# 1/3-Inch CMOS Digital Image Sensor

## AR0330 Data Sheet, Rev. U

For the latest data sheet, please visit www.onsemi.com

## Features

- 2.2 µm pixel with ON Semiconductor A-Pix™ technology
- Full HD support at 60 fps (2304H x 1296V) for maximum video performance
- Superior low-light performance
- 3.4Mp (3:2) and 3.15 Mp (4:3)still images
- Support for external mechanical shutter
- Support for external LED or Xenon flash
- Data interfaces: four-lane serial high-speed pixel interface (HiSPi™) differential signaling (SLVS), four-lane serial MIPI interface, or parallel.
- On-chip phase-locked loop (PLL) oscillator
- Simple two-wire serial interface
- Auto black level calibration
- 12-to-10 bit output A-Law compression
- Slave mode for precise frame-rate control and for synchronizing two sensors

## Applications

- 1080p high-definition digital video camcorder
- Web cameras and video conferencing cameras
- Security

## General Description

The ON Semiconductor AR0330 is a 1/3-inch CMOS digital image sensor with an active-pixel array of 2304Hx1536V. It can support 3.15 megapixel (2048H x 1536V) digital still image capture and a 1080p60+20%EIS (2304H x 1296V) digital video mode. It incorporates sophisticated on-chip camera functions such as windowing, mirroring, column and row subsampling modes, and snapshot modes.

**Table 1: Key Parameters**

| Parameter | | Typical Value |
|---|---|---|
| Optical format | | 1/3-inch (6.0 mm)<br>Entire Array: 6.09 mm<br>Still Image: 5.63 mm (4:3)<br>HD Image: 5.82 mm (16:9) |
| Active pixels | | 2304(H) x 1536(V): (entire array): 5.07 mm (H) x 3.38 mm (V)<br>2048(H) x 1536(V) (4:3, still mode)<br>2304(H) x 1296(V) (16:9, sHD mode) |
| Pixel size | | 2.2 µm x 2.2µm |
| Color filter array | | RGB Bayer |
| Shutter type | | ERS and GRR |
| Input clock range | | 6 – 27 MHz |
| Output clock maximum | | 196 Mp/s (4-lane HiSPi or MIPI) |
| Output Video | 4-lane HiSPi | 2304 x 1296 at 60 fps<br>< 450 mW (Vcm 0.2V, 198MP/s)<br>230 x 1296 at 30 fps<br>< 300 mW (VCM 0.2V, 98 Mp/s) |
| Responsivity | | 2.0 V/lux-sec |
| $SNR_{MAX}$ | | 39 dB |
| Dynamic range | | 69.5 dB |
| Supply voltage | I/O/Digital | 1.7–1.9 V (1.8 V nominal) or<br>2.4–3.1 V (2.8 V nominal) |
| | Digital | 1.7–1.9 V (1.8 V nominal) |
| | Analog | 2.7–2.9 V |
| | HiSPi PHY | 1.7–1.9 V (1.8 V nominal) |
| | HiSPi I/O (SLVS) | 0.3–0.9 V (0.4 or 0.8 V nominal) |
| | HiSPi I/O (HiVCM) | 1.7–1.9 V (1.8 V nominal) |
| Operating temperature (junction) -$T_J$ | | –30°C to + 70° C |
| Package options | | 11.4 mm x 11.4 mm CLCC |
| | | 6.28 mm x 6.65 mm CSP |
| | | Bare die |

# Ordering Information

**Table 2: Available Part Numbers**

| Part Number | Product Description | Orderable Product Attribute Description |
|---|---|---|
| AR0330CM1C00SHAA0-DP | 3 MP 1/3" CIS | Dry Pack with Protective Film |
| AR0330CM1C00SHAA0-DR | 3 MP 1/3" CIS | Dry Pack without Protective Film |
| AR0330CM1C00SHAA0-TP | 3.5 MP 1/3" CIS | Tape & Reel with Protective Film |
| AR0330CM1C00SHKA0-CP | 3 MP 1/3" CIS | Chip Tray with Protective Film |
| AR0330CM1C00SHKA0-CR | 3 MP 1/3" CIS | Chip Tray without Protective Film |
| AR0330CM1C12SHAA0-DP | 3 MP 1/3" CIS | Dry Pack with Protective Film |
| AR0330CM1C12SHAA0-DR | 3 MP 1/3" CIS | Dry Pack without Protective Film |
| AR0330CM1C12SHKA0-CP | 3 MP 1/3" CIS | Chip Tray with Protective Film |
| AR0330CM1C12SHKA0-CR | 3 MP 1/3" CIS | Chip Tray without Protective Film |
| AR0330CM1C21SHKA0-CP | 3 MP 1/3" CIS | Chip Tray with Protective Film |
| AR0330CM1C21SHKA0-CR | 3 MP 1/3" CIS | Chip Tray without Protective Film |

## Table of Contents

## List of Figures

## List of Tables

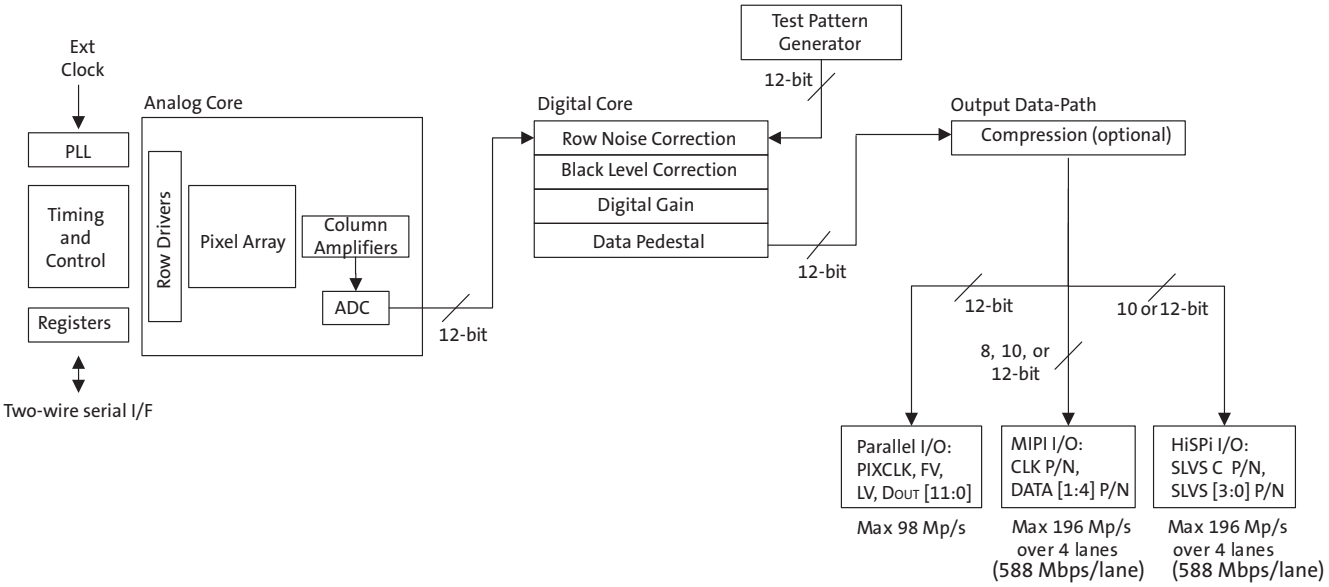
## General Description

The AR0330 can be operated in its default mode or programmed for frame size, exposure, gain, and other parameters. The default mode output is a 2304 x 1296 image at 60 frames per second (fps). The sensor outputs 10- or 12-bit raw data, using either the parallel or serial (HiSPi, MIPI) output ports.

## Functional Overview

The AR0330 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 generate all internal clocks from a single master input clock running between 6 and 27 MHz. The maximum output pixel rate is 196 Mp/s using a 4-lane HiSPi or MIPI serial interface and 98 Mp/s using the parallel interface. 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 3.4Mp 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 signal from the column is amplified in a column amplifier and then digitized in 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).

# Working Modes

The AR0330 sensor working modes are specified from the following aspect ratios:

**Table 3: Available Aspect Ratios in the AR0330 Sensor**

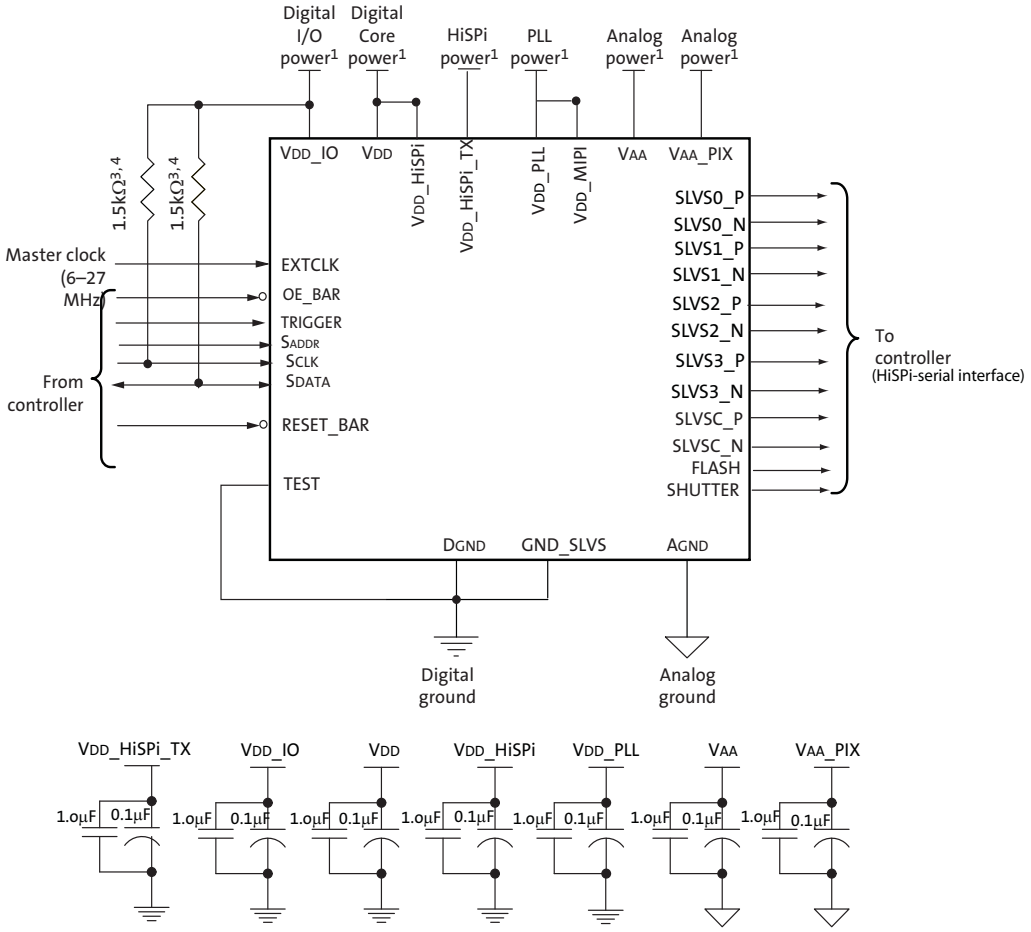
| Aspect Ratio | | Sensor Array Usage |
|---|---|---|
| 3:2 | Still Format #1 | 2256 (H) x 1504 (V) |
| 4:3 | Still Format #2 | 2048(H) x 1536 (V) |
| 16:10 | Still Format #3 | 2256 (H) x 1440(V) |
| 16:9 | HD Format | 2304 (H) x 1296 (V) |

The AR0330 supports the following working modes. To operate the sensor at full speed (196 Mp/s) the sensor must use the 4-lane HiSPi or MIPI interface. The sensor will operate at half-speed (98 Mp/s) when using the parallel interface.

**Table 4: Available Working Modes in the AR0330 Sensor**

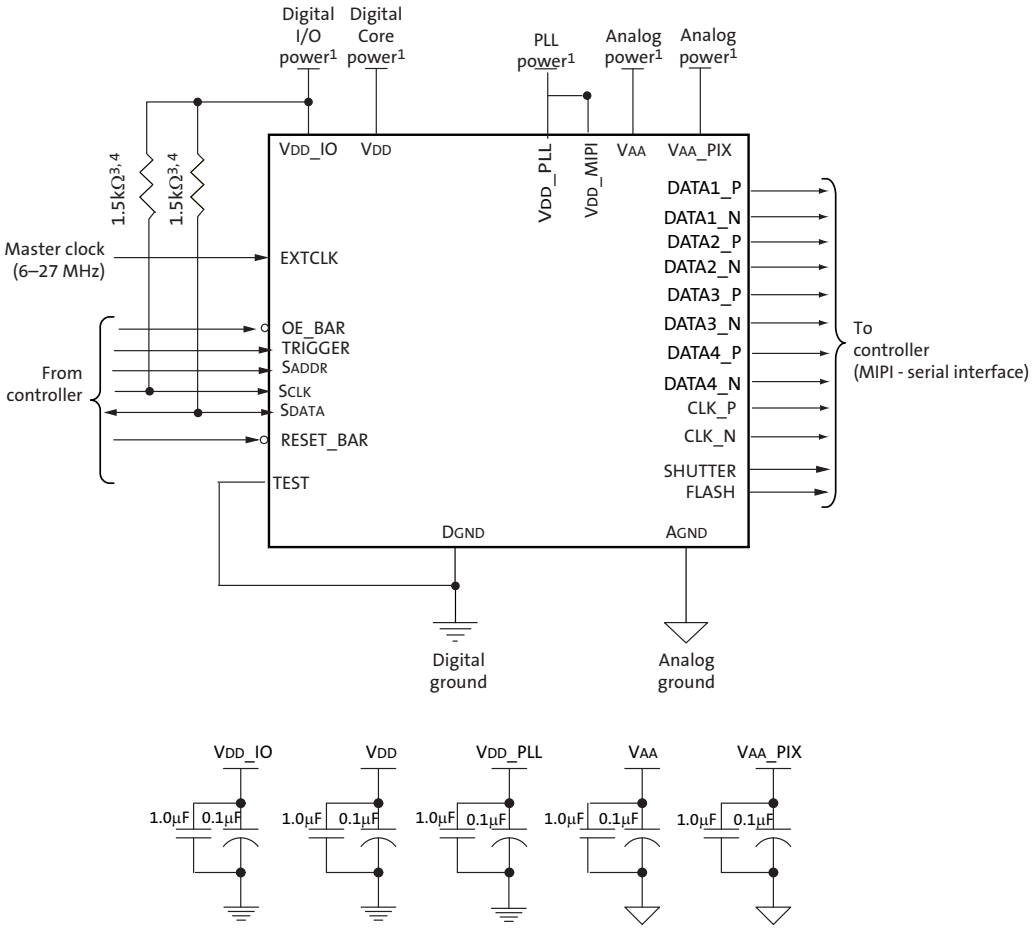
| Mode | Aspect Ratio | Active Readout Window | Sensor Output Resolution | FPS (4-Lane MIPI/ HiSPi Interface) | FPS (Parallel Interface) | Sub-sampling | FOV |
|---|---|---|---|---|---|---|---|
| 1080p + EIS | 16:9 | 2304 x 1296 | 2304 x 1296 | 60 | n/a | - | 100% |
| | | | | 30 | 30 | - | 100% |
| 3M Still | 4:3 | 2048 x 1536 | 2048 x 1536 | 30 | 25 | - | 100% |
| | 3:2 | 2256 x 1504 | 2256 x 1504 | 30 | 25 | - | 100% |
| WVGA + EIS | 16:9 | 2304 x 1296 | 1152 x 648 | 60 | 60 | 2x2 | 100% |
| WVGA + EIS Slow-motion | 16:9 | 2304 x 1296 | 1152 x 648 | 120 | N/A | 2x2 | 100% |
| VGA Video | 16:10 | 2256 x 1440 | 752 x 480 | 60 | 60 | 3x3 | 96% |
| VGA Video Slow-motion | 16:10 | 2256 x 1440 | 752 x 480 | 215 | 107 | 3x3 | 96% |

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



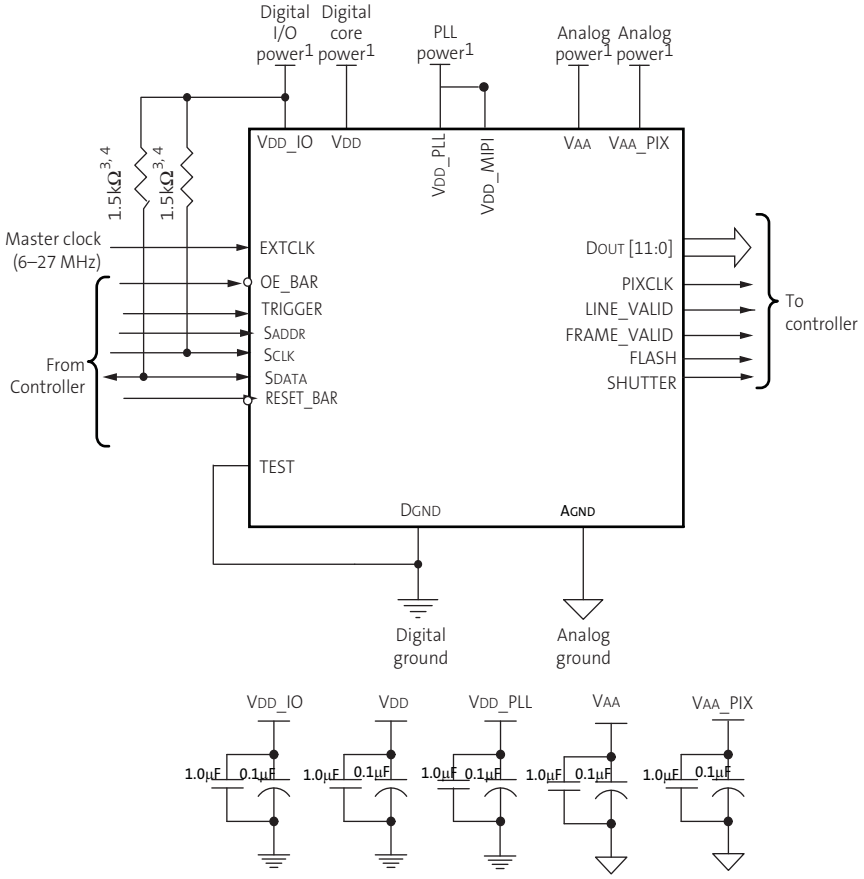

Notes:
1. All power supplies must be adequately decoupled. ON Semiconductor recommends having 1.0µF and 0.1µF decoupling capacitors for every power supply. If space is a concern, then priority must be given in the following order: VAA, VAA_PIX, VDD_PLL, VDD_IO, and VDD. Actual values and results may vary depending on layout and design considerations.
2. To allow for space constraints, ON Semiconductor recommends having 0.1µF decoupling capacitor inside the module as close to the pads as possible. In addition, place a 10µF capacitor for each supply off-module but close to each supply.
3. ON Semiconductor recommends a resistor value of 1.5kΩ, but a greater value may be used for slower two-wire speed.
4. The pull-up resistor is not required if the controller drives a valid logic level on SCLK at all times.
5. ON Semiconductor recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.
6. TEST pin should be tied to DGND.
7. Set High_VCM (R0x306E[9]) to 0 (default) to use the VDD_HiSPi_TX in the range of 0.4 – 0.8V. Set High_VCM to 1 to use a range of 1.7 – 1.9V.
8. The package pins or die pads used for the MIPI data and clock as well as the parallel interface must be left floating.
9. The VDD_MIPI package pin and sensor die pad should be connected to a 2.8V supply as VDD_MIPI is tied to the VDD_PLL supply both in the package routing and also within the sensor die itself.
10. If the SHUTTER or FLASH pins or pads are not used, then they must be left floating.
11. If the TRIGGER or OE_BAR pins or pads are not used, then they should be tied to DGND.
12. The GND_SLVS pad must be tied to DGND. It is connected this way in the CLCC and CSP packages.

**Figure 3: Typical Configuration: Serial MIPI**




Notes:
1. All power supplies must be adequately decoupled. ON Semiconductor recommends having 1.0µF and 0.1µF decoupling capacitors for every power supply. If space is a concern, then priority must be given in the following order: VAA, VAA_PIX, VDD_PLL, VDD_MIPI, VDD_IO, and VDD. Actual values and results may vary depending on layout and design considerations.
2. To allow for space constraints, ON Semiconductor recommends having 0.1µF decoupling capacitor inside the module as close to the pads as possible. In addition, place a 10µF capacitor for each supply off-module but close to each supply.
3. ON Semiconductor recommends a resistor value of 1.5kΩ, but a greater value may be used for slower two-wire speed.
4. The pull-up resistor is not required if the controller drives a valid logic level on SCLK at all times.
5. ON Semiconductor recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.
6. TEST pin must be tied to DGND for the MIPI configuration.
7. ON Semiconductor recommends that GND_MIPI be tied to DGND.
8. VDD_MIPI is tied to VDD_PLL in both the CLCC and the CSP package. ON Semiconductor strongly recommends that VDD_MIPI must be connected to a VDD_PLL in a module design since VDD_PLL and VDD_MIPI are tied together in the die.
9. The package pins or die pads used for the HiSPi data and clock as well as the parallel interface must be left floating.
10. HiSPi Power Supplies (VDD_HiSPi and VDD_HiSPi_TX) can be tied to ground.
11. If the SHUTTER or FLASH pins or pads are not used, then they must be left floating.
12. If the TRIGGER or OE_BAR pins or pads are not used, then they should be tied to DGND.

**Figure 4: Typical Configuration: Parallel Pixel Data Interface**




Notes:
1. All power supplies must be adequately decoupled. ON Semiconductor recommends having 1.0µF and 0.1µF decoupling capacitors for every power supply. If space is a concern, then priority must be given in the following order: VAA, VAA_PIX, VDD_PLL, VDD_IO, and VDD. Actual values and results may vary depending on layout and design considerations.
2. To allow for space constraints, ON Semiconductor recommends having 0.1µF decoupling capacitor inside the module as close to the pads as possible. In addition, place a 10µF capacitor for each supply off-module but close to each supply.
3. ON Semiconductor recommends a resistor value of 1.5kΩ, but a greater value may be used for slower two-wire speed.
4. The pull-up resistor is not required if the controller drives a valid logic level on SCLK at all times.
5. ON Semiconductor recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.
6. TEST pin should be tied to the ground.
7. The data and clock package pins or die pads used for the HiSPi and MIPI interface must be left floating.
8. The VDD_MIPI package pin and sensor die pad should be connected to a 2.8V supply as it is tied to the VDD_PLL supply both in the package routing and also within the sensor die itself. HiSPi Power Supplies (VDD_HiSPi and VDD_HiSPi_TX) can be tied to ground.
9. If the SHUTTER or FLASH pins or pads are not used, then they must be left floating.
10. If the TRIGGER or OE_BAR pins or pads are not used, then they should be tied to DGND.

## HiSPi Power Supply Connections

The HiSPi interface requires two power supplies. The VDD_HiSPi powers the digital logic while the VDD_HiSPi _TX powers the output drivers. The digital logic supply is a nominal 1.8V and ranges from 1.7 to 1.9V. The HiSPi drivers can receive a supply voltage of 0.4 to 0.8V or 1.7 to 1.9V.

The common mode voltage is derived as half of the VDD_HiSPi _TX supply. Two settings are available for the output common mode voltage:

1. SLVS mode. The VDD_HiSPi_Tx supply must be in the range of 0.4 to 0.8V and the high_vcm register bit R0x306E[9] must be set to “0”. The output common mode voltage will be in the range of 0.2 to 0.4V.
2. HiVCM mode. The VDD_HiSPi_Tx supply must be in the range of 1.7 to 1.9V and the high_vcm register bit R0x306E[9] must be set to “1”. The output common mode voltage will be in the range of 0.76 to 1.07V.

Two prior naming conventions have also been used with the VDD_HiSPi and VDD_HiSPi _TX pins:

1. Digital logic supply was named VDD_SLVS while the driver supply was named VDD_SLVS_TX.
2. Digital logic supply was named VDD_PHY while the driver supply was named VDD_SLVS.

