1/3-Inch 3.1M/Full HD Digital Image Sensor

AR0331 Data Sheet

For the latest data sheet, refer to Aptina’s Web site: www.aptina.com

---

Features

- Superior low-light performance
- Latest 2.2um pixel with Aptina A-Pix Technology
- Full HD support at 1080P 60fps for superior video performance
- Linear or high dynamic range capture
- 3.1M (4:3)and 1080P Full HD (16:9) images
- Optional Adaptive Local Tone Mapping
- Interleaved T1/T2 output
- Support for external mechanical shutter
- Support for external LED or xenon flash
- Slow-motion video (VGA 120 fps)
- On-chip phase-locked loop (PLL) oscillator
- Integrated position-based color and lens shading correction
- Slave mode for precise frame-rate control
- Stereo/3D camera support
- Statistics engine
- Data interfaces: four-lane serial high-speed pixel interface (HiSPi™) differential signaling (SLVS), or parallel
- Auto black level calibration
- High-speed Context switching
- Temperature Sensor

Applications

- Video surveillance
- Stereo vision
- Smart vision
- Automation
- Machine vision
- 1080p60 video applications
- High dynamic range imaging

General Description

Aptina's AR0331 is a 1/3-inch CMOS digital image sensor with an active-pixel array of 2052Hx1536V. It captures images in either linear or high dynamic range modes, with a rolling-shutter readout. It includes sophisticated camera functions such as in-pixel binning, windowing and both video and single frame modes. It is designed for both low light and high dynamic range scene performance. It is programmable through a simple two-wire serial interface. The AR0331 produces extraordinarily clear, sharp digital pictures, and its ability to capture both continuous video and single frames makes it the perfect choice for a wide range of applications, including surveillance and HD video.

Table 1: Key Parameters

Parameter		Typical Value
Optical format		1/3-inch (5.8 mm) Note: Sensor optical format will also work with lenses designed for 1/3.2” format.
Active pixels		2052(H) x 1536(V): (entire array) 2048(H) x 1536(V) (4:3, mode)
Pixel size		2.2µm x 2.2 µm
Color filter array		RGB Bayer
Shutter type		Electronic rolling shutter and GRR
Input clock range		6 – 74.25 MHz
Output clock maximum		148.5 Mp/s (4-lane HiSPi) 74.25 Mp/s (Parallel)
Output	Serial	HiSPi 10-, 12-, or 16-bit
	Parallel	10-, 12-bit
Frame rate	Full resolution	30 fps
	1080p	60 fps
Responsivity		1.9 V/lux-sec
SNR_{MAX}		39 dB
Max Dynamic range		Up to 100 dB
Supply voltage	I/O	1.8 or 2.8V
	Digital	1.8 V
	Analog	2.8 V
	HiSPi	0.4V - 0.8V, 1.7V - 1.9V
Power consumption (typical)		<720 mW
Operating temperature (ambient) -T_A		–30°C to + 85° C
Package options		10x10mm 48 pin iLCC 9x9mm 64 pin iBGA

‡Products and specifications discussed herein are for evaluation and reference purposes only and are subject to change by Aptina without notice. Products are only warranted by Aptina to meet Aptina’s production data sheet specifications.

Ordering Information

Table 2: Available Part Numbers

Part Number	Description
AR0331SRSC00SUCA0-E	48 pin iLCC, AS/ES
AR0331SRSC00SUCA0	48 pin iLCC, MP
AR0331SRSC00SUCAH-E	48 pin iLCC, Headboard
AR0331SRSC00SUCAD-E	48 pin iLCC, Demo
AR0331ATSC00XUEA0-E	64 pin iBGA, AS/ES
AR0331ATSC00XUEA0	64 pin iBGA, MP
AR0331ATSC00XUEAH-E	64 pin iBGA, Headboard
AR0331ATSC00XUEAD-E	64 pin iBGA, Demo
AR0331SRSC00SHCA0-E	48 pin iLCC HiSpi, AS/ES
AR0331SRSC00SHCA0	48 pin iLCC HiSpi, MP
AR0331SRSC00SHCAH-E	48 pin iLCC HiSpi, Headboard
AR0331SRSC00SHCAD-E	48 pin iLCC HiSpi, Demo Kit

Table of Contents

List of Figures

List of Tables

General Description

The Aptina® AR0331 can be operated in its default mode or programmed for frame size, exposure, gain, and other parameters. The default mode output is a 1080p-resolution image at 60 frames per second (fps). In linear mode, it outputs 12-bit or 10-bit A-Law compressed raw data, using either the parallel or serial (HiSPi) output ports. In high dynamic range mode, it outputs 12-bit compressed data using parallel output, or 12-bit compressed or 16-bit linearized data using the HiSPi port. The device may be operated in video (master) mode or in single frame trigger mode.

FRAME_VALID and LINE_VALID signals are output on dedicated pins, along with a synchronized pixel clock in parallel mode.

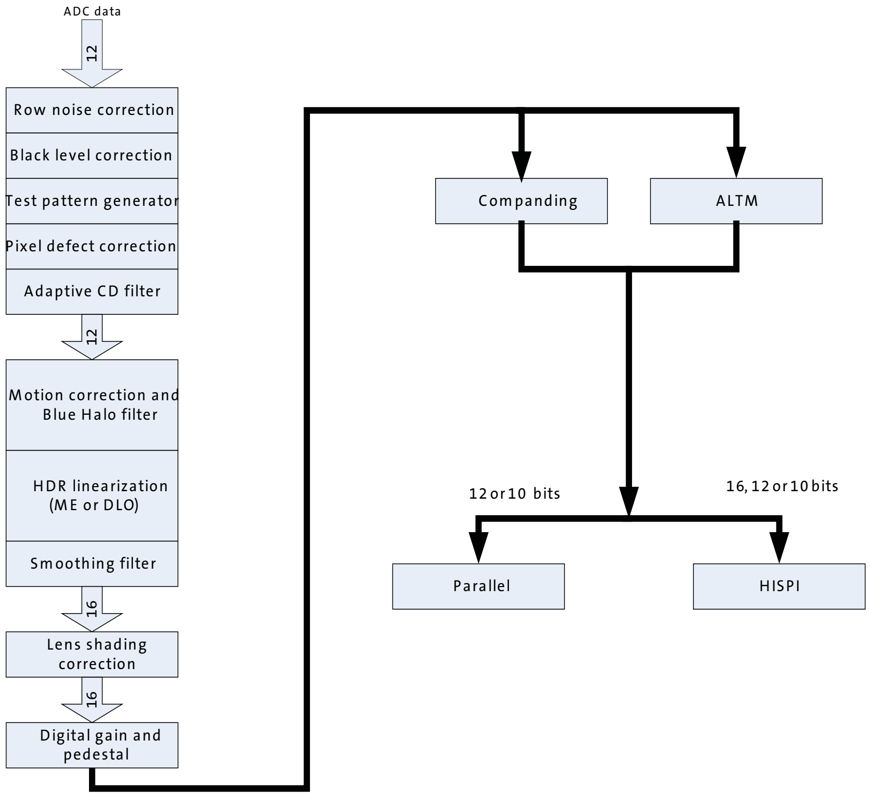
The AR0331 includes additional features to allow application-specific tuning: windowing and offset, auto black level correction, and on-board temperature sensor. Optional register information and histogram statistic information can be embedded in first and last 2 lines of the image frame.

The sensor is designed to operate in a wide temperature range (–30°C to +85°C).

Functional Overview

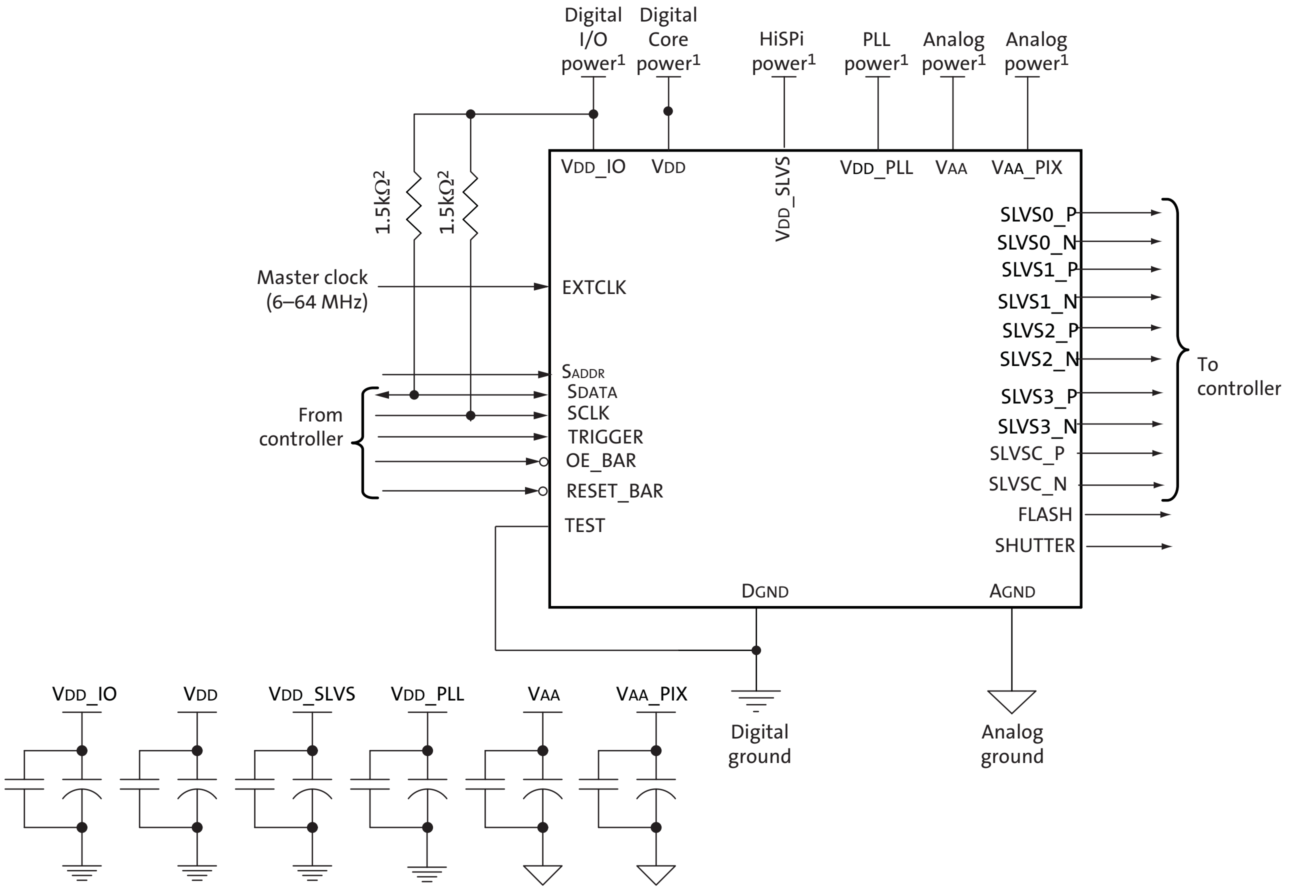
The AR0331 is a progressive-scan sensor that generates a stream of pixel data at a constant frame rate. It uses an on-chip, phase-locked loop (PLL) that can be optionally enabled to generate all internal clocks from a single master input clock running between 6 and 74.25 MHz The maximum output pixel rate is 148.5 Mp/s, corresponding to a clock rate of 74.25 MHz. Figure 1 shows a block diagram of the sensor.

Figure 1: Block Diagram



User interaction with the sensor is through the two-wire serial bus, which communicates with the array control, analog signal chain, and digital signal chain. The core of the sensor is a 3.1 Mp Active- Pixel Sensor array. The timing and control circuitry sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and reading that row, the pixels in the row integrate incident light. The exposure is controlled by varying the time interval between reset and readout. Once a row has been read, the data from the columns is sequenced through an analog signal chain (providing offset correction and gain), and then through an analog-to-digital converter (ADC). The output from the ADC is a 12-bit value for each pixel in the array. The ADC output passes through a digital processing signal chain (which provides further data path corrections and applies digital gain). The sensor also offers a high dynamic range mode of operation where multiple images are combined on-chip to produce a single image at 16-bit per pixel value. A compressing mode is further offered to allow this 16-bit pixel value to be transmitted to the host system as a 12-bit value with close to zero loss in image quality. The pixel data are output at a rate of up to 74.25 Mp/s, in parallel to frame and line synchronization signals.

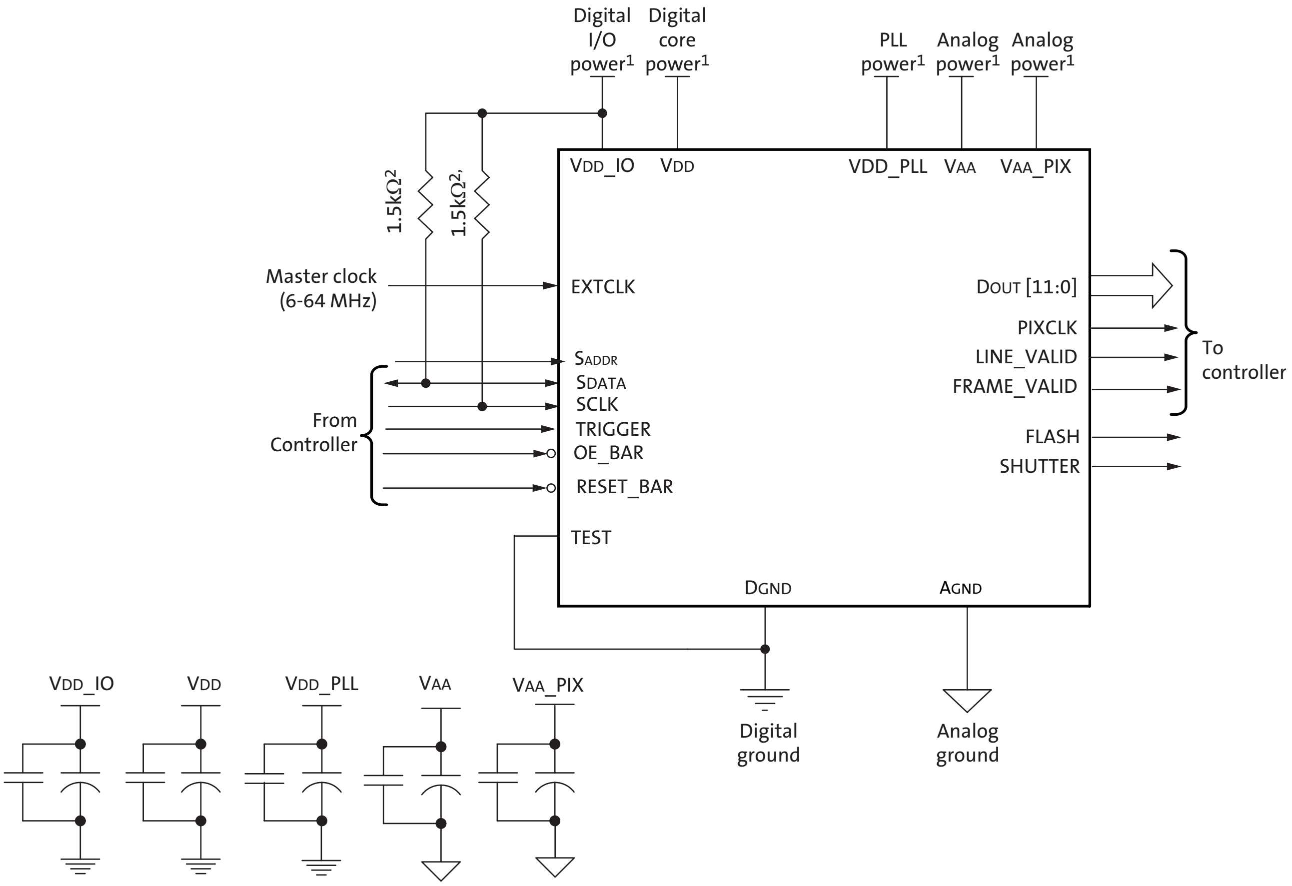
Figure 2: **Typical Configuration: Serial Four-Lane HiSPi Interface**



Notes:
1. All power supplies must be adequately decoupled.
2. Aptina recommends a resistor value of 1.5kΩ, but a greater value may be used for slower two-wire speed.
3. The parallel interface output pads can be left unconnected if the serial output interface is used.

4. Aptina recommends that 0.1µF and 10µF decoupling capacitors for each power supply are mounted as close as possible to the pad. Actual values and results may vary depending on layout and design considerations. Check the AR0331 demo headboard schematics for circuit recommendations.
5. Aptina recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.
6. I/O signals voltage must be configured to match VDD_IO voltage to minimize any leakage currents.

Figure 3: Typical Configuration: Parallel Pixel Data Interface



Notes:
1. All power supplies must be adequately decoupled.
2. Aptina recommends a resistor value of 1.5kΩ, but a greater value may be used for slower two-wire speed.
3. The serial interface output pads and VDDSLVS can be left unconnected if the parallel output interface is used.
4. Aptina recommends that 0.1µF and 10µF decoupling capacitors for each power supply are mounted as close as possible to the pad. Actual values and results may vary depending on layout and design considerations. Check the AR0331 demo headboard schematics for circuit recommendations.
5. Aptina recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.
6. I/O signals voltage must be configured to match VDD_IO voltage to minimize any leakage currents.

Figure 4: 48 iLCC Package, Parallel Output

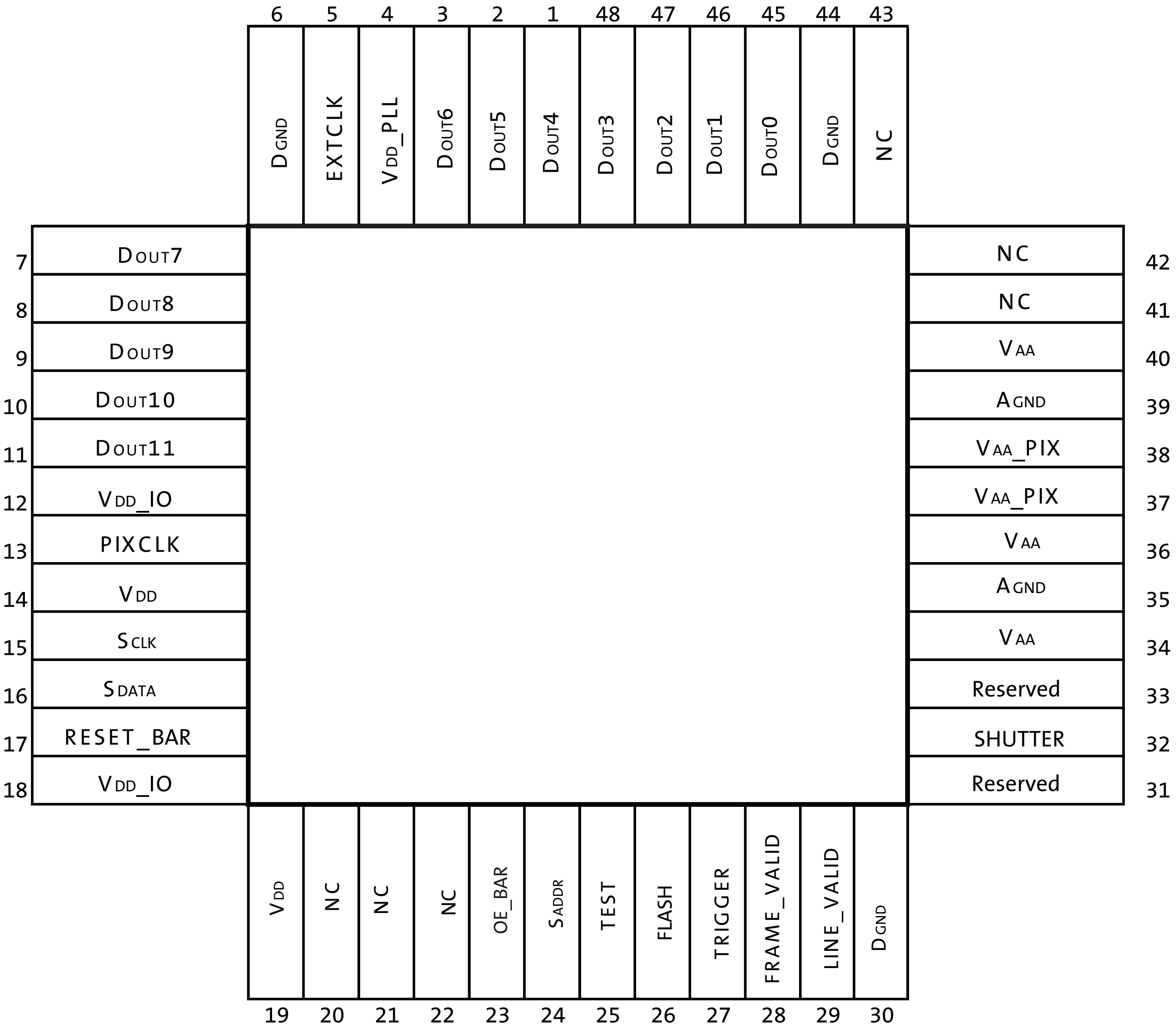


Table 3: Pin Descriptions

Pin Number	Name	Type	Description
1	DOUT4	Output	Parallel pixel data output.
2	DOUT5	Output	Parallel pixel data output.
3	DOUT6	Output	Parallel pixel data output.
4	VDD_PLL	Power	PLL power.
5	EXTCLK	Input	External input clock.
6	DGND	Power	Digital ground.
7	DOUT7	Output	Parallel pixel data output.
8	DOUT8	Output	Parallel pixel data output.
9	DOUT9	Output	Parallel pixel data output.
10	DOUT10	Output	Parallel pixel data output.
11	DOUT11	Output	Parallel pixel data output (MSB).
12	VDD_IO	Power	I/O supply power.
13	PIXCLK	Output	Pixel clock out. DOUT is valid on rising edge of this clock.
14	VDD	Power	Digital power.
15	SCLK	Input	Two-Wire Serial clock input.
16	SDATA	I/O	Two-Wire Serial data I/O.
17	RESET_BAR	Input	Asynchronous reset (active LOW). All settings are restored to factory default.
18	VDD_IO	Power	I/O supply power.
19	VDD	Power	Digital power.
20	NC		
21	NC		
22	NC		
23	OE_BAR	Input	Output enable (active LOW).
24	SADDR	Input	Two-Wire Serial address select. 0: 0x20. 1: 0x30
25	TEST	Input	Manufacturing test enable pin (connect to DGND).
26	FLASH	oUTPUT	Flash output control.
27	TRIGGER	Input	Receives slave mode VD signal for frame rate synchronization and trigger to start a GRR frame.
28	FRAME_VALID	Output	Asserted when DOUT frame data is valid.
29	LINE_VALID	Output	Asserted when DOUT line data is valid.
30	DGND	Power	Digital ground
31	Reserved		
32	SHUTTER	Output	Control for external mechanical shutter. Can be left floating if not used.
33	Reserved		
34	VAA	Power	Analog power.
35	AGND	Power	Analog ground.
36	VAA	Power	Analog power.
37	VAA_PIX	Power	Pixel power.
38	VAA_PIX	Power	Pixel power.
39	AGND	Power	Analog ground.
40	VAA	Power	Analog power.
41	NC		
42	NC		
43	NC		
44	DGND	Power	Digital ground.

Table 3: Pin Descriptions (continued)

Pin Number	Name	Type	Description
45	DOUT0	Output	Parallel pixel data output (LSB)
46	DOUT1	Output	Parallel pixel data output.
47	DOUT2	Output	Parallel pixel data output.
48	DOUT3	Output	Parallel pixel data output.

