERO_MIN | $0 \times 9 \mathrm{~A}$ | RW | Low Byte of Minimum Value for hs_zero, unit: ns hs_zero_real = hs_zero_min_o + Tui*ui_hs_zero_min_o |
| 0x481A | HS_TRAIL_MIN | 0x00 | RW | High Byte of Minimum Value for hs_trail, unit: ns |
| 0x481B | HS_TRAIL_MIN | 0x3C | RW | Low Byte of Minimum Value for hs_trail, unit: ns hs_trail_real = hs_trail_min_o + Tui*ui_hs_trail_min_o |
| 0x481C | CLK_ZERO_MIN | 0x01 | RW | High Byte of Minimum Value for clk_zero, unit: ns |
| 0x481D | CLK_ZERO_MIN | 0x86 | RW | Low Byte of Minimum Value for clk_zero, unit: ns clk_zero_real = clk_zero_min_o + Tui*ui_clk_zero_min_o |
| 0x481E | CLK_PREPARE_ MIN | 0x00 | RW | High Byte of Minimum Value for clk_prepare, unit: ns |
| 0x481F | CLK_PREPARE_ MIN | 0x3C | RW | Low Byte of Minimum Value for clk_prepare, unit: ns clk_prepare_real = clk_prepare_min_o + Tui*ui_clk_prepare_min_o |
| 0x4820 | CLK_POST_MIN | 0x00 | RW | High Byte of Minimum Value for clk_post, unit: ns Bit[1:0]: clk_post_min[9:8] |
| 0x4821 | CLK_POST_MIN | 0x56 | RW | Low Byte of Minimum Value for clk_post, unit: ns clk_post_real = clk_post_min_o + Tui*ui_clk_post_min_o |
| 0x4822 | CLK_TRAIL_MIN | 0x00 | RW | High Byte of Minimum Value for clk_trail, unit: ns Bit[1:0]: clk_trail_min[9:8] |
| 0x4823 | CLK_TRAIL_MIN | 0x3C | RW | Low Byte of Minimum Value for clk_trail clk_trail_real = clk_trail_min_o + Tui*ui_clk_trail_min_o |
| 0x4824 | LPX_P_MIN | $0 \times 00$ | RW | High Byte of Minimum Value for lpx_p, unit: ns Bit[1:0]: lpx_p_min[9:8] |
| 0x4825 | LPX_P_MIN | $0 \times 32$ | RW | Low Byte of Minimum Value for Ipx_p, unit: ns lpx_p_real = lpx_p_min_o + Tui*ui_lpx_p_min_o |
| 0x4826 | HS_PREPARE_MIN | 0x00 | RW | High Byte of Minimum Value of hs_prepare, unit: ns |
| 0x4827 | HS_PREPARE_MIN | $0 \times 32$ | RW | Low Byte of Minimum Value for hs_prepare, unit: ns hs_prepare_real = hs_prepare_min_o + Tui*ui_hs_prepare_min_o |
| 0x4828 | HS_EXIT_MIN | 0x00 | RW | High Byte of Minimum Value for hs_exit, unit: ns Bit[1:0]: hs_exit_min[9:8] |

table 6-1 MIPI top control registers (sheet 7 of 9 )

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4829 | HS_EXIT_MIN | 0x64 | RW | Low Byte of Minimum Value for hs_exit, unit: ns hs_exit_real = hs_exit_min_o + Tui*ui_hs_exit_min_o |
| 0x482A | UI_HS_ZERO_MIN | $0 \times 05$ | RW | Minimum UI Value of hs_zero, unit: UI |
| 0x482B | UI_HS_TRAIL_MIN | 0x04 | RW | Minimum UI Value of hs_trail, unit: UI |
| 0x482C | Ul_CLK_ZERO_MIN | 0x00 | RW | Minimum UI Value of clk_zero, unit: UI |
| 0x482D | UI_CLK_PREPARE_ MIN | 0x00 | RW | Bit[5:4]: ui_clk_prepare_max <br> Maximum UI value of clk_prepare, unit: UI <br> Bit[3:0]: ui_clk_prepare_min <br> Minimum UI value of clk_prepare, unit: UI |
| 0x482E | Ul_CLK_POST_MIN | 0x34 | RW | Minimum UI Value of clk_post, unit: UI |
| 0x482F | Ul_CLK_TRAIL_MIN | 0x00 | RW | Minimum UI Value of clk_trail, unit: UI |
| 0x4830 | Ul_LPX_P_MIN | 0x00 | RW | Minimum UI Value of Ipx_p, unit: UI |
| $0 \times 4831$ | UI_HS_PREPARE_ MIN | 0x04 | RW | Bit[7:4]: ui_hs_prepare_max <br> Maximum UI value of hs_prepare, unit: UI <br> Bit[3:0]: ui_hs_prepare_min <br> Minimum UI value of hs_prepare, unit: UI |
| 0x4832 | Ul_HS_EXIT_MIN | 0x00 | RW | Minimum UI Value of hs_exit, unit: UI |
| 0x4833 | MIPI_REG_MIN_H | 0x00 | RW | MIPI RW Register Address Lower Boundary High Byte |
| 0x4834 | MIPI_REG_MIN_L | 0x00 | RW | MIPI RW Register Address Lower Boundary Low Byte |
| 0x4835 | MIPI_REG_MAX_H | 0xFF | RW | MIPI RW Register Address Top Boundary High Byte |
| 0x4836 | MIPI_REG_MAX_L | 0xFF | RW | MIPI RW Register Address Top Boundary Low Byte |
| 0x4837 | PCLK_PERIOD | 0x19 | RW | Period of Pclk2x, pclk_div = 1, and 1-bit Decimal |
| 0x4838 | WKUP_DLY | 0x02 | RW | Wakeup Delay for MIPI (Mark1 state)/2^12 |
| $0 \times 483 \mathrm{~A}$ | DIR_DLY | 0x08 | RW | Change LP Direction Delay/2 After LP11 |

table 6-1 MIPI top control registers (sheet 8 of 9)

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x483B | MIPI_LP_GPIO | $0 \times 33$ | RW |  |
| 0x483C | MIPI_CTRL3C | 0x4F | RW | $\begin{array}{ll} \text { Bit[7:4]: } & \text { t_lpx } \\ & \text { Unit: SCLK cycle } \\ \text { Bit[3:0]: } & \text { t_clk_pre } \\ & \text { Unit: pclk2x cycle } \end{array}$ |
| 0x483D | T_TA_GO | $0 \times 10$ | RW | Unit: sclk cycle |
| 0x483E | T_TA_SURE | 0x06 | RW | Unit: sclk cycle |
| 0x483F | T_TA_GET | 0x14 | RW | Unit: sclk cycle |
| 0x4846 | MIPI_CLIP_MAX | 0x0F | RW | Bit[3:0]: MIPI output data max value[11:8] |
| 0x4847 | MIPI_CLIP_MAX | 0xFF | RW | Bit[7:0]: MIPI output data max value[7:0] |
| 0x4848 | MIPI_CLIP_MIN | 0x0F | RW | Bit[3:0]: MIPI output data min value[11:8] |
| 0x4849 | MIPI_CLIP_MIN | 0xFF | RW | Bit[7:0]: MIPI output data min value[7:0] |
| 0x4850 | REG_INTR_MAN | - | W | Generate 1 SCLK Cycle Pulse for MCU Interrupt |
| $0 \times 4851$ | REG_TX_WR | - | W | Generate 1 SCLK Cycle Pulse to MIPI_TX_LP_TX, and reg_wdata Will Be Sent Out Through MIPI Escape Mode |
| 0x4852 | REG_TX_STOP | - | W | Generate 1 SCLK Cycle Pulse to MIPI_TX_LP_TX, and MIPI_TX_LP_TX Will Go Back to LP11 |
| $0 \times 4853$ | REG_TA_ACK | - | W | Generate 1 SCLK Cycle Pulse to MIPI_TX_LP_TX to Receive TurnAround Command |
| 0x4854 | REG_TA_REQ | - | W | Generate 1 SCLK Cycle Pulse to MIPI_TX_LP_TX to Send TurnAround Command |
| 0x4861 | HD_SK_REG0 | - | R | MIPI RW, SCCB and MCU Read Only |
| 0x4862 | HD_SK_REG1 | - | R | MIPI RW, SCCB and MCU Read Only |
| 0x4863 | HD_SK_REG2 | - | R | MIPI RW, SCCB and MCU Read Only |

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

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4864 | HD_SK_REG3 | - | R | MIPI RW, SCCB and MCU Read Only |
| 0x4865 | MIPI_ST | - | R | Bit[5]: Ip_rx_sel_i <br> 1: MIPI_LP_RX receiving LP data <br> Bit[4]: tx_busy_i <br> 1: MIPI_TX_LP_TX is busy sending LP data <br> Bit[3]: mipi_lp_p1_i <br> MIPI low power input for lane1 $P$ <br> Bit[2]: mipi_lp_n1_i <br> MIPI low power input for lane1 N <br> Bit[1]: mipi_lp_p2_i <br> MIPI low power input for lane2 $P$ <br> Bit[0]: mipi_lp_n2_i <br> MIPI low power input for lane2 P |

## 6.2 low-voltage differential signaling (LVDS)

LVDS is a common differential signaling interface that has low power consumption, minimal EMI, and excellent noise immunity. The maximum data rate for LVDS is 800 Mbps per lane.

Features include:

- supports 10-bit mode
- supports one lane mode
- supports manual setting sync code (can set different sync code for frame start/end and line start/end)
- supports manual setting dummy data in blanking duration
- supports PCLK inversion
table 6-2 LVDS registers (sheet 1 of 2)

| address | register name | default value | RW | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4A00 | LVDS_R0 | 0x2A | RW | Bit[6]: $\quad$ SYNC code manual mode enable <br> Bit[5]: SYNC code enable when only 1 lane <br> Bit[4]: PCLK invert enable <br> Bit[3]: Channel ID enable in sync per lane mode <br> Bit[2]: F parameter in CCIR656 standard <br> Bit[1]: SAV first enable <br> Bit[0]: SYNC code mode <br> 0: Split <br> 1: Per lane |
| 0x4A02 | LVDS_R2 | $0 \times 00$ | RW | Bit[7:0]: Dummy data0[15:8] |

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

| address | register name | default <br> value | RW | description |
| :--- | :--- | :--- | :--- | :--- |
| $0 \times 4$ A03 | LVDS_R3 | $0 \times 80$ | RW | Bit[7:0]: Dummy data0[7:0] |
| $0 \times 4$ A04 | LVDS_R4 | $0 \times 00$ | RW | Bit[7:0]: |

### 6.2.1 output modes

figure 6-1 LVDS 1-lane mode

note $\mathrm{HO}-\mathrm{H7}=$ header (sync code) $/ \mathrm{PO}-\mathrm{Pn}=$ pixel

### 6.2.2 PHY specification

figure 6-2 PHY specification diagram

table 6-3 PHY specifications

| description | min | typ | max | unit |
| :--- | :--- | :--- | :--- | :--- |
| differential output | 150 | 200 | 250 | mV |
| common level output | 150 | 200 | 250 | mV |
| rise/fall time | 150 | 0.3 UI | ps |  |
| data-clock skew | -0.15 UI | 0.15 UI | ps |  |
| impedance | 40 | 50 | 60 | $\Omega$ |

### 6.2.3 LVDS lane configuration and sync

figure 6-3 LVDS lane configuration and sync

| lane 0 | 3FF | 000 | 000 | SAV | P0 | P1 | P2 | ... | Pn | 3FF | 000 | 000 | EAV | ... |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| active row |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| lane0 | 3FF | 000 | 000 | 200 | P0 | ... | Pn | 3FF | 000 | 000 | 240 |  |  |  |
| blanking row |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| lane0 | 3FF | 000 | 000 | 2 AC | P0 | ... | Pn | 3FF | 000 | 000 | 2 D 8 |  |  | 7750.0.6.6 |

## 7 register tables

The following tables provide descriptions of the device control registers contained in the OV7750/OV7251. For all register enable/disable bits, $\mathrm{ENABLE}=1$ and DISABLE $=0$. The device slave addresses are $0 \times C 0$ for write and $0 \times C 1$ for read.