# Pin Descriptions

**Table 5: Pin Descriptions**

| Name | Type | Description |
|---|---|---|
| RESET_BAR | Input | Asynchronous reset (active LOW). All settings are restored to factory default. |
| EXTCLK | Input | Master input clock, range 6 -27 MHz |
| OE_BAR | Input | Output enable (active LOW). Only available on bare die version. |
| TRIGGER | Input | Receives slave mode VD signal for frame rate synchronization and trigger to start a GRR frame. |
| SADDR | Input | Two-wire serial address select. |
| SCLK | Input | Two-wire serial clock input. |
| SDATA | I/O | Two-wire serial data I/O. |
| PIXCLK | Output | Pixel clock out. DOUT is valid on rising edge of this clock. |
| DOUT[11:0] | Output | Parallel pixel data output. |
| FLASH | Output | Flash output. Synchronization pulse for external light source. Can be left floating if not used. |
| FRAME_VALID | Output | Asserted when DOUT data is valid. |
| LINE_VALID | Output | Asserted when DOUT data is valid. |
| VDD | Power | Digital power. |
| VDD_IO | Power | IO supply power. |
| VDD_PLL | Power | PLL power supply. The MIPI power supply (VDD_MIPI) is tied to VDD_PLL in both packages. |
| DGND | Power | Digital GND. |
| VAA | Power | Analog power. |
| VAA_PIX | Power | Pixel power. |
| AGND | Power | Analog GND. |
| TEST | Input | Enable manufacturing test modes. Tie to DGND for normal sensor operation. |
| SHUTTER | Output | Control for external mechanical shutter. Can be left floating if not used. |
| SLVS0_P | Output | HiSPi serial data, lane 0, differential P. |
| SLVS0_N | Output | HiSPi serial data, lane 0, differential N. |
| SLVS1_P | Output | HiSPi serial data, lane 1, differential P. |
| SLVS1_N | Output | HiSPi serial data, lane 1, differential N. |
| SLVS2_P | Output | HiSPi serial data, lane 2, differential P. |
| SLVS2_N | Output | HiSPi serial data, lane 2, differential N. |
| SLVS3_P | Output | HiSPi serial data, lane 3, differential P. |
| SLVS3_N | Output | HiSPi serial data, lane 3, differential N. |
| SLVSC_P | Output | HiSPi serial DDR clock differential P. |
| SLVSC_N | Output | HiSPi serial DDR clock differential N. |
| DATA1_P | Output | MIPI serial data, lane 1, differential P |
| DATA1_N | Output | MIPI serial data, lane 1, differential N |
| DATA2_P | Output | MIPI serial data, lane 2, differential P |
| DATA2_N | Output | MIPI serial data, lane 2, differential N |
| DATA3_P | Output | MIPI serial data, lane 3, differential P |
| DATA3_N | Output | MIPI serial data, lane 3, differential N |
| DATA4_P | Output | MIPI serial data, lane 4, differential P |
| DATA4_N | Output | MIPI serial data, lane 4, differential N |
| CLK_P | Output | Output MIPI serial clock, differential P |
| CLK_N | Output | Output MIPI serial clock, differential N |

**Table 5: Pin Descriptions (continued)**

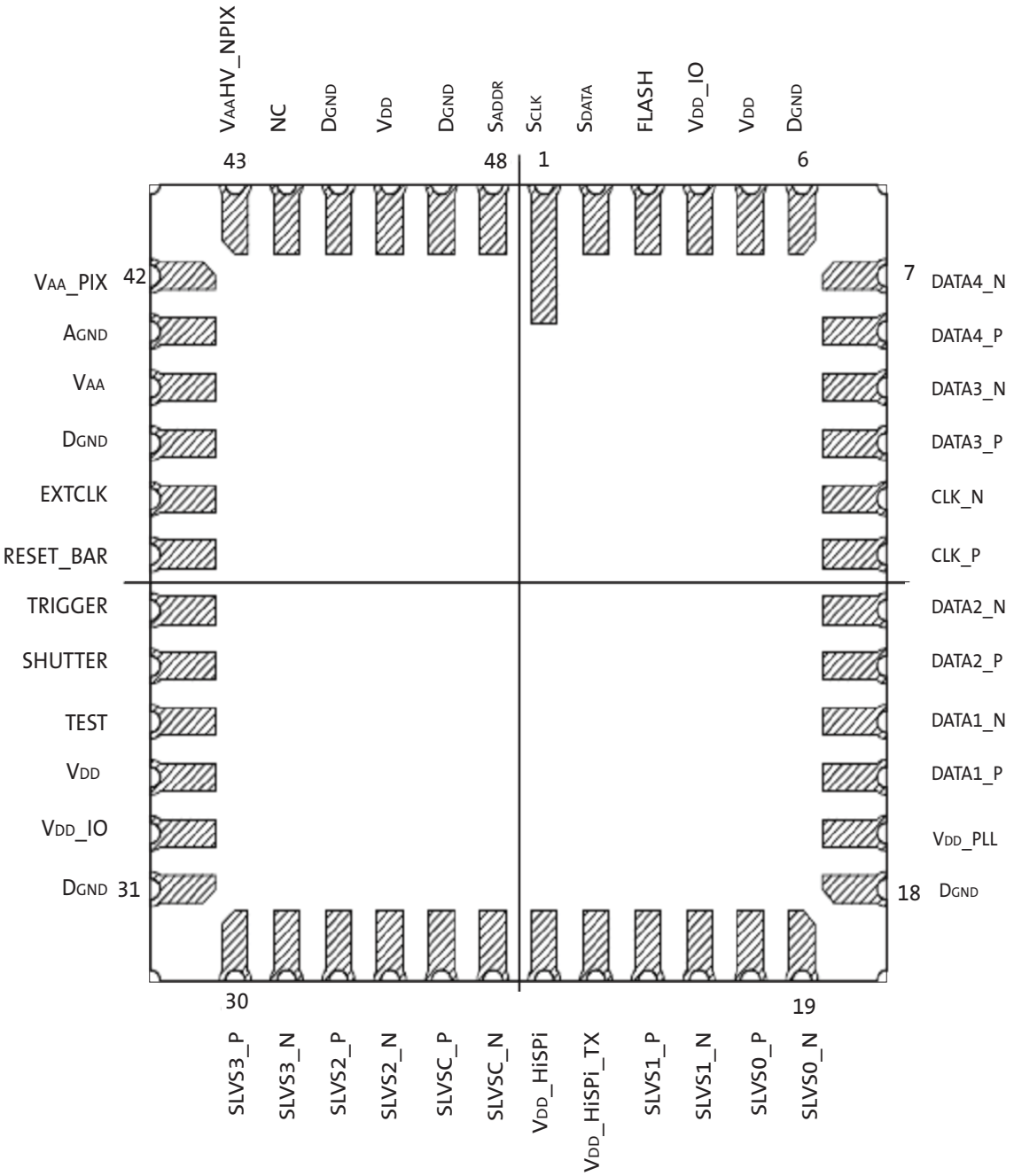
| Name | Type | Description |
|---|---|---|
| VDD_HiSPi | Power | 1.8V power port to HiSPi digital logic |
| VDD_HiSPi_TX | Power | 0.4V-0.8V or 1.7V - 1.9V Refer to "HiSPi Power Supply Connections" on page 11. |
| VAAHV_NPIX | Power | Power supply pin used to program the sensor OTPM (one-time programmable memory). This pin should be open if OTPM is not used. |

**Table 6: CSP (HiSPi/MIPI) Package Pinout**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **A** | VAA | VAAHV_NPIX | AGND | AGND | VAA | VDD | TEST | DGND |
| **B** | DGND | NC | VAA_PIX | DGND | VDD_IO | TRIGGER | RESET_BAR | EXTCLK |
| **C** | VDD | SHUTTER | DGND | SLVSC_P | SLVS3_P | SLVS3_N | SLVS2_N | SLVS2_P |
| **D** | SADDR | SCLK | SDATA | FLASH | SLVSC_N | SLVS1_P | VDD_HiSPi_TX | VDD_HiSPi |
| **E** | VDD_IO | VDD_IO | CLK_N | CLK_P | DGND | SLVS1_N | SLVS0_N | SLVS0_P |
| **F** | DGND | VDD_IO | DGND | DGND | DATA4_P | DATA1_N | DATA_1P | VDD_PLL |
| **G** | VDD_IO | VDD | DGND | VDD_IO | DATA4_N | DATA3_N | DATA2_N | VDD |
| **H** | DGND | VDD_IO | VDD_IO | DGND | VDD_PLL | DATA3_P | DATA2_P | VDD_PLL |

Note: NC = No Connection.

**Figure 5: CLCC Package Pin Descriptions**




Note: Pins labeled NC (Not Connected) should be tied to ground

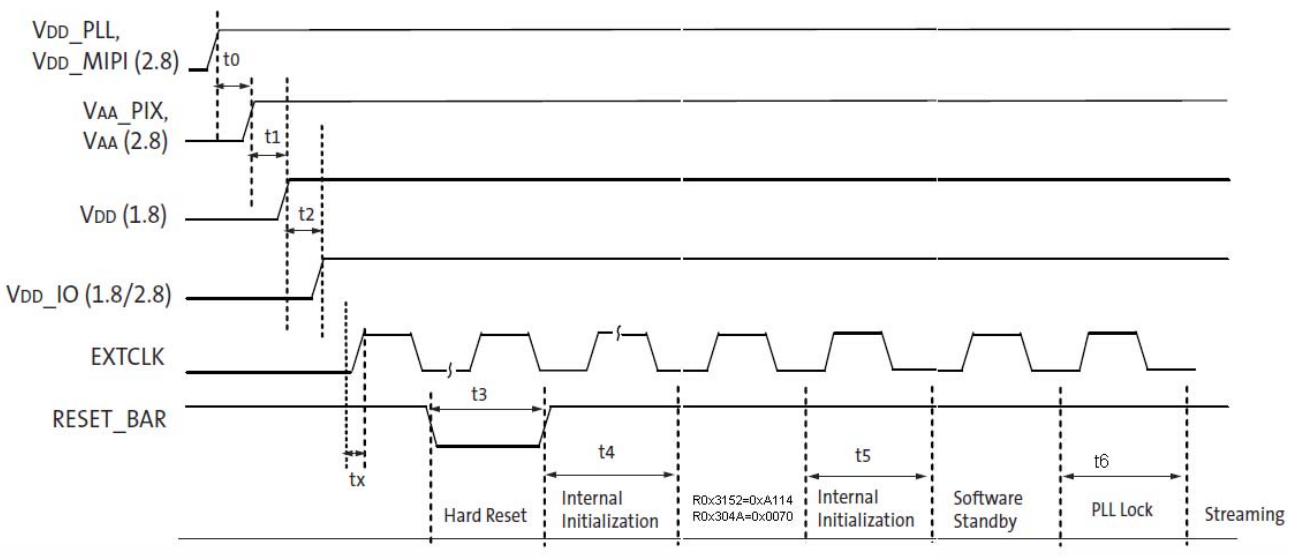
# Sensor Initialization

## Power-Up Sequence

The recommended power-up sequence for the AR0330CS is shown in Figure 6. The available power supplies (VDD_IO, VDD_PLL, VDD_MIPI, VAA, VAA_PIX) must have the separation specified below.

1. Turn on VDD_PLL and VDD_MIPI power supplies
2. After 100µs, turn on VAA and VAA_PIX power supply.
3. After 100µs, turn on VDD power supply.
4. After 100µs, turn on VDD_IO power supply.
5. After the last power supply is stable, enable EXTCLK.
6. Assert RESET_BAR for at least 1ms.
7. Wait 150,000 EXTCLK periods (for internal initialization into software standby.
8. Write R0x3152 = 0xA114 to configure the internal register initialization process.
9. Write R0x304A = 0x0070 to start the internal register initialization process.
10. Wait 150,000 EXTCLK periods
11. Configure PLL, output, and image settings to desired values.
12. Wait 1ms for the PLL to lock.
13. Set streaming mode (R0x301A[2] = 1).

**Figure 6: Power Up**




Notes:
1. A software reset (R0x301A[0] = 1) is not necessary after the procedure described above since a Hard Reset will automatically triggers a software reset. Independently executing a software reset, should be followed by steps seven through thirteen above
2. The sensor must be receiving the external input clock (EXTCLK) before the reset pin is toggled. The sensor will begin an internal initialization sequence when the reset pin toggle from LOW to HIGH. This initialization sequence will run using the external input clock. Power on default state is software standby state, need to apply two-wire serial commands to start streaming. Above power up sequence is a general power up sequence. For different interface configurations, MIPI, and Parallel, some power rails are not needed. Those not needed power rails should be ignored in the general power up sequence.

**Table 7: Power-Up Sequence**

| Definition | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{DD}$_PLL, $V_{DD}$_MIPI to $V_{AA}$/$V_{AA}$_PIX[3] | t0 | 0 | 100 | – | µs |
| $V_{AA}$/$V_{AA}$_PIX to $V_{DD}$ | t1 | 0 | 100 | – | µs |
| $V_{DD}$ to $V_{DD}$_IO | t2 | 0 | 100 | – | µs |
| External clock settling time | tx | – | 30[1] | – | ms |
| Hard Reset | t3 | 1[2] | – | – | ms |
| Internal Initialization | t4 | 150000 | – | – | EXTCLKs |
| Internal Initialization | t5 | 150000 | – | – | EXTCLKs |
| PLL Lock Time | t6 | 1 | – | – | ms |

Notes:
1. External clock 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 $V_{DD}$_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 $V_{DD}$_PLL is powered after other supplies then sensor may have functionality issues and will experience high current draw on this supply.
4. $V_{DD}$_MIPI is tied to $V_{DD}$_PLL in the both the CLCC and CSP packages and must be powered to 2.8 V. The $V_{DD}$_HiSPi and $V_{DD}$_HiSPi_TX supplies do not need to be turned on if the sensor is configured to use the MIPI or parallel interface.

## Power-Down Sequence

The recommended power-down sequence for the AR0330 is shown in Figure 7. The available power supplies (VDD_IO, VDD_HiSPi, VDD_HiSPi_TX, VDD_PLL, VDD_MIPI, 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_HiSPi_TX.
4. Turn off VDD_IO.
5. Turn off VDD and VDD_HiSPi
6. Turn off VAA/VAA_PIX.
7. Turn off VDD_PLL, VDD_MIPI.

**Figure 7: Power Down**

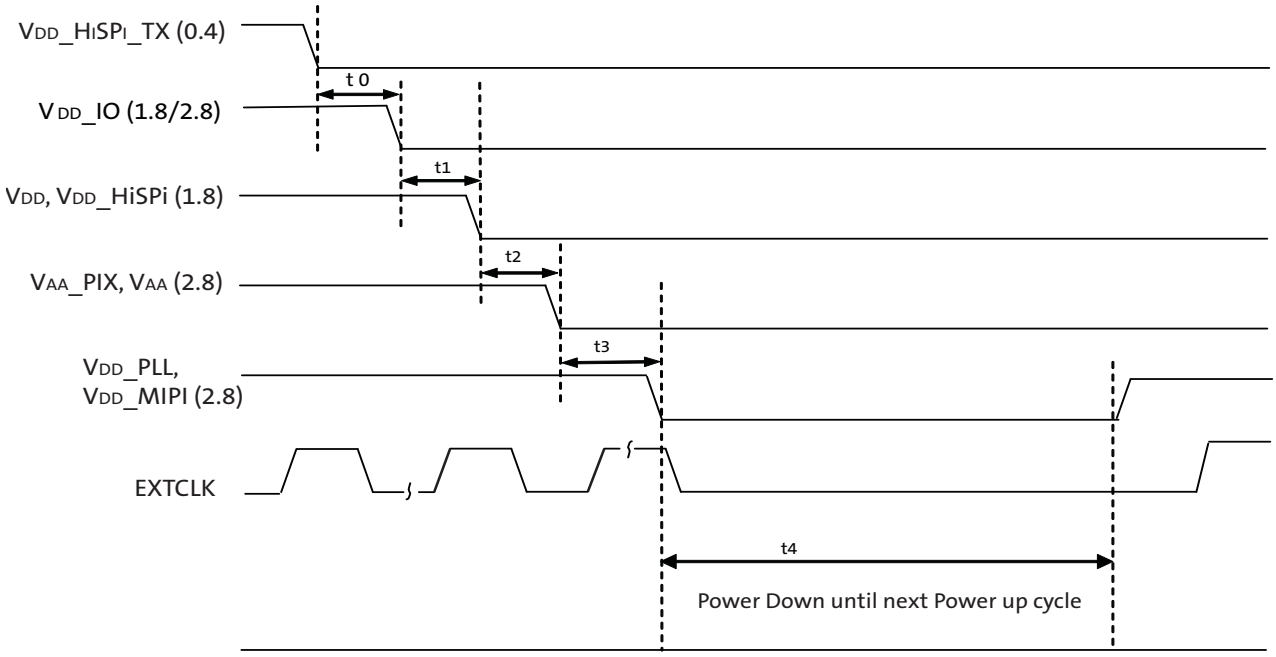



**Table 8: Power-Down Sequence**

| Definition | Symbol | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| VDD_HiSPi _TX to VDD_IO | t0 | 0 | – | – | µs |
| VDD_IO to VDD and VDD_HiSPi | t1 | 0 | – | – | µs |
| VDD and VDD_HiSPi 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.

# Electrical Characteristics

**Table 9: DC Electrical Definitions and Characteristics (MIPI Mode)**

$f_{EXTCLK}$ = 24 MHz; $V_{DD}$ = 1.8V; $V_{DD\_IO}$ = 1.8V; $V_{AA}$ = 2.8V; $V_{AA\_PIX}$ = 2.8V; $V_{DD\_PLL}$ = 2.8V; Output load = 68.5pF; $T_J$ = 60°C; Data Rate =588 Mbps; 2304x1296 at 60 fps

| Definition | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Core digital voltage | $V_{DD}$ | 1.7 | 1.8 | 1.9 | V |
| I/O digital voltage | $V_{DD\_IO}$ | 1.7 | 1.8 | 1.9 | V |
| | | 2.4 | 2.8 | 3.1 | V |
| Analog voltage | $V_{AA}$ | 2.7 | 2.8 | 2.9 | V |
| Pixel supply voltage | $V_{AA\_PIX}$ | 2.7 | 2.8 | 2.9 | V |
| PLL supply voltage | $V_{DD\_PLL}$ | 2.7 | 2.8 | 2.9 | V |
| MIPI supply voltage | $V_{DD\_MIPI}$ | 2.7 | 2.8 | 2.9 | V |
| Digital operating current | I($V_{DD}$) | | 114 | 136 | mA |
| I/O digital operating current | I($V_{DD\_IO}$) | | 0 | 0 | mA |
| Analog operating current | I($V_{AA}$) | | 41 | 53 | mA |
| Pixel supply current | I($V_{AA\_PIX}$) | | 9.9 | 12 | mA |
| PLL supply current | I($V_{DD\_PLL}$) | | 15 | 27 | mA |
| MIPI digital operating current | I($V_{DD\_MIPI}$) | | 35 | 49 | mA |

**Table 10: DC Electrical Definitions and Characteristics (HiSPi Mode)**

$f_{EXTCLK}$ = 24 MHz; $V_{DD}$ = 1.8V; $V_{DD\_IO}$ = 1.8V; $V_{AA}$ = 2.8V; $V_{AA\_PIX}$ = 2.8V; $V_{DD\_PLL}$ = 2.8V; $V_{DD\_HiSPi}$ = 1.8V, $V_{DD\_HiSPi\_TX}$ = 0.4V; Output load = 68.5pF; $T_J$ = 60°C; Data Rate =588 Mbps; DLL set to 0; 2304x1296 at 60 fps

| Definition | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Core digital voltage | $V_{DD}$ | 1.7 | 1.8 | 1.9 | V |
| I/O digital voltage | $V_{DD\_IO}$ | 1.7 | 1.8 | 1.9 | V |
| | | 2.4 | 2.8 | 3.1 | V |
| Analog voltage | $V_{AA}$ | 2.7 | 2.8 | 2.9 | V |
| Pixel supply voltage | $V_{AA\_PIX}$ | 2.7 | 2.8 | 2.9 | V |
| PLL supply voltage | $V_{DD\_PLL}$ | 2.7 | 2.8 | 2.9 | V |
| HiSPi digital voltage | $V_{DD\_HiSPi}$ | 1.7 | 1.8 | 1.9 | V |
| HiSPi I/O digital voltage | $V_{DD\_HiSPi\_TX}$ | 0.3 | 0.4 | 0.9 | V |
| | | 1.7 | 1.8 | 1.9 | V |
| Digital operating current | I($V_{DD}$) | | 96.3 | 137 | mA |
| I/O digital operating current | I($V_{DD\_IO}$) | | 0 | 0 | mA |
| Analog operating current | I($V_{AA}$) | | 45.1 | 53 | mA |
| Pixel supply current | I($V_{AA\_PIX}$) | | 10.5 | 12 | mA |
| PLL supply current | I($V_{DD\_PLL}$) | | 6.4 | 11 | mA |
| HiSPi digital operating current | I($V_{DD\_HiSPi}$) | | 21.8 | 36 | mA |
| HiSPi I/O digital operating current | I($V_{DD\_HiSPi\_TX}$) | | 22.3 | 40 | mA |

**Table 11: DC Electrical Definitions and Characteristics (Parallel Mode)**

$f_{EXTCLK}$ = 24 MHz; $V_{DD}$ = 1.8 V; $V_{DD\_IO}$ = 1.8 V; $V_{AA}$ = 2.8 V; $V_{AA\_PIX}$ = 2.8 V; $V_{DD\_PLL}$ = 2.8 V;
Output load = 68.5 pF; $T_J$ = 60°C; 2304 x 1296 at 30 fps

| Definition | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Core digital voltage | $V_{DD}$ | 1.7 | 1.8 | 1.9 | V |
| I/O digital voltage | $V_{DD\_IO}$ | 1.7 | 1.8 | 1.9 | V |
| | | 2.4 | 2.8 | 3.1 | V |
| Analog voltage | $V_{AA}$ | 2.7 | 2.8 | 2.9 | V |
| Pixel supply voltage | $V_{AA\_PIX}$ | 2.7 | 2.8 | 2.9 | V |
| PLL supply voltage | $V_{DD\_PLL}$ | 2.7 | 2.8 | 2.9 | V |
| Digital operating current | I($V_{DD}$) | | 66.5 | 75 | mA |
| I/O digital operating current | I($V_{DD\_IO}$) | | 24 | 35 | mA |
| Analog operating current | I($V_{AA}$) | | 36 | 44 | mA |
| Pixel supply current | I($V_{AA\_PIX}$) | | 10.5 | 18 | mA |
| PLL supply current | I($V_{DD\_PLL}$) | | 6 | 11 | mA |

**Table 12: Standby Power**

$f_{EXTCLK}$ = 24 MHz; $V_{DD}$ = 1.8 V; $V_{DD\_IO}$ = 1.8 V; $V_{AA}$ = 2.8 V; $V_{AA\_PIX}$ = 2.8 V; $V_{DD\_PLL}$ = 2.8 V; Output load = 68.5 pF; $T_J$ = 60°C

| | Power | Typical | Max | Unit |
|---|---|---|---|---|
| Hard Standby (CLK OFF) | Digital | 19.8 | 35.8 | µA |
| | Analog | 5.8 | 7.0 | µA |
| Soft Standby (CLK OFF) | Digital | 23.5 | 39.7 | µA |
| | Analog | 5.4 | 5.9 | µA |
| Soft Standby (CLK ON) | Digital | 15700 | 16900 | µA |
| | Analog | 5.5 | 5.7 | µA |

**Caution** **Stresses greater than those listed in Table 13 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 13: Absolute Maximum Ratings**

| Symbol | Definition | Min | Max | Unit |
|---|---|---|---|---|
| $V_{DD\_MAX}$ | Core digital voltage | –0.3 | 2.4 | V |
| $V_{DD\_IO\_MAX}$ | I/O digital voltage | –0.3 | 4 | V |
| $V_{AA\_MAX}$ | Analog voltage | –0.3 | 4 | V |
| $V_{AA\_PIX}$ | Pixel supply voltage | –0.3 | 4 | V |
| $V_{DD\_PLL}$ | PLL supply voltage | –0.3 | 4 | V |
| $V_{DD\_MIPI}$ | MIPI supply voltage | –0.3 | 4 | V |
| $V_{DD\_HiSPi\_MAX}$ | HiSPi digital voltage | –0.3 | 2.4 | V |
| $V_{DD\_HiSPi\_TX\_MAX}$ | 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.