Figure 5: 48 iLCC Package, HiSPi Output

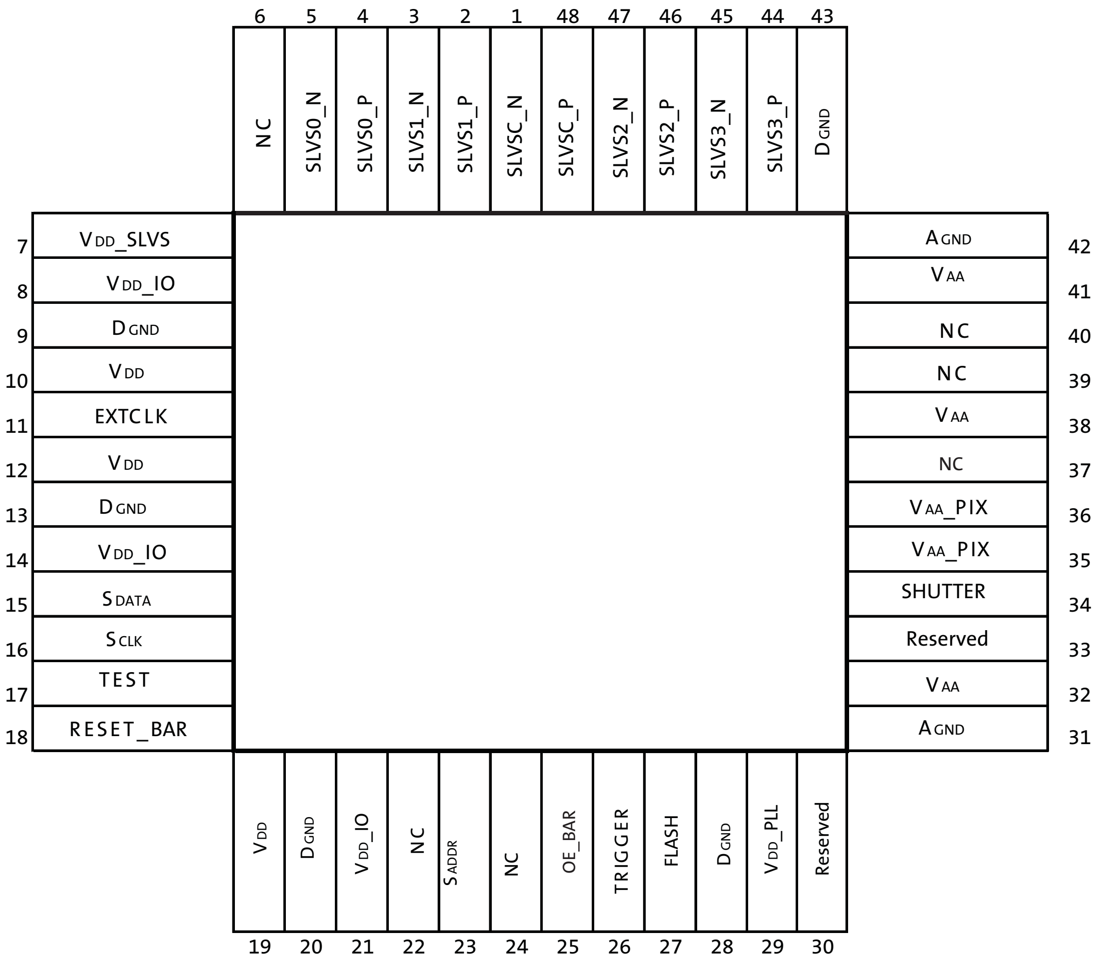


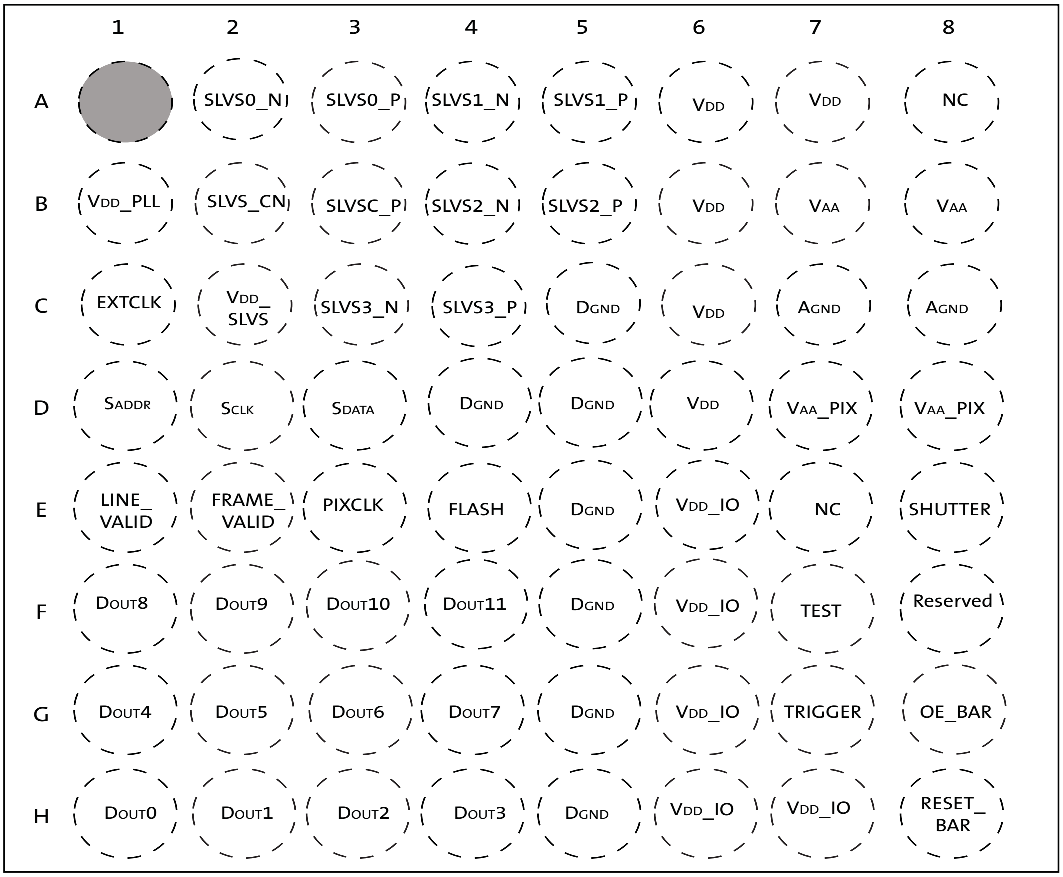
Table 4: Pin Descriptions, 48 iLCC

Pin Number	Name	Type	Description
1	SLVSC_N	Output	HiSPi serial DDR clock differential N.
2	SLVS1_P	Output	HiSPi serial data, lane 1, differential P.
3	SLVS1_N	Output	HiSPi serial data, lane 1, differential N.
4	SLVS0_P	Output	HiSPi serial data, lane 0, differential P.
5	SLVS0_N	Output	HiSPi serial data, lane 0, differential N.
6	NC		

Table 4: Pin Descriptions, 48 iLCC (continued)

Pin Number	Name	Type	Description
7	VDD_SLVS	Power	0.4V-0.8V or 1.7V - 1.9V port to HiSPi Output Driver. Set the High_VCM (R0x306E[9]) bit to 1 when configuring VDD_SLVS to 1.7 – 1.9V.
8	VDD_IO	Power	I/O supply power.
9	DGND	Power	Digital ground.
10	VDD	Power	Digital power.
11	EXTCLK	Input	External input clock.
12	VDD	Power	Digital power.
13	DGND		Digital ground.
14	VDD_IO	Power	I/O supply power.
15	SDATA	I/O	Two-Wire Serial data I/O.
16	SCLK	Input	Two-Wire Serial clock input.
17	TEST		Manufacturing test enable pin (connect to DGND).
18	RESET_BAR	Input	Asynchronous reset (active LOW). All settings are restored to factory default.
19	VDD	Power	Digital power.
20	DGND	Power	Digital ground.
21	VDD_IO	Power	I/O supply power.
22	NC		
23	SADDR	Input	Two-Wire Serial address select. 0: 0x20. 1: 0x30
24	NC		
25	OE_BAR		Output enable (active LOW).
26	TRIGGER	Input	Receives slave mode VD signal for frame rate synchronization and trigger to start a GRR frame.
27	FLASH	Output	Flash output control.
28	DGND	Power	
29	VDD_PLL	Power	PLL power.
30	Reserved		
31	AGND	Power	Analog ground.
32	VAA	Power	Analog power.
33	Reserved		
34	SHUTTER	Output	Control for external mechanical shutter. Can be left floating if not used.
35	VAA_PIX	Power	Pixel power.
36	VAA_PIX	Power	Pixel power.
37	NC		
38	VAA	Power	Analog power.
39	NC		
40	NC		
41	VAA	Power	Analog power.
42	AGND	Power	Analog ground.
43	DGND	Power	Digital ground.
44	SLVS3_P	Output	HiSPi serial data, lane 3, differential P.
45	SLVS3_N	Output	HiSPi serial data, lane 3, differential N.
46	SLVS2_P	Output	HiSPi serial data, lane 2, differential P.
47	SLVS2_N	Output	HiSPi serial data, lane 2, differential N
48	SLVSC_P	Output	HiSPi serial DDR clock differential P.

Figure 6: 9 x 9 mm 64-Ball IBGA Package



	1	2	3	4	5	6	7	8
A	●	SLVS0_N	SLVS0_P	SLVS1_N	SLVS1_P	VDD	VDD	NC
B	VDD_PLL	SLVS_CN	SLVSC_P	SLVS2_N	SLVS2_P	VDD	VAA	VAA
C	EXTCLK	VDD_SLVS	SLVS3_N	SLVS3_P	DGND	VDD	AGND	AGND
D	SADDR	SCLK	SDATA	DGND	DGND	VDD	VAA_PIX	VAA_PIX
E	LINE_VALID	FRAME_VALID	PIXCLK	FLASH	DGND	VDD_IO	NC	SHUTTER
F	DOUT8	DOUT9	DOUT10	DOUT11	DGND	VDD_IO	TEST	Reserved
G	DOUT4	DOUT5	DOUT6	DOUT7	DGND	VDD_IO	TRIGGER	OE_BAR
H	DOUT0	DOUT1	DOUT2	DOUT3	DGND	VDD_IO	VDD_IO	RESET_BAR

Top View
(Ball Down)

Table 5: Pin Descriptions, 9 x 9 mm, 64-ball iBGA Parallel

Name	iBGA Pin	Type	Description
SLVS0_N	A2	Output	HiSPi serial data, lane 0, differential N.
SLVS0_P	A3	Output	HiSPi serial data, lane 0, differential P.
SLVS1_N	A4	Output	HiSPi serial data, lane 1, differential N.
SLVS1_P	A5	Output	HiSPi serial data, lane 1, differential P.
VDD_PLL	B1	Power	PLL power.
SLVSC_N	B2	Output	HiSPi serial DDR clock differential N.
SLVSC_P	B3	Output	HiSPi serial DDR clock differential P.
SLVS2_N	B4	Output	HiSPi serial data, lane 2, differential N.
SLVS2_P	B5	Output	HiSPi serial data, lane 2, differential P.
VAA	B7, B8	Power	Analog power.
EXTCLK	C1	Input	External input clock.
VDD_SLVS	C2	Power	0.4V-0.8V or 1.7V - 1.9V port to HiSPi Output Driver. Set the High_VCM (R0x306E[9]) bit to 1 when configuring VDD_SLVS to 1.7 – 1.9V.
SLVS3_N	C3	Output	HiSPi serial data, lane 3, differential N.
SLVS3_P	C4	Output	HiSPi serial data, lane 3, differential P.

Table 5: Pin Descriptions, 9 x 9 mm, 64-ball iBGA Parallel

Name	iBGA Pin	Type	Description
DGND	C5, D4, D5, E5, F5, G5, H5	Power	Digital ground.
VDD	A6, A7, B6, C6, D6	Power	Digital power.
AGND	C7, C8	Power	Analog ground.
SADDR	D1	Input	Two-Wire Serial address select. 0: 0x20. 1: 0x30
SCLK	D2	Input	Two-Wire Serial clock input.
SDATA	D3	I/O	Two-Wire Serial data I/O.
VAA_PIX	D7, D8	Power	Pixel power.
LINE_VALID	E1	Output	Asserted when DOUT line data is valid.
FRAME_VALID	E2	Output	Asserted when DOUT frame data is valid.
PIXCLK	E3	Output	Pixel clock out. DOUT is valid on rising edge of this clock.
VDD_IO	E6, F6, G6, H6, H7	Power	I/O supply power.
DOUT8	F1	Output	Parallel pixel data output.
DOUT9	F2	Output	Parallel pixel data output.
DOUT10	F3	Output	Parallel pixel data output.
DOUT11	F4	Output	Parallel pixel data output (MSB)
TEST	F7	Input.	Manufacturing test enable pin (connect to DGND).
DOUT4	G1	Output	Parallel pixel data output.
DOUT5	G2	Output	Parallel pixel data output.
DOUT6	G3	Output	Parallel pixel data output.
DOUT7	G4	Output	Parallel pixel data output.
TRIGGER	G7	Input	Exposure synchronization input.
OE_BAR	G8	Input	Output enable (active LOW).
DOUT0	H1	Output	Parallel pixel data output (LSB)
DOUT1	H2	Output	Parallel pixel data output.
DOUT2	H3	Output	Parallel pixel data output.
DOUT3	H4	Output	Parallel pixel data output.
RESET_BAR	H8	Input	Asynchronous reset (active LOW). All settings are restored to factory default.
SHUTTER	E8	Output	Control for external mechanical shutter. Can be left floating if not used.
FLASH	E4	Output	Flash control output.
NC	A8, E7		
Reserved	F8		

Pixel Data Format

Pixel Array Structure

The AR0331 pixel array consists of 2052 columns by 1536 rows of optically active pixels. While the sensor's format is 2048x1536, the additional active columns and active rows are included for use when horizontal or vertical mirrored readout is enabled, to allow readout to start on the same pixel. The pixel adjustment is always performed for monochrome or color versions. The active area is surrounded with optically transparent dummy pixels to improve image uniformity within the active area. Not all dummy pixels or barrier pixels can be read out.

Figure 7: Pixel Array Description

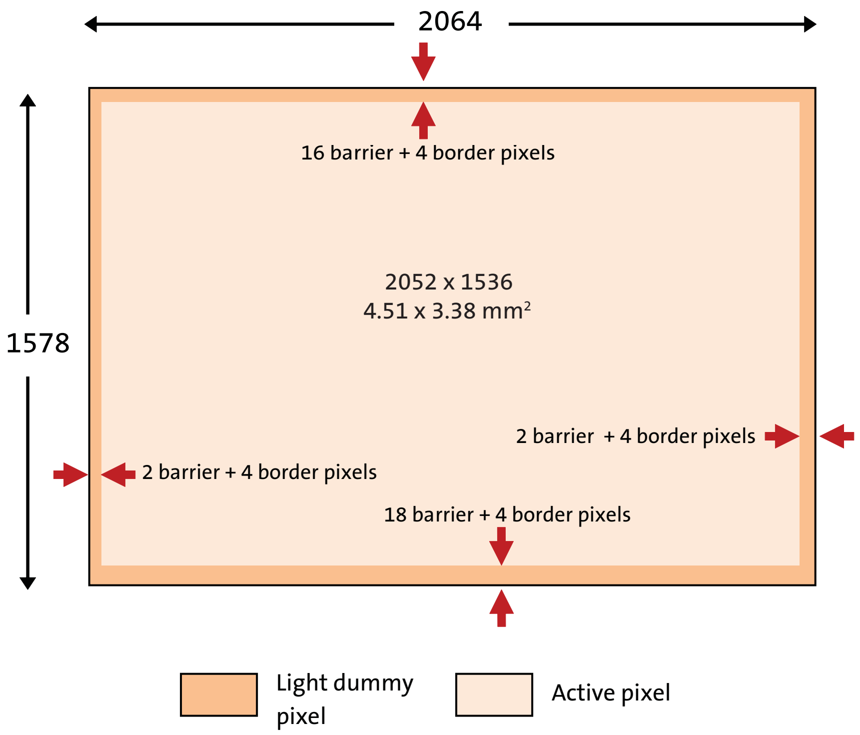
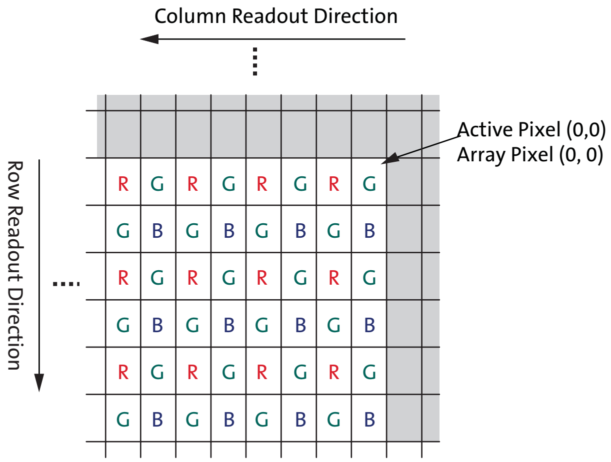


Figure 8: Pixel Color Pattern Detail (Top Right Corner)



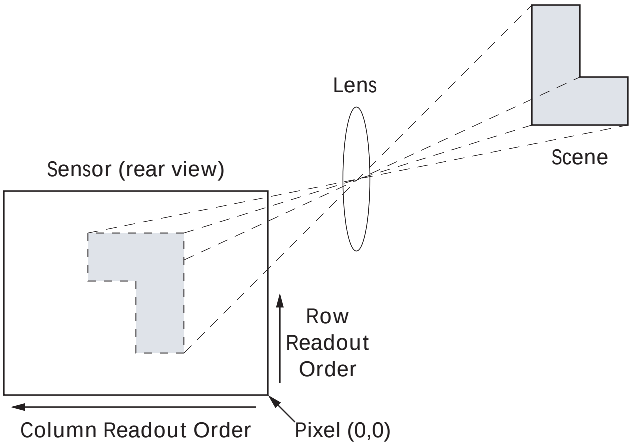
R	G	R	G	R	G	R	G
G	B	G	B	G	B	G	B
R	G	R	G	R	G	R	G
G	B	G	B	G	B	G	B
R	G	R	G	R	G	R	G
G	B	G	B	G	B	G	B

Default Readout Order

By convention, the sensor core pixel array is shown with pixel (0,0) in the top right corner (see Figure 8). This reflects the actual layout of the array on the die. Also, the first pixel data read out of the sensor in default condition is that of pixel (0, 0).

When the sensor is imaging, the active surface of the sensor faces the scene as shown in Figure 9. When the image is read out of the sensor, it is read one row at a time, with the rows and columns sequenced as shown in Figure 9 on page 17.

Figure 9: Imaging a Scene



Positional Gain Adjustments (PGA)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. There are also other factors causing fixed pattern signal gradients in images captured by image sensors. The cumulative result of all these factors is known as image shading. The AR0331 has an embedded shading correction module that can be programmed to counter the shading effects on each individual R, Gb, Gr., and B color signal.

The Correction Function

The correction functions can then be applied to each pixel value to equalize the response across the image as follows:

$$P_{corrected}(row, col) = P_{sensor}(row, col) \times f(row, col) \quad \text{(EQ 1)}$$

where P are the pixel values and f is the color dependent correction functions for each color channel.

Pixel Output Interfaces

Parallel Interface

The parallel pixel data interface uses these output-only signals:

- FV
- LV
- PIXCLK
- DOUT[11:0]

The parallel pixel data interface is disabled by default at power up and after reset. It can be enabled by programming R0x301A. Table 7 shows the recommended settings.

When the parallel pixel data interface is in use, the serial data output signals can be left unconnected. Set reset_register[12] to disable the serializer while in parallel output mode.

Output Enable Control

When the parallel pixel data interface is enabled, its signals can be switched asynchronously between the driven and High-Z under pin or register control, as shown in Table 6.

Table 6: Output Enable Control

OE_BAR Pin	Drive Signals R0x301A–B[6]	Description
Disabled	0	Interface High-Z
Disabled	1	Interface driven
1	0	Interface High-Z
X	1	Interface driven
0	X	Interface driven

Configuration of the Pixel Data Interface

Fields in R0x301A are used to configure the operation of the pixel data interface. The supported combinations are shown in Table 7.

Table 7: Configuration of the Pixel Data Interface

Serializer Disable R0x301 A–B[12]	Parallel Enable R0x301A–B[7]	Standby End-of-Frame R0x301A–B[4]	Description
0	0	1	Power up default. Serial pixel data interface and its clocks are enabled. Transitions to soft standby are synchronized to the end of frames on the serial pixel data interface.
1	1	0	Parallel pixel data interface, sensor core data output. Serial pixel data interface and its clocks disabled to save power. Transitions to soft standby are synchronized to the end of the current row readout on the parallel pixel data interface.
1	1	1	Parallel pixel data interface, sensor core data output. Serial pixel data interface and its clocks disabled to save power. Transitions to soft standby are synchronized to the end of frames in the parallel pixel data interface.

High Speed Serial Pixel Data Interface

The High Speed Serial Pixel (HiSPi) interface uses four data and one clock low voltage differential signaling (LVDS) outputs.

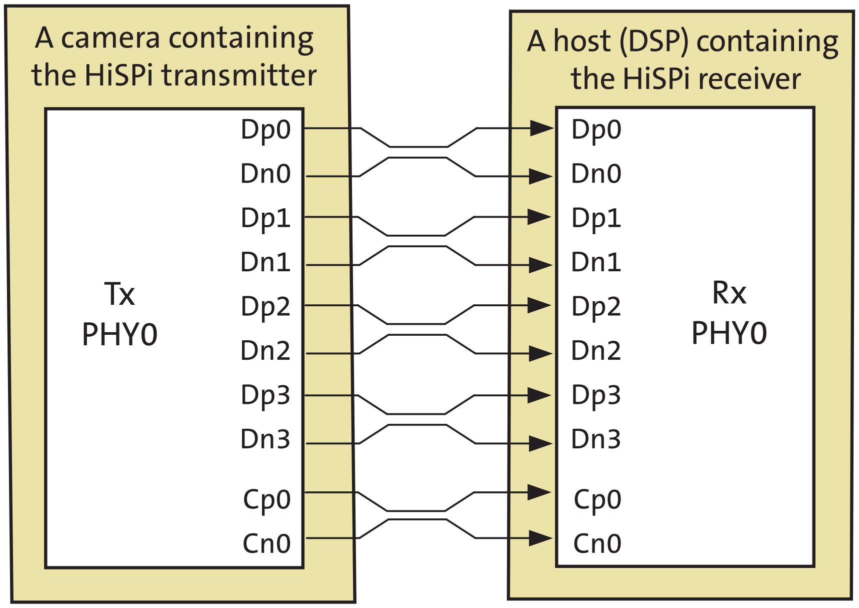
- SLVSC_P
- SLVSC_N
- SLVS0_P
- SLVS0_N
- SLVS1_P
- SLVS1_N
- SLVS2_P
- SLVS2_N
- SLVS3_P
- SLVS3_N

The HiSPi interface supports three protocols, Streaming S, Streaming SP, and Packetized SP. The streaming protocols conform to a standard video application where each line of active or intra-frame blanking provided by the sensor is transmitted at the same length. The Packetized SP protocol will transmit only the active data ignoring line-to-line and frame-to-frame blanking data.

These protocols are further described in the High-Speed Serial Pixel (HiSPi™) Interface Protocol Specification V1.00.00.

The HiSPi interface building block is a unidirectional differential serial interface with four data and one double data rate (DDR) clock lanes. One clock for every four serial data lanes is provided for phase alignment across multiple lanes. Figure 10 shows the configuration between the HiSPi transmitter and the receiver.

Figure 10: **HiSPi Transmitter and Receiver Interface Block Diagram**

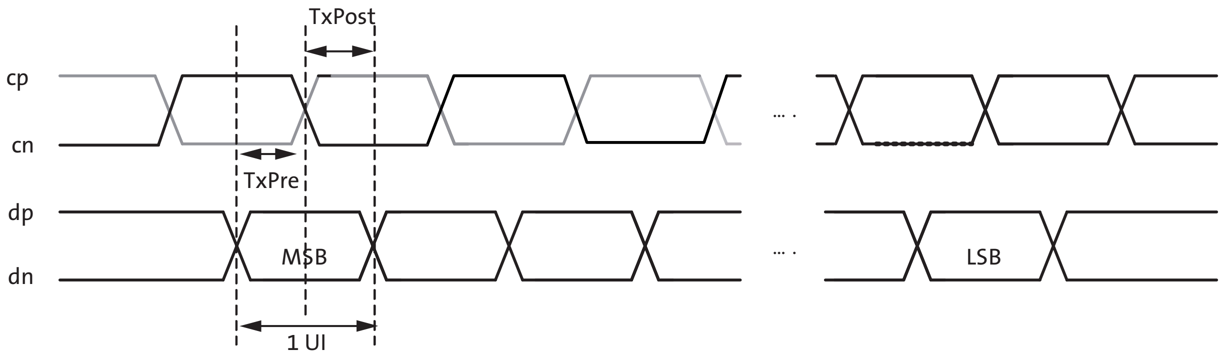


HiSPi Physical Layer

The HiSPi physical layer is partitioned into blocks of four data lanes and an associated clock lane. Any reference to the PHY in the remainder of this document is referring to this minimum building block.

