1/6-Inch 1080P High-Definition (HD) System-On-A-Chip (SOC) Digital Image Sensor

AS0260 Datasheet, Rev. G

For the latest datasheet, please visit www.onsemi.com

Features

- Superior low-light performance
- Ultra-low-power
- 1080p Full HD video at 30 fps
- Internal master clock generated by on-chip phase locked loop (PLL) oscillator
- Electronic rolling shutter (ERS), progressive scan
- Integrated image flow processor (IFP) for single-die camera module
- Automatic image correction and enhancement
- Arbitrary image scaling with anti-aliasing
- Two-wire serial interface providing access to registers and microcontroller memory
- Selectable output data format: YCbCr, JPEG, MJPEG, 565RGB, 555RGB, 444RGB, processed Bayer, BT656, RAW8, RAW8+2-bit, and M420
- Parallel and 1- or 2-lane MIPI data output
- Independently configurable gamma correction
- Adaptive polynomial lens shading correction
- UVC interface support
- Perspective correction
- Multi-camera synchronization

Applications

- Embedded tablet, notebook, and tethered PC cameras
- Game consoles
- Cell phones, mobile devices
- Consumer video communications

General Description

The ON Semiconductor AS0260 is a 1/6-inch 2.0Mp Full HD CMOS digital image sensor with an active-pixel array of 1920H x 1080V. It includes sophisticated camera functions such as auto exposure control, auto white balance, black level control, flicker avoidance, and defect correction. It is designed for low light performance.The AS0260 produces extraordinarily clear, sharp digital pictures, making it the perfect choice for a wide range of applications, including PC and notebook cameras, gaming systems, and mobile phones.

Table 1: Key Parameters

Parameter		Typical Value
Optical format		1/6-inch
Active pixels		1920 x 1080
Pixel size		1.4 µm
Color filter array		RGB Bayer
Shutter type		Electronic rolling shutter (ERS)
Input clock range		6 – 54 MHz
Output pixel clock maximum		96 MHz
Output MIPI data rate maximum		768 Mb/s per lane
Frame Rate	1080p (full res)	30 fps
	720p	60 fps
	VGA	60 fps
	QVGA	120 fps
Responsivity		0.64 V/lux-sec
SNR_{MAX}		33 dB
Pixel dynamic range		65 dB
Supply voltage	Digital	1.7 – 1.95 V
	Analog	2.5 – 3.1 V
	I/O	1.7 – 1.95 V or 2.5 – 3.1 V
	PHY	1.7 – 1.95 V
Power consumption[1]		255 mW
Operating temperature, ambient		–30°C to +70° C
Chief ray angle		28°
Package options		CSP, Bare die

Notes: 1. Power consumption for typical voltages and full resolution output, no MJPEG.

Ordering Information

Table 2: Available Part Numbers

Part Number	Base Description	Variant Description
AS0260CSSC28SUD20	RGB color	Die Sales, 200 µm Thickness
AS0260CSSC28SUKA0-CR	RGB color CSP	Chip Tray without Protective Film
AS0260HQSC28SUD20	RGB color	Die Sales, 200 µm Thickness
AS0260HQSC28SUKA0-CR	RGB color	Chip Tray without Protective Film
AS0260HQSC28SUKAD3-GEVK	RGB color	Demo3 Board
AS0260HQSC28SUKAD-GEVK	RGB color	Demo Kit
AS0260HQSC28SUKAH3-GEVB	RGB color	Demo3 Board
AS0260HQSC28SUKAH-GEVB	RGB color	Demo Board

Table of Contents

Features 1
Applications 1
General Description 1
Ordering Information 2
Functional Description 6
Decoupling Capacitor Recommendations 10
Output Data Format 10
Power-Up Sequence 11
Image Data Output Interface 14
Sensor Control 16
PLL 23
Image Flow Processor 24
Camera Control and Auto Functions 36
JPEG Encoder 42
MJPEG Format 45
UVC Interface 49
Multi-Camera Sync 61
Hardware Functions 65
Spectral Characteristics 71
Chief Ray Angle 72
CSP Package Details 73
Electrical Specifications 75
MIPI Specification Reference 80
Revision History 81

List of Figures

Figure 1: AS0260 Block Diagram . . . 6
Figure 2: Typical Configuration . . . 8
Figure 3: Spatial Illustration of Image Readout . . . 10
Figure 4: Power-Up Sequence . . . 11
Figure 5: Hard Reset Operation . . . 12
Figure 6: Soft Reset Operation . . . 13
Figure 7: Pixel Data Timing Example . . . 14
Figure 8: Row Timing, FV, and LV Signals . . . 14
Figure 9: Sensor Core Block Diagram . . . 16
Figure 10: Pixel Color Pattern Detail (Top Right Corner) . . . 17
Figure 12: Three Pixels in Normal and Column Mirror Readout Mode . . . 18
Figure 13: Six Rows in Normal and Row Mirror Readout Mode . . . 19
Figure 14: Eight Pixels in Normal and Column Skip 2X Readout Mode . . . 19
Figure 15: Pixel Readout (no skipping) . . . 20
Figure 16: Pixel Readout (column skipping) . . . 21
Figure 17: Pixel Readout (row skipping) . . . 21
Figure 18: Pixel Readout (column and row skipping) . . . 22
Figure 19: Pixel Readout (column and row binning) . . . 23
Figure 20: Image Flow Processor . . . 24
Figure 21: Color Bar Test Pattern . . . 26
Figure 22: Gamma Interaction . . . 28
Figure 23: Gamma Reference Variables Against Brightness Metric . . . 30
Figure 24: 0° Hue . . . 32
Figure 25: –22° Hue . . . 33
Figure 26: +22° Hue . . . 33
Figure 27: 5 x 5 Grid . . . 36
Figure 28: JPEG Continuous Data Output . . . 43
Figure 29: JPEG Spoof Mode Timing with Continuous Clock . . . 44
Figure 30: JPEG Status Segment Structure . . . 44
Figure 31: Brightness-Dependent Contrast Control . . . 54
Figure 32: UVC Sharpness Control Flow . . . 57
Figure 33: Multi-Camera Connection . . . 61
Figure 34: Normal Use of SADDR . . . 61
Figure 35: AptiSync2 Hardware Connections . . . 64
Figure 36: Single Read from Random Location . . . 67
Figure 37: Single Read from Current Location . . . 67
Figure 38: Sequential Read, Start from Random Location . . . 68
Figure 39: Sequential Read, Start from Current Location . . . 68
Figure 40: Single Write to Random Location . . . 69
Figure 41: Sequential Write, Start at Random Location . . . 70
Figure 42: Quantum Efficiency vs. Wavelength . . . 71
Figure 43: CSP Mechanical Drawing . . . 74
Figure 44: Parallel Pixel Bus Timing Diagram . . . 77
Figure 45: Two-Wire Serial Bus Timing Parameters . . . 79

List of Tables

Table 1: Key Parameters 1
Table 2: Available Part Numbers 2
Table 3: Pin Descriptions 9
Table 4: Power-Up Signal Timing 11
Table 5: Status of Output Signals During Hard Reset 11
Table 6: Hard Reset 12
Table 7: Soft Reset Signal Timing 13
Table 8: Variables Required for Gamma Knee Point Calculation 29
Table 9: Gamma Curve Selection 30
Table 10: Fade-to-Black Selection 31
Table 11: Hue Control 32
Table 12: Variables Controlling VPC 34
Table 13: YCbCr Output Data Ordering 41
Table 14: RGB Ordering in Default Mode 41
Table 15: 2-Byte Bayer Format 41
Table 16: UVC_Result_Status Codes 50
Table 17: UVC Adjustment of Lowlight Sharpness 57
Table 18: UVC Sharpness vs. Adaptive Sharpness Range 57
Table 19: CHAIN_CONTROL Register 62
Table 20: AUTOSYNC_MODE Register 64
Table 21: Chief Ray Angle Characteristics 72
Table 22: Package Dimension 73
Table 23: Ball Matrix 74
Table 24: Absolute Maximum Ratings 75
Table 25: Operating Conditions 75
Table 26: DC Electrical Characteristics 75
Table 27: Operating Current Consumption (Parallel) 76
Table 28: AC Electrical Characteristics 77
Table 29: Two-Wire Serial Interface Timing Data 79
Table 30: MIPI High-Speed Transmitter DC Characteristics 80
Table 31: MIPI High-Speed Transmitter AC Characteristics 80
Table 32: MIPI Low-Power Transmitter DC Characteristics 80
Table 33: MIPI Low-Power Transmitter AC Characteristics 80

Functional Description

The ON Semiconductor AS0260 is a 1/6-inch 2.0 Mp CMOS digital image sensor with an integrated advanced camera system. This camera system features a microcontroller (MCU), a sophisticated image flow processor (IFP), MIPI and parallel output ports (only one output port can be used). The microcontroller manages all functions of the camera system and sets key operation parameters for the sensor core to optimize the quality of raw image data entering the IFP. The sensor core consists of an active pixel array of 1920 x 1080 pixels with programmable timing and control circuitry. It also includes an analog signal chain with automatic offset correction, programmable gain, and a 10-bit analog-to-digital converter (ADC).

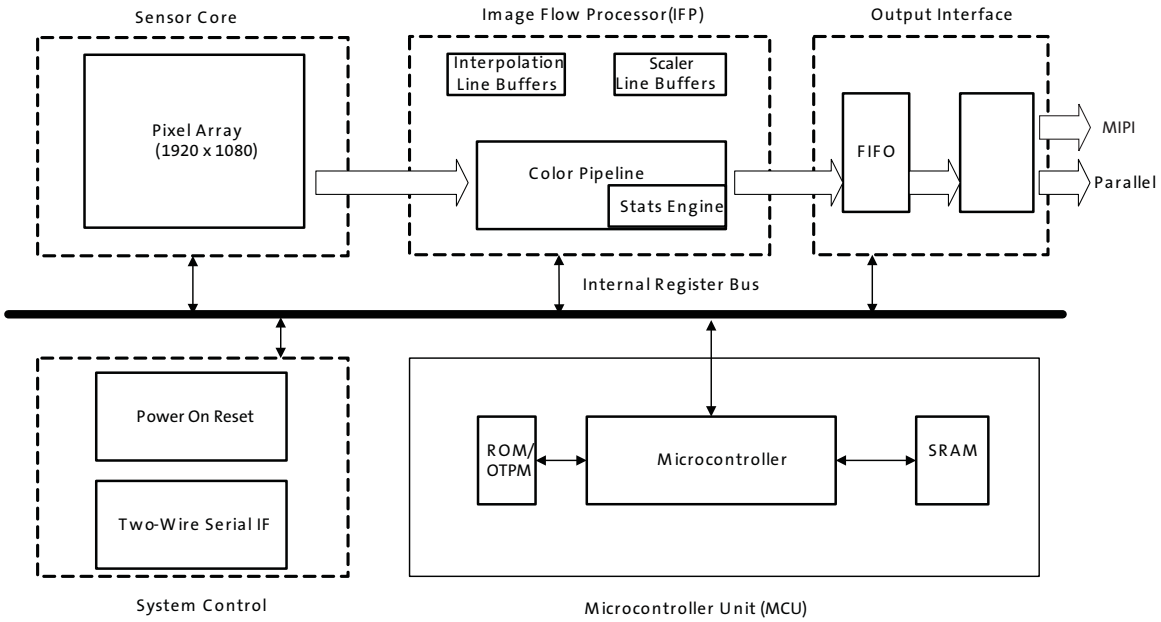
The entire system-on-a-chip (SOC) has superior low-light performance that is particularly suitable for PC camera applications. The AS0260 features ON Semiconductor's breakthrough low-noise CMOS imaging technology that achieves near-CCD image quality (based on signal-to-noise ratio and low-light sensitivity) while maintaining the inherent size, cost, and integration advantages of CMOS.

The ON Semiconductor AS0260 can be operated in its default mode or programmed for frame size, exposure, gain, and other parameters. The default mode output is a 1080P image size at 30 frames per second (fps). It outputs JPEG compressed 8-bit data, using the parallel output port.

Architecture Overview

The AS0260 combines a 2.0 Mp sensor core with an IFP to form a stand-alone solution for both image acquisition and processing. Both the sensor core and the IFP have internal registers that can be controlled by the user. In normal operation, an integrated microcontroller autonomously controls most aspects of operation. The processed image data is transmitted to the host system either through the parallel or MIPI interface. Figure 1 shows the major functional blocks of the AS0260.

Figure 1: AS0260 Block Diagram



Sensor Core

The AS0260 has a color image sensor with a Bayer color filter arrangement and a 2.0Mp active-pixel array with electronic rolling shutter (ERS). The sensor core readout is 10 bits and can be flipped and/or mirrored.

Image Flow Processor (IFP)

The advanced IFP features and flexible programmability of the AS0260 can enhance and optimize the image sensor performance. Built-in optimization algorithms enable the AS0260 to operate with factory settings as a fully automatic and highly adaptable system-on-a-chip (SOC) for most camera systems.

These algorithms include black level conditioning, shading correction, defect correction, color interpolation, edge detection, color correction, aperture correction, hue rotation, perspective correction, and image formatting with cropping and scaling.

The IFP includes special modes to support presence detection and ambient light measurement. These modes can be used to assist the power management of a notebook PC.

Microcontroller Unit (MCU)

The MCU communicates with all functional blocks by way of an internal ON Semiconductor proprietary bus interface. The MCU firmware configures all the registers in the sensor core and IFP.

System Control

The AS0260 has a phase-locked loop (PLL) oscillator that can generate the internal sensor clock from the common system clock. The PLL adjusts the incoming clock frequency up, allowing the AS0260 to run at almost any desired resolution and frame rate within the sensor's capabilities.

The AS0260 provides power-conserving features including a soft standby mode. A two-wire serial interface bus enables read and write access to the AS0260's internal registers and variables. The internal registers control the sensor core, the color pipeline flow, and the output interface. Variables are located in the microcontroller's RAM memory and are used to configure and control the auto-algorithms and camera control functions.

Output Interface

The output interface block can select either raw data or processed data. Image data is provided to the host system either by an 8-bit parallel port or by a dual-lane serial MIPI port. The parallel output port provides 8-bit RGB data or extended 10-bit Bayer data.

