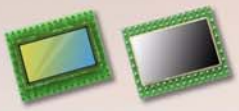


datasheet

PRELIMINARY SPECIFICATION

1/1.79" color CMOS 4 megapixel (2688 x 1520) image sensor
with PureCel®Plus-S and Nyxel® technologies

OS04A10 (rev 1B)

color CMOS 4 megapixel (2688x1520) image sensor with PureCel®Plus-S and Nyxel® technologies

datasheet (fan-out)
PRELIMINARY SPECIFICATION

version 1.0
april 2019

To learn more about OmniVision Technologies, visit www.ovt.com.

applications

- security camera
- action camera
- high resolution consumer camera

ordering information

- **OS04A10-J72A-1B-Z** (color, lead-free)
  72-pin fan-out package

note Since it is impossible to check compatibility with all displays, check interoperability before committing to mass production.

features

- QE enhancement in NIR range
- support for image size: 2688x1520, VGA, QVGA, and any cropped size
- high dynamic range
- high sensitivity
- image sensor processor functions: defective pixel cancelation, DCG combination, automatic black level correction, PWL compression, etc.
- pixel data: 12b RAW RGB
- SCCB for register programming
- programmable GPIOs
- high speed serial data transfer with MIPI CSI-2 or LVDS
- external frame synchronization capability
- embedded temperature sensor
- one-time programmable (OTP) memory

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

key specifications (typical)

- **active array size:** 2688 x 1520
- **power supply:**
  analog: 2.8V
  digital: 1.2V
  I/O pads: 1.8V
- **power requirements:**
  active: TBD
- **temperature range:**
  operating: -30°C to 85°C junction temperature
- **output interfaces:** up to 4-lane MIPI CSI-2 or LVDS
- **input clock frequency:** 6 ~ 36 MHz
- **lens size:** 1/1.79"
- **lens chief ray angle:** 9° (see **figure 10-3**)
- **SCCB speed**: up to 1 MHz
- **scan mode:** progressive
- **shutter:** rolling shutter
- **output formats:** single exposure HDR - 16-bit combined RAW, 12-bit (PWL) compressed combined RAW; dual exposure HDR - 16-bit combined RAW + 12-bit VS RAW, 12-bit (PWL) compressed combined RAW + 12-bit VS RAW; 3-exposure HDR - 12-bit long exposure+12-bit medium exposure+12-bit short exposure
- **maximum image transfer rate:**
  30x3 fps @ 1520p
- **sensitivity:** 32,000 e^-/Lux-sec (green pixel response at 530 nm illumination)
- **max S/N ratio:** TBD
- **dynamic range:**
  >120 dB dual exposure staggered HDR and 3-exposure staggered HDR
- **pixel size:** 2.9 µm x 2.9 µm
- **image array area:** 7841.6 µm x 4454.4 µm
- **package dimensions:** 10680 µm x 8540 µm

note OmniVision recommends fan-out packages use underfill as part of camera assembly process.

note Pixel performance and power requirements are estimates. Values may change based on real measurements.

table of contents

list of figures

list of tables

1 application system

1.1 overview

The OS04A10 color image sensor is a 1/1.79" optical format, 2688x1520 single-chip, CMOS, active-pixel, digital high dynamic range sensor, intended for the high-end security and consumer market.

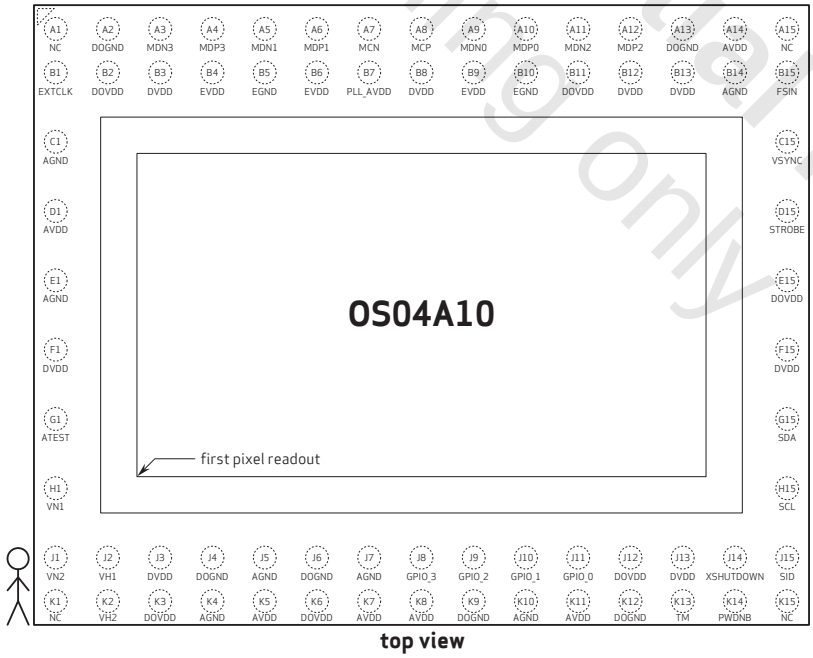
The OS04A10 features OmniVision's PureCel®Plus-S and Nyxel® technologies to extend the dynamic range. The OS04A10 has the option to output up to 12-bit, 30 fps video with 90 dB dynamic range from a single exposure (DCG) plus with very short exposure. The sensor supports 3-exposure, staggered HDR for >120 dB, but in this case, the HDR combination is done externally.

The OS04A10 performs sophisticated camera functions on-chip controlled via the serial camera control bus (SCCB) interface. These functions include black level correction (BLC), dual conversion gain (DCG) combination, defect pixel correction (DPC) and piece-wise linear (PWL) image compression. The OS04A10 enables advanced HDR imaging in a simple, cost effective system.

1.2 signal description and pin assignment

table 1-1 lists the signal descriptions and their corresponding pin numbers for the OS04A10 image sensor. The package information is shown in **section 9**. NC labeled pins are not connected to any other electrical net. For better thermal conduction, they can be tied to the ground plane.

figure 1-1 pin diagram



top view

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

pin number	signal name	pin type	description
A1	NC	–	no connect
A2	DOGND	ground	ground for I/O and digital circuit
A3	**MDN3**	output	MIPI data negative output 3
A4	**MDP3**	output	MIPI data positive output 3
A5	**MDN1**	output	MIPI data negative output 1
A6	**MDP1**	output	MIPI data positive output 1
A7	**MCN**	output	MIPI clock negative output
A8	**MCP**	output	MIPI clock positive output
A9	**MDN0**	output	MIPI data negative output 0
A10	**MDP0**	output	MIPI data positive output 0
A11	**MDN2**	output	MIPI data negative output 2
A12	**MDP2**	output	MIPI data positive output 2
A13	DOGND	ground	ground for I/O and digital circuit
A14	AVDD	power	analog power
A15	NC	–	no connect
B1	**EXTCLK**	input	clock input
B2	DOVDD	power	I/O power
B3	DVDD	power	digital circuit power
B4	EVDD	power	MIPI core logic power
B5	EGND	ground	MIPI supply ground
B6	EVDD	power	MIPI core logic power
B7	PLL_AVDD	power	PLL power supply
B8	DVDD	power	digital circuit power
B9	EVDD	power	MIPI core logic power
B10	EGND	ground	MIPI supply ground
B11	DOVDD	power	I/O power
B12	DVDD	power	digital circuit power
B13	DVDD	power	digital circuit power
B14	AGND	ground	analog ground
B15	**FSIN**	input	frame synchronization input

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

pin number	signal name	pin type	description
C1	AGND	ground	analog ground
C15	**VSYNC**	output	vertical/frame synchronization output signal
D1	AVDD	power	analog power
D15	**STROBE**	output	frame exposure output indicator
E1	AGND	ground	analog ground
E15	DOVDD	power	I/O power
F1	DVDD	power	digital circuit power
F15	DVDD	power	digital circuit power
G1	**ATEST**	analog I/O	analog test I/O
G15	**SDA**	I/O	SCCB interface data pin
H1	VN1	power	negative reference voltage 1
H15	**SCL**	input	SCCB interface input clock
J1	VN2	power	negative reference voltage 2
J2	VH1	power	high/positive reference voltage 1
J3	DVDD	power	digital circuit power
J4	DOGND	ground	ground for I/O and digital circuit
J5	AGND	ground	analog ground
J6	DOGND	ground	ground for I/O and digital circuit
J7	AGND	ground	analog ground
J8	**GPIO_3**	I/O	general purpose I/O 3
J9	**GPIO_2**	I/O	general purpose I/O 2
J10	**GPIO_1**	I/O	general purpose I/O 1
J11	**GPIO_0**	I/O	general purpose I/O 0
J12	DOVDD	power	I/O power
J13	DVDD	power	digital circuit power
J14	**XSHUTDOWN**	input	reset and power down (active low with pull-down resistor)
J15	**SID**	input	sensor SCCB ID switch input 0: SCCB ID address = 0x6C 1: SCCB ID address = 0x20
K1	NC	–	no connect
K2	VH2	power	high/positive reference voltage 2

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

pin number	signal name	pin type	description
K3	DOVDD	power	I/O power
K4	AGND	ground	analog ground
K5	AVDD	power	analog power
K6	DOVDD	power	I/O power
K7	AVDD	power	analog power
K8	AVDD	power	analog power
K9	DOGND	ground	ground for I/O and digital circuit
K10	AGND	ground	analog ground
K11	AVDD	power	analog power
K12	DOGND	ground	ground for I/O and digital circuit
K13	**TM**	input	test mode (active high, internally tied to pull-down resistor)
K14	**PWDNB**	input	power down (active low, internally tied to pull-up resistor)
K15	NC	–	no connect

table 1-2 pin states under various conditions (sheet 1 of 2)

pin	signal name	XSHUTDOWN mode[a]	power down mode[b]	SW_STANDBY mode[c]
A3	MDN3	high-z	high	high
A4	MDP3	high-z	high	high
A5	MDN1	high-z	high	high
A6	MDP1	high-z	high	high
A7	MCN	high-z	high	high
A8	MCP	high-z	high	high
A9	MDN0	high-z	high	high
A10	MDP0	high-z	high	high
A11	MDN2	high-z	high	high
A12	MDP2	high-z	high	high
B1	EXTCLK	input	input	input
B15	FSIN	high-z	low	input (configurable)

table 1-2 pin states under various conditions (sheet 2 of 2)

pin	signal name	XSHUTDOWN mode[a]	power down mode[b]	SW_STANDBY mode[c]
C15	VSYNC	high-z	low	input[d] (configurable)
D15	STROBE	low	low	low
G1	ATEST	high-z	high-z	high-z
G15	SDA	open drain	open drain	I/O
H15	SCL	high-z	high-z	input
J8	GPIO_3	high-z	high	high[d] (configurable)
J9	GPIO_2	high-z	high	high[d] (configurable)
J10	GPIO_1	high-z	high	high[d] (configurable)
J11	GPIO_0	high-z	low	low[d] (configurable)
J14	XSHUTDOWN	input	input	input
J15	SID	input	input	input
K13	TM	input	input	input
K14	PWDNB	input	input	input

a. XSHUTDOWN = 0

b. XSHUTDOWN = 1, PWDNB = 0 from software standby

c. sensor set to software standby from streaming mode

d. default state can change, depending on state of sensor and/or function of pin

1.3 I/O control

The OS04A10 can configure its I/O pins as an input or output. For the output signal, it follows one of two paths: either from the data path or from register control.

table 1-3 GPIO control registers (sheet 1 of 2)

pin	register bit	register function	description
VSYNC	0x3002[7]	VSYNC I/O control	0: Input 1: Output
	0x3008[7]	VSYNC output select	0: Output sensor VSYNC 1: Output controlled by register value
	0x3005[7]	VSYNC output value	Value to be output on pin when option selected
	0x303E[1]	GPIO input value	Read value when input option is selected for pin
FSIN	0x3002[3]	FSIN I/O control	0: Input (used for FSIN) 1: Output
	0x3008[3]	FSIN output select	1: Output controlled by register value
	0x3005[3]	FSIN output value	Value to be output on pin when option selected
	0x303E[2]	GPIO input value	Read value when input option is selected for pin
GPIO_0	0x3002[0]	GPIO I/O control	0: Input 1: Output
	0x3008[0]	GPIO output select	1: Output controlled by register value
	0x3005[0]	GPIO output value	Value to be output on pin when option selected
	0x303F[0]	GPIO input value	Read value when input option is selected for pin
GPIO_1	0x3002[1]	GPIO I/O control	0: Input 1: Output
	0x3008[1]	GPIO output select	1: Output controlled by register value
	0x3005[1]	GPIO output value	Value to be output on pin when option selected
	0x303F[1]	GPIO input value	Read value when input option is selected for pin
GPIO_2	0x3001[0]	GPIO I/O control	0: Input 1: Output
	0x3007[0]	GPIO output select	1: Output controlled by register value
	0x3004[0]	GPIO output value	Value to be output on pin when option selected
	0x303F[2]	GPIO input value	Read value when input option is selected for pin

table 1-3 GPIO control registers (sheet 2 of 2)

pin	register bit	register function	description
GPIO_3	0x3001[1]	GPIO I/O control	0: Input 1: Output
	0x3007[1]	GPIO output select	1: Output controlled by register value
	0x3004[1]	GPIO output value	Value to be output on pin when option selected
	0x303F[3]	GPIO input value	Read value when input option is selected for pin

table 1-4 pad equivalent circuit (sheet 1 of 2)

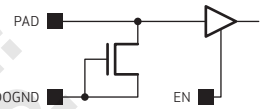
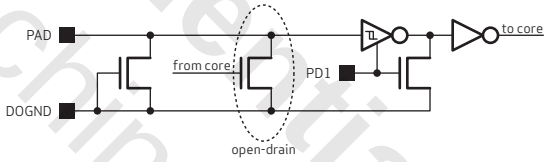
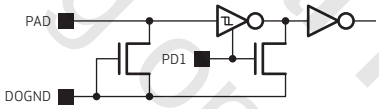
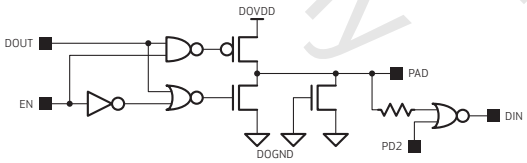
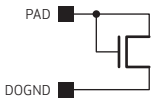
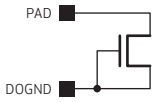
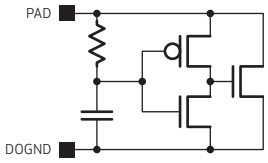
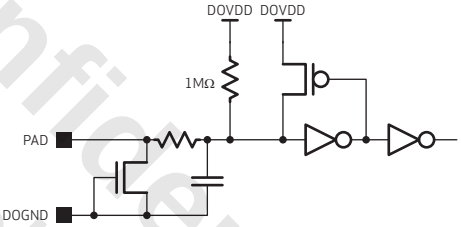
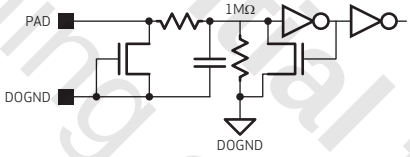
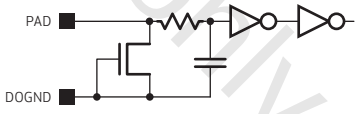
signal name	equivalent circuit
EXTCLK	
SDA	
SCL	
VSYNC, GPIO0, GPIO1, GPIO2, GPIO3, FSIN	
VN1, VN2	

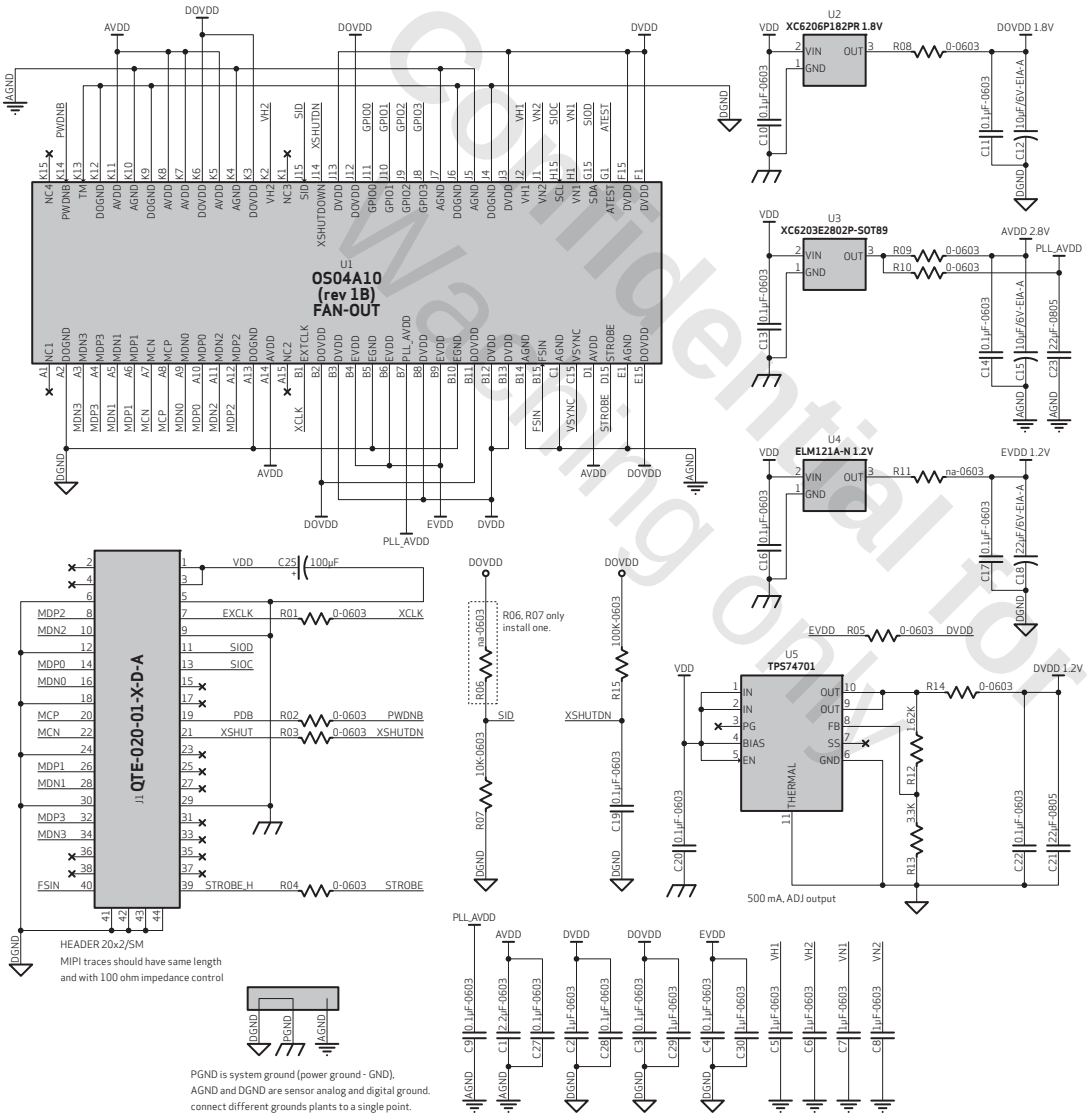
table 1-4 pad equivalent circuit (sheet 2 of 2)

signal name	equivalent circuit
MDP3, MDN3, MDP2, MDN2, MDP1, MDN1, MDP0, MDN0, MCP, MCN, EGND, AGND, DOGND, VH1, VH2, ATEST	
AVDD, DVDD, DOVDD, PLL_AVDD, EVDD	
PWDNB	
TM, XSHUTDOWN	
SID	

1.4 reference design

figure 1-2 shows the power supply and signal connection of the OS04A10 when using MIPI interface. The silicon (chip) revision of the sensor can be read from register 0x302A. The SCCB ID of the sensor is controlled by the SID pin at power up and after hardware reset (XSHUTDOWN pin low). If SID is low, the sensor's default SCCB ID is 0x6C (default) or value defined in register 0x3035. If SID is high, the sensor's default SCCB ID is 0x20 (default) or value defined in register 0x3037. Changing register 0x3035 or register 0x3037, using SCCB command or from OTP loading, will change the SCCB ID thereafter. SID should not be floating.

figure 1-2 OS04A10 MIPI reference schematic

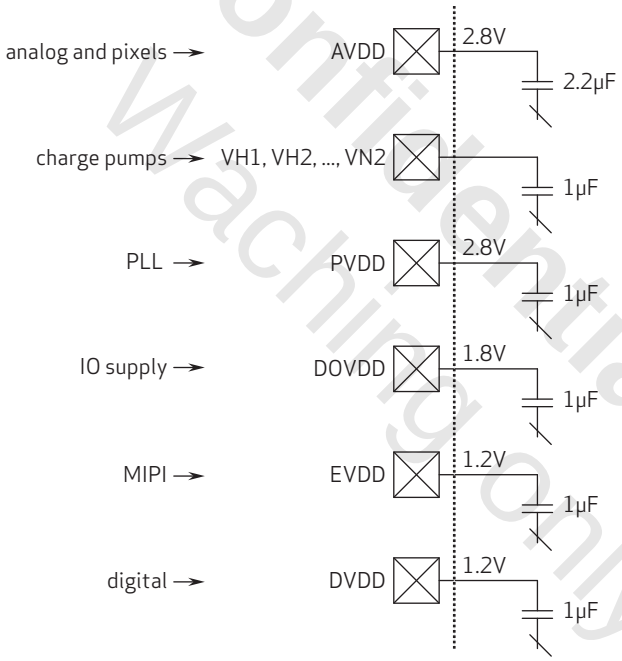


1.4.1 external components

The pixel array and analog readout circuits are powered from 2.8V (AVDD). The PLL core is also powered by 2.8V (PVDD). It is possible to connect both PVDD and AVDD to the same external power supply. The I/O pad power supply voltage level is 1.8V (DOVDD). The digital core logic (DVDD) and the MIPI core logic (EVDD) operate on 1.2V power supply.

The OS04A10 must use external power de-coupling capacitors to reduce noise on the supply lines and for proper operation of its charge pumps. These de-coupling capacitors should be connected as close as possible to the different power supply pins of the chip. For the core power supplies, a minimum capacitance of 1 µF is recommended. A minimum capacitance of 1 µF should be used for the output pins from the embedded charge pumps, which generate high/positive (VH1/VH2/VH3) and negative (VN1/VN2) reference voltages. At power up, power supplies should ramp up in 50 µs or more to avoid in-rush current during ramp-up.

figure 1-3 OS04A10 power supplies and recommended external decoupling



1.5 power management

Based on the system power configuration (XSHUTDOWN), the power up sequence will be different. OmniVision recommends cutting off all power supplies, including the external DVDD, when the sensor is not in use.

1.5.1 power up sequence

To avoid any glitch from a strong external noise source, OmniVision recommends controlling XSHUTDOWN by GPIO.

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

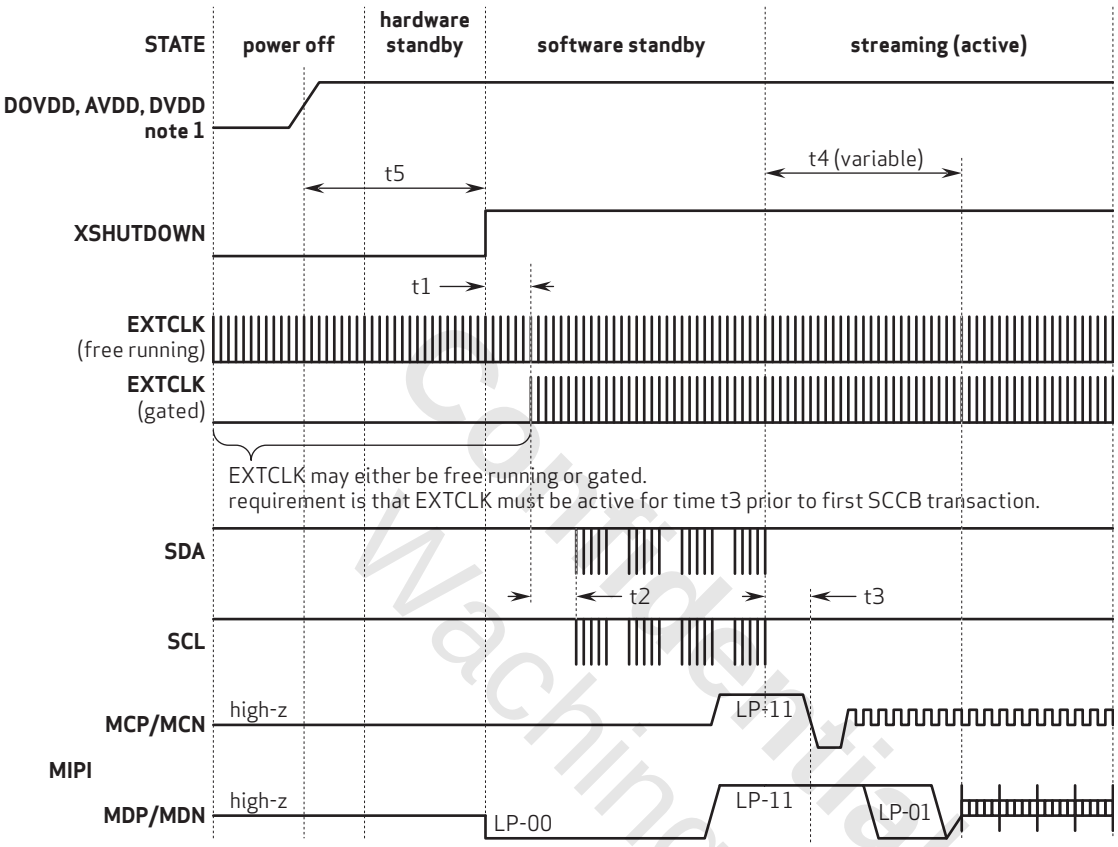
table 1-5 power up sequence

XSHUTDOWN	power up sequence requirement
GPIO	Refer to **figure 1-4** 1. DOVDD, AVDD, and DVDD can rise in any order 2. XSHUTDOWN rising must occur after AVDD, DOVDD, and DVDD are stable

table 1-6 power up sequence timing constraints

constraint	label	min	max	unit
XSHUTDOWN rising – system ready	t1	5		ms
minimum number of EXTCLK cycles prior to first SCCB transaction	t2	8192		EXTCLK cycles
MIPI clock startup time	t3		8192	EXTCLK cycles
entering streaming mode – first frame start sequence (variable part)	t4	delay related to output frame rate and line timing		lines
AVDD, DOVDD, or DVDD, whichever is last – XSHUTDOWN rising	t5	0	∞	ns

figure 1-4 power up sequence timing diagram



note 1 DOVDD, AVDD, and DVDD may rise in any order

1.5.2 power down sequence

Pull XSHUTDOWN low to set the sensor into power down mode. The digital and analog supply voltages can be cut off in any order (e.g., DOVDD, DVDD, then AVDD or AVDD, DVDD, then DOVDD). Similar to the power up sequence, the EXTCLK input clock may be either gated or continuous. To avoid bad frames from MIPI, OmniVision recommends setting the sensor into sleep mode in inter frame first before sending the sensor into power down mode by setting register 0x3021[6:5]=2'b00 and register 0x0100 = 0x00.