The PHY will serialize a 10-, 12-, or 16-bit data word and transmit each bit of data centered on a rising edge of the clock, the second on the falling edge of clock. Figure 11 shows bit transmission. In this example, the word is transmitted in order of MSB to LSB. The receiver latches data at the rising and falling edge of the clock.

Figure 11: **Timing Diagram**



DLL Timing Adjustment

The specification includes a DLL to compensate for differences in group delay for each data lane. The DLL is connected to the clock lane and each data lane, which acts as a control master for the output delay buffers. Once the DLL has gained phase lock, each lane can be delayed in 1/8 unit interval (UI) steps. This additional delay allows the user to increase the setup or hold time at the receiver circuits and can be used to compensate for skew introduced in PCB design.

If the DLL timing adjustment is not required, the data and clock lane delay settings should be set to a default code of 0x000 to reduce jitter, skew, and power dissipation.

Figure 12: Block Diagram of DLL Timing Adjustment

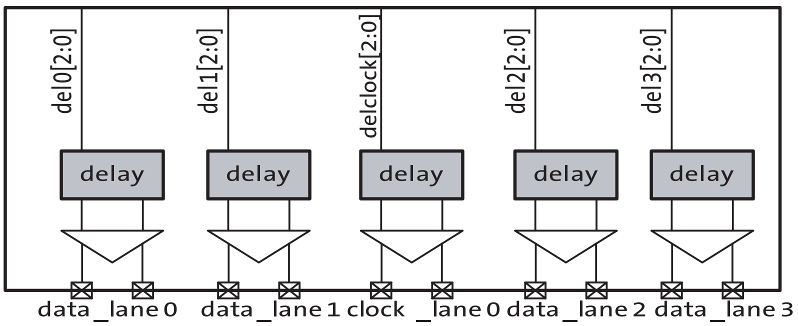


Figure 13: Delaying the clock_lane with Respect to data_lane

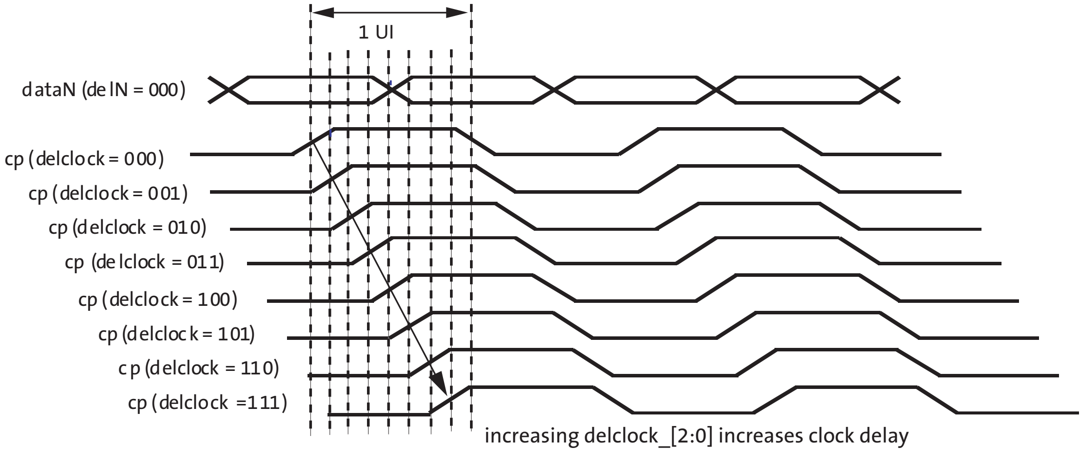
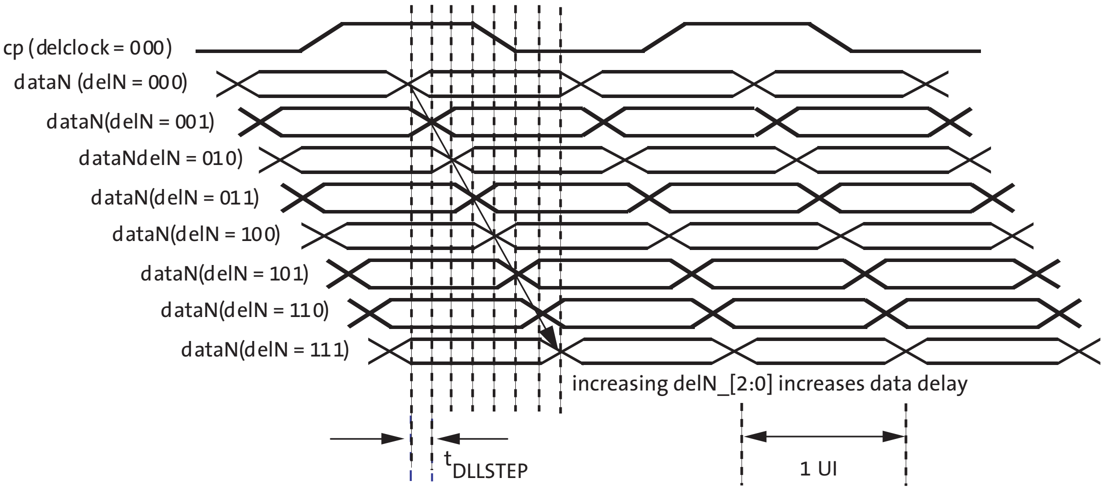


Figure 14: Delaying data_lane with Respect to the clock_lane



HiSPi Streaming Mode Protocol Layer

The HiSPi protocol is described HiSPi Protocol V1.00.00 A.

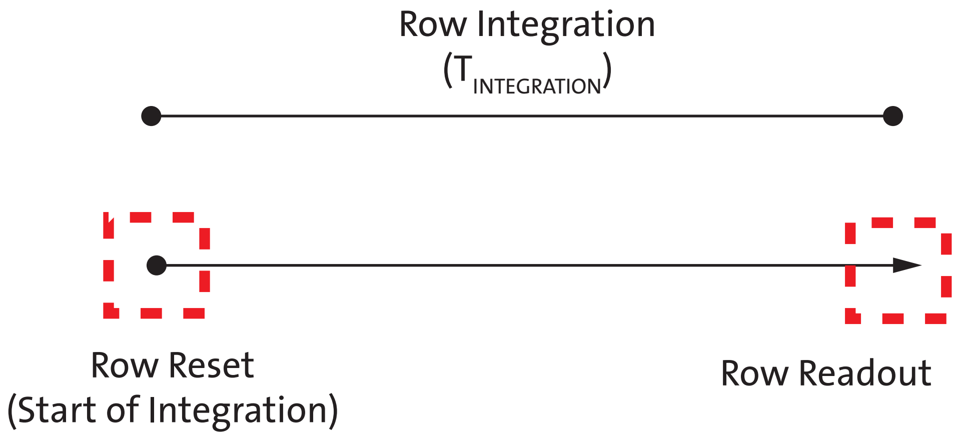
Serial Configuration

The serial format should be configured using R0x31AC. Refer to the AR0331 Register Reference document for more detail regarding this register.

The serial_format register (R0x31AE) register controls which serial interface is in use when the serial interface is enabled (reset_register[12] = 0). The following serial format is supported:

- 0x0304 - Sensor supports quad-lane HiSPi operation
- 0x0302 - Sensor supports dual-lane HiSPi operation
- 0x0301 - Sensor supports single-lane HiSPi operation

Figure 15: Integration Control in ERS Readout



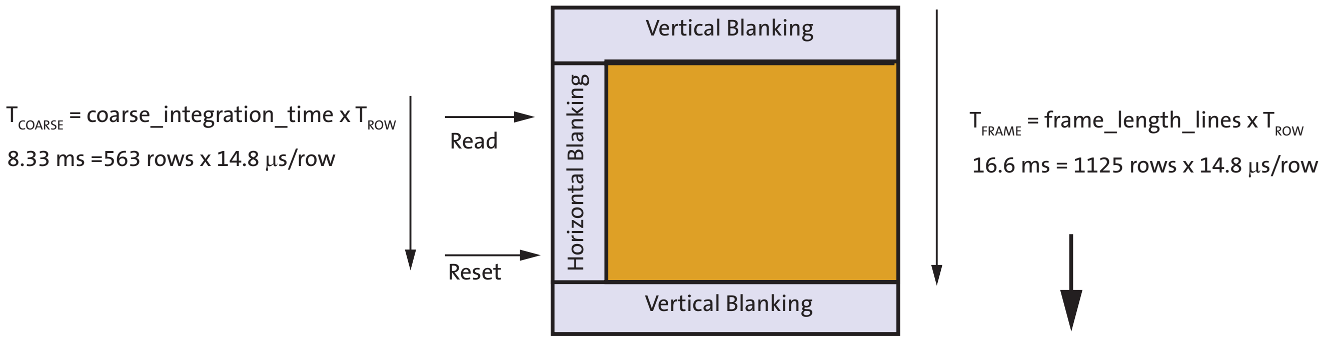
A pixel's integration time is defined by the number of clock periods between a row's reset and read operation. Both the read followed by the reset operations occur within a row period (T_{ROW}) where the read and reset may be applied to different rows. The read and reset operations will be applied to the rows of the pixel array in a consecutive order.

The integration time in an ERS frame is defined as $T_{INTEGRATION} = T_{COARSE} - T_{FINE}$.

The coarse integration time is defined by the number of row periods (T_{ROW}) between a row's reset and the row read. The row period is the defined as the time between row read operations (see Sensor Frame Rate).

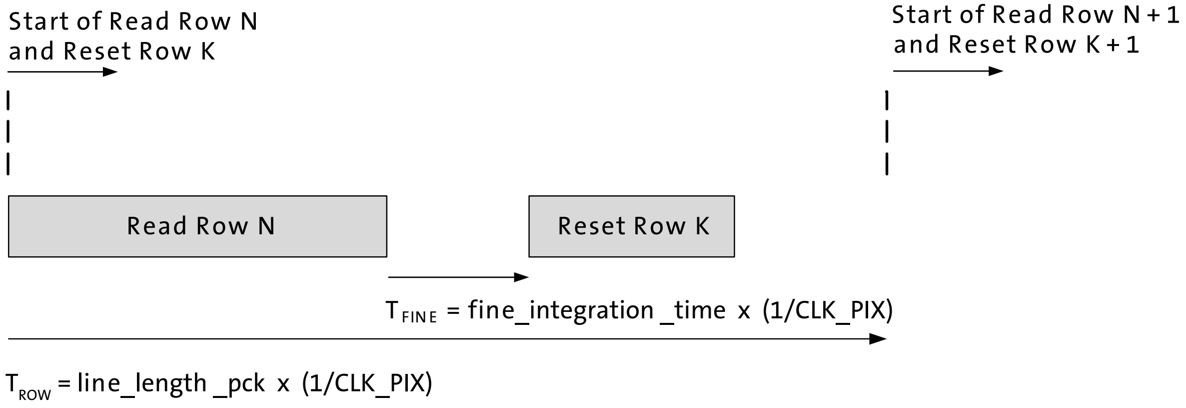
$$T_{COARSE} = T_{ROW} * \text{coarse\_integration\_time} \quad \text{(EQ 2)}$$

Figure 16: Example of 8.33ms Integration in 16.6ms Frame



The fine integration is then defined by the number of pixel clock periods between the row reset and row read operation within T_{ROW}. This period is defined by the *fine_integration_time* register.

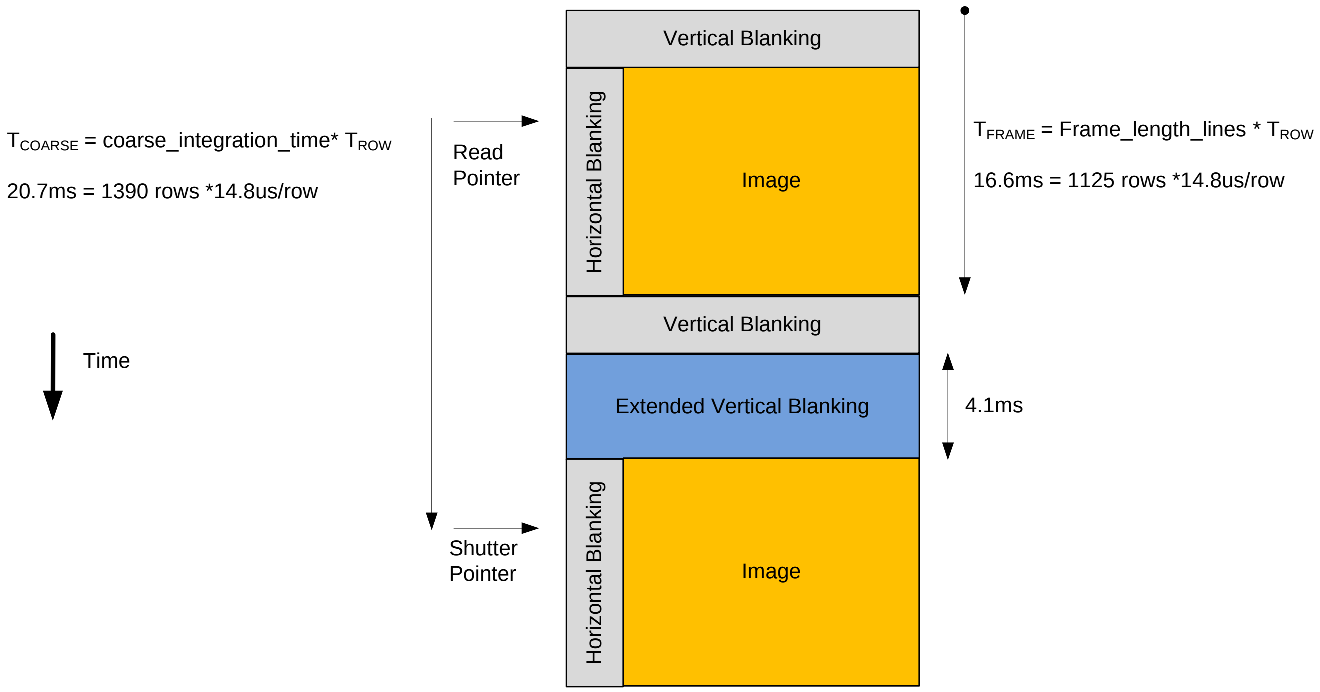
Figure 17: Row Read and Row Reset Showing Fine Integration



$$T_{FINE} = \text{fine\_integration\_time}/\text{clk\_pix} \tag{EQ 3}$$

The maximum allowed value for fine_integration_time is *line_length_pck - 1100*.

Figure 18: The Row Integration Time is Greater Than the Frame Readout Time

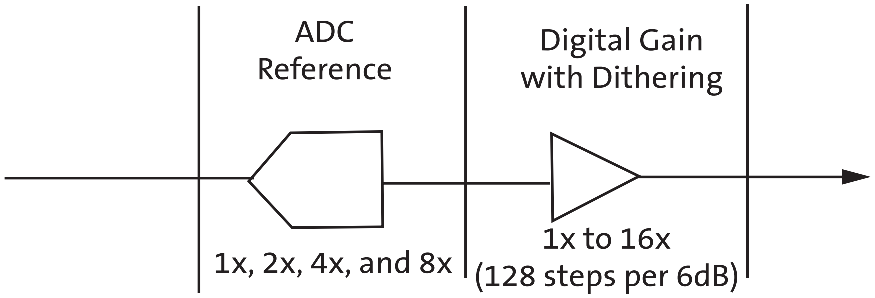


The minimum frame-time is defined by the number of row periods per frame and the row period. The sensor frame-time will increase if the *coarse_integration_time* is set to a value equal to or greater than the f*rame_length_lines*. The maximum integration time can be limited to the frame time by setting R0x30CE[5] to 1.

Gain Stages

The analog gain stages of the AR0331 sensor are shown in Figure 19. The sensor analog gain stage consists of column amplifiers and a variable ADC reference. The sensor will apply the same analog gain to each color channel. Digital gain can be configured to separate levels for each color channel.

Figure 19: Gain Stages in AR0331 Sensor



The level of analog gain applied is controlled by the coarse_gain register. The analog readout circuitry can be configured differently for each analog gain level. The recommended analog gain tables listed in Table 8 can be configured using the registers listed in Table 9. It is recommended that these registers are configured before streaming images.

Table 8: Recommended Sensor Gain

coarse_gain (0x3060[5:4])/ coarse_gain_cb (0x3060[13:12])	fine_gain (0x3060[3:0])/ fine_gain_cb (0x3060[11:8])	ADC Gain
0	0	1.000
0	1	1.032
0	2	1.067
0	3	1.103
0	4	1.143
0	5	1.185
0	6	1.231
0	7	1.280
0	8	1.333
0	9	1.391
0	10	1.455
0	11	1.524
0	12	1.600
0	13	1.684
0	14	1.778
0	15	1.882
1	0	2.000
1	2	2.133
1	4	2.285714286
1	6	2.461538462
1	8	2.666666667
1	10	2.909090909
1	12	3.2
1	14	3.555555556
2	0	4
2	4	4.571428571
2	8	5.333333333
2	12	6.4
3	0	8
3	8	10.66666667

Table 9: Recommended Registers to Configure Sensor Gain Table

Register	Value	Description
R0x3ECE[7:0]	0xFF	Reserved
R0x3ED0[15:4]	0xE4F	Reserved

Each digital gain can be configured from a gain of 0 to 15.875. The digital gain supports 128 gain steps per 6dB of gain. The format of each digital gain register is “xxxx.yyyyyyy” where “xxxx” refers an integer gain of 1 to 15 and “yyyyyyy” is a fractional gain ranging from 0/128 to 127/128.

The sensor includes a digital dithering feature to reduce quantization resulting from using digital gain. can be implemented by setting R0x30BA[5] to 1. The default value is 0.

Refer to “Temperature Sensor” on page 47 for the analog and digital gain registers in both context A and context B modes.

Data Pedestal

The data pedestal is a constant offset that is added to pixel values at the end of datapath. The default offset is 168 and is a 12-bit offset. This offset matches the maximum range used by the corrections in the digital readout path.

The data pedestal value can be changed if the lock register bit (R0x301A[3]) is set to “0”. This bit is set to “1” by default.

High Dynamic Range Mode

By default, the sensor powers up in HDR Mode, however, the AR0331 can be configured to run in Linear mode. The HDR scheme used is multi-exposure HDR. This allows the sensor to handle up to 100dB of dynamic range. The sensor also features a linear mode. In HDR mode, the sensor sequentially captures two exposures by maintaining two separate read and reset pointers that are interleaved within the rolling shutter readout. The intermediate pixel values are stored in line buffers while waiting for the two exposure-values to be present. As soon as a pixel's two exposure values are available, they are combined to create a linearized 16-bit value for each pixel’s response. This 16-bit value is then compressed through the Adaptive Local Tone Mapping (ALTM) block to a 12-bit value for output.

Adaptive Local Tone Mapping

The real world scenes often have very high dynamic range (HDR) that far exceeds the electrical dynamic range of the imager. Dynamic range is defined as the luminance ratio between the brightest and the darkest object in a scene. Even though the AR0331 can capture full dynamic range images, the images are still limited by the low dynamic range display devices. Today’s typical LCD monitor has contrast ratio around 1,000:1 while it is not atypical for an HDR image having contrast ratio around 250,000:1. Therefore, in order to reproduce HDR images on a low dynamic range display device, the captured high dynamic range must be compressed to the available range of the display device. This is commonly called tone mapping. The AR0331 has implemented an adaptive local tone mapping (ALTM) feature to reproduce visually appealing images that increase the local contrast and the visibility of the images. Figure 20 on page 28 shows the HDR data compression: The HDR mode is selected when Operation_Mode_Ctrl, R0x3082[1:0] = 0. Further controls on exposure time limits and compressing are controlled by R0x3082[3:2]and R0x31D0. More details can be found in the AR0331 Register Reference.

In HDR mode, when compression is used, there are two types of knee-points: (i) T1/T2 capture knee-points and (ii) POUT and POUT2 compression knee-points (Figure 20 on page 28). Aligning the capture knee-points on top of the compression knee-points, can avoid code losses (SNR loss) in the compression. Table 10 below shows the knee points for the different modes.

Figure 20: HDR Data Compression

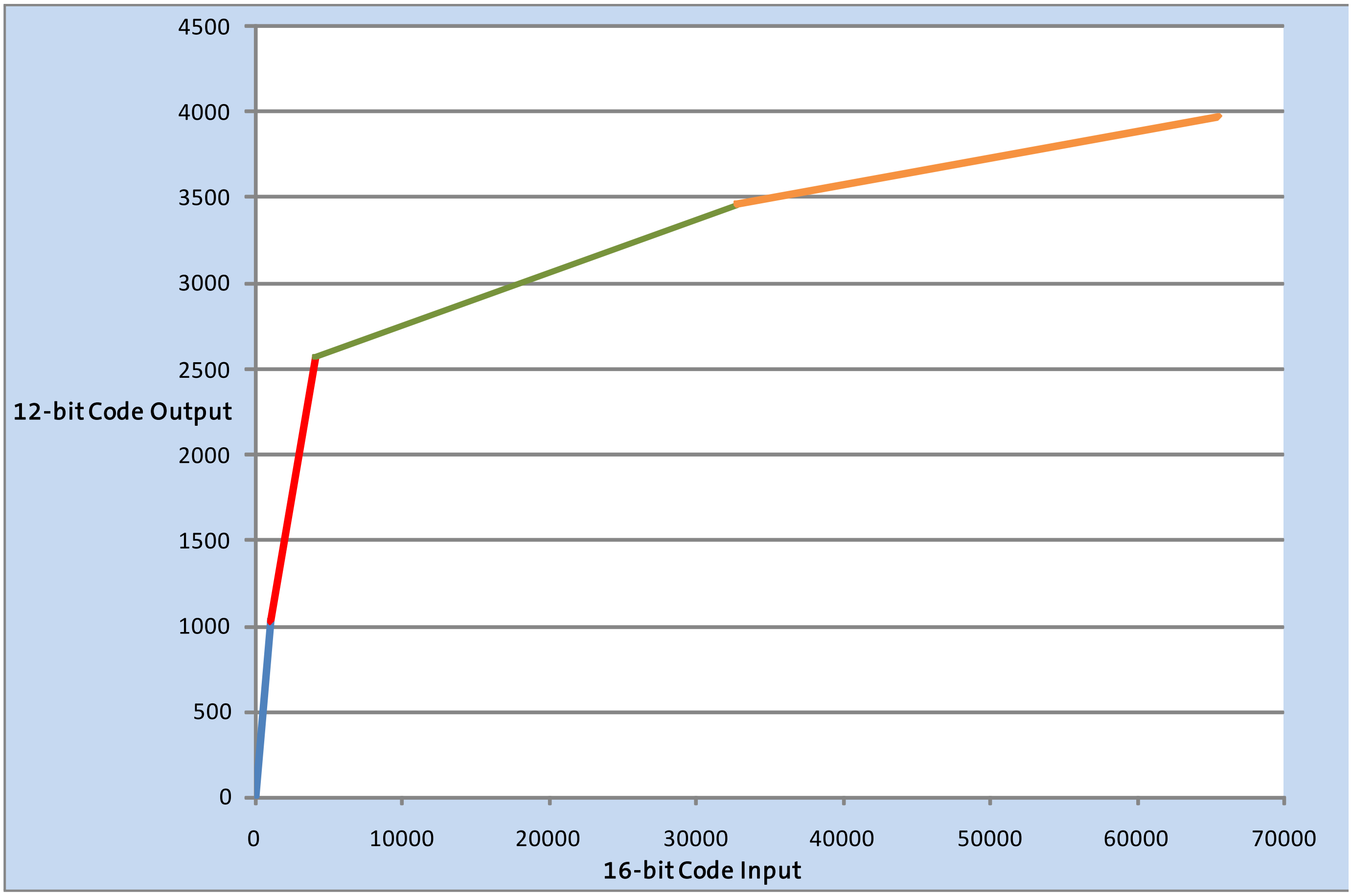


Table 10: Knee Points for Compression from 16 Bits to 12 Bits

T1/T2 Exposure Ratio (R1) R0x3082[3:2]	P1	POUT1 = P1	P2	POUT2= (P2 - P1)/2 + 1024	P3	POUT3= (P3 - P2)/32 + 2560	PMAX	POUTMAX = (PMAX - P3)/64 +3456
4x, 8x, 16x, 32x	2^{10}	1024	2^{12}	2560	2^{15}	3456	2^{16}	3968

As described inTable 10, the AR0331 companding block operates in 16-bit input only. For the exposure ratios that do not result in 16-bit, bit shifting occurs before the data enters the companding block. As a result of the bit shift, data needs to be unshifted after linearization in order to obtain the proper image. Table 11 provides the bit operation that should occur to the data after linearization.