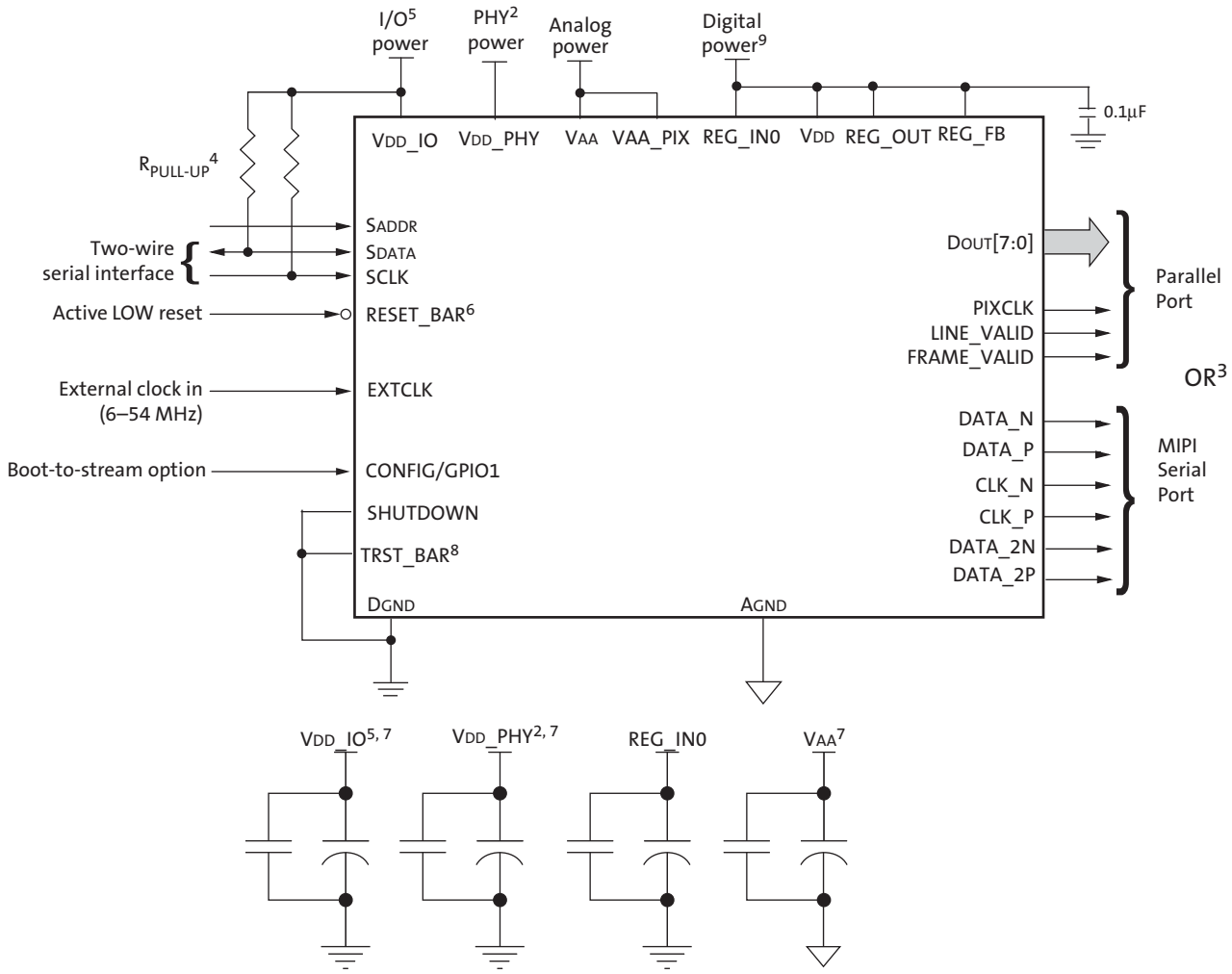
The AS0260 also includes programmable I/O slew rate to minimize EMI.

System Interfaces

Figure 2 on page 3 shows typical AS0260 device connections. For low-noise operation, the AS0260 requires separate power supplies for analog and digital sections of the die. Both power supply rails must be decoupled from ground using capacitors as close as possible to the die. The use of inductance filters is not recommended on the power supplies or output signals.

The AS0260 provides dedicated inputs for digital core, PHY, and I/O power domains that can be at different voltages. The PLL and analog circuitry require clean power sources. Table 1 on page 4 provides the signal descriptions for the AS0260.

Figure 2: Typical Configuration



Notes:
1. This typical configuration shows only one scenario out of multiple possible variations for this sensor.
2. If a MIPI Interface is not required, the MIPI serial port must be left floating. The VDD_PHY power signal must always be connected to the 1.8V supply.
3. Only one of the output modes (serial or parallel) can be used at any time.
4. ON Semiconductor recommends a 1.5kΩ resistor value for the two-wire serial interface RPULL-UP; however, greater values may be used for slower transmission speed.
5. All inputs must be configured with VDD_IO.
6. RESET_BAR has an internal pull-up resistor and can be left floating.
7. ON Semiconductor recommends that 0.1µF and 1µF decoupling capacitors for each power supply are mounted as close as possible to the pad. Actual values and numbers may vary depending on layout and design considerations.
8. TRST_BAR connects to GND for normal operation.
9. Future versions of AS0260 will not require VDD, REG_OUT, and REG_FB to be connected to REG_IN0.

Table 1: Pin Descriptions

Name	Type	Description	Note
EXTCLK	Input	Input clock signal.	
RESET_BAR	Input/PU	Master reset signal, active LOW. This signal has an internal pull up.	
SCLK	Input	Two-wire serial interface clock.	
SDATA	I/O	Two-wire serial interface data.	
SADDR	Input	Selects device address for the two-wire serial interface.	
FRAME_VALID (FV)	Output	Identifies rows in the active image.	
LINE_VALID (LV)	Output	Identifies pixels in the active line.	
PIXCLK	Output	Pixel clock.	
DOUT[7:0]	Output	DOUT[7:0] for 8-bit image data output or DOUT[9:2] for 10-bit image data output.	
CLK_N	Output	Differential MIPI clock (sub-LVDS, negative).	2
CLK_P	Output	Differential MIPI clock (sub-LVDS, positive).	2
DATA_N	Output	Differential MIPI data (sub-LVDS, negative).	2
DATA_P	Output	Differential MIPI data (sub-LVDS, positive).	2
DATA_2N	Output	Differential MIPI data (sub-LVDS, negative).	2
DATA_2P	Output	Differential MIPI data (sub-LVDS, positive).	2
CONFIG/GPIO1	Input/Output	If on power-up CONFIG =1 then the part shall go into streaming else the system will go to suspend state waiting for host to update. This pin can also be re-programmed to support multiple functions.	
CHAIN/GPIO0	Input/Output	To synchronize a number of sensors together. This pin can also be re-programmed to support multiple functions.	
GPIO2	Input/Output	General purpose input/output.	
SHUTDOWN	Input	Low power shutdown control, active HIGH.	
TRST_BAR	Input	Must be tied to GND in normal operation.	
VDD	Supply	Digital power. Must connect to REG_OUT and REG_FB.	4
DGND	Supply	Digital ground.	1
VDD_IO	Supply	I/O power supply.	
VAA	Supply	Analog power.	
VAA_PIX	Supply	Analog pixel power.	
AGND	Supply	Analog ground.	1
VPP	Supply	OTPM programming.	
REG_IN0	Supply	Digital power	
REG_OUT	Supply	Digital power. Must connect to VDD and REG_FB.	4
REG_FB	Supply	Digital power. Must connect to VDD and REG_OUT.	4
VDD_PHY	Supply	I/O power supply for the MIPI interface.	3

1. AGND and DGND are not connected internally.
2. To be left floating if not using feature.
3. Must always be connected even when not using MIPI.
4. The VDD, REG_OUT, and REG_FB pins must be connected together and have a 0.1µF decoupling capacitor attached.

Decoupling Capacitor Recommendations

It is important to provide clean, well regulated power to each power supply. The ON Semiconductor recommendation for capacitor placement and values are based on our internal demo camera design and verified in hardware. Note: Because hardware design is influenced by many factors, such as layout, operating conditions, and component selection, the customer is ultimately responsible to ensure that clean power is provided for their own designs.

In order of preference, ON Semiconductor recommends:

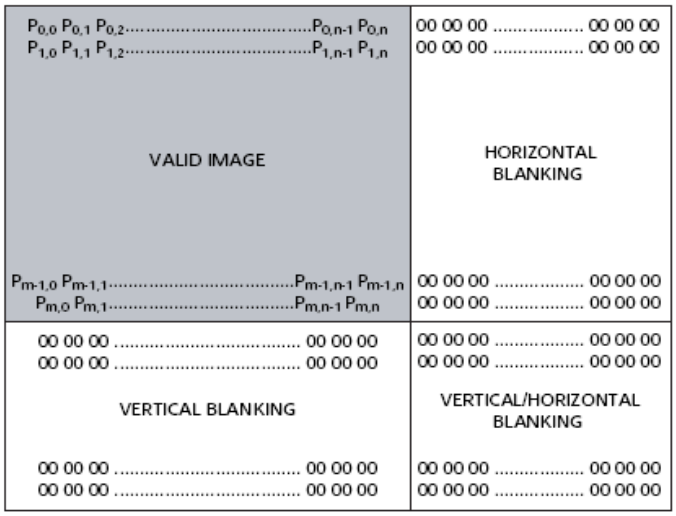
1. Mount 0.1µF and 1µF decoupling capacitors for each power supply as close as possible to the pad and place a 10 µF capacitor nearby off-module.
2. If module limitations allow for only six decoupling capacitors for a three-regulator design use a 0.1µF and 1µF capacitor for each of the three regulated supplies. ON Semiconductor also recommends placing a 10µF capacitor for each supply off-module, but close to each supply.
3. If module limitations allow for only three decoupling capacitors, use a 1µF capacitor (preferred) or a 0.1µF capacitor for each of the three regulated supplies. ON Semiconductor recommends placing a 10µF capacitor for each supply off-module but close to each supply.
4. Give priority to the VAA supply for additional decoupling capacitors.
5. Inductive filtering components are not recommended.
6. Follow best practices when performing physical layout. Refer to technical note TN-09-131.

Output Data Format

The AS0260 image data is read out in a progressive scan. Valid image data is surrounded by horizontal blanking and vertical blanking, as shown in Figure 3.

LINE_VALID is HIGH in the shaded region of the figure.

Figure 3: Spatial Illustration of Image Readout



Power-Up Sequence

Powering up the sensor requires voltages to be applied in a particular order, as seen in Figure 4. The timing requirements are shown in Table 2. The sensor includes a power-on reset feature that initiates a reset upon power up of the sensor.

Figure 4: Power-Up Sequence

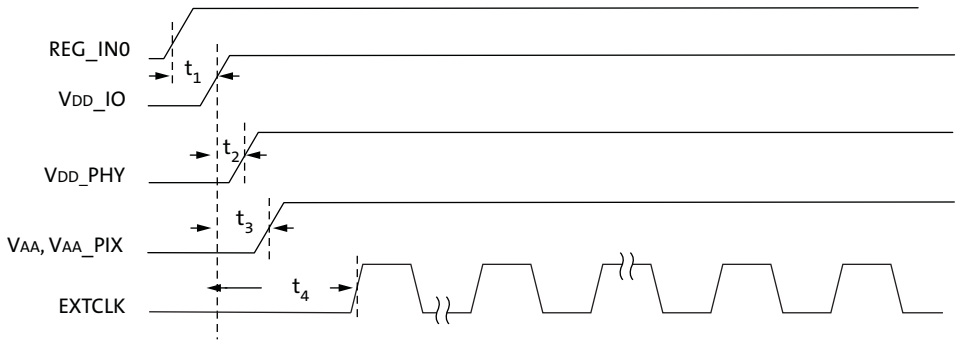


Table 2: Power-Up Signal Timing

Symbol	Parameter	Min	Max	Unit
t_1	Delay from REG_IN0 to VDD_IO	50	200	ms
t_2	Delay from VDD_IO and VDD_PHY	0	50	ms
t_3	Delay from VDD_IO to VAA, VAA_PIX, and VDD_PLL	0	50	ms
t_4	EXTCLK activation	t_2 + 0	–	ms

Power-On Reset

The AS0260 includes a power-on reset feature that initiates a reset upon power-up.

Three types of reset are available:

- A hard reset is issued by toggling the RESET_BAR signal
- A soft reset is issued by writing commands through the two-wire serial interface
- An internal power-on reset

The output states during hard reset are shown in Table 3.

Table 3: Status of Output Signals During Hard Reset

Signal	Reset
DOUT[7:0]	High-Z
PIXCLK	High-Z
LV	High-Z
FV	High-Z
DATA_N	0
DATA_P	0
DATA_2N	0
DATA_2P	0
CLK_N	0

Table 3: Status of Output Signals During Hard Reset

Signal	Reset
CLK_P	0

A soft reset sequence to the sensor has a similar effect as the hard reset and can be activated by writing to a register through the two-wire serial interface. On-chip power-on-reset circuitry can generate an internal reset signal in case an external reset is not provided. The RESET_BAR signal has an internal pull-up resistor and can be left floating.

Hard Reset

The AS0260 enters the reset state when the external RESET_BAR signal is asserted LOW, as shown in Figure 5. Parallel data output signals will be in High-Z state.

Figure 5: Hard Reset Operation

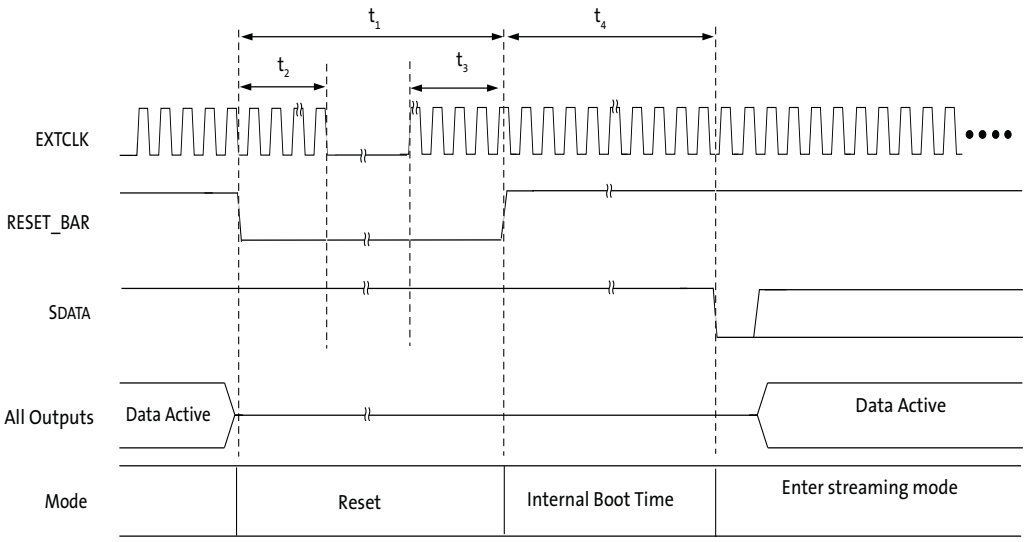


Table 4: Hard Reset

Symbol	Definition	Min	Typ	Max	Unit
t_1	RESET_BAR pulse width	50	–	–	EXTCLK cycles
t_2	Active EXTCLK required after RESET_BAR asserted	10	–	–	
t_3	Active EXTCLK required before RESET_BAR de-asserted	10	–	–	
t_4	Maximum internal boot time[1]	–	–	35	ms

Notes: 1. This delay is dependent on EXTCLK frequency.
2. Assumes that CONFIG/GPIO1 = 1.

Soft Reset

The host processor can reset the AS0260 using the two-wire serial interface by writing to SYSCTL 0x001A. SYSCTL 0x001A[0] is used to reset the AS0260 which is similar to external RESET_BAR signal.