table 1-7 power down sequence

XSHUTDOWN	power down sequence requirement
GPIO	Refer to **figure 1-6** 1. Software standby recommended 2. Pull XSHUTDOWN low for minimum power consumption 3. Pull AVDD, DVDD, and DOVDD low in any order

table 1-8 power down sequence timing constraints

constraint	label	min	max	unit
enter software standby SCCB command device in software standby mode	t0	when a frame of MIPI data is output, wait for MIPI end code before entering software for standby; otherwise, enter software standby mode immediately		
minimum of EXTCLK cycles after last SCCB transaction or MIPI frame end	t1	512		EXTCLK cycles
last SCCB transaction or MIPI frame end, XSHUTDOWN falling	t2	512		EXTCLK cycles
XSHUTDOWN falling – AVDD, DVDD, or DOVDD falling whichever is first	t3	0.0		ns

figure 1-5 software standby sequence

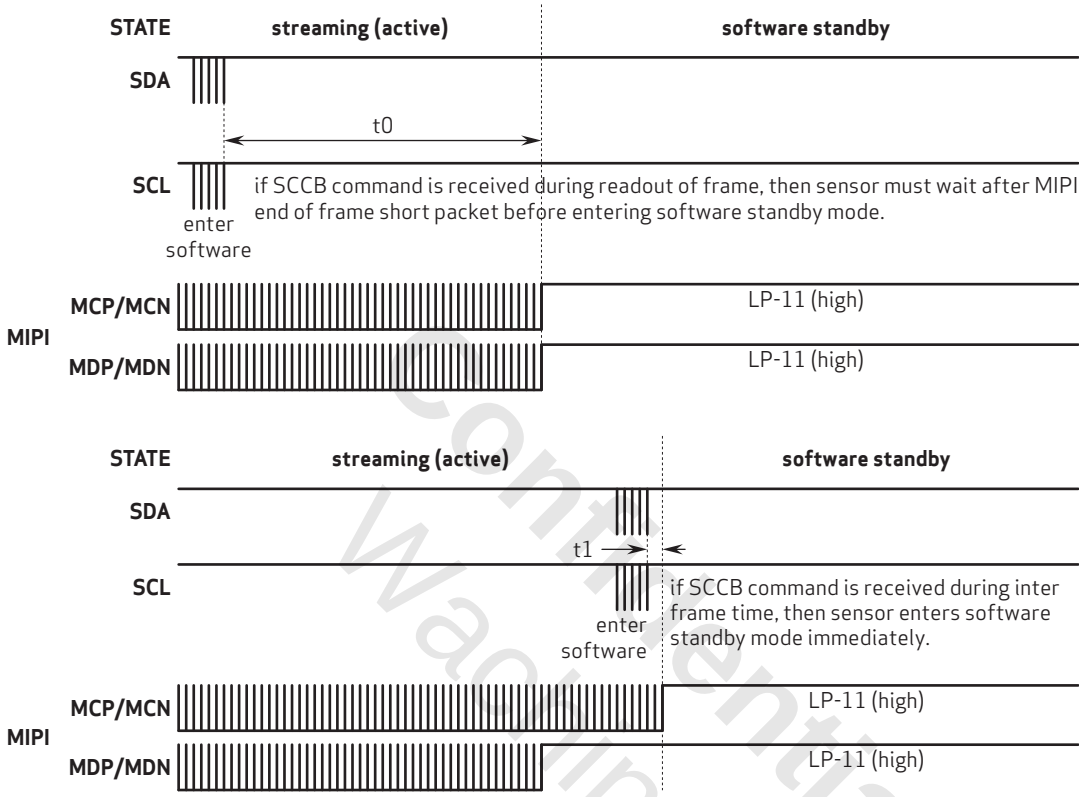
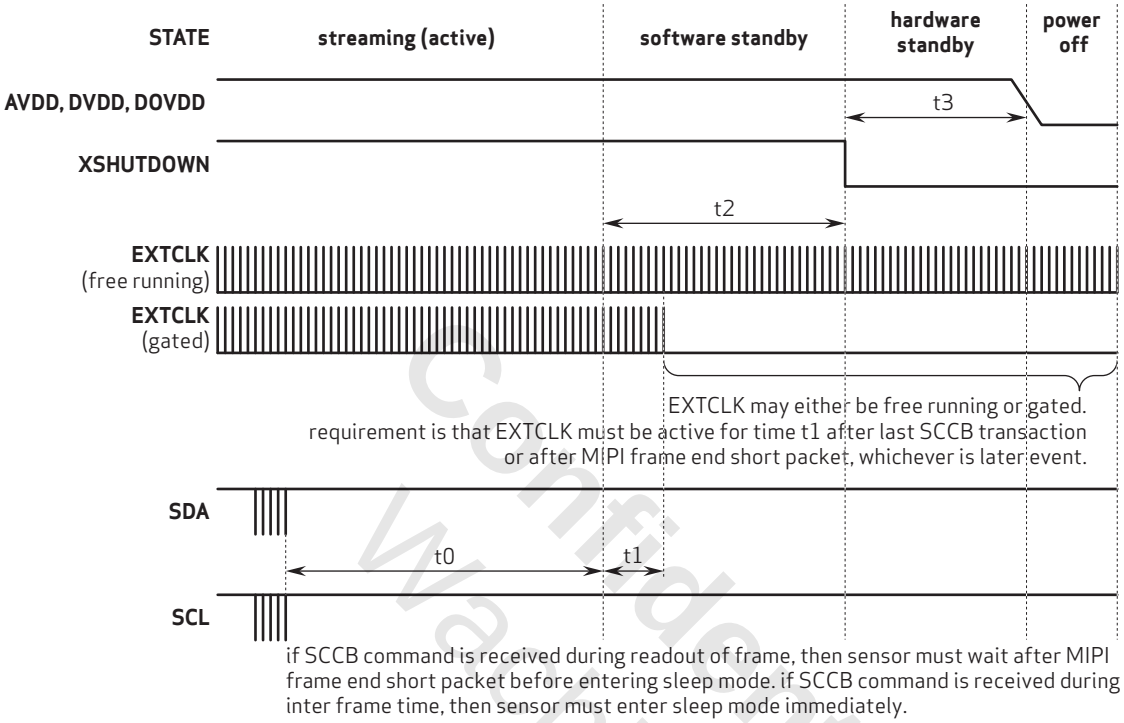


figure 1-6 power down sequence



1.6 hardware and software standby

Two suspend modes are available for the OS04A10:

- hardware standby
- software standby

1.6.1 hardware standby

Dropping any power source (AVDD/DOVDD/DVDD) or if XSHUTDOWN/PWDNB is tied to low, will initiate a hardware standby mode. Refer to **table 1-9** for difference between XSHUTDOWN and PWDNB hardware standby.

1.6.2 software standby

Executing a software power down (0x0100[0]) through the SCCB interface suspends internal circuit activity, but does not halt the device clock. All register content is maintained in standby mode. During the resume state, all the registers are restored to their original values.

table 1-9 hardware standby description

mode	description
hardware standby with XSHUTDOWN	1. Enabled by pulling XSHUTDOWN low 2. Power down all blocks 3. Register values are reset to default values 4. No SCCB communication 5. Minimum power consumption
hardware standby with PWDNB	1. Default mode after power on reset 2. Power down all blocks except SCCB 3. Register values are maintained 4. SCCB communication is available 5. Low power consumption 6. GPIO can be configured as high/low/tri-state

1.7 reset

The whole chip will be reset during power up. Manually applying a hardware reset (XSHUTDOWN=0) upon power up is recommended even though the on-chip power up reset is included. The hardware reset is active low with an asynchronized design. The reset pulse width should be greater than or equal to 2 ms.

1.7.1 power on reset

The power on reset can be controlled from an external pin. Additionally, in this sensor, a power on reset is generated after the core power becomes stable.

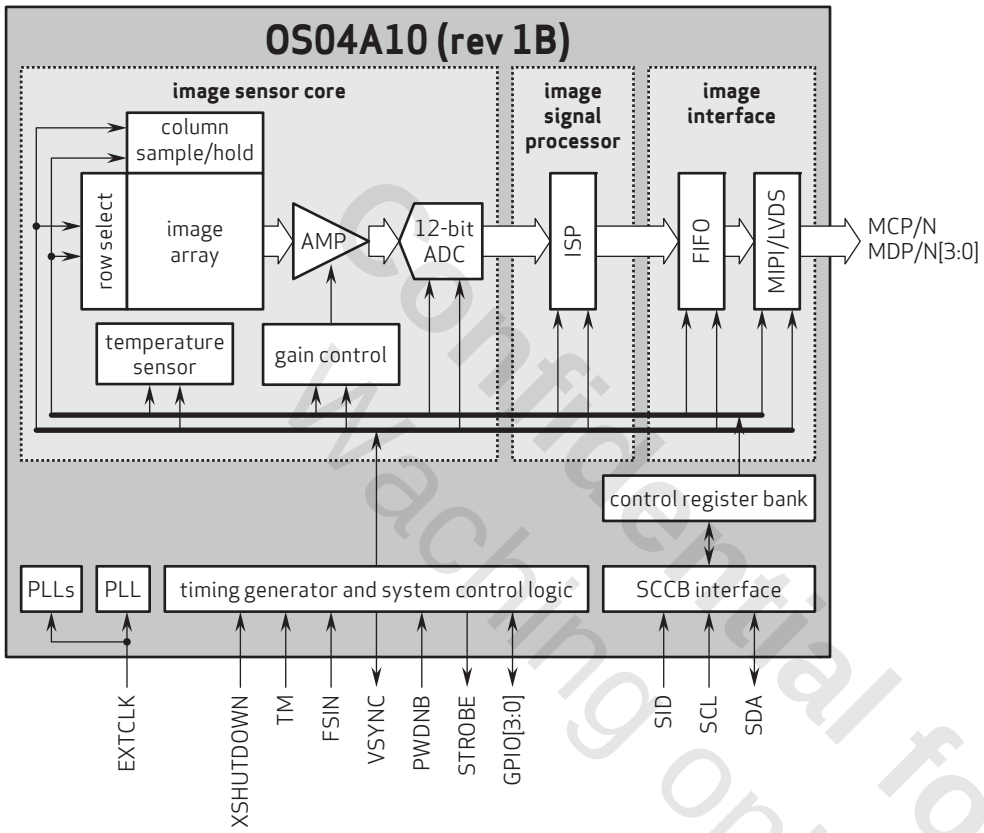
1.7.2 software reset

When register 0x0103[0] is configured as 1, all registers are reset to default value.

2 sensor architecture

figure 2-1 shows the top level block diagram of the OS04A10 sensor.

figure 2-1 OS04A10 block diagram



The sensor consists of three major functional blocks: image sensor core, image signal processor (ISP), and output interface.

The image sensor core receives the photo signal which generates electrical charge collected by the pixel photo diodes (PDs). During readout, the accumulated charge is converted to a voltage signal in the pixel. This signal is then amplified and converted to a digital signal by the analog-to-digital converter (ADC). Dark current and circuit offsets are compensated by the black level correction circuit (BLC). The correction is implemented purely in the digital domain. Dark current increases exponentially with temperature and the BLC can be configured to automatically re-trigger with changes in exposure, output format, applied gain, and out of range offsets, in addition to manually triggering it. A temperature sensor is integrated in the image sensor core.

The OS04A10 supports dual conversion gain (DCG) HDR, which means switchable two conversion gain (CG) in one exposure (integration time) – low conversion gain (LCG) for large charge handling capacity in bright scenes and a high

conversion gain (HCG) mode with increased sensitivity and low read noise for low-light scenes. Higher CG means higher sensitivity, as one signal electron can be more easily detected. Higher CG also means that the sensor will realize a reduction in read noise. The OS04A10 also supports very short (VS) exposure.

The output from the HCG and LCG channels are combined in the sensor to create a 90 dB (16-bit) HDR output image from the single exposure. The sensor also supports dual exposure staggered HDR for >120 dB (20-bit), but in this case the HDR combination with VS exposure is done externally. The ISP also supports defect pixel correction (DPC) and DCG. The processed linear or combined HDR image is formatted and output through the MIPI interface. Two on-chip phase lock loops (PLLs) generate the required clock signals for all blocks from the EXTCLK input clock. The timing generator generates the control signals for the pixel array to reset the PD at the beginning of the exposure, to stop the exposure by reading out the accumulated charge, and also to generate the required control timing for the readout amplifier and ADC. The sensor can output a vertical synchronization signal (VSYNC) to the backend processor. The sensor can also be synchronized by frame synchronization input (FSIN), which is helpful in multi-camera applications.

For security applications, the OS04A10 also supports a motion detection function when the sensor operates in ultra low power mode. The sensor can set a flag to notify the backend processor to take further measures, such as wake up the camera and start video recording.

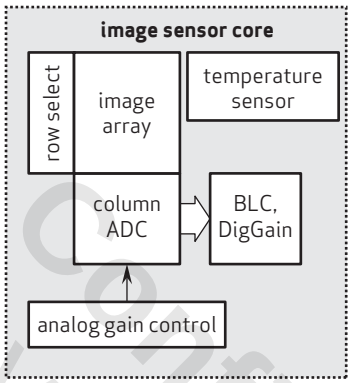
Calibration and identification information can be stored in one-time programmable memory (OTP).

All functional blocks are controlled by registers. The host controller can program and read back through the SCCB interface.

3 image sensor core

figure 3-1 shows the top level block diagram of the OS04A10 image sensor core.

figure 3-1 sensor core block diagram



The image sensor core consists of the active pixel array, row access control circuit, column parallel analog-to-digital converter (ADC) with gain control, and analog readout channel. A single analog readout channel is used for the processing of three capture channels (HCG, LCG, VS). This provides optimal matching between the capture channels. DigGain and BLC are implemented in digital domain.

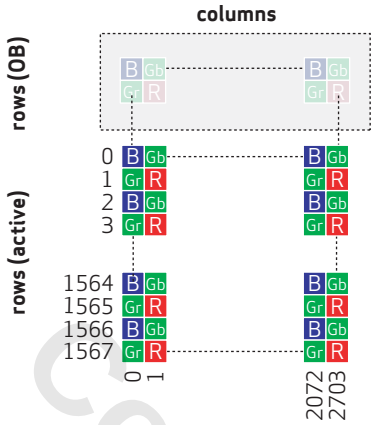
3.1 pixel array structure

The OS04A10 sensor, in total, has an image array of 2704 columns by 1568 rows (with an active image array of 2688x1520 contained within) covered with color filters arranged in a Bayer (B-Gb-Gr-R) pattern. **figure 3-2** shows the pixel array color filter layout. In addition to the active pixel rows, optical black (OB) pixel rows are embedded to serve as reference pixels for the black level correction (BLC). The OB rows are covered with a light shield (solid metal layer). In order to minimize non-ideal edge effects in the output image, it is not recommended to use the pixels at the lowest and highest row and column addresses.

The entire readable column is 2688 active columns + 8 active border columns + 8 extra active border columns = 2704 columns.

The entire readable row is 1520 active rows + 8 active border rows + 8 extra active border rows + 32 BLC rows = 1568 rows.

figure 3-2 pixel array region color filter layout

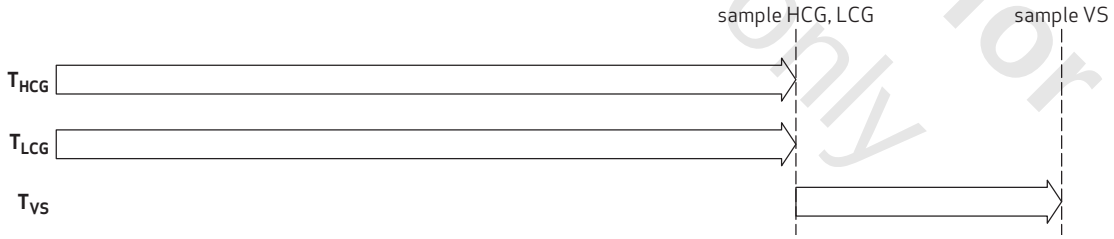


Each pixel has two switchable conversion gains (CG) to extend the dynamic range. Higher CG (HCG) means higher sensitivity, as one signal electron can be more easily detected. Lower CG (LCG) means lower sensitivity. The OS04A10 supports two exposures with three capture readout, with one exposure supporting two captures in the spatial domain and the shortest exposure in the time domain.

The integration time for VS (T_{VS}) will always start after (T_{HCG}/T_{LCG}) finish. The sampling point is (VS exposure time in rows + 1) rows after, but the T_{VS} start can move from end of T_{HCG}/T_{LCG} to just before sampling of TVS (see **figure 3-3**).

The OS04A10 also supports three exposures with three capture output (long exposure, medium exposure, and short exposure). The integration time for medium exposure will always start after long exposure finishes and the integration time for short exposure will always start after medium exposure.

figure 3-3 integration time diagram



The OS04A10 can operate in three modes:

- 3-exposure mode: HDR mode with all three exposures, one capture with long exposure, one with medium exposure, and one with short exposure
- dual exposure mode: HDR mode with two exposures, two captures with DCG (HCG and LCG combined) and VS valid
- single exposure mode: HDR mode with only DCG (HCG and LCG) valid

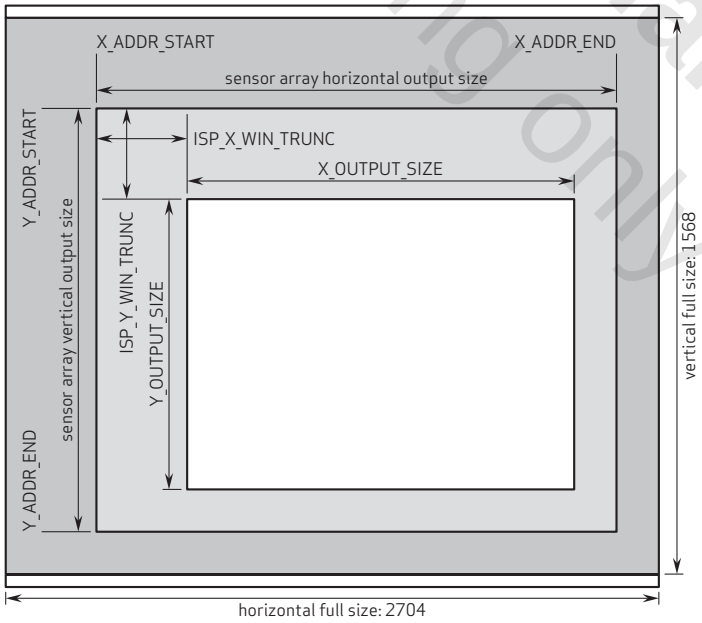
3.2 pixel array access

The readout image array size is fully programmable from 256 to 2704 in steps of 16 in the horizontal direction and 20 to 1536 in steps of 2 in the vertical direction. Start address must be at an even row and column and end address must be at an odd row and column address in order to preserve the Bayer pattern order. It represents the image array area that will be sampled and is defined by four 'crop' parameters - horizontal start (X_ADDR_START) {0x3800, 0x3801}, horizontal end (X_ADDR_END) {0x3804, 0x3805}, vertical start (Y_ADDR_START) {0x3802, 0x3803}, and vertical end (Y_ADDR_END) {0x3806, 0x3807}.

By properly setting the previously mentioned parameters, any portion within the sensor array size can be output as a visible area. This kind of windowing is achieved by masking off the pixels outside of the output window; thus, the original timing is not affected. The 'output window' is defined by an initial 'offset' from the 'crop' window, set in both horizontal (ISP_X_WIN_TRUNC) {0x3810, 0x3811} and vertical (ISP_Y_WIN_TRUNC) {0x3812, 0x3813} directions. Also, the actual image 'window size', that is going to be send out, is specified in width (X_OUTPUT_SIZE) {0x3808, 0x3809} and height (Y_OUTPUT_SIZE) {0x380A, 0x380B}. The offset address should always be an even number, while the output end address should always be an odd number.

The 'crop' window is programmed larger than the processed output image resolution because the ISP uses extra rows and columns for the image processing algorithms (e.g., defect pixel correction). Also, if the embedded row is enabled, the 'crop' window vertical size needs to be increased manually, depending on the number of these rows that had been enabled. These and other non-image array rows (like DTPR, ATPR, and row IDs) if enabled, will increase the 'output window' vertical size (of the output image) automatically, so the user does not need to adjust that in the sensor. It is only the receiver that needs to accommodate for the extra window height produced as a result.

figure 3-4 pixel array access diagram



3.3 mirror and flip

The pixel array can be accessed in the reverse order in column and row directions (i.e., image can be horizontally mirrored and vertically flipped) (see **figure 3-5**). Image flip is controlled by setting register bit 0x3820[2]. Mirror is controlled by register 0x3820[1]. The image mirror and flip setting is shown in **table 3-1**. The sensor needs to be in standby mode when implementing mirror or flip.

When either flip and/or mirror features are enabled, the vertical and/or horizontal crop sizes would have to be redefined by respectively adding two to vertical crop size {0x3806, 0x3807} for flip and incrementing the horizontal crop size {0x3804, 0x3805} for mirror.

table 3-1 mirror and flip registers

address	register name	default value	R/W	description
0x3820	FORMAT1	0x00	RW	Bit[2]: Flip Bit[1]: Mirror

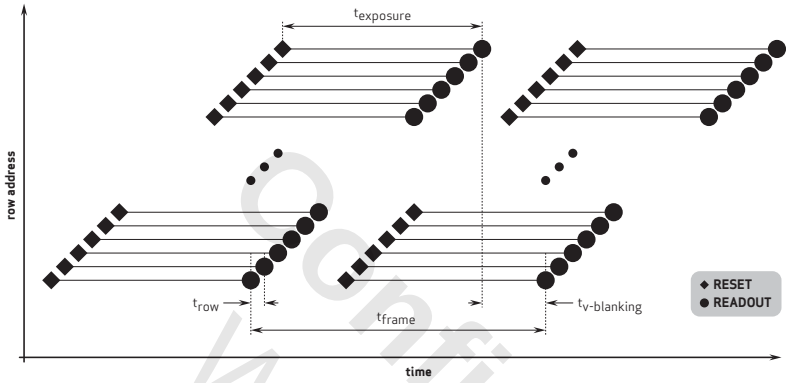
figure 3-5 horizontal mirror and vertical flip samples

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

3.4 frame timing and maximum frame rate

The OS04A10 employs an electronic rolling shutter (ERS) for exposure control (see **figure 3-6**). The pixel array is first reset row by row and when the exposure time has elapsed, the readout of the pixel array is done row by row.

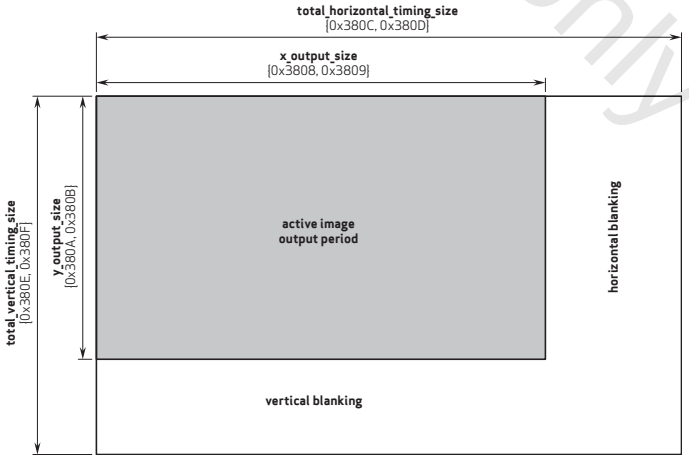
figure 3-6 row address versus time graph



The timing generator generates all the control signals based on a row counter and column counter. Refer to **figure 3-7** for frame timing. The row period consists of an active output period and a horizontal blanking period. A vertical blanking period is also required to perform frame-based operation. The vertical blanking period seen by the backend processor is usually longer than the internal vertical blanking because the BLC is reading the optical black rows required to perform the correction.

A minimum number of clock periods are required to complete all the required operations per row (blanking time). Refer to **section 6** for details.

figure 3-7 frame output timing diagram



The maximum frame rate is determined by the maximum system and pixel clocks, total number of pixels read out of the entire frame and minimum horizontal/vertical blanking time. The system clock and the minimum blanking time are usually fixed for a given design and the frame rate is dependent on the number of pixels read out. If the requested output image size (ISP window size) is smaller than the full pixel array, it is not necessary to read out the whole pixel array; thus, the frame rate can be increased by cropping the pixel array (see **figure 3-3**). **table 3-2** lists the most common image sizes and maximum frame rates that the sensor can achieve. Other image sizes are also possible by cropping. Refer to **section 6,** SCCB interface for details.

table 3-2 supported output formats and frame rates for MIPI

maximum frame rate supported via MIPI (4-lane @ 720 Mbps/lane) interface			
format		resolution (HxV)	maximum frame rate
single exposure linear (linear)	10-bit	2688x1520	90 fps
	12-bit	2688x1520	45 fps
single exposure HDR (DCG)	10-bit x 2	2688x1520	40 fps
	12-bit x 2	2688x1520	20 fps
	16-bit combined	2688x1520	40 fps
	12-bit (PWL) compressed combined	2688x1520	40 fps
dual exposure HDR (DCG + VS)	16-bit combined + 12-bit	2688x1520	15 fps
	12-bit (PWL) compressed combined	2688x1520	15 fps
	10-bit x 3	2688x1520	30 fps
	12-bit x 3	2688x1520	15 fps
3-exposure HDR (3 stagger HDR)	10-bit x 3	2688x1520	30 fps
	12-bit x 3	2688x1520	15 fps
2-exposure HDR (2 stagger HDR)	10-bit x 2	2688x1520	45 fps
	12-bit x 2	2688x1520	23 fps

table 3-3 timing control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x3800	TIMING_CTRL_0	0x00	RW	Bit[7:0]: Array horizontal start address[15:8]
0x3801	TIMING_CTRL_1	0x00	RW	Bit[7:0]: Array horizontal start address[7:0]
0x3802	TIMING_CTRL_2	0x00	RW	Bit[7:0]: Array vertical start address[15:8]

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

address	register name	default value	R/W	description
0x3803	TIMING_CTRL_3	0x00	RW	Bit[7:0]: Array vertical start address[7:0]
0x3804	TIMING_CTRL_4	0x0A	RW	Bit[7:0]: Array horizontal end address[15:8]
0x3805	TIMING_CTRL_5	0x8F	RW	Bit[7:0]: Array horizontal end address[7:0]
0x3806	TIMING_CTRL_6	0x05	RW	Bit[7:0]: Array vertical end address[15:8]
0x3807	TIMING_CTRL_7	0xFF	RW	Bit[7:0]: Array vertical end address[7:0]
0x3808	TIMING_CTRL_8	0x0A	RW	Bit[7:0]: Array horizontal output size for final image[10:8]
0x3809	TIMING_CTRL_9	0x80	RW	Bit[7:0]: Array horizontal output size for final image[7:0]
0x380A	TIMING_CTRL_10	0x05	RW	Bit[7:0]: Array vertical output size for final image[15:8]
0x380B	TIMING_CTRL_11	0xF0	RW	Bit[7:0]: Array vertical output size for final image[7:0]
0x380C	TIMING_CTRL_12	0x02	RW	Bit[7:0]: Array row timing length[15:8] Horizontal timing length (HTS) for (combined) DCG exposure part
0x380D	TIMING_CTRL_13	0xE2	RW	Bit[7:0]: Array row timing length[7:0] Horizontal timing length (HTS) for (combined) DCG exposure part
0x380E	TIMING_CTRL_14	0x06	RW	Bit[7:0]: Array frame timing[15:8] Vertical timing length (VTS) or total rows in one frame
0x380F	TIMING_CTRL_15	0x58	RW	Bit[7:0]: Array frame timing[7:0] Vertical timing length (VTS) or total rows in one frame
0x3810	TIMING_CTRL_16	0x00	RW	Bit[7:0]: ISP window horizontal truncation size[15:8]
0x3811	TIMING_CTRL_17	0x08	RW	Bit[7:0]: ISP window horizontal truncation size[7:0]
0x3812	TIMING_CTRL_18	0x00	RW	Bit[7:0]: ISP window vertical truncation size[15:8]
0x3813	TIMING_CTRL_19	0x08	RW	Bit[7:0]: ISP window vertical truncation size[7:0]

3.5 exposure and gain control

The OS04A10 has three exposures and channels: DCG will occupy one exposure and two data channels. There are two options for three data channels: DCG+VS (DCG for long exposure and VS for medium exposure) or L+M+S (L for long exposure, M for medium exposure, and S for short exposure).

The exposure time can be set manually in registers:

- minimum value for both DCG and VS exposures = 1 line
- maximum exposure value depends on (HDR) type:
  - in one exposure mode (linear mode or DCG only mode): single_exp_max = VTS - 8
  - in dual exposure mode (DCG+VS or L+M): l_exp_max = VTS - 8, m_exp_max = VTS - l_exp_max - 2
  - in 3-exposure mode (L+M+S): l_exp_max = VTS - 8; m_exp_max = VTS - l_exp_max - 2, s_exp_max = VTS - m_exp_max - 2
- {0x3501[7:0], 0x3502[7:0]} for DCG (HCG and LCG) or L exposure time
- {0x3541[7:0], 0x3542[7:0]} for M exposure time
- {0x3581[7:0], 0x3582[7:0]} for S exposure time

All exposure time values are represented by the unit of row time:

$$\text{row time} = \frac{\text{HTS}}{\text{SCLK}}$$

where SCLK is the system (core) clock frequency (Hz) and HTS is the horizontal size in number of SCLK cycles, which is defined by {0x380C, 0x380D} for DCG or long/medium channel and {0x384C, 0x384D} for very short exposure.

The exposure control is double frame-synced, meaning that a change in the setting of the current frame (N) will take place and the actual exposure will change on the following frame (at N+2). This period is fixed and cannot be adjusted.