## Two-Wire Serial Register Interface

The electrical characteristics of the two-wire serial register interface (SCLK, SDATA) are shown in Figure 8 and Table 14.

**Figure 8: 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 14: Two-Wire Serial Bus Characteristics**

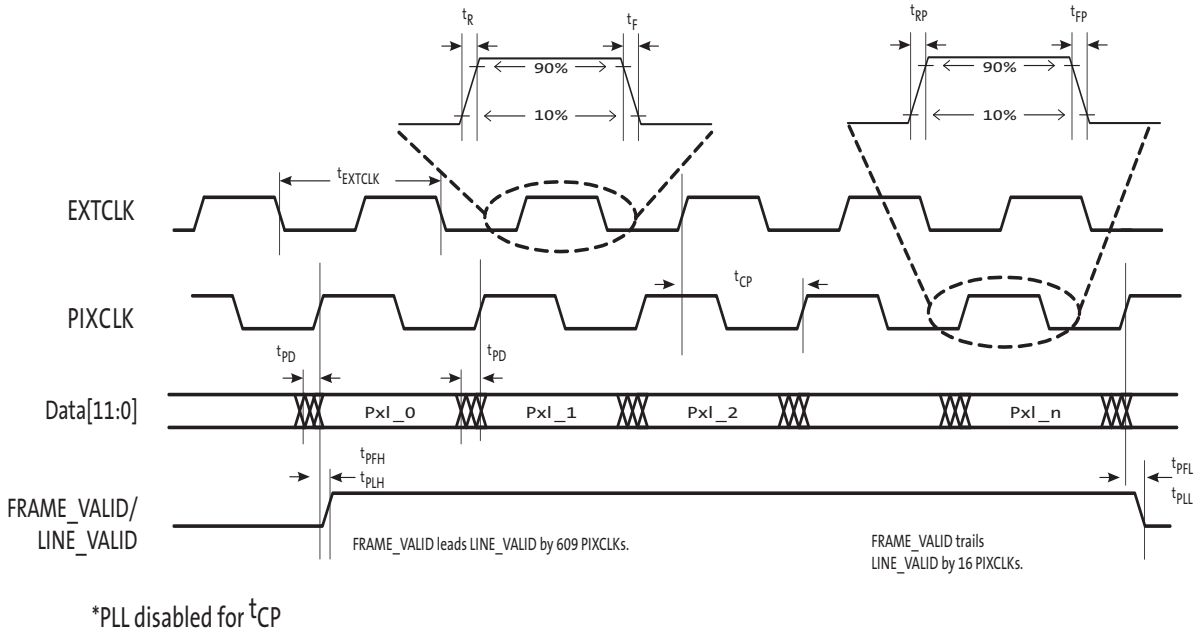
$^{f}$EXTCLK = 27 MHz; $V_{DD}$ = 1.8V; $V_{DD}$_IO = 2.8V; $V_{AA}$ = 2.8V; $V_{AA}$_PIX = 2.8V; $V_{DD}$_PLL = 2.8V; $T_A$ = 25°C

| 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 |
| SDATA pull-up resistor | RSD | 1.5 | 4.7 | 1.5 | 4.7 | KΩ |

Notes:
1. This table is based on I$^2$C standard (v2.1 January 2000). Philips Semiconductor.
2. Two-wire control is I$^2$C-compatible.
3. All values referred to $V_{IHmin}$ = 0.9 $V_{DD}$ and $V_{ILmax}$ = 0.1$V_{DD}$ 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$^2$C-bus device can be used in a Standard-mode I$^2$C-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$^2$C-bus specification) before the SCLK line is released.
7. Cb = total capacitance of one bus line in pF.

**Figure 9: I/O Timing Diagram (Parallel Mode)**




*PLL disabled for $^{t}$CP

**Table 15: I/O Parameters**

$^{f}$EXTCLK = 24 MHz; VDD = 1.8V; VAA = 2.8V; VAA_PIX = 2.8V; VDD_PLL = 2.8V; Output load = 68.5pF; $T_J$ = 60°C; CLK_OP = 98 MPixel/s

| Symbol | Definition | Conditions | Min | Max | Units |
|---|---|---|---|---|---|
| VIH | Input HIGH voltage | VDD_IO = 1.8V | 1.4 | VDD_IO + 0.3 | V |
| | | VDD_IO = 2.8V | 2.4 | | |
| VIL | Input LOW voltage | VDD_IO = 1.8V | GND – 0.3 | 0.4 | |
| | | VDD_IO = 2.8V | GND – 0.3 | 0.8 | |
| IIN | Input leakage current | No pull-up resistor; VIN = VDD or DGND | – 20 | 20 | µA |
| VOH | Output HIGH voltage | At specified IOH | VDD_IO - 0.4V | – | V |
| VOL | Output LOW voltage | At specified IOL | – | 0.4 | V |
| IOH | Output HIGH current | At specified VOH | – | –12 | mA |
| IOL | Output LOW current | At specified VOL | – | 9 | mA |
| IOZ | Tri-state output leakage current | | – | 10 | µA |

**Table 16: I/O Timing**

$^{f}$EXTCLK = 24 MHz; $V_{DD}$ = 1.8V; $V_{DD}$_IO = 1.8V; $V_{AA}$ = 2.8V; $V_{AA}$_PIX = 2.8V; $V_{DD}$_PLL = 2.8V; Output load = 68.5pF; $T_J$ = 60°C; CLK_OP = 98 MPixel/s

| Symbol | Definition | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $^{f}$EXTCLK | Input clock frequency | PLL enabled | 6 | 24 | 27 | MHz |
| $^{t}$EXTCLK | Input clock period | PLL enabled | 166 | 41 | 20 | ns |
| $^{t}$R | Input clock rise time | | 0.5 | – | Sine wave rise time | ns |
| $^{t}$F | Input clock fall time | | 0.5 | – | Sine wavefall time | ns |
| | Clock duty cycle | | 45 | 50 | 55 | % |
| $^{t}$JITTER | Input clock jitter | | – | – | 0.3 | ns |
| Output pin slew | Fastest | $C_{LOAD}$ = 15pF | – | 0.7 | – | V/ns |
| $^{f}$PIXCLK | PIXCLK frequency | Default | – | 80 | – | MHz |
| $^{t}$PD | PIXCLK to data valid | Default | – | – | 3 | ns |
| $^{t}$PFH | PIXCLK to FRAME_VALID HIGH | Default | – | – | 3 | ns |
| $^{t}$PLH | PIXCLK to LINE_VALID HIGH | Default | – | – | 3 | ns |
| $^{t}$PFL | PIXCLK to FRAME_VALID LOW | Default | – | – | 3 | ns |
| $^{t}$PLL | PIXCLK to LINE_VALID LOW | Default | – | – | 3 | ns |

**Table 17: Parallel I/O Rise Slew Rate**

$^{f}$EXTCLK = 24 MHz; $V_{DD}$ = 1.8V; $V_{AA}$ = 2.8V; $V_{AA}$_PIX = 2.8V; $V_{DD}$_PLL = 2.8V; Output load = 68.5pF; $T_J$ = 60°C; CLK_OP = 98 MPixel/s

| | Parallel Slew Rate (R0x306E[15:13]) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $V_{DD}$_IO | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Units |
| 1.70V | 0.069 | 0.115 | 0.172 | 0.239 | 0.325 | 0.43 | 0.558 | 0.836 | V/ns |
| 1.80V | 0.078 | 0.131 | 0.195 | 0.276 | 0.375 | 0.507 | 0.667 | 1.018 | |
| 1.95V | 0.093 | 0.156 | 0.233 | 0.331 | 0.456 | 0.62 | 0.839 | 1.283 | |
| 2.50V | 0.15 | 0.252 | 0.377 | 0.539 | 0.759 | 1.07 | 1.531 | 2.666 | |
| 2.80V | 0.181 | 0.305 | 0.458 | 0.659 | 0.936 | 1.347 | 1.917 | 3.497 | |
| 3.10V | 0.212 | 0.361 | 0.543 | 0.78 | 1.114 | 1.618 | 2.349 | 4.14 | |

# HiSPi Transmitter

**Note:** Refer to "High-Speed Serial Pixel Interface Physical Layer Specification v2.00.00" for further explanation of the HiSPi transmitter specification.

## SLVS Electrical Specifications

**Table 18: Power Supply and Operating Temperature**

| Parameter | Symbol | Min | Typ | Max | Unit | Notes |
|---|---|---|---|---|---|---|
| SLVS Current Consumption | $I_{DD\_HiSPi\_TX}$ | | | n*18 | mA | 1, 2 |
| HiSPi PHY Current Consumption | $I_{DD\_HiSPi}$ | | | n*45 | mA | 1, 2, 3 |
| Operating temperature | $T_J$ | -30 | | 70 | °C | 4 |

Notes:
1. Where 'n' is the number of PHYs
2. Temperature of 25°C
3. Up to 700 Mbps
4. Specification values may be exceeded when outside this temperature range.

**Table 19: SLVS Electrical DC Specification**
Tj = 25°C

| Parameter | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| SLVS DC mean common mode voltage | $V_{CM}$ | 0.45*$V_{DD\_TX}$ | 0.5*$V_{DD\_TX}$ | 0.55*$V_{DD\_TX}$ | V |
| SLVS DC mean differential output voltage | $\|V_{OD}\|$ | 0.36*$V_{DD\_TX}$ | 0.5*$V_{DD\_TX}$ | 0.64*$V_{DD\_TX}$ | V |
| Change in $V_{CM}$ between logic 1 and 0 | $\Delta V_{CM}$ | | | 25 | mV |
| Change in $\|V_{OD}\|$ between logic 1 and 0 | $\|\Delta V_{OD}\|$ | | | 25 | mV |
| $V_{OD}$ noise margin | NM | | | ±30 | % |
| Difference in $V_{CM}$ between any two channels | $\|\Delta V_{CM}\|$ | | | 50 | mV |
| Difference in $V_{OD}$ between any two channels | $\|\Delta V_{OD}\|$ | | | 100 | mV |
| Common-mode AC Voltage (pk) without VCM cap termination | $V_{CM\_AC}$ | | | 50 | mV |
| Common-mode AC Voltage (pk) with VCM cap termination | $V_{CM\_AC}$ | | | 30 | mV |
| Maximum overshoot peak $\|V_{OD}\|$ | $V_{OD\_AC}$ | | | 1.3*$\|V_{OD}\|$ | V |
| Maximum overshoot $V_{diff\ pk\text{-}pk}$ | $V_{diff\_pkpk}$ | | | 2.6*$V_{OD}$ | V |
| Single-ended Output impedance | $R_O$ | 35 | 50 | 70 | Ω |
| Output Impedance Mismatch | $\Delta R_O$ | | | 20 | % |

**Table 20: SLVS Electrical Timing Specification**

| Parameter | Symbol | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| Data Rate | 1/UI | 280 | 700 | Mbps | 1 |
| Bitrate Period | $t_{PW}$ | 1.43 | 3.57 | ns | 1 |
| Max setup time from transmitter | $t_{PRE}$ | 0.3 | | UI | 1, 2 |
| Max hold time from transmitter | $t_{POST}$ | 0.3 | | UI | 1, 2 |
| Eye Width | $t_{EYE}$ | | 0.6 | UI | 1, 2 |
| Data Total Jitter (pk-pk) @1e-9 | $t_{TOTALJIT}$ | | 0.2 | UI | 1, 2 |
| Clock Period Jitter (RMS) | $t_{CKJIT}$ | | 50 | ps | 2 |
| Clock Cycle-to-Cycle Jitter (RMS) | $t_{CYCJIT}$ | | 100 | ps | 2 |
| Rise time (20% - 80%) | tR | 150ps | 0.25 | UI | 3 |
| Fall time (20% - 80%) | tF | 150ps | 0.25 | UI | 3 |
| Clock duty cycle | DCYC | 45 | 55 | % | 2 |
| Mean Clock to Data Skew | $t_{CHSKEW}$ | -0.1 | 0.1 | UI | 1, 4 |
| PHY-to-PHY Skew | $t_{PHYSKEW}$ | | 2.1 | UI | 1, 5 |
| Mean differential skew | $t_{DIFFSKEW}$ | -100 | 100 | ps | 6 |

Notes:
1. One UI is defined as the normalized mean time between one edge and the following edge of the clock.
2. Taken from the 0V crossing point with the DLL off.
3. Also defined with a maximum loading capacitance of 10 pF on any pin. The loading capacitance may also need to be less for higher bitrates so the rise and fall times do not exceed the maximum 0.3 UI.
4. The absolute mean skew between the Clock lane and any Data Lane in the same PHY between any edges.
5. The absolute skew between any Clock in one PHY and any Data lane in any other PHY between any edges.

Differential skew is defined as the skew between complementary outputs. It is measured as the absolute time between the two complementary edges at mean $V_{CM}$ point. Note that differential skew also is related to the $\Delta$VCM_AC spec which also must not be exceeded.

## HiVCM Electrical Specifications

The HiSPi 2.0 specification also defines an alternative signaling level mode called HiVCM. Both $V_{OD}$ and $V_{CM}$ are still scalable with $V_{DD}$_HiSPi_TX, but with $V_{DD}$_HiSPi_TX nominal set to 1.8 V the common-mode is elevated to around 0.9 V.

**Table 21: HiVCM Power Supply and Operating Temperatures**

| Parameter | Symbol | Min | Typ | Max | Unit | Notes |
|---|---|---|---|---|---|---|
| HiVCM Current Consumption | $I_{DD}$_HiSPi_TX | | | n*34 | mA | 1, 2 |
| HiSPi PHY Current Consumption | $I_{DD}$_HiSPi | | | n*45 | mA | 1, 2, 3 |
| Operating temperature | $T_J$ | -30 | | 70 | °C | 4 |

Notes:
1. Where 'n' is the number of PHYs
2. Temperature of 25°C
3. Up to 700 Mbps
4. Specification values may be exceeded when outside this temperature range.

**Table 22: HiVCM Electrical Voltage and Impedance Specification**
Tj = 25° C

| Parameter | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| HiVCM DC mean common mode voltage | $V_{CM}$ | 0.76 | 0.90 | 1.07 | V |
| HiVCM DC mean differential output voltage | $\|V_{OD}\|$ | 200 | 280 | 350 | mV |
| Change in $V_{CM}$ between logic 1 and 0 | $\Delta V_{CM}$ | | | 25 | mV |
| Change in $\|V_{OD}\|$ between logic 1 and 0 | $\Delta \|V_{OD}\|$ | | | 25 | mV |
| $V_{OD}$ noise margin | NM | | | ±30 | % |
| Difference in $V_{CM}$ between any two channels | $\|\Delta V_{CM}\|$ | | | 50 | mV |
| Difference in $V_{OD}$ between any two channels | $\|\Delta V_{OD}\|$ | | | 100 | mV |
| Common-mode AC Voltage (pk) without $V_{CM}$ cap termination | $\Delta V_{CM\_AC}$ | | | 50 | mV |
| Common-mode AC Voltage (pk) with $V_{CM}$ cap termination | $\Delta V_{CM\_AC}$ | | | 30 | mV |
| Maximum overshoot peak $\|V_{OD}\|$ | $V_{OD\_AC}$ | | | $1.3*\|V_{OD}\|$ | V |
| Maximum overshoot Vdiff pk-pk | $V_{diff\_pkpk}$ | | | $2.6*V_{OD}$ | V |
| Single-ended Output impedance | $R_O$ | 40 | 70 | 100 | Ω |
| Output Impedance Mismatch | $\Delta R_O$ | | | 20 | % |

**Table 23: HiVCM Electrical AC Specification**

| Parameter | Symbol | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| Data Rate | 1/UI | 280 | 700 | Mbps | 1 |
| Bitrate Period | $t_{PW}$ | 1.43 | 3.57 | ns | 1 |
| Max setup time from transmitter | $t_{PRE}$ | 0.3 | | UI | 1, 2 |
| Max hold time from transmitter | $t_{POST}$ | 0.3 | | UI | 1, 2 |
| Eye Width | $t_{EYE}$ | | 0.6 | UI | 1, 2 |
| Data Total Jitter (pk-pk) @1e-9 | $t_{TOTALJIT}$ | | 0.2 | UI | 1, 2 |
| Clock Period Jitter (RMS) | $t_{CKJIT}$ | | 50 | ps | 2 |
| Clock Cycle-to-Cycle Jitter (RMS) | $t_{CYCJIT}$ | | 100 | ps | 2 |
| Rise time (20% - 80%) | $t_R$ | 150ps | 0.3 | UI | 3 |
| Fall time (20% - 80%) | $t_F$ | 150ps | 0.3 | UI | 3 |
| Clock duty cycle | $D_{CYC}$ | 45 | 55 | % | 2 |
| Clock to Data Skew | $t_{CHSKEW}$ | -0.1 | 0.1 | UI | 1, 4 |
| PHY-to-PHY Skew | $tP_{HYSKEW}$ | | 2.1 | UI | 1, 5 |
| Mean differential skew | $t_{DIFFSKEW}$ | -100 | 100 | ps | 6 |

Notes:
1. One UI is defined as the normalized mean time between one edge and the following edge of the clock.
2. Taken from the 0 V crossing point with the DLL off.
3. Also defined with a maximum loading capacitance of 10pF on any pin. The loading capacitance may also need to be less for higher bitrates so the rise and fall times do not exceed the maximum 0.3 UI.
4. The absolute mean skew between the Clock lane and any Data Lane in the same PHY between any edges.
5. The absolute mean skew between any Clock in one PHY and any Data lane in any other PHY between any edges.
6. Differential skew is defined as the skew between complementary outputs. It is measured as the absolute time between the two complementary edges at mean VCM point. Note that differential skew also is related to the ΔVCM_AC spec which also must not be exceeded.

## Electrical Definitions

Figure 10 is the diagram defining differential amplitude $V_{OD}$, $V_{CM,}$ and rise and fall times. To measure $V_{OD}$ and $V_{CM}$ use the DC test circuit shown in Figure 11 on page 27 and set the HiSPi PHY to constant Logic 1 and Logic 0. Measure $V_{oa}$, $V_{ob}$ and $V_{CM}$ with voltmeters for both Logic 1 and Logic 0.

**Figure 10: Single-Ended and Differential Signals**

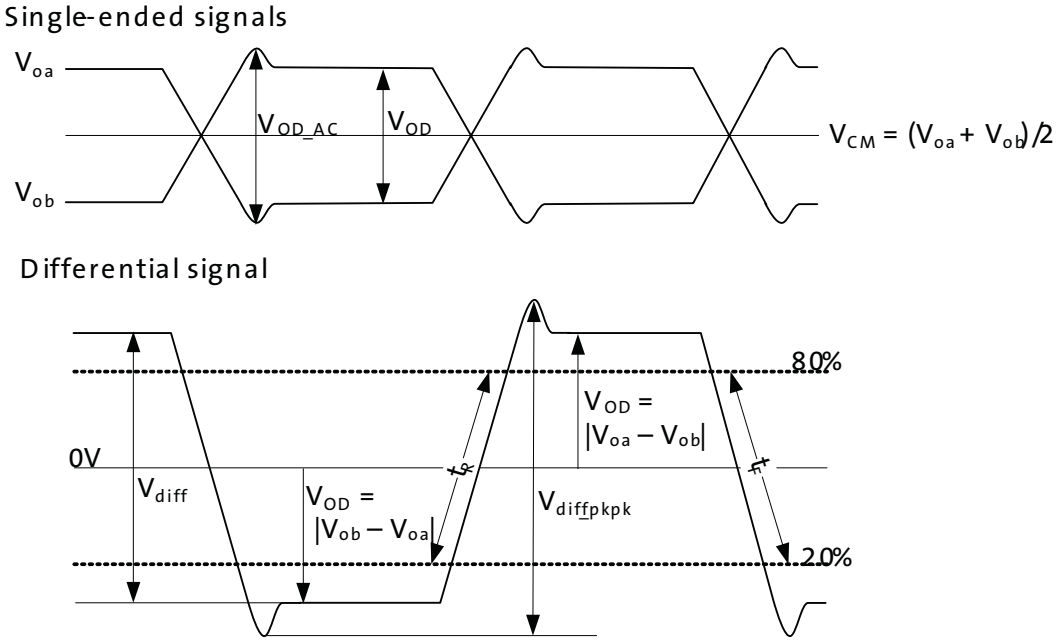
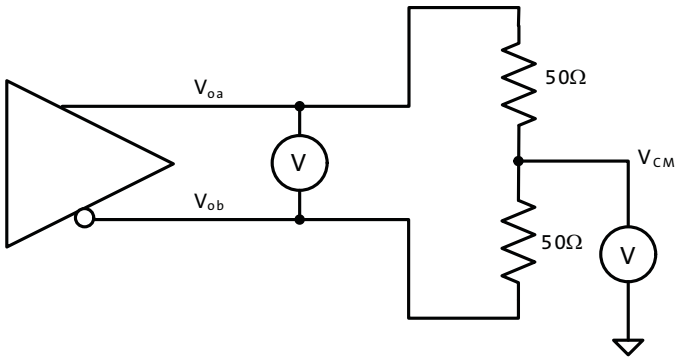



**Figure 11: DC Test Circuit**




$V_{OD}(m) = |V_{oa}(m) - V_{ob}(m)|$ where 'm' is either "1" for logic 1 or "0" for logic 0 (EQ 1)

$$V_{OD} = \frac{V_{OD}(1) + V_{OD}(0)}{2} \quad \textit{(EQ 2)}$$

$$V_{diff} = V_{OD}(1) + V_{OD}(0) \quad \text{(EQ 3)}$$

$$\Delta V_{OD} = |V_{OD}(1) - V_{OD}(0)| \quad \text{(EQ 4)}$$

$$V_{CM} = \frac{V_{CM}(1) + V_{CM}(0)}{2} \quad \textit{(EQ 5)}$$

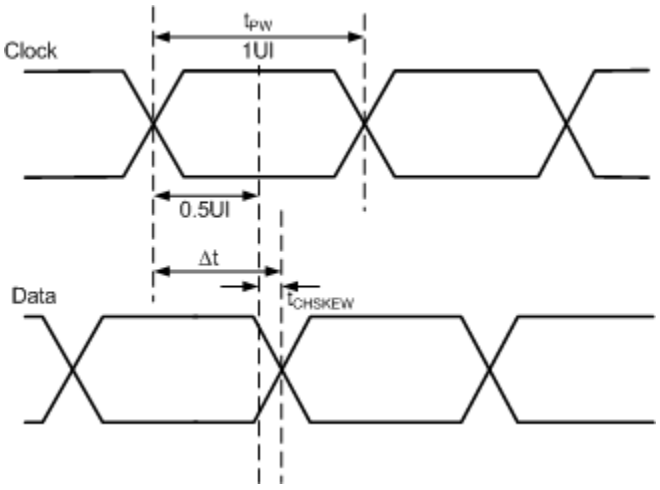
$$\Delta V_{CM} = |V_{CM}(1) - V_{CM}(0)| \quad \text{(EQ 6)}$$

Both $V_{OD}$ and $V_{CM}$ are measured for all output channels. The worst case $\Delta V_{OD}$ is defined as the largest difference in $V_{OD}$ between all channels regardless of logic level. And the worst case $\Delta V_{CM}$ is similarly defined as the largest difference in $V_{CM}$ between all channels regardless of logic level.

## Timing Definitions

1. Timing measurements are to be taken using the Square Wave test mode.
2. Rise and fall times are measured between 20% to 80% positions on the differential waveform, as shown in Figure 10: “Single-Ended and Differential Signals,” on page 27.
3. Mean Clock-to-Data skew should be measured from the 0V crossing point on Clock to the 0V crossing point on any Data channel regardless of edge, as shown in Figure 12 on page 28. This time is compared with the ideal Data transition point of 0.5UI with the difference being the Clock-to-Data Skew (see Equation 7 on page 28).