## 7.1 system control [0x0100-0x010A, 0x3001-0x301F, 0x3023-0x303B, 0x4501]

| system control registers (sheet 1 of 7) |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| address | register name | default value | R/W | description |  |
| 0x0100 | SC_MODE_SELECT | 0x00 | RW | Bit[7:1]: <br> Bit[0]: | Not used <br> Mode select <br> 0 : software_standby <br> 1: Streaming |
| $\begin{aligned} & 0 \times 0101 ~ \\ & 0 \times 0102 \end{aligned}$ | RSVD | - | - | Reserved |  |
| 0x0103 | SC_SOFTWARE_ RESET | 0x00 | RW | Bit[7:1]: Bit[0]: | Not used software_reset <br> 0 : Off <br> 1: On |
| $\begin{aligned} & 0 \times 0104 ~ \\ & 0 \times 0105 \end{aligned}$ | RSVD | - | - | Reserved |  |
| 0x0106 | SC_FAST_STANDBY _CTRL | 0x01 | RW | Bit[7:1]: Bit[0]: | Not used <br> fast_standby <br> 0: Frame completes before mode entry <br> 1: Frame may be truncated before mode entry |
| $\begin{aligned} & \text { 0x0107~ } \\ & 0 \times 0108 \end{aligned}$ | NOT USED | - | - | Not Used |  |
| 0x0109 | SC_SCCB_ID1 | 0xC0 | RW | SCCB ID 0 |  |
| 0x010A | NOT USED | - | - | Not Used |  |
| 0x3001 | SC_REG1 | 0x02 | RW | Bit[7]: <br> Bit[6:5]: <br> Bit[4:0]: | pd_dato_en <br> Output pad drive strength <br> 00: 1 x <br> 01: $2 x$ <br> 10: $3 x$ <br> 11: $4 x$ <br> Debug control |
| $\begin{aligned} & \text { 0x3002~ } \\ & 0 \times 3003 \end{aligned}$ | RSVD | - | - | Reserved |  |

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

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3004 | SC_REG4 | $0 \times 00$ | RW | Bit[7:0]: Debug control |
| $0 \times 3005$ | SC_REG5 | 0x00 | RW | Bit[7:5]: Debug control <br> Bit[3]: <br> Strobe output enable <br>  $0:$ Disable <br> 1: Enable  <br> Bit[2]: PWM output enable <br>  $0:$ Disable <br>  $1:$ Enable <br> Bit[1]: VSYNC output enable <br>  $0:$ Disable <br>  $1:$ Enable <br> Bit[0]: SIOD output enable <br>  <br> $0: ~ D i s a b l e ~$ <br>  $1:$ Enable |
| $\begin{aligned} & 0 \times 3006 ~ \\ & 0 \times 3007 \end{aligned}$ | RSVD | - | - | Reserved |
| 0x3008 | SC_REG8 | 0x00 | RW | Bit[7:0]: Debug control |
| 0x3009 | SC_REG9 | 0x00 | RW | Bit[7:5]: Debug control <br> Bit[3]: Strobe output value, valid only when $0 \times 3027$ [3] is 1 <br> Bit[2]: PWM output value, valid only when $0 \times 3027[2]$ is 1 <br> Bit[1]: VSYNC output value, valid only when $0 \times 3027[1]$ is 1 <br> Bit[0]: SIOD output value, valid only when $0 \times 3027$ [0] is 1 |
| 0x300A | SC_CHIP_ID_HIGH | $0 \times 77$ | R | Chip ID High Byte |
| 0x300B | SC_CHIP_ID_LOW | 0×50 | R | Chip ID Low Byte |
| 0x300C | SC_REG0C | - | R | Revision ID |
| $\begin{aligned} & \text { 0x300D~ } \\ & \text { 0x300E } \end{aligned}$ | RSVD | - | - | Reserved |
| 0x300F | SC_REG0F | 0xF0 | RW | Bit[7]: Debug control <br> Bit[6]: daclk_en <br> Bit[5:4]: Debug control <br> Bit[3]: DACLK same as SCLK <br> 0: DACLK <br> 1: SCLK <br> Bit[2:0]: Debug control |

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

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3010 | SC_REG10 | 0xE1 | RW | Bit[7]: scale_div_man_en <br> Bit[6]: daclk_en <br> Bit[5:4]: daclk_o divider <br>  $00: / 1$ <br>  $01: \quad / 2$ <br>  10: 14 <br> Bit[3]: 11: <br> DACLK same as SCLK  <br> Bit[2:0]: pll_scale_div |
| 0x3011 | RSVD | - | - | Reserved |
| 0x3012 | SC_MIPI_PHY | 0x00 | RW | Bit[7:2]: Debug control <br> Bit[1:0]: mipi_ictl[1:0] |
| 0x3013 | SC_MIPI_PHY | 0x10 | RW | Bit[7:6]: Common mode voltage control for MIPI high speed transmitter <br> Bit[5:4]: Driving strength control of MIPI low power transmitter <br> Bit[3]: bp_d_hs_en_lat <br> Bit[2]: bp_c_hs_en_lat <br> Bit[1]: mipi_pad <br> Bit[0]: Debug control |
| 0x3014 | SC_MIPI_SC_CTRLO | 0x19 | RW |  |
| 0x3015 | SC_MIPI_SC_CTRL1 | 0x10 | RW | Bit[7:0]: MIPI ULPS resume mark1 detect length |

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

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3016 | SC_CLKRST0 | 0xFO | RW | Bit[7]: sclk_fc <br> Bit[6]: Debug control <br> Bit[5]: sclk_aec <br> Bit[4]: sclk_tc <br> Bit[3]: rst_fc <br> Bit[2]: Debug control <br> Bit[1]: rst_aec <br> Bit[0]: rst_tc |
| 0x3017 | SC_CLKRST1 | 0xFO | RW | Bit[7]: sclk_blc <br> Bit[6]: sclk_isp <br> Bit[5]: Debug control <br> Bit[4]: sclk_vfifo <br> Bit[3]: rst_blc <br> Bit[2]: rst_isp <br> Bit[1]: Debug control <br> Bit[0]: rst_vfifo |
| 0x3018 | SC_CLKRST2 | 0xFO | RW | Bit[7]: Debug control <br> Bit[6]: sclk_mipi <br> Bit[5]: sclk_ac <br> Bit[4]: sclk_otp <br> Bit[3]: Debug control <br> Bit[2]: rst_mipi <br> Bit[1]: rst_ac <br> Bit[0]: rst_otp |
| 0x3019 | SC_CLKRST3 | 0xFO | RW | Bit[7:0]: Debug control |
| 0x301A | SC_CLKRST4 | 0xFO | RW | Bit[7:6]: Debug control <br> Bit[5]: pclk_vfifo <br> Bit[4]: pclk_mipi <br> Bit[3]: Debug control <br> Bit[2]: rst_mipi_sc <br> Bit[1:0]: Debug control |
| 0x301B | SC_CLKRST5 | 0xFO | RW | Bit[7]: sclk_src <br> Bit[6]: Debug control <br> Bit[5]: sclk_asram_tst <br> Bit[4]: sclk_snr_sync <br> Bit[3]: rst_src <br> Bit[2]: Debug control <br> Bit[1]: rst_asram_tst <br> Bit[0]: rst_snr_sync |

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

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x301C | SC_CLKRST6 | 0xF2 | RW | $\operatorname{Bit}[7]:$ sclk_bist <br> $\operatorname{Bit[6]:}$ sclk_srb <br> $\operatorname{Bit}[5]:$ sclk_grp <br> $\operatorname{Bit}[4]:$ Debug control <br> $\operatorname{Bit}[3]:$ rst_bist <br> $\operatorname{Bit}[2]:$ Debug control <br> $\operatorname{Bit}[1]:$ rst_grp <br> $\operatorname{Bit[0]:}$ Debug control |
| 0x301D | SC_FREX_RST_ <br> MASKO | $0 \times 00$ | RW | Bit[7:0]: Debug control |
| 0x301E | SC_CLOCK_SEL | 0x0B | RW | Bit[7:4]: Debug control <br> Bit[3]: pclk_source_sel <br>  $0: / 1$ <br>  $1: \quad / 2$ <br> Bit[2:1]: Debug control <br> Bit[0]: sclk2x_source_sel <br>  $0: 11$ <br>  $1: 12$ |
| 0x301F | SC_MISC_CTRL | 0x03 | RW |  |
| 0x3023 | $\begin{aligned} & \text { SC_LOW_PWR_ } \\ & \text { CTRL } \end{aligned}$ | $0 \times 07$ | RW |  |
| $\begin{aligned} & 0 \times 3024 ~ \\ & 0 \times 3025 \end{aligned}$ | RSVD | - | - | Reserved |

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

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| $0 \times 3026$ | SC_REG26 | $0 \times 00$ | RW | Bit[7:5]: Not used <br> Bit[1:0]: Debug control |
| 0x3027 | SC_REG27 | 0x00 | RW | Bit[7]: Debug control <br> Bit[6:5]: io_gpio_sel <br> Bit[3]: io_strobe_sel <br> Bit[2]: io_pwm_sel <br> Bit[1]: io_fsin_sel <br> Bit[0]: io_sda_sel |
| 0x3028 | SC_GP_IO_IN0 | - | R | Bit[7:0]: Debug control |
| 0x3029 | SC_GP_IO_IN1 | - | R |  |
| 0x302A | SC_GP_IO_IN2 | - | R | $\operatorname{Bit}[7: 6]:$ Not used <br> $\operatorname{Bit}[5]:$ Strobe input value <br> $\operatorname{Bit}[3]:$ FSIN input value <br> $\operatorname{Bit}[2]:$ PWM input value <br> Bit[1:0]: Debug control |
| 0x302B | SC_SCCB_ID2 | $0 x E 0$ | RW | SCCB ID 2 |
| 0x302C | SC_AUTO_SLEEP_ PERIOD | $0 \times 01$ | RW | Bit[7:0]: Auto sleep period[31:24] |
| 0x302D | SC_AUTO_SLEEP_ PERIOD | 0x00 | RW | Bit[7:0]: Auto sleep period[23:16] |
| 0x302E | SC_AUTO_SLEEP_ PERIOD | 0x00 | RW | Bit[7:0]: Auto sleep period[15:8] |
| 0x302F | SC_AUTO_SLEEP_ PERIOD | $0 \times 00$ | RW | Bit[7:0]: Auto sleep period[7:0] |
| 0x3030 | SC_LP_CTRL0 | 0x03 | RW | Bit[7]: auto_sleep_en <br> Bit[6]: gpio_sel <br>  $0: \quad$ Not used <br>  $1: \quad$ Sleep can be read by GPIO <br>  $\quad(Y 9)$ <br> Bit[5:4]: Debug control <br> $\operatorname{Bit[3:0]:~}$ frame_on_num |
| 0x3031 | RSVD | - | - | Reserved |
| 0×3032 | SC_IO_OEN_SLEEP | $0 \times 03$ | RW | Output Control in Sleep Mode |