The analog gain for the OS04A10, referred to as manual real gain, can be set individually for each exposure channel and it consists of two bytes. It follows 5.4 bits format, where the first 5b are held in the lower nibble of the first byte and represent an integer value. While the 4b represent the fractional part (units of 1/16) and is MSB aligned in the second (consecutive) byte. The value range of the analog gain is x1 ~ x16, where for the maximum value, only the fifth bit of the integer part needs to be to 1, while the rest of the valid bits across the two bytes need to be set to 0.

The digital gain for the OS04A10 can also be set individually for each exposure channel and it consists of three bytes. It follows 4.10 bits format, where the first 4b are held in the lower nibble of the first byte and represent an integer value. While the 10b represent the fractional part (units of 1/1023) and are distributed across the next two bytes in consecutive order with the MSB held in the second byte and the remaining two bits contained in the higher bits of the last byte. The value range of the digital gain is x1 ~ x15.99, where the maximum value needs to be set to 1 across all the valid bits.

table 3-4 exposure and gain control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x3501	AEC_PK_CORE_HCG_TOP_1	0x00	RW	Bit[7:0]: hcg_exposure_coarse[15:8] DCG exposure in unit of rows
0x3502	AEC_PK_CORE_HCG_TOP_2	0x40	RW	Bit[7:0]: hcg_exposure_coarse[7:0] DCG exposure in unit of rows
0x3508	AEC_PK_CORE_HCG_TOP_8	0x01	RW	Bit[4:0]: HCG real gain[8:4]
0x3509	AEC_PK_CORE_HCG_TOP_9	0x00	RW	Bit[7:4]: HCG real gain[3:0]
0x350A	AEC_PK_CORE_HCG_TOP_10	0x01	RW	Bit[3:0]: HCG digital gain[13:10]
0x350B	AEC_PK_CORE_HCG_TOP_11	0x00	RW	Bit[7:0]: HCG digital gain[9:2]
0x350C	AEC_PK_CORE_HCG_TOP_12	0x00	RW	Bit[7:6]: HCG digital gain[1:0]
0x3548	AEC_PK_CORE_LCG_TOP_8	0x01	RW	Bit[4:0]: LCG real gain[8:4]
0x3549	AEC_PK_CORE_LCG_TOP_9	0x00	RW	Bit[7:4]: LCG real gain[3:0]
0x354A	AEC_PK_CORE_LCG_TOP_10	0x01	RW	Bit[3:0]: LCG digital gain[13:10]
0x354B	AEC_PK_CORE_LCG_TOP_11	0x00	RW	Bit[7:0]: LCG digital gain[9:2]
0x354C	AEC_PK_CORE_LCG_TOP_12	0x00	RW	Bit[7:6]: LCG digital gain[1:0]
0x3581	AEC_PK_CORE_VS_TOP_1	0x00	RW	Bit[7:0]: Very short (VS) exposure in unit of rows[15:8]
0x3582	AEC_PK_CORE_VS_TOP_2	0x40	RW	Bit[7:0]: Very short (VS) exposure in unit of rows[7:0]
0x3588	AEC_PK_CORE_VS_TOP_8	0x01	RW	Bit[4:0]: VS real gain[8:4]
0x3589	AEC_PK_CORE_VS_TOP_9	0x00	RW	Bit[7:4]: VS real gain[3:0]
0x358A	AEC_PK_CORE_VS_TOP_10	0x01	RW	Bit[3:0]: VS digital gain[13:10]
0x358B	AEC_PK_CORE_VS_TOP_11	0x00	RW	Bit[7:0]: VS digital gain[9:2]

table 3-4 exposure and gain control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x358C	AEC_PK_CORE_VS_TOP_12	0x00	RW	Bit[7:6]: VS digital gain[1:0]
0x380C	TIMING_CTRL_12	0x02	RW	Bit[7:0]: Array row timing length[15:8] Horizontal timing length (HTS) for (combined) DCG exposure part
0x380D	TIMING_CTRL_13	0xE2	RW	Bit[7:0]: Array row timing length[7:0] Horizontal timing length (HTS) for (combined) DCG exposure part
0x380E	TIMING_CTRL_14	0x06	RW	Bit[7:0]: Array frame timing[15:8] Vertical timing length (VTS) or total rows in one frame
0x380F	TIMING_CTRL_15	0x58	RW	Bit[7:0]: Array frame timing[7:0] Vertical timing length (VTS) or total rows in one frame
0x384C	TIMING_CTRL_76	0x02	RW	Bit[7:0]: vs_hts[15:8] Horizontal timing length (HTS) for VS exposure part
0x384D	TIMING_CTRL_77	0xE2	RW	Bit[7:0]: vs_hts[7:0] Horizontal timing length (HTS) for VS exposure part

3.6 black level calibration (BLC)

The pixel array contains several optically shielded (black) rows positioned at the lower side of the array. That section of the image matrix is used as reference for black level calibration. The main function of the BLC is to adjust all normal pixel values based on the values of the black levels.

table 3-5 BLC control registers

address	register name	default value	R/W	description
0x4000	BLC_157_TOP_0	0xF8	RW	Bit[7]: off_trig_en Bit[6]: exp_chg_trig_en Bit[5]: gain_chg_trig_en Bit[4]: fmt_chg_trig_en Bit[3]: rst_trig_en Bit[2]: man_trig Bit[1]: off_frz_en Bit[0]: off_always_up
0x4001	BLC_157_TOP_1	0x2B	RW	Bit[7]: zero_ln_out_en Bit[6]: blk_ln_out_en Bit[5]: dither_en Bit[4]: off_man_en Bit[3]: mf_en Bit[2]: v15ne_channel Bit[1]: dc_blc_en Bit[0]: blc_en
0x4004	BLC_157_TOP_4	0x00	RW	Bit[3:0]: blk_lvl_target_HCG[11:8]
0x4005	BLC_157_TOP_5	0x40	RW	Bit[7:0]: blk_lvl_target_HCG[7:0]
0x402E	BLC_157_TOP_46	0x00	RW	Bit[3:0]: blk_lvl_target_LCG[11:8]
0x402F	BLC_157_TOP_47	0x40	RW	Bit[7:0]: blk_lvl_target_LCG[7:0]
0x4030	BLC_157_TOP_48	0x00	RW	Bit[3:0]: blk_lvl_target_VS[11:8]
0x4031	BLC_157_TOP_49	0x40	RW	Bit[7:0]: blk_lvl_target_VS[7:0]

3.7 PLL

PLL settings can only be changed during sensor standby mode (0x0100 = 0).

The OS04A10 implements two PLLs with both inputs connected to the EXTCLK pin. One can support the MIPI bit clock and output clock PCLK, while the other one can support internal SCLK. Two PLLs enable the internal clock (SCLK) to be separate from the output clock (PCLK). Additionally, in MIPI mode, the two PLLs can be used to optimize for EMC and/or minimize the required MIPI frequency.

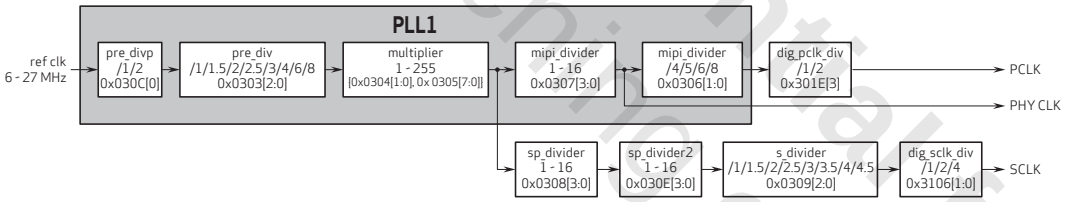
In order to reduce EMI's impact, this PLL also supports spread spectrum mode (SSM). Spread profile is a triangular waveform spread. The spectrum can be up spread, both side spread, or down spread depending on the input setting dsm[19:0]. Tssc is the modulation period. Fssc (1/Tssc) is the modulation frequency, which is about 30 kHz~33 kHz in many interface designs. DeltaF is modulation amplitude, which is normally smaller than 5000 ppm. For example, if normal clock frequency Fo = 1 GHz, then 5000 ppm = 5000 x Fo/1e6 = 5 MHz.

Maximum operating clock frequencies: SCLK = 108 MHz, PCLK = 138 MHz

3.7.1 PLL1

The PLL1 generates a default (maximum) 138 MHz pixel clock (PCLK) and 1104 MHz MIPI serial clock from a 6~36 MHz input clock. The VCO range is from 500 MHz to 1500 MHz. A programmable clock is provided to generate different frequencies.

figure 3-8 PLL1 control diagram



3.7.2 PLL2

The PLL2 generates a default 108 MHz system clock (SCLK) from a 6~36 MHz input clock. The VCO range is from 500 MHz to 1700 MHz. A programmable clock divider is provided to generate different frequencies.

figure 3-9 PLL2 control diagram

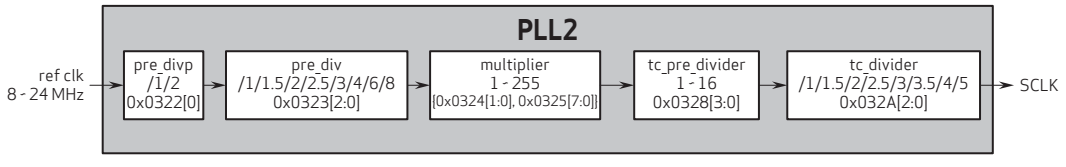


table 3-6 PLL control registers

address	register name	default value	R/W	description
0x0303	PLL_CTRL_TOP_3	0x02	RW	Bit[2:0]: pll1_prediv
0x0304	PLL_CTRL_TOP_4	0x00	RW	Bit[1:0]: pll1_divp[9:8]
0x0305	PLL_CTRL_TOP_5	0x5C	RW	Bit[7:0]: pll1_divp[7:0]
0x0306	PLL_CTRL_TOP_6	0x00	RW	Bit[1:0]: pll1_divmipi
0x0307	PLL_CTRL_TOP_7	0x00	RW	Bit[3:0]: pll1_divm
0x0308	PLL_CTRL_TOP_8	0x04	RW	Bit[3:0]: pll1_divsp
0x0309	PLL_CTRL_TOP_9	0x02	RW	Bit[2:0]: pll1_divs
0x030C	PLL_CTRL_TOP_12	0x00	RW	Bit[0]: pll1_predivp
0x0322	PLL_CTRL_TOP_34	0x00	RW	Bit[0]: pll2_predivp
0x0323	PLL_CTRL_TOP_35	0x02	RW	Bit[2:0]: pll2_prediv
0x0324	PLL_CTRL_TOP_36	0x00	RW	Bit[1:0]: pll2_divp[9:8]
0x0325	PLL_CTRL_TOP_37	0x6C	RW	Bit[7:0]: pll2_divp[7:0]
0x0326	PLL_CTRL_TOP_38	0x00	RW	Bit[1:0]: pll2_reserve1
0x0327	PLL_CTRL_TOP_39	0x05	RW	Bit[3:0]: pll2_divsram
0x0328	PLL_CTRL_TOP_40	0x05	RW	Bit[3:0]: pll2_divst
0x0329	PLL_CTRL_TOP_41	0x01	RW	Bit[3:0]: pll2_divdac
0x032A	PLL_CTRL_TOP_42	0x02	RW	Bit[2:0]: pll2_divt
0x301E	SC_MMN_30	0xB8	RW	Bit[3]: pclk_selection 0: PCLK 1: 1/2 PCLK
0x3106	SB_SCCB_6	0x10	RW	Bit[7:0]: SCLK SRAM clock and counter clock/divided clock selection

3.8 temperature sensor

The OS04A10 has two embedded temperature sensors in the image sensor core to measure its junction temperature. By default, the value of the temperature contains the average of the two. The value can be read directly from two registers, in a direct 8.8b format. Where the first byte (0x4F06) holds the integer part of the value and the second one (0x4F07) holds the decimal number of units of 1/255, all in degrees Celsius. The valid readout temperature range is from -63.996 (-3F.FF) to 192 (C0.00) °C.

To distinguish between + or - temperatures and to translate the register read values into actual temperatures, the procedure below must be followed:

If the read {0x4F06, 0x4F07} ≤ 0xC000, the temperature is positive. It can be interpreted directly with the 8.8b format.

If the read {0x4F06, 0x4F07} > 0xC000, the temperature is negative. The read value needs to be converted by subtracting it by 0xC000. For the remainder, apply 8.8b format again as normal. Keep in mind, the resulting value is absolute and the minus needs to be 'appended' for further calculations.

The temperature readout function is enabled by default. For the data to be accurate, the sensor needs to be set to streaming mode at least once so that the OTP calibration data for the temperature sensor is loaded from memory and can be utilized for the measurements.

table 3-7 temperature sensor registers

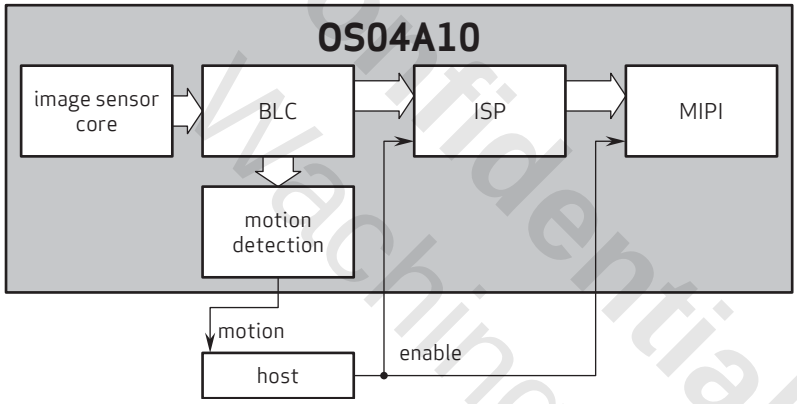
address	register name	default value	R/W	description
0x4F06	TPM_INT	–	R	Bit[7:0]: TPM temperature value integer part
0x4F07	TPM_DEC	–	R	Bit[7:0]: TPM temperature value decimal part

3.9 motion detection

The OS04A10 is designed for use in ultra-low power situations. In most applications, an image sensor is not required to output images all the time, which would be unnecessary and a large waste of energy. Motion detection was designed in order to make the image sensor more efficient on power consumption and more flexible on working mode.

The sensor works in power save mode by default. The image sensor core, BLC and motion detection modules are working, but the data path after BLC and other unused modules are idle. As shown in **figure 3-10**, the motion detection module receives image data from BLC, calculates the average illuminance of the current frame and compares it with the average illuminance of the previous frame. Through the comparison algorithm, the motion detection module determines that there is motion and sends this information to the host chip. The host chip wakes up the ISP, MIPI, and other functional blocks.

figure 3-10 motion detection module



3.9.1 motion detection working mode

3.9.1.1 image splitting options

The image from the BLC will be split into 4x4 or 8x8 sub-images as shown in **figure 3-11**. The image splitting mode setting should be changed during vertical blanking time; otherwise, it may confuse the motion detection module. The module provides options for some data path/image formats, which cannot be split. It provides a user-defined method to separate sub-images by register setting.

The motion detection module will calculate the average pixel illuminance for each zone and compare it with that in the same zone of the previous frame. Hence, it can report in which zone the motion occurs through a register read-out, while it can also disable motion detection in each zone to ignore motion in some areas.

figure 3-11 sub-images from BLC

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

3.9.1.2 image take-in options

This module provides options to determine which frame to take-in for calculation. Users can define every certain frames to take one frame's data into calculation. The current frame pixel illuminance is always compared with the last taken-in frame. Note that the first frame is always taken-in.

3.9.1.3 motion judgement

The difference between illuminances of the same zone of two consecutive detected frames will be used to decide whether it is defined as a "MOTION". The module provides two options by register setting. In fix mode, the difference will be compared with a value also defined by register. In flex mode, if the previous zone illuminance is smaller than a threshold (set by registers), the difference will be compared with some ratio of the previous zone illuminance. This ratio is set by registers. If the previous zone illuminance exceeds the threshold, it will be compared with a value defined by register.

3.9.1.4 motion output

This module will record the motion of each zone if the current frame is taken into calculation. This motion detection result can be read out by registers. Once there is a "MOTION" in a certain zone, a motion flag goes high immediately and can be read out by registers if this zone is not disabled. Another motion valid flag will go high at the end of the image frame if there was "MOTION" previously during current frame timing. The host may read these two flags to decide when to send a wake-up instruction back to the image sensor.

Additionally, this module provides options to perform motion detection continuously or not after the motion valid flag goes high. Thus, whether motion detection results are updated in registers or not, it can be changed by register setting.

3.9.2 motion detection registers

table 3-8 motion detection registers (sheet 1 of 5)

address	register name	default value	R/W	description
0x4380	MOTION_DETECT_TOP_0	0x01	RW	Bit[4]: Sub-image splitting option 0: 16 sub-images 1: 64 sub-images Bit[3]: Comparison threshold option 0: Only user defined comparison threshold 1: Two sub-image comparison threshold according to a ratio of image illuminance or user defined comparison threshold Bit[2:1]: Ratio map 00: Right shift 1 bit 01: Right shift 2 bits 10: Right shift 3 bits 11: Right shift 4 bits Bit[0]: Motion detection mode enable 0: Disable motion detection mode 1: Enable motion detection mode
0x4381	MOTION_DETECT_TOP_1	0x00	RW	Bit[7:0]: Number of frames between two sampled frames
0x4382	MOTION_DETECT_TOP_2	0x00	RW	Bit[7:0]: High byte of threshold of image illuminance change to be defined as a motion[DATA_DW-1:8]
0x4383	MOTION_DETECT_TOP_3	0x00	RW	Bit[7:0]: Low byte of threshold of image illuminance change to be defined as a motion[7:0]
0x4384	MOTION_DETECT_TOP_4	0x00	RW	Bit[7:0]: Below this threshold will trigger threshold ratio mode[7:0]
0x4385	MOTION_DETECT_TOP_5	0x00	RW	Bit[1]: Unused clean up result register 2 Bit[0]: Unused clean up result register 1
0x4386	MOTION_DETECT_TOP_6	0x00	RW	Bit[1]: Dummy register Bit[0]: Continue update sub_image motion result after motion is detected
0x4387	MOTION_DETECT_TOP_7	–	R	Bit[4]: sub_image motion is valid Bit[0]: Readable motion detection result

table 3-8 motion detection registers (sheet 2 of 5)

address	register name	default value	R/W	description
0x4388	MOTION_DETECT_TOP_8	0x00	RW	Bit[7]: Disable sub_img07 Bit[6]: Disable sub_img06 Bit[5]: Disable sub_img05 Bit[4]: Disable sub_img04 Bit[3]: Disable sub_img03 Bit[2]: Disable sub_img02 Bit[1]: Disable sub_img01 Bit[0]: Disable sub_img00
0x4389	MOTION_DETECT_TOP_9	0x00	RW	Bit[7]: Disable sub_img17 Bit[6]: Disable sub_img16 Bit[5]: Disable sub_img15 Bit[4]: Disable sub_img14 Bit[3]: Disable sub_img13 Bit[2]: Disable sub_img12 Bit[1]: Disable sub_img11 Bit[0]: Disable sub_img10
0x438A	MOTION_DETECT_TOP_A	0x00	RW	Bit[7]: Disable sub_img27 Bit[6]: Disable sub_img26 Bit[5]: Disable sub_img25 Bit[4]: Disable sub_img24 Bit[3]: Disable sub_img23 Bit[2]: Disable sub_img22 Bit[1]: Disable sub_img21 Bit[0]: Disable sub_img20
0x438B	MOTION_DETECT_TOP_B	0x00	RW	Bit[7]: Disable sub_img37 Bit[6]: Disable sub_img36 Bit[5]: Disable sub_img35 Bit[4]: Disable sub_img34 Bit[3]: Disable sub_img33 Bit[2]: Disable sub_img32 Bit[1]: Disable sub_img31 Bit[0]: Disable sub_img30
0x438C	MOTION_DETECT_TOP_C	0x00	RW	Bit[7]: Disable sub_img47 Bit[6]: Disable sub_img46 Bit[5]: Disable sub_img45 Bit[4]: Disable sub_img44 Bit[3]: Disable sub_img43 Bit[2]: Disable sub_img42 Bit[1]: Disable sub_img41 Bit[0]: Disable sub_img40

table 3-8 motion detection registers (sheet 3 of 5)

address	register name	default value	R/W	description
0x438D	MOTION_DETECT_TOP_D	0x00	RW	Bit[7]: Disable sub_img57 Bit[6]: Disable sub_img56 Bit[5]: Disable sub_img55 Bit[4]: Disable sub_img54 Bit[3]: Disable sub_img53 Bit[2]: Disable sub_img52 Bit[1]: Disable sub_img51 Bit[0]: Disable sub_img50
0x438E	MOTION_DETECT_TOP_E	0x00	RW	Bit[7]: Disable sub_img67 Bit[6]: Disable sub_img66 Bit[5]: Disable sub_img65 Bit[4]: Disable sub_img64 Bit[3]: Disable sub_img63 Bit[2]: Disable sub_img62 Bit[1]: Disable sub_img61 Bit[0]: Disable sub_img60
0x438F	MOTION_DETECT_TOP_F	0x00	RW	Bit[7]: Disable sub_img77 Bit[6]: Disable sub_img76 Bit[5]: Disable sub_img75 Bit[4]: Disable sub_img74 Bit[3]: Disable sub_img73 Bit[2]: Disable sub_img72 Bit[1]: Disable sub_img71 Bit[0]: Disable sub_img70
0x4390	MOTION_DETECT_TOP_10	–	R	Bit[7]: sub_img07 motion detected Bit[6]: sub_img06 motion detected Bit[5]: sub_img05 motion detected Bit[4]: sub_img04 motion detected Bit[3]: sub_img03 motion detected Bit[2]: sub_img02 motion detected Bit[1]: sub_img01 motion detected Bit[0]: sub_img00 motion detected
0x4391	MOTION_DETECT_TOP_11	–	R	Bit[7]: sub_img17 motion detected Bit[6]: sub_img16 motion detected Bit[5]: sub_img15 motion detected Bit[4]: sub_img14 motion detected Bit[3]: sub_img13 motion detected Bit[2]: sub_img12 motion detected Bit[1]: sub_img11 motion detected Bit[0]: sub_img10 motion detected

table 3-8 motion detection registers (sheet 4 of 5)

address	register name	default value	R/W	description
0x4392	MOTION_DETECT_TOP_12	–	R	Bit[7]: sub_img27 motion detected Bit[6]: sub_img26 motion detected Bit[5]: sub_img25 motion detected Bit[4]: sub_img24 motion detected Bit[3]: sub_img23 motion detected Bit[2]: sub_img22 motion detected Bit[1]: sub_img21 motion detected Bit[0]: sub_img20 motion detected
0x4393	MOTION_DETECT_TOP_13	–	R	Bit[7]: sub_img37 motion detected Bit[6]: sub_img36 motion detected Bit[5]: sub_img35 motion detected Bit[4]: sub_img34 motion detected Bit[3]: sub_img33 motion detected Bit[2]: sub_img32 motion detected Bit[1]: sub_img31 motion detected Bit[0]: sub_img30 motion detected
0x4394	MOTION_DETECT_TOP_14	–	R	Bit[7]: sub_img47 motion detected Bit[6]: sub_img46 motion detected Bit[5]: sub_img45 motion detected Bit[4]: sub_img44 motion detected Bit[3]: sub_img43 motion detected Bit[2]: sub_img42 motion detected Bit[1]: sub_img41 motion detected Bit[0]: sub_img40 motion detected
0x4395	MOTION_DETECT_TOP_15	–	R	Bit[7]: sub_img57 motion detected Bit[6]: sub_img56 motion detected Bit[5]: sub_img55 motion detected Bit[4]: sub_img54 motion detected Bit[3]: sub_img53 motion detected Bit[2]: sub_img52 motion detected Bit[1]: sub_img51 motion detected Bit[0]: sub_img50 motion detected
0x4396	MOTION_DETECT_TOP_16	–	R	Bit[7]: sub_img67 motion detected Bit[6]: sub_img66 motion detected Bit[5]: sub_img65 motion detected Bit[4]: sub_img64 motion detected Bit[3]: sub_img63 motion detected Bit[2]: sub_img62 motion detected Bit[1]: sub_img61 motion detected Bit[0]: sub_img60 motion detected

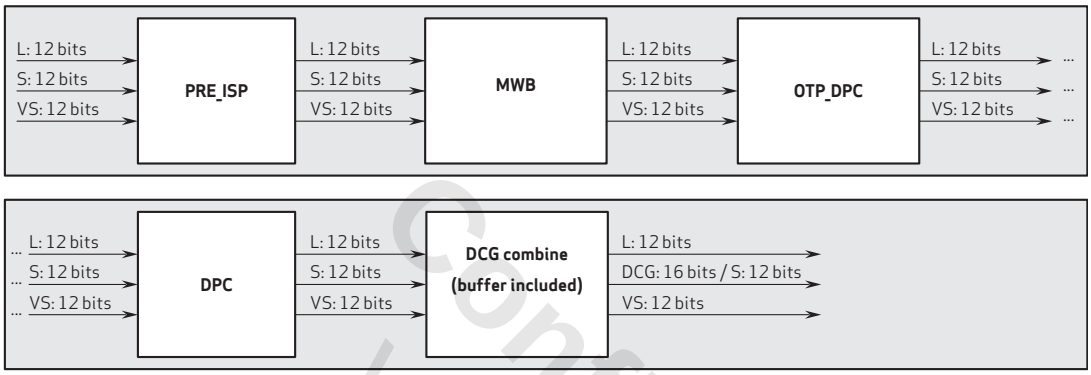
table 3-8 motion detection registers (sheet 5 of 5)

address	register name	default value	R/W	description
0x4397	MOTION_DETECT_TOP_17	–	R	Bit[7]: sub_img77 motion detected Bit[6]: sub_img76 motion detected Bit[5]: sub_img75 motion detected Bit[4]: sub_img74 motion detected Bit[3]: sub_img73 motion detected Bit[2]: sub_img72 motion detected Bit[1]: sub_img71 motion detected Bit[0]: sub_img70 motion detected
0x4398	MOTION_DETECT_TOP_18	0x00	RW	Bit[6:4]: Manual sub-image pixel counts in horizontal direction (should be n of 2**n)[2:0] Bit[3:1]: Manual sub_image pixel counts in vertical direction (should be n of 2**n)[2:0] Bit[0]: manu_div_en

4 image processor

figure 4-1 shows the top level block diagram of the OS04A10 image processor.

figure 4-1 image processor block diagram



The ISP receives image data from the sensor core and includes modules for RAW image processing. The video stream arrives as 12-bit parallel data separated in long exposure (L) or high conversion gain (HCG) in DCG mode, medium exposure (L) or low conversion gain (LCG) in DCG mode, and very short (VS) exposure channels. After processing the data from the ISP, it is configured to the correct output format in the output interface.

One of the first processing steps in the ISP (whether or not any of the available digital test patterns are enabled) is digital gain. Defect pixel and clusters (DPC) are corrected on-the-fly for each capture. After that, the DCG exposures (HCG and LCG) are combined into a 16-bit HDR image (DCG combine). In the transition area, the data is linearly combined. At the end, image windowing is performed if enabled. There is no lens correction block (LENC) in the OS04A10.

The main purpose of the ISP top includes:

- integrate all sub-modules
- create necessary control signals

4.1 test pattern

For testing purposes, the OS04A10 supports three types of test patterns: color bar, square, and random data. The OS04A10 also offers two digital effects: transparent effect and rolling bar effect. Some of the test patterns can be enabled for each of the exposure channels, L, S, and VS, individually. The digital test pattern is controlled by a set of three registers named PRE_ISP_0, each assigned for each channel at addresses: for L - 0x5080, for S - 0x50C0, and for VS - 0x5100.

The output type of the digital test pattern is controlled by the PRE_ISP_0[3:2] register. The digital test pattern function is either enabled/disabled by register PRE_ISP_0[7] for each channel. The data path test patterns are only available if the ISP is enabled (0x5000[0] = 1). The test patterns are processed by the ISP and outputted as regular images. However, they are unaffected by sensor exposure time and digital gain.

4.1.1 color bar

There are four types of color bars, which are switched by bar-style in register PRE_ISP_0[3:2] (see **figure 4-2**).

figure 4-2 color bar types

color bar type 1
PRE_ISP_0[3:2]=2'b00

color bar type 2
PRE_ISP[3:2]=2'b01

color bar type 3
PRE_ISP_0[3:2]=2'b10

color bar type 4
PRE_ISP_0[3:2]=2'b11

4.1.2 square

There are two types of squares: color square and black-white square. The register PRE_ISP_0[4] determines which type of square will be output

figure 4-3 color, black, and white square bars

color square black-white square

4.1.3 random data

There are two types of random data test patterns: frame-changing and frame-fixed random data.