**Figure 12: Clock-to-Data Skew Timing Diagram**




$$t_{CHSKEW}(ps) = \Delta t - \frac{t_{pw}}{2} \quad \text{(EQ 7)}$$

$$t_{CHSKEW}(UI) = \frac{\Delta t}{t_{pw}} - 0.5 \quad \text{(EQ 8)}$$

4. The differential skew is measured on the two single-ended signals for any channel. The time is taken from a transition on $V_{oa}$ signal to corresponding transition on $V_{ob}$ signal at $V_{CM}$ crossing point.

**Figure 13: Differential Skew**

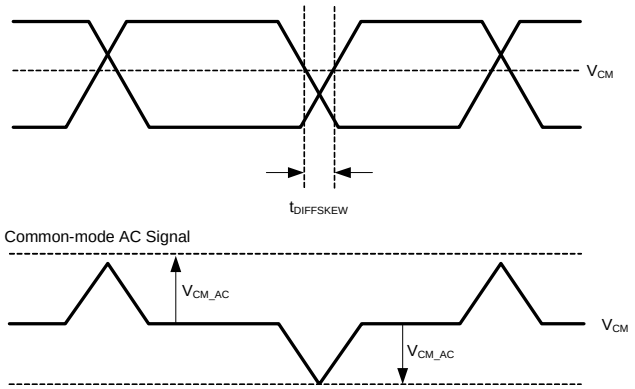



Figure 13 on page 29 also shows the corresponding AC $V_{CM}$ common-mode signal. Differential skew between the $V_{oa}$ and $V_{ob}$ signals can cause spikes in the common-mode, which the receiver needs to be able to reject. $V_{CM\_AC}$ is measured as the absolute peak deviation from the mean DC $V_{CM}$ common-mode.

## Transmitter Eye Mask

**Figure 14: Transmitter Eye Mask**

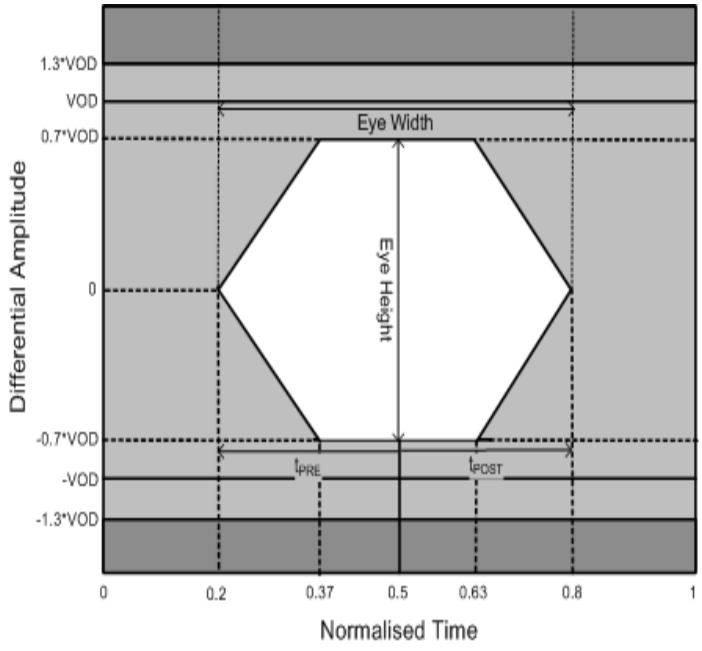
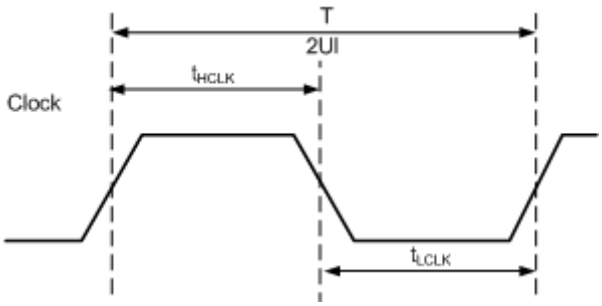



Figure 14 defines the **eye mask** for the transmitter. 0.5 UI point is the instantaneous crossing point of the Clock. The area in white shows the area Data is prohibited from crossing into. The **eye mask** also defines the minimum eye height, the data $\mathbf{t_{pre}}$ and $\mathbf{t_{post}}$ times, and the **total jitter pk-pk +mean skew** ($\mathbf{t_{TJSKEW}}$) for Data.

## Clock Signal

$t_{HCLK}$ is defined as the high clock period, and $t_{LCLK}$ is defined as the low clock period as shown in Figure 15. The clock duty cycle $D_{CYC}$ is defined as the percentage time the clock is either high ($t_{HCLK}$) or low ($t_{LCLK}$) compared with the clock period T.

**Figure 15: Clock Duty Cycle**




$$D_{CYC}(1) = \frac{t_{HCLK}}{T} \quad \text{(EQ 9)}$$

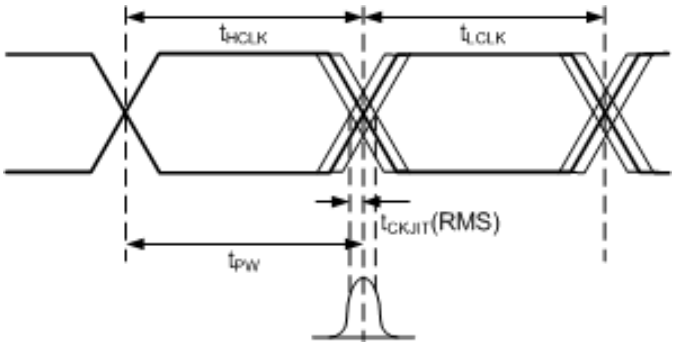
$$D_{CYC}(0) = \frac{t_{LCLK}}{T} \quad \text{(EQ 10)}$$

$$t_{pw} = \frac{T}{2} \quad \text{(i.e, 1 UI)} \quad \text{(EQ 11)}$$

$$Bitrate = \frac{1}{t_{pw}} \quad \text{(EQ 12)}$$

Figure 16 shows the definition of clock jitter for both the period and the cycle-to-cycle jitter.

**Figure 16: Clock Jitter**




Period Jitter ($t_{CKJIT}$) is defined as the deviation of the instantaneous clock $t_{PW}$ from an ideal 1UI. This should be measured for both the clock high period variation $\Delta t_{HCLK,}$ and the clock low period variation $\Delta t_{LCLK}$ taking the RMS or 1-sigma standard deviation and quoting the worse case jitter between $\Delta t_{HCLK}$ and $\Delta t_{LCLK}$.

Cycle-to-cycle jitter ($t_{CYCJIT}$) is defined as the difference in time between consecutive clock high and clock low periods $t_{HCLK}$ and $t_{LCLK}$, quoting the RMS value of the variation $\Delta(t_{HCLK} - t_{LCLK})$.

If pk-pk jitter is also measured, this should be limited to ±3-sigma.

**Table 24: HiVCM Electrical AC Specification**

| Parameter | Symbol | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| Data Rate | 1/UI | 280 | 700 | Mbps | 1 |
| Bitrate Period | $t_{PW}$ | 1.43 | 3.57 | ns | 1 |
| Max setup time from transmitter | $t_{PRE}$ | 0.3 | | UI | 1, 2 |
| Max hold time from transmitter | $t_{POST}$ | 0.3 | | UI | 1, 2 |
| Eye Width | $t_{EYE}$ | | 0.6 | UI | 1, 2 |
| Data Total Jitter (pk-pk) @1e-9 | $t_{TOTALJIT}$ | | 0.2 | UI | 1, 2 |
| Clock Period Jitter (RMS) | $t_{CKJIT}$ | | 50 | ps | 2 |
| Clock Cycle-to-Cycle Jitter (RMS) | $t_{CYCJIT}$ | | 100 | ps | 2 |
| Rise time (20% - 80%) | $t_R$ | 150ps | 0.3 | UI | 3 |
| Fall time (20% - 80%) | $t_F$ | 150ps | 0.3 | UI | 3 |
| Clock duty cycle | $D_{CYC}$ | 45 | 55 | % | 2 |
| Clock to Data Skew | $t_{CHSKEW}$ | -0.1 | 0.1 | UI | 1, 4 |
| PHY-to-PHY Skew | $tP_{HYSKEW}$ | | 2.1 | UI | 1, 5 |
| Mean differential skew | $t_{DIFFSKEW}$ | -100 | 100 | ps | 6 |

Notes:
1. One UI is defined as the normalized mean time between one edge and the following edge of the clock.
2. Taken from the 0 V crossing point with the DLL off.
3. Also defined with a maximum loading capacitance of 10pF on any pin. The loading capacitance may also need to be less for higher bitrates so the rise and fall times do not exceed the maximum 0.3 UI.
4. The absolute mean skew between the clock lane and any data lane in the same PHY between any edges.
5. The absolute mean skew between any clock in one PHY and any data lane in any other PHY between any edges.
6. Differential skew is defined as the skew between complementary outputs. It is measured as the absolute time between the two complementary edges at mean VCM point. Note that differential skew also is related to the ΔVCM_AC spec which also must not be exceeded.

## Sequencer

The sequencer digital block determines the order and timing of operations required to sample pixel data from the array during each row period. It is controlled by an instruction set that is programmed into RAM from the sensor OTPM (One Time Programmable Memory). The OTPM is configured during production.

The instruction set determines the length of the sequencer operation that determines the "ADC Readout Limitation" (Equation 5) listed in the Sensor Frame-Rate section. The instruction set can be shortened through register writes in order to achieve faster frame rates. Instructions for shortening the sequencer can be found in the AR0330 Developer Guide.

The sequencer digital block can be reprogrammed using the following instructions:

Program a new sequencer.

1. Place the sensor in standby.
2. Write 0x8000 to R0x3088 ("seq_ctrl_port").
3. Write each instruction incrementally to R0x3086. Each write must be 16-bit consisting of two bytes {Byte[N], Byte[N+1]}.
4. If the sequencer consists of an odd number of bytes, set the last byte to "0".

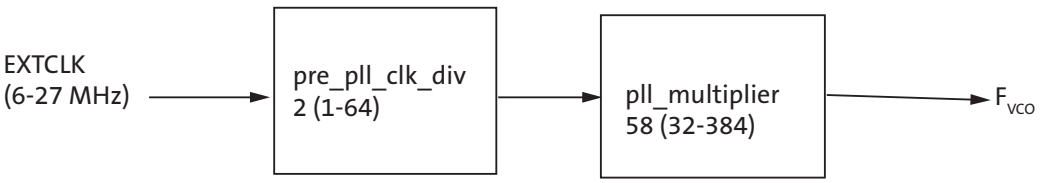
Read the instructions stored in the sequencer.

1. Place the sensor in standby.
2. Write 0xC000 to R0x3088 ("seq_ctrl_port").
3. Sequentially read one byte at a time from R0x3086 with 8-bit read command.

## Sensor PLL

### VCO

**Figure 17: Relationship Between Readout Clock and Peak Pixel Rate**

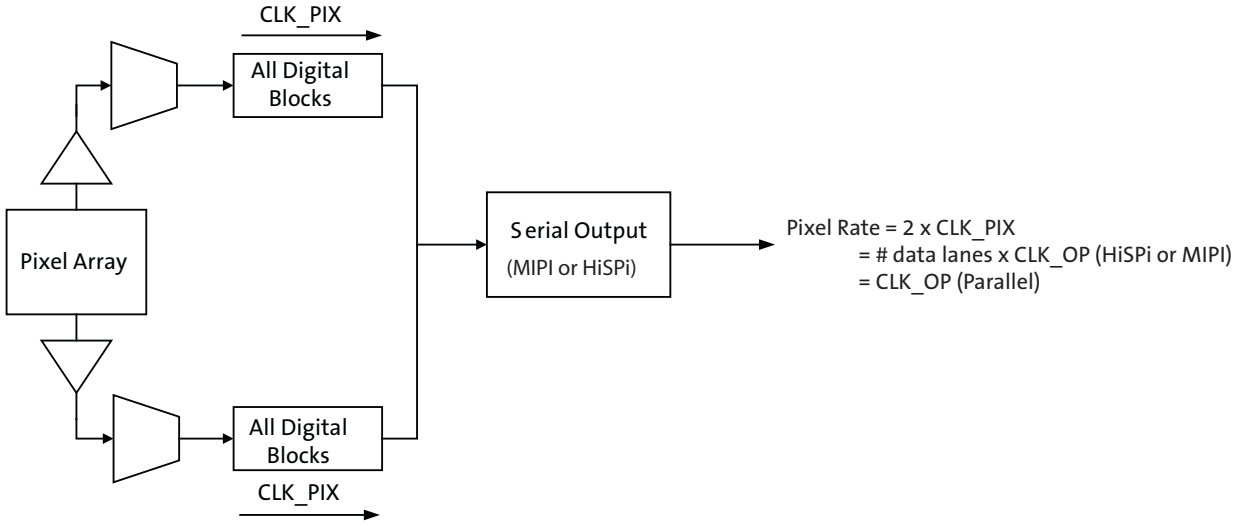



The sensor contains a phase-locked loop (PLL) that is used for timing generation and control. The required VCO clock frequency is attained through the use of a pre-PLL clock divider followed by a multiplier. The multiplier is followed by set of dividers used to generate the output clocks required for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

**Figure 18: Sensor Dual Readout Paths**

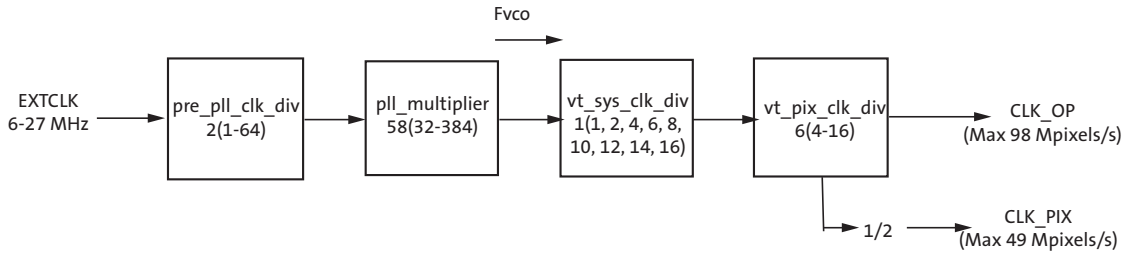



The sensor row timing calculations refers to each data-path individually. For example, the sensor default configuration uses 1248 clocks per row (line_length_pck) to output 2304 active pixels per row. The aggregate clocks per row seen by the receiver will be 2496 clocks (1248 x 2 readout paths).

## Parallel PLL Configuration

**Figure 19: PLL for the Parallel Interface**

The parallel interface has a maximum output data-rate of 98MPixel/s.




The maximum output of the parallel interface is 98 Mpixel/s (CLK_OP). This will limit the readout clock (CLK_PIX) to 49 Mpixel/s. The sensor will not use the $F_{SERIAL}$, $F_{SERIAL\_CLK}$, or CLK_OP when configured to use the parallel interface.

**Table 25: PLL Parameters for the Parallel Interface**

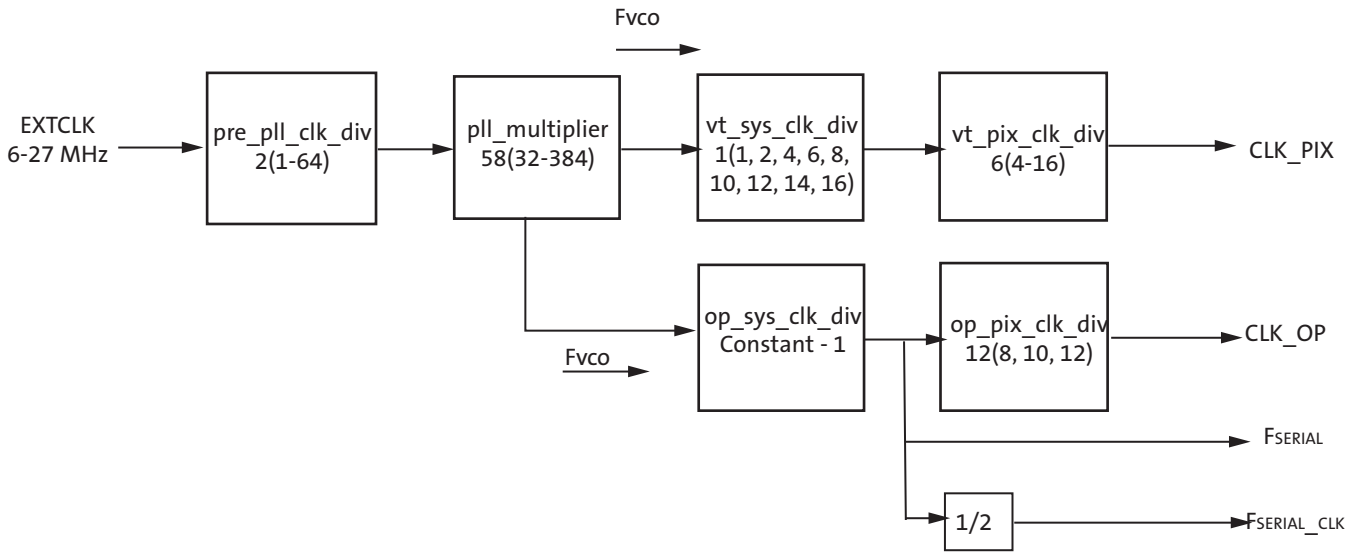
| Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| External Clock | EXTCLK | 6 | 27 | MHz |
| VCO Clock | $F_{VCO}$ | 384 | 768 | MHz |
| Readout Clock | CLK_PIX | | 49 | Mpixel/s |
| Output Clock | CLK_OP | | 98 | Mpixel/s |

**Table 26: Example PLL Configuration for the Parallel Interface**

| Parameter | Value | Output |
|---|---|---|
| $F_{VCO}$ | | 588 MHz (Max) |
| vt_sys_clk_div | 1 | |
| vt_pix_clk_div | 6 | |
| CLK_PIX | | 49 Mpixel/s (= 588 MHz / 12) |
| CLK_OP | | 98 Mpixel/s (= 588 MHz / 6) |
| Output pixel rate | | 98 MPixel/s |

## Serial PLL Configuration

**Figure 20: 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 40.

**Table 27: PLL Parameters for the Serial Interface**

| Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| External Clock | EXTCLK | 6 | 27 | MHz |
| VCO Clock | $F_{VCO}$ | 384 | 768 | MHz |
| Readout Clock | CLK_PIX | | 98 | Mpixel/s |
| Output Clock | CLK_OP | | 98 | Mpixel/s |
| Output Serial Data Rate Per Lane | $F_{SERIAL}$ | 300 (HiSPi)<br>384 (MIPI) | 700 (HiSPi)<br>768 (MIPI) | Mbps |
| Output Serial Clock Speed Per Lane | $F_{SERIAL\_CLK}$ | 150 (HiSPi)<br>192 (MIPI) | 350(HiSPi)<br>384 (MIPI) | MHz |

The serial output should be configured so that it adheres to the following rules:

- The maximum data-rate per lane ($F_{SERIAL}$) is 768Mbps/lane (MIPI) and 700Mbps/lane (HiSPi).
- The output pixel rate per lane (CLK_OP) should be configured 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: 196 Mpixel/s)
  - 2-lane: 2 x CLK_OP = 2 x CLK_PIX = Pixel Rate (max: 98 Mpixel/s)
  - 1-lane: 1 x CLK_OP = 2 x CLK_PIX = Pixel Rate (max: 76 Mpixel/s)

**Table 28: Example PLL Configurations for the Serial Interface**

| Parameter | 4-lane | | 2-lane | | 1-lane | | | Notes |
|---|---|---|---|---|---|---|---|---|
| | 12-bit | 10-bit | 12-bit | 10-bit | 12-bit | 10-bit | 8-bit | |
| $F_{VCO}$ | 588 | 490 | 588 | 490 | 768 | 768 | 768 | MHz |
| vt_sys_clk_div | 1 | 1 | 2 | 2 | 4 | 4 | 4 | |
| vt_pix_clk_div | 6 | 5 | 6 | 5 | 6 | 5 | 4 | |
| op_sys_clk_div | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| op_pix_clk_div | 12 | 10 | 12 | 10 | 12 | 10 | 8 | |
| $F_{SERIAL}$ | 588 | 490 | 588 | 490 | 768 | 768 | 768 | MHz |
| $F_{SERIAL\_CLK}$ | 294 | 245 | 294 | 245 | 384 | 384 | 384 | MHz |
| CLK_PIX | 98 | 98 | 49 | 49 | 32 | 38.4 | 48 | Mpixel/s |
| CLK_OP | 49 | 49 | 49 | 49 | 64 | 76.8 | 96 | Mpixel/s |
| Pixel Rate | 196 | 196 | 98 | 98 | 64 | 76.8 | 96 | Mpixel/s |

# 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 30 on page 36 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 29. OE_BAR pin is only available on the bare die version.

**Table 29: 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 30.

**Table 30: 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.<br>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.

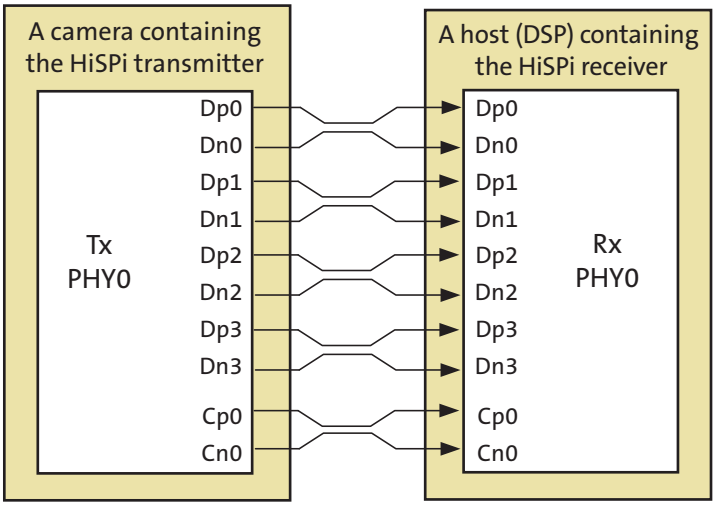
- SLVSC_P
- SLVSC_N
- SLVS0_P
- SLVS0_N
- SLVS1_P
- SLVS1_N
- SLVS2_P
- SLVS2_N
- SLVS3_P
- SLVS3_N

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 21 shows the configuration between the HiSPi transmitter and the receiver.