1. Set SYSCTL 0x001A[0] to 0x1 to initiate internal reset cycle.
2. Reset SYSCTL 0x001A[0] to 0x0 for normal operation.
3. Delay up to 35 ms, depending on EXTCLK frequency.

Figure 6: Soft Reset Operation

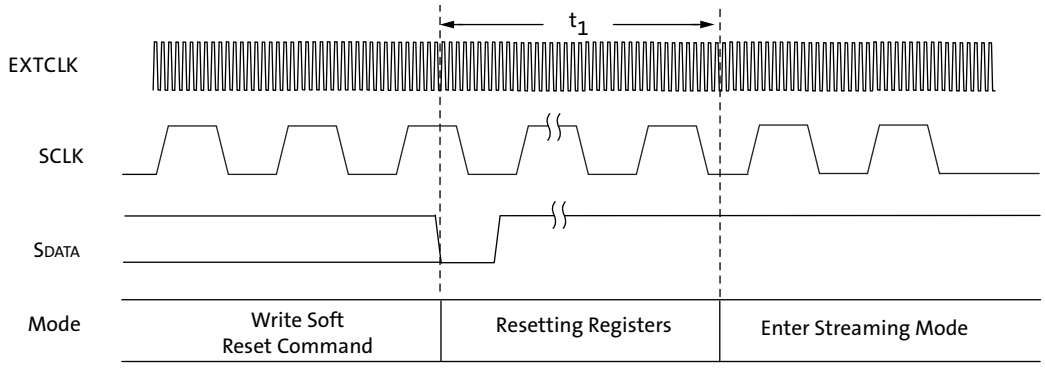


Table 5: Soft Reset Signal Timing

Symbol	Parameter	Min	Typ	Max	Unit
t_1	Maximum soft reset time[1]	–	–	35	ms

Notes:
1. This delay is dependent on EXTCLK frequency.
2. Assumes that CONFIG/GPIO1 = 1.

Shutdown Mode

The shutdown mode is entered when the SHUTDOWN pin is asserted. All power to the AS0260 is disabled and no state, register, or patch information is retained. De-assertion of the SHUTDOWN pin will cause a full POR.

Image Data Output Interface

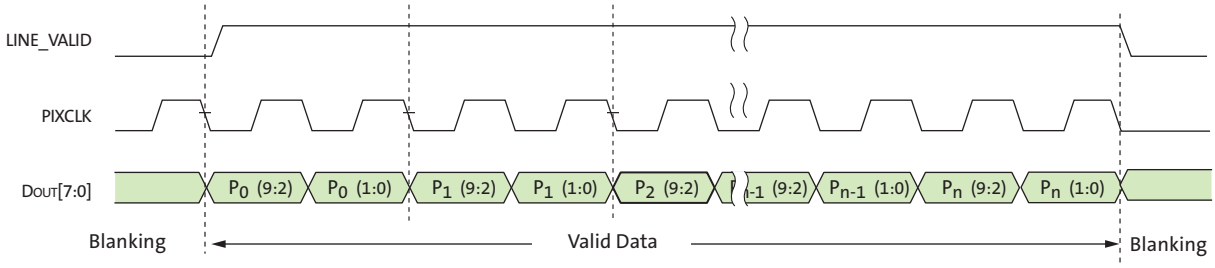
The user can select either the 8-bit parallel or serial MIPI output to transmit the sensor image data to the host system. Only one of the output modes can be used at any time. The AS0260 has an output FIFO to retain a constant pixel output clock.

Parallel Port

The AS0260 image data is read out in a progressive scan mode. Valid image data is surrounded by horizontal blanking and vertical blanking. The amount of horizontal blanking and vertical blanking are programmable.

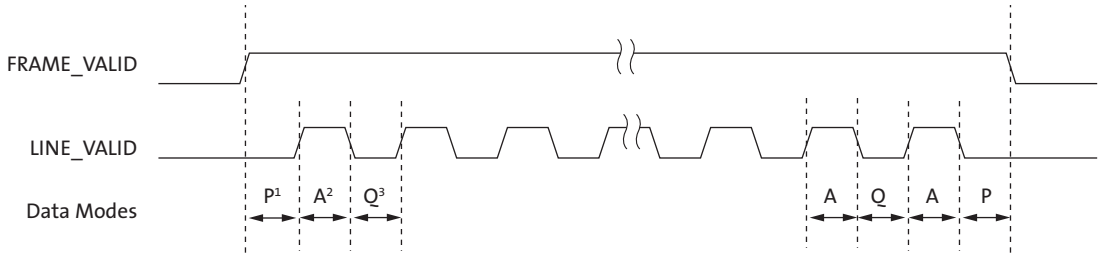
AS0260 output data is synchronized with the PIXCLK output. When LV is HIGH, one pixel value is output on the 8-bit DOUT port every TWO PIXCLK periods as shown in Figure 7. PIXCLK is continuously running, even during the blanking period. PIXCLK phase can be varied by 50 percent, controlled using a register.

Figure 7: Pixel Data Timing Example



Note: Shown is 10-bit Bayer data in 8 + 2 mode.

Figure 8: Row Timing, FV, and LV Signals



Notes:
1. P: Frame start and end blanking time.
2. A: Active data time.
3. Q: Horizontal blanking time.

MIPI Port

The MIPI output implements a serial differential sub-LVDS transmitter capable of up to 1536 Mbps (768 Mbps/lane). It supports multiple formats, error checking, and custom short packets.

When the sensor is in the hardware standby system state or in the software standby system state, the MIPI signals (CLK_P, CLK_N, DATA_P, DATA_N, DATA_2P, DATA_2N) indicate ultra low power state (ULPS) corresponding to (nominal) 0V levels being driven on CLK_P, CLK_N, DATA_P, DATA_N, DATA_2P, and DATA_2N. This is equivalent to signaling code LP-00.

When the sensor enters the streaming system state, the interface goes through the following transitions:

1. After the PLL has locked and the bias generator for the MIPI drivers has stabilized, the MIPI interface transitions from the ULPS state to the ULPS-exit state (signaling code LP–10).
2. After a delay (TWAKEUP), the MIPI interface transitions from the ULPS-exit state to the TX-stop state (signaling code LP–11).
3. After a short period of time (the programmed integration time plus a fixed overhead), frames of pixel data start to be transmitted on the MIPI interface. Each frame of pixel data is transmitted as a number of high-speed packets. The transition from the TX-stop state to the high-speed signaling states occurs in accordance with the MIPI specifications. Between high-speed packets and between frames, the MIPI interface idles in the TX-stop state. The transition from the high-speed signaling states and the TX-stop state takes place in accordance with the MIPI specifications.
4. If the sensor is reset, any frame in progress is aborted immediately and the MIPI signals switch to indicate the ULPS.
5. If the sensor is taken out of the streaming system state and reset_register[4] = 1 (standby end-of-frame), any frame in progress is completed and the MIPI signals switch to indicate the ULPS.

If the sensor is taken out of the streaming system state and reset_register[4] = 0 (standby end-of-frame), any frame in progress is aborted as follows:

1. Any long packet in transmission is completed.
2. The end of frame short packet is transmitted.

After the frame has been aborted, the MIPI signals switch to indicate the ULPS.

Sensor Control

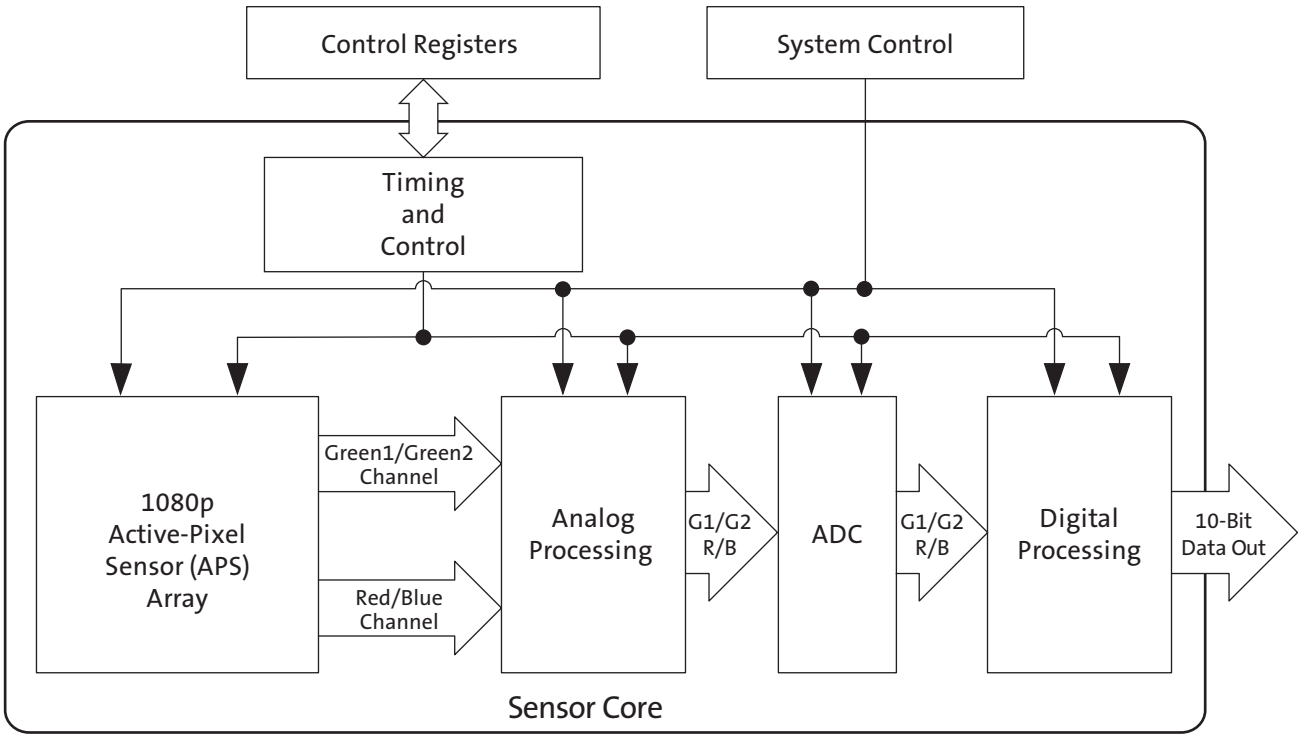
The sensor core of the AS0260 is a progressive-scan sensor that generates a stream of pixel data at a constant frame rate. Figure 9 shows a block diagram of the sensor core. The timing and control circuitry sequences through the rows of the array, resetting and then reading each row in turn. In the time 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 selected, the data from each column is sequenced through an analog signal chain, including offset correction, gain adjustment, and ADC. The final stage of sensor core converts the output of the ADC into 10-bit data for each pixel in the array.

The pixel array contains optically active and light-shielded (dark) pixels. The dark pixels are used to provide data for the offset-correction algorithms (black level control).

The sensor core contains a set of control and status registers that can be used to control many aspects of the sensor behavior including the frame size, exposure, and gain setting. These registers are controlled by the MCU firmware and are also accessible by the host processor through the two-wire serial interface.

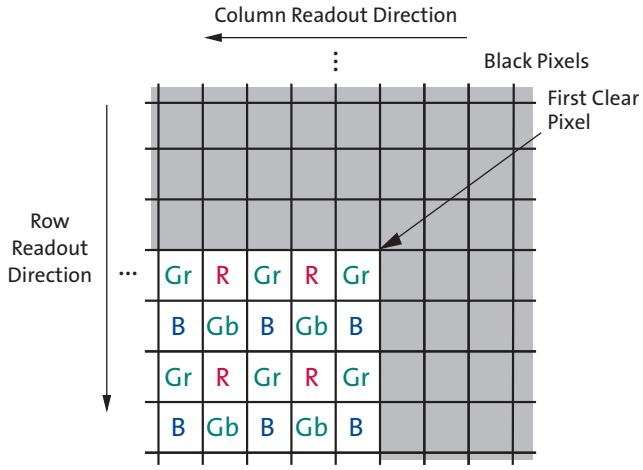
The output from the sensor core is a Bayer pattern; alternate rows are a sequence of either green and red pixels or blue and green pixels. The analog signal chain provides per-color control of the pixel data.

Figure 9: Sensor Core Block Diagram



The sensor core uses a Bayer color pattern, as shown in Figure 10. The even-numbered rows contain green and red pixels; odd-numbered rows contain blue and green pixels. Even-numbered columns contain green and blue pixels; odd-numbered columns contain red and green pixels.

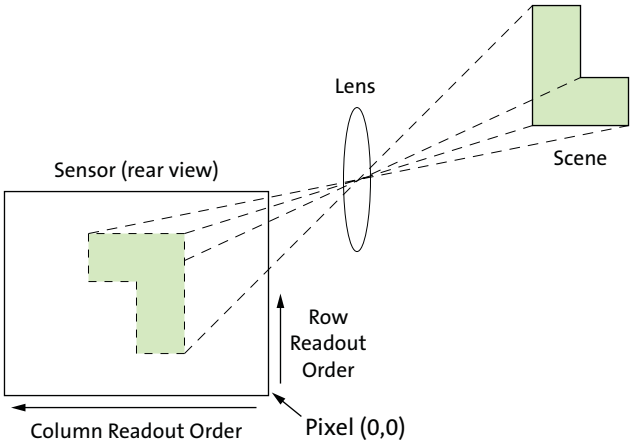
Figure 10: Pixel Color Pattern Detail (Top Right Corner)



The AS0260 sensor core pixel array is shown with pixel (0,0) in the top right corner, which reflects the actual layout of the array on the die. Figure 11 on page 13 shows the image shown in the sensor during normal operation.

When the image is read out of the sensor, it is read one row at a time, with the rows and columns sequenced.

Figure 11: Imaging a Scene



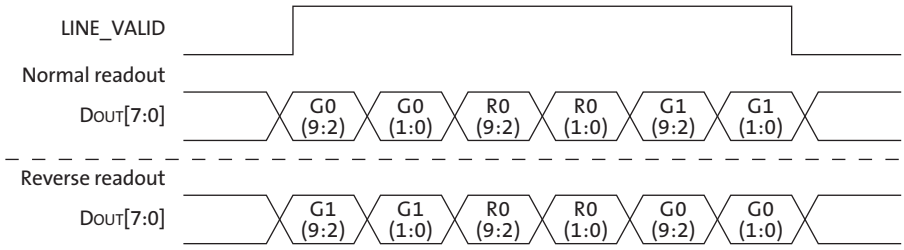
The sensor core supports different readout options to modify the image before it is sent to the IFP. The readout can be limited to a specific window size of the original pixel array.

By changing the readout directions, the image can be flipped in the vertical direction and/or mirrored in the horizontal direction.

The image output size is set by programming row and column start and end address variables.

