



1/3-Inch CMOS Digital Image Sensor

AR0330 Data Sheet

For the latest data sheet, from Sunnyvale's Web site: www.sunnyvale.com

Features

- 2.2µm pixel with Aptina™ A-Pix technology
- Full HD support at 60 fps (2304H x 1296V) for maximum video performance
- Superior low-light performance
- 3.4M (3:2) and 3.15M (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.
- Slow-motion video (WVGA 240 fps)
- On-chip phase-locked loop (PLL) oscillator
- Integrated position-based color and lens shading correction
- Simple two-wire serial interface
- Auto black level calibration
- 12-to-10 bit output A-Law compression
- Slave mode for precise frame-rate control

Applications

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

General Description

Aptina's 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 (5.8 mm) Note: Sensor optical format will also work with lenses designed for 1/3.2" format.

Table 1: Key Parameters (continued)

Parameter		Typical Value
Active pixels		2304(H) x 1536(V): (entire array): 5.07mm(H) x 3.38mm(V) 2048(H) x 1536(V) (4:3, still mode) 2304(H) x 1296(V) (16:9, still mode)
Pixel size		2.2µm x 2.2µm
Color filter array		RGB Bayer
Shutter type		ERS and GRR
Input clock range		6 – 64 MHz
Output clock maximum		196 Mp/s (4-lane HiSPi or MIPI) 98 Mp/s (Parallel)
Output Video	4-lane HiSPi (196 Mp/s)	2304x1296 at 60 fps < 450 mW (HiSPi, V _{CM} 0.2V) 2048x1536 at 54 fps < 450 mW (HiSPi, V _{CM} 0.2V)
	4-lane HiSPi (98 Mp/s)	2304x1296 at 30 fps < 300 mW (HiSPi, V _{CM} 0.2V) 2048x1536 at 25 fps < 300 mW (HiSPi, V _{CM} 0.2V)
Responsivity		1.9 V/lux-sec
SNR _{MAX}		39 dB
Dynamic range		72.4 dB
Supply voltage	I/O	1.8 or 2.8V
	Digital	1.8 V
	Analog	2.8 V
	HiSPi	0.4V - 0.8V, 1.7V - 1.9V
Operating temperature (junction) -T _j		-30°C to +70°C
Package options		11.4 mm x 11.4mm CLCC
		6.28 mm x 6.65 mm CSP
		Bare die

Ordering Information

Table 2: Available Part Numbers

Part Number	Description
AR0330CM1C00SHAA0-E	48pin CLCC (HiSPi/MIPI) 0° CRA
AR0330CM1C00SHKA0-E	CSP (HiSPi/MIPI) 0° CRA
AR0330CM1C21SHKA0-E	CSP (HiSPi/MIPI) 21° CRA
AR0330CM1C25SHKA0-E	CSP (HiSPi/MIPI) 25° CRA



Table of Contents

Features	1
Applications	1
General Description	1
Ordering Information	1
General Description	6
Functional Overview	6
Working Modes	7
Pin Descriptions	11
Sensor Initialization	14
Power-Up Sequence	14
Power-Down Sequence	16
Electrical Characteristics	17
Sequencer	23
Sensor PLL	23
VCO	23
Dual Readout Paths	23
Parallel PLL Configuration	24
Serial PLL Configuration	25
Pixel Output Interfaces	27
Parallel Interface	27
Output Enable Control	27
Configuration of the Pixel Data Interface	27
High Speed Serial Pixel Data Interface	28
HiSPi Physical Layer	29
DLL Timing Adjustment	29
HiSPi Streaming Mode Protocol Layer	30
MIPI Interface	31
Serial Configuration	31
Pixel Sensitivity	32
Gain Stages	34
Data Pedestal	35
Sensor Readout	36
Image Acquisition Modes	36
Window Control	36
Readout Modes	36
Horizontal Mirror	36
Vertical Flip	37
Subsampling	38
Sensor Frame Rate	41
Row Period (T_{ROW})	42
Row Periods Per Frame	42
Slave Mode	43
Frame Readout	46
Changing Sensor Modes	47
Register Changes	47
Real-Time Context Switching	47
Compression	49
Test Patterns	50
Solid Color	50
Vertical Color Bars	50
Walking 1s	50



Two-Wire Serial Register Interface51
Protocol51
Start Condition.....	.51
Stop Condition.....	.51
Data Transfer51
Slave Address/Data Direction Byte51
Message Byte52
Acknowledge Bit52
No-Acknowledge Bit52
Typical Sequence52
Single READ From Random Location53
Single READ From Current Location53
Sequential READ, Start From Random Location.....	.54
Sequential READ, Start From Current Location54
Single WRITE to Random Location54
Sequential WRITE, Start at Random Location55
Spectral Characteristics56
Read the Sensor CRA58
Packages58
CLCC Package59
CSP Package.....	.60
Revision History.....	.61



List of Figures

Figure 1:	Block Diagram	6
Figure 2:	Typical Configuration: Serial Four-Lane HiSPi Interface	8
Figure 3:	Typical MIPI Connections	9
Figure 4:	Typical Configuration: Parallel Pixel Data Interface	10
Figure 5:	CLCC Package Pin Descriptions	13
Figure 6:	Power Up	14
Figure 7:	Power Down	16
Figure 8:	Two-Wire Serial Bus Timing Parameters	18
Figure 9:	I/O Timing Diagram	19
Figure 10:	HiSPi Eye Diagram for Both Clock and Data Signals	21
Figure 11:	HiSPi Skew Between Data Signals Within the PHY	22
Figure 12:	Relationship Between Readout Clock and Peak Pixel Rate	23
Figure 13:	Sensor Dual Readout Paths	24
Figure 14:	PLL for the Parallel Interface	24
Figure 15:	PLL for the Serial Interface	25
Figure 16:	HiSPi Transmitter and Receiver Interface Block Diagram	28
Figure 17:	Timing Diagram	29
Figure 18:	Block Diagram of DLL Timing Adjustment	29
Figure 19:	Delaying the clock_lane with Respect to data_lane	30
Figure 20:	Delaying data_lane with Respect to the clock_lane	30
Figure 21:	Integration Control in ERS Readout	32
Figure 22:	Example of 8.33ms Integration in 16.6ms Frame	33
Figure 23:	Row Read and Row Reset Showing Fine Integration	33
Figure 24:	The Row Integration Time is Greater Than the Frame Readout Time	34
Figure 25:	Gain Stages in AR0330 Sensor	34
Figure 26:	Effect of Horizontal Mirror on Readout Order	37
Figure 27:	Effect of Vertical Flip on Readout Order	37
Figure 28:	Horizontal Binning in the AR0330 Sensor	38
Figure 29:	Vertical Row Binning in the AR0330 Sensor	38
Figure 30:	Frame Period Measured in Clocks	41
Figure 31:	Slave Mode Active State and Vertical Blanking	43
Figure 32:	Slave Mode Example with Equal Integration and Frame Readout Periods	44
Figure 33:	Slave Mode Example Where the Integration Period is Half of the Frame Readout Period	45
Figure 34:	Example of the Sensor Output of a 2304 x 1296 Frame at 60 fps	46
Figure 35:	Example of the Sensor Output of a 2304 x 1296 Frame at 30 fps	47
Figure 36:	Example of Changing the Sensor from Context A to Context B	48
Figure 37:	Single READ From Random Location	53
Figure 38:	Single READ From Current Location	53
Figure 39:	Sequential READ, Start From Random Location	54
Figure 40:	Sequential READ, Start From Current Location	54
Figure 41:	Single WRITE to Random Location	54
Figure 42:	Sequential WRITE, Start at Random Location	55
Figure 43:	Bare Die Quantum Efficiency	56
Figure 44:	CLCC Package	59
Figure 45:	CSP Package	60



List of Tables

Table 1:	Key Parameters	1
Table 2:	Available Part Numbers	1
Table 3:	Available Aspect Ratios in the AR0330 Sensor	7
Table 4:	Available Working Modes in the AR0330 Sensor	7
Table 5:	Pin Descriptions	11
Table 6:	CSP Package Pin Descriptions	12
Table 7:	Power-Up Sequence	15
Table 8:	Power-Down Sequence	16
Table 9:	DC Electrical Definitions and Characteristics	17
Table 10:	Absolute Maximum Ratings	17
Table 11:	Two-Wire Serial Register Interface Electrical Characteristics	18
Table 12:	Two-Wire Serial Register Interface Timing Specification	18
Table 13:	I/O Parameters	19
Table 14:	I/O Timing	20
Table 15:	Parallel I/O Rise Slew Rate	20
Table 16:	HiSPi Rise and Fall Times at 660 Mbps	21
Table 17:	HiSPi Rise and Fall Times at 360 MHz	21
Table 18:	Channel, PHY and Intra-PHY Skew	22
Table 19:	Clock DLL Steps	22
Table 20:	Data DLL Steps	22
Table 21:	PLL Parameters for the Parallel Interface	24
Table 22:	Example PLL Configuration for the Parallel Interface	25
Table 23:	PLL Parameters for the Serial Interface	25
Table 24:	Example PLL Configurations for the Serial Interface	26
Table 25:	Output Enable Control	27
Table 26:	Configuration of the Pixel Data Interface	27
Table 27:	Recommended MIPI Timing Configuration	32
Table 28:	Recommended Sensor Gain Tables	35
Table 29:	Recommended Registers to Configure Sensor Gain Table	35
Table 30:	Available Skip and Bin Modes in the AR0330 Sensor	38
Table 31:	Configuration for Horizontal Subsampling	39
Table 32:	Configuration for Vertical Subsampling	40
Table 33:	Minimum Vertical Blanking Configuration	42
Table 34:	Serial SYNC Codes Included with Each Protocol Included with the AR0330 Sensor	46
Table 35:	List of Configurable Registers for Context A and Context B	48
Table 36:	A-Law Compression Table for 12-10 bits	49
Table 37:	Test Pattern Modes	50
Table 38:	Chief Ray Angle (CRA) 21 °	57
Table 39:	Chief Ray Angle (CRA) 25 °	58
Table 40:	CRA Value	58



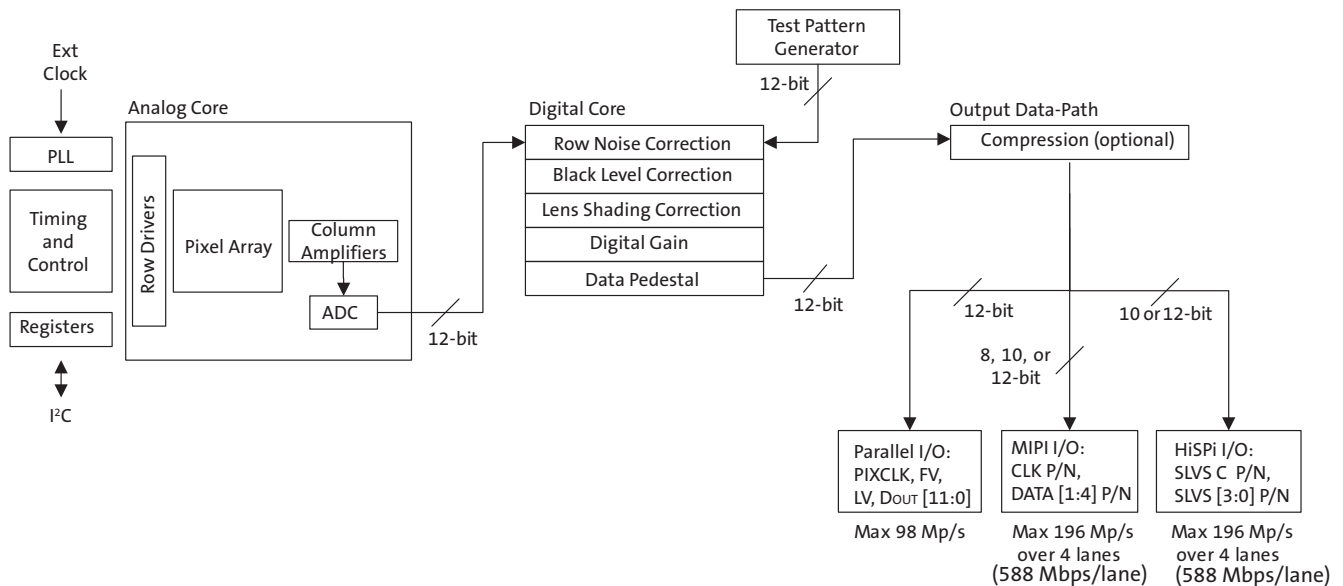
General Description

The Aptina® 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 64 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
4:3	Still Format #1	2256 (H) x 1296 (V)
16:10	Still Format #2	2256 (H) x 1504 (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.

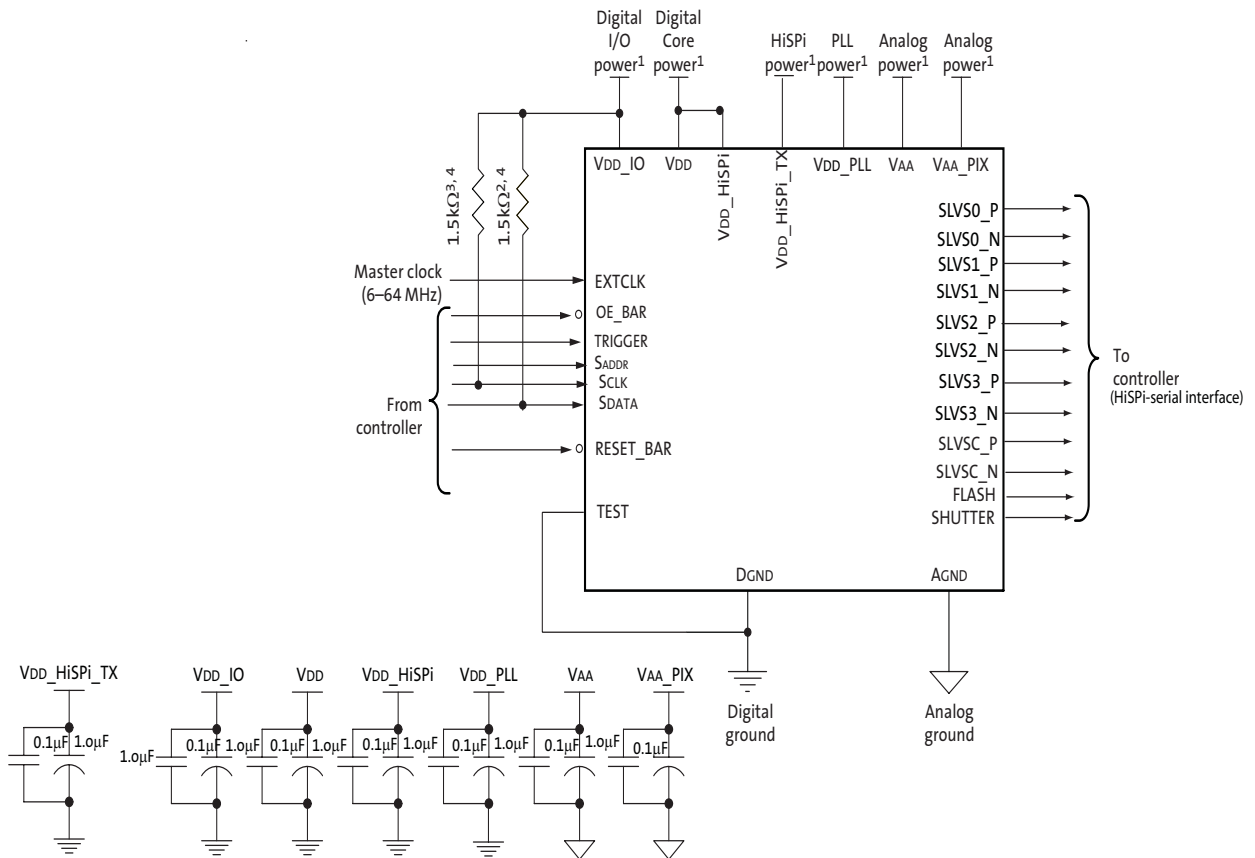
Table 4: Available Working Modes in the AR0330 Sensor

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	30	-	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	120	2x2	100%
VGA Video	16:10	2256 x 1440	752 x 480	60	60	3x3	96%
VGA Video Slow-mo	16:10	2256 x 1440	752 x 480	240	240	3x3	96%



AR0330: 1/3-Inch CMOS Digital Image Sensor Working Modes

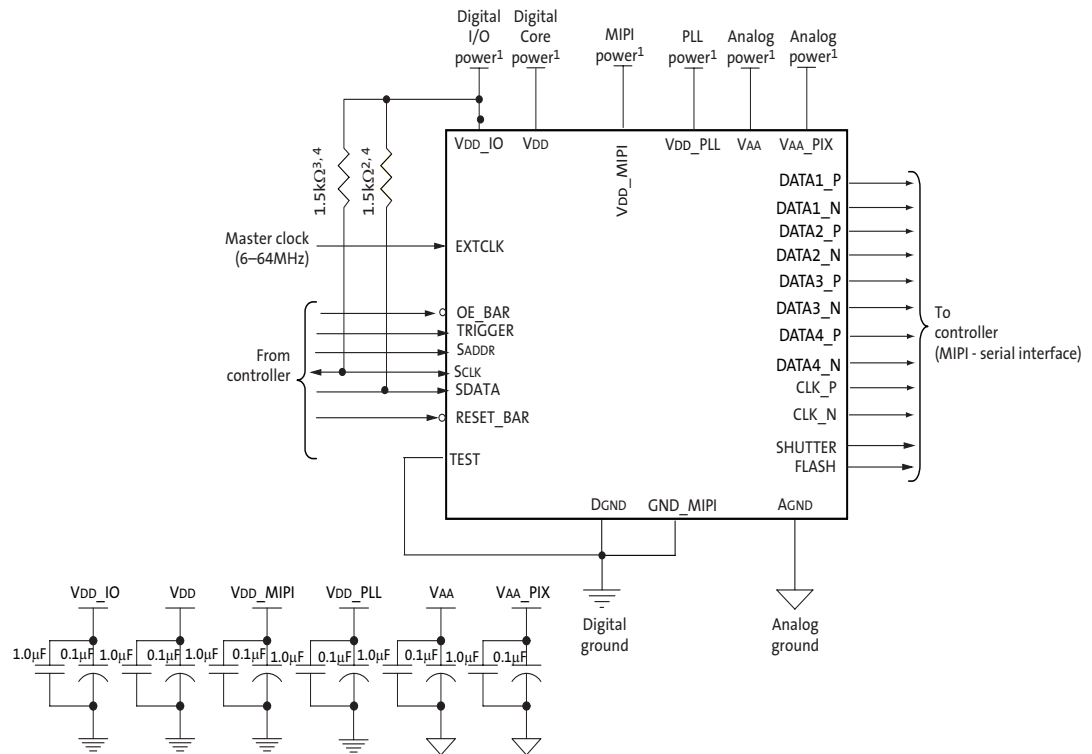
Figure 2: Typical Configuration: Serial Four-Lane HiSPi Interface



- Notes:
1. All power supplies must be adequately decoupled. Aptina 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, Aptina 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. Aptina 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. Aptina 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.