Table 11: Bit Operation After Linearization

ratio_t1_t2 (R0x3082[3:2])/ ratio_t1_t2_cb (R0x3084[3:2])	Bit Shift Operation after Linearization
4x	Right shift 2 bits
8x	Right shift 1 bit
16x	No shift
32x	Left shift 1 bit

HDR-Specific Exposure Settings

In HDR mode, pixel values are stored in line buffers while waiting for both exposures to be available for final pixel data combination. There are 70 line buffers used to store intermediate T1 data. Due to this limitation, the maximum coarse integration time possible is equal to 70*T1/T2 lines.

For example, if R0x3082[3:2] = 2, the sensor is set to have T1/T2 ratio = 16x. Therefore the maximum number of integration lines is 70*16 = 1120 lines. If coarse integration time is greater than this, the T2 integration time will stay at 70. The sensor will calculate the ratio internally, enabling the linearization to be performed. If companding is being used, then relinearization would still follow the programmed ratio. For example if the T1/T2 ratio was programmed to 16x but coarse integration was increased beyond 1120 than one would still use the 16x relinearization formulas.

An additional limitation is the maximum number of exposure lines in relation to the frame_length_lines register. In Linear mode, maximum coarse_integration_time = frame_length_lines - 1. However in HDR mode, since the coarse integration time register controls T1, the max coarse_integration time is frame_length_lines - 71.

Putting the two criteria listed above together, the formula is as follows:

$$maximum\ coarse\_integration\_time = minimum(70 \times T1 / T2,\ frame\_length\_lines - 71) \qquad \text{(EQ 4)}$$

In HDR mode, subline integration is not utilized. As such, fine integration time register changes will have no effect on the image.

There is also a limitation of the minimum number of exposure lines, which is one row time for linear mode. In HDR mode, the minimum number of rows required is half of the ratio T1/T2.

Motion Compensation

In typical multi-exposure HDR systems, motion artifacts can be created when objects move during the T1 or T2 integration time. When this happens, edge artifacts can potentially be visible and might look like a ghosting effect.

To correct this feature, the AR0331 has special 2D motion compensation circuitry that detects motion artifacts and corrects the image.

The motion compensation feature can be enabled by setting R0x318C[14] = 1. Additional parameters are available to control the extent of motion detection and correction as per the requirements of the specific application. These can be set in R0x318C-R0x31A2. For more information, refer to the AR0331 Register Reference document.

Features

See the AR0331 Register Reference for additional details.

Reset

The AR0331 may be reset by using RESET_BAR (active LOW) or the reset register.

Hard Reset of Logic

The RESET_BAR pin can be connected to an external RC circuit for simplicity. The recommended RC circuit uses a 10kΩ resistor and a 0.1μF capacitor. The rise time for the RC circuit is 1μs maximum.

Soft Reset of Logic

Soft reset of logic is controlled by the R0x301A Reset register. Bit 0 is used to reset the digital logic of the sensor while preserving the existing two-wire serial interface configuration. Furthermore, by asserting the soft reset, the sensor aborts the current frame it is processing and starts a new frame. This bit is a self-resetting bit and also returns to “0” during two-wire serial interface reads.

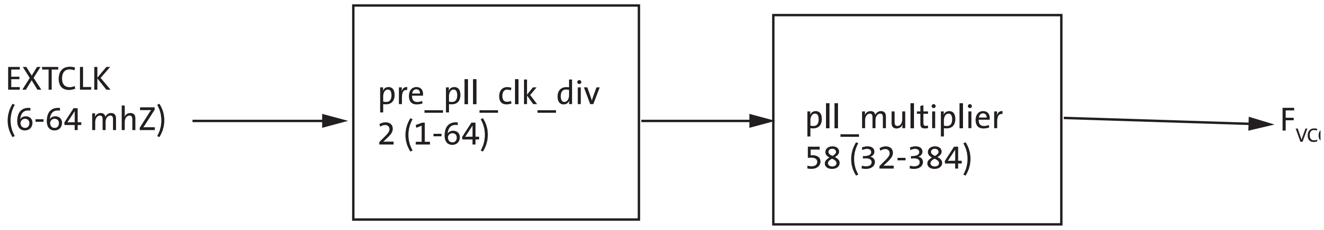
Clocks

The AR0331 requires one clock input (EXTCLK).

Sensor PLL

VCO

Figure 21: Relationship Between Readout Clock and Peak Pixel Rate

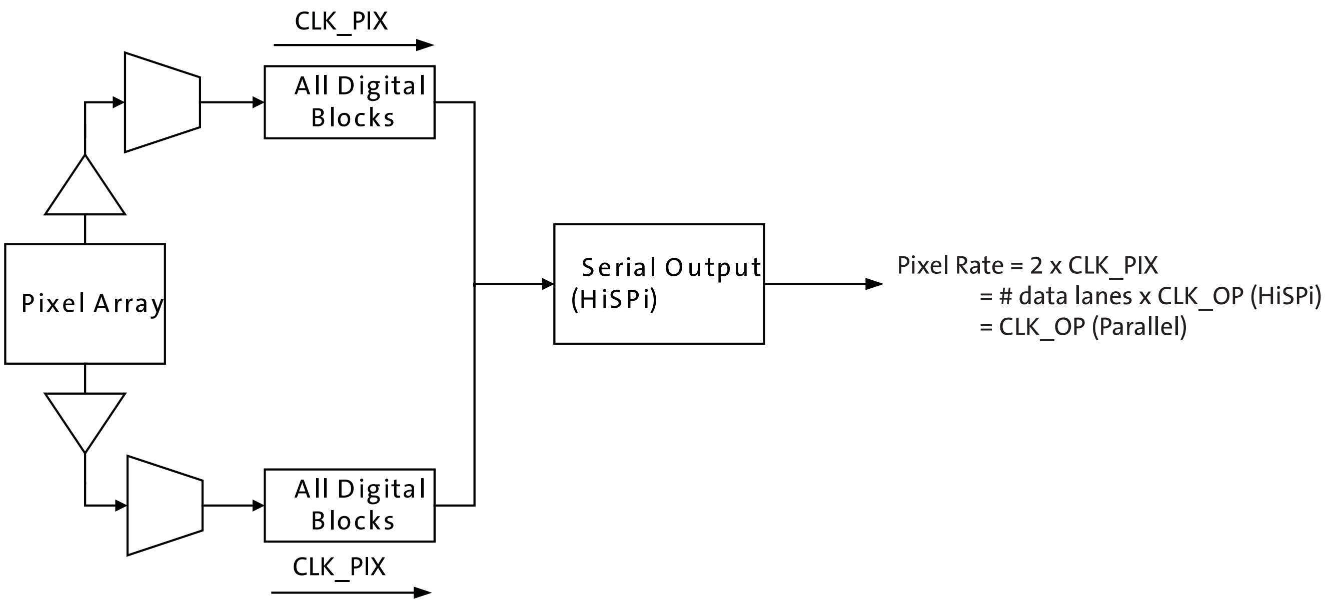


The sensor contains a phase-locked loop (PLL) that is used for timing generation and control. The required VCO clock frequency is attained through the use of a pre-PLL clock divider followed by a multiplier. The multiplier is followed by set of dividers used to generate the output clocks required for the sensor array, the pixel analog and digital readout paths, and the output parallel and serial interfaces.

Dual Readout Paths

There are two readout paths within the sensor digital block. The sensor PLL should be configured so that the total pixel rate across both readout paths equals the total pixel relationship between sensor clock configuration and peak pixel rate.

Figure 22: Sensor Dual Readout Paths

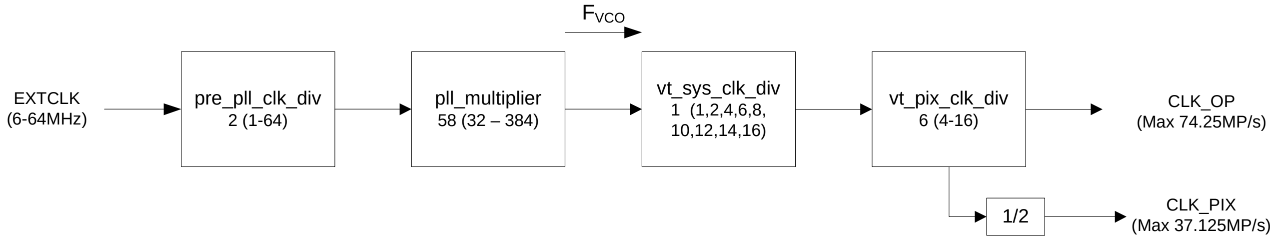


The sensor row timing calculation refers to each data-path individually. For example, the sensor default configuration uses 1100 clocks per row (line_length_pck) to output 1928 active pixels per row. The aggregate clocks per row seen by the receiver will be 2200 clocks (1100 x 2 readout paths).

Parallel PLL Configuration

Figure 23: PLL for the Parallel Interface

The parallel interface has a maximum output data-rate of 74.25MP/s.



The maximum output of the parallel interface is 74.25 MPixel/s (CLK_OP). This will limit the readout clock (CLK_PIX) to 37.125 MPixel/s. The sensor will not use the F_{SERIAL}, F_{SERIAL_CLK}, or CLK_OP when configured to use the parallel interface.

Table 12: PLL Parameters for the Parallel Interface

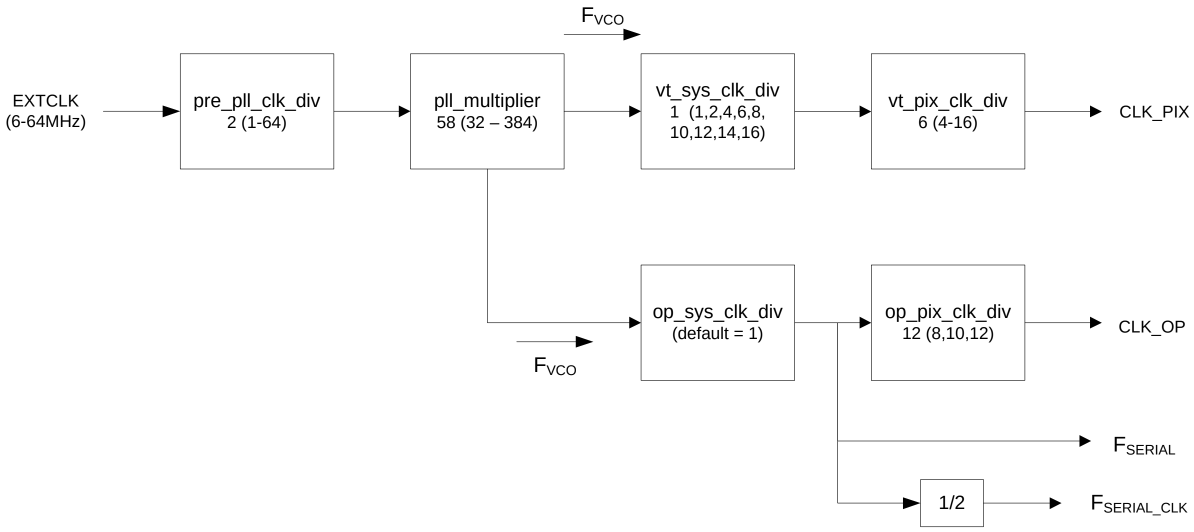
Parameter	Symbol	Min	Max	Unit
External Clock	EXTCLK	6	64	MHz
VCO Clock	F_{VCO}	384	768	MHz
Readout Clock	CLK_PIX		37.125	Mpixel/s
Output Clock	CLK_OP		74.25	Mpixel/s

Table 13: Example PLL Configuration for the Parallel Interface

Parameter	Value	Output
F_{VCO}		445.5 MHz (Max)
vt_sys_clk_div	1	
vt_pix_clk_div	6	
CLK_PIX		37.125 MPixel/s (= 445.5MHz / 12)
CLK_OP		74.25 MPixel/s (= 445.5MHz / 6)
Output pixel rate		74.25 MPixel/s

Serial PLL Configuration

Figure 24: PLL for the Serial Interface



The sensor will use op_sys_clk_div and op_pix_clk_div to configure the output clock per lane (CLK_OP). The configuration will depend on the number of active lanes (1, 2, or 4) configured. To configure the sensor protocol and number of lanes, refer to “Serial Configuration” on page 23.

Table 14: PLL Parameters for the Serial Interface

Parameter	Symbol	Min	Max	Unit
External Clock	EXTCLK	6	64	MHz
VCO Clock	F_{VCO}	384	768	MHz
Readout Clock	CLK_PIX		74.25	Mpixel/s
Output Clock	CLK_OP		37.125	Mpixel/s
Output Serial Data Rate Per Lane	F_{SERIAL}	300 (HiSPi)	700 (HiSPi)	Mbps
Output Serial Clock Speed Per Lane	F_{SERIAL_CLK}	150 (HiSPi)	350(HiSPi)	MHz

Configure the serial output so that it adheres to the following rules:

- The maximum data-rate per lane (F_{SERIAL}) is 700Mbps/lane (HiSPi).
- Configure the output pixel rate per lane (CLK_OP) so that the sensor output pixel rate matches the peak pixel rate (2 x CLK_PIX).
  - 4-lane: 4 x CLK_OP = 2 x CLK_PIX = Pixel Rate (max: 148.5 Mpixel/s)
  - 2-lane: 2 x CLK_OP = 2 x CLK_PIX = Pixel Rate (max: 74.25 Mpixel/s)
  - 1-lane: 1 x CLK_OP = 2 x CLK_PIX = Pixel Rate (max: 37.125 Mpixel/s)

Table 15: Example PLL Configurations for the Serial Interface

Parameter	4-lane			2-lane		1-lane		Units
	16-bit	12-bit	10-bit	12-bit	10-bit	12-bit	10-bit	
F_{VCO}	594	445.5	742.5	445.5	742.5	768	742.5	MHz
vt_sys_clk_div	1	1	2	1	2	2	2	
vt_pix_clk_div	8	6	5	12	10	12	10	
op_sys_clk_div	1	1	2	1	2	1	1	
op_pix_clk_div	16	12	10	12	10	12	10	
F_{SERIAL}	594	445.5	742.5	445.5	742.5	768	742.5	MHz
F_{SERIAL_CLK}	297	222.75	371.25	222.75	371.25	384	376.25	MHz
CLK_PIX	74.25	74.25	74.25	37.125	37.125	64	37.125	Mpixel/s
CLK_OP	37.125	37.125	37.125	37.125	37.125	32	37.125	Mpixel/s
Pixel Rate	148.5	148.5	148.5	74.25	74.25	64	74.25	Mpixel/s

Stream/Standby Control

The sensor supports a Standby mode. In this mode, external clock can be optionally disabled to further minimize power consumption. If this is done, then the “Power-Up Sequence” on page 67 must be followed.

Soft Standby

-Soft Standby is a low-power state that is controlled through register R0x301A[2]. Depending on the value of R0x301A[4], the sensor will go to Standby after completion of the current frame readout (default behavior) or after the completion of the current row readout. When the sensor comes back from Soft Standby, previously written register settings are still maintained. Soft Standby will not occur if Trigger pin is held high.

A specific sequence needs to be followed to enter and exit from Soft Standby.

Entering Soft Standby:

1. Set R0x301A[12] = 1 if serial mode was used

2. Set R0x301A[2] = 0 and drive Trigger pin low.
3. Turn off external clock to further minimize power consumption

Exiting Soft Standby:

1. Enable external clock if it was turned off
2. Set R0x301A[2] = 1 or drive Trigger pin high.
3. Set R0x301A[12] = 0 if serial mode is used

Sensor Readout

Image Acquisition Modes

The AR0331 supports two image acquisition modes:

1. Electronic rolling shutter (ERS) mode

This is the normal mode of operation. When the AR0331 is streaming; it generates frames at a fixed rate, and each frame is integrated (exposed) using the ERS. When the ERS is in use, timing and control logic within the sensor sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and subsequently reading that row, the pixels in the row integrate incident light. The integration (exposure) time is controlled by varying the time between row reset and row readout. For each row in a frame, the time between row reset and row readout is the same, leading to a uniform integration time across the frame. When the integration time is changed (by using the two-wire serial interface to change register settings), the timing and control logic controls the transition from old to new integration time in such a way that the stream of output frames from the AR0331 switches cleanly from the old integration time to the new while only generating frames with uniform integration. See “Changes to Integration Time” in the AR0331 Register Reference.

2. Global reset mode

This mode can be used to acquire a single image at the current resolution. In this mode, the end point of the pixel integration time is controlled by an external electromechanical shutter, and the AR0331 provides control signals to interface to that shutter.

The benefit of using an external electromechanical shutter is that it eliminates the visual artifacts associated with ERS operation. Visual artifacts arise in ERS operation, particularly at low frame rates, because an ERS image effectively integrates each row of the pixel array at a different point in time.

Window Control

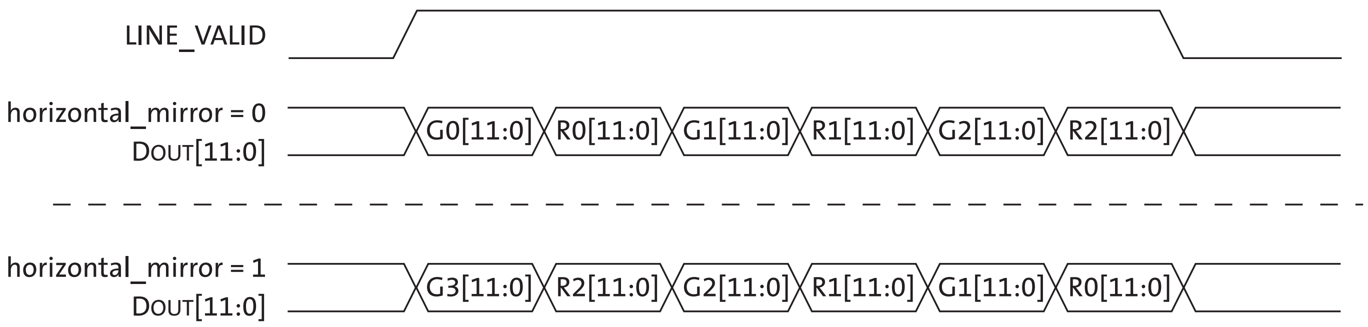
The sequencing of the pixel array is controlled by the x_addr_start, y_addr_start, x_addr_end, and y_addr_end registers.

Readout Modes

Horizontal Mirror

When the horizontal_mirror bit (R0x3040[14]) is set in the image_orientation register, the order of pixel readout within a row is reversed, so that readout starts from *x_addr_end* + 1and ends at *x_addr_start*. Figure 25 on page 35 shows a sequence of 6 pixels being read out with R0x3040[14] = 0 and R0x3040[14] = 1. Changing R0x3040[14] causes the Bayer order of the output image to change; the new Bayer order is reflected in the value of the pixel_order register.

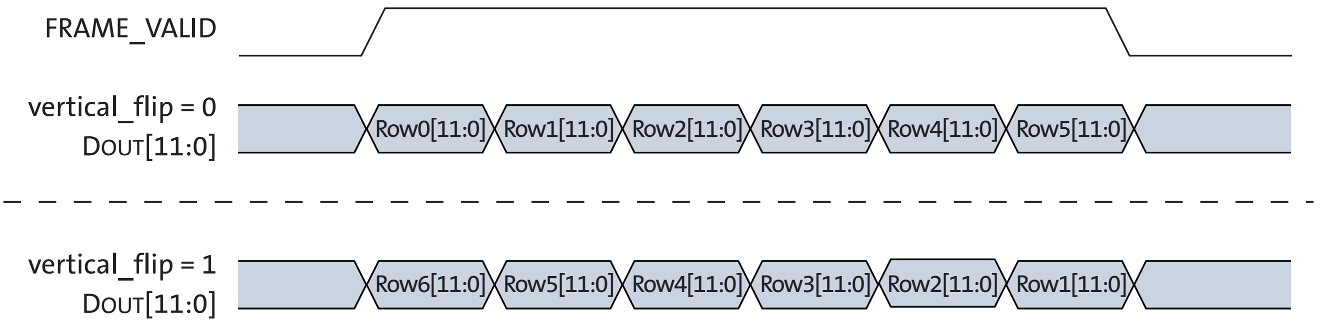
Figure 25: Effect of Horizontal Mirror on Readout Order



Vertical Flip

When the vertical_flip bit (R0x3040[15]) is set in the image_orientation register, the order in which pixel rows are read out is reversed, so that row readout starts from *y_addr_end* and ends at *y_addr_start*. Figure 30 shows a sequence of 6 rows being read out with R0x3040[15] = 0 and R0x3040[15] = 1. Changing this bit causes the Bayer order of the output image to change; the new Bayer order is reflected in the value of the pixel_order register.

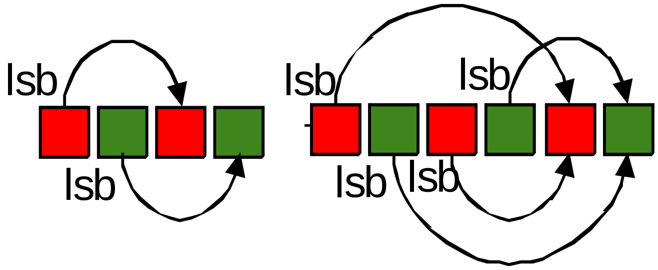
Figure 26: Effect of Vertical Flip on Readout Order



Subsampling

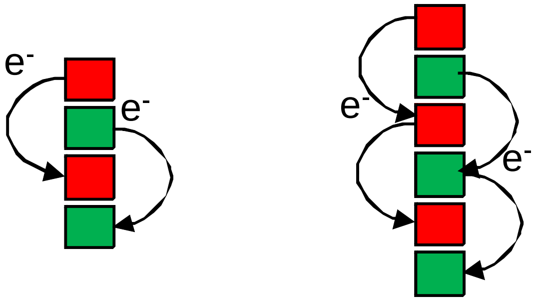
The AR0331 supports subsampling. Subsampling allows the sensor to read out a smaller set of active pixels by either skipping or binning pixels within the readout window. The working modes described in the data sheet that use subsampling are configured to use either 2x2 or 3x3 subsampling.

Figure 27: Horizontal Binning in the AR0331 Sensor



Horizontal binning is achieved either in the pixel readout or the digital readout. The sensor will sample the combined 2x or 3x adjacent pixels within the same color plane.

Figure 28: Vertical Row Binning in the AR0331 Sensor



Vertical row binning is applied in the pixel readout. Row binning can be configured of 2x or 3x rows within the same color plane.

Pixel skipping can be configured up to 2x and 3x in both the x-direction and y-direction. Skipping pixels in the x-direction will not reduce the row time. Skipping pixels in the y-direction will reduce the number of rows from the sensor effectively reducing the frame time. Skipping will introduce image artifacts from aliasing.

Table 16: Available Skip and Bin Modes in the AR0331 Sensor

Subsampling Method	Horizontal	Vertical
Skipping	2x, 3x	2x, 3x
Binning	2x, 3x,	2x, 3x

The sensor increments its x and y address based on the x_odd_inc and y_odd_inc value. The value indicates the addresses that are skipped after each pair of pixels or rows has been read.