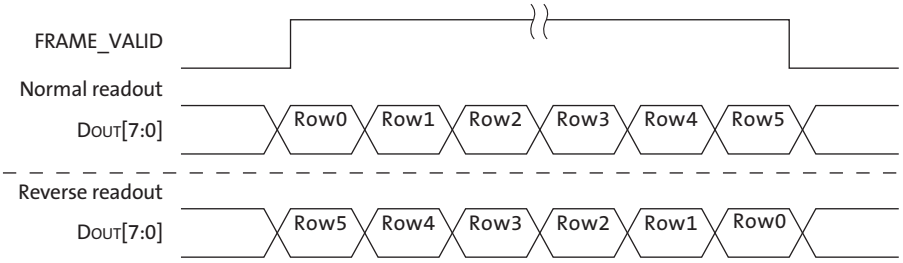
When the sensor is configured to mirror the image horizontally, the order of pixel readout within a row is reversed, so that readout starts from the last column address and ends at the first column address. Figure 12 shows a sequence of 3 pixels being read out with normal readout and reverse readout (Bayer8 + 2 mode shown). This change in sensor core output is corrected by the IFP.

Figure 12: Three Pixels in Normal and Column Mirror Readout Mode



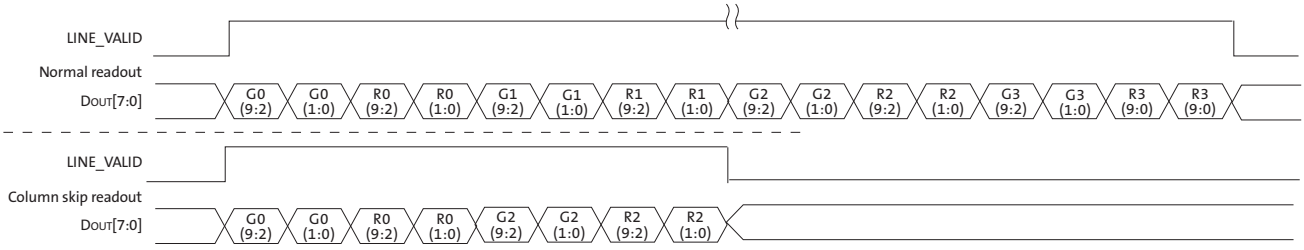
When the sensor is configured to flip the image vertically, the order in which pixel rows are read out is reversed, so that row readout starts from the last row address and ends at the first row address. Figure 13 on page 14 shows a sequence of 6 rows being read out with normal readout and reverse readout. This change in sensor core output is corrected by the IFP.

Figure 13: Six Rows in Normal and Row Mirror Readout Mode



The AS0260 sensor core supports subsampling with skipping to increase the frame rate. The proper image output size and cropped size must be programmed before enabling subsampling mode. Figure 14 shows Bayer 8 + 2 readout with 2X skipping.

Figure 14: Eight Pixels in Normal and Column Skip 2X Readout Mode



Pixel Readouts

The following diagrams show a sequence of data being read out with no skipping. The effect of the different subsampling on the pixel array readout is shown in Figure 15 through Figure 19 on page 18.

Figure 15: Pixel Readout (no skipping)

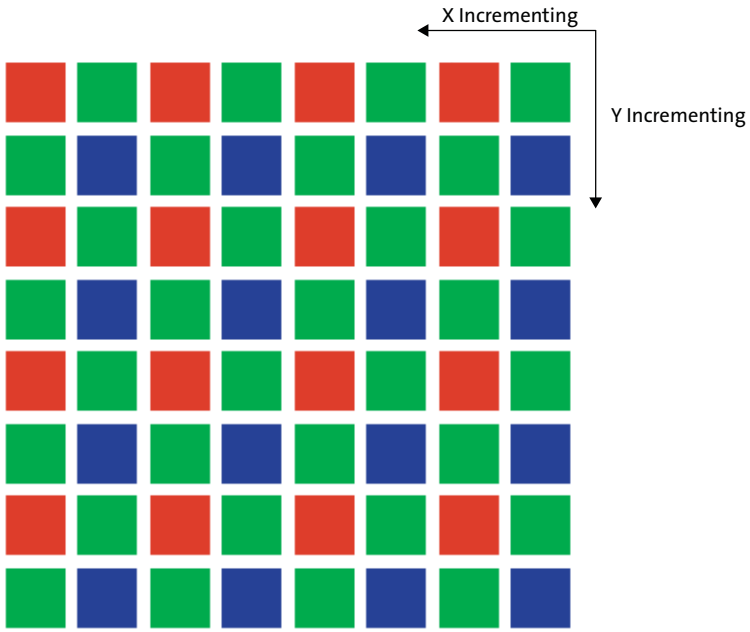


Figure 16: Pixel Readout (column skipping)

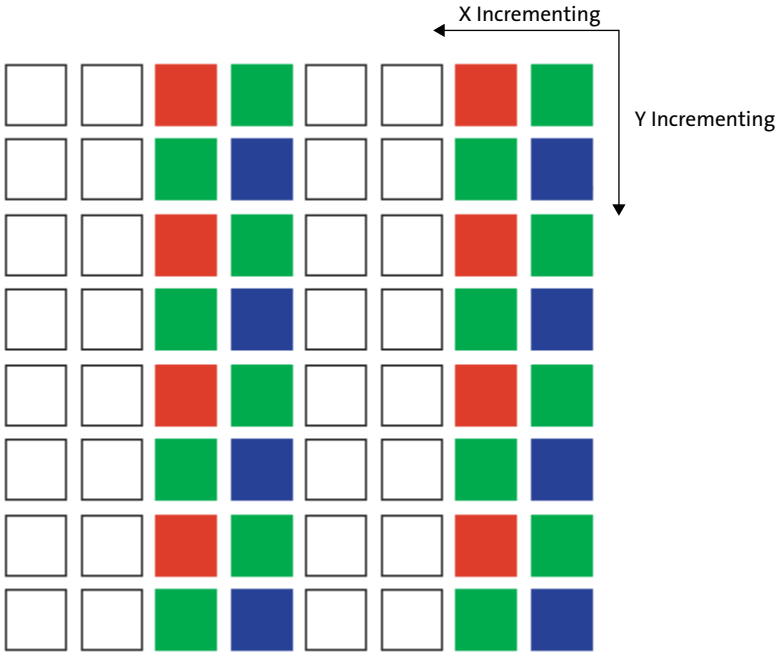


Figure 17: Pixel Readout (row skipping)

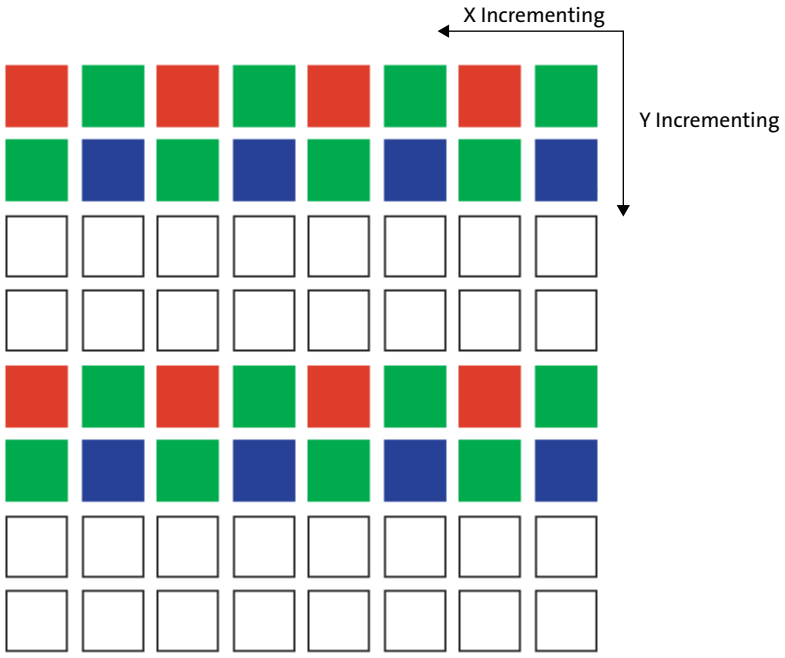
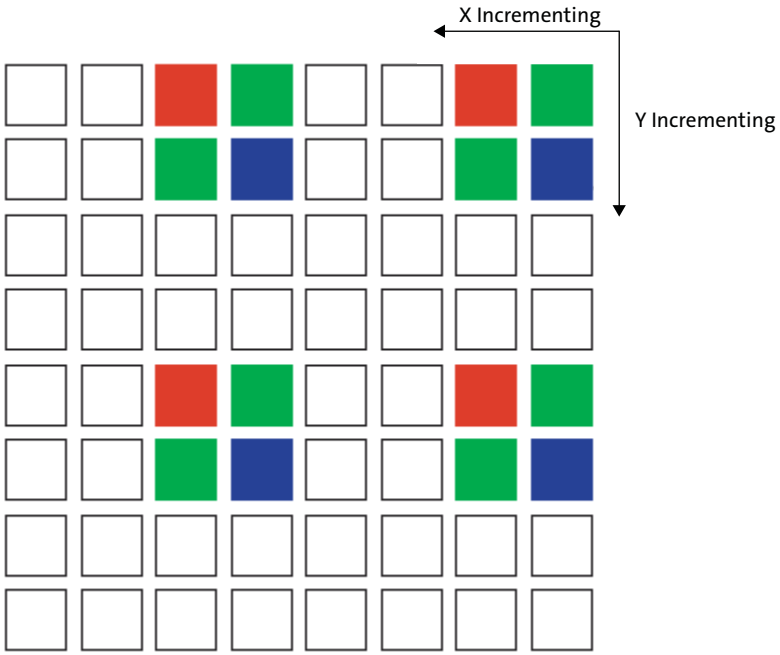


Figure 18: Pixel Readout (column and row skipping)



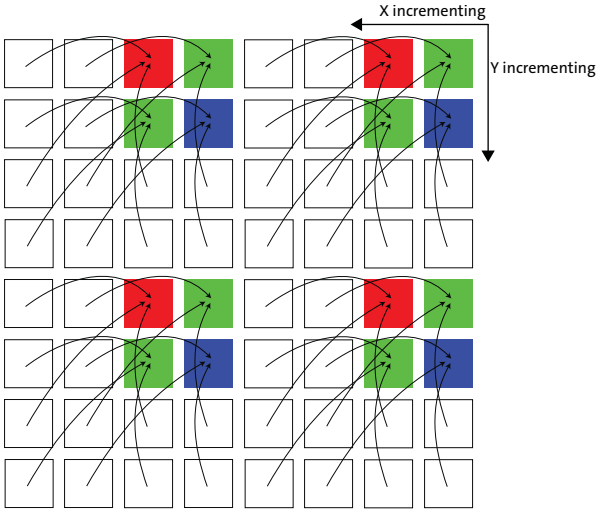
Binning

The AS0260 sensor core supports 2 x 2 binning. Binning has many of the same characteristics as subsampling but because it gathers image data from all pixels in the active window (rather than a subset of them), it achieves superior image quality and avoids the aliasing artifacts that can be a characteristic side effect of subsampling.

Binning is enabled by selecting the appropriate subsampling settings. Subsampling may require sensor window size adjustment when binning is enabled.

The effect of binning is shown in Figure 19 on page 18.

Figure 19: Pixel Readout (column and row binning)



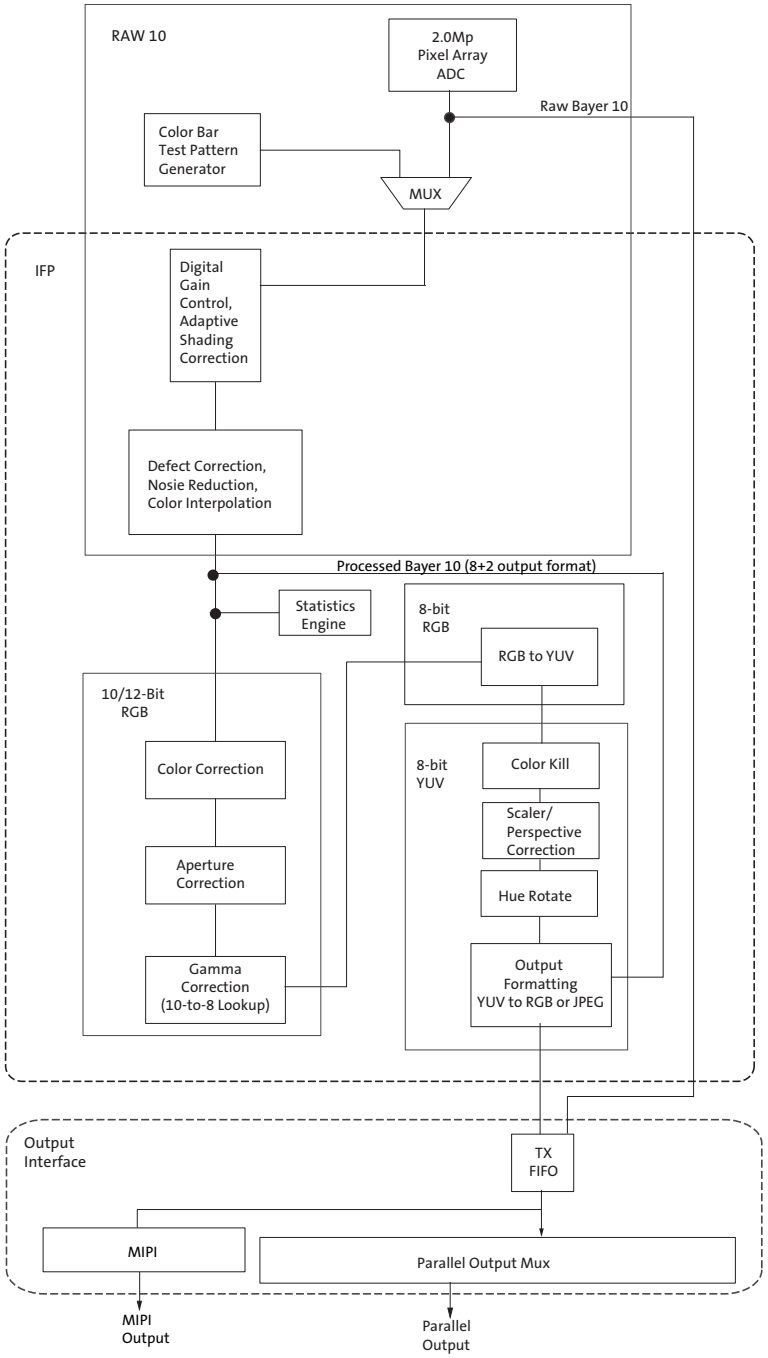
PLL

A PLL is provided to create the required PIXCLK from the input EXTCLK. The PLL is programmed through variable settings.

Image Flow Processor

Image control processing in the AS0260 is implemented in the IFP hardware logic. For normal operation, the microcontroller automatically adjusts the operational parameters of the IFP. Figure 20 shows the image data processing flow within the IFP.