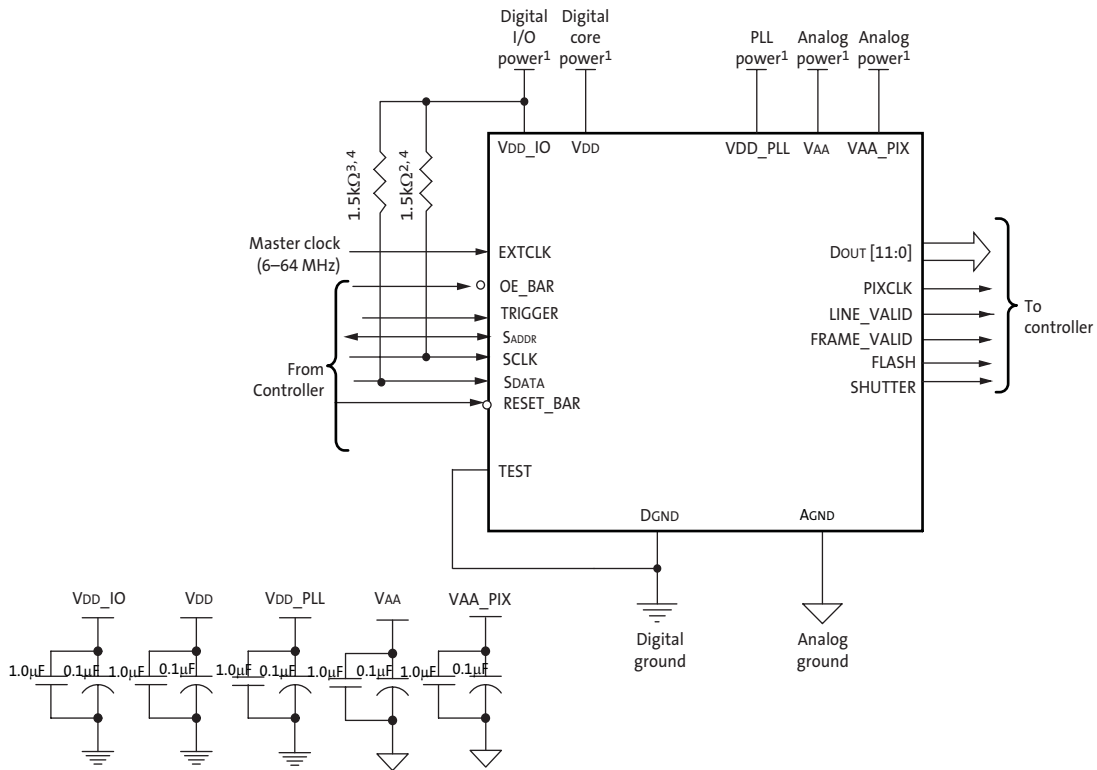
Figure 3: Typical MIPI Connections



- Notes:
1. All power supplies must be adequately decoupled. Aptina 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, Aptina 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. Aptina 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. Aptina 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. Aptina recommends that GND_MIPI be tied to DGND.
 8. VDD_MIPI can be connected to VDD_IO if VDD_IO=2.8V. Otherwise, VDD_MIPI can be tied to VDD_PLL if the Aptina-recommended decoupling capacitors are used. VDD_MIPI is tied to VDD_PLL in both the CLCC and the CSP package. Aptina strongly recommends that VDD_MIPI must be connected to a 2.8V supply and kept separate from VAA and VAA_PIX.



Figure 4: Typical Configuration: Parallel Pixel Data Interface



- Notes:
1. All power supplies must be adequately decoupled. Aptina 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, Aptina 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. Aptina 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. Aptina 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.



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 -64 MHz
OE_BAR	Input	Output enable(active LOW).
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
VDD_HiSPi	Power	1.8V power port to HiSPi PHY
VDD_HiSPi_TX	Power	0.4V-0.8V or 1.7V - 1.9V port to HiSPi Output Driver. Set the High_VCM (R0x306E[9]) bit to 1 when configuring VDD_HiSPi_TX to 1.7 – 1.9V.

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

Name	Type	Description
VAAHV_NPIX	Power	Power supply pin used to program the sensor OTPM (one-time programmable memory). This pin must be connected to ground during normal operation.

Table 6: CSP Package Pin Descriptions

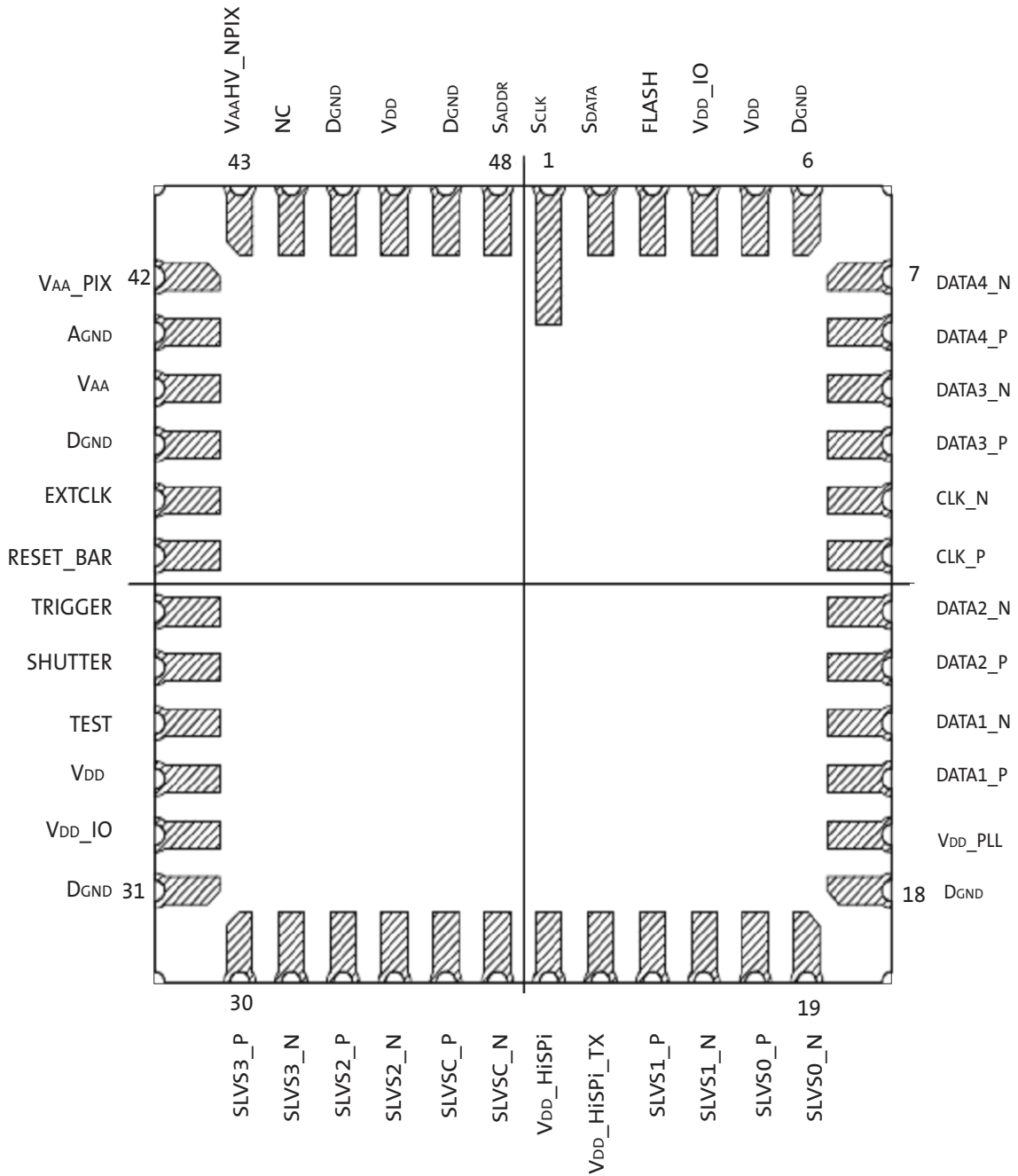
	1	2	3	4	5	6	7	8
A	VAA	VAAHV_NPIX	AGND	AGND	VAA	VDD	TEST	DGND
B	DGND	NC	VAA_PIX	DGND	VDD_IO	TRIGGER	RESET_BAR	EXTCLK
C	VDD	SHUTTER	DGND	SLVSC_P	SLVS3_P	SLVS3_N	SLVS2_N	SLVS2_P
D	SADDR	SCLK	SDATA	FLASH	SLVSC_N	SLVS1_P	VDD_HiSPi_TX	VDD_HiSPi
E	VDD_IO	VDD_IO	CLK_N	CLK_P	DGND	SLVS1_N	SLVS0_N	SLVS0_P
F	DGND	VDD_IO	DGND	DGND	DATA4_P	DATA_N	DATA_P	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: Pins labeled NC (Not Connected) should be left floating.



AR0330: 1/3-Inch CMOS Digital Image Sensor
Pin Descriptions

Figure 5: CLCC Package Pin Descriptions



Note: Pins labeled NC (Not Connected) should be left floating.



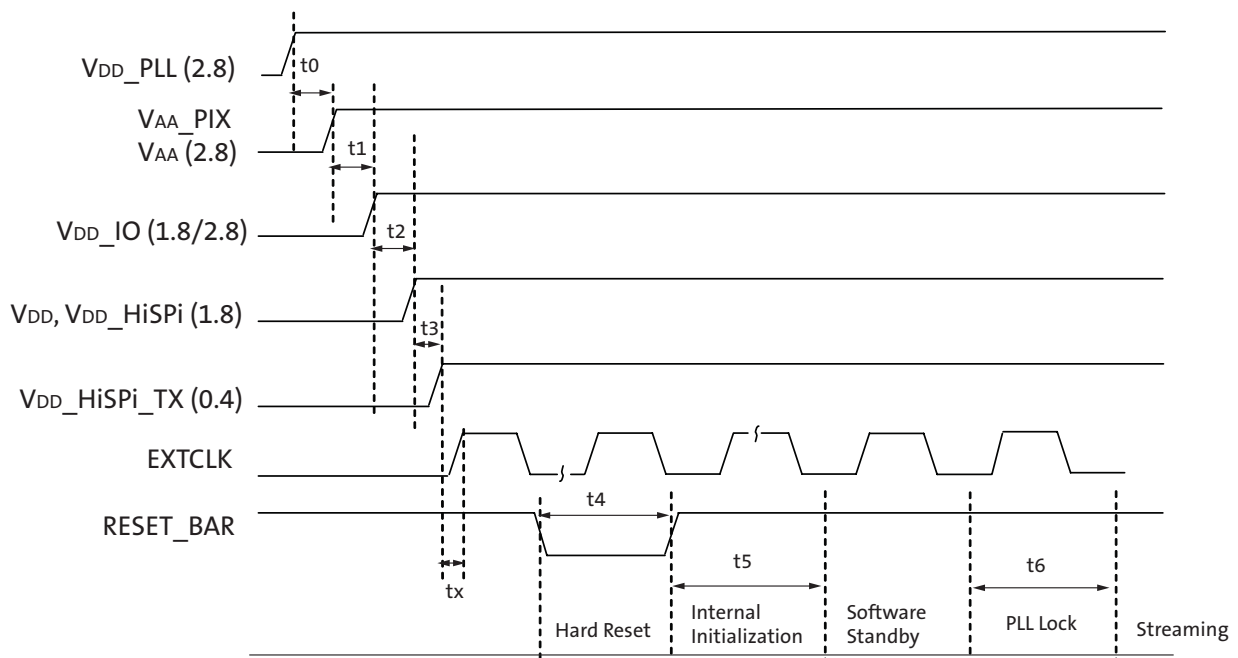
Sensor Initialization

Power-Up Sequence

The recommended power-up sequence for the AR0330 is shown in Figure 6. The available power supplies (V_{DD_IO} , V_{DD_HiSPi} , $V_{DD_HiSPi_TX}$, V_{DD_PLL} , V_{AA} , V_{AA_PIX}) must have the separation specified below.

1. Turn on V_{DD_PLL} power supply.
2. After $100\mu\text{s}$, turn on V_{AA} and V_{AA_PIX} power supply.
3. After $100\mu\text{s}$, turn on V_{DD_IO} power supply.
4. After $100\mu\text{s}$, turn on V_{DD} and V_{DD_HiSPi} power supplies.
5. After $100\mu\text{s}$, turn on $V_{DD_HiSPi_TX}$ power supply.
6. After the last power supply is stable, enable $EXTCLK$.
7. Assert $RESET_BAR$ for at least 1ms.
8. Wait 150000 $EXTCLK$ s (for internal initialization into software standby).
9. Configure PLL, output, and image settings to desired values.
10. Wait 1ms for the PLL to lock.
11. Set streaming mode ($R0x301a[2] = 1$).

Figure 6: Power Up



**Table 7: Power-Up Sequence**

Definition	Symbol	Minimum	Typical	Maximum	Unit
VDD_PLL to VAA/VAA_PIX ³	t0	0	100	–	μS
VAA/VAA_PIX to VDD_IO	t1	0	100	–	μS
VDD_IO to VDD and VDD_HiSPi	t2	10	100	–	μS
VDD and VDD_HiSPi to VDD_HiSPi_TX	t3	10	100	–	μS
External clock settling time	tx	–	30 ¹	–	mS
Hard Reset	t4	1 ²	–	–	mS
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 VDD_PLL is not powered up after the other power supplies. It must be powered before or at least at the same time as the others. If the case happens that VDD_PLL is powered after other supplies then sensor may have functionality issues and will experience high current draw on this supply.
 4. VDD_MIPI is tied to VDD_PLL in both the CLCC and the CSP packages and must be powered to 2.8V. The VDD_HiSPi and VDD_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 (V_{DD_IO} , V_{DD_HiSPi} , $V_{DD_HiSPi_TX}$, V_{DD_PLL} , V_{AA} , V_{AA_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 $V_{DD_HiSPi_TX}$.
4. Turn off V_{DD} and V_{DD_HiSPi} .
5. Turn off V_{DD_IO}
6. Turn off V_{AA}/V_{AA_PIX} .
7. Turn off V_{DD_PLL} .