**Figure 21: HiSPi Transmitter and Receiver Interface Block Diagram**

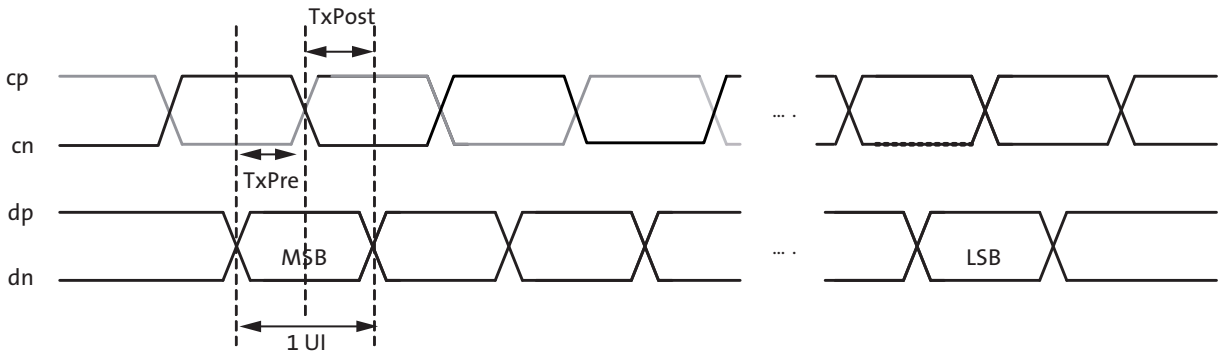


### HiSPi Physical Layer

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-, 14- 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 22 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 22: 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 23: Block Diagram of DLL Timing Adjustment**

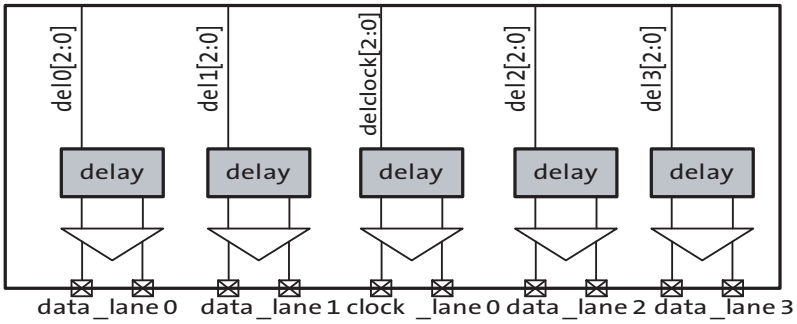


**Figure 24: Delaying the clock_lane with Respect to data_lane**

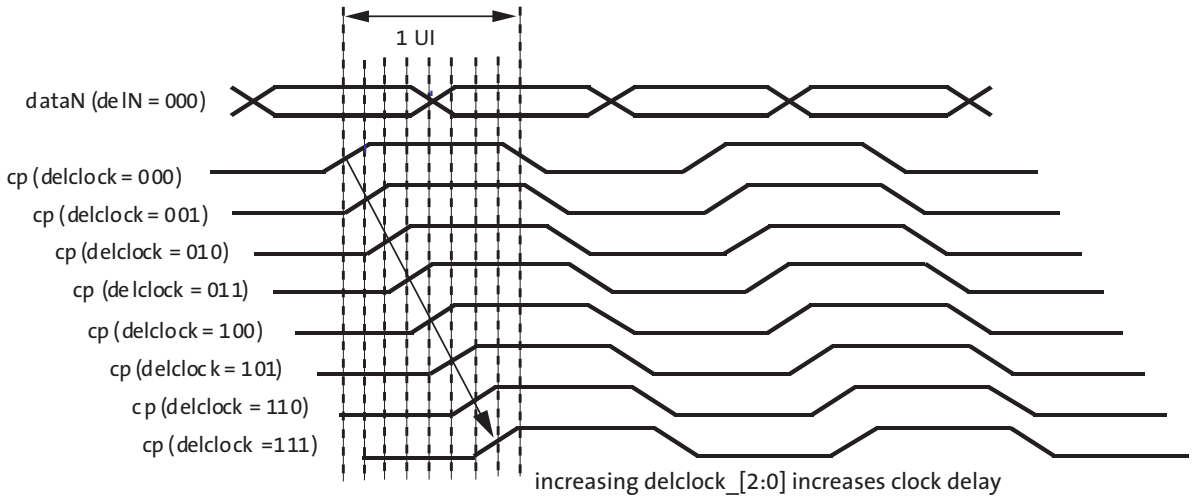
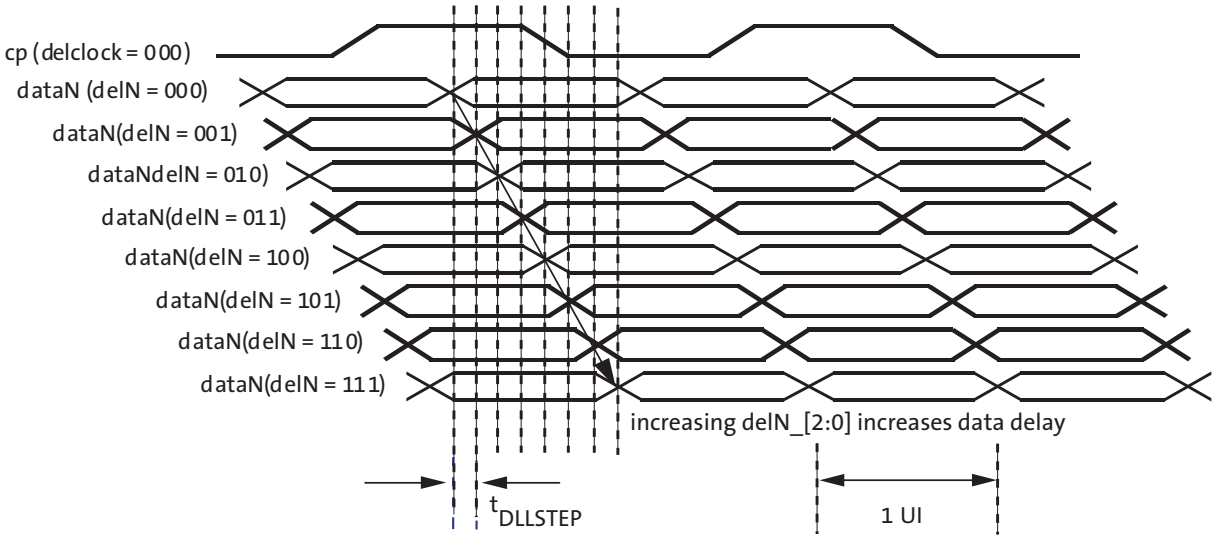



**Figure 25: Delaying data_lane with Respect to the clock_lane**




### HiSPi Streaming Mode Protocol Layer

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

## MIPI Interface

The serial pixel data interface uses the following output-only signal pairs:

- DATA1_P
- DATA1_N
- DATA2_P
- DATA2_N
- DATA3_P
- DATA3_N
- DATA4_P
- DATA4_N
- CLK_P
- CLK_N

The signal pairs use both single-ended and differential signaling, in accordance with the the MIPI Alliance Specification for D-PHY v1.00.00. The serial pixel data interface is enabled by default at power up and after reset.

The DATA0_P, DATA0_N, DATA1_P, DATA1_N, CLK_P and CLK_N pads are set to the Ultra Low Power State (ULPS) if the serial disable bit is asserted (R0x301A-B[12]=1) or when the sensor is in the hardware standby or soft standby system states.

When the serial pixel data interface is used, the LINE_VALID, FRAME_VALID, PIXCLK and dout[11:0] signals (if present) can be left unconnected.

## Serial Configuration

The serial format should be configured using R0x31AC. This register should be programmed to 0x0C0C when using the parallel interface.

The R0x0112-3 register can be programmed to any of the following data format settings that are supported:

- 0x0C0C – Sensor supports RAW12 uncompressed data format
- 0x0C0A – The sensor supports RAW12 compressed format (10-bit words) using 12-10 bit A-LAW Compression. See “Compression” on page 59.
- 0x0A0A – Sensor supports RAW10 uncompressed data format. This mode is supported by discarding all but the upper 10 bits of a pixel value.
- 0x0808 – Sensor supports RAW8 uncompressed data format. This mode is supported by discarding all but the upper 8 bits of a pixel value (MIPI only).

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 formats are supported:

- 0x0201 – Sensor supports single-lane MIPI operation
- 0x0202 – Sensor supports dual-lane MIPI operation
- 0x0204 – Sensor supports quad-lane MIPI operation
- 0x0304 - Sensor supports quad-lane HiSPi operation

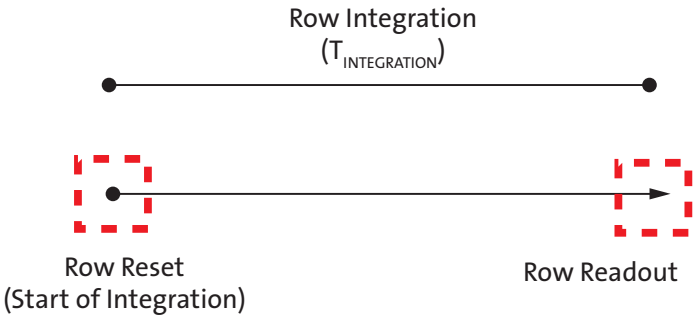
The MIPI timing registers must be configured differently for 10-bit or 12-bit modes. These modes should be configured when the sensor streaming is disabled. See Table 31 on page 41

**Table 31: Recommended MIPI Timing Configuration**

| Register | Configuration | | Description |
|---|---|---|---|
| | 10bit, 490Mbps/lane | 12-bit, 588Mbps/lane | |
| | Clocking: Continuous | | |
| 0x31B0 | 40 | 36 | Frame Preamble |
| 0x31B2 | 14 | 12 | Line Preamble |
| 0x31B4 | 0x2743 | 0x2643 | MIPI Timing 0 |
| 0x31B6 | 0x114E | 0x114E | MIPI Timing 1 |
| 0x31B8 | 0x2049 | 0x2048 | MIPI Timing 2 |
| 0x31BA | 0x0186 | 0x0186 | MIPI Timing 3 |
| 0x31BC | 0x8005 | 0x8005 | MIPI Timing 4 |
| 0x31BE | 0x2003 | 0x2003 | MIPI Config Status |

## Pixel Sensitivity

**Figure 26: 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.

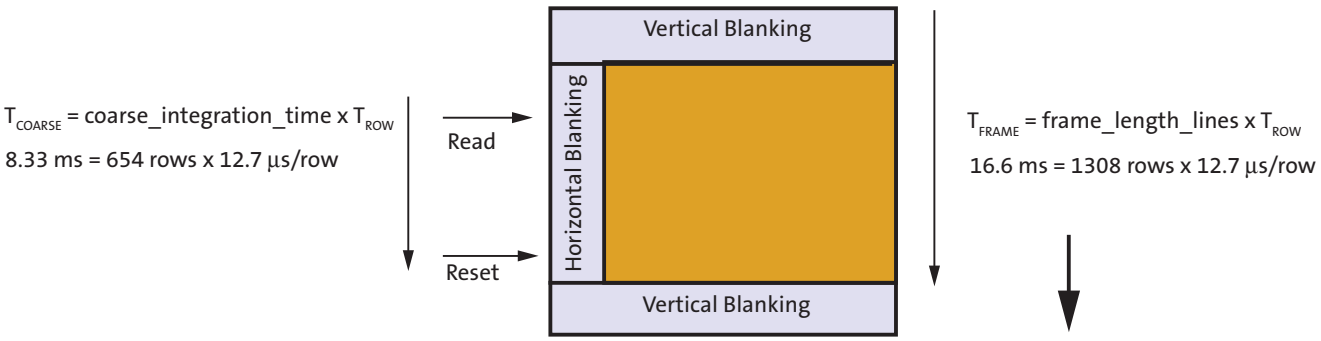
The integration time in an ERS frame is defined as:

$$T_{INTEGRATION} = T_{COARSE} - T_{FINE} \quad \text{(EQ 13)}$$

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).

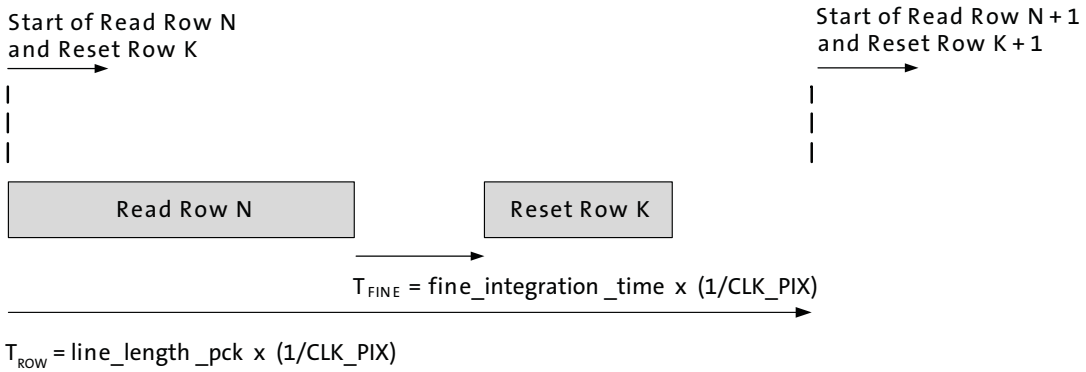
$$T_{COARSE} = T_{ROW} * \text{coarse\_integration\_time} \quad \text{(EQ 14)}$$

**Figure 27: 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 28: Row Read and Row Reset Showing Fine Integration**

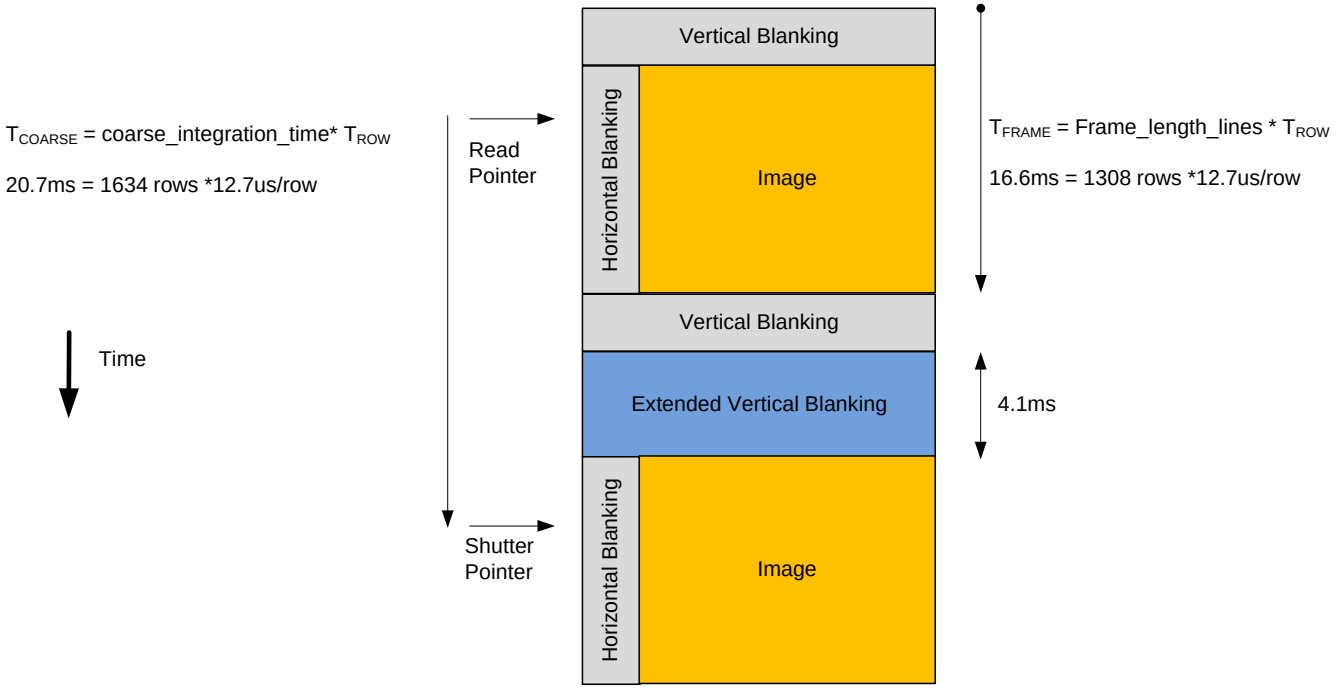



$$T_{FINE} = fine\_integration\_time / clk\_pix \quad \text{(EQ 15)}$$

The maximum allowed value for fine_integration_time is *line_length_pck - 1204*.

ON Semiconductor recommends that the fine_integration_time in the AR0330 be left at zero.

**Figure 29: The Row Integration Time is Greater Than the Frame Readout Time**

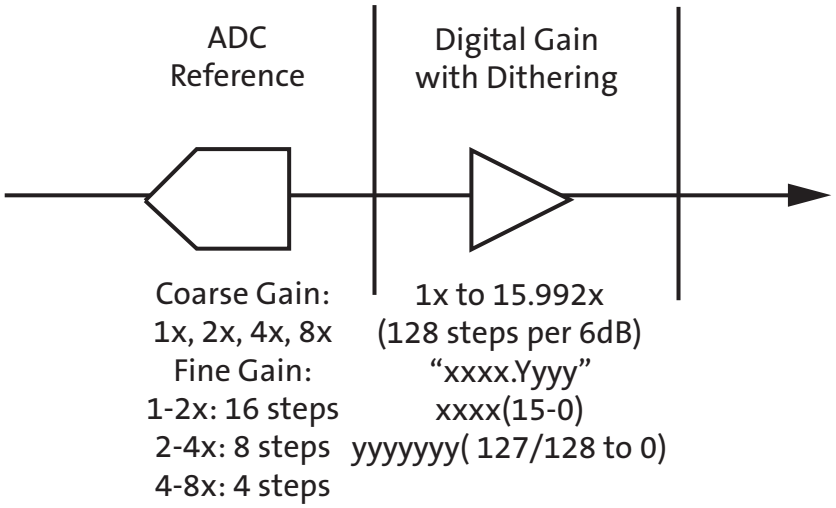



The minimum frame-time is defined by the number of row periods per frame and the row period. The sensor frame-time will increase if the *coarse_integration_time* is set to a value equal to or greater than the f*rame_length_lines*. The maximum integration time can be limited to the frame time by setting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

## Gain Stages

The analog gain stages of the AR0330 sensor are shown in Figure 30. 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 30: Gain Stages in AR0330 Sensor**




The level of analog gain applied is controlled by the coarse_gain and fine_gain registers. The analog readout can be configured differently for each gain level. The recommended gain tables are listed in Table 32. It is recommended that these registers are configured before streaming images.

**Table 32: Recommended Sensor Analog Gain Tables**

| COARSE_GAIN | | FINE_GAIN | | Total Gain | | COARSE_GAIN | | FINE_GAIN | | Total Gain | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| R0x3060[5:4] | Gain (x) | R0x3060 [3:0] | Gain (x) | (x) | (dB) | R0x3060[5:4] | Gain (x) | R0x3060 [3:0] | Gain (x) | (x) | (dB) |
| 0 | 1 | 0 | 1.00 | 1.00 | 0.00 | 0 | 1x | 15 | 1.88 | 1.88 | 5.49 |
| 0 | 1 | 1 | 1.03 | 1.03 | 0.26 | 1 | 2x | 0 | 1.00 | 2.00 | 6.00 |
| 0 | 1 | 2 | 1.07 | 1.07 | 0.56 | 1 | 2x | 2 | 1.07 | 2.13 | 6.58 |
| 0 | 1 | 3 | 1.10 | 1.10 | 0.86 | 1 | 2x | 4 | 1.14 | 2.29 | 7.18 |
| 0 | 1 | 4 | 1.14 | 1.14 | 1.16 | 1 | 2x | 6 | 1.23 | 2.46 | 7.82 |
| 0 | 1 | 5 | 1.19 | 1.19 | 1.46 | 1 | 2x | 8 | 1.33 | 2.67 | 8.52 |
| 0 | 1 | 6 | 1.23 | 1.23 | 1.80 | 1 | 2x | 10 | 1.45 | 2.91 | 9.28 |
| 0 | 1 | 7 | 1.28 | 1.28 | 2.14 | 1 | 2x | 12 | 1.60 | 3.20 | 10.10 |
| 0 | 1 | 8 | 1.33 | 1.33 | 2.50 | 1 | 2x | 14 | 1.78 | 3.56 | 11.02 |
| 0 | 1 | 9 | 1.39 | 1.39 | 2.87 | 2 | 4x | 0 | 1.00 | 4.00 | 12.00 |
| 0 | 1 | 10 | 1.45 | 1.45 | 3.25 | 2 | 4x | 4 | 1.14 | 4.57 | 13.20 |
| 0 | 1 | 11 | 1.52 | 1.52 | 3.66 | 2 | 4x | 8 | 1.33 | 5.33 | 14.54 |
| 0 | 1 | 12 | 1.60 | 1.60 | 4.08 | 2 | 4x | 12 | 1.60 | 6.40 | 16.12 |
| 0 | 1 | 13 | 1.68 | 1.68 | 4.53 | 3 | 8x | 0 | 1.00 | 8.00 | 18.00 |
| 0 | 1 | 14 | 1.78 | 1.78 | 5.00 | | | | | | |

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 “Real-Time Context Switching” on page 47 for the analog and digital gain registers in both context A and context B modes.

Refer to “Real-Time Context Switching” on page 57 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.

# Sensor Readout

## Image Acquisition Modes

The AR0330 supports two image acquisition modes:

- Electronic rolling shutter (ERS) mode
  This is the normal mode of operation. When the AR0330 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 AR0330 switches cleanly from the old integration time to the new while only generating frames with uniform integration. See “Changes to Integration Time” in the AR0330 Register Reference.
- 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 AR0330 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. The x_addr_start equal to 6 is the minimum setting value. The y_addr_start equal to 6 is the minimum setting value. Please refer to Table 33 and Table 34 for details.

**Table 33: Pixel Column Configuration**

| Column Address | Number | Type | Notes |
|---|---|---|---|
| 0–5 | 6 | Active | Border columns |
| 6–2309 | 2304 | Active | Active columns |
| 2310–2315 | 6 | Active | Border columns |

**Table 34: Pixel Row Configuration**

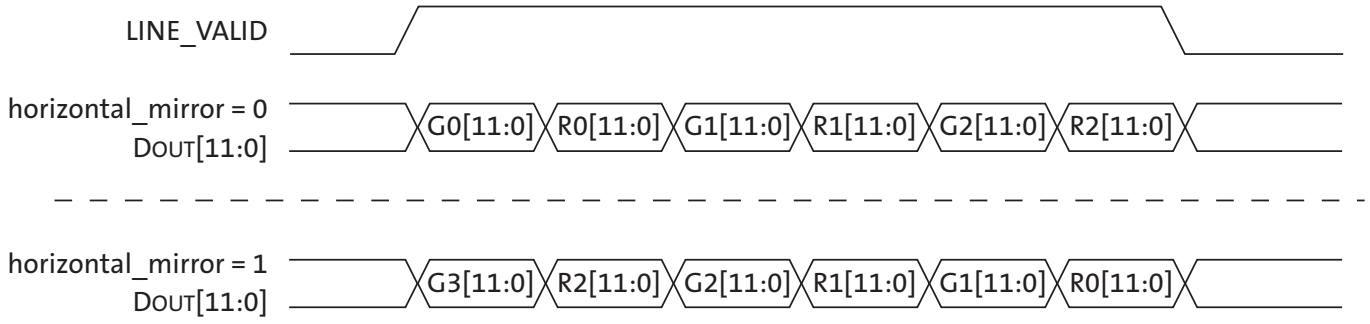
| Row Address | Number | Type | Notes |
|---|---|---|---|
| 2–5 | 4 | Active | Not used in case of “edge effects” |
| 6–1549 | 1544 | Active | Active rows |
| 1550–1555 | 6 | Active | Not used in case of “edge effects” |

## 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 31 on page 47 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 31: Effect of Horizontal Mirror on Readout Order**

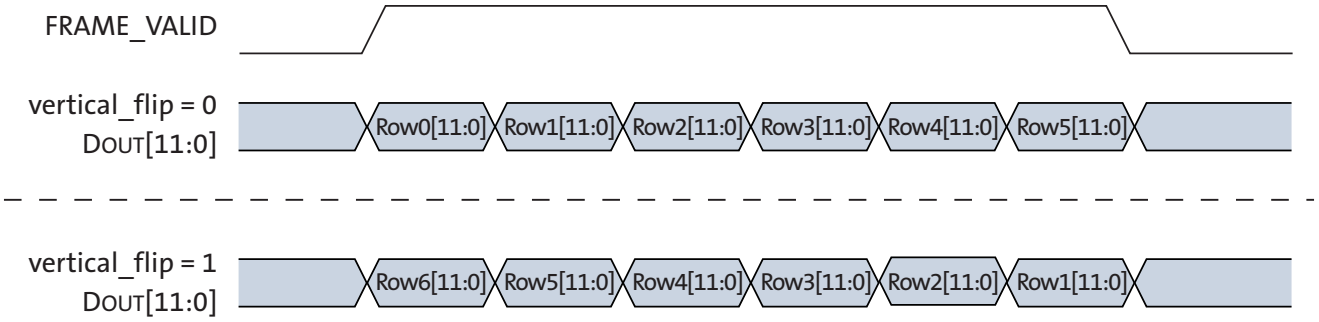



### Vertical Flip

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.