4.1.4 transparent effect

The transparent effect is enabled by PRE_ISP_0[5]. If this register is set, the transparent test pattern will be displayed. **figure 4-4** is an example showing a transparent color bar image.

figure 4-4 transparent effect

4.1.5 rolling bar effect

The rolling bar is enabled by setting the registers PRE_ISP_0[6] for HCG, LCG, and VS to 1, at the same time. If it is enabled, an inverted-color rolling bar will roll from up to down. **figure 4-5** is an example showing a rolling bar on color bar image.

figure 4-5 rolling bar effect

table 4-1 test pattern control registers

address	register name	default value	R/W	description
0x5000	ISP_TOP_0	0x1F	RW	Bit[4]: Manual DCG enable Bit[3]: DPC enable Bit[2]: OTP_DPC enable Bit[1]: AWB gain enable Bit[0]: ISP enable
0x5001	ISP_TOP_1	0x05	RW	Bit[2]: Latch module enable Bit[1]: ISP module work manual mode enable Bit[0]: HREF before first VSYNC mask enable
0x5080	PRE_ISP_0_HCG	0x00	RW	Bit[7]: Test pattern enable Bit[6]: Rolling mode enable Bit[5]: Transparent mode enable Bit[4]: Black white square enable Bit[3:2]: Color bar option Bit[1:0]: Test pattern mode
0x50C0	PRE_ISP_0_LCG	0x00	RW	Bit[7]: Test pattern enable Bit[6]: Rolling mode enable Bit[5]: Transparent mode enable Bit[4]: Black white square enable Bit[3:2]: Color bar option Bit[1:0]: Test pattern mode
0x5100	PRE_ISP_0_VS	0x00	RW	Bit[7]: Test pattern enable Bit[6]: Rolling mode enable Bit[5]: Transparent mode enable Bit[4]: Black white square enable Bit[3:2]: Color bar option Bit[1:0]: Test pattern mode

4.2 white balance gain (WB gain)

The next process in the pipeline is white balance. The RAW red, green, and blue values of a gray object vary with the light source spectrum and the pixel QE spectrum response. Light source spectrum is usually described by "color temperature", which is the surface temperature of a black body radiating equivalent spectrum. In the real world, the light color temperature ranges from very low (reddish) to very high (bluish) value. For example, the color temperature of an incandescent lamp is about 2850K, while the color temperature of an overcast day is about 6500K.

To make sure that a gray image is truly gray regardless of the light spectrum, the sensor needs to adjust the gain for each RGB channel according to color temperature. This process is called white balance (WB) and it can be adjusted manually.

Thus, OS04A10 WB gain registers can be controlled by a separate (external to the sensor) ISP processor.

White balance gain is enabled by default and can be disabled by register 0x5000[7]. It is controlled by registers (allotted for each capture channel and color channel individually), 0x5180~0x5187 for L, 0x51A0~0x51A7 for S, and 0x51C0~0x51C7 for VS, where each range is comprised of 4 byte-pairs (2 bytes) ordered for each color channel, B[0], Gb[1], Gr[2], and R[3], respectively. Each WB gain is set as a 15-bit value occupying the LSBs part of each byte pair. The format is 5.10b, where the integer part is contained in the higher bits [6:2] of the first register, while the fractional part is stored in the lower bits [1:0] of the first register and continues through all the bits [7:0] in the second register. For example, a WB gain of 1x for L, blue color is {0x5180, 0x5181} = 0x0040.

Adjustable white balance gain does not have an internal latch function. Once it is set, it will take effect immediately, during the same (current) frame.

table 4-2 WB control registers (sheet 1 of 2)

address	register name	default value	R/W	description
0x5000	ISP_TOP_0	0x1F	RW	Bit[4]: Manual DCG enable Bit[3]: DPC enable Bit[2]: OTP_DPC enable Bit[1]: AWB gain enable Bit[0]: ISP enable
0x5180	AWB_GAIN_0	0x04	RW	Bit[6:0]: L manual AWB gain B[14:8]
0x5181	AWB_GAIN_1	0x00	RW	Bit[7:0]: L manual AWB gain B[7:0]
0x5182	AWB_GAIN_2	0x04	RW	Bit[6:0]: L manual AWB gain Gb[14:8]
0x5183	AWB_GAIN_3	0x00	RW	Bit[7:0]: L manual AWB gain Gb[7:0]
0x5184	AWB_GAIN_4	0x04	RW	Bit[6:0]: L manual AWB gain Gr[14:8]
0x5185	AWB_GAIN_5	0x00	RW	Bit[7:0]: L manual AWB gain Gr[7:0]
0x5186	AWB_GAIN_6	0x04	RW	Bit[6:0]: L manual AWB gain R[14:8]
0x5187	AWB_GAIN_7	0x00	RW	Bit[7:0]: L manual AWB gain R[7:0]
0x51A0	AWB_GAIN_0	0x04	RW	Bit[6:0]: S manual AWB gain B[14:8]

table 4-2 WB control registers (sheet 2 of 2)

address	register name	default value	R/W	description
0x51A1	AWB_GAIN_1	0x00	RW	Bit[7:0]: S manual AWB gain B[7:0]
0x51A2	AWB_GAIN_2	0x04	RW	Bit[6:0]: S manual AWB gain Gb[14:8]
0x51A3	AWB_GAIN_3	0x00	RW	Bit[7:0]: S manual AWB gain Gb[7:0]
0x51A4	AWB_GAIN_4	0x04	RW	Bit[6:0]: S manual AWB gain Gr[14:8]
0x51A5	AWB_GAIN_5	0x00	RW	Bit[7:0]: S manual AWB gain Gr[7:0]
0x51A6	AWB_GAIN_6	0x04	RW	Bit[6:0]: S manual AWB gain R[14:8]
0x51A7	AWB_GAIN_7	0x00	RW	Bit[7:0]: S manual AWB gain R[7:0]
0x51C0	AWB_GAIN_0	0x04	RW	Bit[6:0]: VS manual AWB gain B[14:8]
0x51C1	AWB_GAIN_1	0x00	RW	Bit[7:0]: VS manual AWB gain B[7:0]
0x51C2	AWB_GAIN_2	0x04	RW	Bit[6:0]: VS manual AWB gain Gb[14:8]
0x51C3	AWB_GAIN_3	0x00	RW	Bit[7:0]: VS manual AWB gain Gb[7:0]
0x51C4	AWB_GAIN_4	0x04	RW	Bit[6:0]: VS manual AWB gain Gr[14:8]
0x51C5	AWB_GAIN_5	0x00	RW	Bit[7:0]: VS manual AWB gain Gr[7:0]
0x51C6	AWB_GAIN_6	0x04	RW	Bit[6:0]: VS manual AWB gain R[14:8]
0x51C7	AWB_GAIN_7	0x00	RW	Bit[7:0]: VS manual AWB gain R[7:0]

4.3 defective pixel cancelation (DPC)

Defect pixels are defined as the pixels with a high possibility to be brighter or darker than its neighboring pixels, which includes both dead pixels and damaged pixels. When sensor gain exposure time or temperature increases, there are more defect pixels.

Defect pixel cancelation (referred to as DPC) is an online defect pixel detection and correction algorithm, which can be enabled by setting 0x5000[3] = 1. The bit must only be toggled while the sensor is in software standby.

The main purpose of the DPC function is to remove these white/black pixels. If the pixel is defective, DPC will use a value calculated from the neighboring normal pixels to replace it. DPC has two work modes: normal mode and stagger HDR mode. In stagger HDR mode, long, short and very short exposures each has its own parameters. Long exposure's register addresses start from 0x5380, short exposure's register addresses start from 0x5400, and very short exposure's register addresses start from 0x5480. Thus, DPC can process three frames separately.

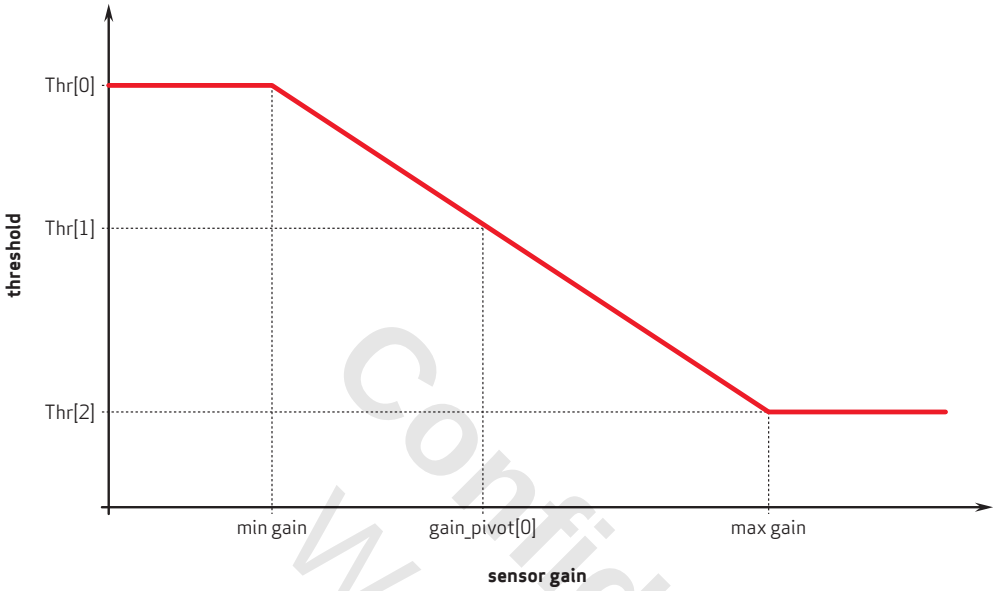
The case reviewed below focuses on L channel registers.

DPC determines whether a pixel is defect or not based on the detection threshold. With a smaller detection threshold, more pixels are detected as white or black pixels and corrected, which results in blurrier images. The defect black pixel cancellation and defect white pixel cancellation are controlled by m_bBlackPixel (0x5380[1]) and m_bWhitePixel (0x5380[0]), respectively. Because a pixel is more likely to be a defect pixel when there is already one defect pixel within its neighbors, DPC needs to remove not only single defect pixels, but also clusters of defect pixels. DPC supports different types of defects. The user can enable these types by setting registers 0x5381, 0x5382, 0x5383, and 0x5384.

In manual mode (when 0x5380[4] = 1'b1), the white pixel threshold is stored in register 0x5396 and black pixel threshold is defined in register 0x5399. In auto mode (when 0x5380[4] = 1'b0), the thresholds are controlled by sensor gain.

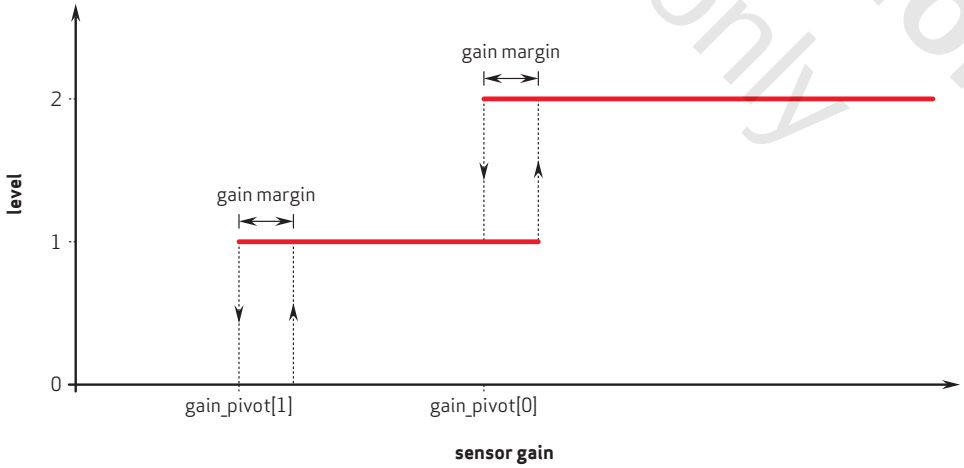
In addition to the previously described detection threshold requirements, the difference between a defect pixel and the reference should also be greater than the difference threshold, which is calculated as (difference_ratio)* reference / 32 (reference is the nearby pixel data). The difference ratio list (registers 0x538E, 0x538F, and 0x5390) and two threshold lists are defined in registers 0x5392 and 0x5393. One threshold list is for white pixel detection (registers 0x5396, 0x5397, and 0x5398) and the other is for black pixel detection (registers 0x5399, 0x539A, and 0x539B). The thresholds and difference ratio work with the current sensor gain for defect pixel detection. They form three piece-wise curves. **figure 4-6** shows an example of the threshold curve.

figure 4-6 example of DPC threshold curve



At different luminance, DPC corrects different types of defect clusters. By default, the least types of defect clusters are corrected under highlight (level 0) and most types of defect clusters are corrected under low light (level 2). The user can configure one set of registers (0x5381, 0x5382, 0x5383, 0x5384, 0x5385, 0x5386) for each level (0, 1, 2) for white and black pixels, respectively. Instead of jumping from one level to another, a transition between two states is defined by using gain_margin (register 0x5394) and gain_pivot (registers {0x5392, 0x5393}). **figure 4-7** shows the relationship between gain and level.

figure 4-7 relationship between gain and level



4.4 DCG combine

The combine_option is configured by register bits 0x5781[5:4]. dcg_combine data is calculated from L and S. In this mode, initial DCG output is equal to (LR*(128-w) + (l_data_i-blc_l_i)*w+16)>>5. DCG will use L exposure's data to calculate weight factor, w, with parameters, m_nCombineTL (0x5802) and m_nCombineTH (0x5803). If m_nCombineTH becomes bigger, long channel data will occupy greater weight for output DCG data.

First, normalize processing, enabled by register bit 0x5780[3]. Use the estimated gain to make saturate pixel in LCG also saturated in the final 16 bits of the image.

Second, apply the dither option, enabled by setting register bit 0x5780[5] to 1.

Finally, add the BLCTarget ({0x5784, 0x5785}), after saturation, to retrieve the data output.

1: dcg_combine data is calculated from L directly.

2: dcg_combine data is calculated from S directly.

table 4-3 combine control register

address	register name	default value	R/W	description
0x3671	ANA_TOP_113	0x0B	RW	Bit[2]: Enable DCG combine

5 image output interface

5.1 image output format

table 5-1 summarizes the output formats which the OS04A10 supports.

table 5-1 image output format summary

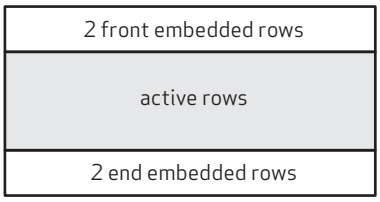
format	resolution	frame rate	MIPI / lane	comments
full (1520p)	2688x1520	90 fps	720 Mbps	Staggered HDR mode: Full size 30x3 fps @ 10-bit Full size 15x3 fps @ 12-bit DCG+VS HDR mode: Full size 30x3 fps @ 10-bit Full size 15x3 fps @ 12-bit DCG_combine+VS HDR mode: Full size 30x2 fps @ 10-bit Full size 15x2 fps @ 12-bit DCG_combine_pwl+VS HDR mode: Full size 30x2 fps @ 10-bit Full size 15x2 fps @ 12-bit DCG_combine only (16b) HDR mode: Full size 45 fps @ 10-bit Full size 24 fps @ 12-bit DCG_combine_pwl only (12b) HDR mode: Full size 45 fps @ 10-bit Full size 24 fps @ 12-bit Linear mode: Full size 90 fps @ 10-bit Full size 45 fps @ 12-bit
720p	1280x720	180 fps	720 Mbps	cropping
360p (VGA)	640x360	360 fps	720 Mbps	2x2 binning and cropping
180p (QVGA)	320x180	720 fps	720 Mbps	2x2 binning, skipping, and cropping
90p (QQVGA)	160x90	1440 fps	720 Mbps	2x2 binning, 2x2 skipping, and cropping

The enabling of each one of the modes holds a large sequence of register writes that has to be performed, which in itself configures multiple logic blocks in the sensor. Thus, individual settings will cover those.

The OS04A10 supports the formats shown in **figure 5-1** with full output size: horizontal (2688) x vertical (1520).

All sub-sizes also support all HDR modes. For two exposure staggered HDR, the frame rate for each exposure frame will be 1/2. For three exposure staggered HDR, the frame rate for each exposure frame will be 1/3.

figure 5-1 output image structure



5.2 data compression algorithm

5.2.1 16-bit to 12-bit

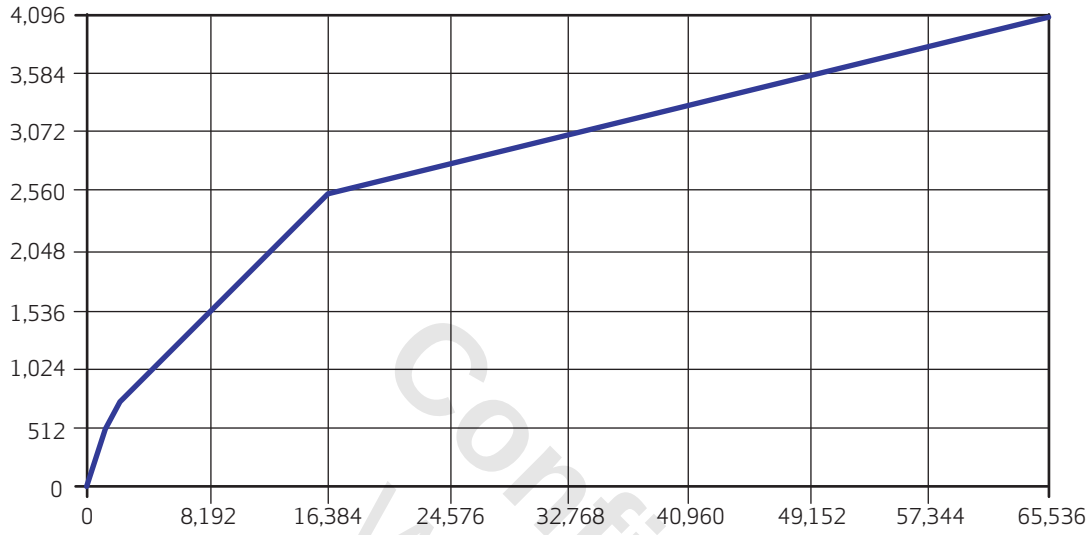
The OS04A10 offers data compression from 16-bit to 12-bit by providing a single piece-wise linear (PWL) method, with three options, utilizing different numbers of different fixed value knee points. The image data compression is enabled by setting 0x3671[5] = 1. The three different options can be selected via 0x3671[4:3] and are defined by the following formulas and curves shown in **figure 5-2** to **figure 5-4**.

5.2.1.1 option A compression – 0x3671[4:3] = 0

There are three fixed knee points, segmenting the value curve into four sections. The compression formula from 16-bit to 12-bit is given below for this particular case:

$$y_{out\_12b} = \begin{cases} \frac{y_{in\_16b}}{2}, & y_{in\_16b} < 1024 \\ \frac{y_{in\_16b}}{4} + 256, & 1024 \le y_{in\_16b} < 2048 \\ \frac{y_{in\_16b}}{8} + 512, & 2048 \le y_{in\_16b} < 16384 \\ \frac{y_{in\_16b}}{32} + 2048, & y_{in\_16b} \ge 16384 \end{cases}$$

figure 5-2 16-bit to 12-bit PWL compression, option A



The backend processor can decompress (option A) 12-bit data to 16-bit data using the following formula:

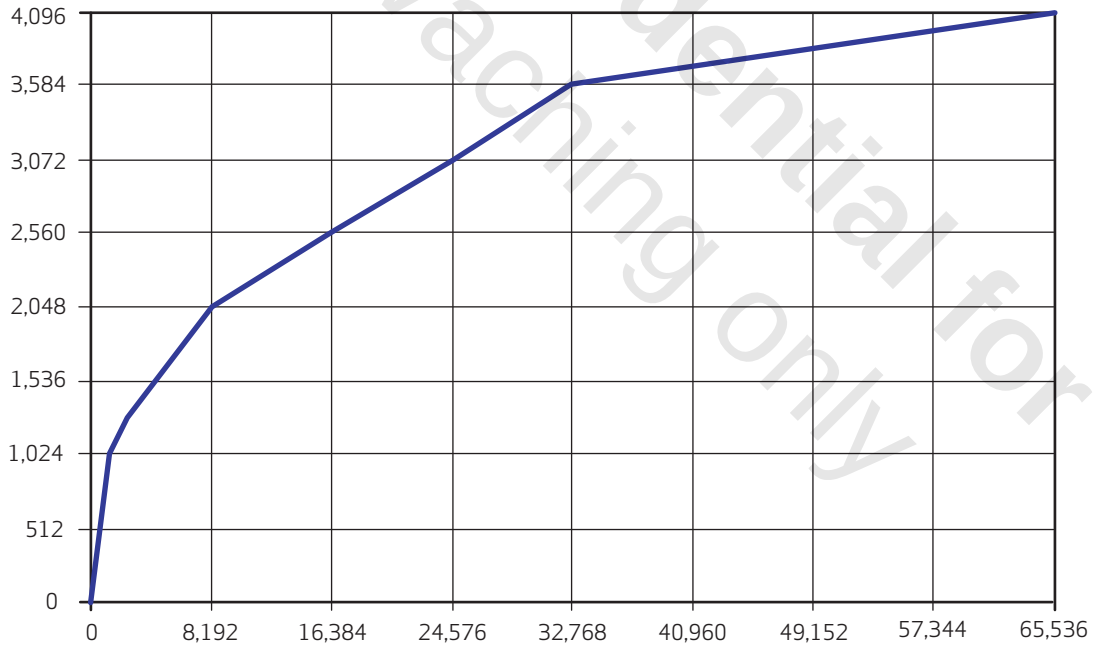
$$y_{out\_16b} = \begin{cases} 2 \times y_{in\_12b} & y_{in\_12b} < 512 \\ 4 \times (y_{in\_12b} - 256), & 512 \le y_{in\_12b} < 768 \\ 8 \times (y_{in\_12b} - 512), & 768 \le y_{in\_12b} < 2560 \\ 32 \times (y_{in\_12b} - 2048), & y_{in\_12b} \ge 2560 \end{cases}$$

5.2.1.2 option B compression – 0x3671[4:3] = 1

There are four fixed knee points, segmenting the value curve into five sections. The compression formula from 16-bit to 12-bit is given below for this particular case:

$$y_{out_{12b}} = \begin{cases} yin16b, & yin16b < 1024 \\ \frac{yin16b}{4} + 768, & 1024 \leq yin16b < 2048 \\ \frac{yin16b}{8} + 1024, & 2048 \leq yin16b < 8192 \\ \frac{yin16b}{16} + 1536, & 8192 \leq yin16b < 32768 \\ \frac{yin16b}{64} + 3072, & yin16b \geq 32768 \end{cases}$$

figure 5-3 16-bit to 12-bit PWL compression, option B



The backend processor can decompress (option B) 12-bit data to 16-bit data using the following formula:

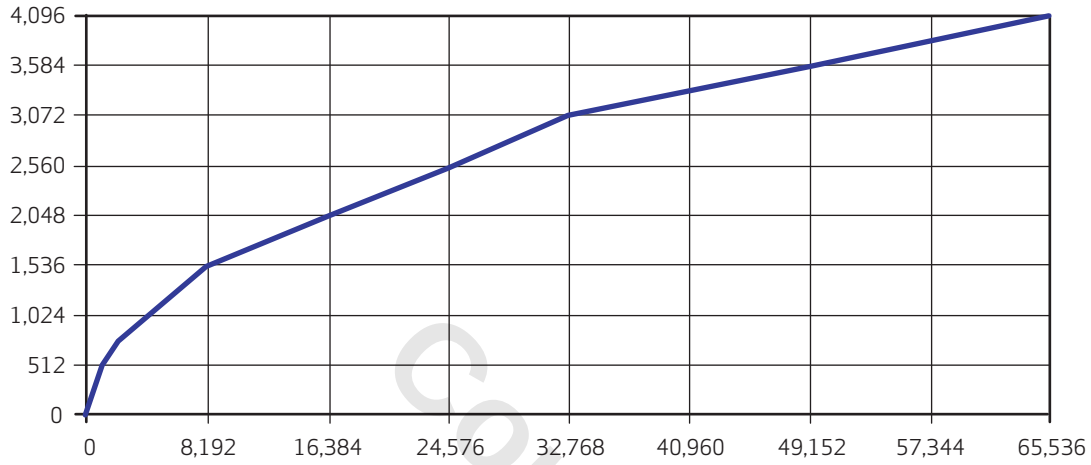
$$y_{out16b} = \begin{cases} y_{in12b}, & y_{in12b} < 1024 \\ 4 \times (y_{in12b} - 768), & 1024 \leq y_{in12b} < 1280 \\ 8 \times (y_{in12b} - 1024), & 1280 \leq y_{in12b} < 2048 \\ 16 \times (y_{in12b} - 1536), & 2048 \leq y_{in12b} < 3584 \\ 64 \times (y_{in12b} - 3072), & y_{in12b} \geq 3584 \end{cases}$$

5.2.1.3 option C compression – 0x3671[4:3] = 2

There are five fixed knee points, segmenting the value curve into six sections. The compression formula from 16-bit to 12-bit is given below for this particular case:

$$y_{out_{12b}} = \begin{cases} yin16b, & yin16b < 256 \\ \frac{yin16b}{2} + 128, & 256 \leq yin16b < 512 \\ \frac{yin16b}{4} + 256, & 512 \leq yin16b < 2048 \\ \frac{yin16b}{8} + 512, & 2048 \leq yin16b < 8192 \\ \frac{yin16b}{16} + 1024, & 8192 \leq yin16b < 32768 \\ \frac{yin16b}{32} + 2048, & yin16b \geq 32768 \end{cases}$$

figure 5-4 16-bit to 12-bit PWL compression, option C



The backend processor can decompress (option C) 12-bit data to 16-bit data using the following formula:

$$y_{out16b} = \begin{cases} y_{in12b}, & y_{in12b} < 256 \\ 2 \times (y_{in12b} - 128), & 256 \le y_{in12b} < 384 \\ 4 \times (y_{in12b} - 256), & 384 \le y_{in12b} < 768 \\ 8 \times (y_{in12b} - 512), & 768 \le y_{in12b} < 1536 \\ 16 \times (y_{in12b} - 1024), & 1536 \le y_{in12b} < 3072 \\ 32 \times (y_{in12b} - 2048), & y_{in12b} \ge 3072 \end{cases}$$

Option A is suitable in high light conditions, when losing one bit in the lower range is acceptable and the user wants to keep the maximum bits in the higher ranges. For low light conditions, option B is preferable. All the bits in the lower range are maintained, while only losing more bits in the higher range where it is more acceptable. Option C is a hybrid of A and B. Option C retains all the bits in the very low range and maintains the maximum bits in the higher ranges.

5.3 HDR output

In staggered HDR mode, HCG/LCG exposure frames are overlapping with VS exposures. This reduces the timing delay between different exposure frames, which will be combined into one HDR frame. It also reduces the frame/line buffer requirement for the backend chip.

5.3.1 MIPI

The MIPI interface supports 1, 2, or 4 lanes. The data output format is the same regardless of how many lanes are used with MIPI. The data packet illustrations in this section are to be interpreted line-wise as one consecutive data stream and show the layout of the data packet. The data packet is divided up between the lanes per byte (8 bits) according to MIPI CSI-2.

For non-staggered HDR, the OS04A10 outputs the combined DCG capture over a single virtual channel, where each capture is output in a single frame as shown in **figure 5-5**. Short-packets denote frame-start (FS) and frame-end (FE) on the respective virtual channel. If only one capture is output from the sensor, virtual channel 0 is used as default.

For staggered HDR, the OS04A10 outputs multiple exposures over individual virtual channels or a single shared virtual channel. In either case, each exposure is output staggered as shown in **figure 5-7** and **figure 5-8**. The exposures are staggered. Short-packets denote FS and FE in their respective virtual channel. If a shared virtual channel is used, only one FS and FE packet will be sent per frame and dummy data will be sent in the absence of actual exposure data as shown in **figure 5-9**.