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

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3033 | SC_IO_OEN_SLEEP | 0xFF | RW | Bit[7]: Debug control <br> Bit[6:5]: io_gpio_oen_sleep[1:0] <br> Bit[3]: io_strobe_oen_sleep <br> Bit[2]: io_pwm_oen_sleep <br> Bit[1]: io_fsin_oen_sleep <br> Bit[0]: io_sda_oen_sleep |
| 0x3034 | RSVD | - | - | Reserved |
| 0x3035 | $\begin{aligned} & \text { SC_IO_Y_OEN_ } \\ & \text { PWDN } \end{aligned}$ | 0x03 | RW | Bit[7:2]: Not used <br> Bit[1:0]: Debug control |
| 0x3036 | $\begin{aligned} & \text { SC_IO_Y_OEN_ } \\ & \text { PWDN } \end{aligned}$ | 0xFF | RW | Bit[7:0]: Not used <br> Bit[1:0]: Debug control |
| 0x3037 | SC_R37 | 0xF0 | RW | Bit[7]: Debug control <br> Bit[6]: sclk_strobe <br> Bit[5]: sclk_fmt <br> Bit[4]: sclk_pwm <br> Bit[3]: Debug control <br> Bit[2]: rst_strobe <br> Bit[1]: rst_fmt <br> Bit[0]: rst_pwm |
| 0x3038 | SC_REG38 | 0x50 | RW | Bit[7]: Debug control <br> Bit[6]: grp_clk_rst sleep en <br> Bit[5]: clk_grp_sel <br> 0: PADCLK <br> 1: SCLK <br> Bit[4]: pwr_switch_pad_clk_en <br> Bit[3]: clk_src_11x_02x <br> 0: $2 x$ <br> 1: $1 x$ <br> Bit[2:0]: Debug control |
| $\begin{aligned} & 0 \times 3039 ~ \\ & 0 \times 303 A \end{aligned}$ | RSVD | - | - | Reserved |
| 0x303B | SC_REG3B | 0x00 | RW | Bit[7:2]: Debug control Bit[1]: sccb_pgm_id_en Bit[0]: sccb_id2_nack_en |
| 0x4501 | SC_REG1501 | 0x08 | RW | Bit[7]: Debug control <br> Bit[6]: Option for power saving during vertical blanking period <br> Bit[0]: Debug control |

### 7.2 PLL control [0x3080-0x3083, 0x3098-0x309F, 0x30B0-0x30B6]

table 7-2 PLL control registers

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- | :--- |

### 7.3 SCCB and group hold control [0x3100-0x3106, 0x31FF - 0x320F]

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

|  |  | default |  |  |
| :---: | :---: | :---: | :---: | :---: |
| address | register name | value | R/W | description |
| $\begin{aligned} & 0 \times 3100 \sim \\ & 0 \times 3105 \end{aligned}$ | DEBUG CTRL | - | - | Debug Control |
| $0 \times 3106$ | SB_SRB_CTRL | 0x12 | RW | $\begin{array}{ll} \text { Bit[7]: } & \text { Enable XVCLK (must be } 1 \text { for video } \\ \text { streaming) } \\ \text { Bit[6:0]: } & \text { Debug control } \end{array}$ |
| 0x31FF | SB_SWITCH | $0 \times 01$ | RW | Bit[7:1]: Debug control <br> Bit[0]: SCCB slave select <br> 0: $\quad$ Select SCCB slave which requires EXTCLK <br> 1: Select SCCB slave which does not require EXTCLK |
| 0x3200 | GROUP ADRO | $0 \times 00$ | RW | Start Address of Group 0 Buffer <br> Actual start address is $\left\{0 \times 3200[3: 0], 4{ }^{\prime} \mathrm{h} 0\right\}$ |
| 0x3201 | GROUP ADR1 | 0x04 | RW | Start Address of Group 1 Buffer Actual start Address Is $\{0 \times 3201$ [3:0], 4'h0\} |
| $\begin{aligned} & 0 \times 3202 ~ \\ & 0 \times 3203 \end{aligned}$ | RSVD | - | - | Reserved |
| 0x3204 | GROUP LEN0 | - | R | Length of Group0 |
| 0x3205 | GROUP LEN1 | - | R | Length of Group1 |
| $\begin{aligned} & 0 \times 3206 ~ \\ & 0 \times 3207 \end{aligned}$ | RSVD | - | - | Reserved |
| 0x3208 | GROUP ACCESS | - | W |  |
| 0x3209 | GROUPO PERIOD | $0 \times 00$ | RW | Number of Frames to Stay in Group0 |
| 0x320A | GROUP1 PERIOD | 0x00 | RW | Number of Frames to Stay in Group1 |

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

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- |

## 7.4 manual AWB gain control [0x3400-0x3406]

table 7-4 manual AWB_gain registers

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- | :--- |

## 7.5 manual AEC/AGC[0x3500-0x350B, 0x5D00-0x5D01,0x5F00-0x5F05]

| address | register name | default value | R/W | description |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0x3500 | AEC EXPO | 0x00 | RW | $\begin{aligned} & \text { Exposure } \\ & \text { Bit[7:4]: } \\ & \text { Bit[3:0]: } \end{aligned}$ | Not used <br> Exposure[15:12] |
| 0x3501 | AEC EXPO | 0x00 | RW | $\begin{aligned} & \text { Exposure } \\ & \text { Bit[7:0]: } \end{aligned}$ | Exposure[11:4] <br> Minimum exposure time is 1 row period. Maximum exposure time is frame length - 20 row periods, where frame length is set by registers $\{0 \times 380 \mathrm{E}, 0 \times 380 \mathrm{~F}\}$. |
| 0x3502 | AEC EXPO | 0x00 | RW | Exposure Bit[7:4]: <br> Bit[3:0]: | Exposure[3:0] <br> Minimum exposure time is 1 row period.Maximum exposure time is frame length - 20 row periods, where frame length is set by registers $\{0 \times 380 \mathrm{E}, 0 \times 380 \mathrm{~F}\}$. Debug control |


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

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| $0 \times 3505$ | MAN SNR GAIN | 0x00 | RW | Manual Sensor Gain <br> Bit[7:0]: Manual sensor gain[7:0] |
| $\begin{aligned} & 0 \times 3506 ~ \\ & 0 \times 3508 \end{aligned}$ | DEBUG CTRL | - | - | Debug Control |
| 0x3509 | AEC GAIN CONVERT | 0x10 | RW | AEC Manual Mode Control <br> Bit[7:5]: Debug control <br> Bit[4]: Gain convert enable <br> 0 : Use sensor gain, $\{0 \times 350 \mathrm{~A}, 0 \times 350 \mathrm{~B}\}$, as sensor gain <br> 1: Use real gain, $\{0 \times 350 \mathrm{~A}, 0 \times 350 \mathrm{~B}\}$, as linear gain <br> Bit[3]: Sensor gain manual enable <br> (BLC cannot be triggered by these gain registers) <br> 0: Disable <br> 1: Manual control for $\{0 \times 3504,0 \times 3505\}$ <br> Bit[2]: Debug control <br> Bit[1]: Gain change delay option <br> 0 : gain_change delay 1 frame <br> 1: gain_change no delay <br> Bit[0]: Debug control |
| 0x350A | AEC AGC ADJ | 0x00 | RW | Gain Output to Sensor <br> Bit[7:2]: Not used <br> Bit[1:0]: Gain[9:8] |
| 0x350B | AEC AGC ADJ | 0x10 | RW | Gain Output to Sensor <br> Bit[7:0]: Gain[7:0] <br> Gain $=$ register 0x350B / 0x10 for linear gain, or <br> Gain $=($ register $0 \times 350 \mathrm{~B}[7]+1) \times($ register $0 \times 350 \mathrm{~B}[6]+1) \times($ register $0 \times 350 \mathrm{~B}[5]+1)$ $\times($ register $0 \times 350 \mathrm{~B}[4]+1) \times($ register $0 \times 350 \mathrm{~B}[3: 0] / 16+1$ ) for sensor gain. <br> Rev. 1A and Rev. 1B must choose sensor gain by setting register 0x3509[4] to 0, and Rev.1C and Rev.1D must choose linear gain by setting register $0 \times 3509[4]$ to 1 . |
| 0x5D00 | GAIN FORMAT 00 | 0x07 | RW | Bit[7:4]: Debug control <br> Bit[3:0]: Analog gain bit control |
| 0x5D01 | GAIN FORMAT 01 | 0x00 | RW | Bit[7:4]: Debug control <br> Bit[3:0]: Digital gain bit control |

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

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x5F00 | DIG_COMP CTRL $00$ | $0 \times 18$ | RW |  |
| 0x5F01 | RSVD | - | - | Reserved |
| 0x5F02 | $\begin{aligned} & \text { DIG_COMP CTRL } \\ & 02 \end{aligned}$ | 0x02 | RW | Bit[7:2]: Debug control <br> Bit[1:0]: dig_comp_man[9:8] |
| 0x5F03 | $\begin{aligned} & \text { DIG_COMP CTRL } \\ & 03 \end{aligned}$ | $0 \times 00$ | RW | Bit[7:0]: dig_comp_man[7:0] |
| 0x5F04 | $\begin{aligned} & \text { DIG_COMP CTRL } \\ & 04 \end{aligned}$ | - | R | Bit[7:2]: Not used <br> Bit[1:0]: dig_comp_auto_i[9:8] |
| 0x5F05 | $\begin{aligned} & \text { DIG_COMP CTRL } \\ & 05 \end{aligned}$ | - | R | Bit[7:0]: dig_comp_auto_i[7:0] |

## 7.6 analog control [0×3600-0x3684]

| address | register name | default value | R/W | description |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{aligned} & 0 \times 3600 ~ \\ & 0 \times 3635 \end{aligned}$ | ANALOG REGISTERS | - | - | Analog Control Registers |  |
| 0x3636 | ANA_CTRL_36 | 0x00 | RW | Bit[7:4]: Bit[3]: Bit[2:0]: | Analog control <br> Internal regulator disable <br> 0 : Enable internal regulator <br> 1: Disable internal regulator <br> Analog control |

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

|  |  | default |  |  |
| :---: | :---: | :---: | :---: | :---: |
| address | register name | value | R/W | description |
| $\begin{aligned} & 0 \times 3637 ~ \\ & 0 \times 3661 \end{aligned}$ | ANALOG REGISTERS | - | - | Analog Control Registers |
| 0x3662 | ANA_CORE_2 | $0 \times 01$ | RW | Bit[7:2]: Debug control <br> Bit[1]: <br>  <br>  <br> RAW8 enable <br> $0: ~ R A W 10 ~$ <br> Bit[0]: 1: RAW8 <br> Debug control  |
| $\begin{aligned} & 0 \times 3663 ~ \\ & 0 \times 3665 \end{aligned}$ | ANALOG REGISTERS | - | - | Analog Control Registers |
| 0x3666 | ANA_CORE_6 | 0x0A | RW | FSIN/VSYNC Input and Output select <br> Bit[7:4]: Output select <br> 0x0: VSYNC <br> Others: For debug only <br> Bit[3:0]: Internal frame sync input select <br> 0x0: From FSIN pin, used for both frame sync and frame trigger function <br> 0xA: Fixed value 0 <br> Others: For debug purposes |
| $\begin{aligned} & 0 \times 3667 ~ \\ & 0 \times 3684 \end{aligned}$ | ANALOG REGISTERS | - | - | Analog Control Registers |