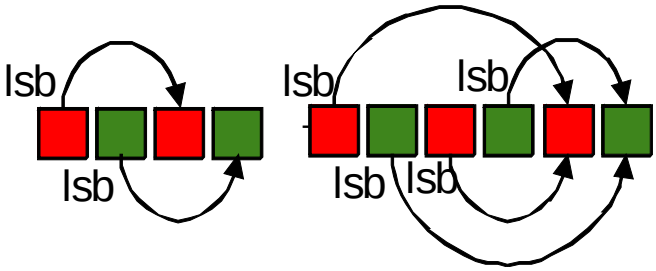
**Figure 32: Effect of Vertical Flip on Readout Order**



## Subsampling

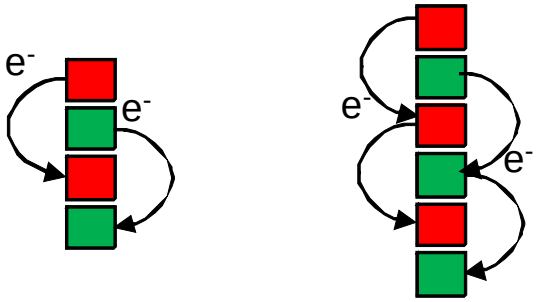
The AR0330 supports subsampling. Subsampling allows the sensor to read out a smaller set of active pixels by either skipping or binning pixels within the readout window. The working modes described in the data sheet that use subsampling are configured to use either 2x2 or 3x3 subsampling.

**Figure 33: Horizontal Binning in the AR0330 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 34: Vertical Row Binning in the AR0330 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. ON Semiconductor recommends not to use 3x binning in AR0330 as it may introduce some image artifacts.

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.

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.

$$\text{x subsampling factor} = \frac{1 + \text{x\_odd\_inc}}{2} \quad \text{(EQ 16)}$$

$$\text{y subsampling factor} = \frac{1 + \text{y\_odd\_inc}}{2} \quad \text{(EQ 17)}$$

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.

**Table 35: Configuration for Horizontal Subsampling**

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

**Table 36: Configuration for Vertical Subsampling**

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

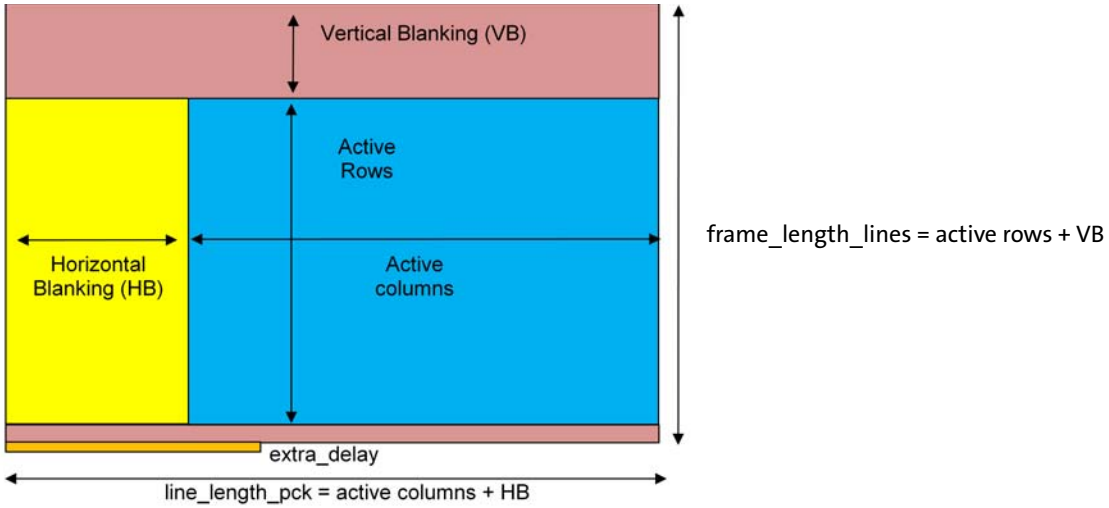
$$fps=1/T_{FRAME} \quad \text{(EQ 18)}$$

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*)

$$T_{FRAME}=1/(CLK\_PIX) \times[frame\_length\_lines \times line\_length\_pck + extra\_delay] \quad \text{(EQ 19)}$$

**Figure 35: 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 18 on page 33, 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:**

$$1204(\text{ADC\_HIGH\_SPEED}) = 0$$
$$or \quad \text{(EQ 20)}$$
$$1116(\text{ADC\_HIGH\_SPEED}) = 1(0)$$

Options to modify this limit, as mentioned in the "Sequencer" section, can be found in the AR0330 Developer Guide.

**Digital Readout Limitation:**

$$\frac{1}{3} \times \left[ \frac{x\_addr\_end - x\_addr\_start}{(x\_odd\_inc + 1) \times 0.5} \right] \quad \textit{(EQ 21)}$$

**Output Interface Limitations:**

$$\frac{1}{2} \times \left[ \frac{x\_addr\_end - x\_addr\_start}{(x\_odd\_inc + 1) \times 0.5} \right] + 96 \quad \textit{(EQ 22)}$$

## 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.

$$Minimum\ frame\_length\_lines = \frac{y\_addr\_end - y\_addr\_start}{(y\_odd\_inc + 1)/2} + minimum\_vertical\_blanking \quad \text{(EQ 23)}$$

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 MIPI CSI-2 Specification V1.00.

**Table 37: Minimum Vertical Blanking Configuration**

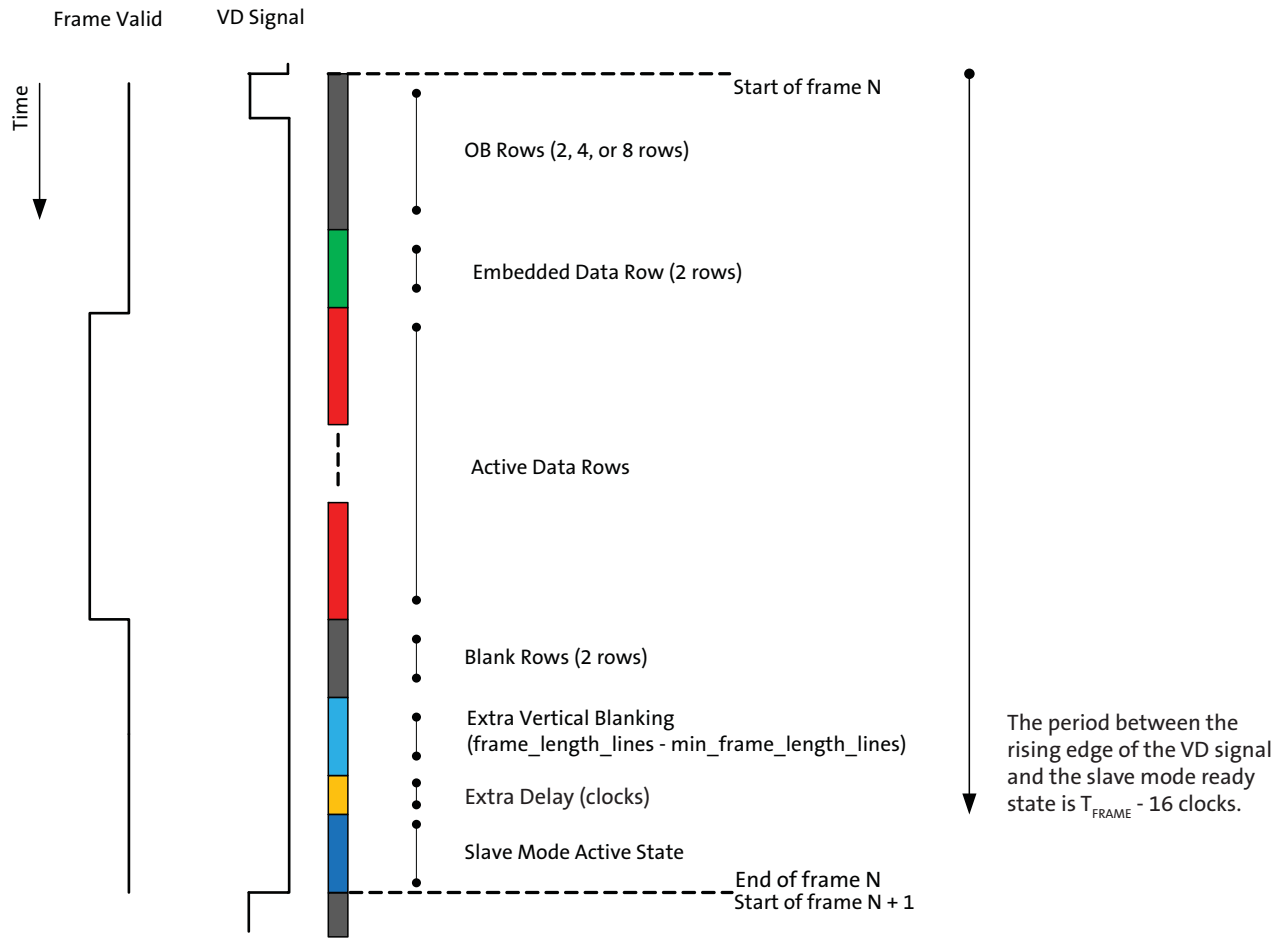
| R0x3180[0x00F0] | OB Rows | minimum_vertical_blanking |
|---|---|---|
| 0x8 (Default) | 8 OB Rows | 8 OB + 4 = 12 |
| 0x4 | 4 OB Rows | 4 OB + 4 = 8 |
| 0x2 | 2 OB Rows | 2 OB + 4 = 6 |

The locations of the OB rows, embedded rows, and blank rows within the frame readout are identified in Figure 36: "Slave Mode Active State and Vertical Blanking," on page 53.

## Slave Mode

The slave mode feature of the AR0330 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. Both the GPI_EN (R0x301A[8]) and the SLAVE_MODE (R0x30CE[4]) bits must be set to "1" to enable the slave mode.

**Figure 36: Slave Mode Active State and Vertical Blanking**

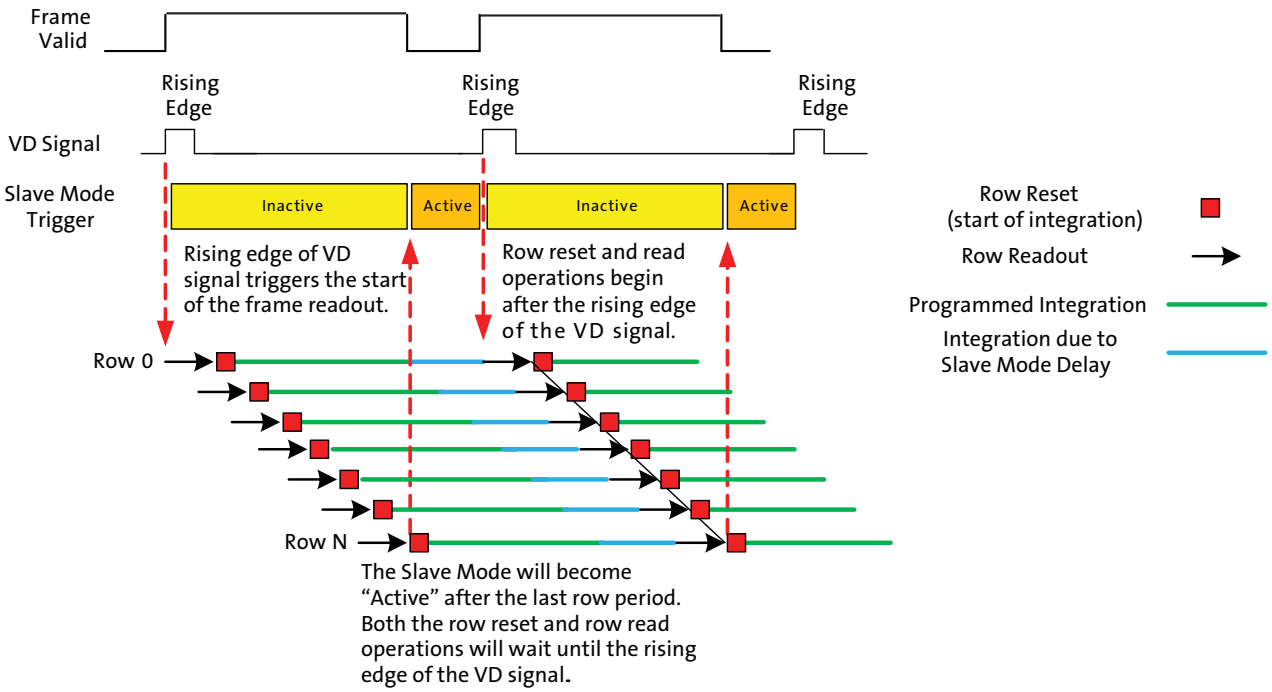



If the slave mode is disabled, the new frame will begin after the extra delay period is finished.

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 37: 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.

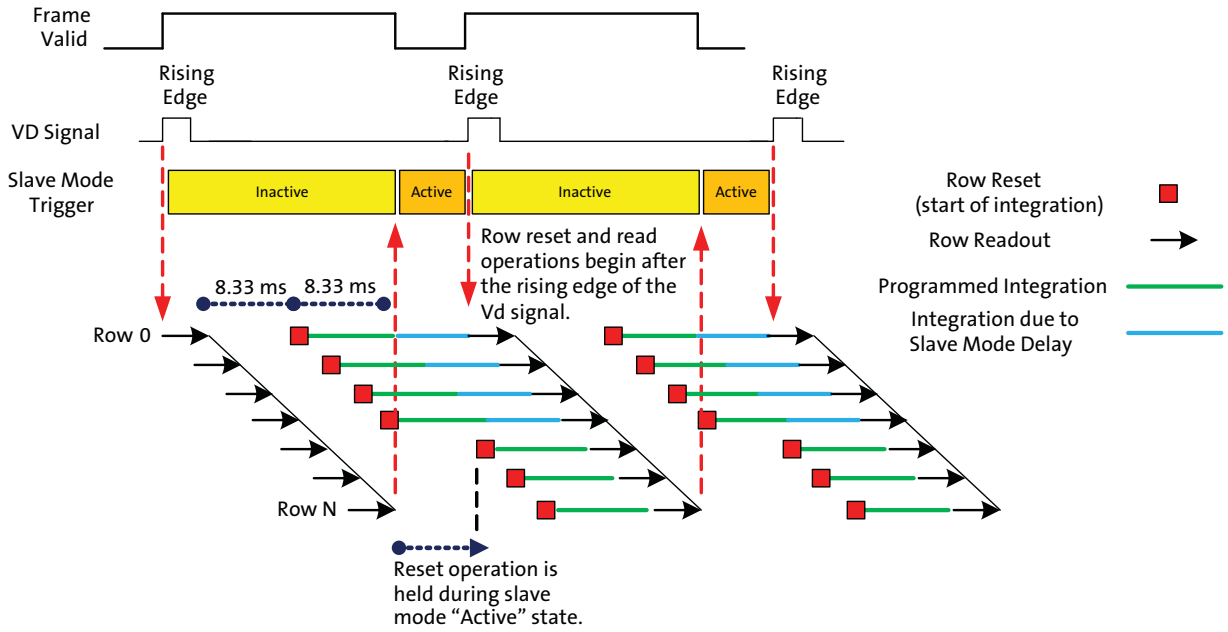



The row shutter and read operations will stop when the slave mode becomes active and is waiting for the VD signal. The following should be considered when configuring the sensor to use the slave mode:

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 38: 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.

When the AR0330 is working in slave mode, the external trigger signal VD must have accurately controlled timing to avoid uneven exposure in the output image. The VD timing control should make the slave mode "wait period" less than 32 pixel clocks.

To avoid uneven exposure, programmed integration time cannot be larger than VD period. To increase integration time more than current VD period, the AR0330 must be configured to work at a lower frame rate and read out image with new VD to match the new timing.

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*). 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 39: Example of the Sensor Output of a 2304 x 1296 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.

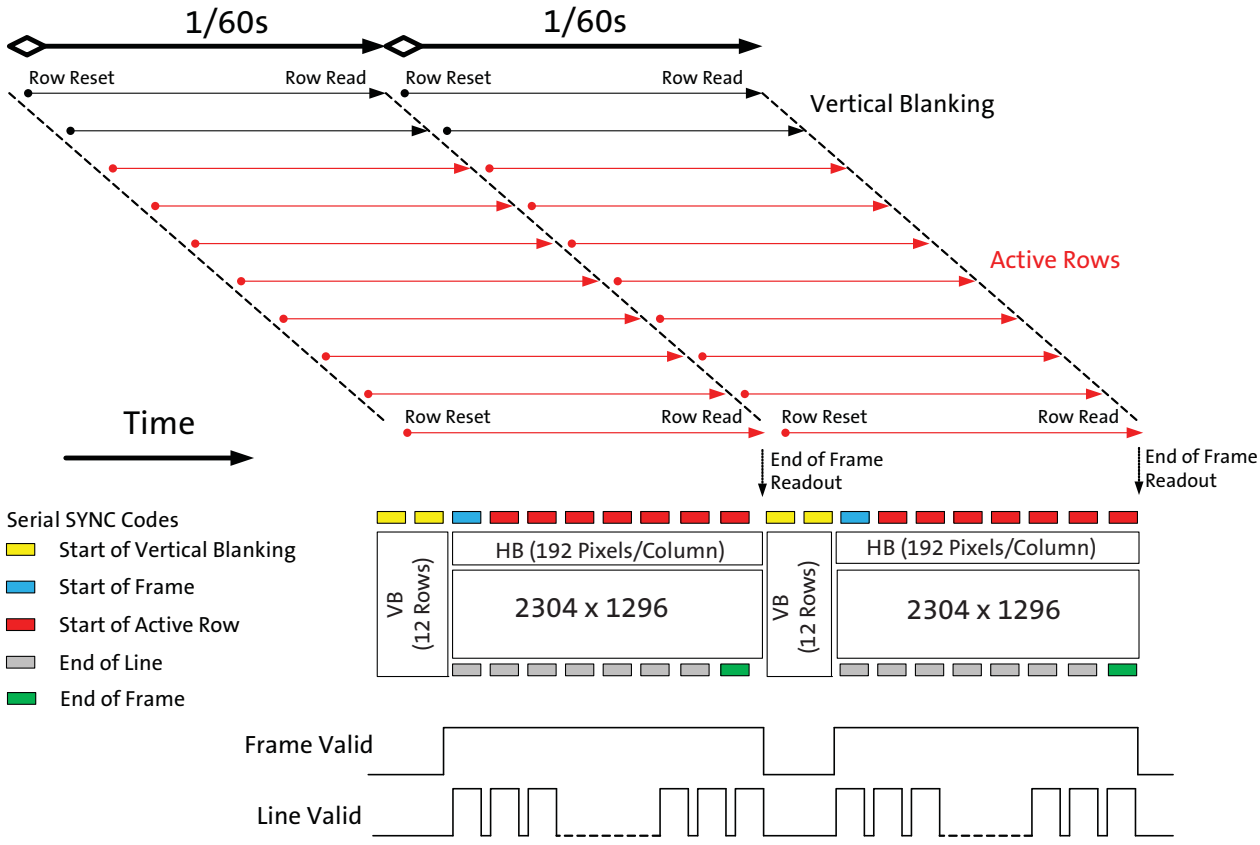



Figure 39 aligns the frame integration and readout operation to the sensor output. It also shows the sensor output using the HiSPi Streaming SP protocol. Different sensor protocols will list different SYNC codes.

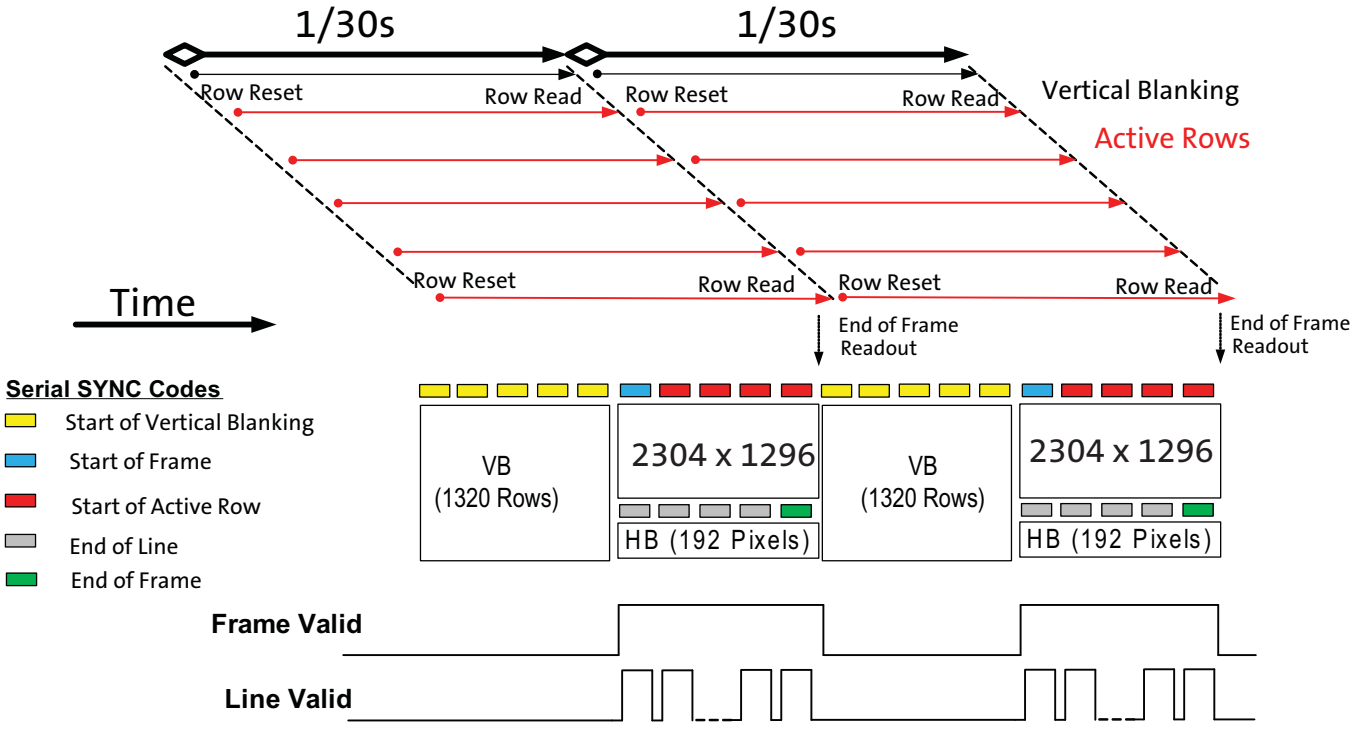
**Table 38: Serial SYNC Codes Included with Each Protocol Included with the AR0330 Sensor**

| 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 |
| MIPI | No SYNC Code | Yes | Yes | Yes | Yes |

Figure 40 illustrates how the sensor active readout time can be minimized while reducing the frame rate. 1308 VB rows were added to the output frame to reduce the 2304 x1296 frame rate from 60 fps to 30 fps without increasing the delay between the readout of the first and last active row.