Figure 7: Power Down

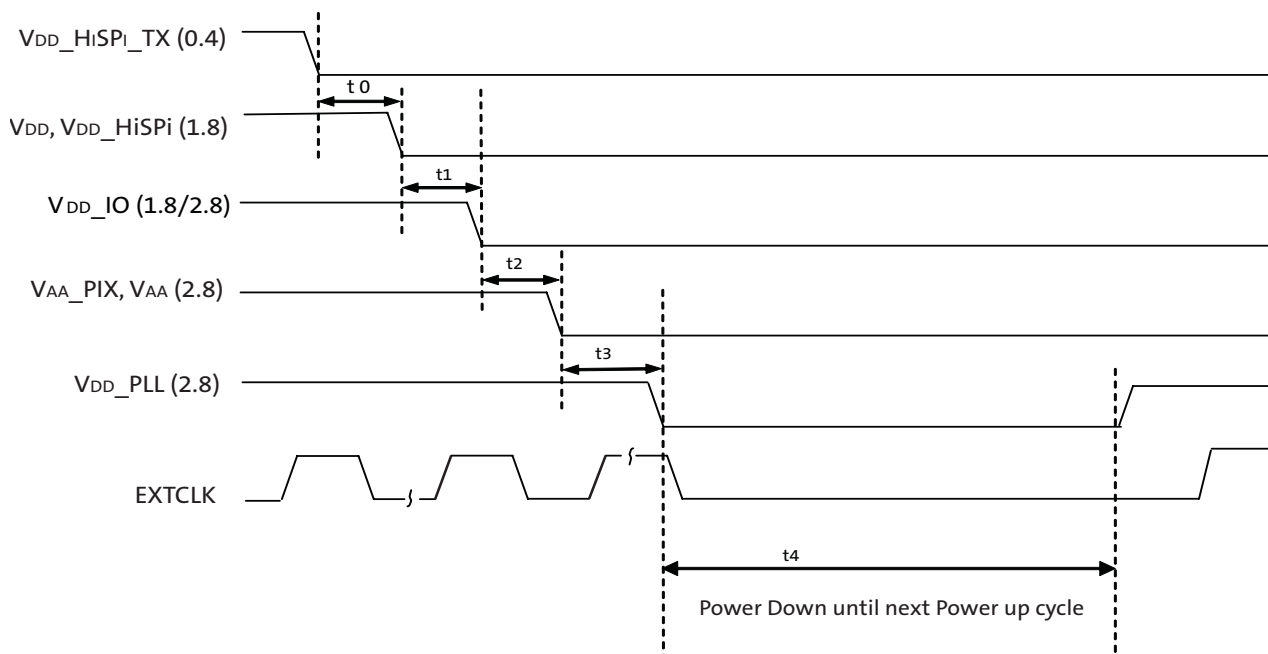


Table 8: Power-Down Sequence

Definition	Symbol	Minimum	Typical	Maximum	Unit
$V_{DD_HiSPi_TX}$ to V_{DD} and V_{DD_HiSPi}	t_0	0	–	–	μs
V_{DD} and V_{DD_HiSPi} to V_{DD_IO}	t_1	0	–	–	μs
V_{DD_IO} to V_{AA}/V_{AA_PIX}	t_2	0	–	–	μs
V_{AA}/V_{AA_PIX} to V_{DD_PLL}	t_3	0	–	–	μs
PwrDn until Next PwrUp Time	t_4	100	–	–	mS

t_4 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

^fEXTCLK = 24 MHz; VDD = 1.8V; VDD_IO = 1.8V; VAA = 2.8V; VAA_PIX = 2.8V; VDD_PLL = 2.8V; VDD_HiSPi = 1.8V, VDD_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	Condition	Symbol	Min	Typ	Max	Unit
Core digital voltage		VDD	1.7	1.8	1.9	V
I/O digital voltage	Parallel pixel data interface	VDD_IO	1.7	1.8	1.9	V
Analog voltage		VAA	2.4	2.8	3.1	V
Pixel supply voltage		VAA_PIX	2.4	2.8	3.1	V
PLL supply voltage		VDD_PLL	2.4	2.8	3.1	V
HiSPi digital voltage		VDD_HiSPi	1.7	1.8	1.9	V
HiSPi I/O digital voltage		VDD_HiSPi_TX	0.3	0.4	0.9	V
Digital operating current				TBD		mA
I/O digital operating current				TBD		mA
Analog operating current				TBD		mA
Pixel supply current				TBD		mA
PLL supply current				TBD		mA
HiSPi digital operating current				TBD		mA
HiSPi I/O digital operating current				TBD		mA
Soft standby (clock on)				TBD		mW

Caution Stresses greater than those listed in Table 10 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.

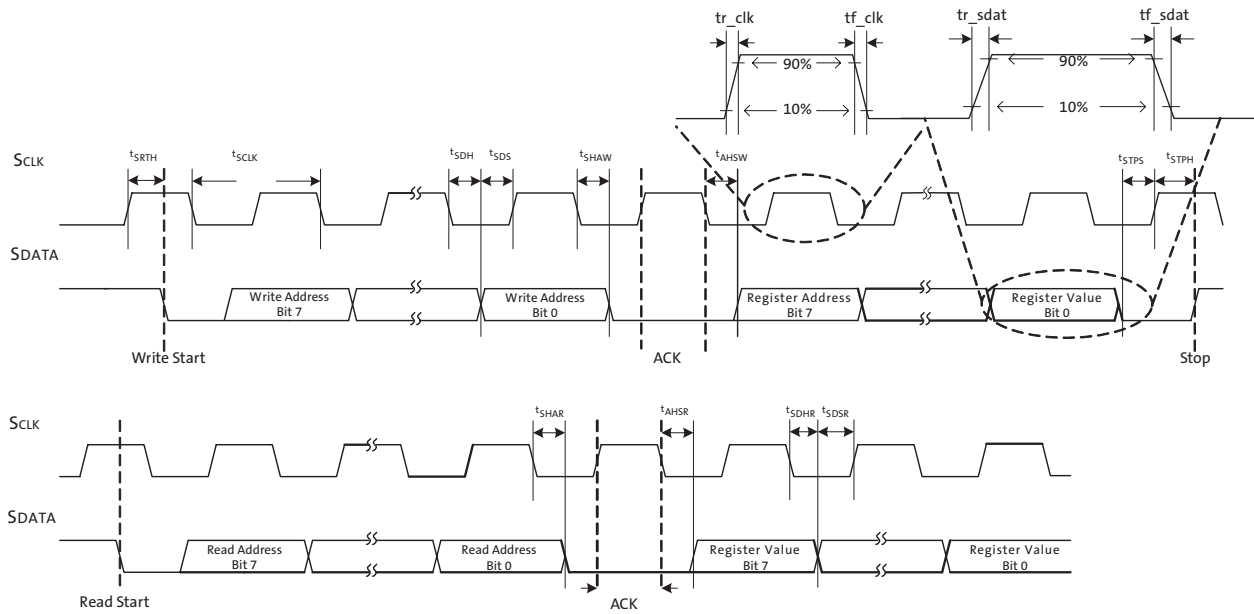
Table 10: Absolute Maximum Ratings

Symbol	Definition	Condition	Min	Max	Unit
VDD_MAX	Core digital voltage		-0.3	2.4	V
VDD_IO_MAX	I/O digital voltage		-0.3	4	V
VAA_MAX	Analog voltage		-0.3	4	V
VAA_PIX	Pixel supply voltage		-0.3	4	V
VDD_PLL	PLL supply voltage		-0.3	4	V
VDD_HiSPi_MAX	HiSPi digital voltage		-0.3	2.4	V
VDD_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.



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 11: Two-Wire Serial Register Interface Electrical Characteristics

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

Symbol	Parameter	Condition	Min	Typ	Max	Unit
V _{IL}	Input LOW voltage		-0.5	0.73	0.3 x V _{DD_IO}	V
I _{IN}	Input leakage current	No pull up resistor; V _{IN} = V _{DD_IO} or DGND	-2		2	μA
V _{OL}	Output LOW voltage	At specified 2mA	0.031	0.032	0.035	V
I _{OL}	Output LOW current	At specified V _{OL} 0.1V			3	mA
C _{IN}	Input pad capacitance				6	pF
C _{LOAD}	Load capacitance					pF

Table 12: Two-Wire Serial Register Interface Timing Specification

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

Symbol	Parameter	Condition	Min	Typ	Max	Unit
f _{SCLK}	Serial interface input clock	—	0	100	400	kHz
	SCLK duty cycle	V _{OD}	45	50	60	%
t _R	SCLK/SDATA rise time				300	μs
t _{SRTS}	Start setup time	Master WRITE to slave	0.6			μs
t _{SRTH}	Start hold time	Master WRITE to slave	0.4			μs
t _{SDH}	SDATA hold	Master WRITE to slave	0.3		0.65	μs
t _{SDS}	SDATA setup	Master WRITE to slave	0.3			μs
t _{SHAW}	SDATA hold to ACK	Master READ to slave	0.15		0.65	μs
t _{AHSW}	ACK hold to SDATA	Master WRITE to slave	0.15		0.70	μs
t _{STPS}	Stop setup time	Master WRITE to slave	0.3			μs



Table 12: Two-Wire Serial Register Interface Timing Specification

$f_{EXTCLK} = 24 \text{ MHz}$; $V_{DD} = 1.8\text{V}$; $V_{DD_IO} = 1.8\text{V}$; $V_{AA} = 2.8\text{V}$; $V_{AA_PIX} = 2.8\text{V}$; $V_{DD_PLL} = 2.8\text{V}$; Output load = 68.5pF; $T_j = 60^\circ\text{C}$

Symbol	Parameter	Condition	Min	Typ	Max	Unit
t_{STPH}	Stop hold time	Master WRITE to slave	0.6			μs
t_{SHAR}	SDATA hold to ACK	Master WRITE to slave	0.3		1.65	μs
t_{AHSR}	ACK hold to SDATA	Master WRITE to slave	0.3		0.65	μs
t_{SDHR}	SDATA hold	Master READ from slave	.012		0.70	μs
t_{SDSR}	SDATA setup	Master READ from slave	0.3			μs

Figure 9: I/O Timing Diagram

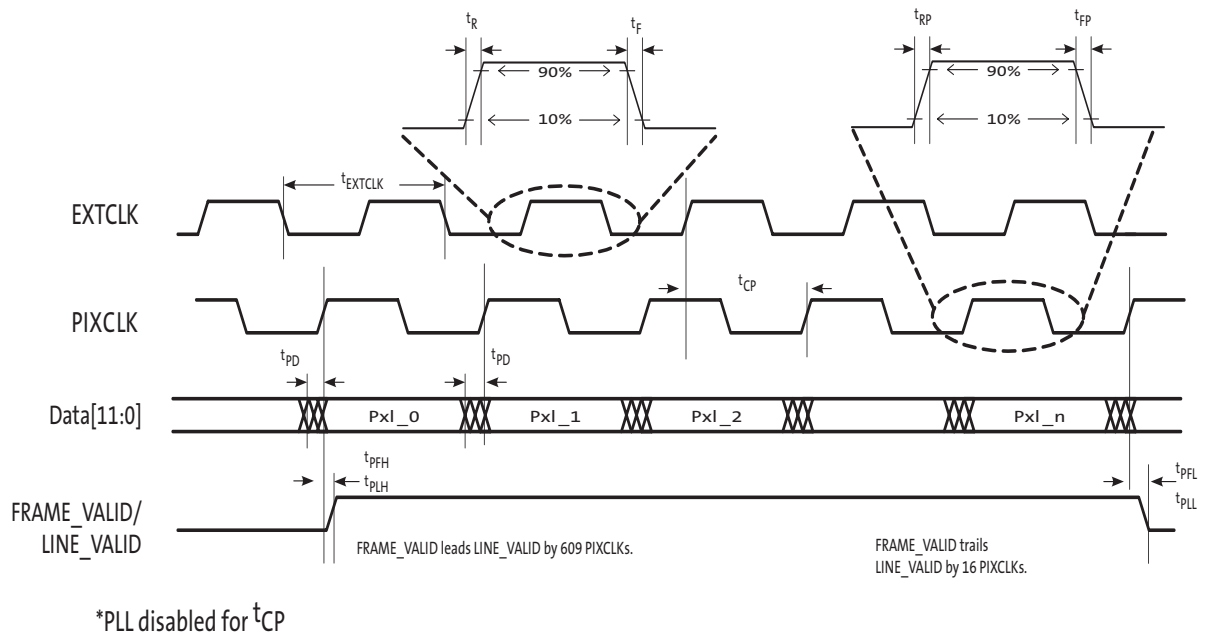


Table 13: I/O Parameters

$f_{EXTCLK} = 24 \text{ MHz}$; $V_{DD} = 1.8\text{V}$; $V_{AA} = 2.8\text{V}$; $V_{AA_PIX} = 2.8\text{V}$; $V_{DD_PLL} = 2.8\text{V}$; Output load = 68.5pF; $T_j = 60^\circ\text{C}$;
 $CLK_OP = 98 \text{ MPixel/s}$

Symbol	Definition	Conditions	Min	Max	Units
V_{IH}	Input HIGH voltage	$V_{DD_IO} = 1.8\text{V}$	1.4	$V_{DD_IO} + 0.3$	V
		$V_{DD_IO} = 2.8\text{V}$	2.4		
V_{IL}	Input LOW voltage	$V_{DD_IO} = 1.8\text{V}$	$GND - 0.3$	0.4	
		$V_{DD_IO} = 2.8\text{V}$	$GND - 0.3$	0.8	
I_{IN}	Input leakage current	No pull-up resistor; $V_{IN} = V_{DD}$ OR $DGND$	-20	20	μA
V_{OH}	Output HIGH voltage	At specified I_{OH}	$V_{DD_IO} - 0.4\text{V}$	-	V
V_{OL}	Output LOW voltage	At specified I_{OL}	-	0.4	V
I_{OH}	Output HIGH current	At specified V_{OH}	-	-12	mA
I_{OL}	Output LOW current	At specified V_{OL}	-	9	mA
I_{OZ}	Tri-state output leakage current		-	10	μA


 AR0330: 1/3-Inch CMOS Digital Image Sensor
 Electrical Characteristics
Table 14: I/O Timing

$f_{EXTCLK} = 24 \text{ MHz}$; $V_{DD} = 1.8\text{V}$; $V_{DD_IO} = 1.8\text{V}$; $V_{AA} = 2.8\text{V}$; $V_{AA_PIX} = 2.8\text{V}$; $V_{DD_PLL} = 2.8\text{V}$; Output load = 68.5pF;
 $T_J = 60^\circ\text{C}$; $CLK_OP = 98 \text{ MPixel/s}$

Symbol	Definition	Conditions	Min	Typ	Max	Units
f_{EXTCLK}	Input clock frequency	PLL enabled	2	24	64	MHz
t_{EXTCLK}	Input clock period	PLL enabled	166	41	20	ns
t_R	Input clock rise time		0.1	–	1	V/ns
t_F	Input clock fall time		0.1	–	1	V/ns
	Clock duty cycle		45	50	55	%
t_{JITTER}	Input clock jitter		–	–	0.3	ns
Output pin slew	Fastest	$C_{LOAD} = 15\text{pF}$	–	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 15: Parallel I/O Rise Slew Rate

$f_{EXTCLK} = 24 \text{ MHz}$; $V_{DD} = 1.8\text{V}$; $V_{AA} = 2.8\text{V}$; $V_{AA_PIX} = 2.8\text{V}$; $V_{DD_PLL} = 2.8\text{V}$; Output load = 68.5pF;
 $T_J = 60^\circ\text{C}$; $CLK_OP = 98 \text{ MPixel/s}$

V _{DD_IO}	Parallel Slew Rate (R0x306E[15:13])								Units
	0	1	2	3	4	5	6	7	
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	



Figure 10: HiSPi Eye Diagram for Both Clock and Data Signals

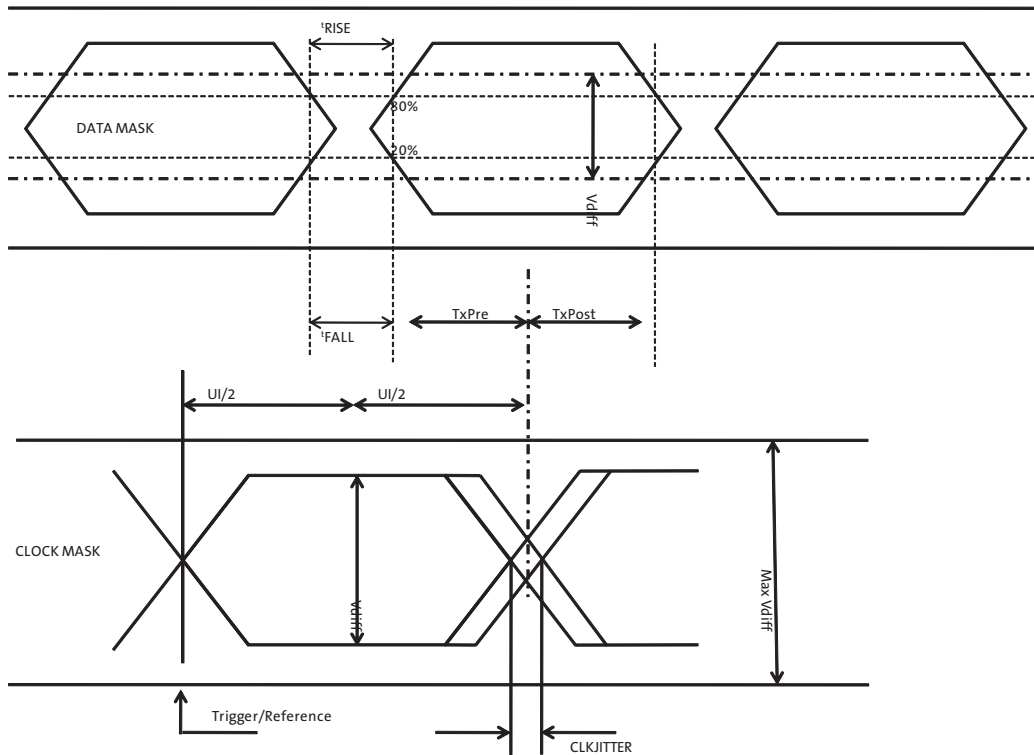


Table 16: HiSPi Rise and Fall Times at 660 Mbps

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

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

Table 17: HiSPi Rise and Fall Times at 360 MHz

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

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



Figure 11: HiSPi Skew Between Data Signals Within the PHY

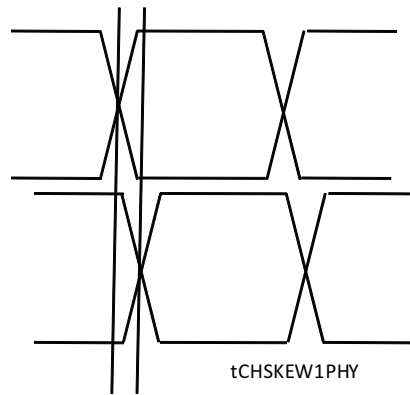


Table 18: Channel, PHY and Intra-PHY Skew

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

Data Lane Skew in Reference to Clock	tCHSKEW1PHY	-150	ps
--------------------------------------	-------------	------	----

Table 19: Clock DLL Steps

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

Clock DLL Step	1	2	3	4	5	Step
Delay at 660 Mbps	0.25	0.375	0.5	0.625	0.75	UI
Eye opening at 660 Mbps	0.85	0.78	0.71	0.71	0.69	UI
Eye opening at 360 MHz	0.89	0.83	0.81	0.60	0.46	UI

Note: The Clock DLL Steps 6 and 7 are not recommended by Aptina for the AR0330 Rev. 1.