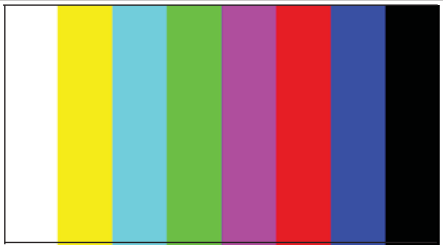
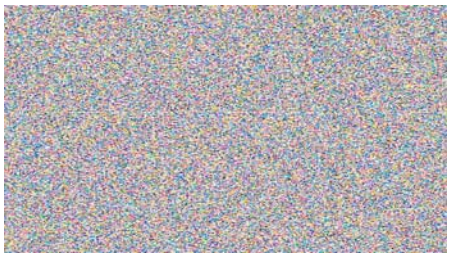
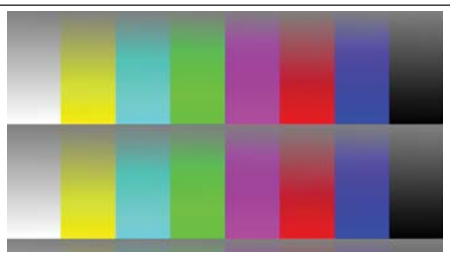
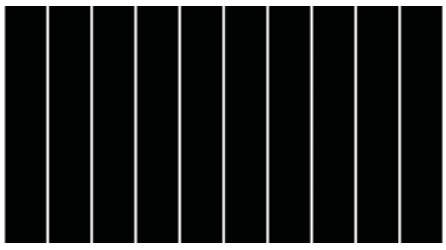
Figure 20: Image Flow Processor



For normal operation of the AS0260, streams of raw image data from the sensor core are continuously fed into the color pipeline. The AS0260 features an automatic color bar test pattern generation function to emulate sensor images as shown in Figure 21: “Color Bar Test Pattern,” on page 21. The color bar test pattern is fed to the IFP for testing the image pipeline without sensor operation.

Color bar test pattern generation can be selected by programming variables. To select enter test pattern mode VAR(0x12,0x4C) or R0xC84C =0x02, to exit this mode VAR (0x12,0x4C) or R0xC84C should be set to 0x00.

Figure 21: Color Bar Test Pattern

Test Pattern	Example
Flat Field VAR = 18, 0x4D, 0x0001	
100% Color Bars VAR = 18, 0x4D, 0x0004	
Pseudo-Random VAR = 18, 0x4D, 0x0005	
Fade-to-Gray Color Bars VAR = 18, 0x4D, 0x0008	
Walking 1s VAR = 18, 0x4D, 0x0009	

Digital Gain

Image stream processing starts with multiplication of all pixel values by a programmable digital gain. Independent color channel digital gain can be adjusted with registers.

Adaptive PGA (APGA)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. There are also other factors causing fixed pattern signal gradients in images captured by image sensors. The cumulative result of all these factors is known as image shading. The AS0260 has an embedded shading correction module that can be programmed to counter the shading effects on each individual R, Gb, Gr, and B color signal.

In some cases, different illuminants can introduce different color shading response. The APGA feature on the AS0260 will compensate for the dependency of the lens shading of the illuminant. The AS0260 will allow for up to three different illuminants to be compensated for.

Color Interpolation and Edge Detection

In the raw data stream fed by the sensor core to the IFP, each pixel is represented by a 10-bit integer, which can be considered proportional to the pixel's response to a one-color light stimulus, red, green, or blue, depending on the pixel's position under the color filter array. Initial data processing steps, up to and including the defect correction, preserve the one-color-per-pixel nature of the data stream, but after the defect correction it must be converted to a three-colors-per-pixel stream appropriate for standard color processing. The conversion is done by an edge-sensitive color interpolation module. The module adds the incomplete color information available for each pixel with information extracted from an appropriate set of neighboring pixels. The algorithm used to select this set and extract the information seeks the best compromise between preserving edges and filtering out high-frequency noise in flat field areas. The edge threshold can be set through variable settings.

Color Correction and Aperture Correction

To achieve good color fidelity of the IFP output, interpolated RGB values of all pixels are subjected to color correction. The IFP multiplies each vector of three pixel colors by a 3 x 3 color correction matrix. The three components of the resulting color vector are all sums of three 10-bit numbers. Since such sums can have up to 12 significant bits, the bit width of the image data stream is widened to 12 bits per color (36 bits per pixel). The color correction matrix can either be programmed by the user or automatically selected by the AWB algorithm implemented in the IFP. Color correction should ideally produce output colors that are independent of the spectral sensitivity and color crosstalk characteristics of the image sensor. The optimal values of the color correction matrix elements depend on those sensor characteristics and on the spectrum of light incident on the sensor. The color correction settings can be adjusted using variables.

To increase image sharpness, a programmable 2D aperture correction (sharpening filter) is applied to color-corrected image data. The gain and threshold for 2D correction can be defined through variable settings.

One-Time Programmable Memory

the AS0260 contains one-time programmable memory (OTPM), suitable for storing separate lens shading correction settings, color calibration, external mechanisms, initialization settings, and module identification, that can be programmed during the module manufacturing process. Programming the OTPM requires the use of a high voltage at the VPP pin. during normal operation, the VPP pin should be left floating. The OTPM can be accessed through the two-wire serial interface. Refer to the AS0260 Developer Guide for programming procedures.

Gamma Correction

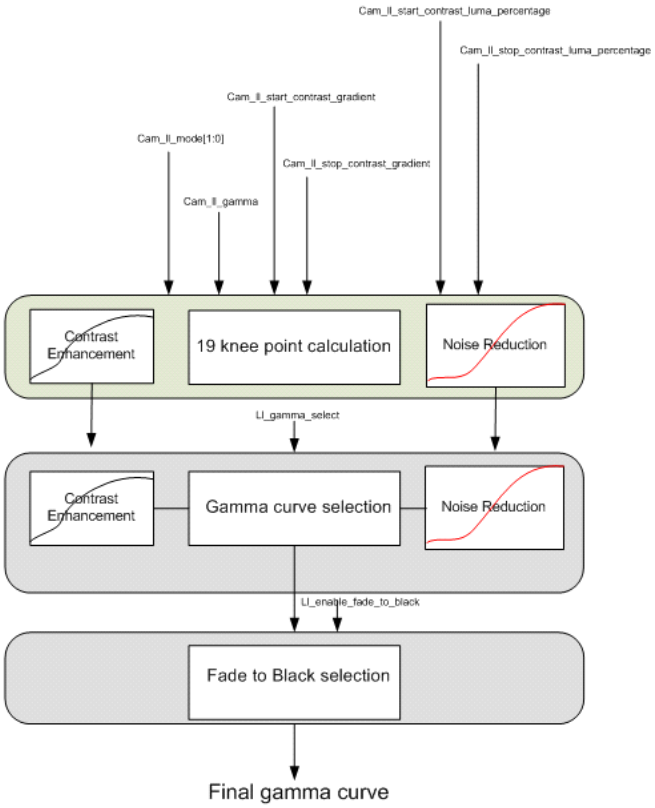
The gamma correction curve (as shown in Figure 22) is implemented as a piecewise linear function with 19 knee points, taking 12-bit arguments and mapping them to 8-bit output. The abscissas of the knee points are fixed at 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. The 8-bit ordinates are programmable through variables.

The AS0260 IFP includes a block for gamma correction that has the capability to adjust its shape, based on brightness, to enhance the performance under certain lighting conditions.

Two custom gamma correction tables may be uploaded, one corresponding to a contrast curve for brighter lighting conditions, the other one corresponding to a noise reduction curve for lower lighting conditions. Also included in this block is a Fade-to Black curve which sets all knee points to zero and causes the image to go black in extreme low light conditions.

The AS0260 has the ability to calculate the 19 point knee points based on a small number of variable inputs from the host, another option is for the host to program one or both of the 19 knee point curves. The diagram below shows how the gamma feature interacts in AS0260.

Figure 22: Gamma Interaction



Gamma Knee Point Calculation

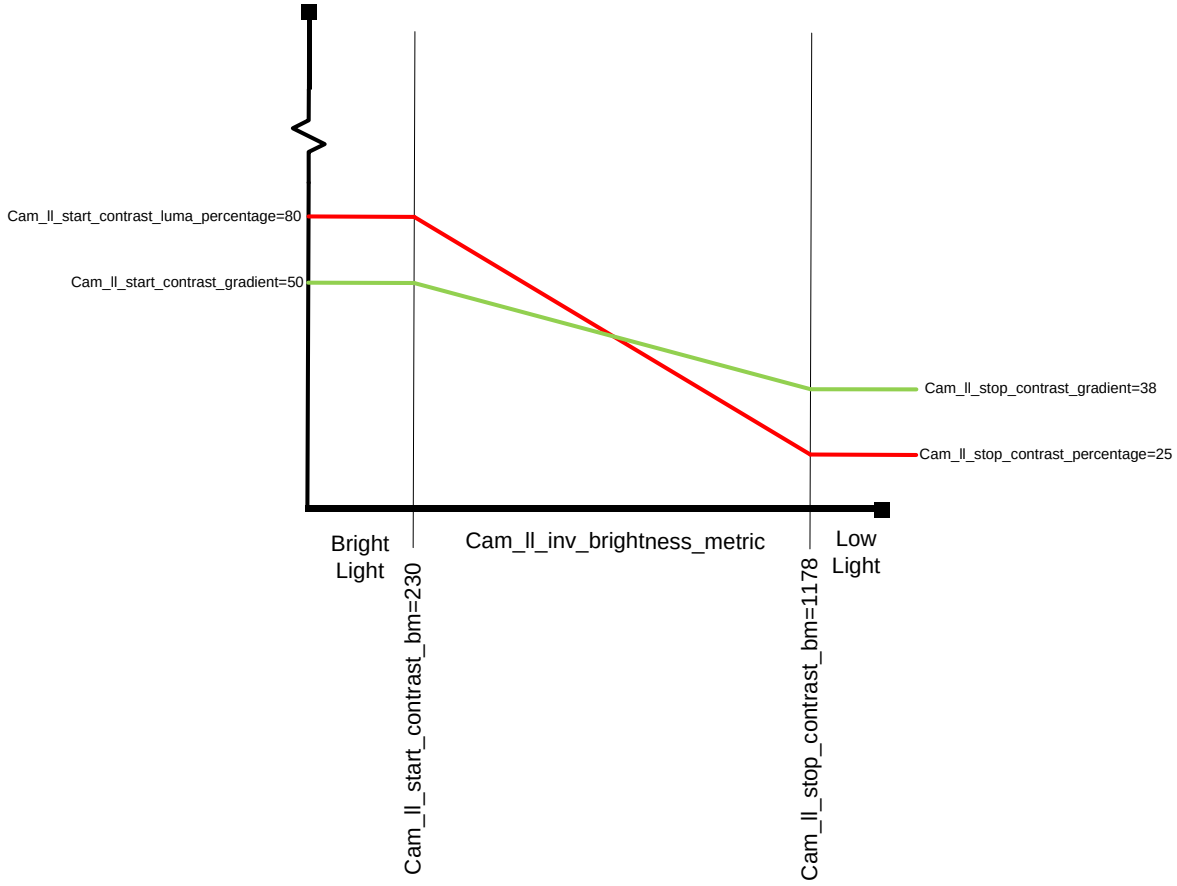
The AS0260 allows for the 19 knee point curves to be programmed based off a small number of variables. The table below shows the variables which are required.

Table 6: Variables Required for Gamma Knee Point Calculation

Variable	Name	Function
VAR(0x12,0x0168) or (R0xC968)	cam_ll_llmode	0x00: User will program 19 knee point gamma curves 0x01: AS0260 will calculate 19 knee point for contrast curve (first curve or table). 0x02: AS0260 will calculate 19 knee point for noise reduction curve (second curve or table). 0x03: AS0260 will calculate both 19 knee point curves.
VAR(0x12,0x01C8) or (R0xC9C8)	cam_ll_start_contrast_bm	Interpolation start point for first curve
VAR(0x12,0x01CA) or (R0xC9CA)	cam_ll_stop_contrast_bm	Interpolation stop point for second curve
VAR(0x12,0x01CC) or (R0xC9CC)	cam_ll_gamma	The value of the gamma curve, this is applied to both 19 knee point curves. The default is 220, this equates to a gamma of 2.2.
VAR(0x12,0x01CE) or (R0xC9CE)	cam_ll_start_contrast_gradient	The value of the contrast gradient that would be used for the first curve
VAR(0x12,0x01CF) or (R0xC93CF)	cam_ll_stop_contrast_gradient	The value of the contrast gradient that would be used for the second curve
VAR(0x12,0x01D0) or (R0xC9D0)	cam_ll_start_contrast_luma_percentage	The percentage of target luma for the inflexion point in the first curve
VAR(0x12,0x01D1) or (R0xC9D1)	cam_ll_start_contrast_luma_percentage	The percentage of target luma for the inflexion point second curve
VAR(0x12,0x01E2) or (R0xC9E2)	cam_ll_inv_brightness_metric	Measure of scene brightness, reference points for cam_ll_start_contrast_bm and cam_ll_stop_contrast_bm

The diagram below shows the interaction of the variables and cam_ll_inv_brightness_metric.

Figure 23: Gamma Reference Variables Against Brightness Metric



ON Semiconductor recommends that cam_ll_start_contrast_bm is set at 100lux and cam_ll_stop_contrast_bm is set at 20lux, but the actual setting is at the discretion of the user.

The recommended setting for cam_ll_llmode is 0x03. This allows the AS0260 to calculate both of the 19 knee point curves based on the user inputs, otherwise the user will have to program the 19 knee point curves.

Gamma Curve Selection

The AS0260 allows the user to select between the two-curve interpolation mode or fixed mode using either of the curves

Table 7: Gamma Curve Selection

Variable	Name	Function
VAR(0x0F,0x0008) or (R0xBC08)	ll_gamma_select	0x00= Auto curve select. The curves will interpolate based on settings of cam_ll_start_contrast_bm and cam_ll_stop_contrast_bm 0x01 = Contrast curve is only used 0x02 =Noise reduction curve is only used

Fade to Black Selection

The final stage of the gamma flow is the enabling and use of fade to black. The AS0260 IFP allows for the image to fade to black under extreme low-light conditions. This feature enables users to optimize the performance of the sensor under low-light conditions. It minimizes the perception of noise and artifacts while the available illumination is diminishing.

This feature has two user set points that reference the brightness of the scene. When the Fade-to-Black starts, it will interpolate to the end point as the light falls until it gets to the end point. When at the end point, the image will be black.

Table 8: Fade-to-Black Selection

Variable	Name	Function
VAR(0x0F,0x0002) or (R0xBC02)	ll_mode	When bit 3=1 this will enable fade to black feature
VAR(0x12,0x01DA) or (R0xC9DA)	cam_ll_start_fade_to_black_luma	Starting point for fade to black to begin
VAR(0x12,0x01DC) or (R0xC9DC)	cam_ll_stop_fade_to_black_luma	End point for fade to black, after this point the image will be black

ON Semiconductor recommends that cam_ll_start_fade_to_black_luma is set at 10 lux and cam_ll_stop_fade_to_black_luma is set at 5lux, but the actual setting is at the discretion of the user.