## 7.7 sensor control [0x3700-0x37AF]

table 7-7 sensor control registers

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- |
| $0 \times 3700 \sim$ <br> $0 \times 37 A 7$ | SENSOR CONTROL <br> REGISTERS | - | - | Sensor Control Registers |
| $0 \times 37 A 8$ | FIFO_CTRL0_H | $0 \times 00$ | RW | Internal FIFO Control |
| $0 \times 37 A 9$ | FIFO_CTRL0_L | $0 \times 60$ | RW | Internal FIFO Control |
| 0x37AA~ <br> $0 \times 37 A F$ | SENSOR CONTROL <br> REGISTERS | - | - | Sensor Control Registers |

## 7.8 timing control [0x3800-0x3835, 0x3837]

table 7-8 timing control registers (sheet 1 of 4)

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3800 | TIMING_X_ADDR_START | 0x00 | RW | Array Horizontal Start Point High Byte |
| 0x3801 | TIMING_X_ADDR_START | 0x04 | RW | Array Horizontal Start Point Low Byte |
| 0x3802 | TIMING_Y_ADDR_START | 0x00 | RW | Array Vertical Start Point High Byte |
| 0x3803 | TIMING_Y_ADDR_START | 0x04 | RW | Array Vertical Start Point Low Byte |
| 0x3804 | TIMING_X_ADDR_END | 0x02 | RW | Array Horizontal End Point High Byte |
| 0x3805 | TIMING_X_ADDR_END | 0x8B | RW | Array Horizontal End Point Low Byte |
| 0x3806 | TIMING_Y_ADDR_END | 0x01 | RW | Array Vertical End Point High Byte |
| 0x3807 | TIMING_Y_ADDR_END | 0xEB | RW | Array Vertical End Point Low Byte |
| 0x3808 | TIMING_X_OUTPUT_SIZE | 0x02 | RW | ISP Horizontal Output Width High Byte |
| 0x3809 | TIMING_X_OUTPUT_SIZE | 0x80 | RW | ISP Horizontal Output Width Low Byte |
| 0x380A | TIMING_Y_OUTPUT_SIZE | $0 \times 01$ | RW | ISP Vertical Output Height High Byte |
| 0x380B | TIMING_Y_OUTPUT_SIZE | $0 x E 0$ | RW | ISP Vertical Output Height Low Byte |
| 0x380C | TIMING_HTS | 0x03 | RW | Total Horizontal Timing Size High Byte |
| 0x380D | TIMING_HTS | 0x04 | RW | Total Horizontal Timing Size Low Byte |
| 0x380E | TIMING_VTS | 0x02 | RW | Total Vertical Timing Size High Byte |
| 0x380F | TIMING_VTS | 0x04 | RW | Total Vertical Timing Size Low Byte |
| 0x3810 | TIMING_ISP_X_WIN | 0x00 | RW | ISP Horizontal Windowing Offset High Byte |
| $0 \times 3811$ | TIMING_ISP_X_WIN | 0x00 | RW | ISP Horizontal Windowing Offset Low Byte |
| 0x3812 | TIMING_ISP_Y_WIN | 0x00 | RW | ISP Vertical Windowing Offset High Byte |
| $0 \times 3813$ | TIMING_ISP_Y_WIN | 0x00 | RW | ISP Vertical Windowing Offset Low Byte |
| $0 \times 3814$ | TIMING_X_INC | $0 \times 11$ | RW | Bit[7:4]: x_odd_inc <br> Bit[3:0]: x_even_inc |
| $0 \times 3815$ | TIMING_Y_INC | $0 \times 11$ | RW | Bit[7:4]: y_odd_inc <br> Bit[3:0]: y_even_inc |
| $\begin{aligned} & 0 \times 3816 \sim \\ & 0 \times 381 F \end{aligned}$ | DEBUG CTRL | - | - | Debug Control |

table 7-8 timing control registers (sheet 2 of 4)

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- | :--- |

table 7-8 timing control registers (sheet 3 of 4)

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- |

table 7-8 timing control registers (sheet 4 of 4)

| address | register name | default <br> value | R/W | Rescription |
| :--- | :--- | :--- | :--- | :--- |

### 7.9 PWM and strobe control [0x3B80-0x3B97]

table 7-9 PWM and strobe control registers (sheet 1 of 2 )

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3B80 | LED_PWM_REG00 | $0 \times 10$ | RW |  |
| 0x3B81 | LED_PWM_REG01 | 0xA5 | RW | Bit[7:0]: strobe_frame pattern for sequential 8 frames <br> 0 : Off for 8 sequential frames <br> 1: On |

table 7-9 PWM and strobe control registers (sheet 2 of 2 )

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3B82 | LED_PWM_REG02 | $0 \times 10$ | RW | Bit[7:0]: pwm_freq_div_cycle_reg[15:8] |
| 0x3B83 | LED_PWM_REG03 | $0 \times 00$ | RW | Bit[7:0]: pwm_freq_div_cycle_reg[7:0] |
| 0x3B84 | LED_PWM_REG04 | $0 \times 08$ | RW | Bit[7:0]: pwm_duty_cycle_reg[15:8] |
| 0x3B85 | LED_PWM_REG05 | $0 \times 00$ | RW | Bit[7:0]: pwm_duty_cycle_reg[7:0] |
| 0x3B86 | LED_PWM_REG06 | $0 \times 01$ | RW | Bit[7:0]: low_limit[15:8] |
| 0x3B87 | LED_PWM_REG07 | $0 \times 00$ | RW | Bit[7:0]: low_limit[7:0] |
| 0x3B88 | LED_PWM_REG08 | 0x00 | RW | Bit[7]: strobe_frame_sign_bit <br>  $0: \quad$ Positive delay <br>  1: Negative delay <br> Bit[6:0]: strobe_frame_shift[31:24] |
| 0x3B89 | LED_PWM_REG09 | 0x00 | RW | Bit[7:0]: strobe_frame_shift[23:16] |
| 0x3B8A | LED_PWM_REG0A | $0 \times 00$ | RW | Bit[7:0]: strobe_frame_shift[15:8] |
| 0x3B8B | LED_PWM_REGOB | 0x05 | RW | Bit[7:0]: strobe_frame_shift[7:0] |
| 0x3B8C | LED_PWM_REG0C | 0x00 | RW | Bit[7:0]: strobe_frame_span[31:24] |
| 0x3B8D | LED_PWM_REGOD | 0x00 | RW | Bit[7:0]: strobe_frame_span[23:16] |
| 0x3B8E | LED_PWM_REG0E | 0x00 | RW | Bit[7:0]: strobe_frame_span[15:8] |
| 0x3B8F | LED_PWM_REG0F | $0 \times 1 \mathrm{~A}$ | RW | Bit[7:0]: strobe_frame_span[7:0] |
| 0x3B90 | LED_PWM_REG10 | $0 \times 01$ | RW | Bit[7:0]: r_strobe_row_st[15:8] |
| 0x3B91 | LED_PWM_REG11 | 0xB4 | RW | Bit[7:0]: r_strobe_row_st[7:0] |
| 0x3B92 | LED_PWM_REG12 | $0 \times 00$ | RW | Bit[7:0]: r_strobe_cs_st[15:8] |
| 0x3B93 | LED_PWM_REG13 | $0 \times 10$ | RW | $\operatorname{Bit}[7: 0]: \quad$ r_strobe_cs_st[7:0] |
| 0x3B94 | LED_PWM_REG14 | 0x05 | RW | Bit[7:0]: step_onerow_man[15:8] |
| 0x3B95 | LED_PWM_REG15 | 0xF2 | RW | Bit[7:0]: step_onerow_man[7:0] |
| 0x3B96 | LED_PWM_REG16 | 0x40 | RW | Bit[7]: r_strobe_frm_pwen <br> Bit[6]: r_strobe_frm_pwst <br> Bit[5]: r_strobe_pol <br> Bit[4]: r_strobe_step_pix <br> Bit[3]: r_step_onerow_precision_man <br> Bit[2:0]: r_strobe_st_opt |
| 0x3B97 | LED_PWM_REG17 | $0 \times 00$ | RW | Bit[7:0]: Debug control |

### 7.10 low power mode control [0x3C00-0x3C0F, 0x4A47-0x4A49]

table 7-10 low power mode control registers (sheet 1 of 2 )

|  |  | default |  |  |
| :---: | :---: | :---: | :---: | :---: |
| address | register name | value | R/W | description |
| 0x3C00 | LOWPWR00 | 0x89 | RW | Bit[7:0\}: Debug control for low power mode Maintains default value of $0 \times 89$ all the time |
| 0x3C01 | LOWPWR01 | $0 \times A B$ | RW | Bit[7:0\}: Power control options <br> 0x63: Low power mode <br> $0 x A B$ : Normal mode |
| 0x3C02 | LOWPWR02 | $0 \times 01$ | RW | Bit[1\}: Idle phase enable <br> Bit[0\}: $\quad$ Streaming phase enable <br> x0: Not allowed <br> 01: Normal streaming mode <br> 11: Enable low power streaming mode |

Low Power Mode Control
Bit[7]: Trigger for internal trigger snapshot mode
A rising edge on $0 \times 3 \mathrm{C} 03[7]$ wakes the sensor up and streams out $\{0 \times 3404$, $0 \times 3405\}$ frames
Bit[6:0]: Mode control
0x00: Low frame rate streaming mode (i.e., repeating the sequence of streaming $\{0 \times 3404,0 \times 3405\}$ frames and then sleeping $\{0 \times 3 \mathrm{C} 06,0 \times 3 \mathrm{C} 07\}$ frame)
0x17: External trigger snapshot mode A rising edge on FSIN pin wakes the sensor up and streams out \{0×3404, 0×3405\} frames
0x03: Internal trigger snapshot mode A rising edge on 0x3C03[7] wakes the sensor up and streams out $\{0 \times 3404,0 \times 3405\}$ frames
Others: For debug only

| $0 \times 3$ C04 | LOWPWR04 | $0 \times 00$ | RW | Bit[7:0\}: Number of active frames[15:8] |
| :--- | :--- | :--- | :--- | :--- |
| $0 \times 3$ C05 | LOWPWR05 | $0 \times 03$ | RW | Bit[7:0\}: Number of active frames[7:0] |
| $0 \times 3$ C06 | LOWPWR06 | $0 \times 00$ | RW | Bit[7:0\}: Number of idle frames[15:8] |
| $0 \times 3 C 07$ | LOWPWR07 | $0 \times 05$ | RW | Bit[7:0\}: Number of idle frames[7:0] |
| $0 \times 3 C 08 \sim$ <br> $0 \times 3 C 0 B$ | NOT USED | - | - | Not Used |

table 7-10 low power mode control registers (sheet 2 of 2 )

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x3C0C | LOWPWR0C | 0x00 | RW | Bit[7:0\}: Row period[15:8] in units of input clock period |
| 0x3C0D | LOWPWR0D | 0x00 | RW | Bit[7:0\}: Row period[7:0] in units of input clock period |
| 0x3C0E | LOWPWR0E | 0x00 | RW | Bit[7:0\}: Number of rows per base frame[15:8] usually set to the same value as \{0×380E, 0x380F\} |
| 0x3C0F | LOWPWR0F | 0x00 | RW | $\operatorname{Bit}[7: 0\}$ : Number of rows per base frame[7:0] usually set to the same value as \{0x380E, 0x380F\} |
| $\begin{aligned} & \text { 0x4A47~ } \\ & \text { 0x4A49 } \end{aligned}$ | LOWPWR CTRL REGISTERS | - | RW | Low Power Control Registers |

### 7.11 OTP control [0x3D80-0x3D87, 0x3D8B]

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

| address | register name | default <br> value | R/W | Rescription |
| :--- | :--- | :--- | :--- | :--- | :--- |