Table 20: Data DLL Steps

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

Data DLL Step	1	2	4	6	Step
Delay at 660 Mbps	0.25	0.375	0.625	0.875	UI
Eye opening at 660 Mbps	0.79	0.84	0.71	0.61	UI
Eye opening at 360 MHz	0.85	0.83	0.82	0.77	UI

Note: The Data DLL Steps 3, 5, and 7 are not recommended by Aptina for the AR0330 Rev. 1.



Sequencer

The sequencer digital block determines the order and timing of operations required to sample pixel data from the array. During sensor initialization, the sequencer will be programmed into memory from the sensor OTPM (One Time Programmable Memory). The OTPM is configured during production.

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".

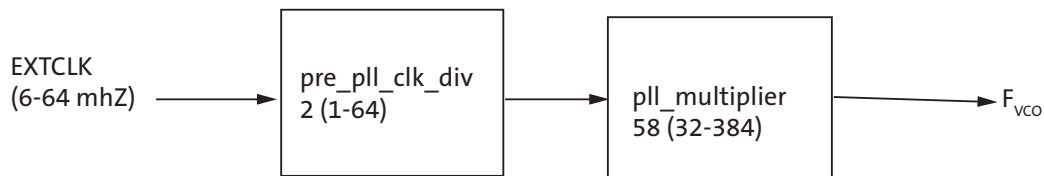
Read the instructions stored in the sequencer.

1. Place the sensor in standby.
2. Write 0xC000 to R0x3088 ("seq_ctrl_port").
3. Sequentially read 2-bytes at a time from R0x3086.

Sensor PLL

VCO

Figure 12: Relationship Between Readout Clock and Peak Pixel Rate

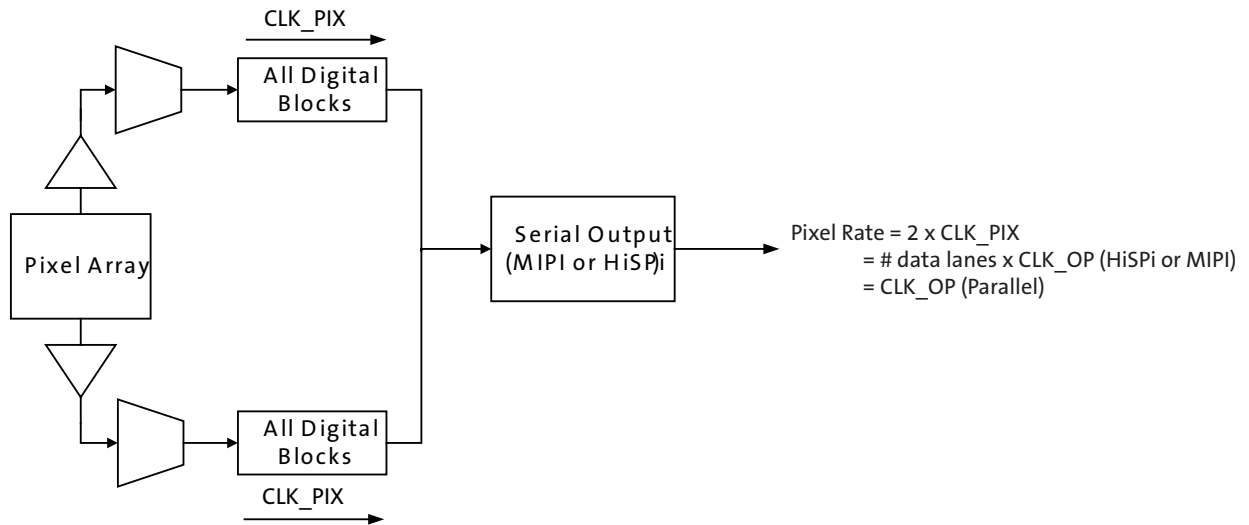


The sensor contains a phase-locked loop (PLL) that is used for timing generation and control. The required VCO clock frequency is attained through the use of a pre-PLL clock divider followed by a multiplier. The multiplier is followed by set of dividers used to generate the output clocks required for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

Dual Readout Paths

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

Figure 13: Sensor Dual Readout Paths

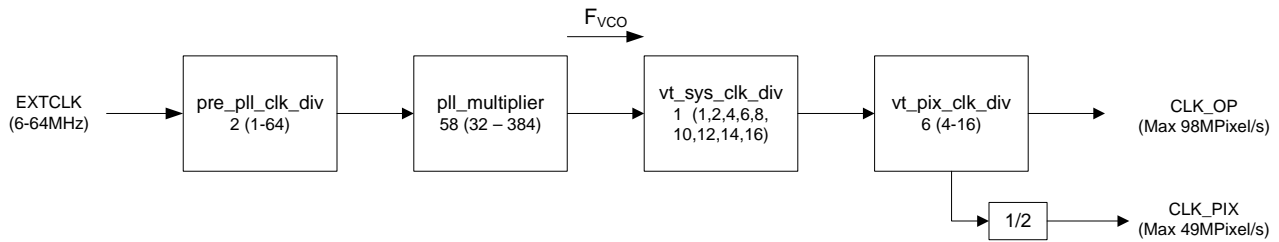


The sensor row timing calculations refers to each data-path individually. For example, the sensor default configuration uses 1248 clocks per row (line_length_pck) to output 2304 active pixels per row. The aggregate clocks per row seen by the receiver will be 2496 clocks (1248 x 2 readout paths).

Parallel PLL Configuration

Figure 14: 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 21: PLL Parameters for the Parallel Interface

Parameter	Symbol	Min	Max	Unit
External Clock	EXTCLK	6	64	MHz
VCO Clock	F_{VCO}	384	768	MHz
Readout Clock	CLK_PIX		49	Mpixel/s
Output Clock	CLK_OP		98	Mpixel/s

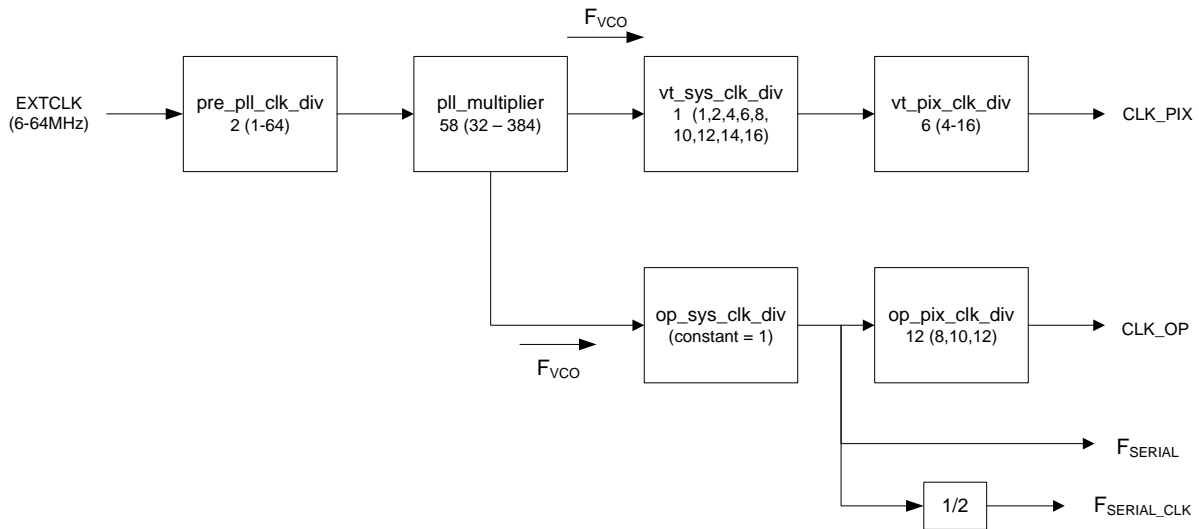


Table 22: Example PLL Configuration for the Parallel Interface

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

Serial PLL Configuration

Figure 15: 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 31.

Table 23: PLL Parameters for the Serial Interface

Parameter	Symbol	Min	Max	Unit
External Clock	EXTCLK	6	64	MHz
VCO Clock	F_{VCO}	384	768	MHz
Readout Clock	CLK_PIX		98	Mpixel/s
Output Clock	CLK_OP		49	Mpixel/s
Output Serial Data Rate Per Lane	F_{SERIAL}	300 (HiSpi) 384 (MIPI)	700 (HiSpi) 768 (MIPI)	Mbps
Output Serial Clock Speed Per Lane	F_{SERIAL_CLK}	150 (HiSpi) 192 (MIPI)	350 (HiSpi) 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 ($2 \times \text{CLK_PIX}$).
 - 4-lane: $4 \times \text{CLK_OP} = 2 \times \text{CLK_PIX} = \text{Pixel Rate}$ (max: 196 Mpixel/s)
 - 2-lane: $2 \times \text{CLK_OP} = 2 \times \text{CLK_PIX} = \text{Pixel Rate}$ (max: 98 Mpixel/s)
 - 1-lane: $1 \times \text{CLK_OP} = 2 \times \text{CLK_PIX} = \text{Pixel Rate}$ (max: 76 Mpixel/s)

Table 24: 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
$F_{\text{SERIAL_CLK}}$	294	245	294	245	384	384	384	MHz
CLK_PIX	98	98	49	49	64	76.8	96	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 26 on page 27 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 25.

Table 25: 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 26.

Table 26: Configuration of the Pixel Data Interface

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



High Speed Serial Pixel Data Interface

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

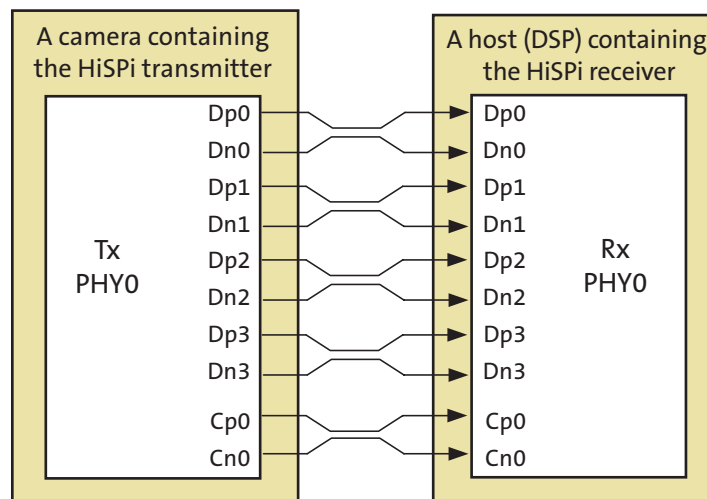
- SLVSC_P
- SLVSC_N
- SLVS0_P
- SLVS0_N
- SLVS1_P
- SLVS1_N
- SLVS2_P
- SLVS2_N
- SLVS3_P
- SLVS3_N

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

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

The HiSPi interface building block is a unidirectional differential serial interface with four data and one double data rate (DDR) clock lanes. One clock for every four serial data lanes is provided for phase alignment across multiple lanes. Figure 16 shows the configuration between the HiSPi transmitter and the receiver.

Figure 16: HiSPi Transmitter and Receiver Interface Block Diagram



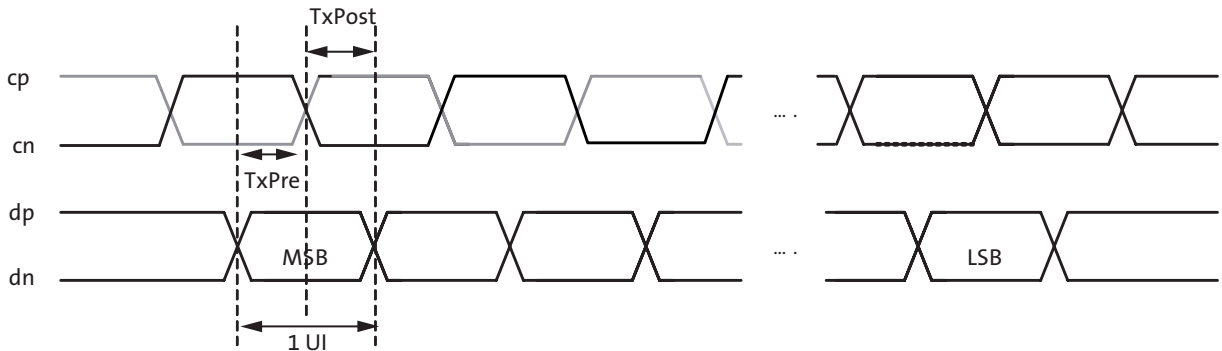


HiSPi Physical Layer

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 17 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 17: 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 18: Block Diagram of DLL Timing Adjustment