The sensor will increment x and y addresses in multiples of 2. This indicates that a GreenR and Red pixel pair will be read together. As well, that the sensor will read a Gr-R row first followed by a B-Gb row.

$$x\ subsampling\ factor = \frac{1 + x\_odd\_inc}{2} \tag{EQ 5}$$

$$y\ subsampling\ factor = \frac{1 + y\_odd\_inc}{2} \tag{EQ 6}$$

A value of 1 is used for *x_odd_inc* and *y_odd_inc* when no pixel subsampling is indicated. In this case, the sensor is incrementing x and y addresses by 1 + 1 so that it reads consecutive pixel and row pairs. To implement a 2x skip in the x direction, the *x_odd*_inc is set to 3 so that the x address increment is 1+3, meaning that sensor will skip every other Gr-R pair.

$$\frac{\text{x\_addr\_end} - (\text{x\_addr\_start} + 1)}{} = \text{even number}$$

Table 17: Configuration for Horizontal Subsampling

	x_odd_inc	Restrictions
No subsampling	x_odd_inc = 1 skip = (1+1)*0.5 = 1x	The horizontal FOV must be programmed to meet the following rule: $\frac{\text{x\_addr\_end} - \text{x\_addr\_start} + 1}{(\text{x\_odd\_inc} + 1)/2} = \text{even number}$
Skip 2x	x_odd_inc = 3 skip = (1+3)*0.5 = 2x	
Skip 3x	x_odd_inc = 5 skip = (1+5)*0.5 = 3x	
Analog Bin 2x	x_odd_inc = 3 skip = (1+3)*0.5 =2x col_sf_bin_en = 1	
Analog Bin 3x	x_odd_inc = 5 skip = (1+5)*0.5 = 3x col_sf_bin_en = 1	
Digital Bin 2x	x_odd_inc = 3 skip = (1+3)*0.5 =2x col_bin =1	
Digital Bin 3x	x_odd_inc = 5 skip = (1+5)*0.5 = 3x col_bin = 1	

Table 18: Configuration for Vertical Subsampling

	y_odd_inc	**Restrictions:**
No subsampling	y_odd_inc = 1 skip = (1+1)*0.5 = 1x row_bin = 0	The vertical FOV must be programmed to meet the following rule: $\frac{y\_addr\_end - y\_addr\_start + 1}{(y\_odd\_inc + 1)/2}$ = even number
Skip 2x	y_odd_inc = 3 skip = (1+3)*0.5 =2x row_bin = 0	
Skip 3x	y_odd_inc = 5 skip = (1+5)*0.5 =3x row_bin = 0	
Analog Bin 2x	y_odd_inc = 3 skip = (1+3)*0.5 =2x row_bin = 1	
Analog Bin 3x	y_odd_inc = 5 skip = (1+5)*0.5 =3x row_bin = 1	

Sensor Frame Rate

The time required to read out an image frame (T_{FRAME}) can be derived from the number of clocks required to output each image and the pixel clock.

The frame-rate is the inverse of the frame period.

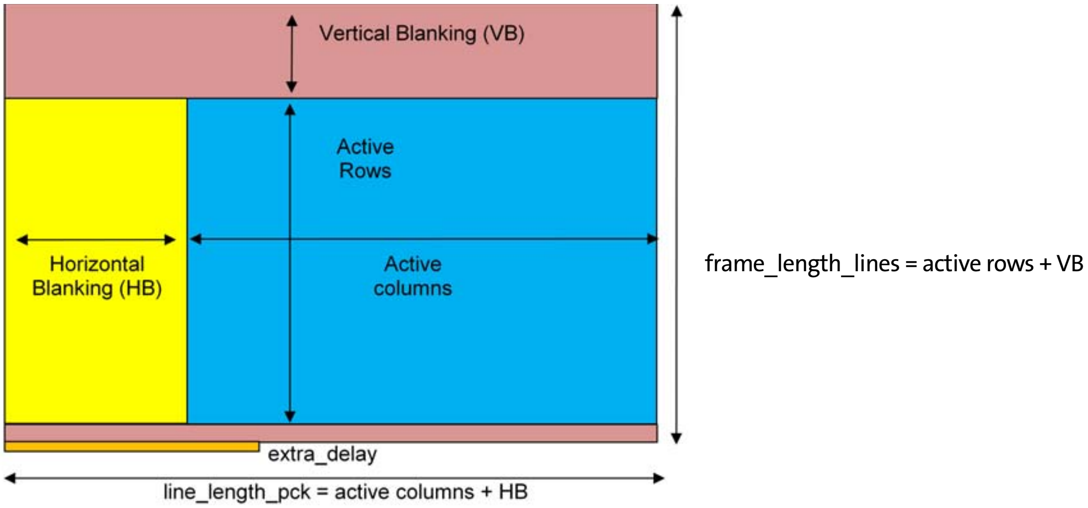
$$fps = 1/T_{FRAME} \quad \text{(EQ 7)}$$

The number of clocks can be simplified further into the following parameters:

- The number of clocks required for each sensor row (*line_length_pck*)
  This parameter also determines the sensor row period when referenced to the sensor readout clock. (T_{ROW} = line_length_pck x 1/CLK_PIX)
- The number of row periods per frame (*frame_length_lines*)
- An extra delay between frames used to achieve a specific output frame period (*extra_delay*)

$$T_{FRAME} = 1/(CLK\_PIX) \times [frame\_length\_lines \times line\_length\_pck + extra\_delay] \quad \text{(EQ 8)}$$

Figure 29: Frame Period Measured in Clocks



Row Period (T_{ROW})

The *line_length_pck* will determine the number of clock periods per row and the row period (T_{ROW}) when combined with the sensor readout clock. The *line_length_pck* includes both the active pixels and the horizontal blanking time per row. The sensor utilizes two readout paths, as seen in Figure 22 on page 31, allowing the sensor to output two pixels during each pixel clock.

The minimum *line_length_pck* is defined as the maximum of the following three equations:

ADC Readout Limitation:

$$line\_length\_pck \geq 1100 \qquad \text{(EQ 9)}$$

line_length_pck >= 1100

Digital Readout Limitation:

$$\frac{1}{3} \times \left[\frac{x\_addr\_end - x\_addr\_start}{(x\_odd\_inc + 1) \times 0.5}\right] \qquad \text{(EQ 10)}$$

Output Interface Limitations:

$$\frac{1}{2} \times \left[\frac{x\_addr\_end - x\_addr\_start}{(x\_odd\_inc + 1) \times 0.5}\right] + 96 \qquad \text{(EQ 11)}$$

Row Periods Per Frame

The *frame_length_lines* determines the number of row periods (T_{ROW}) per frame. This includes both the active and blanking rows. The *minimum_vertical_blanking* value is defined by the number of OB rows read per frame, two embedded data rows, and two blank rows.

$$Minimum\ frame\_length\_lines = \frac{y\_addr\_end - y\_addr\_start}{(y\_odd\_inc + 1)/2} + min\_frame\_length\_lines \qquad \text{(EQ 12)}$$

The sensor is configured to output frame information in two embedded data rows by setting R0x3064[8] to 1 (default). If R0x3064[8] is set to 0, the sensor will instead output two blank rows. The data configured in the two embedded rows is defined in “Embedded Data and Statistic” on page 48.

Table 19: Minimum Vertical Blanking Configuration

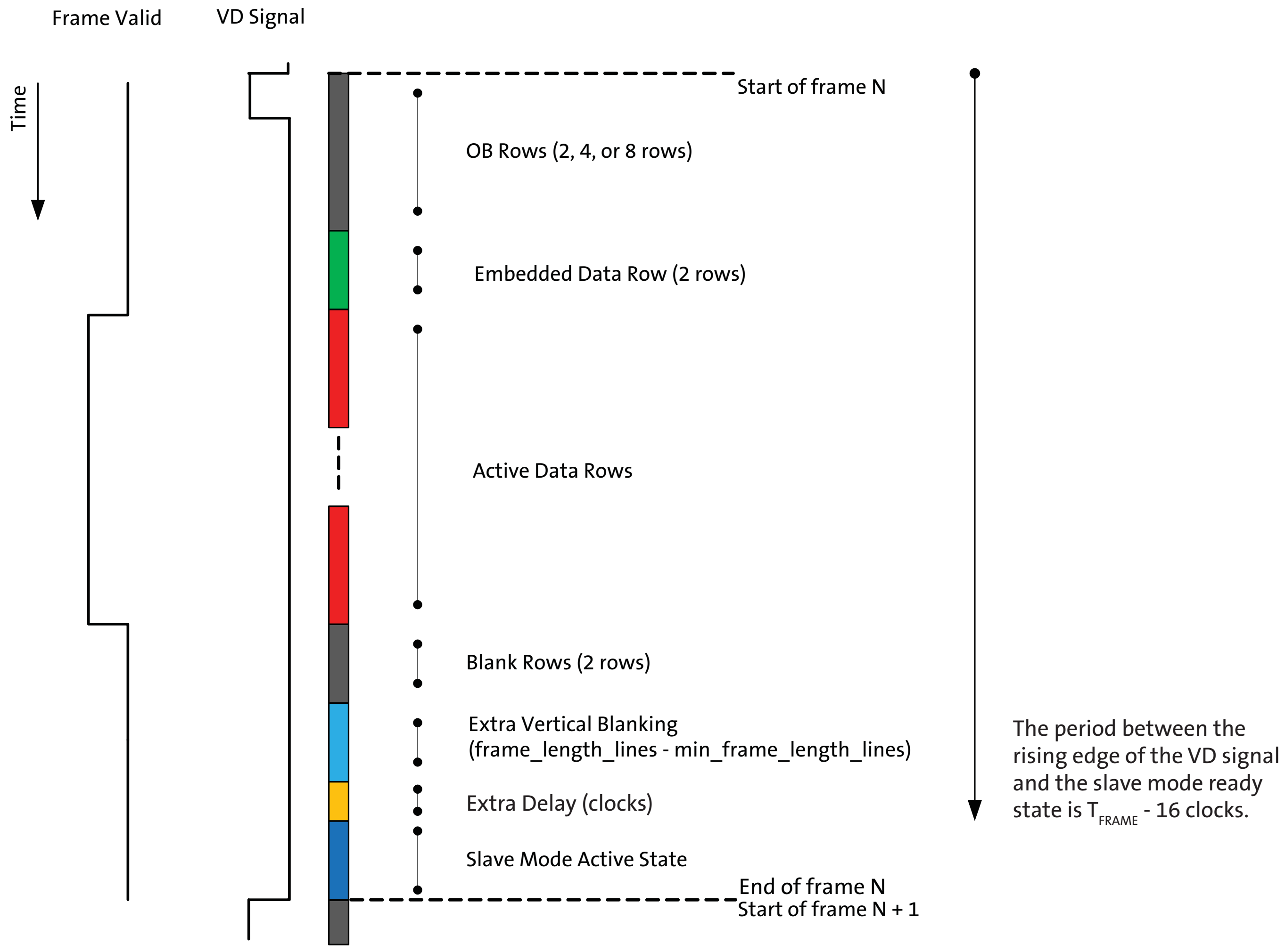
R0x3180[0x00F0]	OB Rows	min_frame_length_lines
0x8 (Default)	8 OB Rows	8 OB + 10 = 18
0x4	4 OB Rows	4 OB + 10 = 14
0x2	2 OB Rows	2 OB + 10 = 12

The locations of the OB rows, embedded rows, and blank rows within the frame readout are identified in Figure 30: “Slave Mode Active State and Vertical Blanking,” on page 41.

Slave Mode

The slave mode feature of the AR0331 supports triggering the start of a frame readout from a VD signal that is supplied from an external ASIC. The slave mode signal allows for precise control of frame rate and register change updates. The VD signal is input to the trigger pin.

Figure 30: Slave Mode Active State and Vertical Blanking

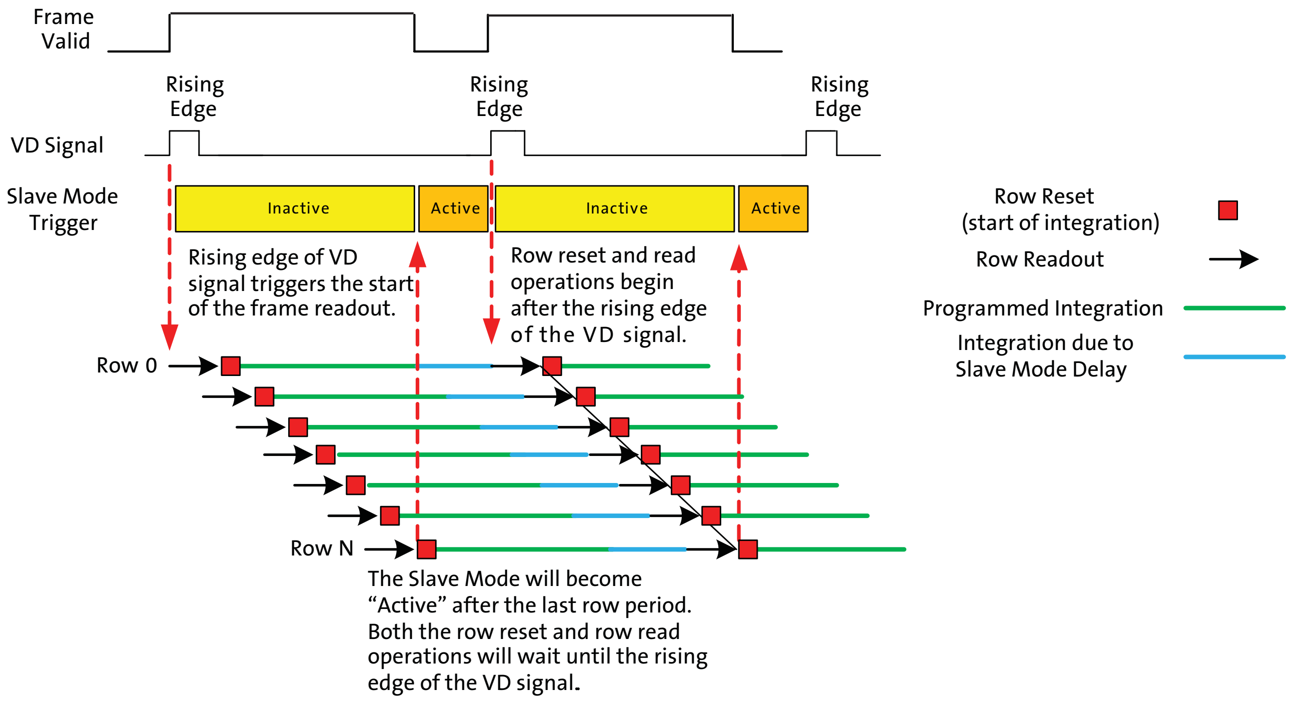


If the slave mode is disabled, the new frame will begin after the extra delay period is finished.

The slave mode will react to the rising edge of the input VD signal if it is in an active state. When the VD signal is received, the sensor will begin the frame readout and the slave mode will remain inactive for the period of one frame time minus 16 clock periods (T_{FRAME} - (16 / CLK_PIX)). After this period, the slave mode will re-enter the active state and will respond to the VD signal.

Figure 31: Slave Mode Example with Equal Integration and Frame Readout Periods

The integration of the last row is therefore started before the end of the programmed integration for the first row.

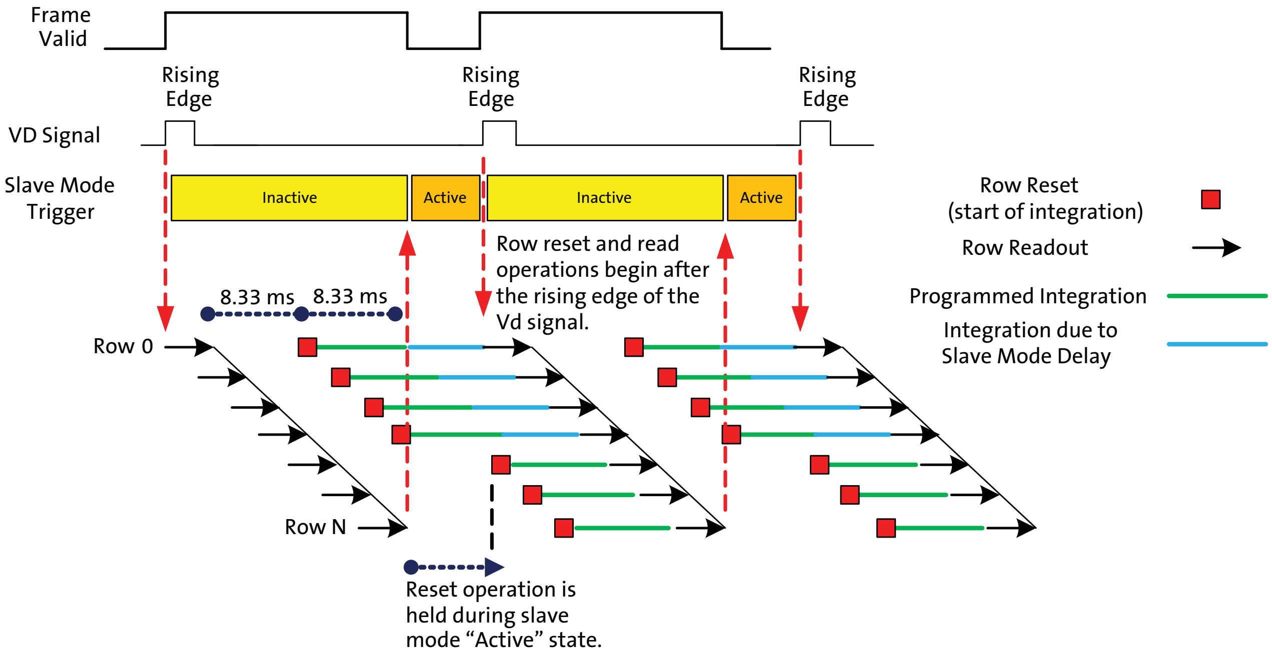


The row shutter and read operations will stop when the slave mode becomes active and is waiting for the VD signal. The following should be considered when configuring the sensor to use the slave mode:

1. The frame period (T_{FRAME}) should be configured to be less than the period of the input VD signal. The sensor will disregard the input VD signal if it appears before the frame readout is finished.
2. If the sensor integration time is configured to be less than the frame period, then the sensor will not have reset all of the sensor rows before it begins waiting for the input VD signal. This error can be minimized by configuring the frame period to be as close as possible to the desired frame rate (period between VD signals).

Figure 32: Slave Mode Example Where the Integration Period is Half of the Frame Readout Period

The sensor read pointer will have paused at row 0 while the shutter pointer pauses at row N/2. The extra integration caused by the slave mode delay will only be seen by rows 0 to N/2. The example below is for a frame readout period of 16.6ms while the integration time is configured to 8.33ms.



When the slave mode becomes active, the sensor will pause both row read and row reset operations. (Note: The row integration period is defined as the period from row reset to row read.) The frame-time should therefore be configured so that the slave mode "wait period" is as short as possible. In the case where the sensor integration time is shorter than the frame time, the "wait period" will only increase the integration of the rows that have been reset following the last VD pulse.

The period between slave mode pulses must also be greater than the frame period. If the rising edge of the VD pulse arrives while the slave mode is inactive, the VD pulse will be ignored and will wait until the next VD pulse has arrived.

Frame Readout

The sensor readout begins with vertical blanking rows followed by the active rows. The frame readout period can be defined by the number of row periods within a frame (*frame_length_lines*) and the row period (*line_length_pck/clk_pix*). The sensor will read the first vertical blanking row at the beginning of the frame period and the last active row at the end of the row period.

Figure 33: Example of the Sensor Output of a 1920 x 1080 Frame at 60 fps

The frame valid and line valid signals mentioned in this diagram represent internal signals within the sensor. The SYNC codes represented in this diagram represent the HiSPi Streaming SP protocol.

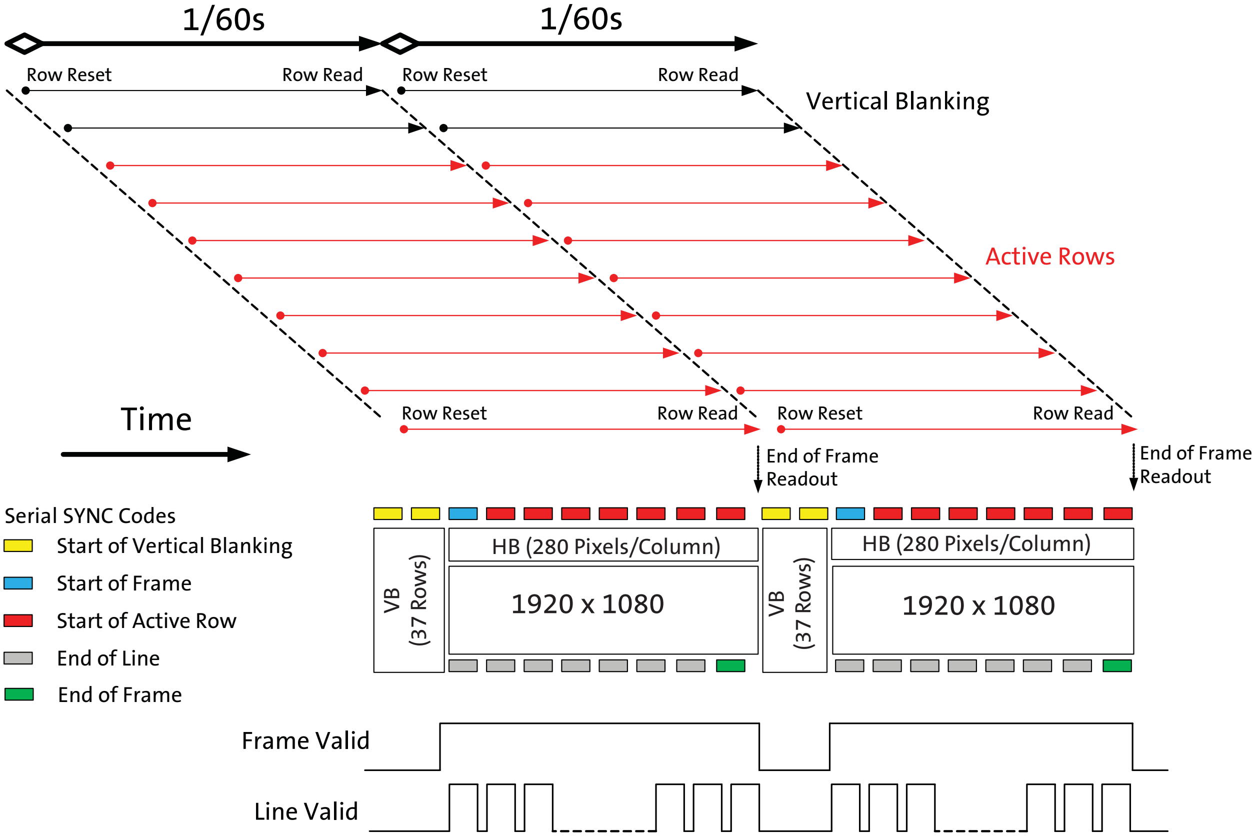


Figure 33 aligns the frame integration and readout operation to the sensor output. It also shows the sensor output using the HiSPi Streaming SP protocol. Different sensor protocols will list different SYNC codes.

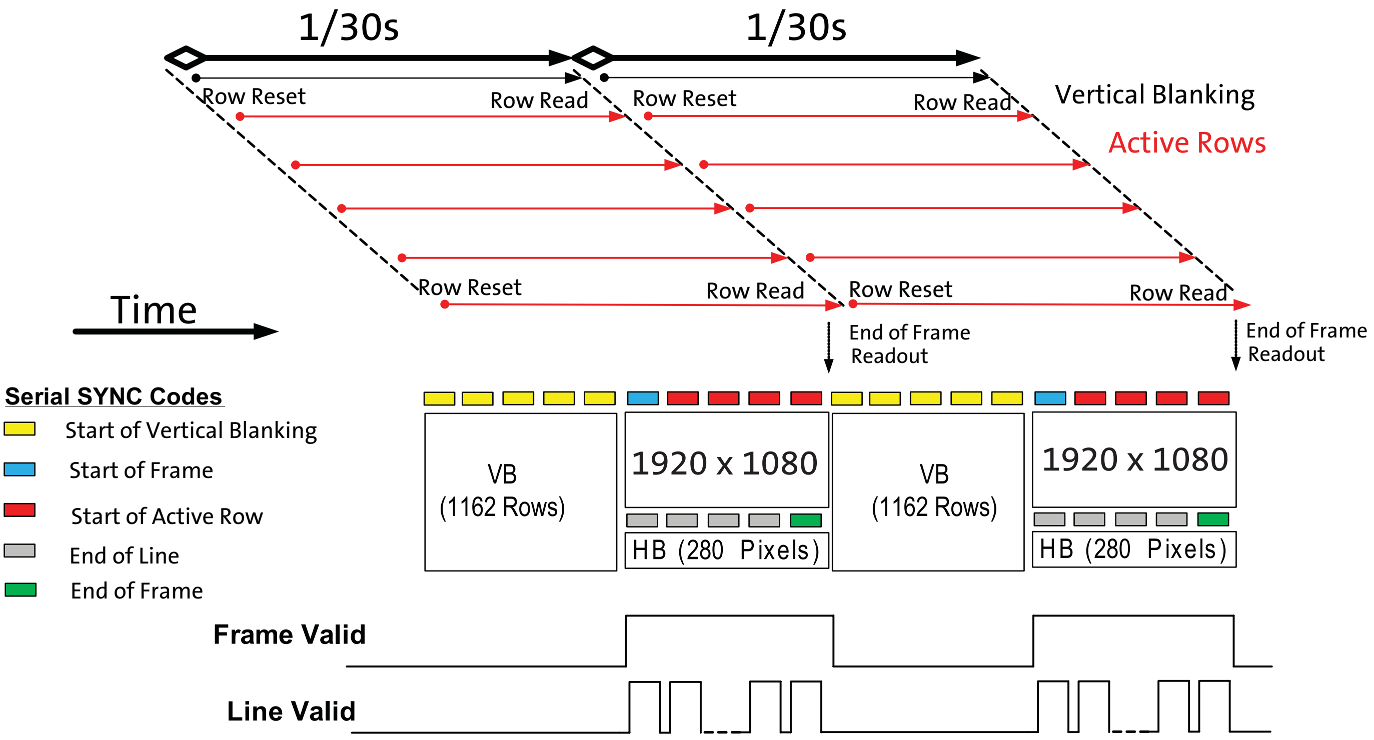
Table 20: Serial SYNC Codes Included with Each Protocol Included with the AR0331 Sensor

Interface/Protocol	Start of Vertical Blanking Row (SOV)	Start of Frame (SOF)	Start of Active Line (SOA)	End of Line (EOL)	End of Frame (EOF)
Parallel	Parallel interface uses FRAME VALID (FV) and LINE VALID (LV) outputs to denote start and end of line and frame.				
HiSPi Streaming S	Yes	Send SOV	Yes	No SYNC Code	No SYNC Code
HiSPi Streaming SP	Yes	Yes	Yes	Yes	Yes
HiSPi Packetized SP	No SYNC Code	Yes	Yes	Yes	Yes

Figure 34 illustrates how the sensor active readout time can be minimized while reducing the frame rate. 1125 VB rows were added to the output frame to reduce the 1920 x1080 frame rate from 60 fps to 30 fps without increasing the delay between the readout of the first and last active row.