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

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- | :--- |

### 7.12 BLC control [0×4000-0x4051]

table 7-12 BLC control registers (sheet 1 of 4)

table 7-12 BLC control registers (sheet 2 of 4)

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4002 | BLC AUTO | 0xC5 | RW | Bit[7]: Format change enable <br> $0: \quad$ BLC keep same after format change <br>  1: BLC will redo after format change <br> Bit[6]: BLC auto enable <br>  <br> $0: \quad$ Get the black level from manual <br>  <br>  <br>  <br> 1: $\quad$ register <br> Calculate the black level from auto <br> Bit[5:0]: <br>  <br> Reset frame number <br> Frames BLC continue after reset |
| 0x4003 | BLC FREEZE | 0x01 | RW | Bit[7]: Manual redo enable <br> Bit[6]: Freeze enable <br>  $0: \quad$ Normal <br>  1: $\quad$ BLC black level will not update; <br>  Priority lower than always update <br> Bit[5:0]: Manual frame number <br>  <br> BLC redo frame number |
| 0x4004 | BLC NUM | 0x04 | RW | Bit[7:6]: Debug control <br> Bit[5:0]: Number of black lines used |
| 0x4005 | BLC MAN CTRL | 0x00 | RW | Bit[7:6]: Debug control <br> Bit[5]: One line slope mode <br> Bit[4]: Median filter option <br> 0 : Image will not pass median filter <br> 1: Image will pass median filter <br> Bit[3]: blc_man_1_en <br> Apply one channel offset ( $0 \times 400 \mathrm{C}$, <br> $0 \times 400 \mathrm{D}$ ) to all manual BLC channels <br> Bit[2]: Release black line enable <br> 0: Disable <br> 1: Enable <br> Bit[1]: blc_always_up_en <br> 0: BLC will continue several frames after reset; after that, it will no longer change until gain changes (controlled by bit[0]), or format changes (controlled by register $0 \times 4002[7]$ ). <br> 1: BLC always updates in every frame <br> Bit[0]: Debug control <br> 0 : agc_change generated by BLC pre <br> 1: agc_change from system |
| 0x4006 | DEBUG CTRL | - | - | Debug Control |

table 7-12 BLC control registers (sheet 3 of 4)

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4007 | BLC WIN | 0x20 | RW | Bit[7:6]: Debug control <br> Bit[5]: r_gain_change_enable <br> 0 : Disable <br> 1: Enable <br> Bit[4:3]: Window selection <br> 00: Full image <br> 01: Windows do not contain the first 16 pixels and the end 16 pixels <br> 10: Windows do not contain the first $1 / 16$ image and the end $1 / 16$ image <br> 11: Windows do not contain the first $1 / 8$ image and the end $1 / 8$ image <br> Bit[2:0]: Bypass mode <br> 000: Bypass data_i after limit bits <br> 001: Bypass data_i[9:0]: <br> 010: Bypass data_i[10:1]: <br> 011: Bypass debug data bbrr <br> 100: Bypass debug data gggg <br> 101~111: Not used |
| 0x4008 | BLC STABLE RANGE | 0xFF | RW | $\begin{array}{ll}\mathrm{Bit}[7]: & \text { BLC stable range option enable } \\ \text { Bit[6]: } & \text { BLC stable range value } \\ \text { Bit[5:0]: } & \text { BLC stable range value }\end{array}$ |
| 0x4009 | BLC TARGET | 0x10 | RW | Black Level Target |
| $\begin{aligned} & \text { 0x400A~ } \\ & \text { 0x400B } \end{aligned}$ | DEBUG CTRL | - | - | Debug Control |
| 0x400C | BLC MAN LEVELO | 0x00 | RW | Bit[7:4]: Debug control <br> Bit[3:0]: BLC manual level channel 0[11:8] |
| 0x400D | BLC MAN LEVELO | 0x00 | RW | Bit[7:0]: BLC manual level channel 0[7:0] |
| 0x400E | BLC MAN LEVEL1 | 0x00 | RW | Bit[7:4]: Debug control <br> Bit[3:0]: BLC manual level channel 1[11:8] |
| 0x400F | BLC MAN LEVEL1 | 0x00 | RW | Bit[7:0]: BLC manual level channel 1[7:0] |
| 0x4010 | BLC MAN LEVEL2 | 0x00 | RW | Bit[7:4]: Debug control <br> Bit[3:0]: BLC manual level channel 2[11:8] |
| 0x4011 | BLC MAN LEVEL2 | 0x00 | RW | Bit[7:0]: BLC manual level channel 2 [ $7: 0]$ |
| 0x4012 | BLC MAN LEVEL3 | 0x00 | RW | Bit[7:4]: Debug control <br> Bit[3:0]: BLC manual level channel 3[11:8] |
| 0x4013 | BLC MAN LEVEL3 | 0x00 | RW | Bit[7:0]: BLC manual level channel 3[7:0] |
| $\begin{aligned} & 0 \times 4014 ~ \\ & 0 \times 402 \mathrm{~B} \end{aligned}$ | NOT USED | - | - | Not Used |
| 0x402C | BLC LEVEL 0 | - | R | Bit[7:6: Debug control <br> Bit[5:0]: Black level channel 0[13:8] |

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


### 7.13 frame control [0x4240-0x4244]

table 7-13 frame registers

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4240 | FC CTRLO | $0 \times 00$ | RW | $\operatorname{Bit[7:4]:}$ Not used <br> $\operatorname{Bit}[3]:$ sof_sel <br> $\operatorname{Bit}[2]:$ fcnt_eof_sel <br> $\operatorname{Bit[1]:~}$ fcnt_mask_dis <br> $\operatorname{Bit[0]:}$ fcnt_reset |
| $0 \times 4241$ | FRAME ON NUM | 0x00 | RW | Bit[7:4]: Not used <br> Bit[3:0]: Frame on number |
| $0 \times 4242$ | FRAME OFF NUM | 0x00 | RW | Bit[7:4]: Not used <br> Bit[3:0]: Frame off number |
| 0x4243 | FC CTRL3 | 0x00 | RW | Bit[7]: Not used <br> Bit[6]: rblue_mask_dis <br> Bit[5]: data_mask_dis <br> Bit[4]: valid_mask_dis <br> Bit[3]: href_mask_dis <br> Bit[2]: eof_mask_dis <br> Bit[1]: sof_mask_dis <br> Bit[0]: all_mask_dis |
| $0 \times 4244$ | FRAME COUNTER | - | R | Bit[7:0]: Frame count |

### 7.14 format control [0x4300-0x4307, 0x4310-0x4316, 0x4320-0x4329]

table 7-14 format control registers (sheet 1 of 3)

| address | register name | default <br> value | R/W | description |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $0 \times 4300$ | DATA_MAX H | 0xFF | RW | Bit[7:0]: | Data max[9:2] |
| $0 \times 4301$ | DATA_MIN H | $0 \times 00$ | RW | Bit[7:0]: | Data min[9:2] |
| $0 \times 4302$ | CLIP L | 0x0C | RW | Bit[7:4]: <br> Bit[3:2]: <br> Nit[1:0]: used <br> Data max[1:0] <br> Data min[1:0] |  |

table 7-14 format control registers (sheet 2 of 3)

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4303 | FORMAT CTRL3 | 0x00 | RW | Bit[7]: r_inc_en <br> Bit[6]: r_inc_pattern <br> Bit[5]: r_pad_Isb <br> Bit[4]: r_bar_mux <br> Bit[3]: r_bar_en <br> Bit[2]: r_bit_shift_tst_en <br> Bit[1]: r_tst_bit8 <br> Bit[0]: r_bit_shift_tst_md |
| 0x4304 | FORMAT CTRL4 | $0 \times 08$ | RW | Bit[7]: Not used <br> Bit[6:4]: data_bit_swap <br> Bit[3]: tst_full_win <br> Bit[2:0]: bar_pad |
| $\begin{aligned} & 0 \times 4305 ~ \\ & 0 \times 4306 \end{aligned}$ | RSVD | - | - | Reserved |
| $0 \times 4307$ | EMBED CTRL | $0 \times 30$ | RW | Bit[7:0]: Debug control |
| $0 \times 4310$ | DEBUG CTRL | - | - | Debug Control |
| $0 \times 4311$ | VSYNC_WIDTH_H | $0 \times 04$ | RW | Bit[7:0]: VSYNC width[15:8] <br> (in terms of pixel numbers) |
| $0 \times 4312$ | VSYNC_WIDTH_L | 0x00 | RW | Bit[7:0]: VSYNC width[7:0] <br> (in terms of pixel numbers) |
| $0 \times 4313$ | VSYNC_CTRL | 0x00 | RW | Bit[7:5]: Not used <br> Bit[4]: VSYNC polarity <br> Bit[3:2]: VSYNC output select <br> Bit[1]: VSYNC mode 3 <br> Bit[0]: VSYNC mode 2 |
| $0 \times 4314$ | VSYNC_DELAY1 | 0x00 | RW | Bit[7:0]: VSYNC trigger to VSYNC delay[23:16] |
| 0x4315 | VSYNC_DELAY2 | $0 \times 01$ | RW | Bit[7:0]: VSYNC trigger to VSYNC delay[15:8] |
| 0×4316 | VSYNC_DELAY3 | $0 \times 00$ | RW | Bit[7:0]: VSYNC trigger to VSYNC delay[7:0] |
| $0 \times 4320$ | TST PATTERN CTRL | 0x80 | RW | Bit[7:6]: pixel_order <br>  $00:$ <br>  01: <br>  RR/BG <br>  $10:$ <br>  11: GG/GR <br> Bit[5]: byte_swap <br> Bit[4:2]: Debug control <br> Bit[1]: solid_color_en <br> Bit[0]: Debug control |
| 0x4321 | RSVD | - | - | Reserved |
| 0x4322 | SOLID_B_H | $0 \times 00$ | RW | Blue Value for Solid Color Test Pattern MSB |
| 0x4323 | SOLID_B_L | $0 \times 00$ | RW | Blue Value for Solid Color Test Pattern LSB |

table 7-14 format control registers (sheet 3 of 3 )

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- |
| $0 \times 4324$ | SOLID_GB_H | $0 \times 00$ | RW | Gb Value for Solid Color Test Pattern MSB |
| $0 \times 4325$ | SOLID_GB_L | $0 \times 00$ | RW | Gb Value for Solid Color Test Pattern LSB |
| $0 \times 4326$ | SOLID_R_H | $0 \times 00$ | RW | Red Value for Solid Color Test Pattern MSB |
| $0 \times 4327$ | SOLID_R_L | $0 \times 00$ | RW | Red Value for Solid Color Test Pattern LSB |
| $0 \times 4328$ | SOLID_GR_H | $0 \times 00$ | RW | Gr Value for Solid Color Test Pattern MSB |
| $0 \times 4329$ | SOLID_GR_L | $0 \times 00$ | RW | Gr Value for Solid Color Test Pattern LSB |

### 7.15 VFIFO control [0x4600-0x4604]

table 7-15 VFIFO control registers

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4600 | READ START H | 0x00 | RW | VFIFO Read Start Point High Byte |
| 0x4601 | READ START L | 0x28 | RW | VFIFO Read Start Point Low Byte |
| 0x4602 | VFIFO CTRL2 | 0xFO | RW | Bit[7:4]: r_rm <br> Bit[3]: r_test1 <br> Bit[2]: Debug control <br> Bit[1]: Frame reset enable <br> Bit[0]: RAM bypass enable |
| 0x4603 | VFIFO CTRL3 | 0x11 | RW | Bit[7:5]: Debug control <br> Bit[4]: man_start_mode <br> Bit[3:2]: Debug control <br> Bit[1:0]: start_offset |
| 0x4604 | VFIFO STATUS | - | R | Bit[7:4]: Debug control <br> Bit[3]: ram_full <br> Bit[2]: ram_empty <br> Bit[1]: fo_full <br> Bit[1:0]: fo_empty |