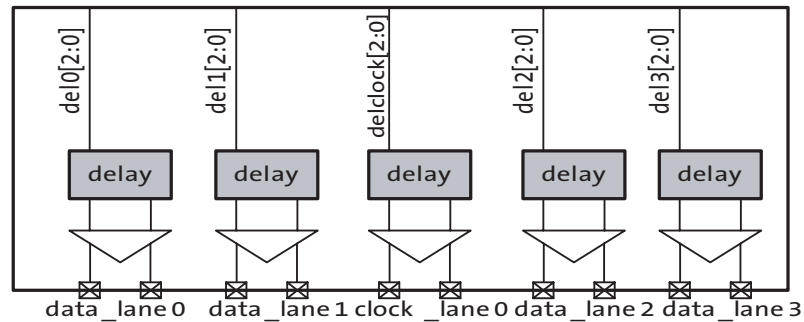




Figure 19: Delaying the clock_lane with Respect to data_lane

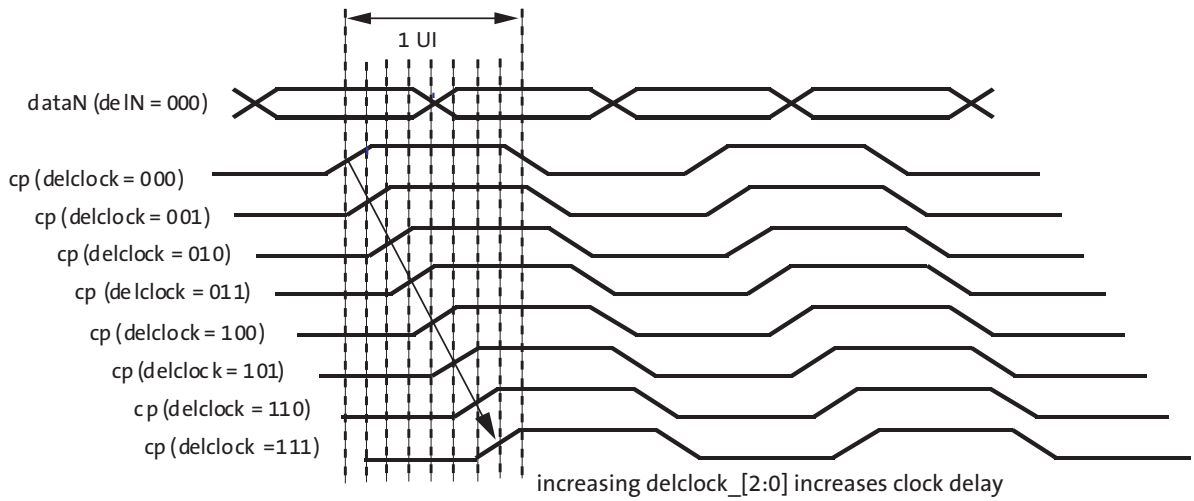
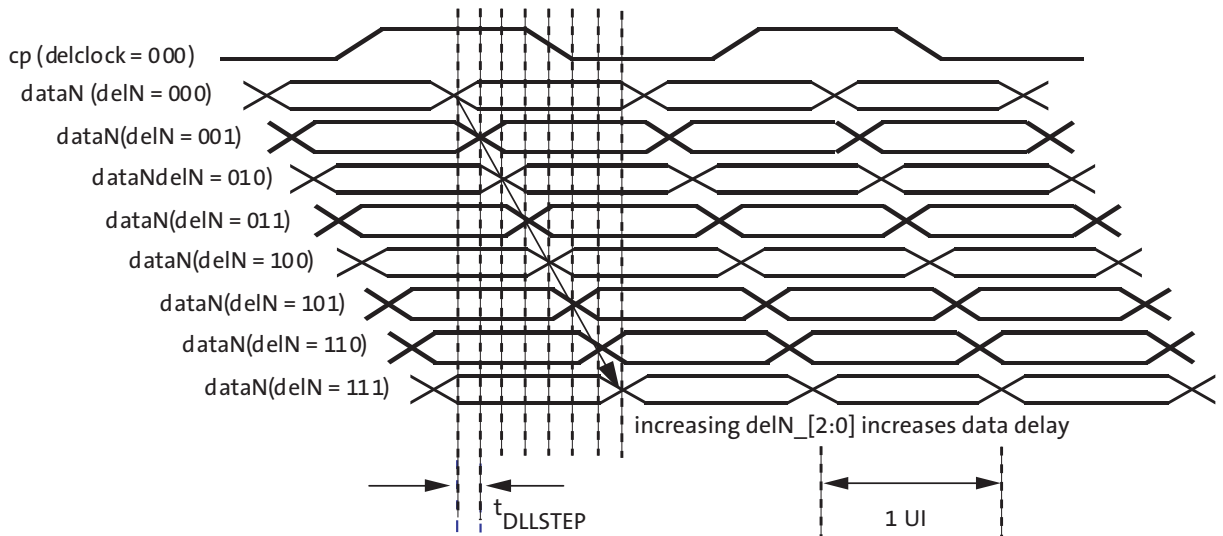


Figure 20: Delaying data_lane with Respect to the clock_lane



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 49.
- 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

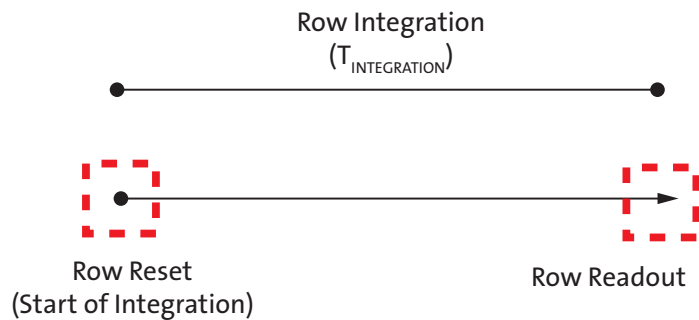
The MIPI timing registers must be configured differently for 10-bit or 12-bit modes. These modes should be configured when the sensor streaming is disabled. See Table 27 on page 32

Table 27: 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 21: Integration Control in ERS Readout



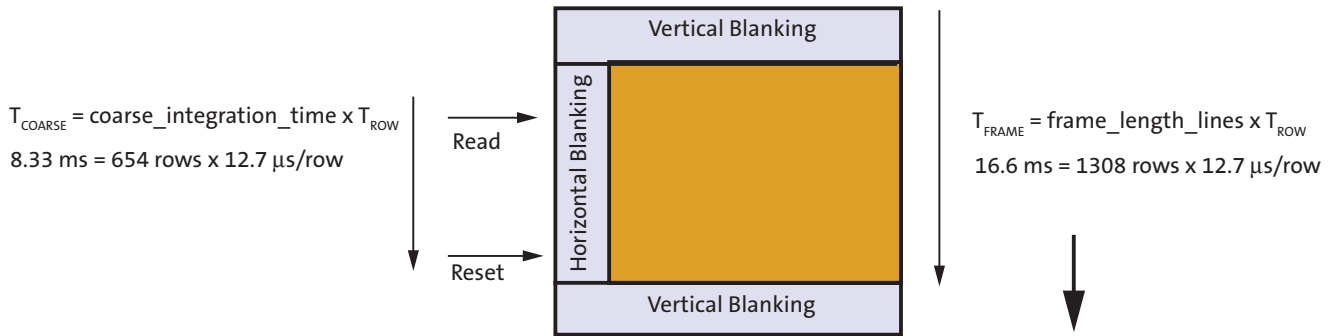
A pixel's integration time is defined by the number of clock periods between a row's reset and read operation. Both the read followed by the reset operations occur within a row period (T_{ROW}) where the read and reset may be applied to different rows. The read and reset operations will be applied to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

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

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

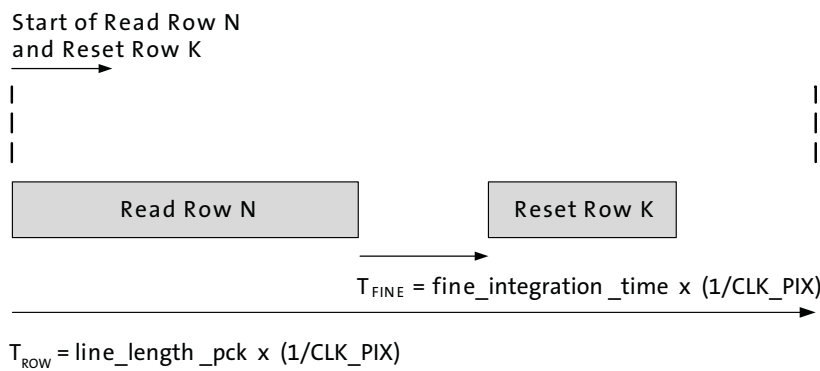
$$T_{COARSE} = T_{ROW} * \text{coarse_integration_time} \quad (\text{EQ 1})$$

Figure 22: 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 23: Row Read and Row Reset Showing Fine Integration

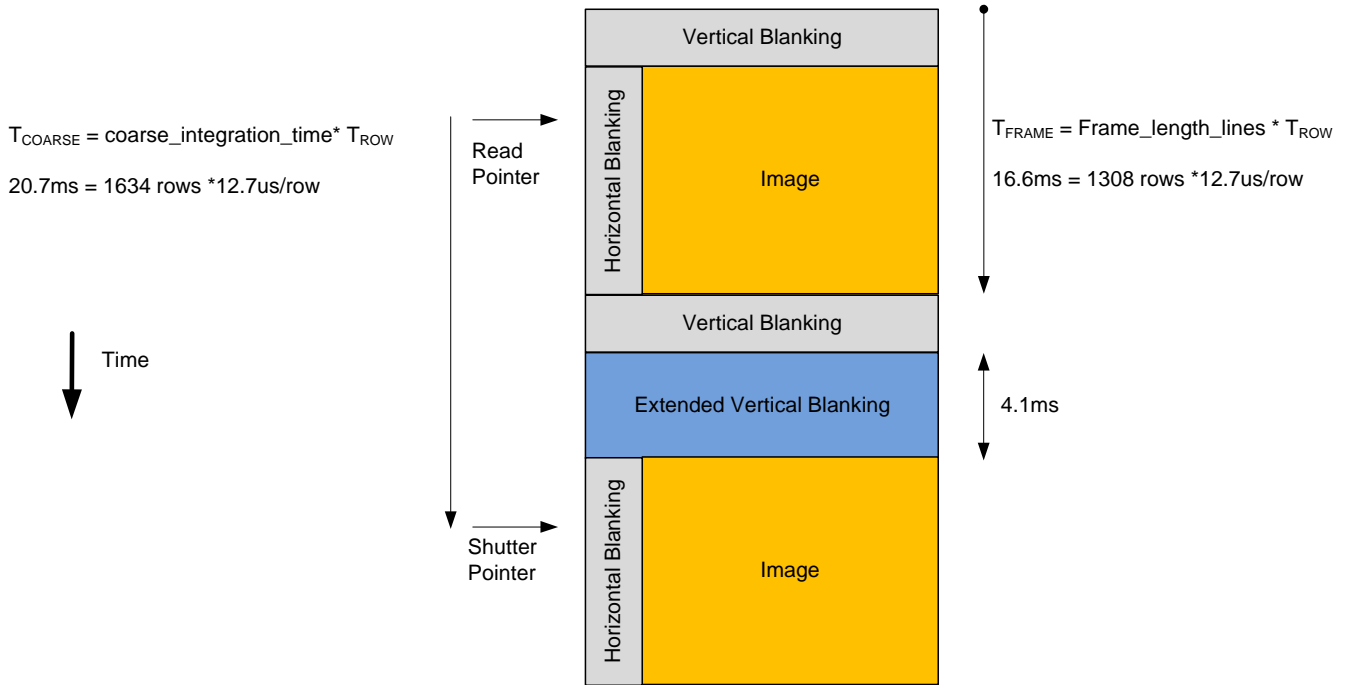


$$T_{FINE} = \text{fine_integration_time} / \text{clk_pix} \tag{EQ 2}$$

The maximum allowed value for *fine_integration_time* is *line_length_pck* - 920.



Figure 24: The Row Integration Time is Greater Than the Frame Readout Time

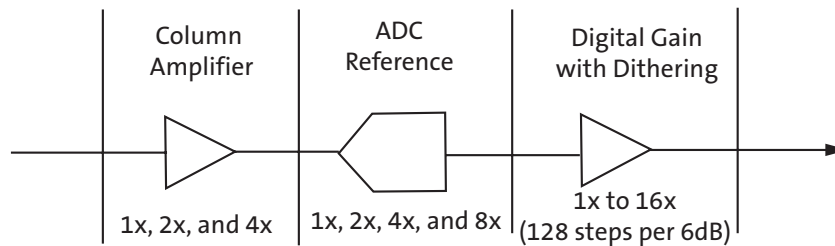


The minimum frame-time is defined by the number of row periods per frame and the row period. The sensor frame-time will increase if the *coarse_integration_time* is set to a value equal to or greater than the *frame_length_lines*. The maximum integration time can be limited to the frame time by setting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

Gain Stages

The analog gain stages of the AR0330 sensor are shown in Figure 25. 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 25: Gain Stages in AR0330 Sensor





The level of analog gain applied is controlled by the coarse_gain register. The analog readout circuitry can be configured differently for each analog gain level. The recommended analog gain tables listed in Table 28 can be configured using the registers listed in Table 29. It is recommended that these registers are configured before streaming images.

Table 28: Recommended Sensor Gain Tables

Coarse_gain register	Global Analog Gain	
'00	1x	1x ADC Gain
'01	2x	2x ADC Gain
'02	4x	4x ADC Gain
'03	8x	8x ADC Gain

Table 29: Recommended Registers to Configure Sensor Gain Table

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

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

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:

1. 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.

2. Global reset mode

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

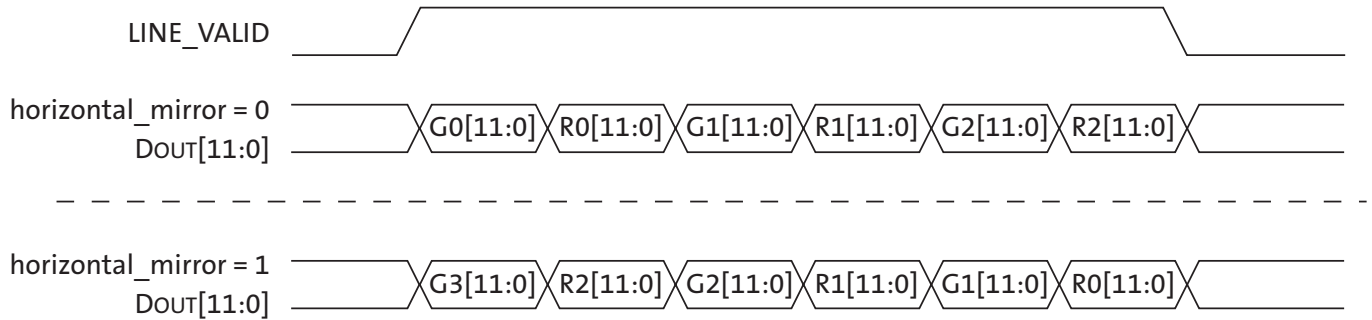
Readout Modes

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 + 1` and ends at `x_addr_start`. Figure 26 on page 37 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.



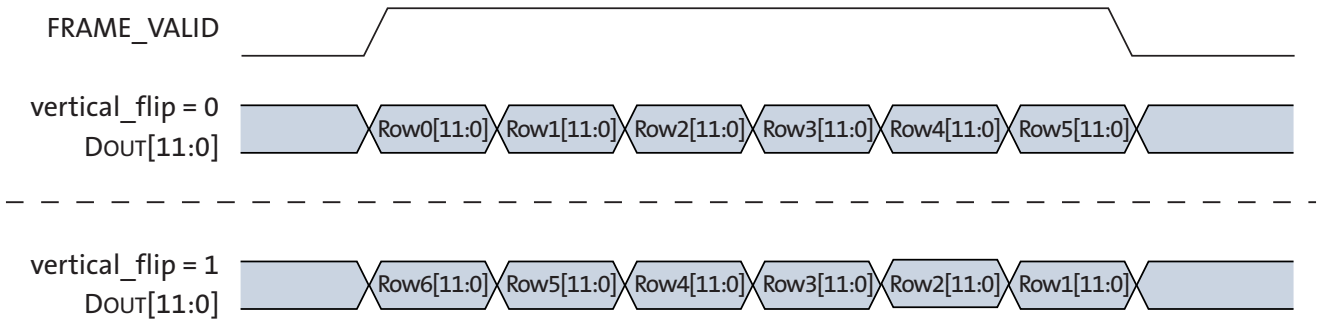
Figure 26: Effect of Horizontal Mirror on Readout Order



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.

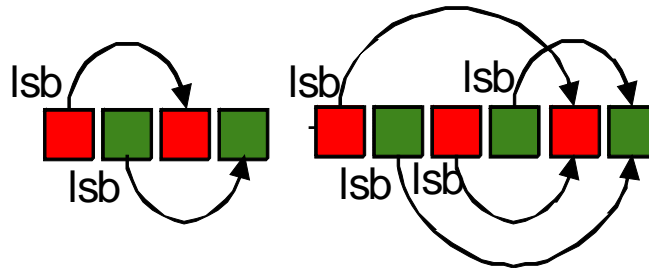
Figure 27: Effect of Vertical Flip on Readout Order



Subsampling

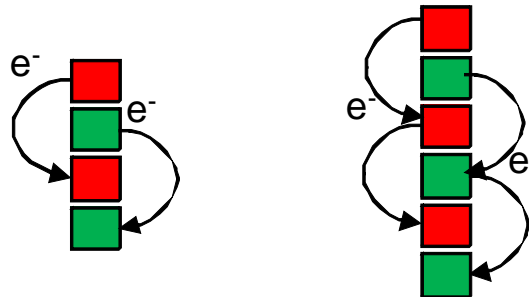
The AR0330 supports subsampling. Subsampling allows the sensor to read out a smaller set of active pixels by either skipping or binning pixels within the readout window. The working modes described in the data sheet that use subsampling are configured to use either 2x2 or 3x3 subsampling.

Figure 28: 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 29: 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.