Figure 34: Example of the Sensor Output of a 1920 x1080 Frame at 30 fps

The frame valid and line valid signals mentioned in this diagram represent internal signals within the sensor. The SYNC codes represented in this diagram represent the HiSPi Streaming SP protocol.



Changing Sensor Modes

Register Changes

All register writes are delayed by 1x frame. A register that is written to during the readout of frame *n* will not be updated to the new value until the readout of frame *n+2*. This includes writes to the sensor gain and integration registers.

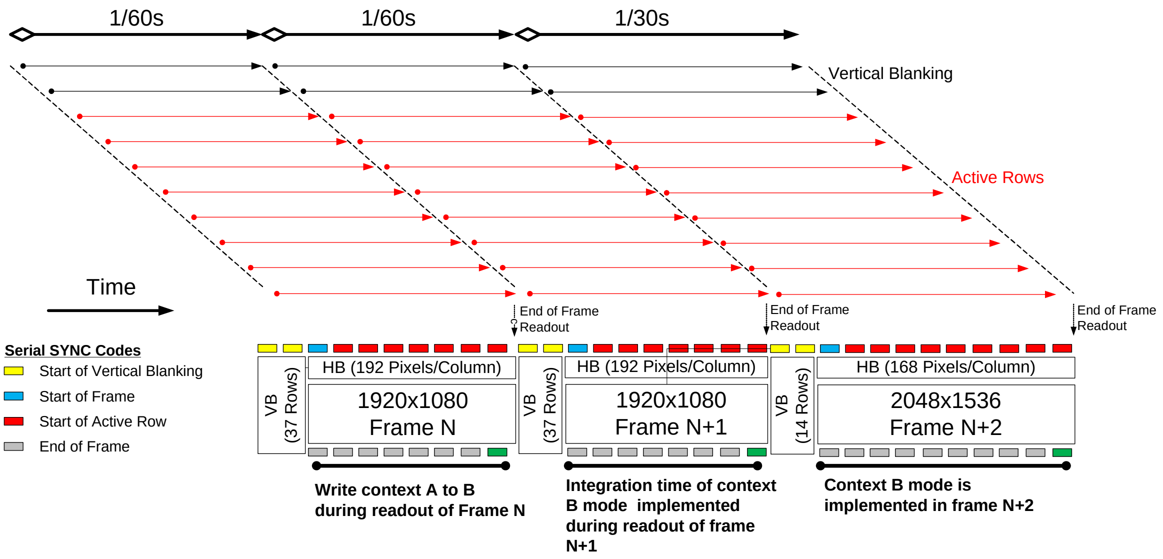
Real-Time Context Switching

In the AR0331, the user may switch between two full register sets A and B by writing to a context switch change bit in R0x30B0[13]. When the context switch is configured to context A the sensor will reference the context A registers. If the context switch is changed from A to B during the readout of frame *n*, the sensor will then reference the context B *coarse_integration_time* registers in frame *n+1* and all other context B registers at the beginning of reading frame *n+2*. The sensor will show the same behavior when changing from context B to context A.

Table 21: List of Configurable Registers for Context A and Context B

Context A		Context B	
Register Description	**Address**	**Register Description**	**Address**
coarse_integration_time	0x3012	coarse_integration_time_CB	0x3016
fine_integration_time	0x3014	fine_integration_time_CB	0x3018
line_length_pck	0x300C	line_length_pck_CB	0x303E
frame_length_lines	0x300A	frame_length_lines_CB	0x30AA
COL_SF_BIN_EN	0x3040[9]	COL_SF_BIN_EN_CB	0x3040[8]
ROW_BIN	0x3040[12]	ROW_BIN_CB	0x3040[10]
COL_BIN	0x3040[13]	COL_BIN_CB	0x3040[11]
FINE_GAIN	0x3060[3:0]	FINE_GAIN_CB	0x3060[11:8]
COARSE_GAIN	0x3060[5:4]	COARSE_GAIN_CB	0x3060[13:12]
x_addr_start	0x3004	x_addr_start_CB	0x308A
y_addr_start	0x3002	y_addr_start_CB	0x308C
x_addr_end	0x3008	x_addr_end_CB	0x308E
y_addr_end	0x3006	y_addr_end_CB	0x3090
Y_odd_inc	0x30A6	Y_odd_inc_CB	0x30A8
X_odd_inc	0x30A2	X_odd_inc_CB	0x30AE
GREEN1_GAIN	0x3056	GREEN1_GAIN_CB	0x30BC
BLUE_GAIN	0x3058	BLUE_GAIN_CB	0x30BE
RED_GAIN	0x305A	RED_GAIN_CB	0x30C0
GREEN2_GAIN	0x305C	GREEN2_GAIN_CB	0x30C2
GLOBAL_GAIN	0x305E	GLOBAL_GAIN_CB	0x30C4
operation_mode_ctrl	0x3082	operation_mode_ctrl_CB	0x3084
bypass_pix_comb	0x318E[13:12]	bypass_pix_comb_CB	0x318E[15:14]

Figure 35: Example of Changing the Sensor from Context A to Context B



Compression

When the AR0331 is configured for linear mode operation, the sensor can optionally compress 12-bit data to 10-bit using A-law compression. The compression is applied after the data pedestal has been added to the data. See “Data Pedestal” on page 27.

The A-law compression is disabled by default and can be enabled by setting R0x31D0 from “0” to “1”.

Table 22: A-Law Compression Table for 12-10 bits

Input Range	Input Values												Compressed Codeword									
	11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
0 to 127	0	0	0	0	0	a	b	c	d	e	f	g	0	0	0	a	b	c	d	e	f	g
128 to 255	0	0	0	0	1	a	b	c	d	e	f	g	0	0	1	a	b	c	d	e	f	g
256 to 511	0	0	0	1	a	b	c	d	e	f	g	X	0	1	0	a	b	c	d	e	f	g
512 to 1023	0	0	1	a	b	c	d	e	f	g	X	X	0	1	1	a	b	c	d	e	f	g
1024 to 2047	0	1	a	b	c	d	e	f	g	h	X	X	1	0	a	b	c	d	e	f	g	h
2048 to 4095	1	a	b	c	d	e	f	g	h	X	X	X	1	1	a	b	c	d	e	f	g	h

Temperature Sensor

The AR0331 sensor has a built-in PTAT-based temperature sensor, accessible through registers, that is capable of measuring die junction temperature.

The temperature sensor can be enabled by writing R0x30B4[0]=1 and R0x30B4[4]=1. After this, the temperature sensor output value can be read from R0x30B2[10:0].

The value read out from the temperature sensor register is an ADC output value that needs to be converted downstream to a final temperature value in degrees Celsius. Since the PTAT device characteristic response is quite linear in the temperature range of operation required, a simple linear function in the format of listed in the equation below can be used to convert the ADC output value to the final temperature in degrees Celsius.

$$Temperature = slope \times R0x30B2[10:0] + T_0 \qquad \text{(EQ 13)}$$

For this conversion, a minimum of two known points are needed to construct the line formula by identifying the slope and y-intercept "T_0". These calibration values can be read from registers R0x30C6 and R0x30C8, which correspond to value read at 70°C and 55°C respectively. Once read, the slope and y-intercept values can be calculated and used in Equation 13.

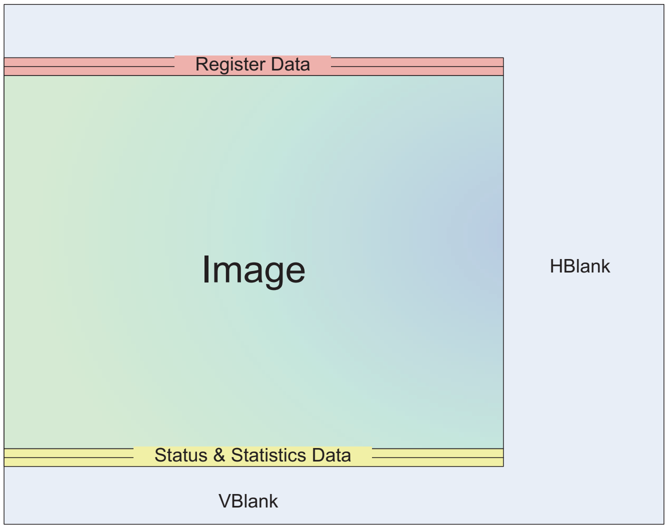
For more information on the temperature sensor registers, refer to the AR0331 Register Reference.

Embedded Data and Statistic

The AR0331 has the capability to output image data and statistics embedded within the frame timing. There are two types of information embedded within the frame readout.

1. Embedded Data: If enabled, these are displayed on the 2 rows immediately before the first active pixel row is displayed.
2. Embedded Statistics: If enabled, these are displayed on the 2 rows immediately after the last active pixel row is displayed.

Figure 36: Frame Format with Embedded Data Lines Enabled



Embedded Data

The embedded data contains the configuration of the image being displayed. This includes all register settings used to capture the current frame. The registers embedded in these rows are as follows:

Line 1: Registers R0x3000 to R0x312F

Line 2: Registers R0x3136 to R0x31BF, R0x31D0 to R0x31FF

Note: All undefined registers will have a value of 0.

The format of the embedded register data transmission is defined per the embedded data section of the SMIA Function Specification.

In parallel mode, since the pixel word depth is 12 bits/pixel, the sensor 16-bit register data will be transferred over 2 pixels where the register data will be broken up into 8 MSB and 8 LSB. The alignment of the 8-bit data will be on the 8 MSB bits of the 12-bit pixel word. For example, if a register value of 0x1234 is to be transmitted, it will be transmitted over two, 12-bit pixels as follows: 0x120, 0x340.

Embedded Statistics

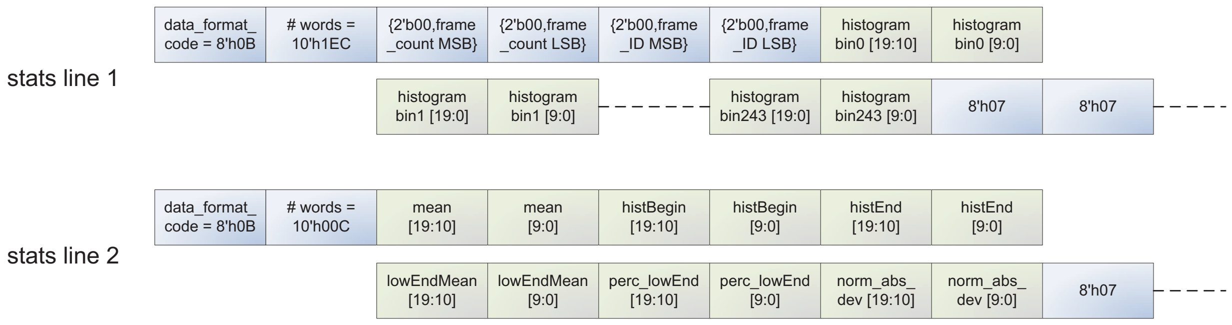
The embedded statistics contain frame identifiers and histogram information of the image in the frame. This can be used by downstream auto-exposure algorithm blocks to make decisions about exposure adjustment.

This histogram is divided into 244 bins with a bin spacing of 64 evenly spaced bins for digital code values 0 to 2^8, 120 evenly spaced bins for values 2^8 to 2^{12}, 60 evenly spaced bins for values 2^{12} to 2^{16}. In HDR with a 16x exposure ratio, this approximately corresponds to the T1 and T2 exposures respectively.

The first pixel of each line in the embedded statistics is a tag value of 0x0B0. This signifies that all subsequent statistics data is 10 bit data aligned to the MSB of the 12-bit pixel.

Figure 37 summarizes how the embedded statistics transmission looks like. It should be noted that data, as shown in Figure 37, is aligned to the MSB of each word:

Figure 37: Format of Embedded Statistics Output within a Frame



The statistics embedded in these rows are as follows:

Line 1:

- 0x0B0 - identifier
- Register 0x303A - frame_count
- Register 0x31D2 - frame ID
- Histogram data - histogram bins 0-243

Line 2:

- 0x0B0 (TAG)
- Mean
- Histogram Begin
- Histogram End
- Low End Histogram Mean
- Percentage of Pixels Below Low End Mean
- Normal Absolute Deviation

Test Patterns

The AR0331 has the capability of injecting a number of test patterns into the top of the datapath to debug the digital logic. With one of the test patterns activated, any of the datapath functions can be enabled to exercise it in a deterministic fashion. Test patterns are selected by Test_Pattern_Mode register (R0x3070). Only one of the test patterns can be enabled at a given point in time by setting the Test_Pattern_Mode register according to Table 23. When test patterns are enabled the active area will receive the value specified by the selected test pattern and the dark pixels will receive the value in Test_Pattern_Green (R0x3074 and R0x3078) for green pixels, Test_Pattern_Blue (R0x3076) for blue pixels, and Test_Pattern_Red (R0x3072) for red pixels.

Note: Turn off black level calibration (BLC) when Test Pattern is enabled.

Table 23: Test Pattern Modes

Test_Pattern_Mode	Test Pattern Output
0	No test pattern (normal operation)
1	Solid color test pattern
2	100% Vertical Color Bars test pattern
3	Fade-to-Gray Vertical Color Bars test pattern
256	Walking 1s test pattern (12-bit)

Solid Color

When the color field mode is selected, the value for each pixel is determined by its color. Green pixels will receive the value in Test_Pattern_Green, red pixels will receive the value in Test_Pattern_Red, and blue pixels will receive the value in Test_Pattern_Blue.

Vertical Color Bars

When the vertical color bars mode is selected, a typical color bar pattern will be sent through the digital pipeline.

Walking 1s

When the walking 1s mode is selected, a walking 1s pattern will be sent through the digital pipeline. The first value in each row is 1.

Two-Wire Serial Register Interface

The two-wire serial interface bus enables read/write access to control and status registers within the AR0331. This interface is designed to be compatible with the electrical characteristics and transfer protocols of the two-wire serial interface specification.

The interface protocol uses a master/slave model in which a master controls one or more slave devices. The sensor acts as a slave device. The master generates a clock (SCLK) that is an input to the sensor and is used to synchronize transfers. Data is transferred between the master and the slave on a bidirectional signal (SDATA). SDATA is pulled up to VDD_IO off-chip by a 1.5kΩ resistor. Either the slave or master device can drive SDATA LOW—the interface protocol determines which device is allowed to drive SDATA at any given time.

The protocols described in the two-wire serial interface specification allow the slave device to drive SCLK LOW; the AR0331 uses SCLK as an input only and therefore never drives it LOW.

Protocol

Data transfers on the two-wire serial interface bus are performed by a sequence of low-level protocol elements:

1. a (repeated) start condition
2. a slave address/data direction byte
3. an (a no) acknowledge bit
4. a message byte
5. a stop condition

The bus is idle when both SCLK and SDATA are HIGH. Control of the bus is initiated with a start condition, and the bus is released with a stop condition. Only the master can generate the start and stop conditions.

Start Condition

A start condition is defined as a HIGH-to-LOW transition on SDATA while SCLK is HIGH. At the end of a transfer, the master can generate a start condition without previously generating a stop condition; this is known as a "repeated start" or "restart" condition.

Stop Condition

A stop condition is defined as a LOW-to-HIGH transition on SDATA while SCLK is HIGH.

Data Transfer

Data is transferred serially, 8 bits at a time, with the MSB transmitted first. Each byte of data is followed by an acknowledge bit or a no-acknowledge bit. This data transfer mechanism is used for the slave address/data direction byte and for message bytes.

One data bit is transferred during each SCLK clock period. SDATA can change when SCLK is LOW and must be stable while SCLK is HIGH.

Slave Address/Data Direction Byte

Bits [7:1] of this byte represent the device slave address and bit [0] indicates the data transfer direction. A "0" in bit [0] indicates a WRITE, and a "1" indicates a READ. The default slave addresses used by the AR0331 are 0x20 (write address) and 0x21 (read address) in accordance with the specification. Alternate slave addresses of 0x30 (write address) and 0x31 (read address) can be selected by enabling and asserting the SADDR input.

An alternate slave address can also be programmed through R0x31FC.

Message Byte

Message bytes are used for sending register addresses and register write data to the slave device and for retrieving register read data.

Acknowledge Bit

Each 8-bit data transfer is followed by an acknowledge bit or a no-acknowledge bit in the SCLK clock period following the data transfer. The transmitter (which is the master when writing, or the slave when reading) releases SDATA. The receiver indicates an acknowledge bit by driving SDATA LOW. As for data transfers, SDATA can change when SCLK is LOW and must be stable while SCLK is HIGH.

No-Acknowledge Bit

The no-acknowledge bit is generated when the receiver does not drive SDATA LOW during the SCLK clock period following a data transfer. A no-acknowledge bit is used to terminate a read sequence.

Typical Sequence

A typical READ or WRITE sequence begins by the master generating a start condition on the bus. After the start condition, the master sends the 8-bit slave address/data direction byte. The last bit indicates whether the request is for a read or a write, where a "0" indicates a write and a "1" indicates a read. If the address matches the address of the slave device, the slave device acknowledges receipt of the address by generating an acknowledge bit on the bus.

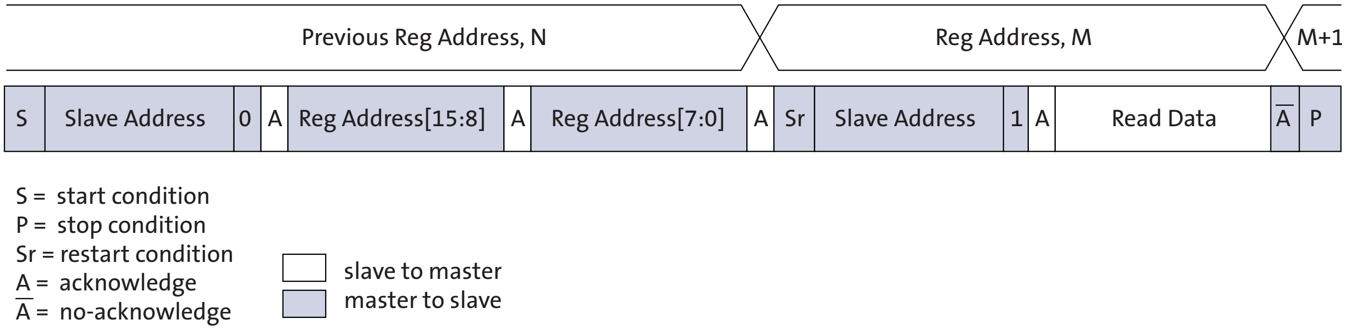
If the request was a WRITE, the master then transfers the 16-bit register address to which the WRITE should take place. This transfer takes place as two 8-bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master then transfers the data as an 8-bit sequence; the slave sends an acknowledge bit at the end of the sequence. The master stops writing by generating a (re)start or stop condition.

If the request was a READ, the master sends the 8-bit write slave address/data direction byte and 16-bit register address, the same way as with a WRITE request. The master then generates a (re)start condition and the 8-bit read slave address/data direction byte, and clocks out the register data, 8 bits at a time. The master generates an acknowledge bit after each 8-bit transfer. The slave's internal register address is automatically incremented after every 8 bits are transferred. The data transfer is stopped when the master sends a no-acknowledge bit.

Single READ from Random Location

This sequence (Figure 38) starts with a dummy WRITE to the 16-bit address that is to be used for the READ. The master terminates the WRITE by generating a restart condition. The master then sends the 8-bit read slave address/data direction byte and clocks out one byte of register data. The master terminates the READ by generating a no-acknowledge bit followed by a stop condition. Figure 38 shows how the internal register address maintained by the AR0331 is loaded and incremented as the sequence proceeds.

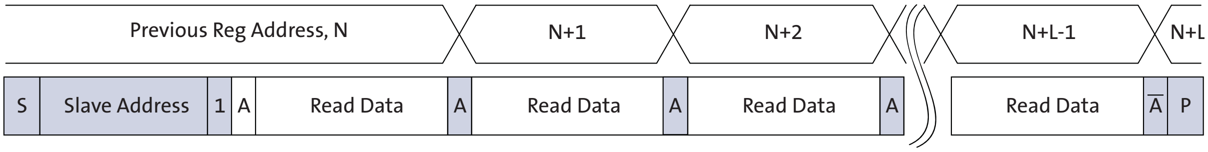
Figure 38: Single READ from Random Location



Single READ from Current Location

This sequence (Figure 39) performs a read using the current value of the AR0331 internal register address. The master terminates the READ by generating a no-acknowledge bit followed by a stop condition. The figure shows two independent READ sequences.

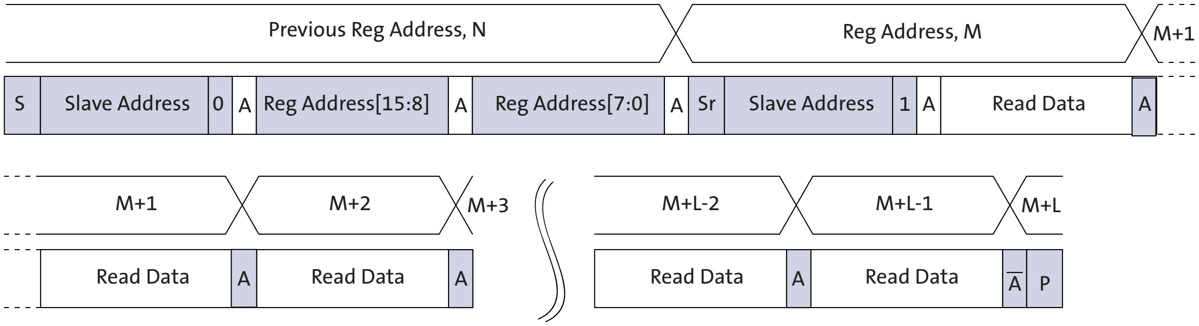
Figure 39: Single READ from Current Location



Sequential READ, Start from Random Location

This sequence (Figure 40) starts in the same way as the single READ from random location (Figure 38). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte READs until “L” bytes have been read.

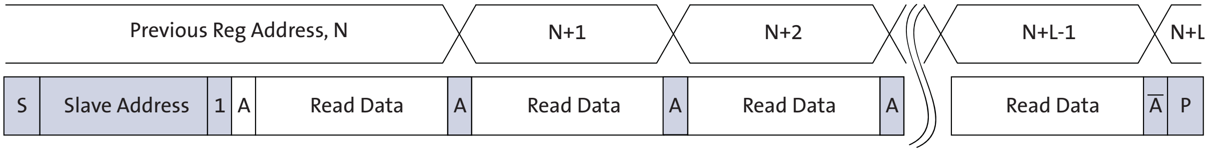
Figure 40: Sequential READ, Start from Random Location



Sequential READ, Start from Current Location

This sequence (Figure 41) starts in the same way as the single READ from current location (Figure 39). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte READs until "L" bytes have been read.