### 7.16 MIPI top [0x4800 - 0x4806, 0x4810 - 0x4849, 0x4850-0x4854, 0x4860-0x4865]

| address | register name | default value | R/W | description |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0x4800 | MIPI CTRL 00 | 0x44 | RW | MIPI Control 00 |  |
|  |  |  |  | Bit[7]: <br> Bit[6]: | Debug control |
|  |  |  |  |  | ck_mark1_en |
|  |  |  |  |  | 0 0 Not used |
|  |  |  |  |  | 1: Enable clock lane mark1 when resume |
|  |  |  |  | Bit[5]: | Clock lane gate enable |
|  |  |  |  |  | 0 : Clock lane is free running |
|  |  |  |  |  | 1: Gate clock lane when no packet to transmit |
|  |  |  |  | Bit[4]: | Line sync enable |
|  |  |  |  |  | 0 : Do not send line short packet for each line |
|  |  |  |  | $\begin{aligned} & \text { Bit[3]: } \\ & \text { Bit[2]: } \end{aligned}$ | 1: Send line short packet for each line |
|  |  |  |  |  | Debug control |
|  |  |  |  |  | Idle status |
|  |  |  |  |  | 0 : MIPI bus will be LPOO when no packet to transmit |
|  |  |  |  |  | 1: MIPI bus will be LP11 when no packet to transmit |
|  |  |  |  | $\begin{aligned} & \text { Bit[1]: } \\ & \text { Bit[0]: } \end{aligned}$ | Debug control |
|  |  |  |  |  | clk_lane_dis |
|  |  |  |  |  | 0 : Not used |
|  |  |  |  |  | 1: Set clock lane to LP mode manually |

table 7-16 MIPI top control registers (sheet 2 of 9)

table 7-16 MIPI top control registers (sheet 3 of 9)

| address | register name | default value | R/W | description |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0x4802 | MIPI CTRL 02 | 0x00 | RW | MIPI Control 02 |  |
|  |  |  |  | Bit[7]: | hs_prepare_sel |
|  |  |  |  |  | 0 : Auto calculate T_hs_prepare, unit: pclk2x |
|  |  |  |  |  | 1: Use hs_prepare_min_o[7:0] clk_prepare_sel |
|  |  |  |  | Bit[6]: | 0: Auto calculate T_clk_prepare, unit: pclk2x |
|  |  |  |  | Bit[5]: | 1: Use clk_prepare_min_o[7:0] |
|  |  |  |  |  | clk_post_sel |
|  |  |  |  |  | 0: Auto calculate T_clk_post, unit: pclk2x |
|  |  |  |  | Bit[4]: | 1: Use clk_post_min_o[7:0] |
|  |  |  |  |  | clk_trail_sel |
|  |  |  |  |  | 0 : Auto calculate T_clk_trail, unit: pclk2x <br> 1. Use clk trail min o[7:0] |
|  |  |  |  | Bit[3]: | hs_exit_sel |
|  |  |  |  |  | 0 : Auto calculate T_hs_exit, unit: pclk 2 x |
|  |  |  |  |  | 1: Use hs_exit_min_o[7:0] |
|  |  |  |  | Bit[2]: | hs_zero_sel |
|  |  |  |  |  | 0 : Auto calculate T_hs_zero, unit: pclk2x <br> 1: Use hs zero_min_o[7:0] |
|  |  |  |  | Bit[1]: | hs_trail_sel - |
|  |  |  |  |  | 0 : Auto calculate T_hs_trail, unit: pclk 2 x |
|  |  |  |  |  | 1: Use hs_trail.min_o[7:0] |
|  |  |  |  | Bit[0]: | clk_zero_sel |
|  |  |  |  |  | 0: Auto calculate T_clk_zero, unit: pclk2x |
|  |  |  |  |  | 1: Use clk_zero_min_o[7:0] |


|  |  | $0 \times 50$ | RW | MIPI Control 03 |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Bit[7:6]: |  | Ip_glitch_nu |
|  |  |  |  | 0: Use 2d of lp_in |
|  |  | Bit[5:4]: |  | 1: Mask one SCLK cycle glitch of Ip_in cd_glitch_nu |
|  |  |  |  | 0: Use 2d of lp_cd_in |
|  |  |  |  | 1: Mask one SCLK cycle glitch of lp cd in |
|  |  | Bit[3]: |  | cd1_int_en |
| 0x4803 | MIPI CTRL 03 |  |  |  | 0: Disable cd plus of data lane1 |
|  |  |  |  |  | 1: Enable cd plus of data lane1 |
|  |  |  |  | Bit[2]: | cd2_int_en |
|  |  |  |  |  | 0: Disable cd plus of data lane2 |
|  |  |  |  |  | 1: Enable cd plus of data lane2 |
|  |  |  |  | Bit[1]: | lp_cd1_en |
|  |  |  |  |  | 0: Disable cd of data_lane1 from PHY |
|  |  |  |  |  | 1: Enable cd of data_lane1from PHY |
|  |  |  |  | Bit[0]: | lp_cd2_en |
|  |  |  |  |  | 0: Disable cd of data_lane2 from PHY |
|  |  |  |  |  | 1: Enable cd of data_lane2 from PHY |

table 7-16 MIPI top control registers (sheet 4 of 9)

table 7-16 MIPI top control registers (sheet 5 of 9)

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4805 | MIPI CTRL 05 | $0 \times 10$ | RW |  |
| 0x4806 | MIPI CTRL 06 | 0x0F | RW | Bit[7]: mipi_test <br> Bit[6]: prbs_en <br> Bit[5:4]: Test mode <br> Debug control  <br> Bit[3]: mipi_slp_man_st <br>  <br>  <br>  <br> MIPI bus status manual control enable in <br> sleep mode <br> Bit[2]: <br> Bit[1]:lane_state <br> clk_la_lane2_state <br> Bit[0]: <br> data_lane1_state  |
| 0x4810 | MIPI MAX FRAME COUNT | 0xFF | RW | High Byte of Maximum Frame Count of Frame Sync Short Packet |
| $0 \times 4811$ | MIPI MAX FRAME COUNT | 0xFF | RW | Low Byte of Maximum Frame Count of Frame Sync Short Packet |
| 0x4812 | MIPI SHORT PKT COUNTER | $0 \times 00$ | RW | High Byte of Manual Short Packet Word Counter |
| 0x4813 | MIPI SHORT PKT COUNTER | $0 \times 00$ | RW | Low Byte of Manual Short Packet Word Counter |

table 7-16 MIPI top control registers (sheet 6 of 9 )

|  |  | default |  |  |
| :---: | :---: | :---: | :---: | :---: |
| address | register name | value | R/W | description |
| 0x4814 | MIPI CTRL14 | 0x2A | RW | MIPI Control 14 <br> Bit[7:6]: Virtual channel of MIPI <br> Bit[5:0]: Data type in manual mode |
| 0x4815 | MIPI_DT_SPKT | 0x40 | RW | Bit[7]: Not used <br> Bit[6]: pclk_inv <br> 0: Use mipi_pclk_o rising edge <br> 1: Use mipi_pclk_o falling edge <br> Bit[5:0]: Not used |
| $\begin{aligned} & \text { 0x4816~ } \\ & 0 \times 4817 \end{aligned}$ | RSVD | - | - | Reserved |
| 0x4818 | HS_ZERO_MIN | 0x00 | RW | High Byte of Minimum Value for Hs_zero, unit: ns |
| 0x4819 | HS_ZERO_MIN | 0x9A | RW | Low Byte of Minimum Value for Hs_zero, unit: ns hs_zero_real = hs_zero_min_o + Tui*ui_hs_zero_min_o |
| 0x481A | HS_TRAIL_MIN | 0x00 | RW | High Byte of Minimum Value for Hs_trail, unit: ns |
| 0x481B | HS_TRAIL_MIN | 0x3C | RW | Low Byte of Minimum Value for Hs_trail, hs_trail_real = hs_trail_min_o + Tui*ui_hs_trail_min_o |
| 0x481C | CLK_ZERO_MIN | 0x01 | RW | High Byte of Minimum Value for Clk_zero, unit: ns |
| 0x481D | CLK_ZERO_MIN | 0x86 | RW | Low Byte of Minimum Value for clk_zero, clk_zero_real = clk_zero_min_o + Tui*ui_clk_zero_min_o |
| 0x481E | CLK_PREPARE_ MIN | 0x00 | RW | High Byte of Minimum Value for clk_prepare, unit: ns |
| 0x481F | CLK_PREPARE MIN | 0x3C | RW | Low Byte of Minimum Value for clk_prepare clk_prepare_real = clk_prepare_min_o + Tui*ui_clk_prepare_min_o |
| 0x4820 | CLK_POST_MIN | 0x00 | RW | High Byte of Minimum Value for clk_post, unit: ns Bit[7:2]: Not used <br> Bit[1:0]: clk_post_min[9:8] |
| 0x4821 | CLK_POST_MIN | 0x56 | RW | Low Byte of Minimum Value for clk_post clk_post_real = clk_post_min_o + Tui*ui_clk_post_min_o |
| 0x4822 | CLK_TRAIL_MIN | 0x00 | RW | High Byte of Minimum Value for clk_trail, unit: ns Bit[7:2]: Not used <br> Bit[1:0]: clk_trail_min[9:8] |
| 0x4823 | CLK_TRAIL_MIN | 0x3C | RW | Low Byte of Minimum Value for clk_trail clk_trail_real = clk_trail_min_o + Tui*ui_clk_trail_min_o |
| 0x4824 | LPX_P_MIN | 0x00 | RW | High Byte of Minimum Value for Ipx_p, unit: ns <br> Bit[7:2]: Not used <br> Bit[1:0]: Ipx_p_min[9:8] |

table 7-16 MIPI top control registers (sheet 7 of 9)

| address | register name | default | value | R/W | description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0x4825 | LPX_P_MIN | $0 \times 32$ | RW | Low Byte of Minimum Value for lpx_p <br> lpx_p_real $=$ Ipx_p_min_o + Tui*ui_lpx_p_min_o |  |
| $0 \times 4826$ | HS_PREPARE_ <br> MIN | $0 \times 00$ | RW | High Byte of Minimum Value of hs_prepare, unit: ns |  |

table 7-16 MIPI top control registers (sheet 8 of 9)