Image Scaling and Cropping

To ensure that the size of images output by the AS0260 can be tailored to the needs of all users, the IFP includes a scaler module. When enabled, this module performs rescaling of incoming images—shrinks them to arbitrarily selected width and height without reducing the field of view and without discarding any pixel values.

By configuring the cropped and output windows to various sizes, different zooming levels including 4X, 2X, and 1X can be achieved. The location of the cropped window is configurable so that panning is also supported. The height and width definitions for the output window must be equal to or smaller than the cropped image. The image cropping and scaler module can be used together to implement a digital zoom and pan.

Hue Rotate

The AS0260 has integrated hue rotate. This feature will help for improving the color image quality and give customers the flexibility for fine color adjustment and special color effects.

Table 9: Hue Control

Variable	Name	Function
R0x3210[9]	Enable Hue Rotate	Setting this bit to 1 enables hue rotate
VAR(0x12,0x73)	Hue Angle	Adjusts the global hue angle adjustment (if enabled). 0xEA = –22° 0x00 = 0° 0x16 = +22°

Figure 24: 0° Hue

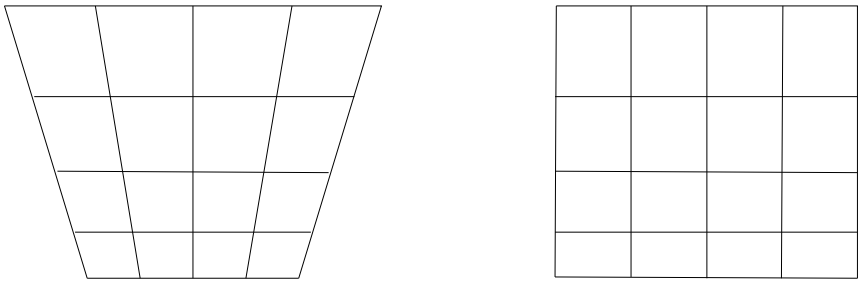
Figure 25: –22° Hue

Figure 26: +22° Hue

Vertical Perspective Correction

The AS0260 has vertical perspective correction (VPC), this allows the user to correct (within limits) for an off-horizontal axis camera.

Original Image *VPC Corrected Image*

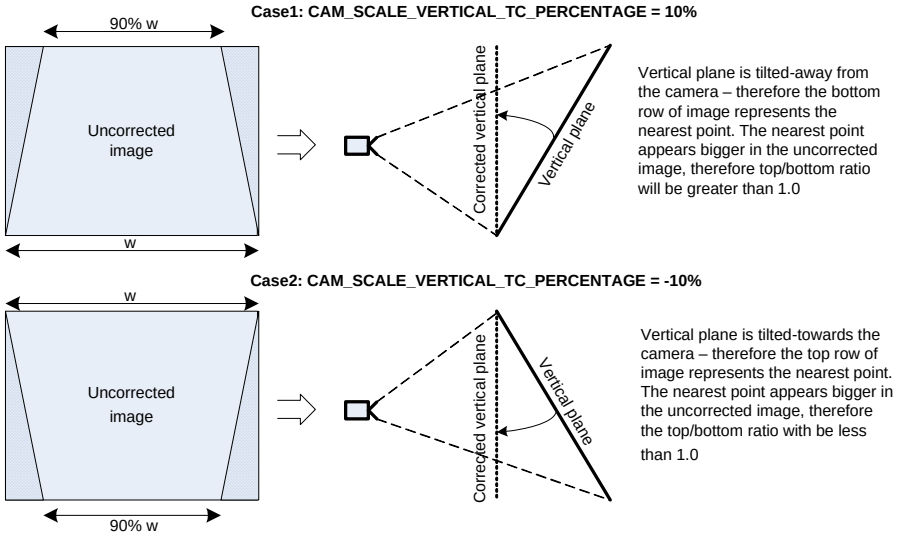


VPC is performed using a mixture of scale and crop, the variables which control this are:

Table 10: Variables Controlling VPC

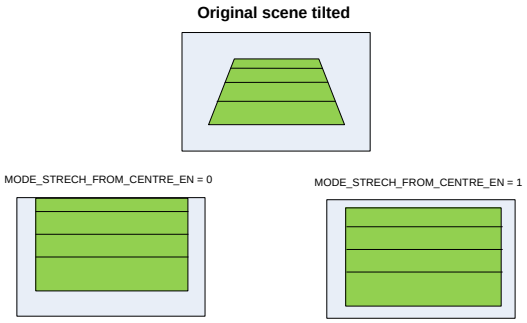
Variable	Name	Function
VAR(0x12,0x005E) or (R0xC85E)	cam_scale_vertical_tc_mode	When bit 0 is set will depend if the cropping is done through the center or top or bottom
VAR(0x12,0x0060) or (R0xC860)	cam_scale_vertical_tc_percentage	The amount of tilt (perspective) correction to be applied. If negative, this value represents% of FOV reduction with the bottom line unaffected. If positive, this value represents% of FOV reduction with the top line unaffected
VAR(0x12,0x0062) or (R0xC862)	cam_scale_vertical_tc_stretch_factor	Ratio of vertical stretching against the percentage applied. Vertical stretching = stretch factor x percentage/2

The effect of using cam_scale_vertical_tc_percentage can be seen below.



Cam_scale_vertical_tc_percentage will define how much tilt needs to be corrected for in percentage terms. When used in conjunction with cam_scale_vertical_tc_stretch_factor, which will stretch the image vertically.

The effect of using cam_scale_vertical_tc_mode can be seen below.



Camera Control and Auto Functions

General Purpose I/Os

The three general purpose I/Os (GPIOs) of the AS0260 can be configured in multiple ways. Each of the I/Os can be used for multiple purposes and can be programmed from the host. The GPIOs are powered by their own power supply.

Auto Exposure

The auto exposure algorithm performs automatic adjustments of the image brightness by controlling exposure time and analog gains of the sensor core as well as digital gains applied to the image.

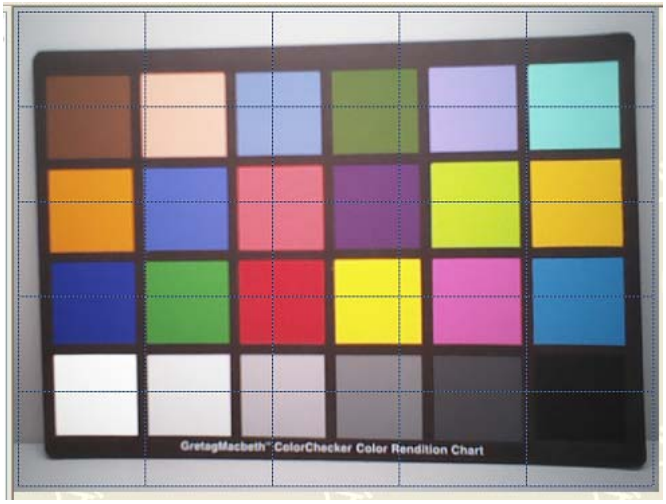
Auto exposure is implemented by a firmware driver that analyzes image statistics collected by the exposure measurement engine, makes a decision, and programs the sensor core and color pipeline to achieve the desired exposure. The measurement engine subdivides the image into 25 windows organized as a 5 x 5 grid.

Two auto exposure algorithm modes are available:

- Average brightness tracking (ABT) or Average Y (ae_rule_algo=0x00)
  The average brightness tracking AE uses a constant average tracking algorithm where a target brightness value is compared to a current brightness value, and the gain and integration time are adjusted accordingly to meet the target requirement.
- Weighted Average Brightness (ae_rule_algo=0x01)
  Each of the 25 windows can be assigned a weight, which can be changed independently of each other. The effect of these weights will allow the center of the image to be weighted higher than the periphery. See Figure 27.

Figure 27: 5 x 5 Grid

W 0,0	W 0,1	W 0,2	W 0,3	W 0,4
W 1,0	W 1,1	W 1,2	W 1,3	W 1,4
W 2,0	W 2,1	W 2,2	W 2,3	W 2,4
W 3,0	W 3,1	W 3,2	W 3,3	W 3,4
W 4,0	W 4,1	W 4,2	W 4,3	W 4,4



- Adaptive Weighted AE for highlights (ae_rule_algo=0x02)- The scene will be exposed based on zone luma and will adapt for highlights. This would expose an image when the background is dark.
- Adaptive Weighted AE for lowlights(ae_rule_algo=0x03)- The scene will be exposed based on zone luma and will adapt for lowlights. This would expose an image when the background is brighter.

Some sample images which show the benefits of the different AE modes

Light Background

Average Brightness Tracking or Average Y

Weighted Average Brightness (center)

Adaptive weighted based on zone luma (highlights)

Adaptive weighted based on zone luma (lowlights)

In the use case above the Adaptive weighted for lowlights exposes the face slightly better when compared to the Weighted Average Brightness. The face is moved off axis and the images are retaken.

Weighted Average Brightness (center) (lowlights)

Adaptive weighted based on zone luma

This shows the advantage of using the Adaptive weighted AE (lowlights), when the face moves off center it still is exposed correctly.

Dark Background

Average Brightness Tracking or Average Y

Weighted Average Brightnes s(centre)

Adaptive weighted based on zone luma (highlights)

Adaptive weighted based on zone luma (lowlights)

In this use case the Adaptive weighted for highlights will expose the face the best when compared to the other options.

AE Driver

Other algorithm features include the rejection of fast fluctuations in illumination (time averaging), control of speed of response, and control of the sensitivity to the small changes. While the default settings are adequate in most situations, the user can program target brightness, measurement window, and other parameters described above.

The driver changes AE parameters (integration time, gains, and so on) to drive brightness to the programmable target. The value of the single step approach to the target value can be controlled.

To avoid unwanted reaction of AE on small fluctuations of scene brightness or momentary scene changes, the AE driver uses a temporal filter for luma and a threshold around the AE luma target. The driver changes AE parameters only if the buffered luma is larger than the AE target step and pushes the luma beyond the threshold.

Exposure Control

To achieve the required amount of exposure, the AE driver adjusts the sensor integration time, gains and IFP digital gains. In addition, a variable is available for the user to adjust the overall brightness of the scene. To reject flicker, integration time is typically adjusted in increments of steps. The incremental step specifies the duration in row times equal to one flicker period. Thus, flicker is rejected if integration time is kept a natural factor of the flicker period.

Auto White Balance

The AS0260 has a built-in AWB algorithm designed to compensate for the effects of changing spectra of the scene illumination on the quality of the color rendition. The algorithm consists of two major parts: a measurement engine performing statistical analysis of the image and a driver performing the selection of the optimal color correction matrix and SOC digital gain. While default settings of these algorithms are adequate in most situations, the user can reprogram base color correction matrices, place limits on color channel gains, and control the speed of both matrix and gain adjustments. The AS0260 AWB displays in color temperature, the range of which is defined by the programming of the CCM matrixes.

Flicker Detection and Avoidance

Flicker occurs when the integration time is not an integer multiple of the period of the light intensity. The AS0260 can be programmed to detect and avoid flicker for 50 or 60 Hz. For integration times below the light intensity period (10ms for 50Hz environment), flicker cannot be avoided. The AS0260 supports an indoor AE mode, that will ensure flicker-free operation. The AS0260 will calculate all flicker parameter based on the sensor settings which are programmed in the cam control variables.

Ambient Light Measurement

To facilitate the measurement of the ambient light lux level used in the dimming of notebook computer LCD screens and other light-sensitive peripherals, the AS0260 has an ambient light measurement mode. This mode takes the image data from the scene and translates it to a value that can be read by the host system over the two-wire serial interface. This ambient light measurement can be made during normal video streaming or during soft-standby.

Presence Detection

The AS0260 offers a presence detection mode to reduce the amount of processing the host system needs to do in power regulation (LCD dimming or on/off controls) functions. This presence detection mode operates only when the AS0260 is placed in soft-standby mode. When the presence of a large object (like a person sitting down in front of a notebook computer) is detected by the image sensor, a register bit is changed to indicate an object's presence has been detected. During presence detection mode, the host system is expected to regularly poll this bit over the two-wire serial interface to determine if and when an object's presence has been detected.

Output Conversion and Formatting

The YUV data stream can either exit the color pipeline as is or be converted before exit to an alternative YUV or RGB data format.

Color Conversion Formulas

Y'U'V'

This conversion is BT 601 scaled to make YUV range from 0 through 255. This setting is recommended for JPEG encoding and is the most popular, although it is not well defined and often misused in various operating systems.

$$Y' = 0.299 \times R' + 0.587 \times G' + 0.114 \times B' \quad \text{(EQ 1)}$$

$$U' = 0.564 \times (B' - Y') + 128 \quad \text{(EQ 2)}$$

$$V' = 0.713 \times (R' - Y') + 128 \quad \text{(EQ 3)}$$

There is an option where 128 is not added to U'V'.

Y'Cb'Cr' Using sRGB Formulas

The AS0260 implements the sRGB standard. This option provides YCbCr coefficients for a correct 4:2:2 transmission.

Note: 16 < Y601< 235; 16 < Cb < 240; 16 < Cr < 240; and 0 < = RGB < = 255

$$Y' = (0.2126 \times R' + 0.7152 \times G' + 0.0722 \times B') \times (219/256) + 16 \quad \text{(EQ 4)}$$

$$Cb' = 0.5389 \times (B' - Y') \times (224/256) + 128 \quad \text{(EQ 5)}$$

$$Cr' = 0.635 \times (R' - Y') \times (224/256) + 128 \quad \text{(EQ 6)}$$

Y'U'V' Using sRGB Formulas

These are similar to the previous set of formulas, but have YUV spanning a range of 0 through 255.

$$Y' = 0.2126 \times R' + 0.7152 \times G' + 0.0722 \times B' \quad \text{(EQ 7)}$$

$$U' = 0.5389 \times (B' - Y') + 128 = -0.1146 \times R' - 0.3854 \times G' + 0.5 \times B' + 128 \quad \text{(EQ 8)}$$

$$V' = 0.635 \times (R' - Y') + 128 = 0.5 \times R' - 0.4542 \times G' - 0.0458 \times B' + 128 \quad \text{(EQ 9)}$$

There is an option to disable adding 128 to U'V'. The reverse transform is as follows:

$$R' = Y + 1.5748 \times (V - 128) \quad \text{(EQ 10)}$$

$$G' = Y - 0.1873 \times (U - 128) - 0.4681 \times (V - 128) \quad \text{(EQ 11)}$$

$$B' = Y + 1.8556 \times (U - 128) \quad \text{(EQ 12)}$$

Uncompressed YUV/RGB Data Ordering

The AS0260 supports swapping YCbCr mode, as illustrated in Table 11.

Table 11: YCbCr Output Data Ordering

Mode	Data Sequence			
Default (no swap)	Cb_i	Y_i	Cr_i	Y_{i+1}
Swapped CrCb	Cr_i	Y_i	Cb_i	Y_{i+1}
Swapped YC	Y_i	Cb_i	Y_{i+1}	Cr_i
Swapped CrCb, YC	Y_i	Cr_i	Y_{i+1}	Cb_i

The RGB output data ordering in default mode is shown in Table 12. The odd and even bytes are swapped when luma/chroma swap is enabled. R and B channels are bitwise swapped when chroma swap is enabled.