The VS exposure plus HCG/LCG exposure should be less than vertical total size (VTS):

Max_exposure_VS + Max_exposure_HCG/LCG < VTS - 10

For 3-exposure stagger HDR mode, the sum of the three exposures should be less than the vertical total size:

Max_exposure_L + max_exposure_M + max_exposure_S < VTS - 12

figure 5-5 non-staggered HDR with MIPI virtual channel diagram

figure 5-6 non-staggered HDR with MIPI virtual channel detail diagram

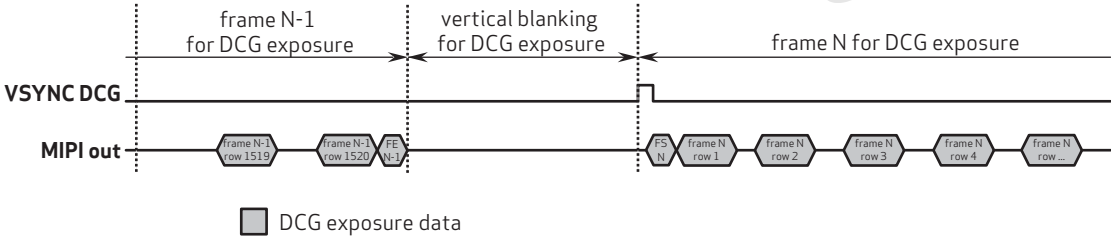


figure 5-7 staggered HDR with MIPI virtual channel diagram

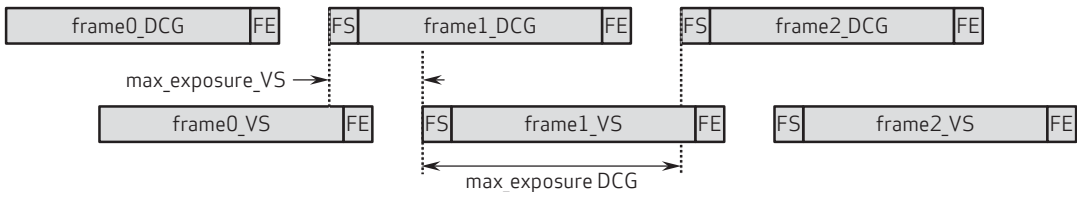


figure 5-8 staggered HDR with MIPI virtual channel detail diagram

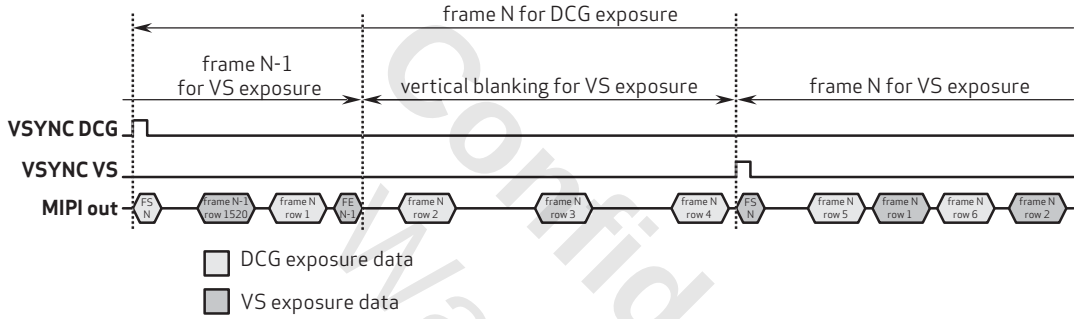


figure 5-9 staggered HDR with (single) shared MIPI virtual channel overview diagram

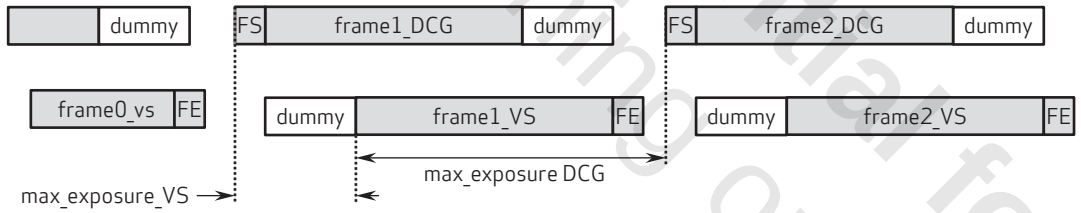
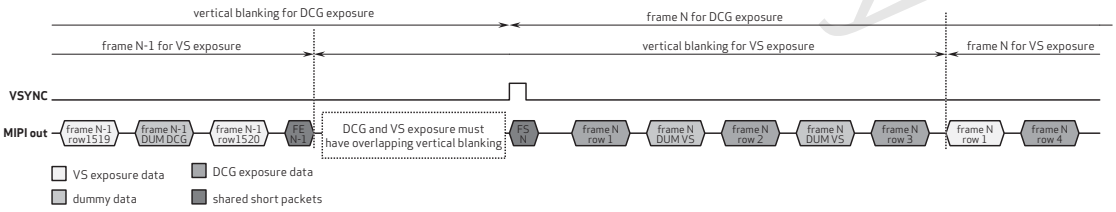


figure 5-10 staggered HDR with (single) shared MIPI virtual channel detail diagram



As a part of the MIPI protocol, when each row of the image (sensor) data is transmitted to the receiver at the beginning of each package, there is header information appending the rest of the row data being sent out. This header consists of 6 bits (wide), and its value can be specified. The application side of this feature implies that, if the MIPI receiver has the capability to recognize this header individually for each row, it can distinguish which type of output image data each row of incoming MIPI data belongs to (see **figure 5-1**).

table 5-2 defines the default MIPI RAW image data type codes (the header value), which belong to the active rows, when being sent out (see **figure 5-1**). The data type codes, can change and/or be fixed values, depending on of the MIPI RAW image data (packetizing) format.

Similar to **table 5-2**, **table 5-3** describes the data type codes which can be set for the rest (other type) of the image output rows. It is important to note that when the DCG16b + VS12b mode is used, the two exposures have different image data widths and will carry different MIPI data types as a result. Therefore, another item to consider is the effect on the other (non-image array) rows that will be output on these exposure channels. They may need to have two different data types as well, one for DCG and for VS. For example, the embedded data can be displayed on both DCG and VS channels at the same time. This depends on how the MIPI receiver (application processor or ASIC) implementation works. To accommodate for a scenario where it is desirable to distinguish which non-image row belongs to which exposure, registers 0x3665~0x3669 specify the MIPI DT for these (non-image) rows on VS. For the same type of rows on DCG, their DT can be set to be the same as those on VS, by 0x3665[7] = 0 or can be set to be different, by using the VS's MIPI DT of those rows with the DT bits[2:1] inverted, by 0x3665[7] = 1.

table 5-2 MIPI RAW image data types for active rows

default data type	register address	description
2C	–	RAW 12 (there is no register to change data type, it is fixed value)
2A	0x3673[5:0]	RAW 16

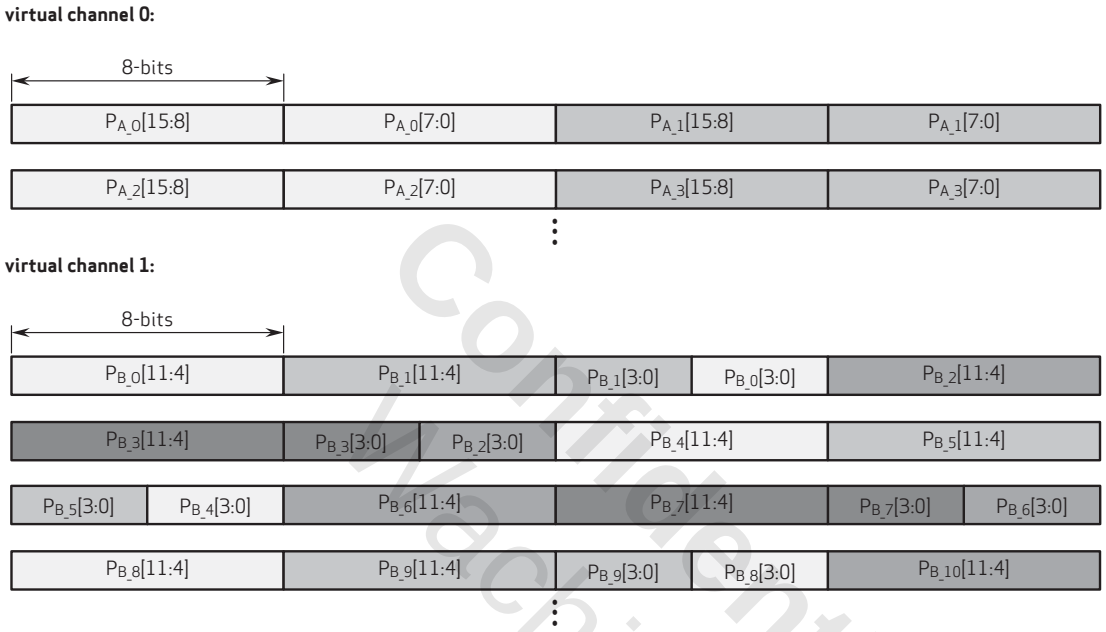
table 5-3 MIPI RAW image data types for other rows

default data type	register address	description
12	0x3665[5:0]	for (front and end) embedded data rows - data type

5.3.1.1 16b DCG + 12b VS dual HDR

Two values per pixel: PA (16-bit combined DCG), PB (12-bit RAW)

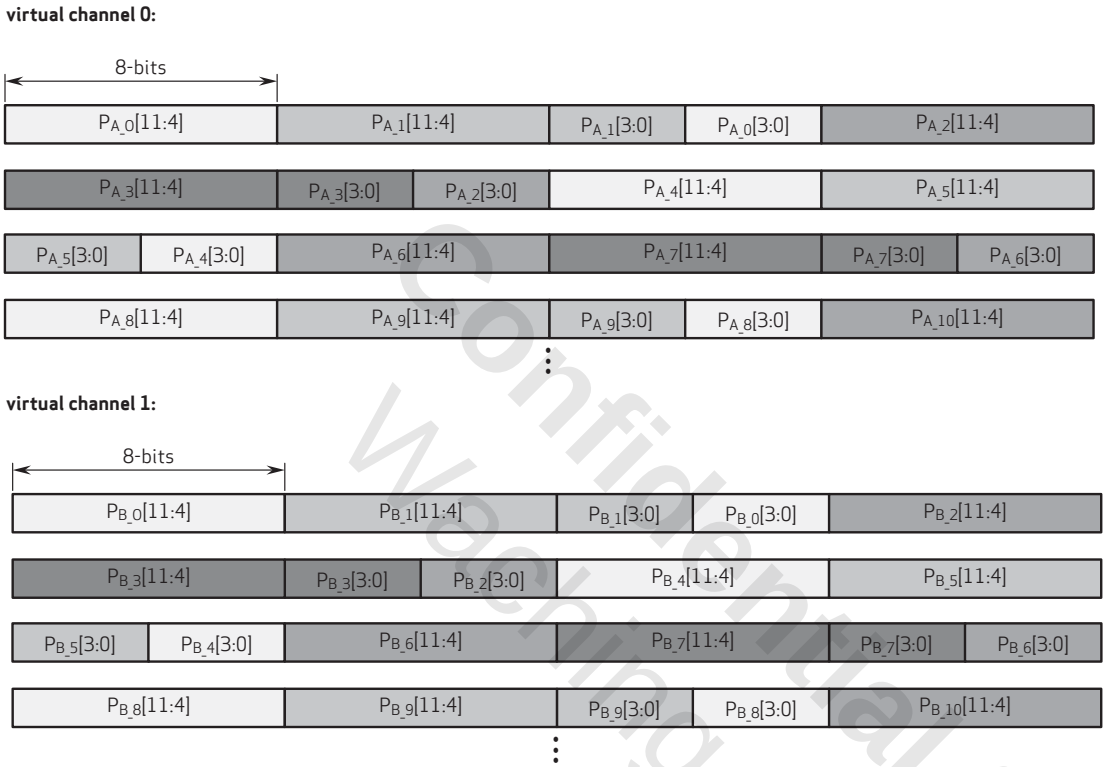
figure 5-11 16b DCG + 12b dual HDR diagram



5.3.1.2 12b compressed DCG + 12b VS dual HDR

Two values per pixel: PA (12-bit, compressed from 16-bit combined DCG), PB (12-bit RAW)

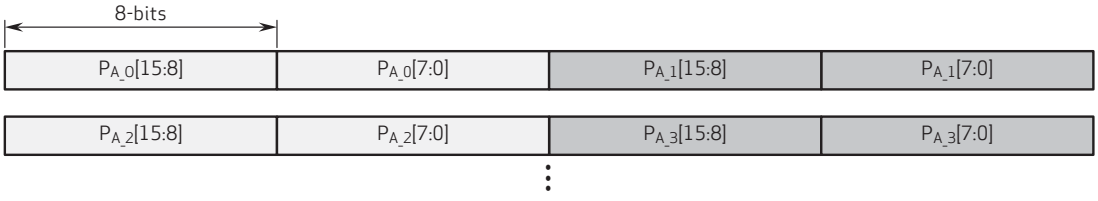
figure 5-12 12b compressed DCG + 12b dual HDR diagram



5.3.1.3 16b DCG single HDR

One value per pixel: PA (16-bit combined DCG)

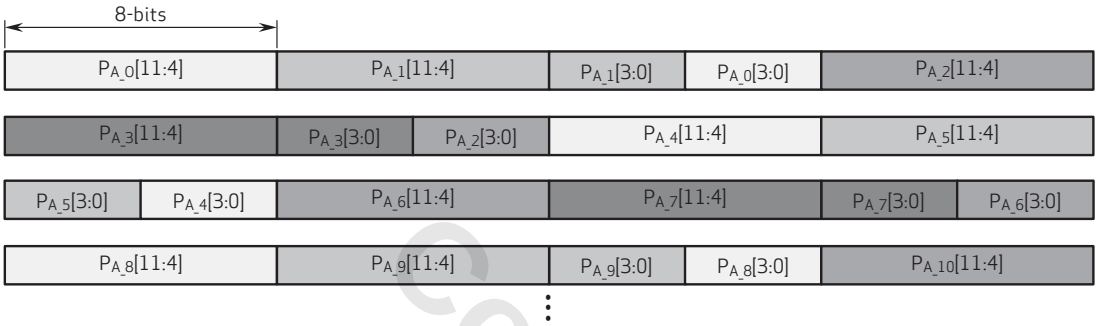
figure 5-13 16b DCG single HDR diagram



5.3.1.4 12b compressed DCG single HDR

One value per pixel: PA (12-bit, compressed from 16-bit combined DCG)

figure 5-14 12b compressed DCG single HDR diagram



If the MIPI speed is changed outside the values provided in the original setting files, the MIPI PCLK period has to be manually (externally) recalculated and set in register 0x4837 for the sensor, according to the formula:

pclk_period_MIPI = 2000/pclk_freq_MHz

For example:

If the MIPI PCLK frequency is set to 162 MHz, pclk_period_MIPI = 2000/162 ≈ 12 = 0x0C to 0x4837

The OS04A10 also supports 3-exposure HDR. Refer to **figure 5-15** to check that the MIPI outputs the 3-exposure image capture with three virtual channels.

figure 5-15 virtual channel for sequential 3-exposure HDR

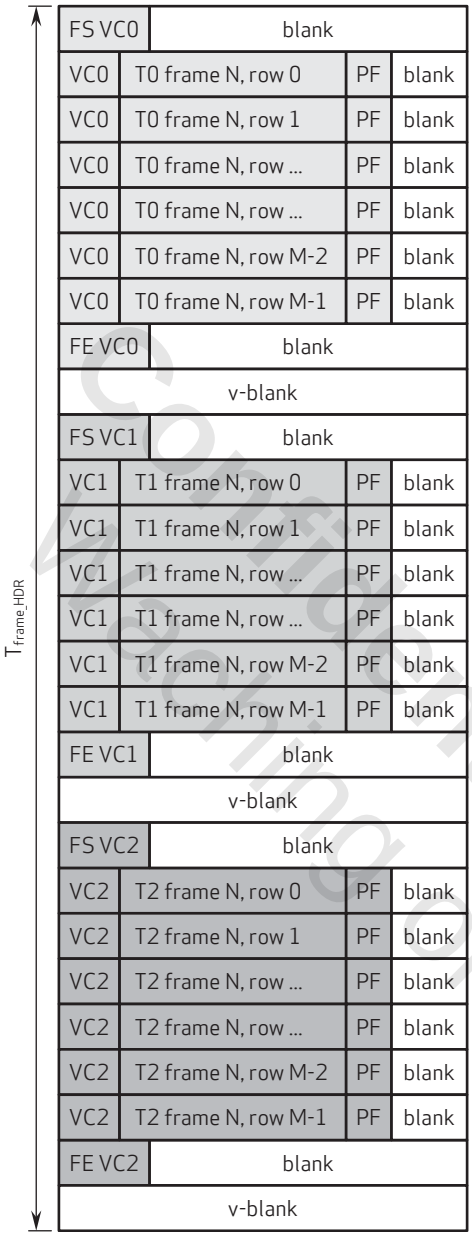


table 5-4 MIPI control registers (sheet 1 of 6)

address	register name	default value	R/W	description
0x4800	MIPI_CORE_0	0x04	RW	Bit[7]: Select sc_valid Bit[6]: Enable vertical clock gating Bit[5]: Enable clock gating Bit[4]: Enable line sync Bit[2]: Invert output PCLK Bit[1]: First bit is 1 Bit[0]: Select manual parameter or auto parameter
0x4802	MIPI_CORE_2	0x00	RW	Bit[7]: Select manual parameter or auto parameter Bit[6]: Select manual parameter or auto parameter Bit[5]: Select manual parameter or auto parameter Bit[4]: Select manual parameter or auto parameter Bit[3]: Select manual parameter or auto parameter Bit[2]: Select manual parameter or auto parameter Bit[1]: Select manual parameter or auto parameter Bit[0]: Select manual parameter or auto parameter
0x4803	MIPI_CORE_3	0x00	RW	Bit[7]: High speed mode only enable Bit[6]: PRBS test enable Bit[5]: prbs_rp_ph_en 1: PRBS pattern replace packet header Bit[4]: FIFO read speed
0x4804	MIPI_CORE_4	0x08	RW	Bit[3:0]: Select height for each virtual channel
0x4805	MIPI_CORE_5	0x00	RW	Bit[3:0]: Retiming
0x4806	MIPI_CORE_6	0x00	RW	Bit[4]: Enable power up mark Bit[3]: Remote reset Bit[2]: Suspend Bit[1]: Enable SMIA lane Bit[0]: Output TX LSB first
0x4807	MIPI_CORE_7	0x03	RW	Bit[3:0]: Ultra low power mode state delay[3:0]
0x4808	MIPI_CORE_8	0x18	RW	Bit[7:0]: Wake-up delay[7:0]
0x480E	MIPI_CORE_14	0x00	RW	Bit[3]: Enable data type2 Bit[2]: Enable img2
0x4810	MIPI_CORE_16	0xFF	RW	Bit[7:0]: Max frame count value[15:8]
0x4811	MIPI_CORE_17	0xFF	RW	Bit[7:0]: Max frame count value[7:0]
0x4813	MIPI_CORE_19	0x00	RW	Bit[7:6]: Virtual channel ID Bit[5:4]: Virtual channel ID Bit[3:2]: Virtual channel ID Bit[1:0]: Virtual channel ID
0x4814	MIPI_CORE_20	0x2A	RW	Bit[6]: Data type select Bit[5:0]: Data type[5:0]
0x4815	MIPI_CORE_21	0x2B	RW	Bit[5:0]: Data type[5:0]

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

address	register name	default value	R/W	description
0x4816	MIPI_CORE_22	0x2B	RW	Bit[5:0]: Data type[5:0]
0x4818	MIPI_CORE_24	0x00	RW	Bit[1:0]: hs_zero_min[9:8] Minimum value of hs_zero, unit ns
0x4819	MIPI_CORE_25	0x70	RW	Bit[7:0]: hs_zero_min[7:0] Minimum value of hs_zero hs_zero_real = hs_zero_min_o + Tui*ui_hs_zero_min_o
0x481A	MIPI_CORE_26	0x00	RW	Bit[1:0]: hs_trail_min[9:8] Minimum value of hs_trail, unit ns
0x481B	MIPI_CORE_27	0x3C	RW	Bit[7:0]: hs_trail_min[7:0] Minimum value of hs_trail hs_trail_real = hs_trail_min_o + Tui*ui_hs_trail_min_o
0x481C	MIPI_CORE_28	0x01	RW	Bit[1:0]: clk_zero_min[9:8] Minimum value of clk_zero, unit ns
0x481D	MIPI_CORE_29	0x2C	RW	Bit[7:0]: clk_zero_min[7:0] Minimum value of clk_zero clk_zero_real = clk_zero_min_o + Tui*ui_clk_zero_min_o
0x481E	MIPI_CORE_30	0x5F	RW	Bit[7:0]: clk_prepare_max[7:0] Maximum value of clk_prepare, unit ns
0x481F	MIPI_CORE_31	0x26	RW	Bit[7:0]: clk_prepare_min[7:0] Minimum value of clk_prepare clk_prepare_real = clk_prepare_min_o + Tui*ui_clk_prepare_min_o
0x4820	MIPI_CORE_32	0x00	RW	Bit[1:0]: clk_post_min[9:8] Minimum value of clk_post, unit ns
0x4821	MIPI_CORE_33	0x3C	RW	Bit[7:0]: clk_post_min[7:0] Minimum value of clk_post clk_post_real = clk_post_min_o + Tui*ui_clk_post_min_o
0x4822	MIPI_CORE_34	0x00	RW	Bit[1:0]: clk_trail_min[9:8] Minimum value of clk_trail, unit ns
0x4823	MIPI_CORE_35	0x3C	RW	Bit[7:0]: clk_trail_min[7:0] Minimum value of clk_trail clk_trail_real = clk_trail_min_o + Tui*ui_clk_trail_min_o
0x4824	MIPI_CORE_36	0x00	RW	Bit[1:0]: lpx_p_min[9:8] Minimum value of lpx_p, unit ns

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

address	register name	default value	R/W	description
0x4825	MIPI_CORE_37	0x32	RW	Bit[7:0]: lpx_p_min[7:0] Minimum value of lpx_p lpx_p_real = lpx_p_min_o + Tui*ui_lpx_p_min_o
0x4826	MIPI_CORE_38	0x32	RW	Bit[7:0]: hs_prepare_min[7:0] Minimum value of hs_prepare, unit ns
0x4827	MIPI_CORE_39	0x55	RW	Bit[7:0]: hs_prepare_max[7:0] Maximum value of hs_prepare hs_prepare_real = hs_prepare_max_o + Tui*ui_hs_prepare_max_o
0x4828	MIPI_CORE_40	0x00	RW	Bit[1:0]: hs_exit_min[9:8] Minimum value of hs_exit, unit ns
0x4829	MIPI_CORE_41	0x64	RW	Bit[7:0]: hs_exit_min[7:0] Minimum value of hs_exit hs_exit_real = hs_exit_min_o + Tui*ui_hs_exit_min_o
0x482A	MIPI_CORE_42	0x06	RW	Bit[5:0]: ui_hs_zero_min[5:0] Minimum UI value of hs_zero, unit UI
0x482B	MIPI_CORE_43	0x04	RW	Bit[5:0]: ui_hs_trail_min[5:0] Minimum UI value of hs_trail, unit UI
0x482C	MIPI_CORE_44	0x00	RW	Bit[5:0]: ui_clk_zero_min[5:0] Minimum UI value of clk_zero, unit UI
0x482D	MIPI_CORE_45	0x00	RW	Bit[7:4]: ui_clk_prepare_max Maximum value of clk_prepare, unit UI Bit[3:0]: ui_clk_prepare_min Minimum value of clk_prepare, unit UI
0x482E	MIPI_CORE_46	0x34	RW	Bit[5:0]: ui_clk_post_min[5:0] Minimum UI value of clk_post, unit UI
0x482F	MIPI_CORE_47	0x00	RW	Bit[5:0]: ui_clk_trail_min[5:0] Minimum UI value of clk_trail, unit UI
0x4830	MIPI_CORE_48	0x00	RW	Bit[5:0]: ui_lpx_p_min[5:0] Minimum UI value of lpx_p (pclk2x domain), unit UI
0x4831	MIPI_CORE_49	0x64	RW	Bit[7:4]: ui_hs_prepare_max Maximum UI value of hs_prepare, unit UI Bit[3:0]: ui_hs_prepare_min Minimum UI value of hs_prepare, unit UI
0x4832	MIPI_CORE_50	0x00	RW	Bit[5:0]: ui_hs_exit_min[5:0] Minimum UI value of hs_exit, unit UI
0x4833	MIPI_CORE_51	0x10	RW	Bit[5:0]: Packet FIFO ready mark[5:0]

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

address	register name	default value	R/W	description
0x4837	MIPI_CORE_55	0x0E	RW	Bit[7:0]: MIPI PCLK period[7:0] Should be changed when PCLK frequency changes
0x4838	MIPI_CORE_56	0x00	RW	Bit[7]: Low power select for lane0 Bit[6]: Low power direction for lane0 Bit[5]: LP p0 for lane0 Bit[4]: LP n0 for lane0 Bit[3]: Low power select for lane1 Bit[2]: Low power direction for lane1 Bit[1]: LP p0 for lane1 Bit[0]: LP n0 for lane1
0x4839	MIPI_CORE_57	0x00	RW	Bit[7]: Low power select for lane2 Bit[6]: Low power direction for lane2 Bit[5]: LP p0 for lane2 Bit[4]: LP n0 for lane2 Bit[3]: Low power select for lane3 Bit[2]: Low power direction for lane3 Bit[1]: LP p0 for lane3 Bit[0]: LP n0 for lane0
0x483C	MIPI_CORE_60	0x10	RW	Bit[7:0]: MIPI parameter value[7:0]
0x483D	MIPI_CORE_61	0x00	RW	Bit[3]: Low power select for clock lane Bit[1]: LP clock lane p Bit[0]: LP clock lane n
0x483E	MIPI_CORE_62	–	R	Bit[7:0]: Frame count value[15:8]
0x483F	MIPI_CORE_63	–	R	Bit[7:0]: Frame count value[7:0]
0x484A	MIPI_CORE_74	0x3F	RW	Bit[5]: Sleep power down select Bit[4]: lp_pon data Bit[3]: lp_pon clock lane data Bit[2]: Manual sleep state Bit[1]: Clock lane sleep state Bit[0]: Data lane sleep state
0x484B	MIPI_CORE_75	0x27	RW	Bit[6]: pa_cal_1time 0: Global timing calculation every frame 1: Global timing calculation only at wake-up Bit[5]: EOF select Bit[4]: Virtual channel select Bit[3]: Data scramble enable Bit[2]: EOF busy select Bit[1]: Clock start select Bit[0]: SOF select
0x484C	MIPI_CORE_76	0x00	RW	Bit[0]: Disable frame count
0x484E	MIPI_CORE_78	0x10	RW	Bit[7:0]: Frame end delay

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

address	register name	default value	R/W	description
0x4850	MIPI_CORE_80	0x40	RW	Bit[6]: DPHY1.2 frame blank select 0: Skew end 1: Frame end Bit[5]: Deskew output enable Bit[4]: Frame deskew disable Bit[3]: Initial deskew disable Bit[2]: De-skew trail data Bit[1]: Perform deskew one lane at a time Bit[0]: DPHY1.2 enable
0x4851	MIPI_CORE_81	0xAA	RW	Bit[7:0]: Deskew data[7:0]
0x4852	MIPI_CORE_82	0xFF	RW	Bit[7:0]: Deskew sync data[7:0]
0x4853	MIPI_CORE_83	0x8A	RW	Bit[7:0]: Deskew start delay[7:0]
0x4854	MIPI_CORE_84	0x08	RW	Bit[7:0]: Frame deskew width[7:0]
0x4855	MIPI_CORE_85	0x30	RW	Bit[7:0]: Initial deskew width[7:0]
0x4856	MIPI_CORE_86	0x01	RW	Bit[7:0]: Frame deskew interval[7:0]
0x4860	MIPI_CORE_87	0x00	RW	Bit[7:6]: mipi_cphy_disable Bit[0]: mipi_cphy_en
0x4861	MIPI_CORE_88	0xA0	RW	Bit[7]: esc_flag_sel2 Bit[6]: esc_flag_sel1 Bit[5]: cphy_trdat Bit[4]: crc_sel
0x4862	MIPI_CORE_89	0x01	RW	Bit[7:0]: t_pre_begin
0x4863	MIPI_CORE_90	0x01	RW	Bit[7:0]: t_pre_end
0x4864	MIPI_CORE_91	0x02	RW	Bit[7:0]: t_prog_seq
0x4865	MIPI_CORE_92	0x66	RW	Bit[7:0]: prog_dat1
0x4866	MIPI_CORE_93	0x99	RW	Bit[7:0]: prog_dat2
0x4867	MIPI_CORE_94	0x88	RW	Bit[7:0]: prog_dat3
0x4868	MIPI_CORE_95	0xAA	RW	Bit[7:0]: prog_dat4
0x4869	MIPI_CORE_96	0xFF	RW	Bit[7:0]: preamb_data1
0x486A	MIPI_CORE_97	0x3F	RW	Bit[7:0]: preamb_data2
0x486B	MIPI_CORE_98	0x84	RW	Bit[7:0]: sync_dat1
0x486C	MIPI_CORE_99	0x36	RW	Bit[7:0]: sync_dat2
0x486D	MIPI_CORE_100	0x00	RW	Bit[7:0]: cphy_rsvdat
0x486E	MIPI_CORE_101	0x84	RW	Bit[7:0]: esc_data1
0x486F	MIPI_CORE_102	0x36	RW	Bit[7:0]: esc_data2

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

address	register name	default value	R/W	description
0x4870	MIPI_CORE_103	0x00	RW	Bit[2]: giic_tinit_man_en Bit[1]: giic_wakeup_man_en Bit[0]: giic_pclk_prd_man_en

5.4 instructions for backend control

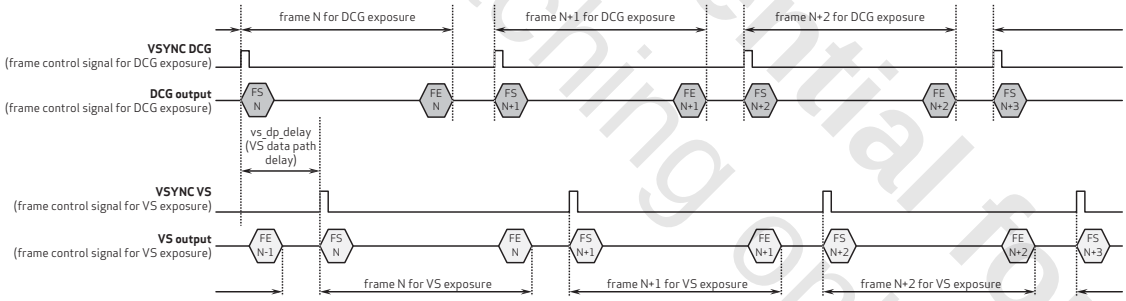
5.4.1 VS data path delay

In staggered mode, VS VSYNC/FS has specific numbers of delayed rows after DCG VSYNC/FS.

vs_dp_delay = integer (TexpVS) + vs_sample_delay

vs_dp_delay is the time delay between DCG VSYNC/FS and VS VSYNC/FS, which is defined by the integer part of the VS exposure time, TexpVS plus vs_sample_delay (a fixed delay between the end of DCG sampling and start of VS precharge state) equal to 2. All values are in units of row time. vs_dp_delay value cannot be read internally and has to be calculated off-sensor (manually) by the application processor.

figure 5-16 sensor frame control signals diagram



The VS data path delay matters only when the sensor is set to use a shared MIPI virtual channel. To output both the DCG and VS exposures on the same channel ID, there is a single FS and FE packet data, but the final image is comprised of line-interleaved image data (rows) of each exposure, toggling one over the other. After the last row of DCG exposure image is sent, a dummy sequence of rows is going to follow (the delay) until only the VS image data remains so it can be transmitted. At this point the user must know vs_dp_delay to define the quantities of dummy data rows. They can be either ignored or accommodated and then taken into account for the final image presentation.