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

Table 30: Available Skip and Bin Modes in the AR0330 Sensor

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

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



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

$$x \text{ subsampling factor} = \frac{1 + x_odd_inc}{2}$$

$$y \text{ subsampling factor} = \frac{1 + y_odd_inc}{2}$$

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 31: Configuration for Horizontal Subsampling

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



Table 32: Configuration for Vertical Subsampling

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



Sensor Frame Rate

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

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

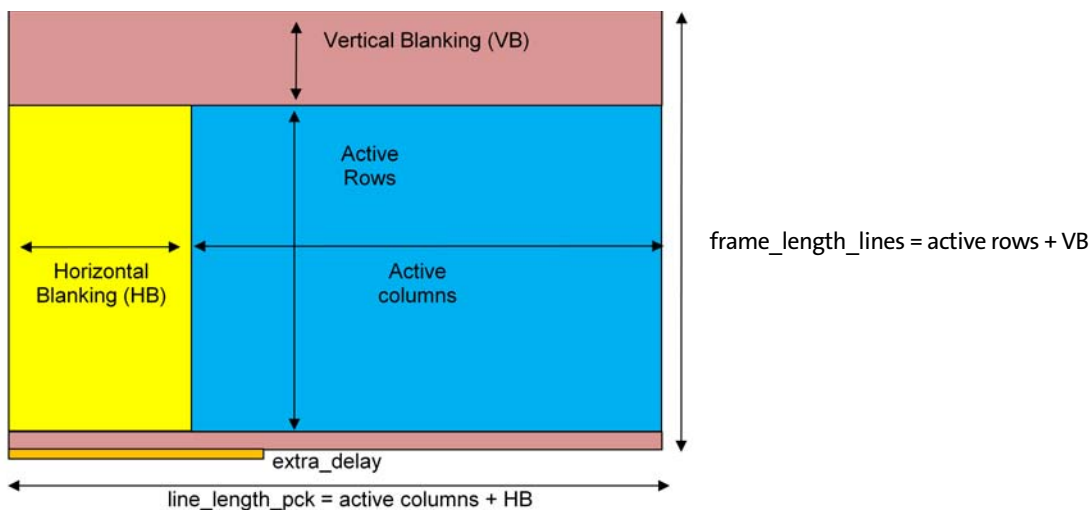
$$fps = 1/T_{FRAME} \quad (EQ\ 3)$$

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. ($T_{ROW} = line_length_pck \times 1/CLK_PIX$)
- 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*)

$$T_{FRAME} = 1 / (CLK_PIX) \times [frame_length_lines \times line_length_pck + extra_delay] \quad (EQ\ 4)$$

Figure 30: 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 13 on page 24, 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:

$$920(\text{ADC_HIGH_SPEED}) = 0$$

or

$$832(\text{ADC_HIGH_SPEED}) = 1(0) \quad (\text{EQ } 5)$$

Digital Readout Limitation:

$$\frac{1}{3} \times \left[\frac{x_addr_end - x_addr_start}{(x_odd_inc + 1) \times 0.5} \right] \quad (\text{EQ } 6)$$

Output Interface Limitations:

$$\frac{1}{2} \times \left[\frac{x_addr_end - x_addr_start}{(x_odd_inc + 1) \times 0.5} \right] + 96 \quad (\text{EQ } 7)$$

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.

$$\text{Minimum_frame_length_lines} = \frac{y_addr_end - y_addr_start}{(y_odd_inc + 1)/2} + \text{min_frame_length_lines} \quad (\text{EQ } 8)$$

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 33: Minimum Vertical Blanking Configuration

R0x3180[0x00F0]	OB Rows	min_frame_length_lines
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

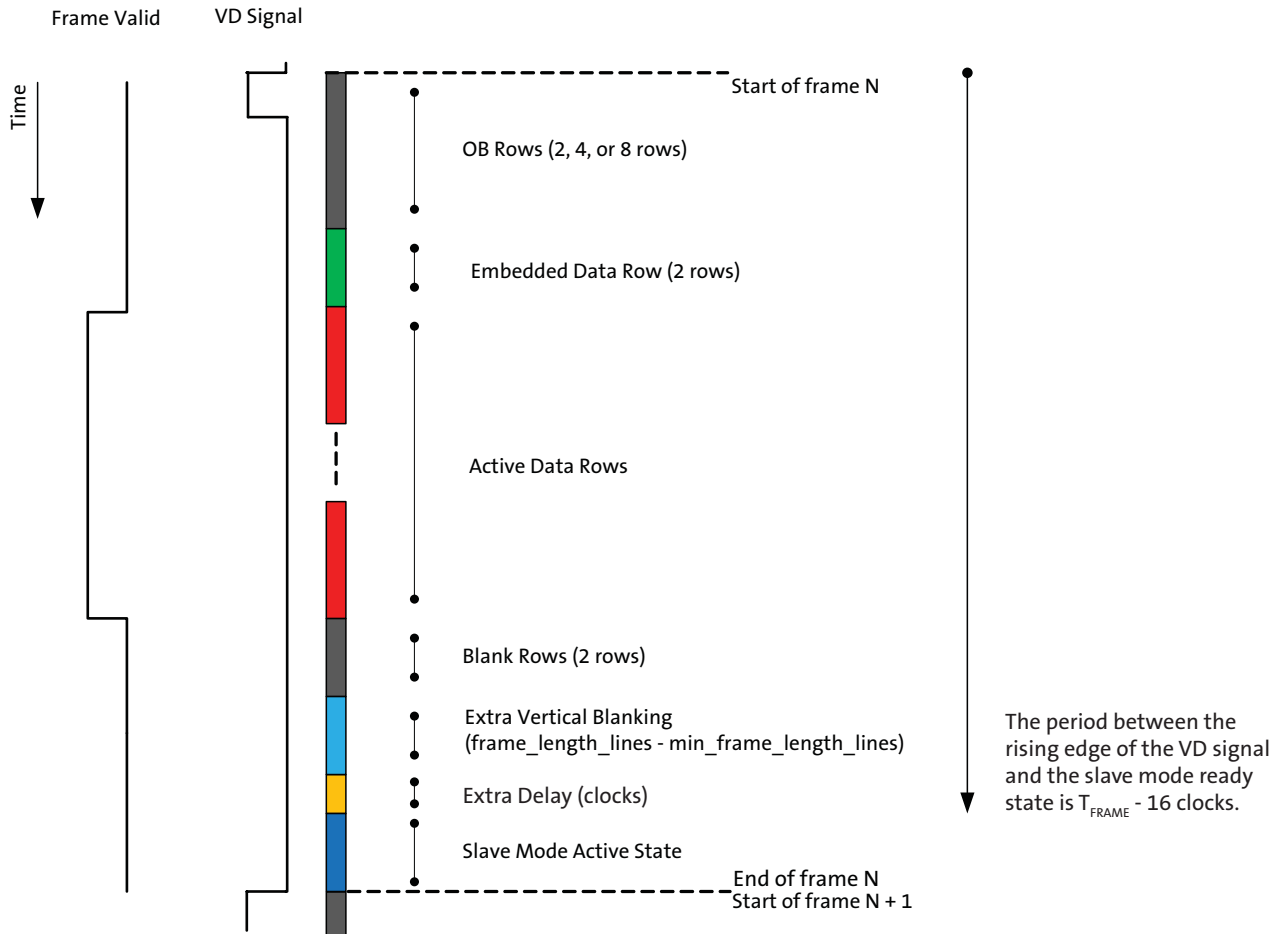
The locations of the OB rows, embedded rows, and blank rows within the frame readout are identified in Figure 31: “Slave Mode Active State and Vertical Blanking,” on page 43.



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.

Figure 31: Slave Mode Active State and Vertical Blanking



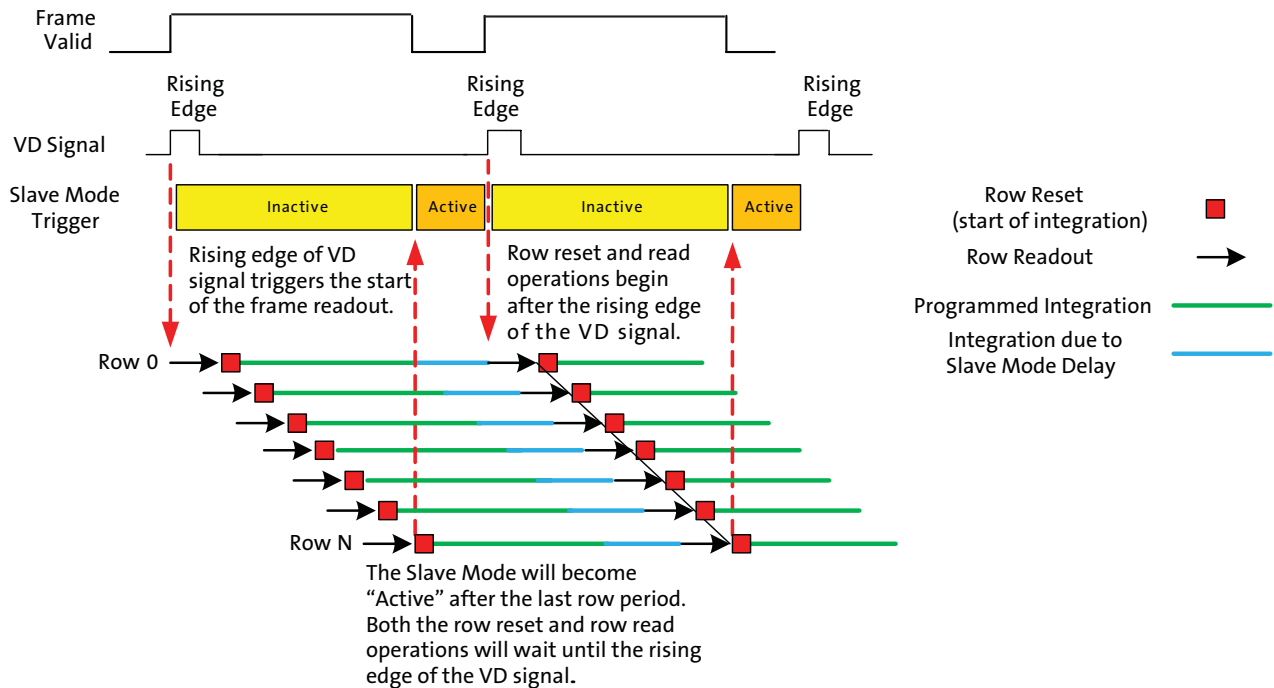
If the slave mode is disabled, the new frame will begin after the extra delay period is finished.

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 32: 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.



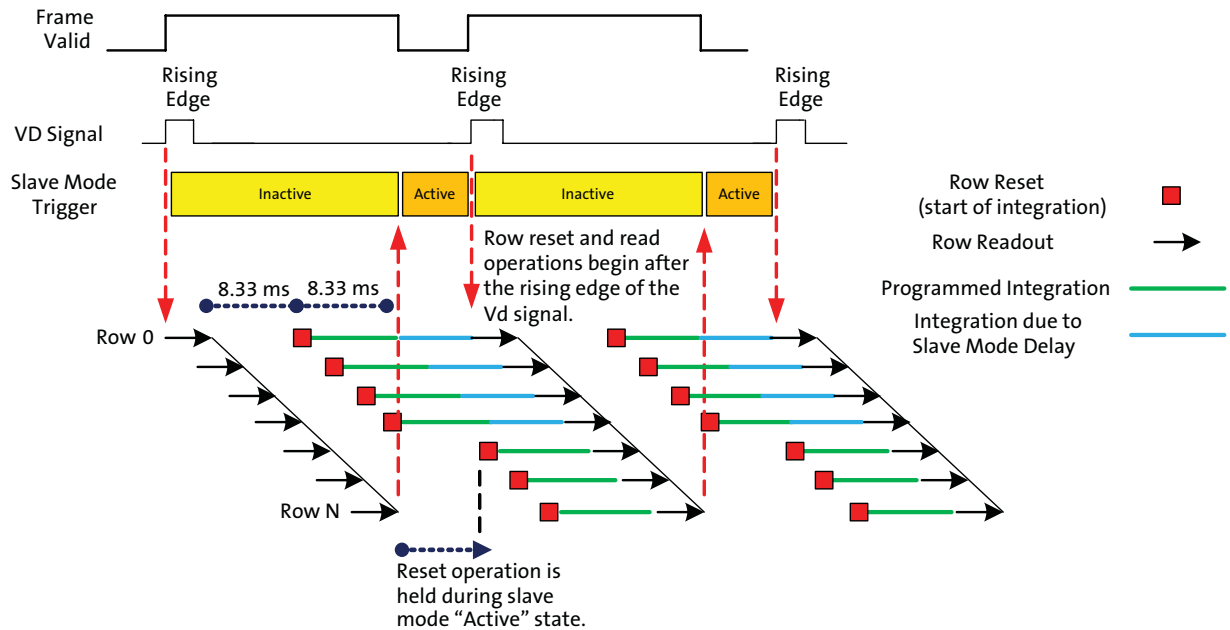
The row shutter and read operations will stop when the slave mode becomes active and is waiting for the VD signal. The following should be considered when configuring the sensor to use the slave mode:

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

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



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

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



Frame Readout

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

Figure 34: 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.

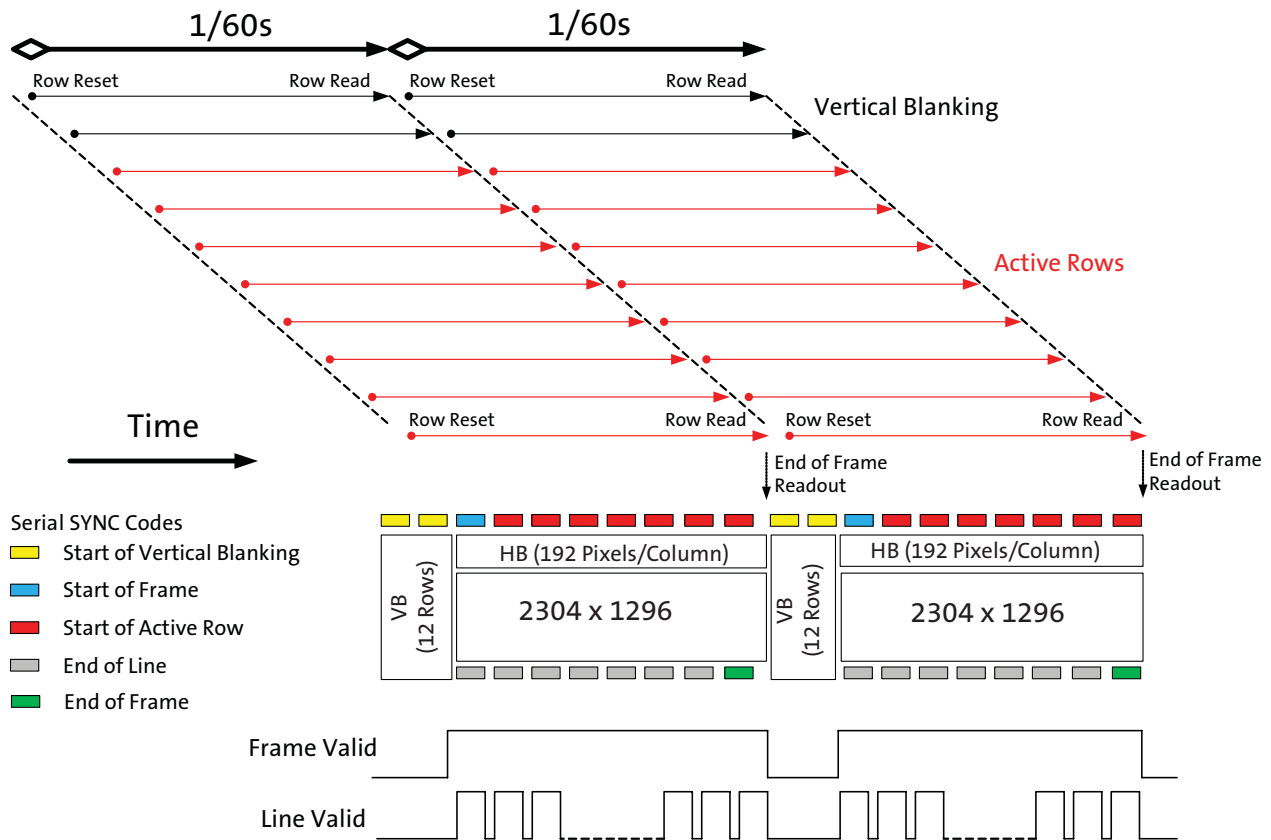


Figure 34 aligns the frame integration and readout operation to the sensor output. It also shows the sensor output using the HiSPi Streaming SP protocol. Different sensor protocols will list different SYNC codes.