**Figure 40: Example of the Sensor Output of a 2304 x1296 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

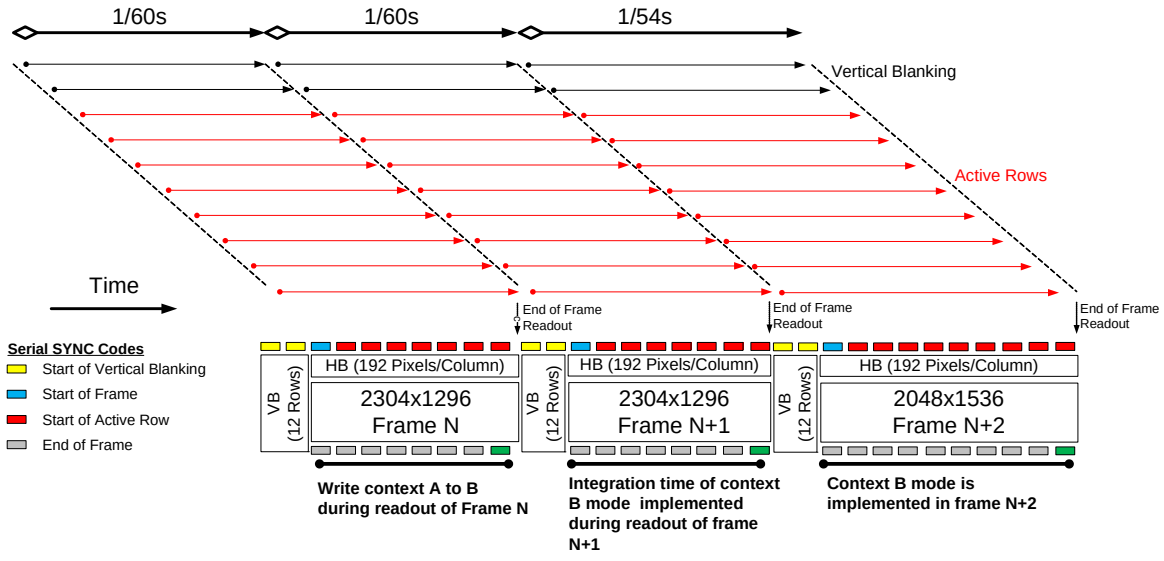
In the AR0330, the user may switch between two full register sets A and B by writing to a context switch change bit in R0x30B0[13]. When the context switch is configured to context A the sensor will reference the “Context A Registers”. If the context switch is changed from A to B during the readout of 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 39: 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 |
| ADC_HIGH_SPEED | 0x30BA[6] | ADC_HIGH_SPEED_CB | 0x30BA[7] |
| 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 |

Note: ON Semiconductor recommends leaving fine_integration_time at 0.

**Figure 41: Example of Changing the Sensor from Context A to Context B**



## Compression

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 Figure 1: “Block Diagram,” on page 6.

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

**Table 40: 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 |

## Test Patterns

The AR0330 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 41. 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.

**Table 41: Test Pattern Modes**

| Test_Pattern_Mode | Test Pattern Output |
|---|---|
| 0 | No test pattern (normal operation) |
| 1 | Solid Color |
| 2 | 100% Vertical Color Bars |
| 3 | Fade-to-Gray Vertical Color Bars |
| 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 AR0330. This interface is designed to be compatible with the electrical characteristics and transfer protocols of the $I^2C$ 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 AR0330 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 both 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 AR0330 sensor are 0x20 (write address) and 0x21 (read address). Alternate slave addresses of 0x30 (WRITE address) and 0x31 (READ address) can be selected by asserting the SADDR signal (tie HIGH).

Alternate slave addresses 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.

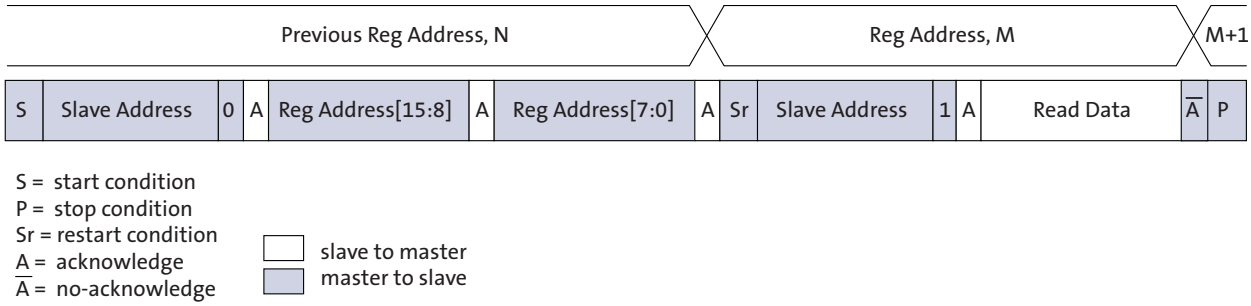
If the request was a WRITE, the master then transfers the 16-bit register address to which the WRITE should take place. This transfer takes place as two 8-bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master then transfers the 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, eight 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 42) 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 42 shows how the internal register address maintained by the AR0330 is loaded and incremented as the sequence proceeds.

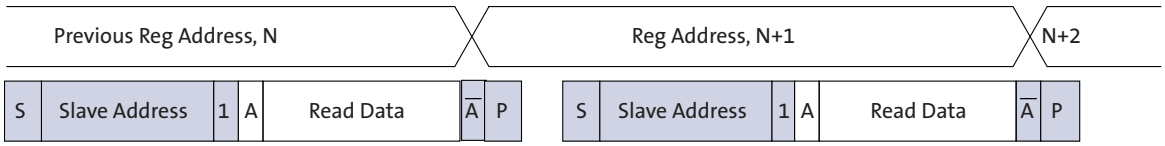
**Figure 42: Single READ From Random Location**




## Single READ From Current Location

This sequence (Figure 43) performs a read using the current value of the AR0330 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.

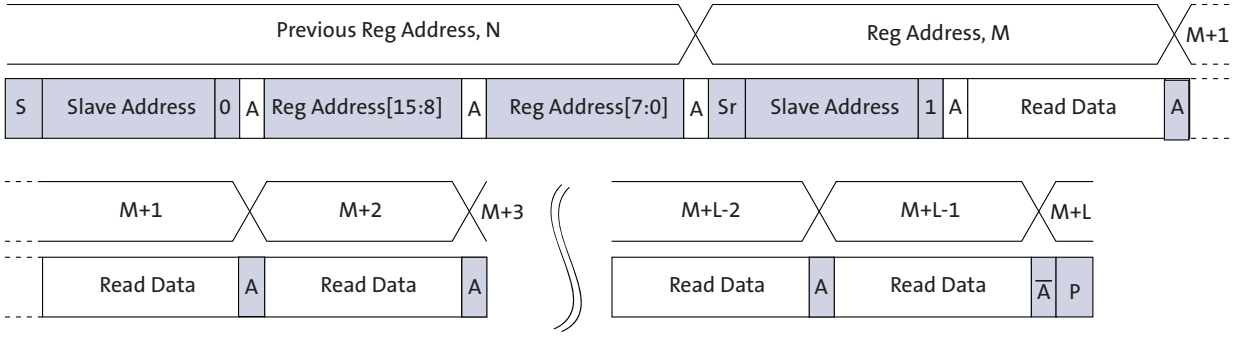
**Figure 43: Single READ From Current Location**



## Sequential READ, Start From Random Location

This sequence (Figure 44) starts in the same way as the single READ from 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 READs until "L" bytes have been read.

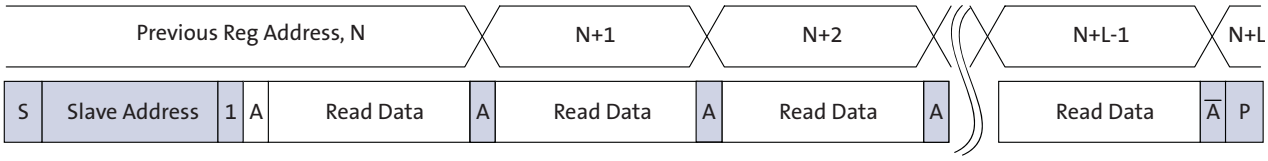
**Figure 44: Sequential READ, Start From Random Location**




## Sequential READ, Start From Current Location

This sequence (Figure 45) starts in the same way as the single READ from current location (Figure 43 on page 62). 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.

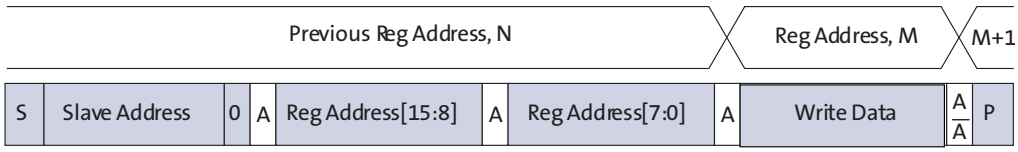
**Figure 45: Sequential READ, Start From Current Location**




## Single WRITE to Random Location

This sequence (Figure 46) 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.

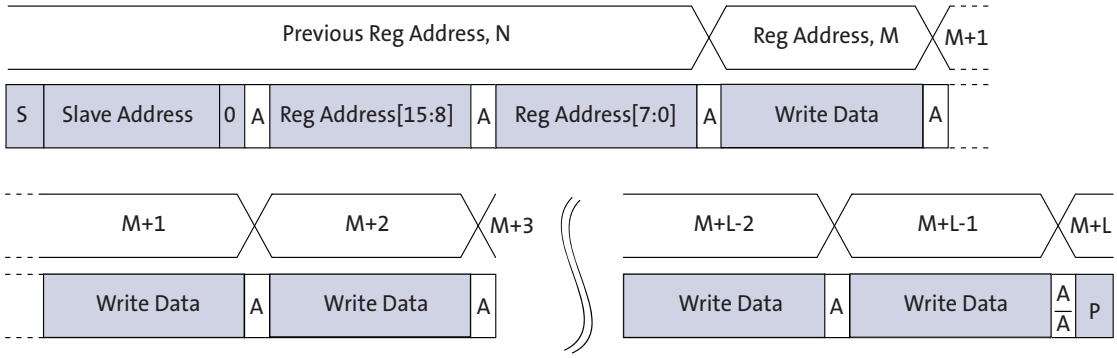
**Figure 46: Single WRITE to Random Location**



## Sequential WRITE, Start at Random Location

This sequence (Figure 47) starts in the same way as the single WRITE to random location (Figure 46 on page 63). Instead of generating a stop condition after the first byte of data has been transferred, the master continues to perform byte WRITEs until 'L' bytes have been written. The WRITE is terminated by the master generating a stop condition.

**Figure 47: Sequential WRITE, Start at Random Location**



## Spectral Characteristics

**Figure 48: Bare Die Quantum Efficiency**

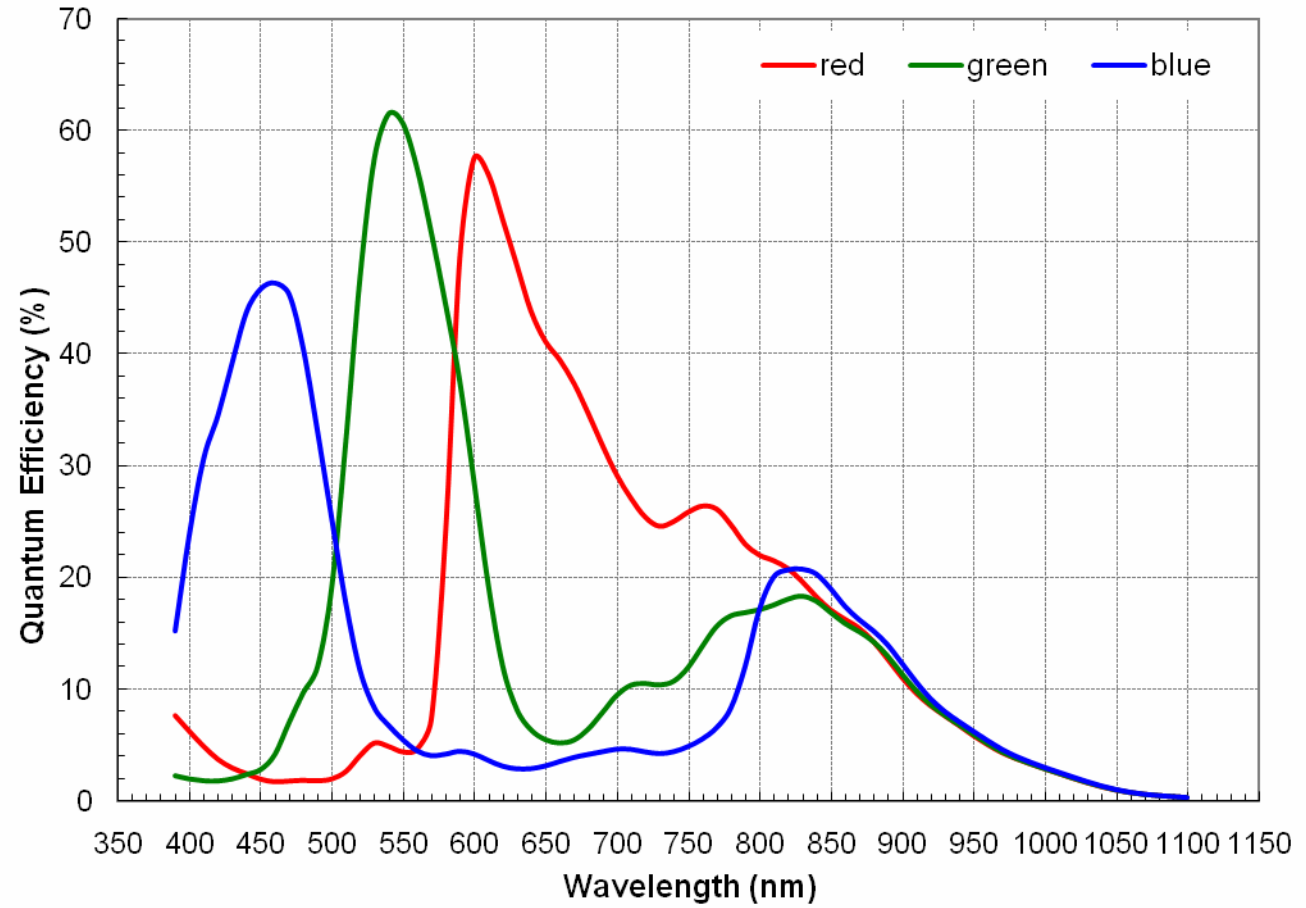
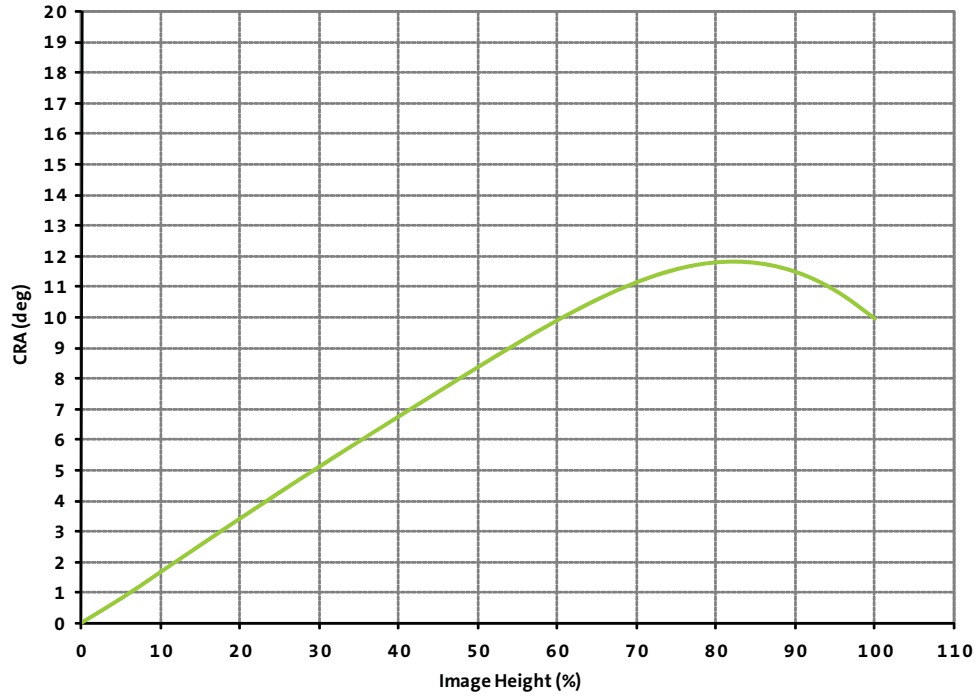


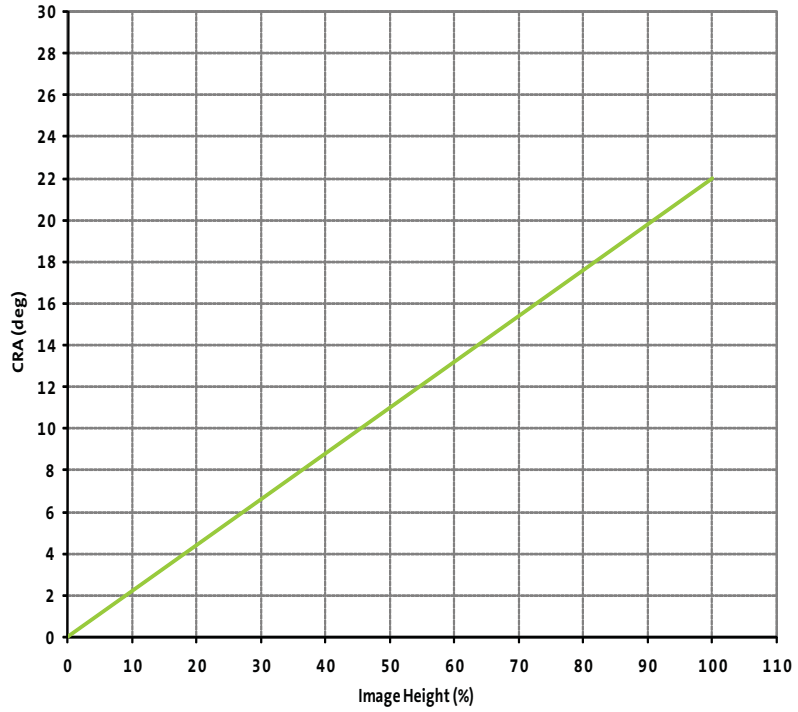
**Table 42: Chief Ray Angle (CRA) 12 °**




| Image Height | | CRA |
|---|---|---|
| % | mm | deg. |
| 0 | 0 | 0 |
| 5 | 0.152 | .80 |
| 10 | 0.305 | 1.66 |
| 15 | 0.457 | 2.54 |
| 20 | 0.609 | 3.42 |
| 25 | 0.761 | 4.28 |
| 30 | 0.914 | 5.11 |
| 35 | 1.066 | 5.94 |
| 40 | 1.218 | 6.75 |
| 45 | 1.371 | 7.57 |
| 50 | 1.523 | 8.37 |
| 55 | 1.675 | 9.16 |
| 60 | 1.828 | 9.90 |
| 65 | 1.980 | 10.58 |
| 70 | 2.132 | 11.15 |
| 75 | 2.284 | 11.57 |
| 80 | 2.437 | 11.80 |
| 85 | 2.589 | 11.78 |
| 90 | 2.741 | 11.48 |
| 95 | 2.894 | 10.88 |
| 100 | 3.046 | 9.96 |

Note: The CRA listed in the advanced data sheet described the 2048x1536 field of view (2.908mm image height). This information was sufficient for configuring the sensor to read both the 4:3 (2048x1536) and 16:9 (2304x1296) aspect ratios. The CRA information listed in the data sheet has now been updated to represent the entire pixel array (2304x1536).

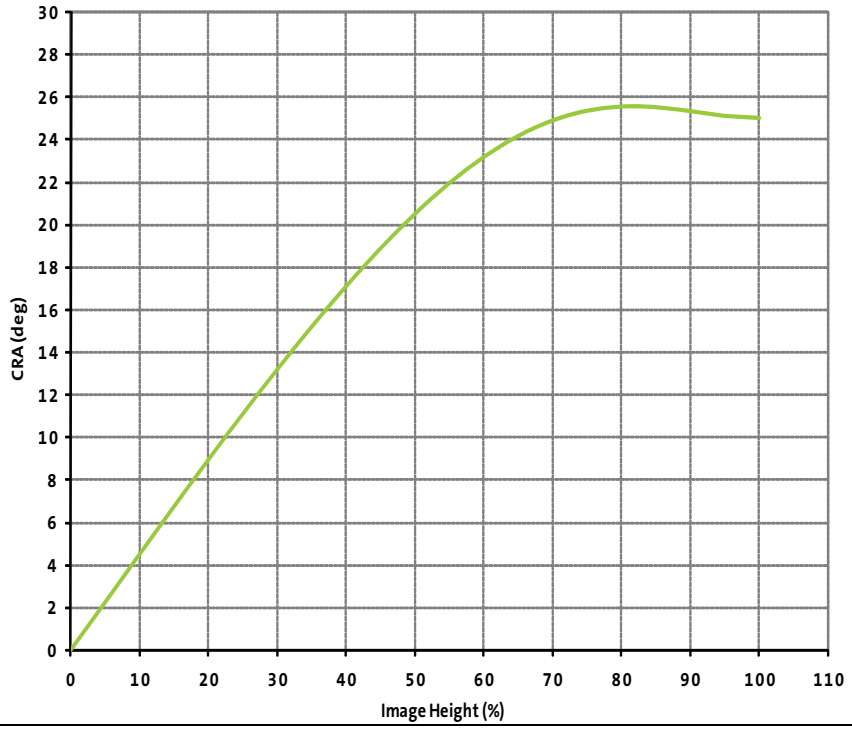
**Table 43: Chief Ray Angle (CRA) 21 °**




| Image Height | | CRA |
|---|---|---|
| % | mm | deg. |
| 0 | 0 | 0 |
| 5 | 0.152 | 1.10 |
| 10 | 0.305 | 2.20 |
| 15 | 0.457 | 3.30 |
| 20 | 0.609 | 4.40 |
| 25 | 0.761 | 5.50 |
| 30 | 0.914 | 6.60 |
| 35 | 1.066 | 7.70 |
| 40 | 1.281 | 8.80 |
| 45 | 1.371 | 9.90 |
| 50 | 1.523 | 11.00 |
| 55 | 1.675 | 12.10 |
| 60 | 1.828 | 13.20 |
| 65 | 1.980 | 14.30 |
| 70 | 2.132 | 15.40 |
| 75 | 2.284 | 16.50 |
| 80 | 2.437 | 17.60 |
| 85 | 2.589 | 18.70 |
| 90 | 2.741 | 19.80 |
| 95 | 2.894 | 20.90 |
| 100 | 3.046 | 22.00 |

Note: The CRA listed in the advanced data sheet described the 2048x1536 field of view (2.908mm image height). This information was sufficient for configuring the sensor to read both the 4:3 (2048x1536) and 16:9 (2304x1296) aspect ratios. The CRA information listed in the data sheet has now been updated to represent the entire pixel array (2304x1536).

**Table 44: Chief Ray Angle (CRA) 25 °**




| Image Height | | CRA |
|---|---|---|
| % | mm | deg. |
| 0 | 0 | 0 |
| 5 | 0.152 | 2.24 |
| 10 | 0.305 | 4.50 |
| 15 | 0.457 | 6.75 |
| 20 | 0.609 | 0.895 |
| 25 | 0.761 | 11.11 |
| 30 | 0.914 | 13.19 |
| 35 | 1.066 | 15.20 |
| 40 | 1.218 | 17.10 |
| 45 | 1.371 | 18.88 |
| 50 | 1.523 | 20.50 |
| 55 | 1.675 | 21.95 |
| 60 | 1.828 | 23.18 |
| 65 | 1.980 | 24.17 |
| 70 | 2.132 | 24.89 |
| 75 | 2.284 | 25.35 |
| 80 | 2.437 | 25.54 |
| 85 | 2.589 | 25.51 |
| 90 | 2.741 | 25.33 |
| 95 | 2.894 | 25.11 |
| 100 | 3.046 | 25.01 |

Note: The CRA listed in the advanced data sheet described the 2048x1536 field of view (2.908mm image height). This information was sufficient for configuring the sensor to read both the 4:3 (2048x1536) and 16:9 (2304x1296) aspect ratios. The CRA information listed in the data sheet has now been updated to represent the entire pixel array (2304x1536).

## Read the Sensor CRA

Follow the steps below to obtain the CRA value of the Image Sensor:

1. Set the register bit field R0x301A[5] = 1.
2. Read the register bit fields R0x31FA[11:9].
3. Determine the CRA value according to Table 45.