5.5 register writing

5.5.1 suggestion for writing register value after VSYNC or FS

In order to avoid register settings for one frame being split into two frames by mistake, the register value should be written as early as possible after VSYNC or FS. If the register values are written during frame N, new parameters can be seen in frame:

- n+1: AWB gain, BLC offset, etc.
- n+2: exposure, analog gain, digital gain, and conversion gain

5.6 embedded data

Additional information about the set up and configuration of the sensor can be embedded in the video stream. The embedded data contains the values of a programmable list of registers to describe the current state of the sensor (e.g., frame counter, exposure time and gain, etc.). The embedded data is added before (at the front) and/or after (at the end) of the video stream. Depending on the output format used and the type of embedded data, the output (crop and window) size must be increased by two or more rows, to accommodate for the extra lines of data. **figure 5-17** and **figure 5-18** show the structure of the outputted image when embedded (both front and end) data is enabled for non-staggered or staggered HDR mode.

Embedded information can be displayed in the image by setting bits 0x366F[4] and/or 0x366F[5]. It will be read using a specific read bus from the registers. The normal register bus will not be used since it will be occupied.

figure 5-17 image output, non-staggered HDR

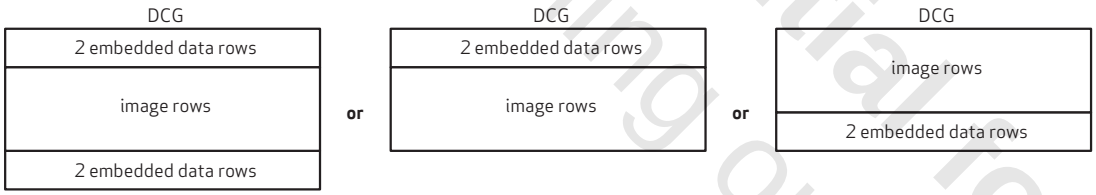
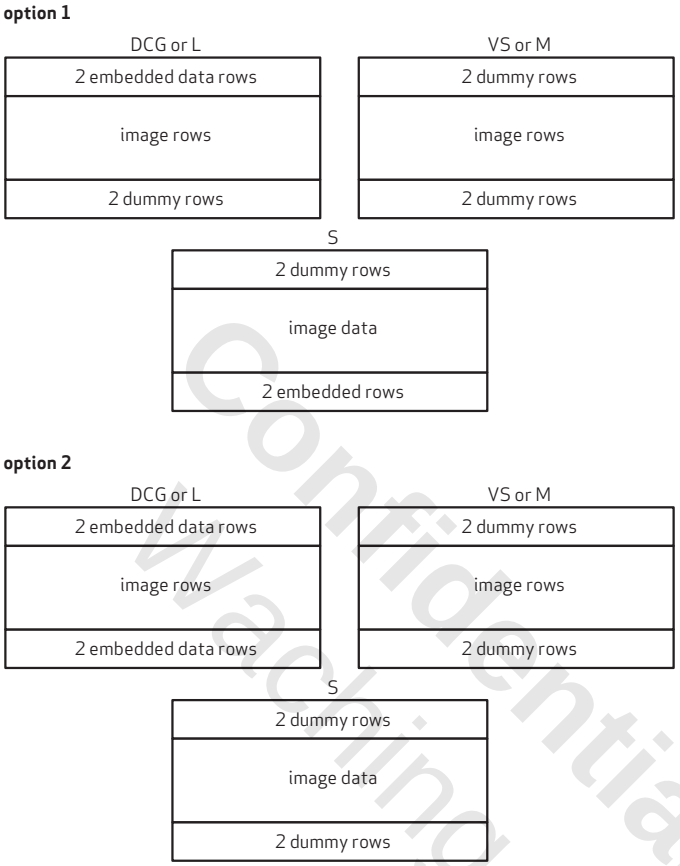


figure 5-18 image output, 2 or 3-staggered HDR (DCG+VS, L+M, or L+M+S)



5.6.1 embedded data format at output – mode 1

In this mode, the 'front' embedded data is added to the first capture (DCG or L) in the beginning of the image. Configured by register 0x3665[7], 'end' embedded data can be added to first capture (DCG or L) or last capture (LCG, S, or VS) at the end of the image. Dummy data (0's) are added to the embedded row for the other blank rows.

The number of registers output in embedded data is configured separately for front and end embedded row by:

- start address
- number of addresses to read

The start address (2 bytes) and the number of addresses (1 byte) will be stored in SRAM. By default, there will be 256 bytes saved for front embedded rows and 256 bytes saved for end embedded rows. Since each address range requires three bytes, 85 different address ranges can be set separately for front and end embedded rows. If additional address ranges are needed, group hold memory allocation can be compromised.

All the data values in the embedded rows are preceded with a tag and two address bytes before each data value. Three padding values (configurable by 0x3207) are inserted after each of these significant bytes due to internal sampling delays.

The embedded data format is the same among the same row positions of the embedded 'front' and 'end'. The content of the rows follows the order below:

|tag|8`hx|8`hx|8`hx|addr_h|8`hx|8`hx|8`hx|addr_l|8`hx|8`hx|8`hx|data|8`hx|8`hx|8`hx|tag|8`hx|8`hx|8`hx|...|8'hx|

where:

- 8'hx = padding data (default = 0x00, configurable)
- tag = tag data (default = 0x55, configurable after initialization)
- addr_h = register address high byte
- addr_l = register address low byte
- data = register content

The total registers output for two front embedded rows are:

((x_output_size/4) + (x_output_size/4))/4

Maximum total registers output for two rows (front embedded data) are (given x_output_ size = 2688): 336.

Maximum total registers output for four rows (front + end embedded data) are (given x_output_ size = 2688): 672.

The following procedure gives an example on how to configure and enable the embedded data functionality for outputting the register data from arbitrary address ranges 0x3800~0x3802 and 0x4800~0x4801 (in the 'front'), 0x5000~0x500F and 0x0304~0x031C (in the 'end'), while utilizing both 'front' and 'end' embedded rows.

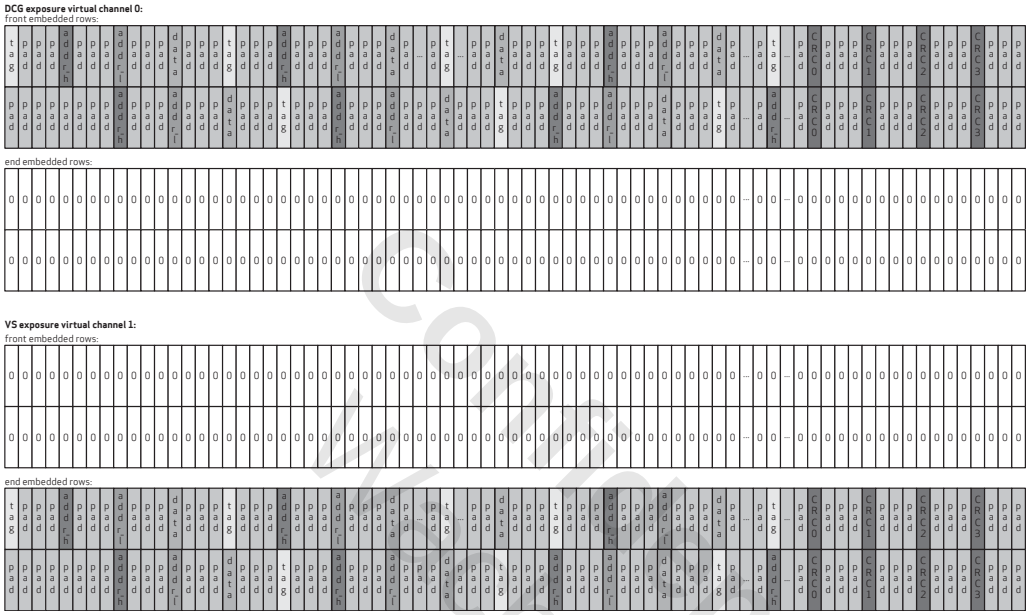
table 5-5 embedded data registers and initialization mode 1

step	registers	comment
1		Embedded rows are configured through group hold functional block. There are two banks (group 4 and group 5). Group 4 is for 'front' embedded row, group 5 is for 'end' embedded row. Set address ranges that are going to be outputted on 'front' embedded rows.
	0x3208 = 0x04	1. Set group 4 to start recording register ranges.
	SCCB_wr (0x3800, 0x03)	2. Execute a SCCB write - configuring three consecutive registers values starting from 0x3800 address to be outputted over embedded row.
	SCCB_wr (0x4800, 0x02)	3. Execute a SCCB write - configuring two consecutive registers values starting from 0x4800 address to be outputted over embedded row.
	0x3208 = 0x14	4. Set group 4 to stop recording register ranges.
		Set address ranges that are going to be outputted on 'end' embedded rows.
	0x3208 = 0x05	5. Set group 5 to start recording register ranges.
	SCCB_wr (0x5000, 0x10)	6. Execute a SCCB write - configuring 16 consecutive registers values starting from 0x5000 address, to be outputted over embedded row.
	SCCB_wr (0x0304, 0x19)	7. Execute a SCCB write - configuring 25 consecutive registers values starting from 0x0304 address, to be outputted over embedded row.
	0x3208 = 0x15	8. Set group 5 to stop recording register ranges.
2	0x3217[a] = 0x00 (default)	9. Select a value to be used for padding (dummy) data.[a]
	0x3219 = 0x55 (default)	10. Select a value to be used for TAGs.
3		After finishing configuration, enable all embedded data lines:
	0x366F[4] = 1	11. Enable 'front' 2 rows of embedded data.
	0x366F[5] = 1	12. Enable 'end' 2 rows of embedded data.
		Streamed image at this point should contain embedded data.
4	0x3218[0] = 1 0x3218[1] = 1 0x3218[2] = 1 0x3218[5] = 1 0x3216[4] = 1	13. Group hold embedded line settings.
5	0x3666 = 0x01	14. Shift embedded data to high 8-bit of 10/12-bit output.
6	0x3669[0] = 0	15. Enable embedded line mode 1.
7	0x3665[7] = 0	16. Set to 0: front embedded data at beginning of first image (L), end embedded data at end of last image (S or VS), see **figure 5-19**.

a. optional

figure 5-19 shows the embedded row composition for a multi-capture mode (3665[7] = 0 in mode 1). If the number of registers is smaller, the remaining space will be replaced by padding bytes (0x3217 value).

figure 5-19 embedded data format for mode 1 in stagger HDR

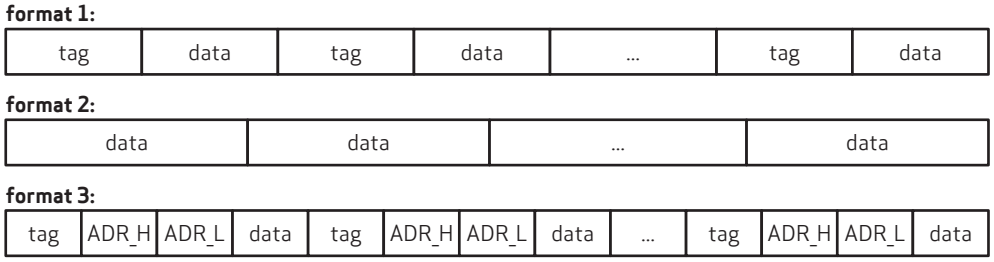


Depending on the image resolution, most importantly the horizontal size, the embedded data adjacent to each section of the image, is going to be distributed along each of the two rows in a continuous/sequential way. It will keep the same order as it has been written in the group hold, filling the first rows first, then spilling over to the second one, if there is more data to accommodate for. It is not possible to specify which register ranges allotted to either the 'front' or 'end' embedded rows, should occupy the first or second rows of the same.

5.6.2 embedded data format at output – mode 2

There is another mode for embedded rows which can be enabled by setting register 0x3669[0] = 1. In this mode, set register 0x3665[7] = 1, all embedded rows are from the long exposure image. This mode has three data formats as shown in **figure 5-20**.

figure 5-20 three data formats



tag: 8-bit, set by register 0x3674[7:0], default is 8'h55

data: effective embedded data

ADR_H: high byte of the address of the following embedded data

ADR_L: low byte of the address of the following embedded data

The three data formats modes are selected by register 0x3671[7:6], 2'b00 for format 1, 2 'b10 for format 2, and 2'b01 for format 3.

table 5-6 embedded data registers and initialization mode 2

step	registers	comment
1		Embedded rows are configured through group hold functional block. There are two banks (group 4 and group 5). Group 4 is for 'front' embedded row, group 5 is for 'end' embedded row. Set address ranges that are going to be outputted on 'front' embedded rows.
	0x3208 = 0x04	1. Set group 4 to start recording register ranges.
	SCCB_wr (0x3800, 0x03)	2. Execute a SCCB write - configuring three consecutive registers values starting from 0x3800 address to be outputted over embedded row.
	SCCB_wr (0x4800, 0x02)	3. Execute a SCCB write - configuring two consecutive registers values starting from 0x4800 address to be outputted over embedded row.
	0x3208 = 0x14	4. Set group 4 to stop recording register ranges.
		Set address ranges that are going to be outputted on 'end' embedded rows.
	0x3208 = 0x05	5. Set group 5 to start recording register ranges.
	SCCB_wr (0x5000, 0x10)	6. Execute a SCCB write - configuring 16 consecutive registers values starting from 0x5000 address, to be outputted over embedded row.
	SCCB_wr (0x0304, 0x19)	7. Execute a SCCB write - configuring 25 consecutive registers values starting from 0x0304 address, to be outputted over embedded row.
	0x3208 = 0x15	8. Set group 5 to stop recording register ranges.
2	0x3217[a] = 0x00 (default) 0x3674 = 0x55	9. Select a value to be used for padding (dummy) data.[a] 10. Select tag value (by 0x3674 in mode 2) to be used in embedded line.
3	0x366F[4] = 1 0x366F[5] = 1	After finishing configuration, enable all embedded data lines: 11. Enable 'front' 2 rows of embedded data. 12. Enable 'end' 2 rows of embedded data. Streamed image at this point should contain embedded data.
4	0x3218[0] = 1 0x3218[1] = 1 0x3218[2] = 1 0x3218[5] = 1	13. Group hold embedded line settings.
5	0x3666 = 0x01	14. Shift embedded data to high-8 bit of 10/12-bit output.
6	0x3669[0] = 1	15. Enable mode 2.
7	0x3665[7] = 8	16. Set to 1: front and end embedded data at first image (L), mode 2 must be set to 1

a. optional

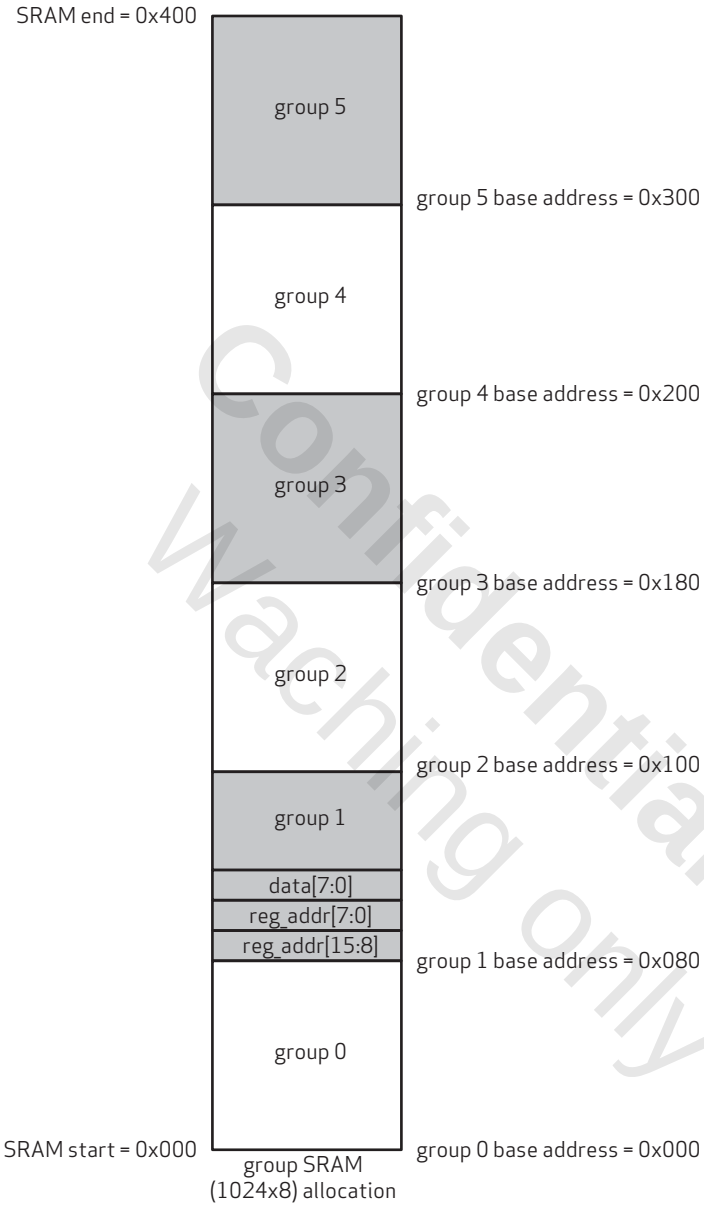
5.7 group hold

The OS04A10 supports a group hold function, which is used to manipulate registers programmed into the sensor. Multiple sets of register settings, which are called groups, can be stored into different virtual memory spaces (hold operation) during sensor initiation and programmed into individual registers later on when desired (launch operation).

The idea is to allow for better synchronization and frame alignment of setting changes between frames to ensure that a certain set of register changes occurs in a single time window, in a single frame or over a number of frames, depending on the way the group hold had been set. Overall, the set of registers are guaranteed to be written prior to the internal latch at the frame boundary. In summation, the group hold functionality allows configuring of many of the sensor's parameters in a single instance, something that cannot be applied or guaranteed when using direct SCCB register writes.

The OS04A10 supports up to six groups (0 to 5) and they all share 1024 bytes of memory (SRAM) used to store register settings. The maximum size of each group is programmable by adjusting the start address amongst the groups (see **figure 5-21**). By default, groups 0 to 3 are assigned 128 bytes each. Groups 4 and 5 are assigned 256 bytes each. Regarding groups 0 to 3, for each register in the setting, three bytes of SRAM space are required for storage of a setting, the register high byte address, the register low byte address, and the desired byte value to be written. So, utilizing the first four groups, a total of 170 (768 bytes/3) register values can be changed. Groups 4 and 5 have somewhat different behaviors and structures as they are tied to one specific function (see **section 5.6**). They cannot be reconfigured to perform group hold operations as in the other groups' (ordinary register settings write) case.

figure 5-21 groups default SRAM allocation



5.7.1 group configuration

The starting address of each group (group base address) is programmed by writing to registers addresses 0x3200~0x3205. The value writing into these registers is the physical SRAM address divided by 16. For example, for the SRAM configuration of groups 0 to 3, registers 0x3200~0x3203, can be set to 0x00, 0x10, 0x28, and 0x38, respectively.

By properly setting the group base address registers previously mentioned, the SRAM can be configured into 1~4 groups. For example, if it is desired to use group 1 and group 3, set register 0x3201 to 0x00 and register 0x3203 to 0x38. The values of registers 0x3200 and 0x3202 do not matter as long as the hold operations are not performed on these two groups.

5.7.2 group hold

After the groups are configured properly, users can perform the hold operation to store register settings into the SRAM of each group. The hold of each group starts and ends with the group access control register 0x3208. The lower four bits of 0x3208 control which group to access. The upper 4 bits control the start (0x0: hold start) and end (0x1: hold end) of the hold operation.

The example setting below shows the sequence to hold group 0 and group 1:

```
6C 3208 00; group 0 hold start

6C 3800 11; first register into group 0

6C 3911 22; second register into group 0

6C 3208 10; group 0 hold end


6C 3208 01; group 1 hold start

6C 3810 11; first register into group 1

6C 3921 22; second register into group 1

6C 3931 33; third register into group 1

6C 3208 10; group 1 hold end
```

Group 0 holds two registers and group 1 holds three registers in the example shown above. Keep in mind not to hold more than the maximum register numbers for each group as defined in the group configuration step.

5.7.3 group launch

After the contents of each group are defined in the hold operation, all registers belonging to each group are stored in the SRAM and ready to be written into the target registers (i.e., launch of that group). There are five launch modes as described in the sections below, using the group hold setup from the previous section.

5.7.3.1 launch mode 1 – quick manual launch

Manual launch is enabled by setting the group switch control register 0x320D to 0.

Quick manual launch is achieved by writing to control register 0x3208. The value written into this register is 0xEX, the upper 4 bits (0xE) are the quick launch command and the lower four bits (0xX) are the group number. For example, if it is desired to launch group 0, write the value 0xE0 to 0x3208. Then, the contents of group 0 will be

written to the target registers immediately after the sensor receives this command through the SCCB. The following is a setting example:

```
6C 320D 00; manual launch on

6C 3208 E0; quick launch group 0

6C 3208 E1; quick launch group 1
```

5.7.3.2 launch mode 2 – delay manual launch

Delay manual launch is achieved by writing to register 0x3208. The value written into this register is 0xAX, where the upper four bits (0xA) are the delay launch command and the lower four bits (0xX) are the group number. For example, if it is desired to launch group 1, write the value 0xA1 to 0x3208. Then the contents of group 1 will be written to the target registers. The difference with mode 1 is that the writing will wait for some internally defined time spot in vertical blanking, thus delayed until the current frame sampling ends. The following is a setting example:

```
6C 320D 00; manual launch on

6C 3208 A0; delay launch group 0

6C 3208 A1; delay launch group 1
```

5.7.3.3 launch mode 3 – quick auto launch

Quick auto launch works like mode 1, but it is different in that it will return to a specified group automatically. This is controlled by the register 0x320D, where bit[2] enables mode and bit[1:0] controls which group to return. The auto launch will automatically switch from group 0 to 1 to 2 to 3. If group auto launch starts from the low number group, returns to high number group, and it is desired to skip the middle group, set that group number to 0. If it is not set to 0, the sensor will switch to the middle group first. Also, the user will need to set a target return group number to be non- zero. Frame numbers 0x3209, 0x320A, 0x320B, and 0x320C for groups 0,1, 2, and 3, respectively, control how many frames to stay before returning.

Register bits 0x320D[7:4] control how many loops of this context switch launch sequence (e.g., grp0 to grp1 to grp2 is a context switch launch sequence; set bit[7:4] to N, sequence will be repeated by N+1 times), register bit 0x320D[2] controls context switch enable, and register bit 0x320D[3] controls repeat mode enable. For auto launch, set register bit 0x320D[3] to 1'b0 and set register bit 0x320D[2] to 1'b1.

The context switch command register address is 0x320E (e.g., 6C 320E E0). It mainly supports combination of context switch launch and manual launch. For hold operation, use 0x3208 only.

The quick auto launch can be better understood with the following setting example:

```
6C 3209 04; group 0 stays 4 frames

6C 320A 01; group 1 stays 1 frame

6C 320D 05; [1:0]: 1, return to group 1; [2]: 1, context switch on; [7:4]: 0, loop
number is 1

6C 320E E0; quick launch group 0
```

5.7.3.4 launch mode 4 – delay auto launch

Delay auto launch works like mode 2 in the delay launch part and like mode 3 in the return part.

The operation can be better understood with a setting example:

```
6C 3209 04; group 0 stays 4 frames
6C 320A 01; group 1 stays 1 frame
6C 320D 05; [1:0]: 1, return to group 1; [2]: 1, context switch on; [7:4]: 0, loop number is 1
6C 320E A0; delay launch group 0
```

In this example, the sensor will delay launch group 0, stay at group 0 for four frames, and then return to group 1.

5.7.3.5 launch mode 5 – repeat launch

Repeat launch is similar to auto launch, with the only difference being that the context switch launch sequence is repeated by infinite loops. Register 0x320D should be configured differently and the configuration is much simpler; set bit[3] to 1'b1 and set bit[2] to 1'b1, there is no need to configure return group and loop number.

The operation can be better understood with a setting example:

```
6C 3209 04; group 0 stays 4 frames
6C 320A 04; group 1 stays 4 frames
6C 320D 0C; [3]: 1, repeat mode on, [2]: 1, context switch on
6C 320E A0; delay launch group 0
```

In this example, the sensor will repeatedly launch group 0 and group 1. Each group stays for 4 frames. Note that the first launch of group 0 is delay launch.

In manual mode, a register write burst can be triggered with a SCCB write and the specified group's register settings will be written to the register interface. In automatic mode, two groups are selected and the number of frames each group should be active. The sensor will continuously update the register settings accordingly.

It can be noted that if a certain group hold number is desired to be disabled, the adjacent register for 'stay for a number of frames' for that group number (0x3209~ 0x320C), can be set to 0. Eliminating the possibility for that group to execute.