Table 12: RGB Ordering in Default Mode

Mode (Swap Disabled)	Byte	D7D6D5D4D3D2D1D0
565RGB	Odd	R7R6R5R4R3G7G6G5
	Even	G4G3G2B7B6B5B4B3
555RGB	Odd	0 R7R6R5R4R3G7G6
	Even	G4G3G2B7B6B5B4B3
444xRGB	Odd	R7R6R5R4G7G6G5G4
	Even	B7B6B5B4 0 0 0 0
x444RGB	Odd	0 0 0 0 R7R6R5R4
	Even	G7G6G5G4B7B6B5B4

Uncompressed Bayer Bypass Output

Raw or processed 10-bit Bayer data from the sensor core can be output in bypass mode by:

1. Using both DOUT[7:0] and DOUT_LSB[1:0].
2. Using only DOUT[7:0] with a special 8 + 2 data format, shown in Table 13.

Table 13: 2-Byte Bayer Format

2-Byte Bayer Format	Bits Used	Bit Sequence
Odd bytes	8 data bits	$D_9D_8D_7D_6D_5D_4D_3D_2$
Even bytes	2 data bits + 6 unused bits	0 0 0 0 0 0 D_1D_0

JPEG Encoder

The JPEG compression engine in the AS0260 is a highly integrated, high-performance solution that provides for low power consumption and programmability of JPEG compression parameters for image quality control.

The JPEG encoding block is designed for continuous image flow and is ideal for low power applications. After initial configuration for a target application, it can be controlled easily for instantaneous stop or restart. A flexible configuration and control interface allows for full programmability of various JPEG-specific parameters and tables.

JPEG Encoding Highlights

- Sequential DCT (baseline) ISO/IEC 10918-1 JPEG-compliant
- Grayscale and YCbCr 4:2:2 format compression
- Support for JPEG 4:2:0 output for image widths that are less than 960 pixels
- Support for two pairs of programmable quantization tables
- Support for user-defined quantization tables
- Quality/compression ratio control capability
- 30 fps JPEG capability at full resolution with or without JFIF-compliant header
- Programmable automatic control of compression ratio
- JPEG encoded stream can work in continuous mode or spoof mode
- JPEG encoded stream working in continuous mode can only transmit on the parallel output port
- In spoof mode, data is output with programmed spoof frame sizes; dummy pixels may be padded as necessary
- Support for Scalado SpeedTags™
- MIPI data types can be used to output a status segment with a different datatype code than the JPEG data
- Spoof-frame height can be ignored in spoof mode
- Optional JFIF header generation

JPEG Output Interface

JPEG Data

JPEG data can be output in both the parallel and the serial MIPI streams. In the parallel output interface, JPEG data is output on the 8-bit parallel bus DOUT[7:0], with FV, LV, and PIXCLK. JPEG output data is valid when both FV and LV are asserted. When the JPEG data output for the frame completes, LV and FV are de-asserted.

The AS0260 can transmit JPEG data using two different formats: JPEG continuous stream and JPEG spoof stream. In both formats, JPEG status segments containing information (resolution, file size, and status) about the image can be inserted into the output streams. The following sections describe the two streaming methods.

JPEG Continuous Stream

JPEG continuous stream goes out only through the parallel output interface, and supports the following features:

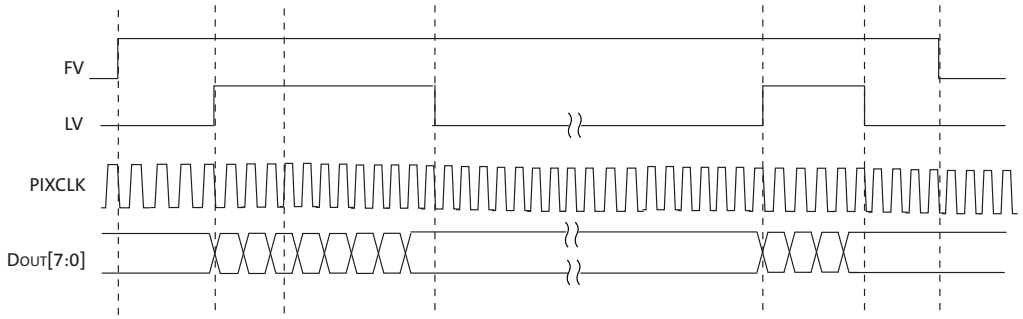
- Duplicate FV on LV
- Append JPEG status segment at the end of the data stream

When enabled, the pixel clock output can be generated continuously during invalid data periods (between FV and between LV). In this streaming mode, the amount of valid data within each line (LV = 1) is variable. Figure 28 through Figure 29 on page 39 are examples of the JPEG stream through the parallel output interface.

Figure 28 illustrates data output when the pixel clock output is generated continuously during invalid data periods. LV is of variable length based on data output rate.

In default mode, data transitions on the falling edge of PIXCLK and the host must capture data on the rising edge of PIXCLK. The PIXCLK is also configurable and its polarity can be reversed through the use of register settings.

Figure 28: JPEG Continuous Data Output



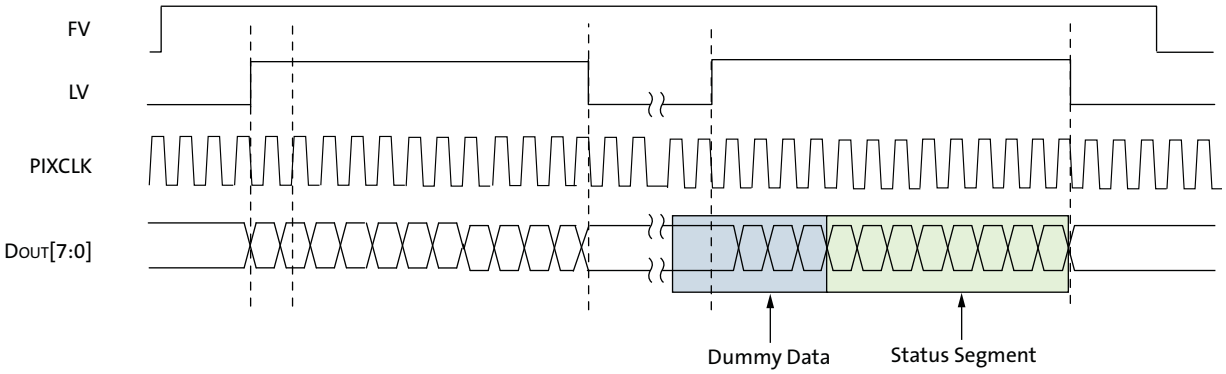
JPEG Spoof Stream

The JPEG compressed data can be output in spoof mode. The amount of expected pixel data is defined by the spoof width and spoof height registers. If the valid JPEG data is less than expected size defined, a dummy data pattern with a value of 0xFF will be padded. There is an option to ignore spoof height so dummy data padding is limited to less than one spoof line.

When enabled, the pixel clock output can be generated continuously during invalid data periods (between FV and between LV). In spoof streaming mode, the amount of valid data within each line (LV = 1) is constant.

Figure 29 illustrates the JPEG spoof output when pixel clock is generated continuously during invalid data periods between LV. The status segment is inserted at the end of the stream.

Figure 29: JPEG Spoof Mode Timing with Continuous Clock



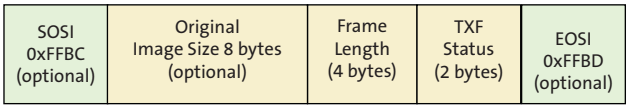
JPEG Spoof Stream in MIPI Output Mode

In MIPI output mode, only the JPEG spoof stream can be output. Similar to the parallel output interface, the amount of expected pixel data is defined by the spoof width and spoof height registers. If the valid JPEG data is less than expected size defined, dummy data will be padded.

JPEG Status Segment

To provide the user quick knowledge of the status when the JPEG is enabled, a JPEG status segment can be appended at the end of frame. The status segment can be enclosed by SOSI/EOSI codes, as shown in Figure 30.

Figure 30: JPEG Status Segment Structure



The contents of the status segment are summarized as follows:

- SOSI, start of status information, which is coded as 0xFFBC
- Original image size:
  - Reserved (4 bytes, 0x00)
  - The width of uncompressed full image (2 bytes)
  - The height of uncompressed full image (2 bytes)
- Byte count in compressed JPEG frame (4 bytes)
- Status (2 bytes)
- EOSI, end of status information, which is coded as 0xFFBD

There are configurable options that can be set to match legacy parts.

Scalado SpeedTags™ Support

The AS0260 supports Scalado SpeedTags™ by inserting markers into the JPEG stream. This is enabled by the register bit TX_SS.jpeg_ctrl.jpeg_insert_rajpeg_markers.

MJPEG Format

The AS0260 supports an MJPEG formatted data output stream. The descriptions following are based on what the SOC will deliver to a host USB bridge device with which the SOC would be combined to form a camera for PC applications. The USB Bridge host is responsible for assembling the individual frames from the SOC into a USB Video Class (UVC) video data stream.

Stream Breakdown

An MJPEG video stream consists of the following sequence of data sections. Each JPEG frame must have the following characteristics:

- Color Encoding is YCbCr
- 8 bits per color component, (24 bits/pixel before subsampling)
- 422 Subsampling
- Baseline sequential DCT (SOF0)

Video Stream Header

This section of the video stream is not implemented in the AS0260. The content of this section is determined by the host.

MJPEG Frame Header

This section of the video stream is not implemented in the AS0260. The content of this section is determined by the host. This section is just 8 bytes of information at the start of each frame. The first 4 bytes are:

0x30 0x30 0x64 0x62 # 00db

The next 4 bytes are the length of the following JPEG frame including all the bytes described in sections 3.3 - 3.5. The 4 byte count value is output LSB first. For example, if the JPEG data was 0x0002_51dc bytes long, the last 4 bytes in the MJPEG frame header would be:

0xdc, 0x51, 0x02, 0x00

Since this field contains the byte count of the compressed JPEG data, it cannot be added by the AS0260, but must be added by the host after frame compression is complete and the byte count known.

JPEG Header Without Huffman Tables

This is a normal JPEG header except for the fact that the DHT segment (Define Huffman Table) is not included. The Huffman table is not included because the MJPEG spec defines the Huffman table to be fixed for all frames. The header segments that will be included are listed below including examples. Note that data values in the examples are in hex. Comments are in decimal.

- SOI, Start of Image. 2 bytes.
  ff d8
- APP0, Application Segment 0. N bytes. Example JFIF marker:
  ff e0 00 10
  4a 46 49 46 00 01 02 00 00 01 00 01 00 00
- DQT, Define Quantization Tables. 134 bytes. Example:
  ff db 00 84

 # 8-bit, Table 0

```
00
10 0b 0c 0e 0c 0a 10 0e 0d 0e 12 11 10 13 18 28
1a 18 16 16 18 31 23 25 1d 28 3a 33 3d 3c 39 33
38 37 40 48 5c 4e 40 44 57 45 37 38 50 6d 51 57
5f 62 67 68 67 3e 4d 71 79 70 64 78 5c 65 67 63
```

 # 8-bit, Table 1

```
01
11 12 12 18 15 18 2f 1a 1a 2f 63 42 38 42 63 63
63 63 63 63 63 63 63 63 63 63 63 63 63 63 63 63
63 63 63 63 63 63 63 63 63 63 63 63 63 63 63 63
63 63 63 63 63 63 63 63 63 63 63 63 63 63 63 63
```

The quantization table can be adjusted for each frame for more or less compression.

- DRI, Define Restart Interval. 6 bytes. Example:
  ff dd 00 04 00 78

This segment is optional. The host will determine whether to include Restart markers and at what interval.

- SOF0, Start of Frame 0. 19 bytes. Example:
  ff c0 00 11

```
08          # Sample precision
04  38      # Number of rows = 1080
07  80      # Number of columns = 1920

03          # Number of components
01  21  00  # Component 1: HSF= 2, VSF = 1, Q Table = 0
02  11  01  # Component 2: HSF= 1, VSF = 1, Q Table = 1
03  11  01  # Component 3: HSF= 1, VSF = 1, Q Table = 1
```

- SOS, Start of Scan. 14 bytes. Example:
  ff da 00 0c

```
03      # Number of components
01  00  # Component 1: DC table 0, AC table 0
02  11  # Component 2: DC table 1, AC table 1
03  11  # Component 3: DC table 1, AC table 1
00      # Start of spectral selection
3f      # End of spectral selection
00      # Successive approximation high/low
```

Compressed Data With or Without Restart Markers

This is the compressed binary data of the frame which can be decoded to display the captured image. The SOC can be configured to insert restart marker at programmable intervals.

EOI

This is the End of Image code. It is only 2 bytes long.

ff d9

Optional Padding

Padding between frames is optional and may be added by the host if desired. The data values used for padding are not defined.

Video Stream Footer

The items from MJPEG header to Optional Padding (inclusive) described above are repeated once per frame until the end of the video stream, after which a data section may be added at the end of the video stream. Addition of this section is left to the host.

Huffman Table

JPEG implementations exists in many ON Semiconductor parts, include the Huffman table (in the DHT segment.) The AS0260 will use the Huffman Table defined in the MJPEG specification (listed below) and will not include the Huffman Table in the header, also as defined by that specification.