|  |  | default |  |  |
| :---: | :---: | :---: | :---: | :---: |
| address | register name | value | R/W | description |
| 0x4838 | WKUP_DLY | 0x02 | RW | Wakeup Delay for MIPI (MARK1 state)/2^12 |
| 0x4839 | RSVD | - | - | Reserved |
| 0x483A | DIR_DLY | 0x08 | RW | Change LP Direction Delay/2 After LP11 |
| 0x483B | MIPI_LP_GPIO | 0x33 | RW | Bit[7]: Ip_sel1 <br>  $0: \quad$ Auto generate mipi_Ip_dir1_o <br>  1: Use Ip_dir_man1 to be mipi_lp_dir1_o <br> Bit[6]: Ip_dir_man1 <br>  $0:$ Input <br>  1: Output <br> Bit[5]: Ip_p1_o <br> Bit[4]: Ip_n1_o <br> Bit[3]: Ip_sel2 <br>  $0:$ Auto generate mipi_Ip_dir2_o <br>  1: Use Ip_dir_man2 to be mipi_lp_dir2_o <br> Bit[2]: Ip_dir_man2 <br>  $0:$ Input <br>  $1:$ Output <br> Bit[1]: Ip_p2_o <br> Bit[0]: Ip_n2_o |
| 0x483C | MIPI_CTRL3C | 0x4F | RW | ```Bit[7:4]: t_lpx Unit: SCLK cycle Bit[3:0]: t_clk_pre Unit: pclk2x cycle``` |
| 0x483D | T_TA_GO | 0x10 | RW | Unit: sclk cycle |
| 0x483E | T_TA_SURE | 0x06 | RW | Unit: sclk cycle |
| 0x483F | T_TA_GET | 0x14 | RW | Unit: sclk cycle |
| $\begin{aligned} & 0 \times 4840 ~ \\ & 0 \times 4845 \end{aligned}$ | RSVD | - | - | Reserved |
| 0x4846 | MIPI_CLIP_MAX | 0x0F | RW | Bit[7:4]: Not used <br> Bit[3:0]: MIPI output data max value[11:8] |
| $0 \times 4847$ | MIPI_CLIP_MAX | 0xFF | RW | Bit[7:0]: MIPI output data max value[7:0] |
| 0x4848 | MIPI_CLIP_MIN | 0x0F | RW | Bit[7:4]: Not used <br> Bit[3:0]: MIPI output data min value[11:8] |
| 0x4849 | MIPI_CLIP_MIN | 0xFF | RW | Bit[7:0]: MIPI output data min value[7:0] |
| 0x4850 | REG_INTR_MAN | - | W | Generate 1 SCLK Cycle Pulse for MCU Interrupt |
| 0x4851 | REG_TX_WR | - | W | Generate 1 SCLK Cycle Pulse to MIPI_TX_LP_TX, and reg_wdata Will Be Sent Out Through MIPI Escape Mode |
| 0x4852 | REG_TX_STOP | - | W | Generate 1 SCLK Cycle Pulse to MIPI_TX_LP_TX, and MIPI_TX_LP_TX Will Go Back to LP11 |

table 7-16 MIPI top control registers (sheet 9 of 9 )

| address | register name | default <br> value | R/W | description |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $0 \times 4853$ | REG_TA_ACK | - | W | Generate 1 SCLK Cycle Pulse to MIPI_TX_LP_TX to <br> Receive TurnAround Command |
| $0 \times 4854$ | REG_TA_REQ | - | W | Generate 1 SCLK Cycle Pulse to MIPI_TX_LP_TX to <br> Send TurnAround Command |
| $0 \times 4860$ | DEBUG CTRL | - | - | Debug Control |

### 7.17 LVDS control [0x4A00, 0x4A02-0x4AOF]

table 7-17 LVDS registers

| address | register name | default value | RW | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x4A00 | LVDS_R0 | 0x2A | RW | Bit[6]: SYNC code manual mode enable <br> Bit[5]: SYNC code enable when only 1 lane <br> Bit[4]: PCLK invert enable <br> Bit[3]: Channel ID enable in sync per lane mode <br> Bit[2]: F parameter in CCIR656 standard <br> Bit[1]: SAV first enable <br> Bit[0]: SYNC code mode <br> 0: Split <br> 1: Per lane |
| 0x4A02 | LVDS_R2 | $0 \times 00$ | RW | Bit[7:0]: Dummy data0[15:8] |
| 0x4A03 | LVDS_R3 | 0x80 | RW | Bit[7:0]: Dummy data0[7:0] |
| 0x4A04 | LVDS_R4 | 0x00 | RW | Bit[7:0]: Dummy data1[15:8] |
| 0x4A05 | LVDS_R5 | $0 \times 10$ | RW | Bit[7:0]: Dummy data1[7:0] |
| 0x4A06 | LVDS_R6 | 0xAA | RW | Blanking line_start Sync Code in Manual Sync Code Mode |
| 0x4A07 | LVDS_R7 | 0x55 | RW | Blanking line_end Sync Code in Manual Sync Code Mode |
| 0x4A08 | LVDS_R8 | 0x99 | RW | Video line_start Sync Code in Manual Sync Code Mode |
| 0x4A09 | LVDS_R9 | 0x66 | RW | Video line_end Sync Code in Manual Sync Code Mode |
| $\begin{aligned} & 0 \times 4 A 0 A \sim \\ & 0 \times 4 A 0 F \end{aligned}$ | DEBUG CTRL | - | - | Debug Control |

### 7.18 ISP top [0x5000-0x5017, 0x5E00-0x5E08]

table 7-18 ISP top registers (sheet 1 of 2 )

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x5000 | ISP CTRL 00 | 0x85 | RW |  |
| $0 \times 5001$ | ISP CTRL 01 | $0 \times 00$ | RW |  |
| 0x5002 | ISP CTRL 02 | $0 \times 00$ | RW | $\begin{array}{ll}\text { Bit[7:1]: } & \text { Debug control } \\ \text { Bit[0]: } & \text { manual_x_addr_st[8] }\end{array}$ |
| 0x5003 | ISP CTRL 03 | $0 \times 00$ | RW | Bit[7:0]: manual_x_addr_st[7:0] |
| 0x5004 | ISP CTRL 04 | $0 \times 00$ | RW | Bit[7:1]: Debug control <br> Bit[0]: manual_y_addr_st[8] |

table 7-18 ISP top registers (sheet 2 of 2)

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| $0 \times 5005$ | DEBUG CTRL | - | - | Debug Control |
| 0x5006 | ISP CTRL 06 | 0x00 | RW | Bit[7:1]: Debug control <br> Bit[0]: manual_x_addr_end[8] |
| 0x5007 | ISP CTRL 07 | 0x00 | RW | Bit[7:0]: manual_x_addr_end[7:0] |
| 0x5008 | ISP CTRL 08 | 0x00 | RW | Bit[7:1]: Debug control <br> Bit[0]: manual_y_addr_end[8] |
| 0x5009 | ISP CTRL 09 | 0x00 | RW | Bit[7:0]: manual_y_addr_end [7:0] |
| $\begin{aligned} & 0 \times 500 \mathrm{~A} \\ & 0 \times 500 \mathrm{~F} \end{aligned}$ | DEBUG CTRL | - | - | Debug Control |
| 0x5010 | SENSOR BIAS | - | R | Sensor Bias Debug |
| $0 \times 5011$ | LINEAR GAIN | - | R | AEC Linear Gain Debug |
| 0x5012 | AWB_GAIN_R | - | R | Bit[3:0]: AWB gain R channel[11:8] |
| 0x5013 | AWB_GAIN_R | - | R | Bit[7:0]: AWB gain R channel[7:0] |
| 0x5014 | AWB_GAIN_G | - | R | Bit[3:0]: AWB gain G channel[11:8] |
| 0x5015 | AWB_GAIN_G | - | R | Bit[7:0]: AWB gain G channel[7:0] |
| 0x5016 | AWB_GAIN_B | - | R | Bit[3:0]: AWB gain B channel[11:8] |
| $0 \times 5017$ | AWB_GAIN_B | - | R | Bit[7:0]: AWB gain B channel[7:0] |
| 0x5E00 | PRE_ISP 00 | 0x0C | R/W | Bit[7]: Color bar enable <br> 0: Disable <br> 1: Enable <br> Bit[6]: Debug control <br> Bit[5]: Mirror option for x offset <br> Bit[4]: Flip option for y offset <br> Bit[3]: Mirror order, bg or gb <br> Bit[2]: Flip order, br or rb <br> Bit[1:0]: Debug control |
| 0x5E01 | PRE_ISP 01 | - | R | Window X Offset High Byte |
| 0x5E02 | PRE_ISP 02 | - | R | Window X Offset Low byte |
| 0x5E03 | PRE_ISP 03 | - | R | Window Y Offset High Byte |
| 0x5E04 | PRE_ISP 04 | - | R | Window X Offset Low byte |
| 0x5E05 | PRE_ISP 05 | - | R | Window X Output Size High Byte |
| 0x5E06 | PRE_ISP 06 | - | R | Window X Output Size Low Byte |
| 0x5E07 | PRE_ISP 07 | - | R | Window Y Output Size High Byte |
| 0x5E08 | PRE_ISP 08 | - | R | Window Y Output Size Low Byte |

### 7.19 window control [ $0 \times 5 \mathrm{~A} 00$ - $0 \times 5 \mathrm{AOC}]$

table 7-19 window control registers

| address | register name | default value | R/W | description |
| :---: | :---: | :---: | :---: | :---: |
| 0x5A00 | MAN_XSTART_OFF | 0x00 | RW | Bit[7:5]: Not used <br> Bit[4:0]: X start offset[12:8] |
| 0x5A01 | MAN_XSTART_OFF | 0x00 | RW | Bit[7:0]: X start offset[7:0] |
| 0x5A02 | MAN_YSTART_OFF | 0x00 | RW | Bit[7:4]: Not used <br> Bit[3:0]: Y start offset[11:8] |
| 0x5A03 | MAN_YSTART_OFF | 0x00 | RW | Bit[7:0]: Y start offset[7:0] |
| 0x5A04 | MAN_WIN_WIDTH | 0x10 | RW | Bit[7:5]: Not used <br> Bit[4:0]: Window width[12:8] |
| 0x5A05 | MAN_WIN_WIDTH | 0xA0 | RW | Bit[7:0]: Window width[7:0] |
| 0x5A06 | MAN_WIN_HEIGHT | 0x0C | RW | Bit[7:4]: Not used <br> Bit[3:0]: Window height[11:8] |
| 0x5A07 | MAN_WIN_HEIGHT | 0x78 | RW | Bit[7:0]: Window height[7:0] |
| 0x5A08 | WIN_MAN | 0x00 | RW | Bit[7:1]: Not used <br> Bit[0]: Window size manual <br> 0 : Disable <br> 1: Enable |
| 0x5A09 | WIN_PX_CNT | - | R | Bit[7:5]: Not used <br> Bit[4:0]: Pixel counter[12:8] |
| 0x5A0A | WIN_PX_CNT | - | R | Bit[7:0]: Pixel counter[7:0] |
| 0x5A0B | WIN_LN_CNT | - | R | Bit[7:4]: Not used <br> Bit[3:0]: Line counter[11:8] |
| 0x5A0C | WIN_LN_CNT | - | R | Bit[7:0]: Line counter[7:0] |

## 8 operating specifications

## 8.1 absolute maximum ratings

table 8-1 absolute maximum ratings

| parameter |  | absolute maximum rating ${ }^{\mathrm{a}}$ |
| :--- | :--- | :--- |
| ambient storage temperature | $-40^{\circ} \mathrm{C}$ to $+125^{\circ} \mathrm{C}$ |  |
|  | $\mathrm{V}_{\mathrm{DD}-\mathrm{A}}$ | 4.5 V |
| supply voltage (with respect to ground) | $\mathrm{V}_{\mathrm{DD}-\mathrm{D}}$ | 3 V |
|  | $\mathrm{~V}_{\mathrm{DD}-\mathrm{IO}}$ | 4.5 V |
| electro-static discharge (ESD) | human body model | 2000 V |
| all input/output voltages (with respect to ground) |  | 200 V |
| I/O current on any input or output pin |  | -0.3 V to $\mathrm{V}_{\mathrm{DD}-\mathrm{o}}+1 \mathrm{~V}$ |
| peak solder temperature (10 second dwell time) |  | $\pm 200 \mathrm{~mA}$ |