Additionally, the OS04A10 supports group record and group launch, not only while in stream mode, but also while in sleep mode, in order to make the first frame with the updated configuration when the sensor is put into streaming. This is especially emphasized for exposure and gain (analog and digital) changes.

table 5-7 group hold control registers (sheet 1 of 3)

address	register name	default value	R/W	description
0x3200	GRP_HOLD_0	0x00	RW	Bit[7:0]: Group 0 base address[7:0]
0x3201	GRP_HOLD_1	0x10	RW	Bit[7:0]: Group 1 base address[7:0]
0x3202	GRP_HOLD_2	0x20	RW	Bit[7:0]: Group 2 base address[7:0]
0x3203	GRP_HOLD_3	0x30	RW	Bit[7:0]: Group 3 base address[7:0]
0x3204	GRP_HOLD_4	0x40	RW	Bit[7:0]: Group 4 base address[7:0]
0x3205	GRP_HOLD_5	0x50	RW	Bit[7:0]: Group 5 base address[7:0]
0x3206	GRP_HOLD_6	0x00	RW	Bit[7]: re_launch_dis Bit[5]: first_frame_context_en Bit[4]: FSIN enable in context switch Bit[1:0]: FSIN group number
0x3208	GROUP_ACCESS	–	W	Bit[7:4]: group_ctrl 0000: Group hold start 0001: Group hold end 0110: Manual launch at hblk 1010: Manual launch in frame blanking 1110: Manual launch directly Bit[3:0]: group_id 0000: Group bank 0, default start from address 11'h000 0001: Group bank 1, default start from address 11'h000 0010: Group bank 2, default start from address 11'h200 0011: Group bank 3, default start from address 11'h300
0x3209	GRP_HOLD_9	0x00	RW	Bit[7:0]: Number of frames staying in group 0
0x320A	GRP_HOLD_10	0x00	RW	Bit[7:0]: Number of frames staying in group 1
0x320B	GRP_HOLD_11	0x00	RW	Bit[7:0]: Number of frames staying in group 2
0x320C	GRP_HOLD_12	0x00	RW	Bit[7:0]: Number of frames staying in group 3
0x320D	GRP_SWCTRL	0x01	RW	Bit[7:4]: Auto launch loop number Bit[3]: Repeat switch mode Bit[2]: Context switch enable Bit[1:0]: Switch back group

table 5-7 group hold control registers (sheet 2 of 3)

address	register name	default value	R/W	description
0x320E	CONTEXT_SW GROUP_ACCESS	–	W	Bit[7:4]: Auto/repeat launch control 0110: Auto/repeat launch at HBLK 1010: Auto/repeat launch in frame blanking 1110: Auto/repeat launch directly Bit[3:0]: group_id 0000: Group bank 0, default start from address 11'h000 0001: Group bank 1, default start from address 11'h000 0010: Group bank 2, default start from address 11'h200 0011: Group bank 3, default start from address 11'h300
0x3210	GRP_HOLD_16	–	R	Bit[7:0]: Group 0 length
0x3211	GRP_HOLD_17	–	R	Bit[7:0]: Group 1 length
0x3212	GRP_HOLD_18	–	R	Bit[7:0]: Group 2 length
0x3213	GRP_HOLD_19	–	R	Bit[7:0]: Group 3 length
0x3214	GRP_HOLD_20	–	R	Bit[7:0]: Group 4 length
0x3215	GRP_HOLD_21	–	R	Bit[7:0]: Group 5 length
0x3216	GRP_HOLD_22	0x01	RW	Bit[4]: emb_href_sel Bit[3:0]: Number of embedded rows
0x3217	GRP_HOLD_23	0x00	RW	Bit[7:0]: Padding data
0x3218	GRP_HOLD_24	0x00	RW	Bit[7:6]: hblk_ctrl Bit[5]: addr_en Bit[4]: padding_md2 Bit[3]: frame_cnt_trig Bit[2]: emline_sof_en Bit[1]: emline_sof_en Bit[0]: tag_en
0x3219	GRP_HOLD_25	0x55	RW	Bit[7:0]: Tag data
0x321A	GRP_HOLD_26	–	R	Bit[7:0]: Selected group
0x321C	GRP_HOLD_28	–	R	Bit[7:0]: Frame counter group 0
0x321D	GRP_HOLD_29	–	R	Bit[7:0]: Frame counter group 1
0x321E	GRP_HOLD_30	–	R	Bit[7:0]: Frame counter group 2
0x321F	GRP_HOLD_31	–	R	Bit[7:0]: Frame counter group 3
0x3220	GRP_HOLD_32	0x00	RW	Bit[5:0]: Tst[13:8]
0x3221	GRP_HOLD_33	0x00	RW	Bit[7:0]: Tst[7:0]

table 5-7 group hold control registers (sheet 3 of 3)

address	register name	default value	R/W	description
0x3222	GRP_HOLD_34	0x02	RW	Bit[7:0]: gain_ctrl
0x3223	GRP_HOLD_35	0x58	RW	Bit[7:0]: Group 6 base address[7:0]
0x3224	GRP_HOLD_36	0x5A	RW	Bit[7:0]: Group 7 base address[7:0]
0x3225	GRP_HOLD_37	0x5C	RW	Bit[7:0]: Group 8 base address[7:0]
0x3226	GRP_HOLD_38	0x5E	RW	Bit[7:0]: Group 9 base address[7:0]
0x3227	GRP_HOLD_39	–	R	Bit[7:0]: Group 6 length
0x3228	GRP_HOLD_40	–	R	Bit[7:0]: Group 7 length
0x3229	GRP_HOLD_41	–	R	Bit[7:0]: Group 8 length
0x322A	GRP_HOLD_42	–	R	Bit[7:0]: Group 9 length

5.8 LVDS specification

5.8.1 features

- parameterized
- supports 2/4-lane mode
- supports 10/12/16-bit mode
- supports only one sync code split in every line or four byte code every line
- SAV or EAV first
- LVDS VC mode

5.8.2 LVDS registers

table 5-8 LVDS register

address	register name	default value	R/W	description
0x4640	LVDS R0	0x2A	RW	Bit[7]: Two sync code enable in 8-lane mode Bit[6]: Sync code manual mode enable Bit[5]: Sync code control Bit[4]: lvds_pclk_inv Bit[3]: Channel ID enable in sync per lane mode Bit[2]: CCIR parameter Bit[1]: SAV first enable Bit[0]: Sync code mode 0: Split 1: Per lane

5.8.3 output mode

Sync mode 1 or 2 can be set by register 0x4640[0].

5.8.3.1 2-lane modes

1 sync code mode:

```
Lane0: FFF 000  P0 P2 ... Pn-1 FFF 000  ... FFF 000  P0 P2 ... Pn-1 FFF 000  ...
Lane1: 000 SAV0 P1 P3 ... Pn   000 EAV0 ... 000 SAV0 P1 P3 ... Pn   000 EAV0 ...
```

2 sync code mode (lane0 and lane1 sync code are same):

```
Lane0: FFF 000 000 SAV0 P0 P2 ... Pn-1 FFF 000 000 EAV0 ... FFF 000 000 SAV0 P0 P2 ...
       Pn-1 FFF 000 000 EAV0 ...
Lane1: FFF 000 000 SAV0 P1 P3 ... Pn   FFF 000 000 EAV0 ... FFF 000 000 SAV0 P1 P3 ...
       Pn   FFF 000 000 EAV0 ...
```

5.8.3.2 4-lane modes

1 sync code mode:

```
Lane0: FFF P0 P4 ... Pn-3 FFF  ... FFF  P0 P4 ... Pn-3 FFF  ...
Lane1: 000 P1 P5 ... Pn-2 000  ... 000  P1 P5 ... Pn-2 000  ...
Lane2: 000 P2 P6 ... Pn-1 000  ... 000  P2 P6 ... Pn-1 000  ...
Lane3: SAV0 P3 P7 ... Pn  EAV0 ... SAV0 P3 P7 ... Pn   EAV0 ...
```

4 sync code mode:

```
Lane0: FFF 000 000 SAV0 P0 P4 ... Pn-3 FFF 000 000 EAV0 ... FFF 000 000 SAV0 P0 P4 ...
       Pn-3 FFF 000 000 EAV0 ...
Lane1: FFF 000 000 SAV0 P1 P5 ... Pn-2 FFF 000 000 EAV0 ... FFF 000 000 SAV0 P1 P5 ...
       Pn-2 FFF 000 000 EAV0 ...
Lane2: FFF 000 000 SAV0 P2 P6 ... Pn-1 FFF 000 000 EAV0 ... FFF 000 000 SAV0 P2 P6 ...
       Pn-1 FFF 000 000 EAV0 ...
Lane3: FFF 000 000 SAV0 P3 P7 ... Pn   FFF 000 000 EAV0 ... FFF 000 000 SAV0 P3 P7 ...
       Pn   FFF 000 000 EAV0 ...
```

5.8.4 EAV or SAV first

EAV or SAV first can be controlled by register 0x4640[1]. No matter whether EAV or SAV is first, the OS04A10 has two blanking lines at the end of each frame.

5.8.4.1 EAV first

figure 5-22 frame timing relationship between SAV and EAV when EAV is first

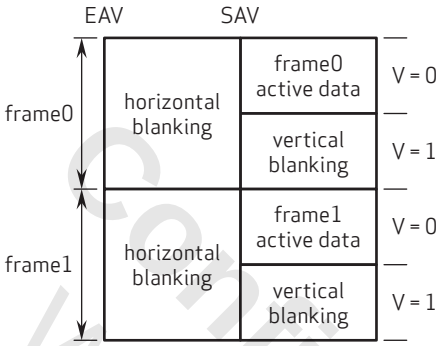
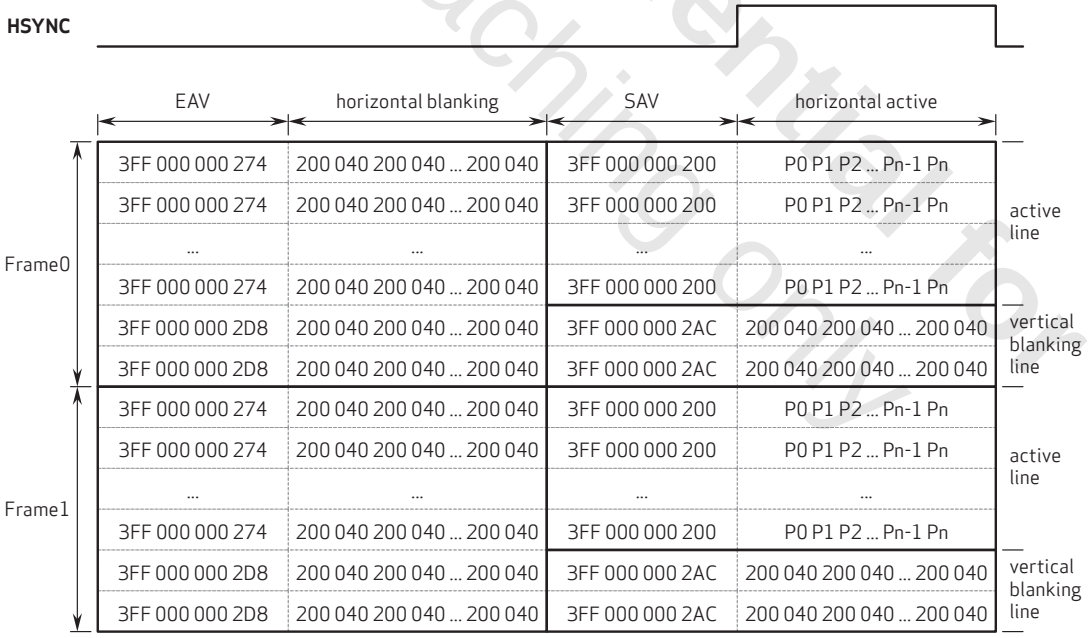


figure 5-23 10-bit data and sync code description for EAV first



The blanking data 200/040 can be adjusted using registers {0x4642, 0x4643}/{0x4644, 0x4645}.

5.8.4.2 SAV first

figure 5-24 frame timing relationship between SAV and EAV when SAV is first

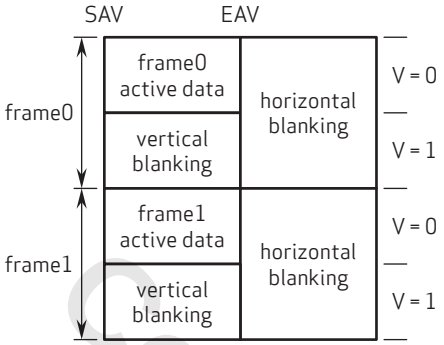
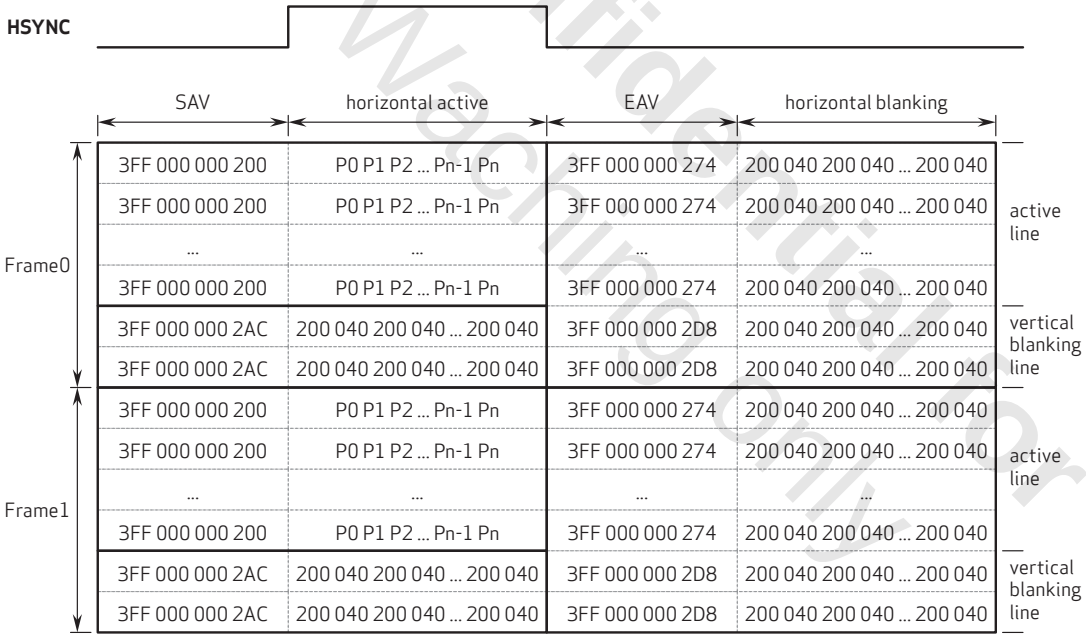


figure 5-25 10-bit data and sync code description for SAV first



The blanking data 200/040 can be adjusted using registers {0x4642, 0x4643}/{0x4644, 0x4645}.

5.8.5 LVDS VC mode

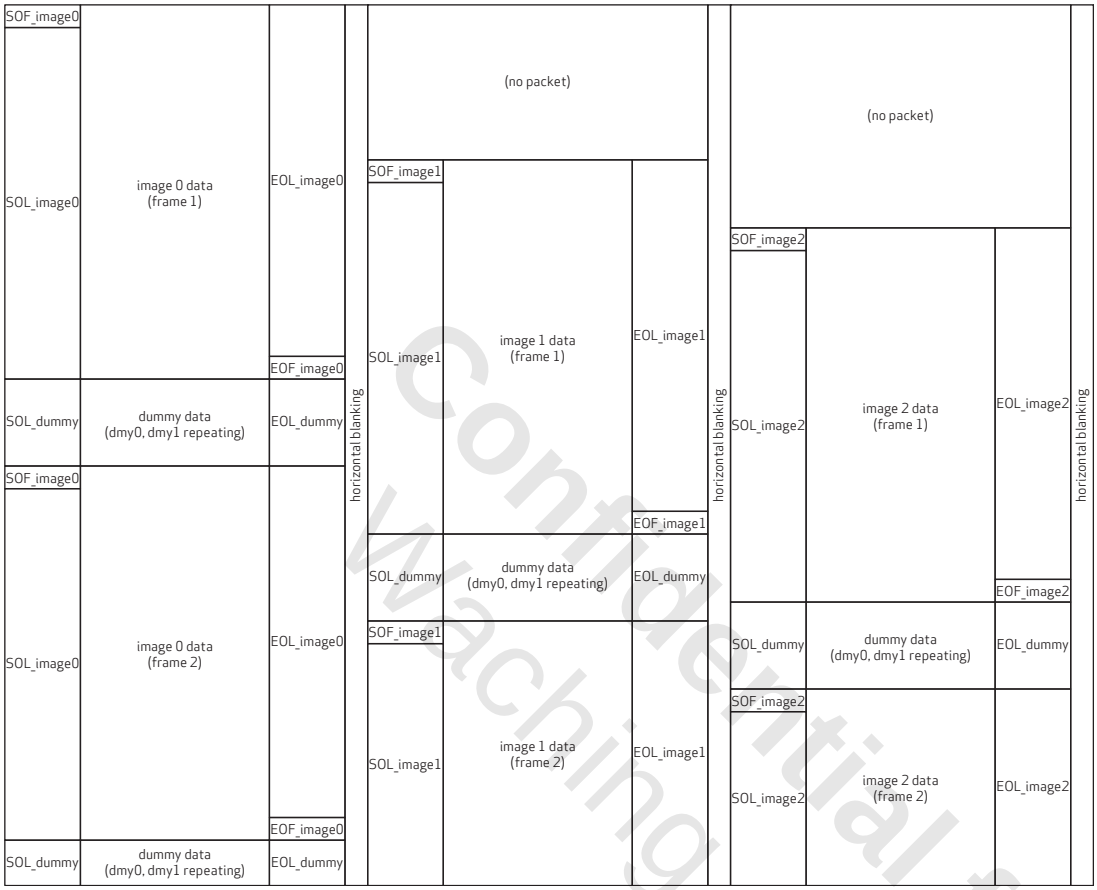
LVDS VC mode is achieved by sending different SAV/EAV sync codes for different exposures. Sync code manual is used. The sync codes are set by registers 0x4646~0x4649 (for long exposure), registers 0x4686~0x4689 (for medium exposure), and registers 0x46C6~0x46C9 (for short exposure).

Blanking line sync code can also be set separately for different virtual channels by using registers 0x4654~0x4655 (for long exposure), registers 0x4694~4695 (for medium exposure), and registers 0x46D4~0x46D5 (for short exposure).

table 5-9 LVDS VC mode registers

address	register name	default value	R/W	description
long exposure				
0x4646	R6	0xAA	RW	Bit[7:0]: frame_st[7:0]
0x4647	R7	0x55	RW	Bit[7:0]: frame_ed[7:0]
0x4648	R8	0x99	RW	Bit[7:0]: line_st[7:0]
0x4649	R9	0x66	RW	Bit[7:0]: line_ed[7:0]
0x4654	R14	0x11	RW	Bit[7:0]: blk_line_st[7:0]
0x4655	R15	0x22	RW	Bit[7:0]: blk_line_ed[7:0]
medium exposure				
0x4646	R6	0xAA	RW	Bit[7:0]: frame_st[7:0]
0x4647	R7	0x55	RW	Bit[7:0]: frame_ed[7:0]
0x4648	R8	0x99	RW	Bit[7:0]: line_st[7:0]
0x4649	R9	0x66	RW	Bit[7:0]: line_ed[7:0]
0x4694	R14	0x11	RW	Bit[7:0]: blk_line_st[7:0]
0x4695	R15	0x22	RW	Bit[7:0]: blk_line_ed[7:0]
short exposure				
0x4646	R6	0xAA	RW	Bit[7:0]: frame_st[7:0]
0x4647	R7	0x55	RW	Bit[7:0]: frame_ed[7:0]
0x4648	R8	0x99	RW	Bit[7:0]: line_st[7:0]
0x4649	R9	0x66	RW	Bit[7:0]: line_ed[7:0]
0x46D4	R14	0x11	RW	Bit[7:0]: blk_line_st[7:0]
0x46D5	R15	0x22	RW	Bit[7:0]: blk_line_ed[7:0]

figure 5-26 LVDS VC mode structure



5.8.6 LVDS 16-bit mode

5.8.6.1 sync per lane format

```
Lane0: FF FF 00 00 00 00 sav_h sav_l P0_h ... Pn-2_h FF FF 00 00 00 00 eav_h eav_l
Lane1: FF FF 00 00 00 00 sav_h sav_l P0_l ... Pn-2_l FF FF 00 00 00 00 eav_h eav_l
Lane2: FF FF 00 00 00 00 sav_h sav_l P1_h ... Pn-1_h FF FF 00 00 00 00 eav_h eav_l
Lane3: FF FF 00 00 00 00 sav_h sav_l P1_l ... Pn-1_l FF FF 00 00 00 00 eav_h eav_l
```

5.8.6.2 sync code split format

```
Lane0: FF 00    P0_h ... Pn-2_h FF 00
Lane1: FF 00    P0_l ... Pn-2_l FF 00
Lane2: 00 sav_h P1_h ... Pn-1_h 00 eav_h
Lane3: 00 sav_l P1_l ... Pn-1_l 00 eav_l
```

5.8.7 PHY specification diagram

figure 5-27 PHY specification diagram

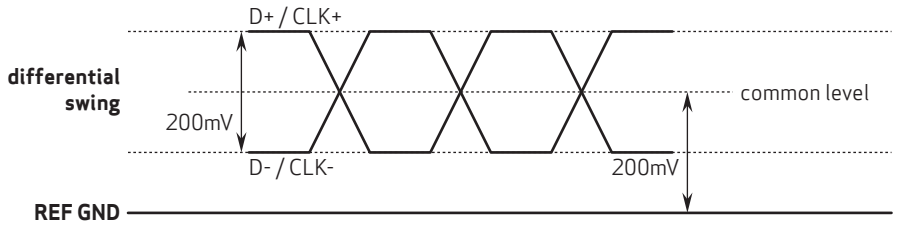


table 5-10 PHY specification table

parameter	min	typ	max	unit
differential output	150	200	250	mV
common level output	150	200	250	mV
rise/fall time	150		0.3UI	ps
data clock skew	-0.15UI		0.15UI	ps
impedance	40	50	60	Ω

6 SCCB interface

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

The SCCB ID is controlled by the SID pin. If SID is low, the sensor's SCCB ID is 0x6C. If SID is high, the sensor's SCCB ID is 0x20. The SCCB ID can also be programmed by registers. When 0x303F[0] is 1, the ID comes from register 0x3035 when SID = 0 and register 0x3037 when SID = 1, which allows SCCB ID to be modified to any valid address by a register write after power on.

6.1 SCCB timing

The maximum allowed capacitive loading on the SCCB bus lines is 25 pF.

figure 6-1 SCCB interface timing

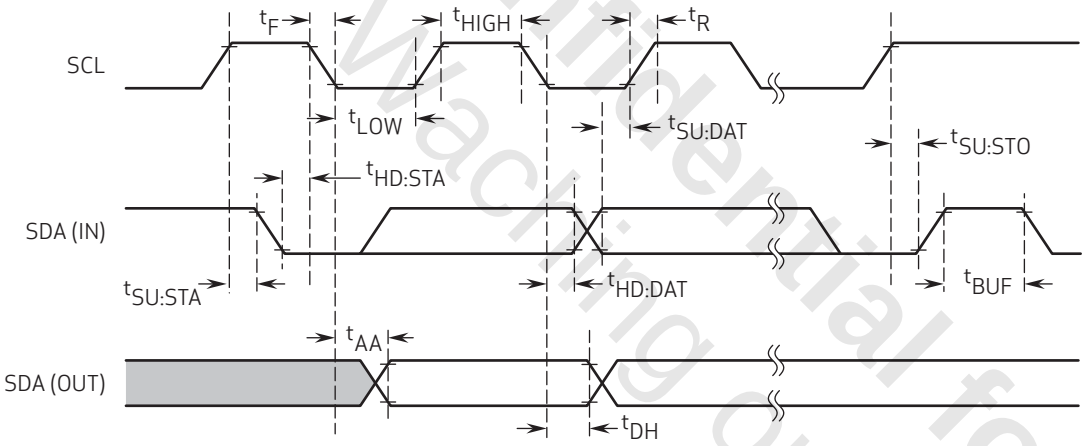


table 6-1 SCCB interface timing specifications[ab]

symbol	parameter	min	typ	max	unit
f_{SCL}	clock frequency		400	1000[c]	kHz
t_{LOW}	clock low period	1.3			µs
t_{HIGH}	clock high period	0.6			µs
t_{AA}	SCL low to data out valid	0.1		0.9	µs
t_{BUF}	bus free time before new start	1.3			µs
$t_{HD:STA}$	start condition hold time	0.6			µs

table 6-1 SCCB interface timing specifications[ab]

symbol	parameter	min	typ	max	unit
$t_{SU:STA}$	start condition setup time	0.6			µs
$t_{HD:DAT}$	data in hold time	0			µs
$t_{SU:DAT}$	data in setup time	0.1			µs
$t_{SU:STO}$	stop condition setup time	0.6			µs
t_R, t_F	SCCB rise/fall times			0.3	µs
t_{DH}	data out hold time	0.05			µs

a. SCCB timing is based on 400 kHz mode

b. timing measurement shown at beginning of rising edge or end of falling edge signifies 30%,
timing measurement shown in middle of rising/falling edge signifies 50%,
timing measurement shown at end of rising edge or beginning of falling edge signifies 70%

c. for 1000 kHz mode, minimum input clock is 10 MHz; for 400 kHz or less, minimum input clock is 6 MHz

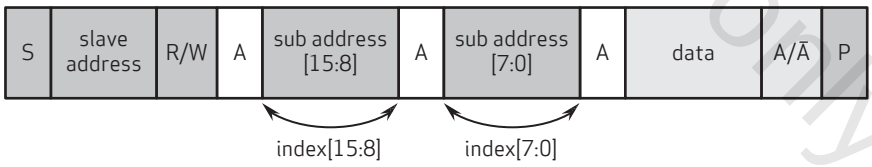
6.2 direct access mode

6.2.1 message format

The OS04A10 supports the message format shown in **figure 6-2**. The repeated START (Sr) condition is shown in **figure 6-3** and **figure 6-5**.

figure 6-2 message type

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



from slave to master — S START condition — A acknowledge

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

direction depends on operation — Sr repeated START condition

6.2.2 read / write operation

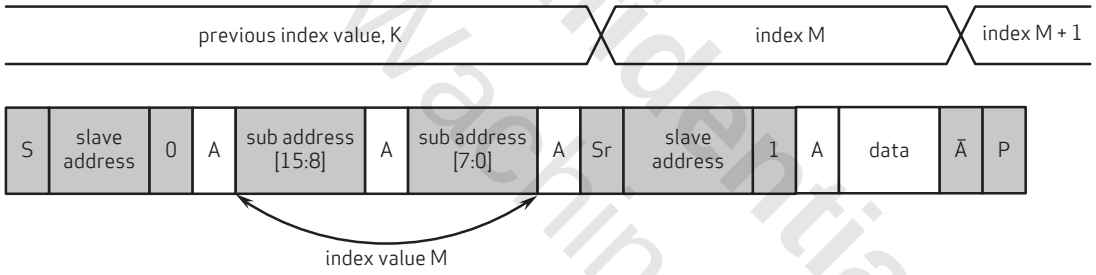
The OS04A10 supports four different read operations and two different write operations:

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

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

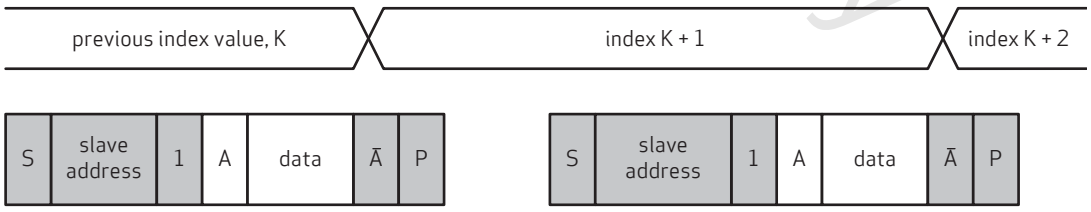
In a single read from random locations, the master does a dummy write operation to desired sub-address, issues a repeated start condition and then addresses the camera again with a read operation. After acknowledging its slave address, the camera starts to output data onto the SDA line as shown in **figure 6-3**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 6-3 SCCB single read from random location



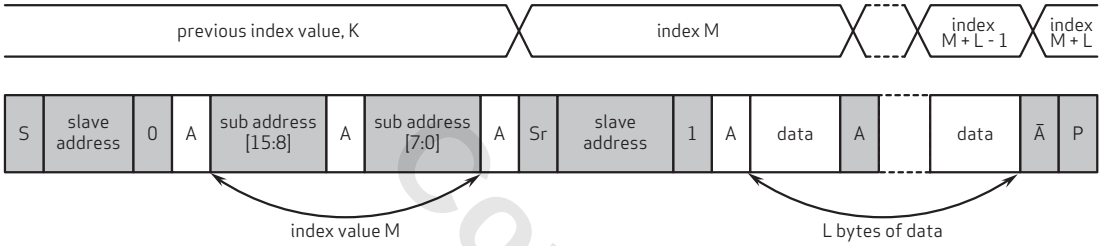
If the host addresses the camera with read operation directly without the dummy write operation, the camera responds by setting the data from last used sub-address to the SDA line as shown in **figure 6-4**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 6-4 SCCB single read from current location