Table 34: Serial SYNC Codes Included with Each Protocol Included with the AR0330 Sensor

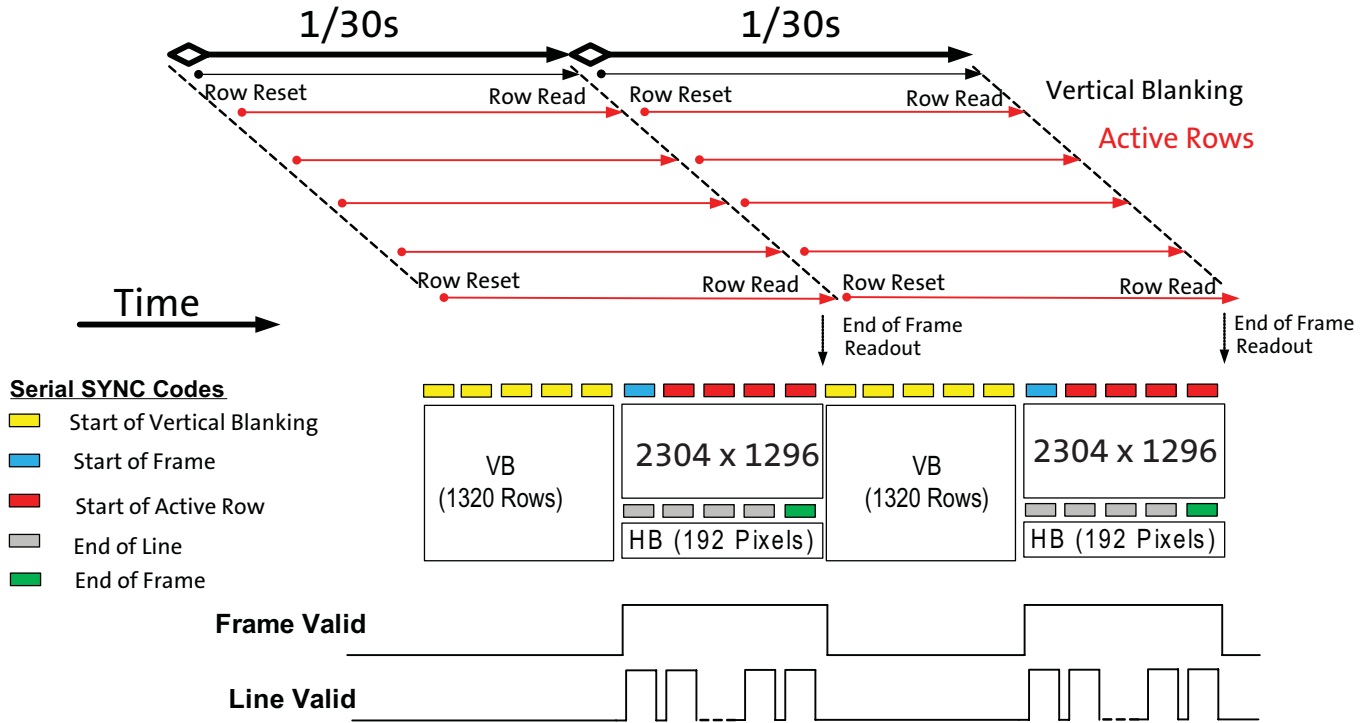
Interface/Protocol	Start of Vertical Blanking Row (SOV)	Start of Frame (SOF)	Start of Active Line (SOA)	End of Line (EOL)	End of Frame (EOF)
Parallel	Parallel interface uses FRAME VALID(FV) and LINE VALID (LV) outputs to denote start and end of line and frame.				
HiSPi Streaming S	Yes	Send SOV	Yes	No 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 35 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 35: 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

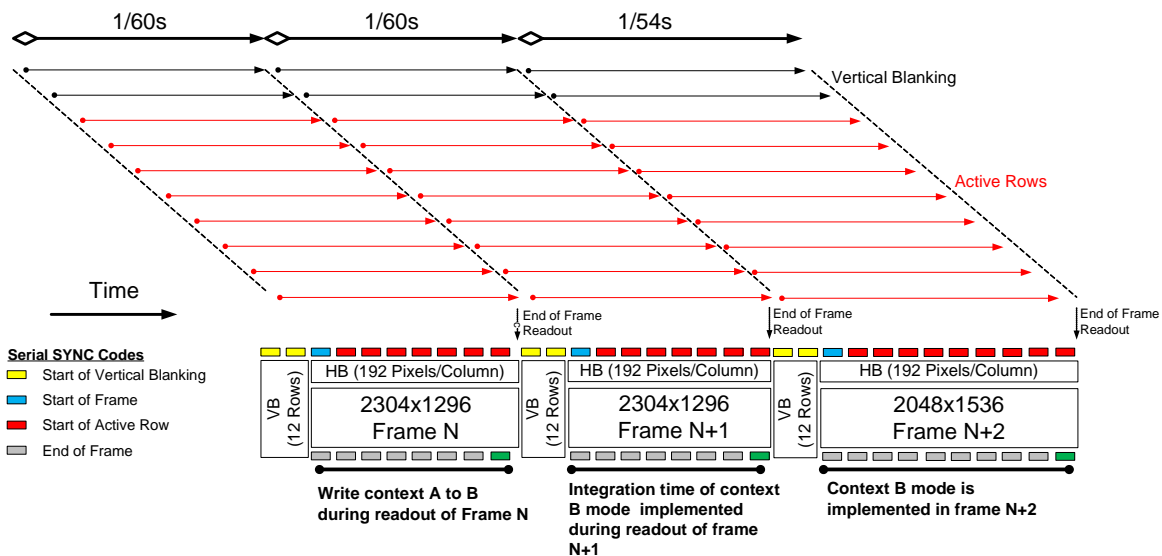
In the AR0330, the user may switch between two full register sets A and B by writing to a context switch change bit in R0x30B0[13]. When the context switch is configured to context A the sensor will reference the “Context A Registers”. If the context switch is changed from A to B during the readout of 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 35: 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

Figure 36: 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 36: 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 37. 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 37: 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 I²C 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.

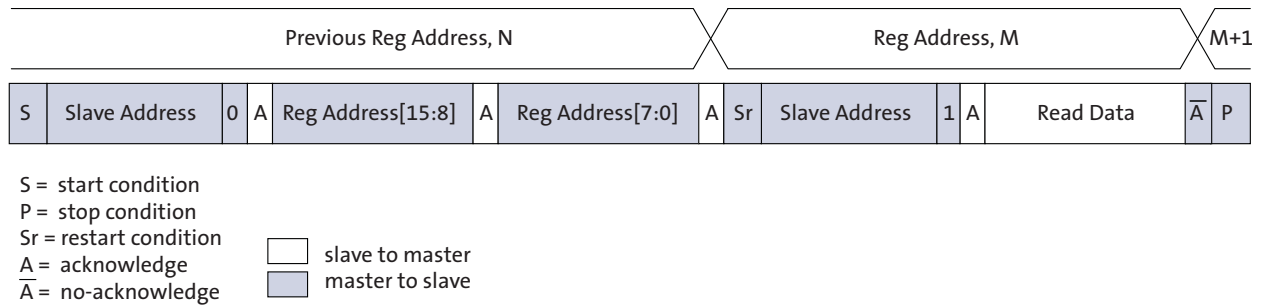
If the request was a READ, the master sends the 8-bit write slave address/data direction byte and 16-bit register address, the same way as with a WRITE request. The master then generates a (re)start condition and the 8-bit read slave address/data direction byte, and clocks out the register data, eight 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 37) 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 37 shows how the internal register address maintained by the AR0330 is loaded and incremented as the sequence proceeds.

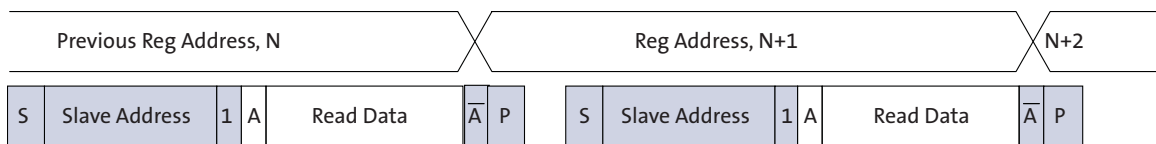
Figure 37: Single READ From Random Location



Single READ From Current Location

This sequence (Figure 38) 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 38: Single READ From Current Location

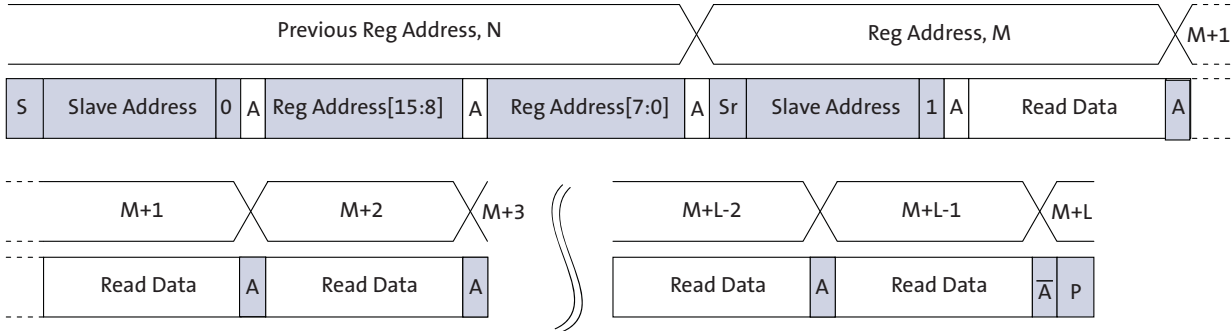




Sequential READ, Start From Random Location

This sequence (Figure 39) starts in the same way as the single READ from random location (Figure 37). 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.

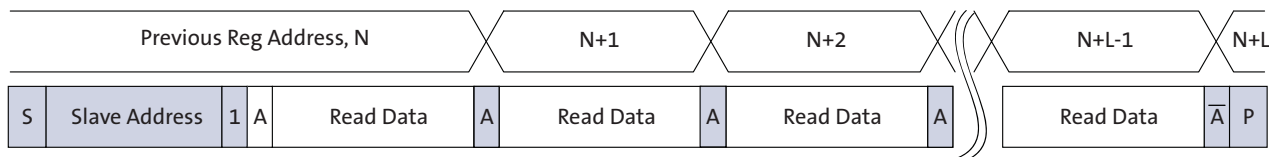
Figure 39: Sequential READ, Start From Random Location



Sequential READ, Start From Current Location

This sequence (Figure 40) starts in the same way as the single READ from current location (Figure 38 on page 53). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte READs until “L” bytes have been read.

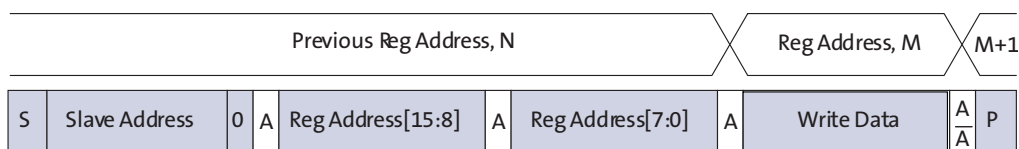
Figure 40: Sequential READ, Start From Current Location



Single WRITE to Random Location

This sequence (Figure 41) 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 41: Single WRITE to Random Location

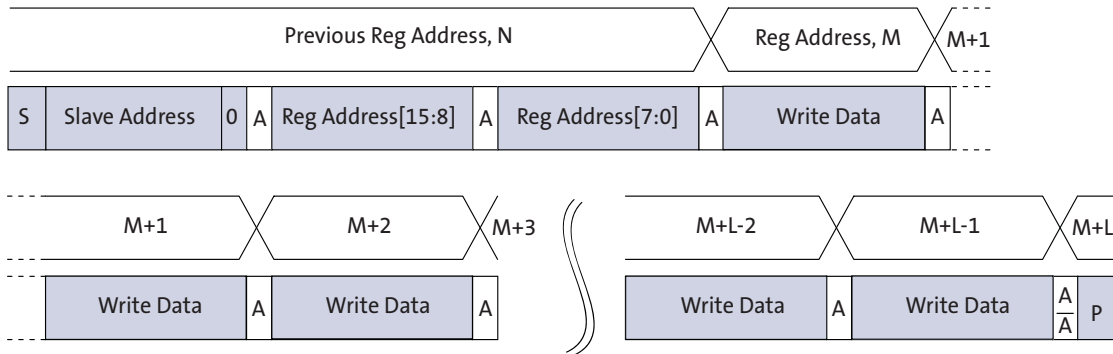




Sequential WRITE, Start at Random Location

This sequence (Figure 42) starts in the same way as the single WRITE to random location (Figure 41 on page 54). 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 42: Sequential WRITE, Start at Random Location