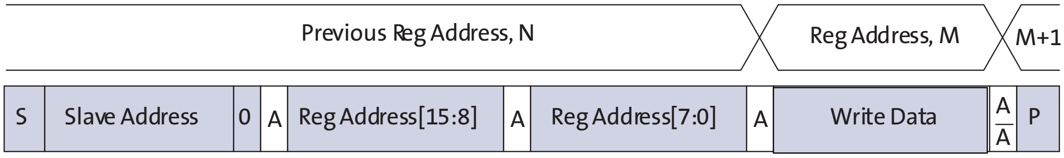
Figure 41: Sequential READ, Start from Current Location



Single WRITE to Random Location

This sequence (Figure 42) begins with the master generating a start condition. The slave address/data direction byte signals a WRITE and is followed by the HIGH then LOW bytes of the register address that is to be written. The master follows this with the byte of write data. The WRITE is terminated by the master generating a stop condition.

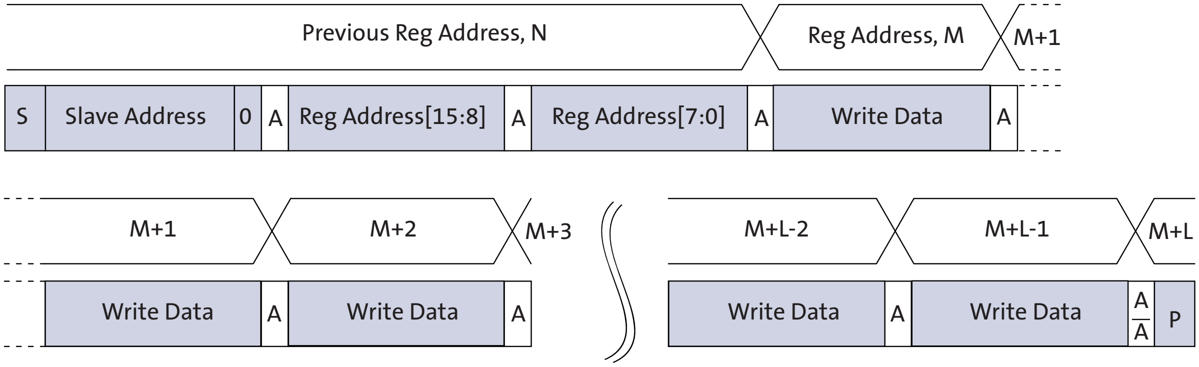
Figure 42: Single WRITE to Random Location



Sequential WRITE, Start at Random Location

This sequence (Figure 43) starts in the same way as the single WRITE to random location (Figure 42). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte WRITEs until "L" bytes have been written. The WRITE is terminated by the master generating a stop condition.

Figure 43: Sequential WRITE, Start at Random Location



Electrical Specifications

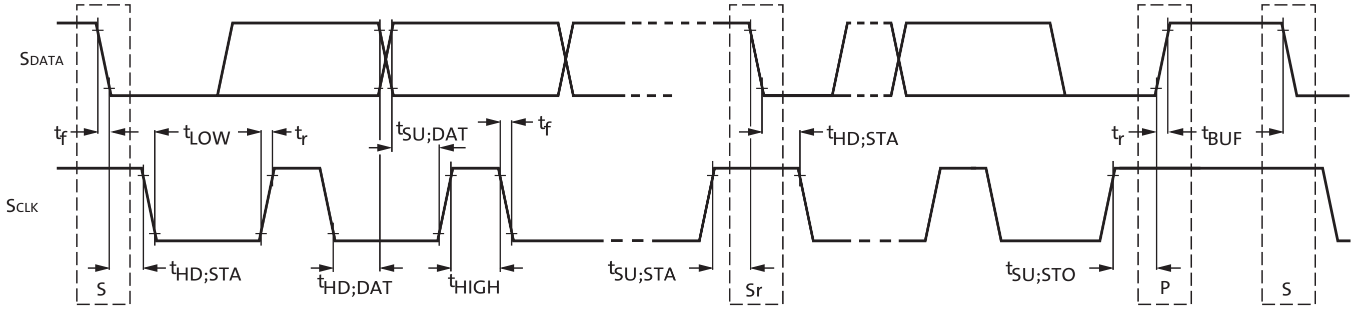
Unless otherwise stated, the following specifications apply to the following conditions:

VDD = 1.8V – 0.10/+0.15; VDD_IO = VDD_PLL = VAA = VAA_PIX = 2.8V ± 0.3V;
VDD_SLVS = 0.4V – 0.1/+0.2; T_A = -30°C to +85°C; output load = 10pF;
frequency = 74.25 MHz; HiSPi off.

Two-Wire Serial Register Interface

The electrical characteristics of the two-wire serial register interface (SCLK, SDATA) are shown in Figure 44 and Table 24.

Figure 44: Two-Wire Serial Bus Timing Parameters



Note: Read sequence: For an 8-bit READ, read waveforms start after WRITE command and register address are issued.

Table 24: Two-Wire Serial Bus Characteristics

^{f}EXTCLK = 27 MHz; VDD = 1.8V; VDD_IO = 2.8V; VAA = 2.8V; VAA_PIX = 2.8V;
VDD_PLL = 2.8V; VDD_DAC = 2.8V; T_A = 25°C

Parameter	Symbol	Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	
SCLK Clock Frequency	^{f}SCL	0	100	0	400	KHz
Hold time (repeated) START condition.						
After this period, the first clock pulse is generated	^{t}HD;STA	4.0	-	0.6	-	µS
LOW period of the SCLK clock	^{t}LOW	4.7	-	1.3	-	µS
HIGH period of the SCLK clock	^{t}HIGH	4.0	-	0.6	-	µS
Set-up time for a repeated START condition	^{t}SU;STA	4.7	-	0.6	-	µS
Data hold time:	^{t}HD;DAT	0^4	3.45^5	0^6	0.9^5	µS
Data set-up time	^{t}SU;DAT	250	-	100^6	-	nS
Rise time of both SDATA and SCLK signals	^{t}r	-	1000	$20 + 0.1Cb^7$	300	nS
Fall time of both SDATA and SCLK signals	^{t}f	-	300	$20 + 0.1Cb^7$	300	nS
Set-up time for STOP condition	^{t}SU;STO	4.0	-	0.6	-	µS
Bus free time between a STOP and START condition	^{t}BUF	4.7	-	1.3	-	µS
Capacitive load for each bus line	Cb	-	400	-	400	pF
Serial interface input pin capacitance	CIN_SI	-	3.3	-	3.3	pF
SDATA max load capacitance	CLOAD_SD	-	30	-	30	pF

Table 24: Two-Wire Serial Bus Characteristics (continued)

^{f}EXTCLK = 27 MHz; VDD = 1.8V; VDD_IO = 2.8V; VAA = 2.8V; VAA_PIX = 2.8V;
VDD_PLL = 2.8V; VDD_DAC = 2.8V; T_A = 25°C

Parameter	Symbol	Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	
SDATA pull-up resistor	RSD	1.5	4.7	1.5	4.7	KΩ

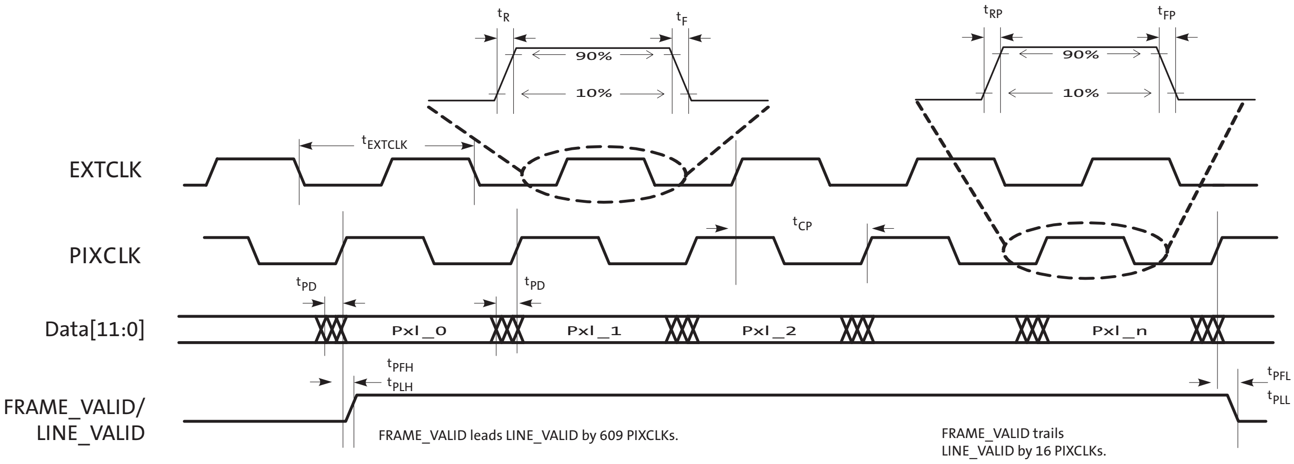
Notes:
1. This table is based on I^2C standard (v2.1 January 2000). Philips Semiconductor.
2. Two-wire control is I^2C-compatible.
3. All values referred to V_{IHmin} = 0.9 VDD and V_{ILmax} = 0.1VDD levels. Sensor EXCLK = 27 MHz.
4. A device must internally provide a hold time of at least 300 ns for the SDATA signal to bridge the undefined region of the falling edge of SCLK.
5. The maximum ^{t}HD;DAT has only to be met if the device does not stretch the LOW period (^{t}LOW) of the SCLK signal.
6. A Fast-mode I^2C-bus device can be used in a Standard-mode I^2C-bus system, but the requirement ^{t}SU;DAT 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLK signal. If such a device does stretch the LOW period of the SCLK signal, it must output the next data bit to the SDATA line ^{t}r max + ^{t}SU;DAT = 1000 + 250 = 1250 ns (according to the Standard-mode I^2C-bus specification) before the SCLK line is released.
7. Cb = total capacitance of one bus line in pF.

I/O Timing

By default, the AR0331 launches pixel data, FV, and LV with the rising edge of PIXCLK. The expectation is that the user captures DOUT[11:0], FV, and LV using the falling edge of PIXCLK.

See Figure 45 and Table 25 for I/O timing (AC) characteristics.

Figure 45: I/O Timing Diagram



*PLL disabled for ^{t}CP

Table 25: I/O Timing Characteristics

Symbol	Definition	Condition	Min	Typ	Max	Unit
^{f}EXTCLK1	Input clock frequency	PLL enabled	6	–	64	MHz
^{t}EXTCLK1	Input clock period	PLL enabled	20	–	166	ns
^{f}EXTCLK2	Input clock frequency	PLL disabled	6	–	74.25	MHz
^{t}EXTCLK2	Input clock period	PLL disabled	13.4	–	166	ns
^{t}R	Input clock rise time		–	3	–	ns
^{t}F	Input clock fall time		–	3	–	ns
^{t}RP	Pixclk rise time		–	4	–	ns
^{t}FP	Pixclk fall time		–	4	–	ns
	Clock duty cycle		40	50	60	%
^{t}(PIX JITTER)	Jitter on PIXCLK		–	1		ns
^{t}CP	EXTCLK to PIXCLK propagation delay	Nominal voltages, PLL Disabled	–	11.3	–	ns
^{f}PIXCLK	PIXCLK frequency	Default, Nominal Voltages	6		74.25	MHz
^{t}PD	PIXCLK to data valid	Default, Nominal Voltages	–	2.3	–	ns
^{t}PFH	PIXCLK to FV HIGH	Default, Nominal Voltages	–	1.5	–	ns
^{t}PLH	PIXCLK to LV HIGH	Default, Nominal Voltages	–	2.3	–	ns
^{t}PFL	PIXCLK to FV LOW	Default, Nominal Voltages	–	1.5	–	ns
^{t}PLL	PIXCLK to LV LOW	Default, Nominal Voltages	–	2	–	ns
CLOAD	Output load capacitance		–	<10	–	pF
CIN	Input pin capacitance		–	2.5	–	pF

DC Electrical Characteristics

The DC electrical characteristics are shown in the tables below.

Table 26: DC Electrical Characteristics

Symbol	Definition	Condition	Min	Typ	Max	Unit
VDD	Core digital voltage		1.7	1.8	1.95	V
VDD_IO	I/O digital voltage		1.7/2.5	1.8/2.8	1.9/3.1	V
VAA	Analog voltage		2.5	2.8	3.1	V
VAA_PIX	Pixel supply voltage		2.5	2.8	3.1	V
VDD_PLL	PLL supply voltage		2.5	2.8	3.1	V
VDD_SLVS	HiSPi supply voltage		0.3	0.4	0.6	V
VIH	Input HIGH voltage		VDD_IO*0.7	–	–	V
VIL	Input LOW voltage		–	–	VDD_IO*0.3	V
IIN	Input leakage current	No pull-up resistor; VIN = VDD_IO or DGND	20	–	–	µA
VOH	Output HIGH voltage		VDD_IO-0.3	–	–	V
VOL	Output LOW voltage		–	–	0.4	V
IOH	Output HIGH current	At specified VOH	-22	–	–	mA
IOL	Output LOW current	At specified VOL	–	–	22	mA

Caution **Stresses greater than those listed in Table 14 may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.**

Table 27: Absolute Maximum Ratings

Symbol	Definition	Condition	Min	Max	Unit
VDD_MAX	Core digital voltage		–0.3	2.4	V
VDD_IO_MAX	I/O digital voltage		–0.3	4	V
VAA_MAX	Analog voltage		–0.3	4	V
VAA_PIX	Pixel supply voltage		–0.3	4	V
VDD_PLL	PLL supply voltage		–0.3	4	V
VDD_SLVS_MAX	HiSPi I/O digital voltage		–0.3	2.4	V
t_{ST}	Storage temperature		–40	85	°C

Note: Exposure to absolute maximum rating conditions for extended periods may affect reliability.

Table 28: Operating Current Consumption in Parallel Output and Linear Mode

Definition	Condition	Symbol	Min	Typ	Max	Unit
Digital operating current	Streaming, 2048x1536 30fps	IDD1	–	TBD	TBD	mA
I/O digital operating current	Streaming, 2048x1536 30fps	IDD_IO	–	TBD	TBD	mA
Analog operating current	Streaming, 2048x1536 30fps	IAA	–	TBD	TBD	mA
Pixel supply current	Streaming, 2048x1536 30fps	IAA_PIX	–	TBD	TBD	mA
PLL supply current	Streaming, 2048x1536 30fps	IDD_PLL	–	TBD	TBD	mA
Digital operating current	Streaming, 1080p 60fps	IDD1	–	TBD	TBD	mA
I/O digital operating current	Streaming, 1080p 60fps	IDD_IO	-	TBD	TBD	mA
Analog operating current	Streaming, 1080p 60fps	IAA	–	TBD	TBD	mA
Pixel supply current	Streaming, 1080p 60fps	IAA_PIX	–	TBD	TBD	mA
PLL supply current	Streaming, 1080p 60fps	IDD_PLL	–	TBD	TBD	mA

Notes: 1. Operating currents are measured at the following conditions:
VAA = VAA_PIX = VDD_IO = VDD_PLL = 2.8V
VDD = 1.8V
PLL Enabled and PIXCLK = 74.25 Mhz
T_A = 25°C

Table 29: Operating Current Consumption in Parallel Output and HDR Mode

Definition	Condition	Symbol	Min	Typ	Max	Unit
Digital operating current	Streaming, 2048x1536 30fps	IDD	–	TBD	TBD	mA
I/O digital operating current	Streaming, 2048x1536 30fps	IDD_IO	–	TBD	TBD	mA
Analog operating current	Streaming, 2048x1536 30fps	IAA	–	TBD	TBD	mA
Pixel supply current	Streaming, 2048x1536 30fps	IAA_PIX	–	TBD	TBD	mA
PLL supply current	Streaming, 2048x1536 30fps	IDD_PLL	–	TBD	TBD	mA
Digital operating current	Streaming, 1080p 60fps	IDD	–	TBD	TBD	mA
I/O digital operating current	Streaming,1080p 60fps	IDD_IO	–	TBD	TBD	mA
Analog operating current	Streaming, 1080p 60fps	IAA	–	TBD	TBD	mA
Pixel supply current	Streaming,1080p 60fps	IAA_PIX	–	TBD	TBD	mA
PLL supply current	Streaming, 1080p 60fps	IDD_PLL	–	TBD	TBD	mA

Notes: 1. Operating currents are measured at the following conditions:
VAA = VAA_PIX = VDD_IO = VDD_PLL = 2.8V
VDD = 1.8V
PLL Enabled and PIXCLK = 74.25 Mhz
T_A = 25°C

Table 30: Operating Currents in HiSPi Output and Linear Mode

Definition	Condition	Symbol	Min	Typ	Max	Unit
Digital Operating Current	Streaming, 2048x1536 30fps	IDD	–	TBD	TBD	mA
I/O digital operating current	Streaming, 2048x1536 30fps	IDD_IO	–	TBD	TBD	µA
Analog operating current	Streaming, 2048x1536 30fps	IAA	–	TBD	TBD	mA
Pixel Supply Current	Streaming, 2048x1536 30fps	IAA_PIX	–	TBD	TBD	mA
PLL Supply Current	Streaming, 2048x1536 30fps	IDD_PLL	–	TBD	TBD	mA
SLVS Supply Current	Streaming, 2048x1536 30fps	IDD_SLVS	–	TBD	TBD	mA
Digital Operating Current	Streaming,1080p 60fps	IDD	–	TBD	TBD	mA
I/O digital operating current	Streaming, 1080p 60fps	IDD_IO	–	TBD	TBD	µA
Analog operating current	Streaming,1080p 60fps	IAA	–	TBD	TBD	mA
Pixel Supply Current	Streaming, 1080p 60fps	IAA_PIX	–	TBD	TBD	mA
PLL Supply Current	Streaming, 1080p 60fps	IDD_PLL	–	TBD	TBD	mA
SLVS Supply Current	Streaming,1080p 60fps	IDD_SLVS	–	TBD	TBD	mA

Notes: 1. Operating currents are measured at the following conditions:
VAA=VAA_PIX=VDD_IO=VDD_PLL=2.8V
VDD=1.8V
VDD_SLVS = 0.4V
PLL Enabled and PIXCLK=74.25Mhz
T_A = 25°C

Table 31: Operating Current in HiSPi Output and HDR Mode

Definition	Condition	Symbol	Min	Typ	Max	Unit
Digital Operating Current	Streaming, 2048x1536 30fps	IDD	–	TBD	TBD	mA
I/O digital operating current	Streaming, 2048x1536 30fps	IDD_IO	–	TBD	TBD	µA
Analog operating current	Streaming, 2048x1536 30fps	IAA	–	TBD	TBD	mA
Pixel Supply Current	Streaming, 2048x1536 30fps	IAA_PIX	–	TBD	TBD	mA
PLL Supply Current	Streaming, 2048x1536 30fps	IDD_PLL	–	TBD	TBD	mA
SLVS Supply Current	Streaming, 2048x1536 30fps	IDD_SLVS	–	TBD	TBD	mA
Digital Operating Current	Streaming,1080p 60fps	IDD	–	TBD	TBD	mA
I/O digital operating current	Streaming,1080p 60fps	IDD_IO	–	TBD	TBD	µA
Analog operating current	Streaming,1080p 60fps	IAA	–	TBD	TBD	mA
Pixel Supply Current	Streaming,1080p 60fps	IAA_PIX	–	TBD	TBD	mA
PLL Supply Current	Streaming,1080p 60fps	IDD_PLL	–	TBD	TBD	mA
SLVS Supply Current	Streaming,1080p 60fps	IDD_SLVS	–	TBD	TBD	mA

Notes: 1. Operating currents are measured at the following conditions:
VAA=VAA_PIX=VDD_IO=VDD_PLL=2.8V
VDD=1.8V
VDD_SLVS = 0.4V
PLL Enabled and PIXCLK=74.25MHz
T_A = 25°C

Table 32: Standby Current Consumption

Definition	Condition	Symbol	Min	Typ	Max	Unit
Soft standby (clock off)	Analog, 2.8V	-	–	TBD	–	µA
	Digital, 1.8V	-	–	TBD	–	µA
Soft standby (clock on)	Analog, 2.8V	-	–	TBD	–	µA
	Digital, 1.8V	-	–	TBD	–	mA

Notes:
1. Analog – VAA + VAA_PIX + VDD_PLL + VDD_PHY
2. Digital – VDD + VDD_IO + VDD_SLVS

HiSPi Electrical Specifications

The HiSPi transmitter electrical specifications are listed at 700 MHz.

Table 33: Input Voltage Levels and Operating Temperatures PWRHiSPi

Parameter	Symbol	Min	Typ	Max	Unit
HiSPi power supply	VDD_SLVS	0.4	0.4	0.8	V
Operating temperature	TA	–30	–	85	°C

Table 34: Input Voltage and Current
Measurement Conditions: HiSPi Power Supply 0.4 V, Max Freq 700 MHz

Parameter	Symbol	Min	Typ	Max	Unit
Supply current (PWRHiSPi) (driving 100Ω load)	IDD_SLVS	–	10	15	mA
HiSPi common mode voltage (driving 100Ω load)	VCMD	VCMDtyp x 0.8	VDD_SLVS/2	VCMDtyp x 1.2	V
HiSPi differential output voltage (driving 100Ω load)	VDIFF	VDIFFtyp x 0.8	VDD_SLVS/2	VDIFFtyp x 1.2	V
Output impedance	–	35	50	70	Ω

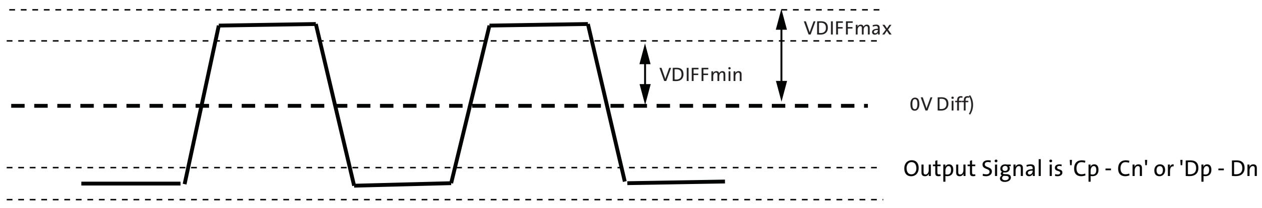
Table 35: Input Voltage Levels and Operating Temperatures HiVCM

Parameter	Symbol	Min	Typ	Max	Unit
HiSPi power supply	VDD_SLVS	1.7	1.8	1.9	V
Operating temperature	TA	–30	–	85	°C

Table 36: Input Voltage and Current
Measurement Conditions: HiSPi Power Supply 1.8 V, Max Freq 700 MHz

Parameter	Symbol	Min	Typ	Max	Unit
Supply current (PWRHiSPi) (driving 100Ω load)	IDD_HiVCM	–	15	TBD	mA
HiSPi common mode voltage (driving 100Ω load)	VCMD	VCMDtyp x 0.8	VDD_SLVS/2	VCMDtyp x 1.2	V
HiSPi differential output voltage (driving 100Ω load)	VDIFF	VDIFFtyp x 0.8	VDD_SLVS/2	VDIFFtyp x 1.2	V
Output impedance	–	–	–	–	Ω

Figure 46: Differential Output Voltage for Clock or Data Pairs



;

Table 37: HiSPi Rise and Fall Times at 660 Mbps

Measurement Conditions: VDD_HiSPi = 1.8V, VDD_HiSPi_TX = 0.8V; Data Rate = 660 Mbps; DLL set to 0

Parameter	Name	Value	Unit
Max Setup Time from Transmitter	TxPRE	0.44	UI
Max Hold Time from Transmitter	TxPost	0.44	UI
Rise time t	^{t}RISE	350	ps
Fall time t	^{t}FALL	350	ps
Output impedance		66	Ω

;

Table 38: HiSPi Rise and Fall Times at 360 Mbps

Measurement Conditions: VDD_HiSPi = 1.8V;VDD_HiSPi_TX = 0.8V; Data Rate =360 Mbps; DLL set to 0 DLL set to 0

Parameter	Name	Value	Unit
Max Setup Time from Transmitter	TxPRE	0.48	UI
Max Hold Time from Transmitter	TxPost	0.42	UI
Rise time t	^{t}RISE	350	ps
Fall time t	^{t}FALL	350	ps
Output impedance		66	Ω