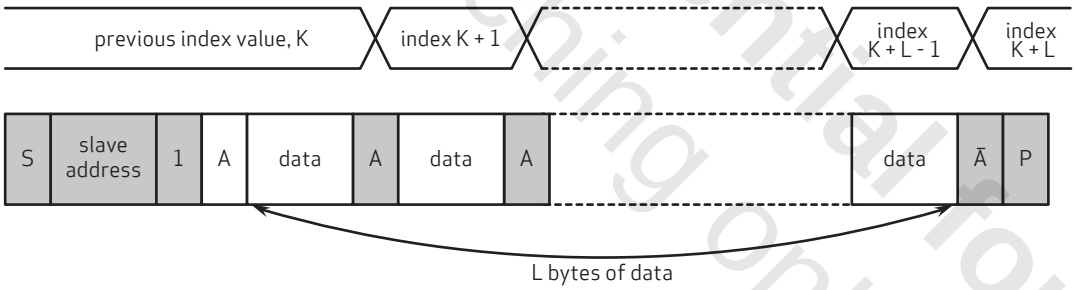
The sequential read from a random location is illustrated in **figure 6-5**. The master does a dummy write to the desired sub-address, issues a repeated start condition after acknowledge from slave and addresses the slave again with read operation. If a master issues an acknowledge after receiving data, 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 6-5 SCCB sequential read from random location



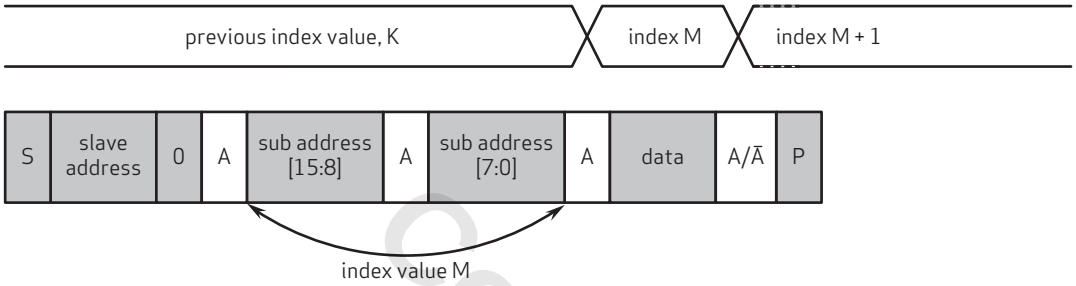
The sequential read from current location is similar to a sequential read from a random location. The only exception is that there is no dummy write operation as shown in **figure 6-6**. The master terminates the read operation by setting a negative acknowledge and stop condition.

figure 6-6 SCCB sequential read from current location



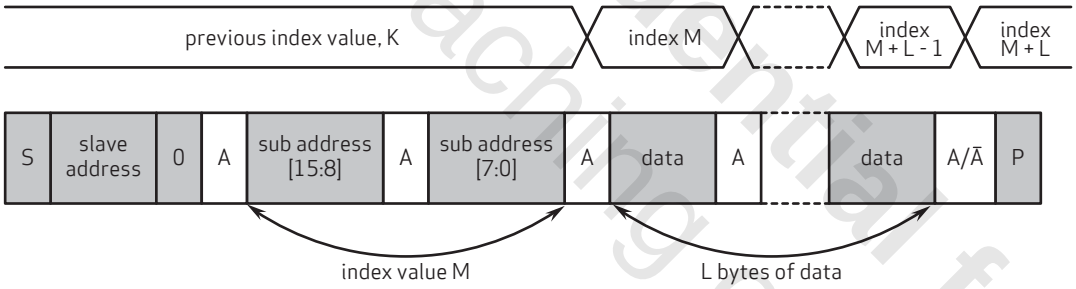
The write operation to a random location is illustrated in **figure 6-7**. The master issues a write operation to the slave, sets the sub-address and data correspondingly after the slave has acknowledged. The write operation is terminated with a stop condition from the master.

figure 6-7 SCCB single write to random location



The sequential write is illustrated in **figure 6-8**. The slave automatically increments the sub-address after each data byte. The sequential write operation is terminated with stop condition from the master.

figure 6-8 SCCB sequential write to random location



7 one-time programmable (OTP) memory

The OS04A10 supports a maximum of 1024 bytes of one-time programmable (OTP/eFUSE) memory. Chip, module identification, register auto-loading or calibration information can be stored. Out of a total of 1024 bytes, TBD bytes are reserved for the user. 1024 bytes of dedicated registers (0x7000~0x73FF) are assigned as the data buffer for OTP.

7.1 OTP write

The following shows how to write data into specific bytes of OTP. As part of the OTP that has already been programmed during manufacturing, this is the recommend way for the user to write to OTP.

1. Set 0x3D84[6] = 1; enable partial OTP mode. Do not run any OTP load or BIST.
2. Set {0x3D88, 0x3D89} = [start address]. For example: {0x3D88, 0x3D89} = 0x7380 sets OTP write start address to 0x7380.
3. Set {0x3D8A, 0x3D8B} = [end address]. For example: {0x3D8A, 0x3D8B} = 0x73FF sets OTP write end address to 0x73FF.
4. Write data into the dedicated buffer defined in **step 2** and **step 3**. For example: write data into 0x7380~0x73FF.
5. Set 0x3D80[0] = 1; OTP write starts.
6. Loop until 0x3D80[7] = 0; When OTP is being programmed, 0x3D80[7] will be 1. It will change to 0 when programming ends.

The OS04A10 supports register auto-loading, which can map a certain length of OTP content to sensor register. This feature is shown as follows:

1. Write command code (An or 5n, 1 byte) + start address of sensor register (2 bytes, optional) + data (n+1 byte) in OTP.
2. Specify start address of OTP in this feature {0x3D8C, 0x3D8D}.
3. Set OTP load option so that the corresponding OTP bytes will be loaded. See **section 7.2**.
   For example, if (0x3D8C, 0x3D8D) = 0x73C0 is OTP buffer start address for register auto-loading, 0x73C0 and hereafter shall be written as shown in **table 7-1**.

table 7-1 data to be written in OTP if allocated for register auto-loading

buffer address	value	note
0x73C0	0xA3	3 + 1 = 4 bytes for auto-loading data. If it is 0x53, it means 4 bytes to be loaded right after previous read/write operation
0x73C1	0x40	auto-loading start address
0x73C2	0x30	
0x73C3	0x01	data to be loaded into 0x4030~0x4033
0x73C4	0x02	
0x73C5	0x03	
0x73C6	0x04	

7.2 OTP read

OTP load can be triggered when the sensor wakes up from software standby (auto load) as well as by register (manual load). In either mode, partial loading and all-byte loading are available. Register auto-load can be enabled independently by 0x3D85[1] = 1 for autoload or 0x3D85[0] = 1 for manual load.

The following shows how to manually load OTP content to OTP buffer and enable register auto-loading.

1. Set 0x3D84[6]; 0 = all-byte load, 1 = partial load
2. Set {0x3D88, 0x3D89} = [start address]. For example: (0x3D88, 0x3D89) = 0x7380 sets OTP write start address to 0x7380.
3. Set (0x3D8A, 0x3D8B) = [end address]; example: (0x3D8A, 0x3D8B) = 0x73FF sets OTP write end address to 0x73FF.
4. Set 0x3D85[0] = 1; 0 = do not use register auto-load in manual load, 1 = use register auto-load in manual load.
5. Set {0x3D8C, 0x3D8D}; start of OTP buffer for register auto-loading.
6. Set 0x3D81 = 0x01; manual load OTP and copy certain OTP buffer values to sensor registers with register auto-loading function.
7. Loop until 0x3D81[7] = 0. When loading OTP 0x3D81[7] will be 1. It will change to 0 when loading finishes.

The OTP memory access conditions are based on typical conditions: sensor wake up, 2.8V AVDD, 1.2V DVDD, and a 108 MHz system clock. To use OTP memory under different operating conditions, please contact your local OmniVision FAE.

8 operating specifications

8.1 absolute maximum ratings

table 8-1 absolute maximum ratings

parameter		absolute maximum rating[a]
ambient storage temperature		-50°C to +125°C
supply voltage (with respect to ground)	V_{DD-A}	4.5V
	V_{DD-D}	3V
	V_{DD-IO}	4.5V
all input/output voltages (with respect to ground)		-0.3V to V_{DD-IO} + 1V
I/O current on any input or output pin		± 200 mA
peak solder temperature (10 second dwell time)		245°C

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

8.2 functional temperature

table 8-2 functional temperature

parameter	range
operating temperature[a]	-30°C to +85°C sensor junction temperature

a. sensor functions in the operating range; however, some image quality changes may be noticed at the temperature extremes

8.3 DC characteristics

table 8-3 DC characteristics (-40°C < T_J < 125°C)

symbol	parameter	min	typ	max	unit
supply					
$V_{DD-2.8}$	supply voltage (analog) AVDD and (PLL) PVDD	2.7	2.8	2.9	V
$V_{DD-1.2}$	supply voltage (digital circuit) DVDD and (MIPI core) EVDD	1.14	1.2	1.26	V
$V_{DD-1.8}$	supply voltage (digital I/O) - DOVDD	1.7	1.8	1.9	V
$I_{DD-2.8}$	active (operating) current - sensor is streaming		TBD	56	mA
$I_{DD-1.2}$			TBD	155	mA
$I_{DD-1.8}$			TBD	0.1	mA
$I_{DDS-PWDN-2.8}$[a]	standby current[b]		TBD		µA
$I_{DDS-PWDN-1.2}$			TBD		µA
$I_{DDS-PWDN-1.8}$			TBD		µA
internally derived voltages[c]					
V_{H1}	positive reference voltage 1	3.3	3.6	3.8	V
V_{H2}	positive reference voltage 2	2.4	2.8	3.1	V
V_{H3}	positive reference voltage 3	2.4	2.8	3.1	V
V_{N1}	negative reference voltage 1	-1.6	-1.4	-0.8	V
V_{N2}	negative reference voltage 2	-1.6	-1.4	-0.8	V
digital inputs (typical conditions: AVDD and PVDD = 2.8V, DOVDD = 1.8V, DVDD and EVDD = 1.2V)					
V_{IL}	input voltage LOW			0.54	V
V_{IH}	input voltage HIGH	1.26			V
C_{IN}	input capacitor			10	pF
digital outputs (standard loading 25 pF)					
V_{OH}	output voltage HIGH	1.62			V
V_{OL}	output voltage LOW			0.18	V
serial interface inputs					
V_{IL}[d]	SCL and SDA	-0.5	0	0.54	V
V_{IH}[d]	SCL and SDA	1.28	1.8	2.32	V

a. standby current without input clock
b. standby current based on room temperature
c. not to be in contact with external devices, except for adjacent capacitors
d. based on DOVDD = 1.8V

8.4 AC characteristics

figure 8-1 clock specification illustration

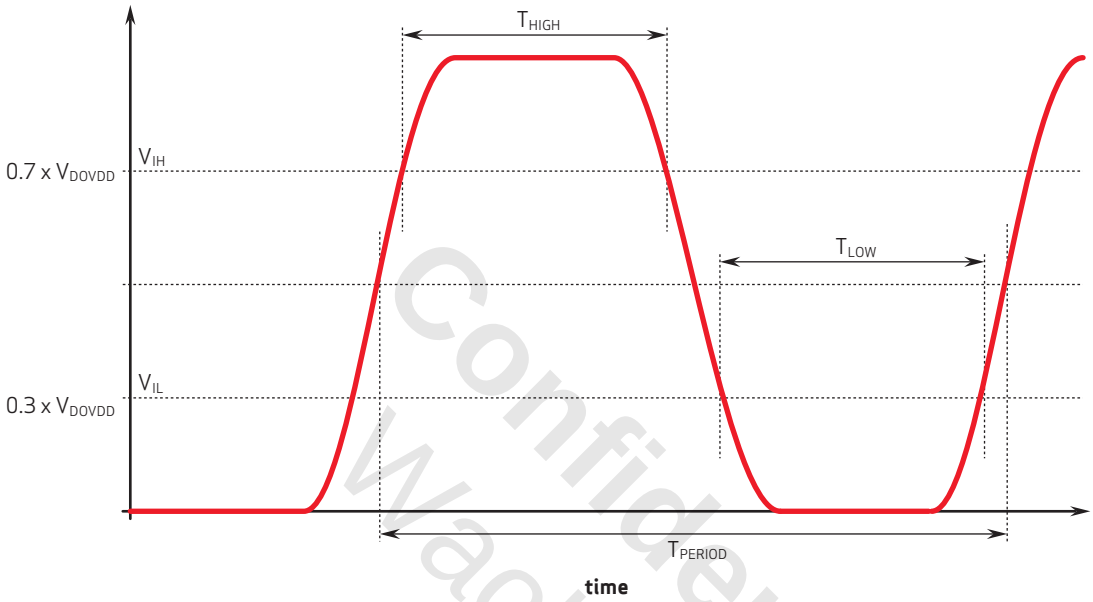


table 8-4 timing characteristics

symbol	parameter	min	typ	max	unit
oscillator and clock input					
f_{EXTCLK}	frequency (EXTCLK)	6	24	36	MHz
T_EXTCLK_jitter	allowed RMS jitter (EXTCLK)			100	ps
C_EXTCLK	input capacitance (EXTCLK)			1.5	pF
T_{PERIOD}	period (EXTCLK)	27.8	41.7	166.7	ns
T_{LOW}	low level width (EXTCLK)	0.35×T_period		0.65×T_period	ns
T_{HIGH}	high level width (EXTCLK)	0.35×T_period		0.65×T_period	ns

9 mechanical specifications

9.1 physical specifications

figure 9-1 package specifications

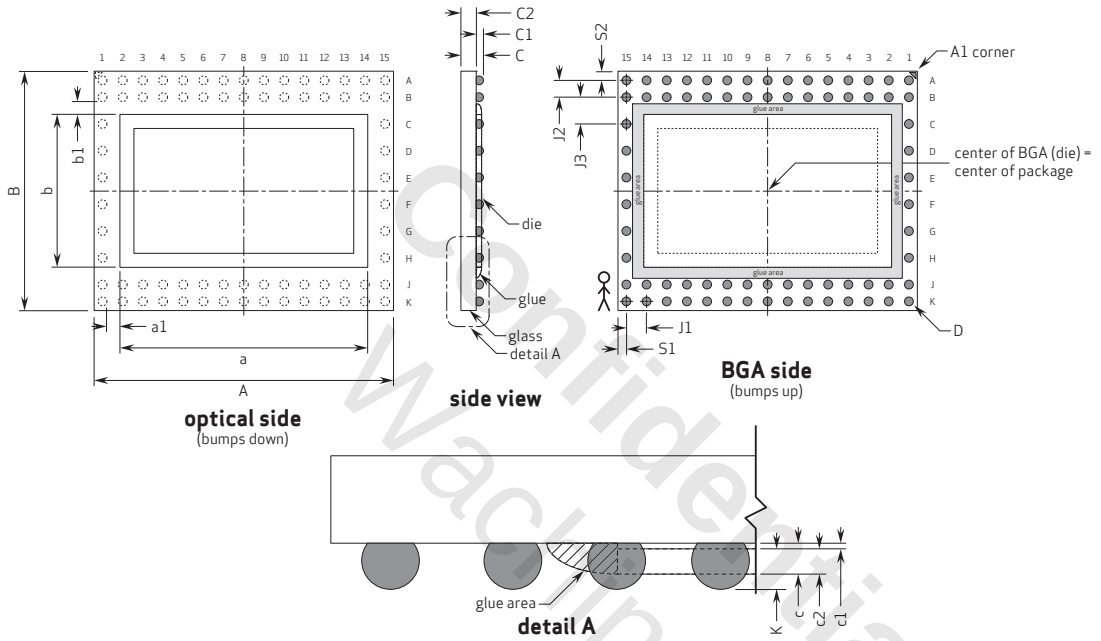


table 9-1 package dimensions (sheet 1 of 2)

parameter	symbol	min	typ	max	unit
package body dimension x	A	10630	10680	10730	µm
package body dimension y	B	8490	8540	8590	µm
package height[a]	C	649	695	741	µm
ball height	C1	218	240	262	µm
package body thickness	C2	431	455	479	µm
ball diameter	D	306	314	322	µm
die body dimension x	a		8842		µm
die edge to ball edge dimension x	a1	444	469	494	µm
die body dimension y	b		5455		µm
die edge to ball edge dimension y	b1	440	465	490	µm

table 9-1 package dimensions (sheet 2 of 2)

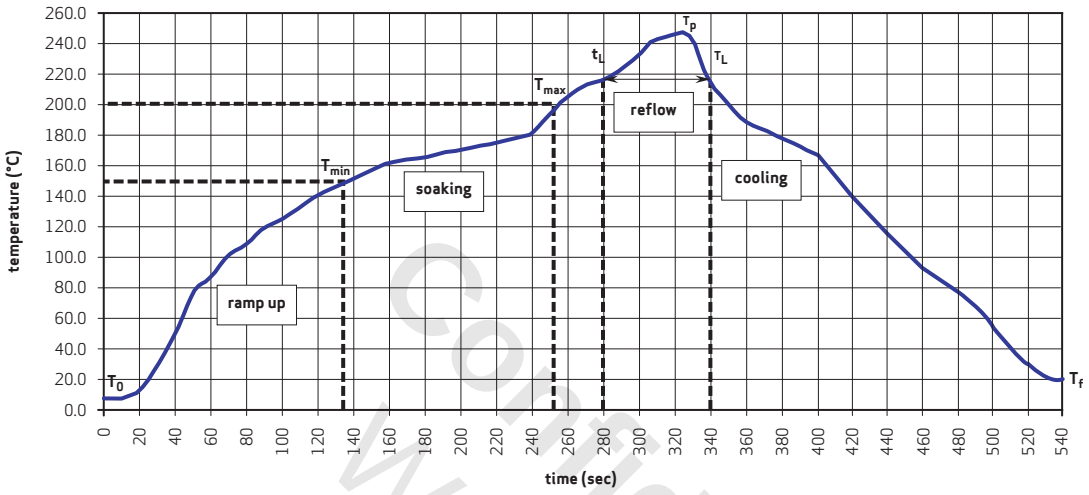
parameter	symbol	min	typ	max	unit
die package height	c	152	167	182	mm
gold bump height	c1	12	17	22	µm
Si thickness	c2	140	150	160	µm
ball pitch x axis	J1		720		µm
ball pitch y1 axis	J2		600		µm
ball pitch y2 axis	J3		955		µm
image plane height	K	200	248	296	µm
total ball count[b]	N		72 (4 NC)		
edge to ball center distance along x	S1	225	300	375	µm
edge to ball center distance along y	S2	253	328	403	µm

a. glass material thickness is 400µm

b. ball placement: solder ball diameter 300µm

9.2 IR reflow specifications

figure 9-2 IR/solder reflow ramp rate profile requirements



note
The OS04A10 uses a lead-free package.

table 9-2 reflow conditions[ab]

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

a. maximum number of reflow cycles = 3

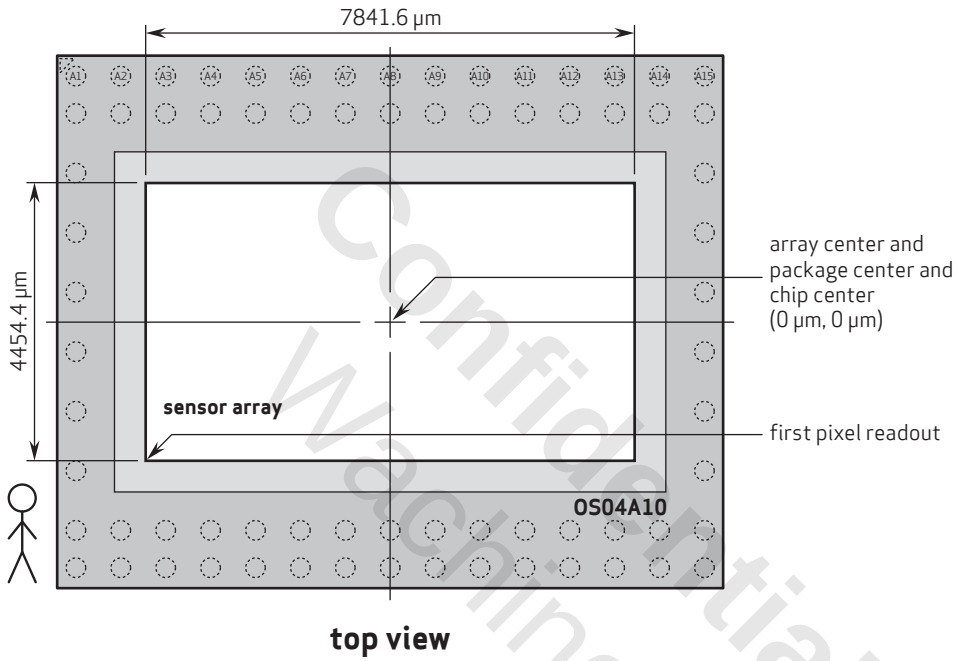
b. N2 gas reflow or control O2 gas PPM<500 as recommendation

note
OmniVision recommends fan-out packages use underfill as part of camera assembly process.

10 optical specifications

10.1 sensor array center

figure 10-1 sensor array center

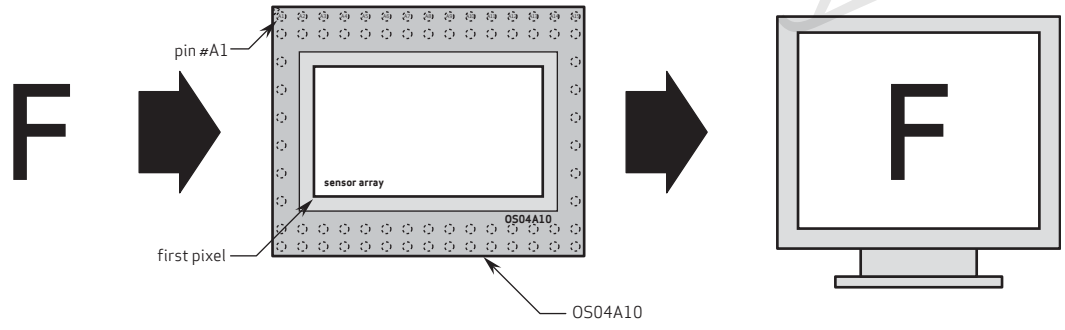


top view

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

note 2 as most optical assemblies flip the image, the chip is typically mounted as above with first pixel at the bottom/left corner.

figure 10-2 final image output



original image **output image**

10.2 lens chief ray angle (CRA)

figure 10-3 chief ray angle (CRA)

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

field (%)	image height (mm)	CRA (degrees)
0.00	0.00	0.00
0.05	0.23	0.45
0.10	0.45	0.90
0.15	0.68	1.35
0.20	0.90	1.80
0.25	1.13	2.25
0.30	1.35	2.70
0.35	1.58	3.15
0.40	1.80	3.60
0.45	2.03	4.05

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

field (%)	image height (mm)	CRA (degrees)
0.50	2.25	4.50
0.55	2.48	4.95
0.60	2.71	5.40
0.65	2.93	5.85
0.70	3.16	6.30
0.75	3.38	6.75
0.80	3.61	7.20
0.85	3.83	7.65
0.90	4.06	8.10
0.95	4.28	8.55
1.00	4.51	9.00

appendix A register table

A.1 module name and address range

table A-1 module name and address range (sheet 1 of 2)

module name	address range
SYSTEM CONTROL (SC)	0x0100~0x010B, 0x3000~0x303F
PLL CONTROL (PLL_CTRL)	0x0300~0x032F
SCCB CONTROL (SCCB TOP)	0x3100~0x3106
GROUP HOLD CONTROL (GRP HOLD)	0x3200~0x322A
ASRAM TST (ASRAM TST)	0x3301~0x3317
PSV CTRL (PSV CTRL)	0x3400~0x342B
EXPOSURE AND GAIN (AEC PK CORE): HCG, LCG, VS	0x3501~0x3538, 0x3541~0x3578, 0x3581~0x35B8
ANALOG CONTROL (ANA TOP)	0x3600~0x3681
SENSOR TOP (SENSOR TOP)	0x3700~0x37FF, 0x3C00~0x3CFF
TIMING CONTROL (TIMING CTRL)	0x3800~0x3888
AEC_101 (AEC_101)	0x3A01~0x3A0F
AVG_12 (AVG_12)	0x3A80~0x3A96
ULPM_PULSE (ULPM_PULSE)	0x3AC0~0x3AED
STROBE (STROBE)	0x3B00~0x3B05
ILLUM_PWM (ILLUM_PWM)	0x3B40~0x3B52
ONE-TIME PROGRAMMABLE CONTROL (OTP_SC)	0x3D80~0x3DA7
SRAM BIST CONTROL (BIST 231 TOP)	0x3E00~0x3E1F
PROGRAMMABLE SRAM CONTROL (PSRAM)	0x3F00~0x3F0F
BLACK LEVEL CORRECTION CONTROL (BLC 157)	0x4100~0x41CF
NEW BLACK LEVEL CORRECTION CONTROL (BLC NEW)	0x4100~0x41A7
ISP FRAME CONTROL (FC TOP)	0x4220~0x4223
WINDOW CONTROL (WINDOW TOP)	0x4280~0x429B
TESTMODE CONTORL (TESTMODE)	0x4300~0x4314
MOTION DETECT CONTROL (MOTION DETECT)	0x4380~0x4398
SYNC FIFO CONTROL (SYNC FIFO)	0x4500~0x4509
VFIFO CONTROL (VFIFO TOP)	0x4600~0x460E

table A-1 module name and address range (sheet 2 of 2)

module name	address range
LVDS1 CONTROL (LVDS1 TOP)	0x4640~0x4656
LVDS2 CONTROL (LVDS2 TOP)	0x4680~0x4696
LVDS3 CONTROL (LVDS3 TOP)	0x46C0~0x46D6
MIPI CORE CONTROL (MIPI CORE TOP)	0x4800~0x4870
MIPI PHY CONTROL (MIPI PHY)	0x4880~0x4887
GIIC MIPI CONTORL (GIIC MIPI)	0x0808~0x0832
WATCHDOG CONTROL (WATCHDOG)	0x4F00~0x4F07
TEMPERATURE MONITOR CONTROL (TPM)	0x4D00~0x4D24
ISP CONTROL (ISP_TOP)	0x5000~0x50AD
PRE ISP CONROL (PRE_ISP): HCG, LCG, VS	0x5080~0x50AD, 0x50C0~0x50ED, 0x5100~0x512D
AWB GAIN CONTROL (AWB_GAIN): HCG, LCG, VS	0x5180~0x518C, 0x51A0~0x51AC, 0x51C0~0x51CC
OTP DPC CONTROL (OTP_DPC)	0x5200~0x5239
DPC CONTROL (OTP_DPC): HCG, LCG, VS	0x5380~0x53D5, 0x5400~0x5455, 0x5480~0x54D5
DPC CONTROL HD (OTP_DPC): HCG, LCG, VS	0x3900~0x3937, 0x3940~0x3977, 0x3980~0x39B0
DPC DUMMY (DPC DUMMY)	0x5680~0x5690
DCG COMBINE CONTROL (DCG_COMBINE)	0x5780~0x57B3

revision history

version 1.0 04.12.2019

- initial release

Future in Sight®

OmniVision Technologies, Inc.

UNITED STATES

4275 Burton Drive
Santa Clara, CA 95054
tel: +1 408 567 3000
Chicago +1 847 508 0217
Detroit +1 248 919 6900
Optics Division +1 303 449 5593

UNITED KINGDOM

Hants +44 1252 761601

GERMANY

Munich +49 89 6381 9988

CHINA

Shanghai + 86 21 6175 9888
+86 21 6154 8000
Beijing + 86 10 8642 4200
Shenzhen + 86 755 8435 9733
Wuhan +86 27 5975 7800
Xian +86 29 8885 2537

INDIA

Bangalore +91 80 4112 8966

KOREA

Seoul + 82 2 3472 3769

JAPAN

Yokohama +81 45 478 7977
Kyoto +81 75 708 5352

NORWAY

Oslo +47 2295 8100

SINGAPORE +65 6933 1933

TAIWAN

Taipei +886 2 2657 9800
Hsin-chu +886 3 5656688

website: **www.ovt.com**