Spectral Characteristics

Figure 43: Bare Die Quantum Efficiency

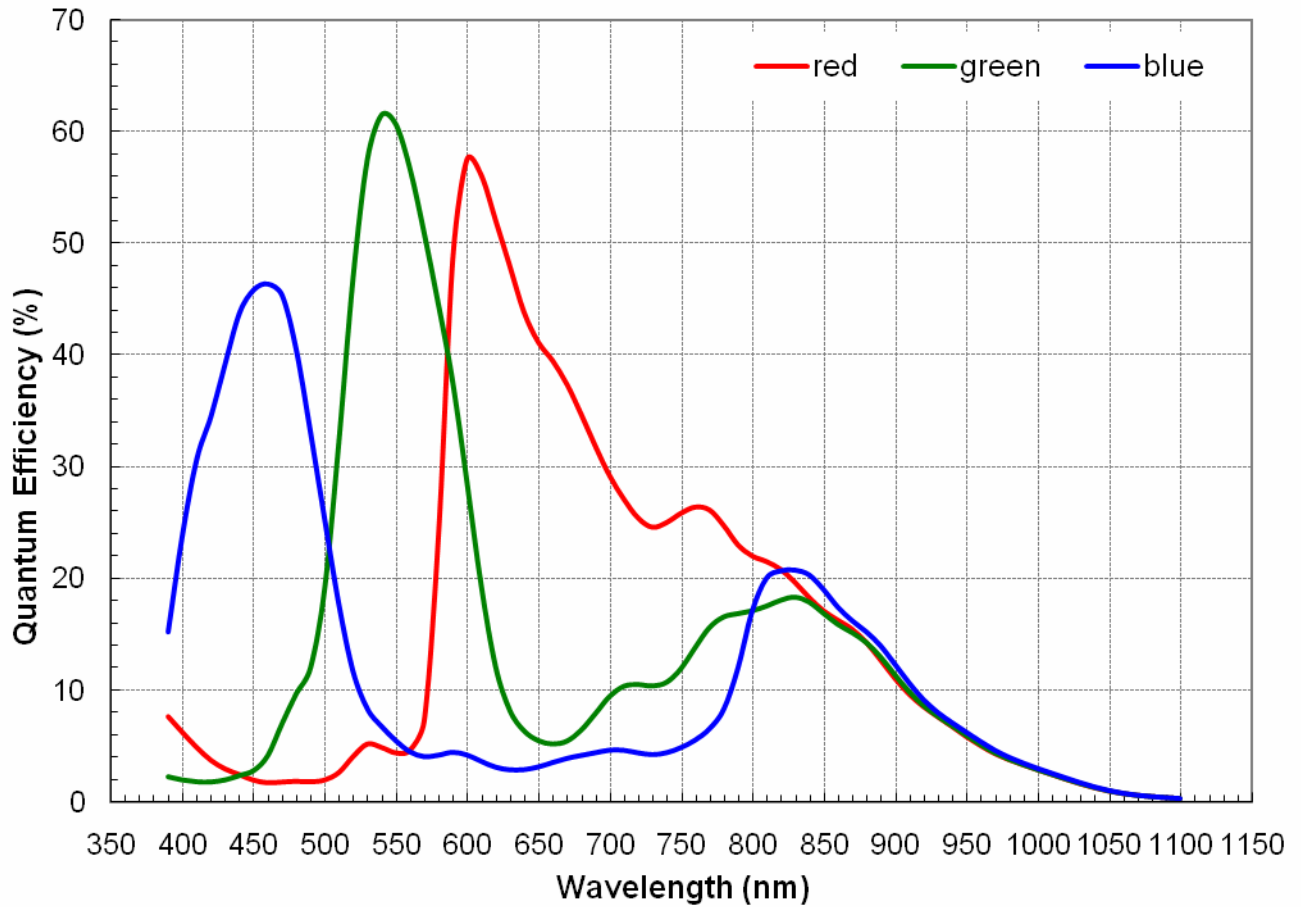




Table 38: Chief Ray Angle (CRA) 21 °

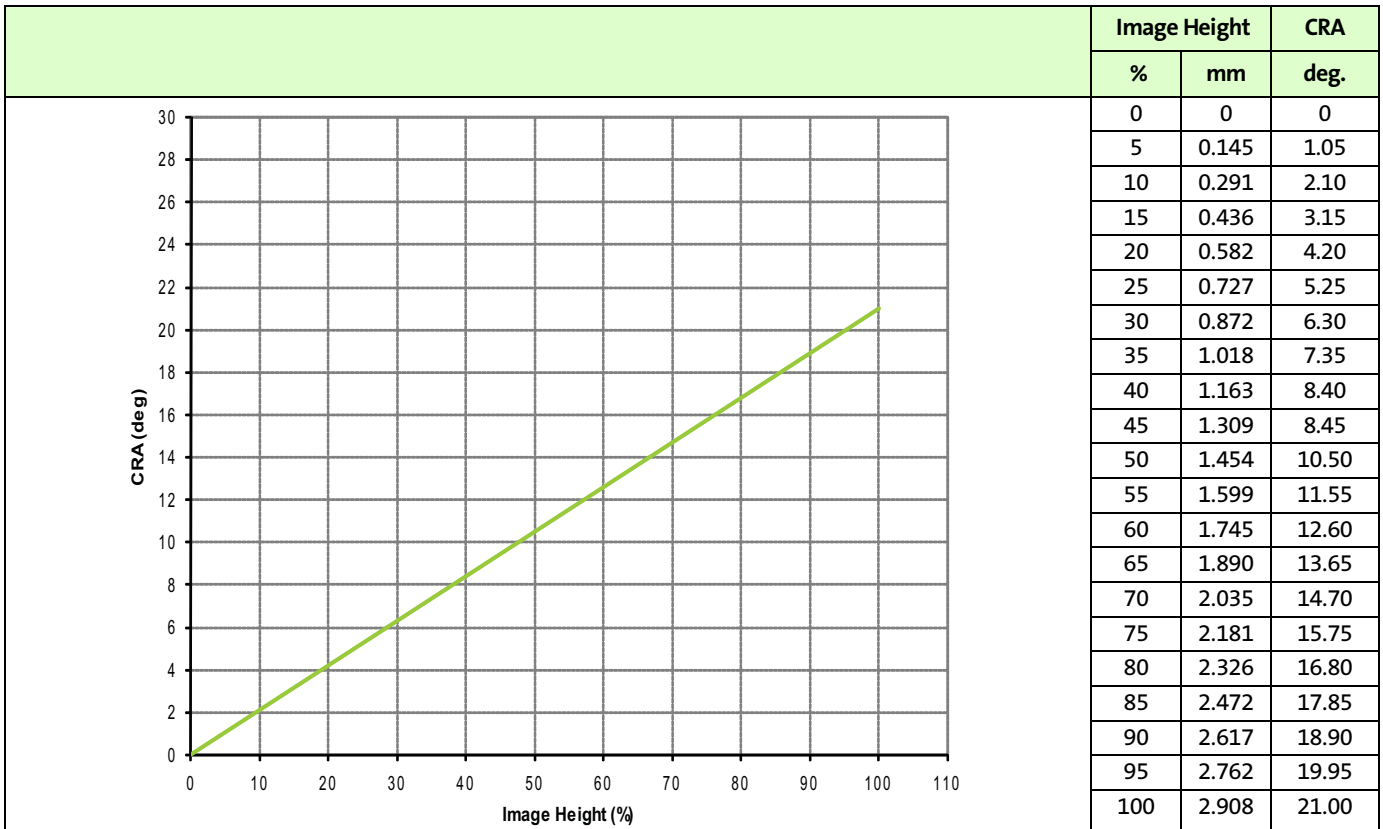
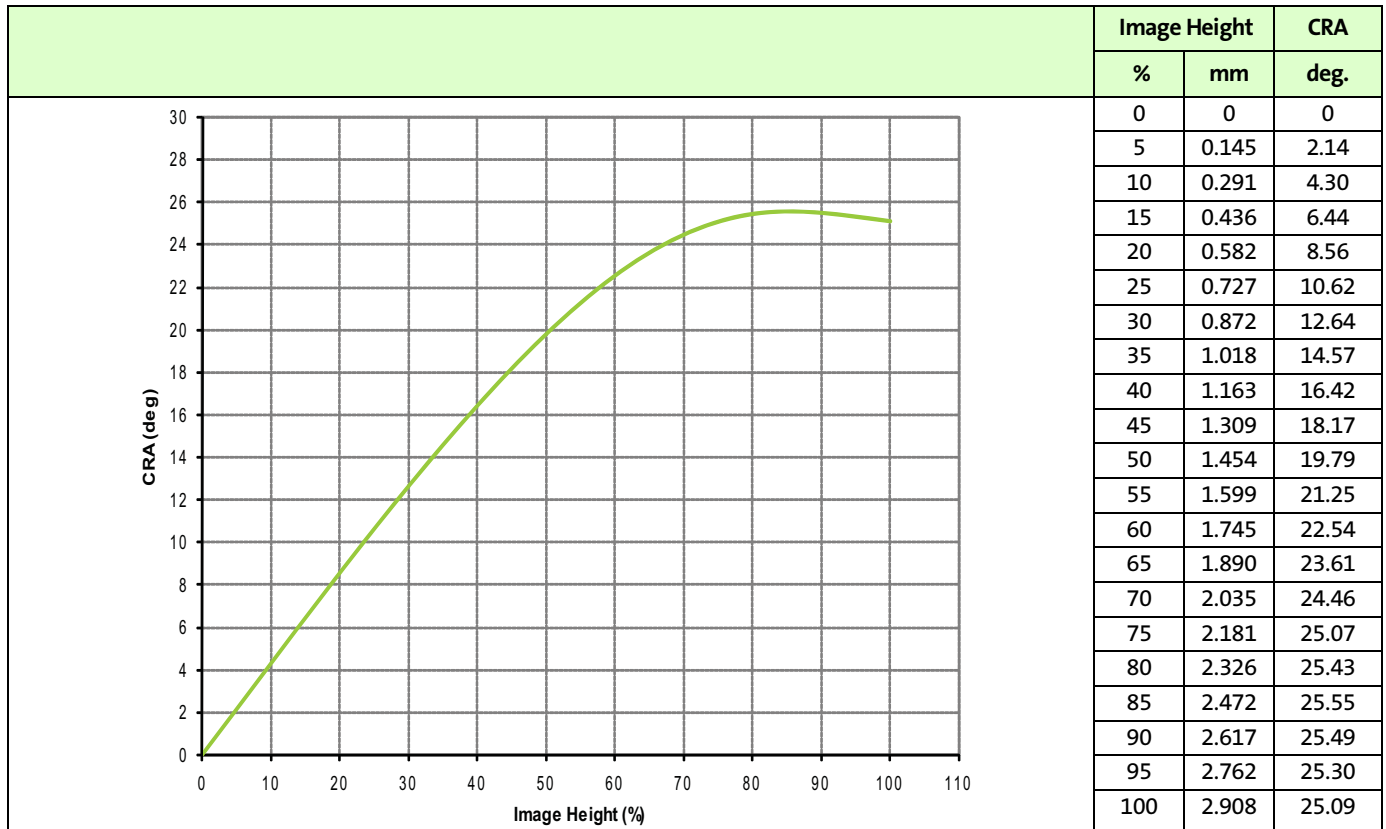




Table 39: Chief Ray Angle (CRA) 25 °



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 40.

Table 40: CRA Value

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

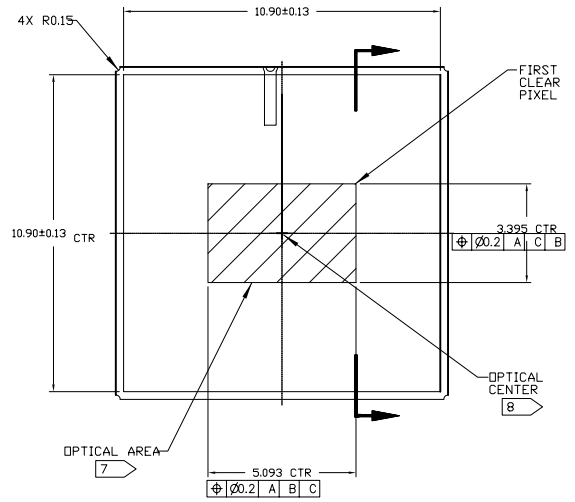
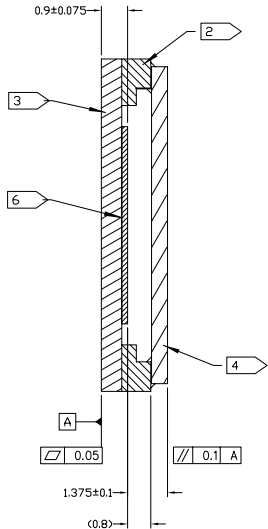
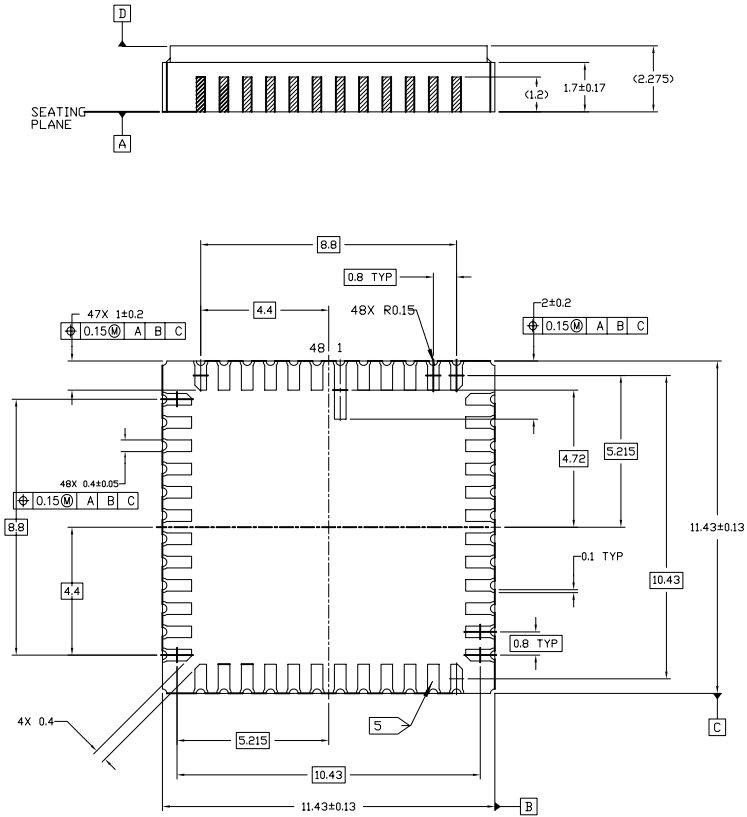
Packages

The AR0330 comes in two packages:

- CLCC Package
- CSP Package

CLCC Package

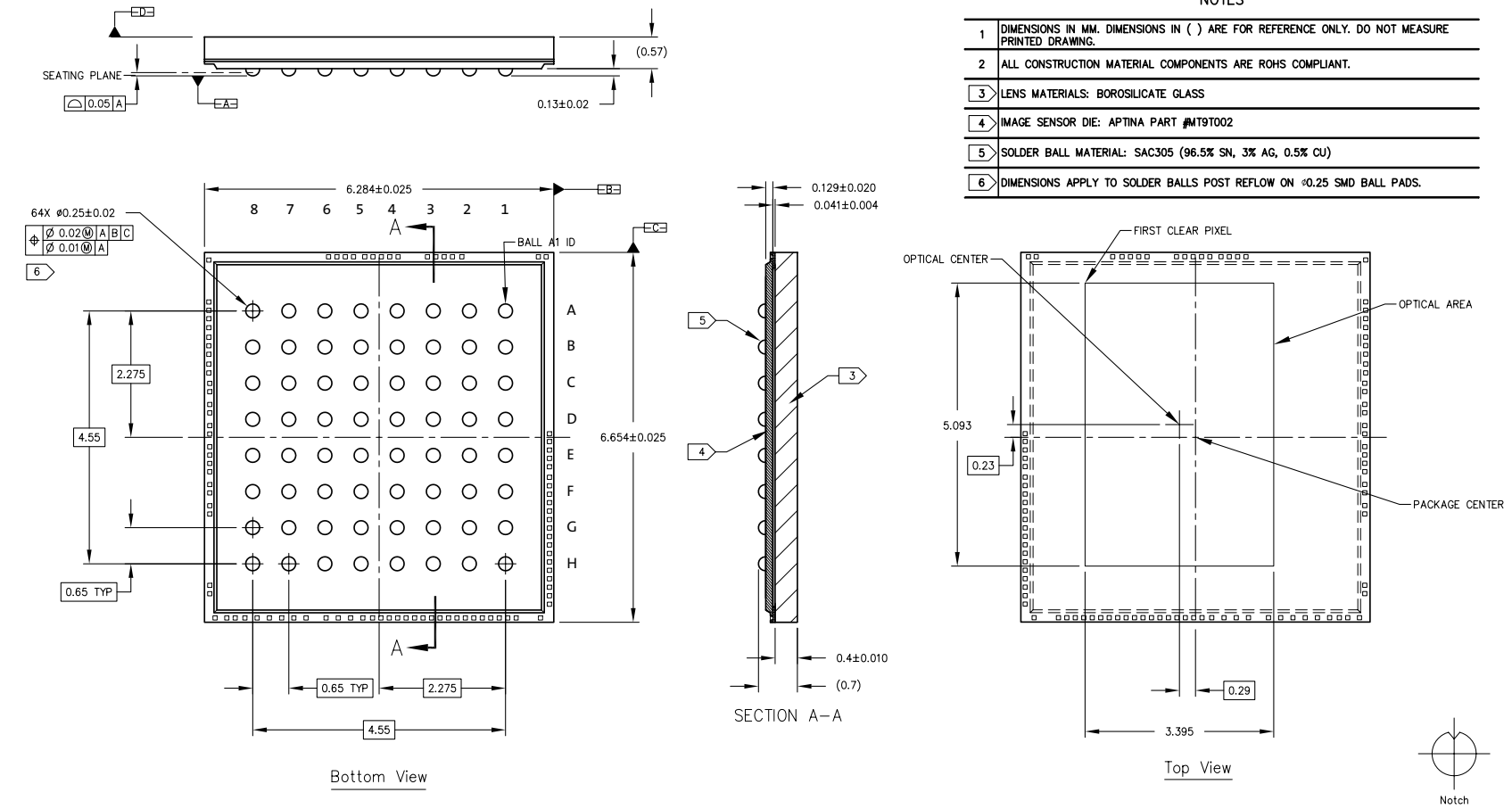
Figure 44: CLCC Package



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 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 MICRON
8	OPTICAL CENTER = PACKAGE CENTER.

CSP Package

Figure 45: CSP Package





Revision History

Rev. F, Advance	1/5/11
	<ul style="list-style-type: none"> • Updated Figure 2: “Typical Configuration: Serial Four-Lane HiSPi Interface,” on page 8 • Updated Table 6, “CSP Package Pin Descriptions,” on page 12 • Updated Figure 5: “CLCC Package Pin Descriptions,” on page 13
Rev. E, Advance	12/17/10
	<ul style="list-style-type: none"> • Changed part number from AC0330 to AR0330 • Applied updated Aptina template • Updated “Power-Up Sequence” on page 14 • Updated Figure 6: “Power Up,” on page 14 • Updated Table 23, “PLL Parameters for the Serial Interface,” on page 25 • Updated column 1 heading in Table 25, “Output Enable Control,” on page 27 • Updated Table 28, “Recommended Sensor Gain Tables,” on page 35 • Updated Figure 25: “Gain Stages in AR0330 Sensor,” on page 34 • Updated Figure 45: “CSP Package,” on page 60
Rev. D, Advance	11/1/10
	<ul style="list-style-type: none"> • Changed part number from MT9T002 to AC0330 • Updated “Features” on page 1 • Updated Table 1, “Key Parameters,” on page 1 • Updated Table 2, “Available Part Numbers,” 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 MIPI Connections,” 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 11 • Added Table 6, “CSP Package Pin Descriptions,” on page 12 • Added Figure 5: “CLCC Package Pin Descriptions,” on page 13 • Added “Electrical Characteristics” on page 17 • Added “Sensor Initialization” on page 14 • Added “Sequencer” on page 23 • Added “Sensor PLL” on page 23 • Added “Pixel Output Interfaces” on page 27 • Added “Sensor Readout” on page 36 • Updated “Subsampling” on page 38 • Added “Sensor Frame Rate” on page 41 • Added “Sensor Frame Rate” on page 41 • Updated “Slave Mode” on page 43 • Added “Frame Readout” on page 46 • Added “Two-Wire Serial Register Interface” on page 51 • Added “Packages” on page 58



Rev. C, Advance	5/3/10
<ul style="list-style-type: none"> • Updated Table 5 on page 11. • Added Figure 2 on page 3. • Updated Figure 1 on page 6. • Updated pins on Figure 2 on page 8 , Figure 3 on page 9 and Figure 4 on page 10. • Changed input clock range to 6-64 MHz • Removed high dynamic range from general description • Removed STANDBY pad from Figure 2 on page 8 and Figure 4 on page 10 • Changed HiSPi to SLVS in Figure 5 on page 11 • Updated slave mode section • Updated Figure 36 on page 45 • Updated notes for Figure 2 on page 8, Figure 3 on page 9, and Figure 4 on page 10. • Updated Table 1 on page 1 	
Rev. B, Advance	4/08/10
<ul style="list-style-type: none"> • Updated key parameters and general description • Updated Table 3 • Removed two-wire serial interface • Added subsampling section • Updated Fig. 1 and Fig. 2 • Added Fig. 3 	
Rev. A	2/10
<ul style="list-style-type: none"> • Initial release 	

Sales: Shenzhen Sunnywale Inc, www.sunnywale.com , awin@sunnywale.com , Wechat: 9308762

10 Eunos Road 8 13-40, Singapore Post Center, Singapore 408600 prodmtg@aptina.com www.aptina.com
Aptina, Aptina Imaging, and the Aptina logo are the property of Aptina Imaging Corporation
All other trademarks are the property of their respective owners.

Advance: This data sheet contains initial descriptions of products still under development.