**Table 45: CRA Value**

| Binary Value of R0x31FA[11:9] | CRA Value |
|---|---|
| 000 | 0 |
| 001 | 21 |
| 010 | 25 |
| 011 | 12 |

# Packages

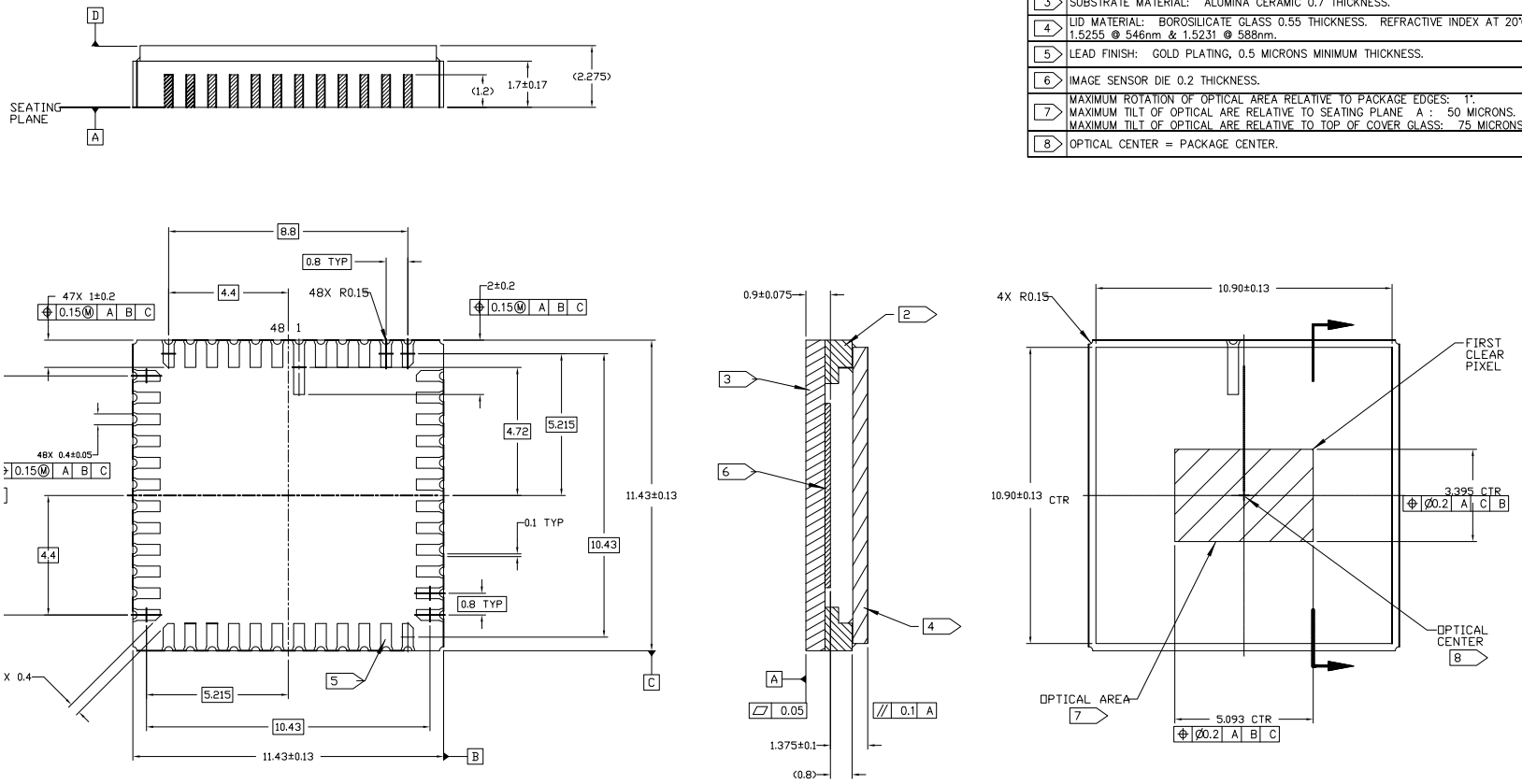
The AR0330 comes in two packages:

- CLCC Package
- CSP HiSPi/MIPI Package

## CLCC Package

### Figure 49: CLCC Package

| NOTES | |
|---|---|
| 1 | DIMENSIONS IN MM. DIMENSIONS IN ( ) ARE FOR REFERENCE ONLY. DO NOT MEASURE PRINTED DRAWING. |
| 2 | WALL MATERIAL: ALUMINA CERAMIC |
| 3 | SUBSTRATE MATERIAL: ALUMINA CERAMIC 0.7 THICKNESS. |
| 4 | LID MATERIAL: BOROSILICATE GLASS 0.55 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 ARE RELATIVE TO SEATING PLANE A : 50 MICRONS. MAXIMUM TILT OF OPTICAL ARE RELATIVE TO TOP OF COVER GLASS: 75 MICRONS. |
| 8 | OPTICAL CENTER = PACKAGE CENTER. |



## CSP Package

### Figure 50: CSP HiSPi Package

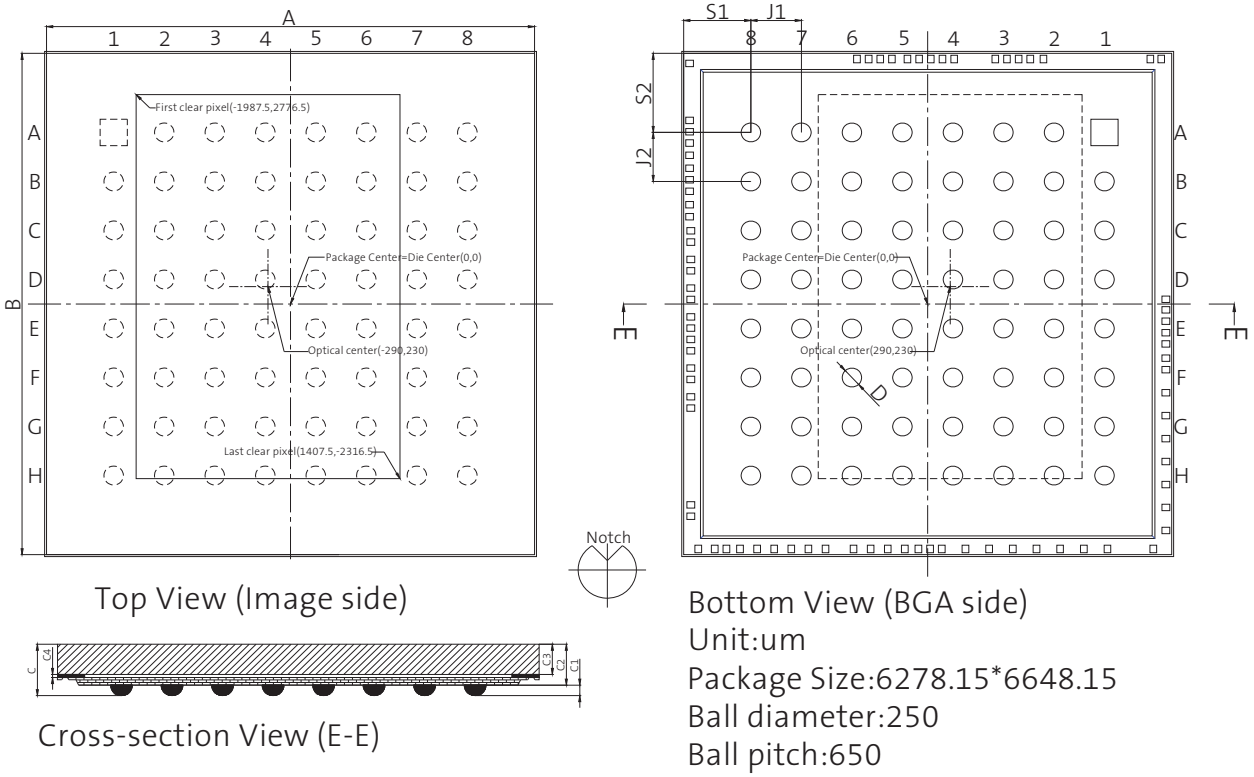


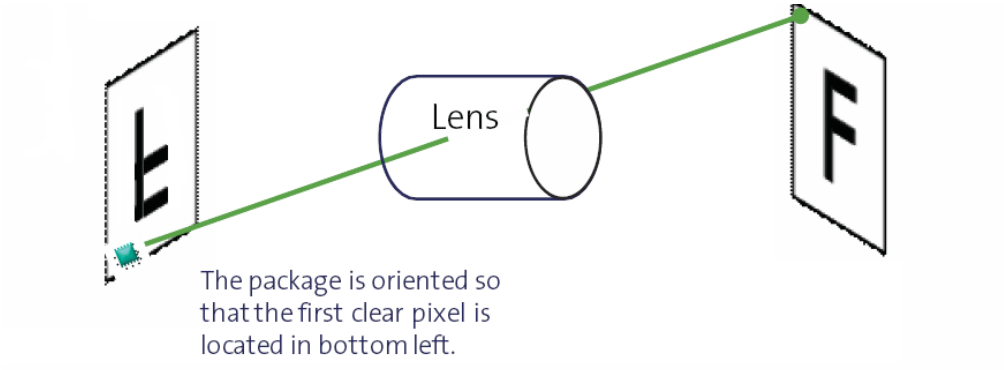
**Table 46: CSP (MIPI/HiSPi) Package Dimensions**

| Parameter | Symbol | Nominal | Min | Max | Nominal | Min | Max |
|---|---|---|---|---|---|---|---|
| | | Millimeters | | | Inches | | |
| Package Body Dimension X | A | 6.278 | 6.253 | 6.303 | 0.247 | 0.246 | 0.248 |
| Package Body Dimension Y | B | 6.648 | 6.623 | 6.673 | 0.262 | 0.261 | 0.263 |
| Package Height | C | 0.700 | 0.645 | 0.745 | 0.028 | 0.025 | 0.029 |
| Cavity height (glass to pixel distance) | C4 | 0.041 | 0.037 | 0.045 | 0.002 | 0.001 | 0.002 |
| Glass Thickness | C3 | 0.400 | 0.390 | 0.410 | 0.016 | 0.015 | 0.016 |
| Package Body Thickness | C2 | 0.570 | 0.535 | 0.605 | 0.022 | 0.021 | 0.024 |
| Ball Height | C1 | 0.130 | 0.100 | 0.160 | 0.005 | 0.004 | 0.006 |
| Ball Diameter | D | 0.250 | 0.220 | 0.280 | 0.010 | 0.009 | 0.011 |
| Total Ball Count | N | 64 | | | | | |
| Ball Count X axis | N1 | 8 | | | | | |
| Ball Count Y axis | N2 | 8 | | | | | |
| UBM | U | 0.280 | 0.270 | 0.290 | 0.011 | 0.011 | 0.011 |
| Pins Pitch X axis | J1 | 0.650 | | | 0.026 | | |
| Pins Pitch Y axis | J2 | 0.650 | | | 0.026 | | |
| BGA ball center to package center offset in X-direction | X | 0.000 | -0.025 | 0.025 | 0.000 | -0.001 | 0.001 |
| BGA ball center to package center offset in Y-direction | Y | 0.000 | -0.025 | 0.025 | 0.000 | -0.001 | 0.001 |
| BGA ball center to chip center offset in X-direction | X1 | 0.000 | -0.014 | 0.014 | 0.000 | -0.001 | 0.001 |
| BGA ball center to chip center offset in Y-direction | Y1 | 0.000 | -0.014 | 0.014 | 0.000 | -0.001 | 0.001 |
| Edge to Ball Center Distance along X | S1 | 0.864 | 0.834 | 0.894 | 0.034 | 0.033 | 0.035 |
| Edge to Ball Center Distance along Y | S2 | 1.049 | 1.019 | 1.079 | 0.041 | 0.040 | 0.042 |

## Package Orientation in Camera Design

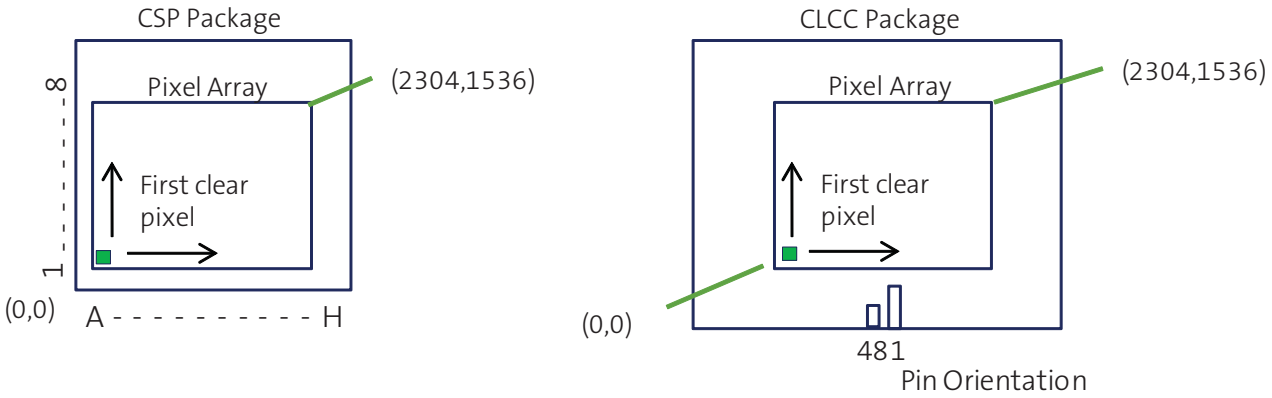
In a camera design, the package should be placed in a PCB so that the first clear pixel is located at the bottom left of the package (look at the package). This orientation will ensure that the image captured using a lens will be oriented correctly.

**Figure 51: Image Orientation With Relation To Camera Lens**




The package pin locations after the sensor has been oriented correctly can be shown below.

**Figure 52: First Clear Pixel and Pin Location**
(Looking Down on Cover Glass)



## Revision History

**Rev. U . . . . . 4/13/15**

- Updated “Ordering Information” on page 2

**Rev. T . . . . . 3/10/15**

- Updated to ON Template and Legal Disclaimer
- Updated Table 5, Pin Descriptions and Table 6, “CSP (HiSPi/MIPI) Package Pinout,” on page 13 names for consistency on page 13
- Added HiSPi voltage information to Figure 6: “Power Up,” on page 15
- Updated Table 9, “DC Electrical Definitions and Characteristics (MIPI Mode),” on page 18
- Added Parallel output information and MIPI information to Table 11, “DC Electrical Definitions and Characteristics (Parallel Mode),” on page 19
- Updated Table 12, “Standby Power,” on page 19
- Updated Two Wire Serial Interface description for consistency - no change to the part specification on page 20
- Updated HiSPi power names for consistency on pages 24 and 25
- Added Table 12, “Standby Power,” on page 19

**Rev. R . . . . . 9/25/13**

- Updated Table 3, “Available Aspect Ratios in the AR0330 Sensor,” on page 7
- Updated Table 5, “Pin Descriptions,” on page 12
- Updated “Power-Up Sequence” on page 15
- Updated “Dual Readout Paths” on page 33
- Updated “Output Enable Control” on page 36
- Updated Figure 30: “Gain Stages in AR0330 Sensor,” on page 44
- Updated Table 32, “Recommended Sensor Analog Gain Tables,” on page 44
- Deleted Table 34, “Available Skip and Bin Modes in the AR0330 Sensor”
- Updated Equation 23 on page 52
- Updated Table 37, “Minimum Vertical Blanking Configuration,” on page 52
- Updated “Frame Readout” on page 56

**Rev. Q . . . . . 3/8/13**

- Updated master clock range in:
  - Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8
  - Figure 3: “Typical Configuration: Serial MIPI,” on page 9
  - Figure 4: “Typical Configuration: Parallel Pixel Data Interface,” on page 10
  - Table 5, “Pin Descriptions,” on page 12
- Updated note for Table 6, “CSP (HiSPi/MIPI) Package Pinout,” on page 13
- Updated Table 9, “DC Electrical Definitions and Characteristics (MIPI Mode),” on page 18
- Updated Table 10, “DC Electrical Definitions and Characteristics (HiSPi Mode),” on page 18
- Updated Table 16, “I/O Timing,” on page 22
- Updated Figure 19: “PLL for the Parallel Interface,” on page 33
- Updated Figure 20: “PLL for the Serial Interface,” on page 34
- Updated “Slave Address/Data Direction Byte” on page 61

**Rev. P . . . . . 10/17/12**

- Updated “Features” on page 1
- Table 1, “Available Part Numbers,” on page 1
- Updated Figure 1: “Block Diagram,” on page 6
- Updated Figure 51: “Image Orientation With Relation To Camera Lens,” on page 72

**Rev. N . . . . . 5/29/12**

- Removed Parallel/MIPI information:
  - deleted Table 7, “CSP (Parallel/MIPI) Package Pinout,” on page 14
  - deleted

**Rev. M. . . . . 2/17/12**

- Updated Table 6, “CSP (HiSPi/MIPI) Package Pinout,” on page 13
- Updated Table 7, “CSP (Parallel/MIPI) Package Pinout,” on page 14
- Updated Table 9, “DC Electrical Definitions and Characteristics (MIPI Mode),” on page 18
- Updated trademarks

**Rev. L . . . . . 12/22/11**

- Updated title of Figure 3: “Typical Configuration: Serial MIPI,” on page 9
- Changed title of Table 5, “Pin Descriptions” to “CLCC Package Pinout”
- Replaced Table 6, “CSP Package Pin Descriptions” with Table 6, CSP (HiSPi/MIPI) Package Pinout and Table 7, “CSP (Parallel/MIPI) Package Pinout,” on page 14
- Updated “Packages” on page 69
- Replaced Figure 52, CSP Package with Figure 50: “CSP HiSPi Package,” on page 70 and Figure 53: “CSP Parallel/MIPI Package Outline Drawing,” on page 74
- Replaced Table 44, “CSP Package Dimensions” with Table 46, CSP (MIPI/HiSPi) Package Dimensions and Table 46, “CSP (MIPI/HiSPi) Package Dimensions,” on page 71

**Rev. K . . . . . 10/26/11**

- Updated to Production
- Updated Table 1, “Available Part Numbers,” on page 1
- Updated Table 1, “Key Parameters,” on page 1
- Updated Figure 1: “Block Diagram,” on page 6
- Updated Table 4, “Available Working Modes in the AR0330 Sensor,” on page 7
- Updated notes for Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8
- Updated notes for Figure 3: “Typical Configuration: Serial MIPI,” on page 9
- Updated notes for Figure 4: “Typical Configuration: Parallel Pixel Data Interface,” on page 10
- Updated Table 5, “Pin Descriptions,” on page 12
- Updated “Power-Up Sequence” on page 15
- Updated Figure 6: “Power Up,” on page 15
- Updated Table 7, “Power-Up Sequence,” on page 16
- Updated “Power-Down Sequence” on page 17
- Updated Table 8, “Power-Down Sequence,” on page 17
- Updated Figure 7: “Power Down,” on page 17
- Added Table 9, “DC Electrical Definitions and Characteristics (MIPI Mode),” on page 18

- Updated Table 10, “DC Electrical Definitions and Characteristics (HiSPi Mode),” on page 18
- Updated Table 12, “Two-Wire Serial Interface Electrical Characteristics,” on page 20
- Updated Table 13, “Two-Wire Serial Interface Timing Specifications,” on page 20
- Updated Figure 8: “Two-Wire Serial Bus Timing Parameters,” on page 20
- Updated Table 16, “I/O Timing,” on page 22
- Updated Figure 17: “Relationship Between Readout Clock and Peak Pixel Rate,” on page 32
- Updated Table 27, “PLL Parameters for the Serial Interface,” on page 35
- Updated Table 28, “Example PLL Configurations for the Serial Interface,” on page 35
- Added sentence to first paragraph under Figure 34: “Vertical Row Binning in the AR0330 Sensor,” on page 49
- Updated Figure 39: “Example of the Slave Mode with a Flat-field Illumination,” on page 57

**Rev. J . . . . . 7/5/11**

- Updated Table 10, “DC Electrical Definitions and Characteristics (HiSPi Mode),” on page 18

**Rev. H . . . . . 6/7/11**

- Updated Figure 50: “CSP HiSPi Package,” on page 70
- Added Table 46, “CSP (MIPI/HiSPi) Package Dimensions,” on page 71

**Rev.G . . . . . 5/26/11**

- Updated Table 1, “Available Part Numbers,” on page 1
- Updated Table 1, “Key Parameters,” on page 1
- Updated Notes 8 and 10 in Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8
- Updated Notes 8 and 10 in Figure 3: “Typical Configuration: Serial MIPI,” on page 9
- Updated Notes 7 and 9 in Figure 4: “Typical Configuration: Parallel Pixel Data Interface,” on page 10

**Rev. F, Advance. . . . . 1/5/11**

- Updated Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8
- Updated Table 6, “CSP (HiSPi/MIPI) Package Pinout,” on page 13
- Updated Figure 5: “CLCC Package Pin Descriptions,” on page 14

**Rev. E, Advance . . . . . 12/17/10**

- Changed part number from AC0330 to AR0330
- Applied updated Aptina template
- Updated “Power-Up Sequence” on page 15
- Updated Figure 6: “Power Up,” on page 15
- Updated Table 27, “PLL Parameters for the Serial Interface,” on page 35
- Updated column 1 heading in Table 29, “Output Enable Control,” on page 36
- Updated Table 28, “Recommended Sensor Gain Tables,” on page 44
- Updated Figure 30: “Gain Stages in AR0330 Sensor,” on page 44
- Updated Figure 50: “CSP HiSPi Package,” on page 70

**Rev. D, Advance . . . . . 11/1/10**

- Changed part number from MT9T002 to AC0330

- Updated “Features” on page 1
- Updated Table 1, “Available Part Numbers,” on page 1
- Updated Table 1, “Key Parameters,” on page 1
- Removed Figure 2: Gain Stages
- Updated first paragraph of “General Description” on page 6
- Moved Working Modes section to follow Functional Overview
- Updated Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8
- Updated Figure 3: “Typical Configuration: Serial MIPI,” on page 9
- Updated Figure 4: “Typical Configuration: Parallel Pixel Data Interface,” on page 10
- Updated Table 5, Pin Descriptions; moved it under new section “Pin Descriptions” on page 12
- Added Table 6, “CSP (HiSPi/MIPI) Package Pinout,” on page 13
- Added Figure 5: “CLCC Package Pin Descriptions,” on page 14
- Added “Electrical Characteristics” on page 18
- Added “Sensor Initialization” on page 15
- Added “Sequencer” on page 32
- Added “Sensor PLL” on page 32
- Added “Pixel Output Interfaces” on page 36
- Added “Sensor Readout” on page 46
- Updated “Subsampling” on page 49
- Added “Sensor Frame Rate” on page 51
- Added “Sensor Frame Rate” on page 51
- Updated “Slave Mode” on page 53
- Added “Frame Readout” on page 56
- Added “Two-Wire Serial Register Interface” on page 60
- Added “Packages” on page 69

**Rev. C, Advance . . . . . 5/3/10**

- Updated Table 5, “Pin Descriptions,” on page 12.
- Added Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8.
- Updated Figure 1: “Block Diagram,” on page 6.
- Updated pins and notes for Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8, Figure 3: “Typical Configuration: Serial MIPI,” on page 9 and Figure 4: “Typical Configuration: Parallel Pixel Data Interface,” on page 10.
- Changed input clock range to 6-64 MHz
- Removed high dynamic range from general description
- Removed STANDBY pad from Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8 and Figure 4: “Typical Configuration: Parallel Pixel Data Interface,” on page 10
- Changed HiSPi to SLVS in Table 5, “Pin Descriptions,” on page 12
- Updated slave mode section
- Updated Figure 36: “Slave Mode Active State and Vertical Blanking,” on page 53
- Updated Table 1, “Available Part Numbers,” on page 1

**Rev. B, Advance . . . . . 4/08/10**

- Updated key parameters and general description
- Updated Table 3
- Removed two-wire serial interface

- Added subsampling section
- Updated Figure 1: “Block Diagram,” on page 6 and Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8
- Added Fig. 3

**Rev. A . . . . . 2/10**

- Initial release

LifeElectronics
Живое партнерство

**ООО «ЛайфЭлектроникс»** **“LifeElectronics” LLC**

*ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703*

*Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.*

*С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров*

*Мы предлагаем:*

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

*В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.*

*Конструкторский отдел помогает осуществить:*

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

*Тел: +7 (812) 336 43 04 (многоканальный)*
*Email: org@lifeelectronics.ru*

***www.lifeelectronics.ru***