Figure 47: Eye Diagram for Clock and Data Signals

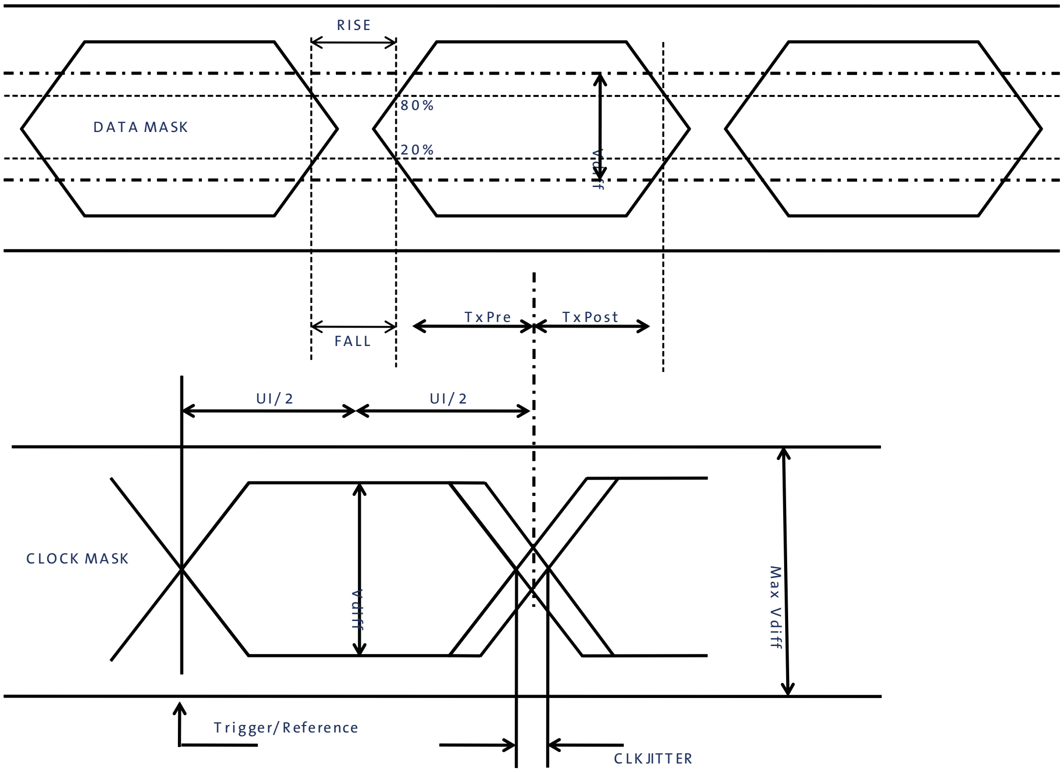


Table 39: Channel, PHY and Intra-PHY Skew

Measurement Conditions: VDD_HiSPi = 1.8V;VDD_HiSPi_TX = 0.8V; Data Rate =480 Mbps; DLL set to 0

Data Lane Skew in Reference to Clock	tCHSKEW1PHY	-150	ps

Figure 48: HiSPi Skew Between Data Signals Within the PHY

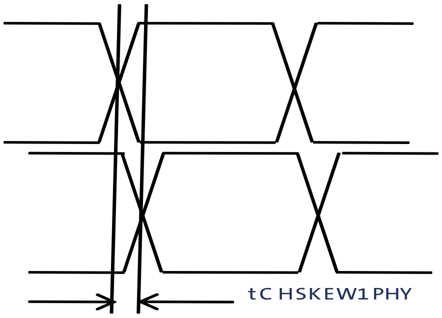


Table 40: Clock DLL Steps

Measurement Conditions: VDD_HiSPi = 1.8V;VDD_HiSPi_TX = 0.8V; Data DLL set to 0

Clock DLL Step	1	2	3	4	5	Step
Delay at 660 Mbps	0.25	0.375	0.5	0.625	0.75	UI

Table 40: **Clock DLL Steps**
Measurement Conditions: VDD_HiSPi = 1.8V;VDD_HiSPi_TX = 0.8V; Data DLL set to 0

Clock DLL Step	1	2	3	4	5	Step
Eye_opening at 660 Mbps	0.85	0.78	0.71	0.71	0.69	UI

Note: The Clock DLL Steps 6 and 7 are not recommended by Aptina for the AR0331 Rev. 1.

Table 41: **Data DLL Steps**
Measurement Conditions: VDD_HiSPi = 1.8V;VDD_HiSPi_TX = 0.8V; Clock DLL set to 0

Data DLL Step	1	2	4	6	Step
Delay at 660 Mbps	0.25	0.375	0.625	0.875	UI
Eye opening at 660 Mbps	0.79	0.84	0.71	0.61	UI

Note: The Data DLL Steps 3, 5, and 7 are not recommended by Aptina for the AR0331 Rev. 1.

Power-On Reset and Standby Timing

Power-Up Sequence

The recommended power-up sequence for the AR0331 is shown in Figure 49. The available power supplies (VDD_IO, VDD, VDD_SLVS, VDD_PLL, VAA, VAA_PIX) must have the separation specified below.

1. Turn on VDD_PLL power supply.
2. After 100µs, turn on VAA and VAA_PIX power supply.
3. After 100µs, turn on VDD_IO power supply.
4. After 100µs, turn on VDD power supply.
5. After 100µs, turn on VDD_SLVS power supply.
6. After the last power supply is stable, enable EXTCLK.
7. Assert RESET_BAR for at least 1ms.
8. Wait 150000 EXTCLKs (for internal initialization into software standby.
9. Configure PLL, output, and image settings to desired values.
10. Wait 1ms for the PLL to lock.
11. Set streaming mode (R0x301a[2] = 1).

Figure 49: **Power Up**

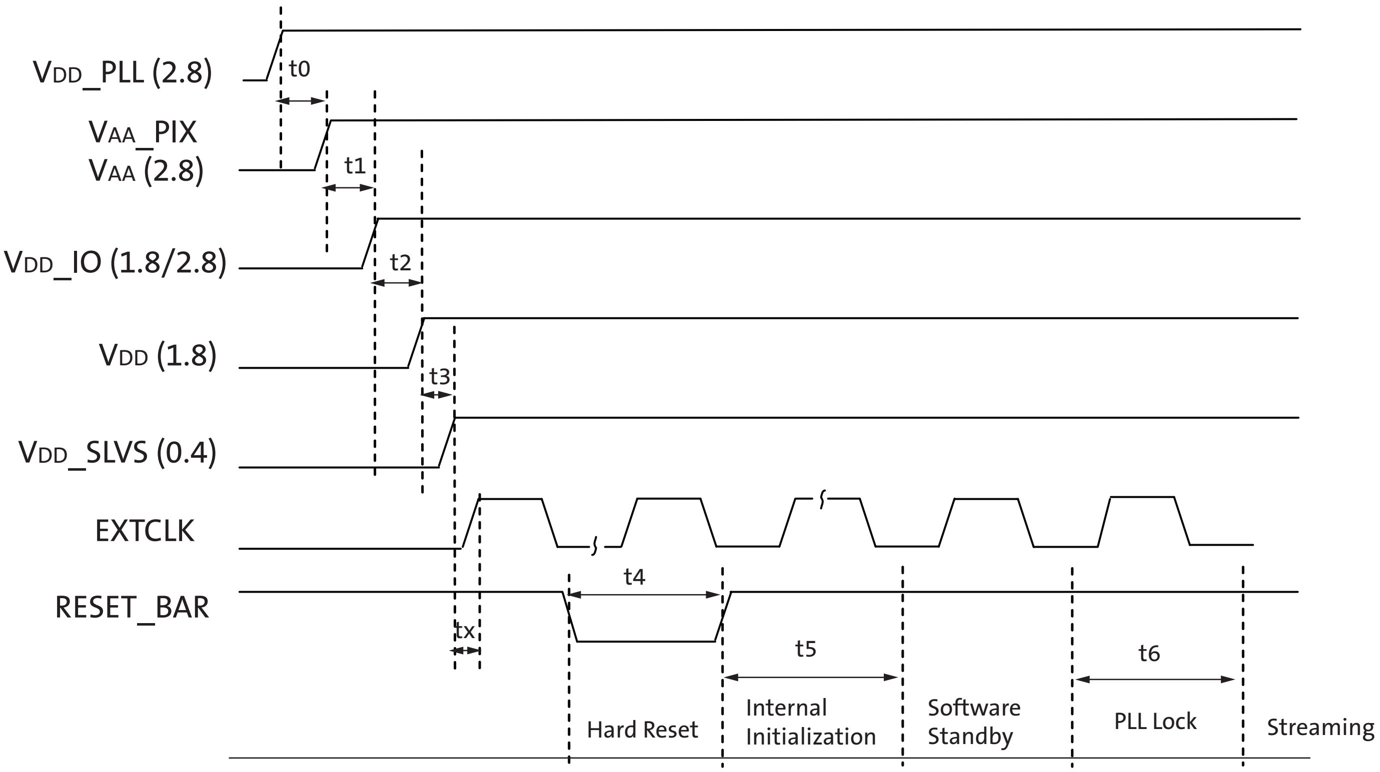


Table 42: Power-Up Sequence

Definition	Symbol	Minimum	Typical	Maximum	Unit
VDD_PLL to VAA/VAA_PIX[3]	t0	0	100	–	µS
VAA/VAA_PIX to VDD_IO	t1	0	100	–	µS
VDD_IO to VDD	t2	0	100	–	µS
VDD to VDD_SLVS	t3	0	100	–	µS
Xtal settle time	tx	–	30[1]	–	mS
Hard Reset	t4	1[2]	–	–	mS
Internal Initialization	t5	150000	–	–	EXTCLKS
PLL Lock Time	t6	1	–	–	mS

Notes:
1. Xtal settling time is component-dependent, usually taking about 10 – 100 mS.
2. Hard reset time is the minimum time required after power rails are settled. In a circuit where Hard reset is held down by RC circuit, then the RC time must include the all power rail settle time and Xtal settle time.
3. It is critical that VDD_PLL is not powered up after the other power supplies. It must be powered before or at least at the same time as the others. If the case happens that VDD_PLL is powered after other supplies then sensor may have functionality issues and will experience high current draw on this supply.

Power-Down Sequence

The recommended power-down sequence for the AR0331 is shown in Figure 50. The available power supplies (VDD_IO, VDD, VDD_SLVS, VDD_PLL, VAA, VAA_PIX) must have the separation specified below.

1. Disable streaming if output is active by setting standby R0x301a[2] = 0
2. The soft standby state is reached after the current row or frame, depending on configuration, has ended.
3. Turn off VDD_SLVS.
4. Turn off VDD.
5. Turn off VDD_IO
6. Turn off VAA/VAA_PIX.
7. Turn off VDD_PLL.

Figure 50: Power Down

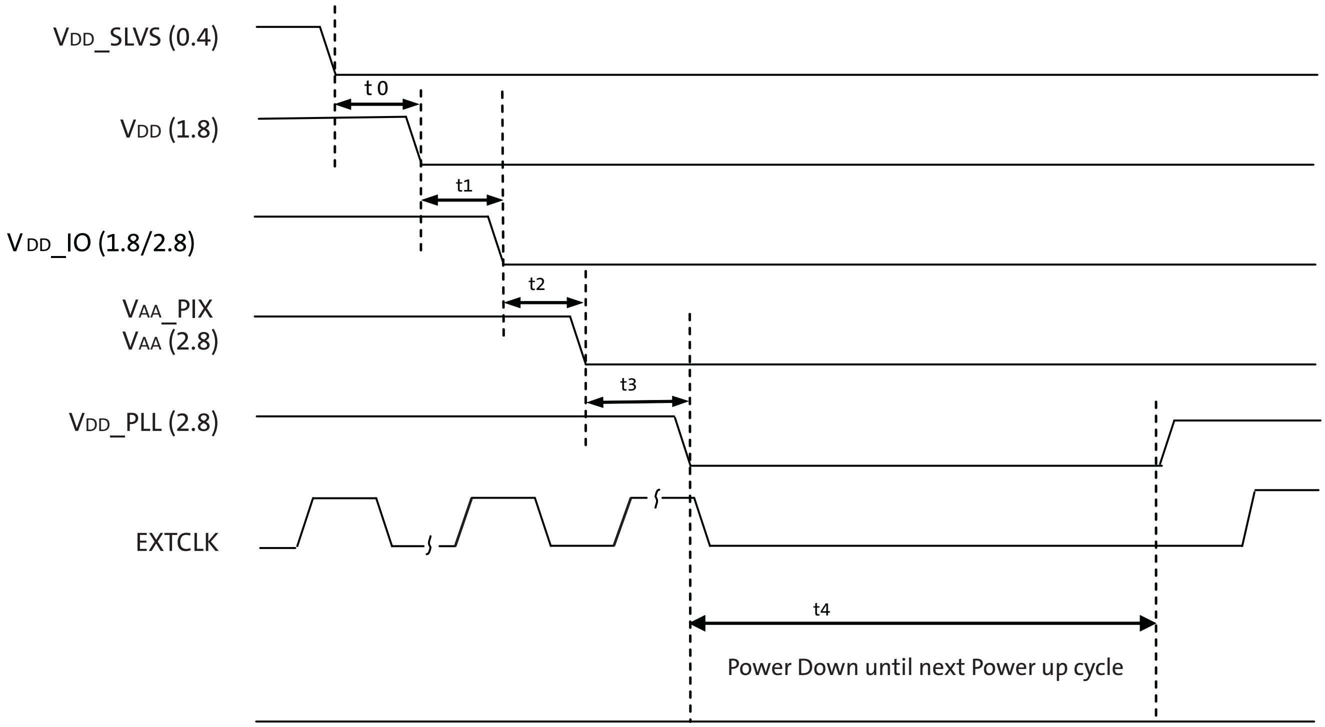


Table 43: Power-Down Sequence

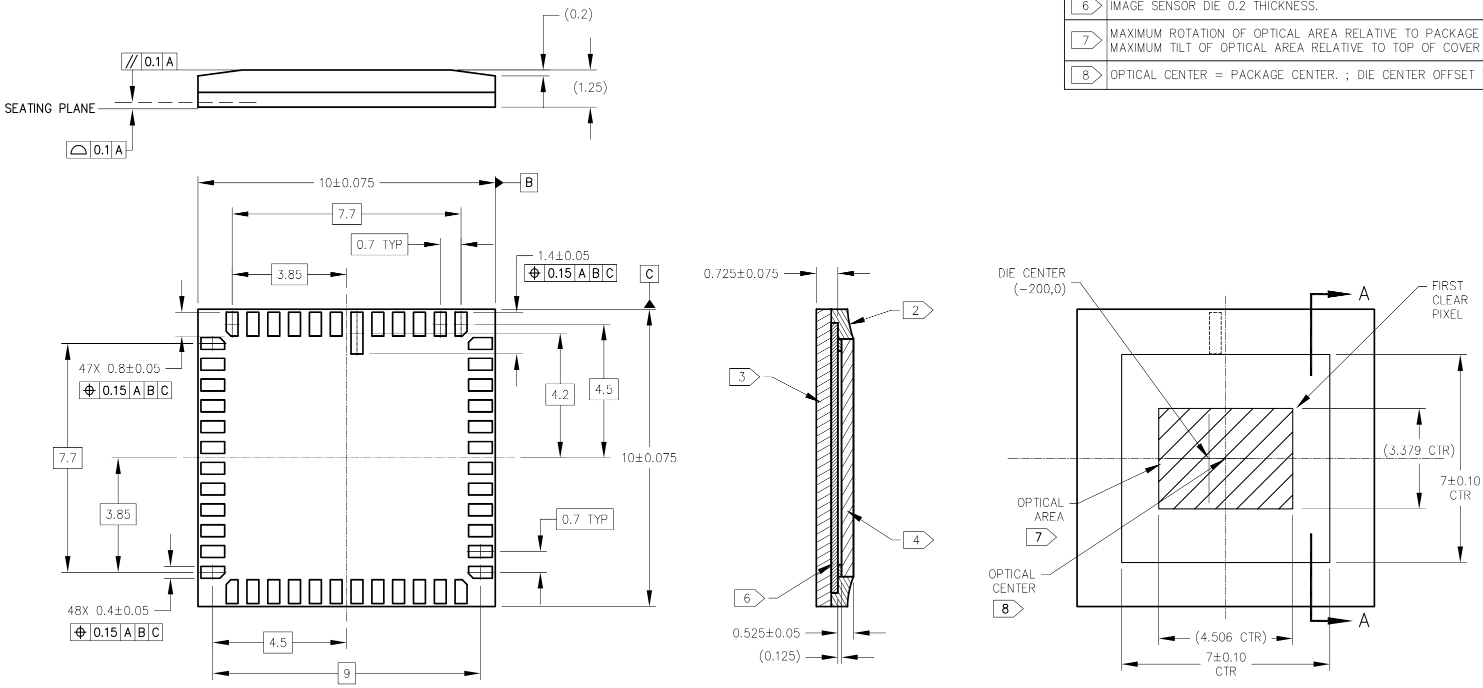
Definition	Symbol	Minimum	Typical	Maximum	Unit
VDD_SLVS to VDD	t0	0	–	–	µS
VDD to VDD_IO	t1	0	–	–	µS
VDD_IO to VAA/VAA_PIX	t2	0	–	–	µS
VAA/VAA_PIX to VDD_PLL	t3	0	–	–	µS
PwrDn until Next PwrUp Time	t4	100	–	–	mS

Note: t4 is required between power down and next power up time; all decoupling caps from regulators must be completely discharged.

Package Dimensions

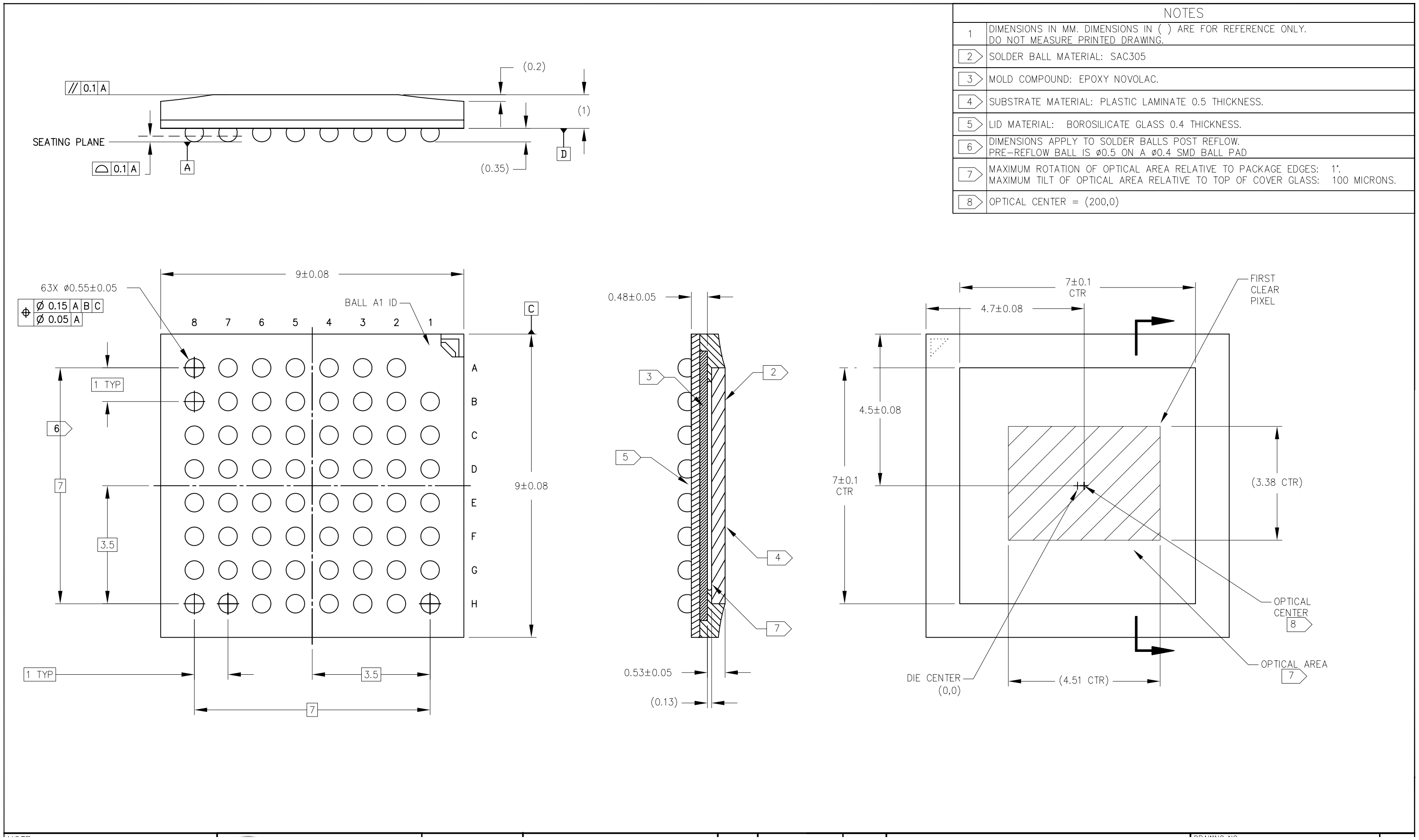
Figure 51: 48 iLCC Package Outline Drawing

	NOTES
1	DIMENSIONS IN MM. DIMENSIONS IN ( ) ARE FOR REFERENCE ONLY. DO NOT MEASURE PRINTED DRAWING.
2	ENCAPSULANT: EPOXY.
3	SUBSTRATE MATERIAL: PLASTIC LAMINATE 0.5 THICKNESS.
4	LID MATERIAL: BOROSILICATE GLASS 0.4 THICKNESS. REFRACTIVE INDEX AT 20°C = 1.5255 @ 546nm & 1.5231 @ 588nm.
5	LEAD FINISH: GOLD PLATING, 0.5 MICRONS MINIMUM THICKNESS.
6	IMAGE SENSOR DIE 0.2 THICKNESS.
7	MAXIMUM ROTATION OF OPTICAL AREA RELATIVE TO PACKAGE EDGES: 1°. MAXIMUM TILT OF OPTICAL AREA RELATIVE TO TOP OF COVER GLASS: 100 MICRONS
8	OPTICAL CENTER = PACKAGE CENTER. ; DIE CENTER OFFSET TO (−200,0)



Note: All dimensions are in millimeters.

Figure 52: 64-Ball iBGA Package Outline Drawing



	NOTES
1	DIMENSIONS IN MM. DIMENSIONS IN ( ) ARE FOR REFERENCE ONLY. DO NOT MEASURE PRINTED DRAWING.
2	SOLDER BALL MATERIAL: SAC305
3	MOLD COMPOUND: EPOXY NOVOLAC.
4	SUBSTRATE MATERIAL: PLASTIC LAMINATE 0.5 THICKNESS.
5	LID MATERIAL: BOROSILICATE GLASS 0.4 THICKNESS.
6	DIMENSIONS APPLY TO SOLDER BALLS POST REFLOW. PRE-REFLOW BALL IS ø0.5 ON A ø0.4 SMD BALL PAD
7	MAXIMUM ROTATION OF OPTICAL AREA RELATIVE TO PACKAGE EDGES: 1°. MAXIMUM TILT OF OPTICAL AREA RELATIVE TO TOP OF COVER GLASS: 100 MICRONS.
8	OPTICAL CENTER = (200,0)

Revision History

Rev. B 3/8/11

- Updated “Features” on page 1
- Updated Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8
- Updated Figure 3: “Typical Configuration: Parallel Pixel Data Interface,” on page 9
- Updated Table 2, “Available Part Numbers,” on page 2
- Updated Table 3, “Pin Descriptions,” on page 11
- Updated Table 4, “Pin Descriptions, 48 iLCC,” on page 12
- Updated Table 5, “Pin Descriptions, 9 x 9 mm, 64-ball iBGA Parallel,” on page 14
- Added “Positional Gain Adjustments (PGA)” on page 18
- Added “The Correction Function” on page 18
- Updated “Serial Configuration” on page 23
- Updated Table 8, “Recommended Sensor Gain,” on page 26
- Updated Table 10, “Knee Points for Compression from 16 Bits to 12 Bits,” on page 28
- Added Table 11, “Bit Operation After Linearization,” on page 28with introductory text above it
- Updated Table 25, “I/O Timing Characteristics,” on page 58
- Updated Figure 51: “48 iLCC Package Outline Drawing,” on page 70
- Updated Figure 52: “64-Ball iBGA Package Outline Drawing,” on page 71

Rev. A 1/26/11

- Initial release

10 Eunos Road 8 13-40, Singapore Post Center, Singapore 408600 prodmktg@aptina.com www.aptina.com
Aptina, Aptina Imaging, and the Aptina logo are the property of Aptina Imaging Corporation
All other trademarks are the property of their respective owners.
Advance: This data sheet contains initial descriptions of products still under development.