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

## 8.2 functional temperature

table 8-2 functional temperature

| parameter | range |
| :--- | :--- |
| operating temperature $^{\mathrm{a}}$ | $-30^{\circ} \mathrm{C}$ to $+70^{\circ} \mathrm{C}$ junction temperature |
| ${\text { stable image temperature }{ }^{\mathrm{b}}}$ | $0^{\circ} \mathrm{C}$ to $+50^{\circ} \mathrm{C}$ junction temperature |

a. sensor functions but image quality may be noticeably different at temperatures outside of stable image range
b. image quality remains stable throughout this temperature range

### 8.3 DC characteristics

table 8-3 $\quad D C$ characteristics ( $\left.T_{A}=23^{\circ} \mathrm{C} \pm 2^{\circ} \mathrm{C}\right)$

| symbol | parameter | min | typ | max | unit |
| :---: | :---: | :---: | :---: | :---: | :---: |
| supply |  |  |  |  |  |
| $\mathrm{V}_{\mathrm{DD}-\mathrm{A}}$ | supply voltage (analog) | 2.6 | 2.8 | 3.0 | V |
| $V_{\text {DD-IO }}$ | supply voltage (digital I/O) | 1.7 | 1.8 | 3.0 | V |
|  | active (operating) current ${ }^{\text {a }}$ | 12 | 16 | 20 | mA |
| $\mathrm{I}_{\mathrm{DD}-10}$ |  | 28 | 40 | 55 | mA |
| $\mathrm{I}_{\text {DDS-SCCB }}{ }^{\text {b }}$ | standby current | 0.50 | 1.2 | 1.5 | mA |
| $\mathrm{I}_{\text {DDS-SCCB }}$ |  | 20 | 50 | 100 | $\mu \mathrm{A}$ |
| IDDS-XSHUTDOwn |  | 2 | 10 | 30 | $\mu \mathrm{A}$ |
| digital inputs (typical conditions: $\mathrm{AVDD}=2.8 \mathrm{~V}, \mathrm{DVDD}=1.5 \mathrm{~V}, \mathrm{DOVDD}=1.8 \mathrm{~V}$ ) |  |  |  |  |  |
| $\mathrm{V}_{\text {IL }}$ | input voltage LOW |  |  | 0.54 | V |
| $\mathrm{V}_{\mathrm{IH}}$ | input voltage HIGH | 1.26 |  |  | V |
| $\mathrm{C}_{\text {IN }}$ | input capacitor |  |  | 10 | pF |
| digital outputs (standard loading 25 pF ) |  |  |  |  |  |
| $\mathrm{V}_{\mathrm{OH}}$ | output voltage HIGH | 1.62 |  |  | V |
| $\mathrm{V}_{\text {OL }}$ | output voltage LOW |  |  | 0.18 | V |
| serial interface inputs |  |  |  |  |  |
| $\mathrm{V}_{\text {IL }}{ }^{\text {c }}$ | SIOC and SIOD | -0.5 | 0 | 0.54 | V |
| $\mathrm{V}_{\mathrm{IH}}$ | SIOC and SIOD | 1.28 | 1.8 | 3.0 | V |

a. $640 \times 480 @ 100 \mathrm{fps}$
b. with XEXTCLK
c. based on DOVDD $=1.8 \mathrm{~V}$

## 8.4 timing characteristics

table 8-4 timing characteristics

| symbol | parameter | min | typ | $\max$ | unit |
| :--- | :--- | :--- | :--- | :--- | :--- |
| oscillator and clock input |  |  |  |  |  |
| $\mathrm{f}_{\text {osc }}$ | frequency (EXTCLK) | 6 | 24 | 27 | MHz |
| $\mathrm{t}_{\mathrm{r}} \mathrm{t}_{\mathrm{f}}$ | clock input rise/fall time |  |  | $5\left(10^{\mathrm{a}}\right)$ | ns |

a. if using internal PLL

## 9 mechanical specifications

## 9.1 physical specifications

figure 9-1 package specifications

table 9-1 package dimensions

| parameter | symbol | min | typ | max | unit |
| :---: | :---: | :---: | :---: | :---: | :---: |
| package body dimension $x$ | A | 3885 | 3910 | 3935 | $\mu \mathrm{m}$ |
| package body dimension y | B | 3385 | 3410 | 3435 | $\mu \mathrm{m}$ |
| package height | C | 700 | 760 | 820 | $\mu \mathrm{m}$ |
| ball height | C1 | 100 | 130 | 160 | $\mu \mathrm{m}$ |
| package body thickness | C2 | 585 | 630 | 675 | $\mu \mathrm{m}$ |
| thickness of glass surface to wafer | C3 | 425 | 445 | 465 | $\mu \mathrm{m}$ |
| image plane height | C4 | 260 | 315 | 370 | $\mu \mathrm{m}$ |
| ball diameter | D | 220 | 250 | 280 | $\mu \mathrm{m}$ |
| total pin count | N |  | 35 (9 NC) |  |  |
| pin count $x$-axis | N1 |  | 7 |  |  |
| pin count y-axis | N2 |  | 6 |  |  |
| pins pitch x -axis | J1 |  | 500 |  | $\mu \mathrm{m}$ |
| pins pitch y-axis | J2 |  | 520 |  | $\mu \mathrm{m}$ |
| edge-to-pin center distance along $x$ | S1 | 425 | 455 | 485 | $\mu \mathrm{m}$ |
| edge-to-pin center distance along y | S2 | 375 | 405 | 435 | $\mu \mathrm{m}$ |
| air gap between die and glass |  | 40 | 45 | 50 | $\mu \mathrm{m}$ |

### 9.2 IR reflow specifications

figure 9-2 IR reflow ramp rate requirements

table 9-2 reflow conditions ${ }^{\text {ab }}$

| zone | description | exposure |
| :--- | :--- | :--- |
| ramp up $\mathrm{A}\left(\mathrm{T}_{0}\right.$ to $\left.\mathrm{T}_{\text {min }}\right)$ | heating from room temperature to $150^{\circ} \mathrm{C}$ | temperature slope $\leq 3^{\circ} \mathrm{C}$ per second |
| soaking | heating from $150^{\circ} \mathrm{C}$ to $200^{\circ} \mathrm{C}$ | $90 \sim 150$ seconds |
| ramp up $\mathrm{B}\left(t_{\mathrm{L}}\right.$ to $\left.\mathrm{T}_{\mathrm{p}}\right)$ | heating from $217^{\circ} \mathrm{C}$ to $245^{\circ} \mathrm{C}$ | temperature slope $\leq 3^{\circ} \mathrm{C}$ per second |
| peak temperature | maximum temperature in SMT | $245^{\circ} \mathrm{C}+0 /-5^{\circ} \mathrm{C}($ duration max 30 sec$)$ |
| reflow (t to $\left.\mathrm{T}_{\mathrm{L}}\right)$ | temperature higher than $217^{\circ} \mathrm{C}$ | $30 \sim 120$ seconds |
| ramp down $\mathrm{A}\left(\mathrm{T}_{\mathrm{p}}\right.$ to $\left.\mathrm{T}_{\mathrm{L}}\right)$ | cooling from $245^{\circ} \mathrm{C}$ to $217^{\circ} \mathrm{C}$ | temperature slope $\leq 3^{\circ} \mathrm{C}$ per second |
| ramp down $\mathrm{B}\left(\mathrm{T}_{\mathrm{L}}\right.$ to $\left.\mathrm{T}_{f}\right)$ | cooling from $217^{\circ} \mathrm{C}$ to room temperature | temperature slope $\leq 2^{\circ} \mathrm{C}$ per second |
| $\mathrm{T}_{0}$ to $\mathrm{T}_{\mathrm{p}}$ | room temperature to peak temperature | $\leq 8$ minutes |

a. maximum number of reflow cycles $=3$
b. N 2 gas reflow or control O 2 gas $\mathrm{PPM}<500$ as recommended

## 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 $\mathbf{2}$ as most optical assemblies invert and mirror the image, the chip is typically mounted with pins A1 to A7 oriented down on the PCB.

## 10.2 lens chief ray angle (CRA)

figure 10-2 chief ray angle (CRA)

table 10-1 CRA versus image height plot

| field (\%) | image height (mm) | CRA (degrees) |
| :--- | :--- | :--- |
| 0 | 0 | 0 |
| 0.085 | 0.102 | 2.3 |
| 0.17 | 0.204 | 4.7 |
| 0.255 | 0.306 | 7.2 |
| 0.34 | 0.408 | 9.7 |
| 0.425 | 0.51 | 12.3 |
| 0.51 | 0.612 | 14.9 |
| 0.595 | 0.714 | 17.5 |
| 0.68 | 0.816 | 20.1 |
| 0.765 | 0.918 | 22.7 |
| 0.85 | 1.02 | 25.2 |
| 0.91666667 | 1.1 | 27.1 |
| 1 | 1.2 | 29 |

### 10.3 IR cut off wavelength

Wavelength above 975 nm must be cut off for both monochrome and color sensors to avoid package structure ghosting. For color sensors, it is recommended to cut wavelengh at 650 nm or shorter for good color reproduction.

## 10.4 spectrum response

figure 10-3 OV7251 spectrum response curve


## revision history

```
version 1.0
```

07.25.2014

- initial release
08.12.2014
- changed datasheet from Preliminary Specification to Product Specification
- in key specifications, changed dark current to $350 \mathrm{e}^{-/ s} @ 50^{\circ} \mathrm{C}$ junction temperature and changed sidebar note to "Maximum integration time...max dark current is around $3.5 e^{-}$at $50^{\circ} \mathrm{C}$ )."
version 2.01
11.18.2014
- in table 7-5, changed description of register 0x350A to "Bit[7:2]: Not used; Bit[1:0]: Gain[9:8]"
- in table 7-6, added register $0 \times 3636$
version $2.02 \quad$ 01.19.2015
- in table 7-6, changed description of register 0x3636 from Bit[3:0]: Internal regulator disable" to "Bit[3]: Internal regulator disable"
version 2.1
04.15 .2015
- in table 5-3 and table 7-5, changed description of register bit 0x3500[3:0] to Exposure[15:12], register bits $0 \times 3501[7: 0$ ] to Exposure[11:4], register bits $0 \times 3502[7: 4]$ to Exposure[3:0], and register bits $0 \times 3502$ [3:0] to Debug control
- in table 5-3 and table 7-5, added "Minimum exposure time is 1 row period. Maximum exposure time is frame length -20 row periods, where frame length is set by registers $\{0 \times 380 \mathrm{E}, 0 \times 380 \mathrm{~F}\}$." to descriptions of registers $0 \times 3501$ and $0 \times 3502$
- in chapter 10, added section 10.3
version 2.11
07.14.2015
- in chapter 10, added section 10.4
version 2.12
10.12.2015
- in table 2-5, changed min value for t 3 and t 4 to 65536
- in section 4.8.3, changed last two sentences of second paragraph to "... rising to integration, $t_{\text {Exp_ }}$ Dly, equals to $16388 \times t_{\text {xVCLK }}+11$ trow. . The frame start short packet is sent out about 8 row periods after integration finishes."
- in table 7-6, added description of register 0x3662


## defining the future of digital imaging ${ }^{\text {rTM }}$

## OmniVision Technologies, Inc.

## UNITED STATES

4275 Burton Drive
Santa Clara, CA 95054
tel: +1 4085673000
fax: +1 4085673001
email: sales@ovt.com

## UNITED KINGDOM

Hook, Hampshire +44 1256744610

GERMANY
Munich +4989 63819988

INDIA
Bangalore +91 8041128966

CHINA
Beijing + 861065801690
Shanghai + 862161759888
+86 2157749288
Shenzhen + 8675583849733

KOREA
Seoul + 82234723769

SINGAPORE +65 69331933

TAIWAN
Taipei +886 226579800
Hsin-chu +886 35656688