The required Huffman table (copied from BMPDIB.TXT) is:

```
/* Default DHT Segment */
MJPGHDTSEG_STORAGE BYTE MJPGDHTSeg[0x1A0] = {
 /* JPEG DHT Segment for YCrCb omitted from MJPG data */
 0xFF   0xC4   0x01   0xA2
 0x00   0x00   0x01   0x05   0x01   0x01   0x01   0x01   0x01   0x01   0x00   0x00   0x00   0x00   0x00
 0x00   0x00   0x00   0x01   0x02   0x03   0x04   0x05   0x06   0x07   0x08   0x09   0x0A   0x0B   0x01
 0x00   0x03   0x01   0x01   0x01   0x01   0x01   0x01   0x01   0x01   0x01   0x00   0x00   0x00   0x00

 0x00   0x00   0x01   0x02   0x03   0x04   0x05   0x06   0x07   0x08   0x09   0x0A   0x0B   0x10   0x00
 0x02   0x01   0x03   0x03   0x02   0x04   0x03   0x05   0x05   0x04   0x04   0x00   0x00   0x01   0x7D
 0x01   0x02   0x03   0x00   0x04   0x11   0x05   0x12   0x21   0x31   0x41   0x06   0x13   0x51   0x61
 0x07   0x22   0x71   0x14   0x32   0x81   0x91   0xA1   0x08   0x23   0x42   0xB1   0xC1   0x15   0x52
 0xD1   0xF0   0x24   0x33   0x62   0x72   0x82   0x09   0x0A   0x16   0x17   0x18   0x19   0x1A   0x25
 0x26   0x27   0x28   0x29   0x2A   0x34   0x35   0x36   0x37   0x38   0x39   0x3A   0x43   0x44   0x45
 0x46   0x47   0x48   0x49   0x4A   0x53   0x54   0x55   0x56   0x57   0x58   0x59   0x5A   0x63   0x64

 0x65   0x66   0x67   0x68   0x69   0x6A   0x73   0x74   0x75   0x76   0x77   0x78   0x79   0x7A   0x83
 0x84   0x85   0x86   0x87   0x88   0x89   0x8A   0x92   0x93   0x94   0x95   0x96   0x97   0x98   0x99
 0x9A   0xA2   0xA3   0xA4   0xA5   0xA6   0xA7   0xA8   0xA9   0xAA   0xB2   0xB3   0xB4   0xB5   0xB6
 0xB7   0xB8   0xB9   0xBA   0xC2   0xC3   0xC4   0xC5   0xC6   0xC7   0xC8   0xC9   0xCA   0xD2   0xD3
 0xD4   0xD5   0xD6   0xD7   0xD8   0xD9   0xDA   0xE1   0xE2   0xE3   0xE4   0xE5   0xE6   0xE7   0xE8
 0xE9   0xEA   0xF1   0xF2   0xF3   0xF4   0xF5   0xF6   0xF7   0xF8   0xF9   0xFA   0x11   0x00   0x02
 0x01   0x02   0x04   0x04   0x03   0x04   0x07   0x05   0x04   0x04   0x00   0x01   0x02   0x77   0x00

 0x01   0x02   0x03   0x11   0x04   0x05   0x21   0x31   0x06   0x12   0x41   0x51   0x07   0x61   0x71
 0x13   0x22   0x32   0x81   0x08   0x14   0x42   0x91   0xA1   0xB1   0xC1   0x09   0x23   0x33   0x52
 0xF0   0x15   0x62   0x72   0xD1   0x0A   0x16   0x24   0x34   0xE1   0x25   0xF1   0x17   0x18   0x19
 0x1A   0x26   0x27   0x28   0x29   0x2A   0x35   0x36   0x37   0x38   0x39   0x3A   0x43   0x44   0x45
 0x46   0x47   0x48   0x49   0x4A   0x53   0x54   0x55   0x56   0x57   0x58   0x59   0x5A   0x63   0x64
 0x65   0x66   0x67   0x68   0x69   0x6A   0x73   0x74   0x75   0x76   0x77   0x78   0x79   0x7A   0x82
 0x83   0x84   0x85   0x86   0x87   0x88   0x89   0x8A   0x92   0x93   0x94   0x95   0x96   0x97   0x98
```

```
0x99  0x9A  0xA2  0xA3  0xA4  0xA5  0xA6  0xA7  0xA8  0xA9  0xAA  0xB2  0xB3  0xB4  0xB5
0xB6  0xB7  0xB8  0xB9  0xBA  0xC2  0xC3  0xC4  0xC5  0xC6  0xC7  0xC8  0xC9  0xCA  0xD2
0xD3  0xD4  0xD5  0xD6  0xD7  0xD8  0xD9  0xDA  0xE2  0xE3  0xE4  0xE5  0xE6  0xE7  0xE8
0xE9  0xEA  0xF2  0xF3  0xF4  0xF5  0xF6  0xF7  0xF8  0xF9  0xFA
};
```

Host Stream Assembly

The output from the SOC will be in frames. Each frame will contain the following data; JPEG header without Huffman tables, Compressed data with or without restart markers and EOI. The host will need to add the following data; Video stream header, MJPEG frame header, Optional padding and Video stream footer and assemble all the data components in the correct sequence to create the UVC compliant stream.

UVC Interface

The AS0260 supports a set of UVC (USB Video Class) controls, in order to simplify the integration of the AS0260 with a host's USB bridge (or ISP) device.

The AS0260 firmware includes a 'UVC Control' component that augments the CamControl variables. The UVC Control component sits above the CamControl interface (in terms of abstraction) and acts as a 'virtual host'. The intention is that CamControl and all other components are unaware of the UVC Control component.

UVC Control exposes a new 'UVC control' page of shared variables to the host. This page contains variables compliant with the UVC 1.1 specification (where possible). The variables on this page are named to match the UVC specification, and have matching data sizes, units and ranges as required. Each UVC variable is 'virtual' - it does not control any AS0260 function directly.

AS0260 therefore provides a 'dual-personality' host interface:

The primary CamControl interface, this interface exposes the full feature-set of the device.

The secondary UVC Control interface, which simplifies integration of AS0260 into a PC-Cam application.

Constraints

There are a number of constraints imposed on the host in order to simplify the implementation of the UVC feature; the following sections will detail the limitations.

No Simultaneous Operation There is a constraint that these two interfaces should not be used by the host simultaneously. The assumption is that the host will use the CamControl interface at start-up to configure the part as desired. The host can then continue to use the CamControl interface, or it can use the UVC Control interface. The reason for this constraint is that as stated earlier, the UVC Control component acts as a virtual host - the other firmware components do not know of its existence. UVC Control modifies selected CamControl variables in order to control the AS0260.

No Coherency The AS0260 cannot guarantee coherency between the UVC Control interface and the CamControl interface. The value of variables on the UVC Control page may only reflect the last change made by the host (or the default value) - there is no immediate coherency between a UVC Control variable and its equivalent CamControl counterpart.

For example, suppose the host sets the desired scene brightness via the UVC Control page. The AS0260 then sets the target brightness via the CamControl page. Reading the current desired brightness on the UVC Control will return the original UVC value, not the actual value being used by the AS0260.

Note however that the converse is not true; changes to UVC Control variables are reflected in the CamControl Control variables, because the UVC Control feature acts as a virtual host - it modifies the CamControl variables itself.

The CamControl and UVC Control interfaces will be coherent (where applicable) on completion of a 'Refresh' command. If a UVC variable's coherency is not applicable this will be stated in the variable's description.

No Multivariable Atomic Changes All UVC control variable changes will be independent - there is no mechanism to 'group' a set of changes to variables together (as in the 'Refresh' command for the CamControl variables). If multiple UVC control variables are changed, there is no guarantee that all changes will occur on the same frame.

Indeterminate Change Latency The latency from when a UVC variable is changed, to when the change takes effect, is indeterminate, and is dependent on where within the frame the UVC change is made. The worse-case latency is two frames. The AS0260 implements the 'Wait For Event' command to allow the host to synchronize to the AS0260 frame timing, and to be sure that a UVC change has been applied.

UVC Control Interface

The following subsections detail the variables exposed by the UVC page. Each variable is documented in its own subsection, including its valid range and default value.

Note: The default value of most UVC control variables is dependent upon the underlying CamControl interface configuration, which is determined by the host at start-up.

All UVC control variables must indicate whether a change was accepted via the UVC_RESULT_STATUS variable (R0xCC24 or VAR(0x13,0x0024)). This variable is provided for diagnostic purposes only, to help track down why changes to UVC variables are being ignored. It does not form part of the UVC 1.1 standard.

Whenever a change is made to a UVC variable, the firmware will process the change and indicate the result of the change in UVC_RESULT_STATUS. Typically, a value of ENOERR will indicate the change was accepted. Any other value indicates the change was rejected. Table 1 shows the result status codes and their typical interpretations. Where the typical interpretation does not match Table 1, this will be indicated within the individual UVC variable documentation.

Table 14: UVC_Result_Status Codes

Value	Mnemonic	Typical Interpretation (each variable may re-interpret)
0x00	ENOERR	No error - change was accepted and acted upon
0x08	EACCES	Permission denied
0x09	EBUSY	Entity busy, cannot support operation
0x0C	EINVAL	Invalid argument
0x0E	ERANGE	Parameter out-of-range
0x0F	ENOSYS	Operation not supported

The host must be aware that UVC_RESULT_STATUS will always indicate the result of the last-changed UVC variable; the previous value of UVC_RESULT_STATUS will be overwritten by each subsequent change. If the host simultaneously modifies multiple UVC variables during the same frame, UVC_RESULT_STATUS will only indicate ENOERR if all changes were accepted. If any change is rejected, there is no mechanism for the host to determine which change it was. It is therefore strongly recommended that during development, the host only modify one UVC variable per-frame.

Auto-Exposure Mode

This mode enables or disables the auto-exposure function of the AS0260.

Variable	Name	Type	Default
R0xCC00 VAR(0x13,0x0000)	UVC_AE_MODE_CONTROL	BITFIELD8	Dependent upon CAM configuration

The UVC_AE_MODE_CONTROL variable controls the AS0260 auto-exposure algorithm. When auto-exposure is enabled, the AS0260 will automatically choose the appropriate frame interval, exposure time and gain to achieve the desired image brightness.

Image brightness is controlled through the UVC_BRIGHTNESS_CONTROL variable. When auto-exposure is disabled, the frame interval, exposure time and gain can be manually controlled through the UVC_FRAME_INTERVAL_CONTROL, UVC_EXPOSURE_TIME_ABSOLUTE_CONTROL and UVC_GAIN_CONTROL variables.

A valid setting for this control will only have one bit set (bits 0 to 3) - any other combination of bits will be rejected with EINVAL; the current setting will not be changed.

When the host switches from auto-exposure mode to manual-exposure mode, the SOC will update the values of UVC_FRAME_INTERVAL_CONTROL, UVC_EXPOSURE_TIME_ABSOLUTE_CONTROL and UVC_GAIN_CONTROL to reflect the current settings (the previous contents of these variables will be lost).

The range of valid values for this control is chosen to map directly to the UVC 1.1 standard. UVC combines auto- and manual iris control with auto- and manual exposure control - therefore there is some duplication within the bits supported. The assumption is that the USB bridge/ISP device will implement auto/manual iris control where required; this function is not supported by the AS0260.

Auto-Exposure Priority

Controls the operation of the frame-rate control function of the Auto-Exposure algorithm.

Variable	Name	Type	Default
0xCC02 VAR(0x13,0x0002)	UVC_AE_PRIORITY_CONTROL	UINT8	Dependent upon CAM configuration

The auto-exposure priority control is only active when auto-exposure is enabled (see Auto-Exposure Mode). When auto-exposure is disabled, changes to auto-exposure priority will be rejected with EACCES.

The AS0260 auto-exposure algorithm supports two variable frame-rate modes, controlled via the CAM_AET_AEMODE[CAM_AET_MODE_DISCRETE_FRAME_RATE] flag. The variable frame-rate mode selected when UVC_AE_PRIORITY_CONTROL is VARIABLE_FRAME_RATE depends upon the current CAM_AET_AEMODE setting. The minimum frame-rate is also controlled by the CamControl variables.

The assumption is that the host will configure the variable frame-rate support at device start-up, via the CamControl interface. If the configuration is such that variable frame-rate is disabled, attempts to set UVC_AE_PRIORITY_CONTROL to VARIABLE_FRAME_RATE will be rejected with ENOSYS.

Note that changing the auto-exposure priority setting to CONSTANT_FRAME_RATE will result in a 'restart' of the AE algorithm. This may result in a number of poorly exposed frames.

Exposure Time (Absolute)

Variable	Name	Type	Default
0xCC04 VAR(0x13,0x0004)	UVC_EXPOSURE_TIME_ABSOLUTE_CONTROL	UINT32	Undefined until AE mode disabled

The exposure time (absolute) control is only active when auto-exposure is disabled (see Auto-Exposure Mode). When auto-exposure is enabled, changes to exposure time (absolute) will be rejected with EBUSY.

The default value of UVC_EXPOSURE_TIME_ABSOLUTE_CONTROL is undefined when auto-exposure is enabled. In the event that auto-exposure is disabled, the default value is the active exposure time at the time the change is made.

The permitted range of exposure times is dependent upon the current CamControl configuration (pixel clock speed and sensor configuration). Attempts to set an exposure time outside this range will be accepted, but clamped to the current minimum or maximum. UVC_RESULT_STATUS will be set to ERANGE to indicate a clamp has occurred.

The UVC_MANUAL_EXPOSURE_CONFIG variable configures whether the exposure time can exceed the current frame interval (as set by UVC_FRAME_INTERVAL_CONTROL). The configuration variable also determines whether the host can set any exposure time, or only multiples of the power line frequency period (to avoid flicker). Note that any clamping of UVC_EXPOSURE_TIME_ABSOLUTE_CONTROL due to a frame interval limitation, or a flicker avoidance limitation, is silent; UVC_RESULT_STATUS will not be affected.

Backlight Compensation

Variable	Name	Type	Default
0xCC08 VAR(0x13,0x0008)	UVC_BACKLIGHT_COMPENSATION_CONTROL	UINT16	Dependent upon CAM configuration

The backlight compensation control helps the AS0260 auto-exposure (AE) algorithm correctly exposure the image for typical backlit scenes. The AS0260 has an AE 'window' that controls which parts of the scene should be considered by the auto-exposure algorithm. This window control is not part of the UVC interface (see CAM_STAT_AE_INITIAL_WINDOW_xxx). The AS0260 divides the AE window into a grid of 5 x 5 'zones'. Each zone has a weighting factor, which allows the host to prioritize the average brightness of some zones more than others (when being considered by the AE algorithm).

- When backlight compensation is disabled, the AS0260 will apply the same weighting to all zones - this is equivalent to taking the average brightness of the entire AE window.
- When backlight compensation level 1 is enabled, the AS0260 will apply a 'backlight compensation' map of weights to the zones, in order to prioritize the central zones over outlying zones (on the assumption that the region-of-interest is within the centre of the AE window). The zone weight map is static, configured by the AE_RULE_AE_WEIGHT_TABLE_N_M variables, and is not under UVC Control.
- When backlight compensation level 2 is enabled, the AS0260 employs an adaptive algorithm, which uses the average brightness of each zone to determine the zone weighting. Darker zones have more weighting. Level 2 uses medium strength adaptive weighting, where the zone weighting applied is a 50/50 blend of the static and adaptive weighting.
- Backlight compensation level 3 is similar to level 2, except only the adaptive weighting is applied to the zone weighting.
- Backlight compensation level 4 uses the static weight map as in level 1, but centres the AE 'window' to the central 9 zones. The average brightness of the 16 outlier zones is not calculated.

Changes to this control will be rejected with EACCES when auto-exposure mode is disabled.

Brightness

Variable	Name	Type	Default
R0xCC0A VAR(0x13,0x000A)	UVC_BRIGHTNESS_CONTROL	UINT16	Dependent upon CAM configuration

The brightness control is used to set the desired brightness of the scene when auto-exposure mode is enabled. When auto-exposure is disabled, any change will be rejected with EACCES.

The brightness of a scene is measured by the average luma of the pixels enclosed by the AE window The auto-exposure algorithm will attempt to keep the average luma of these pixels at the desired brightness (within configurable thresholds for smoothing the adaption rate).

Increasing the desired brightness of a scene may result in a change in frame-rate if UVC_AE_PRIORITY_CONTROL is set to VARIABLE_FRAME_RATE. Conversely, reducing the desired brightness may increase frame-rate.

The AS0260 does not maintain coherency between UVC_BRIGHTNESS_CONTROL and CAM_AET_TARGET_AVG_LUMA. Issue a Refresh command to force coherency.

Contrast

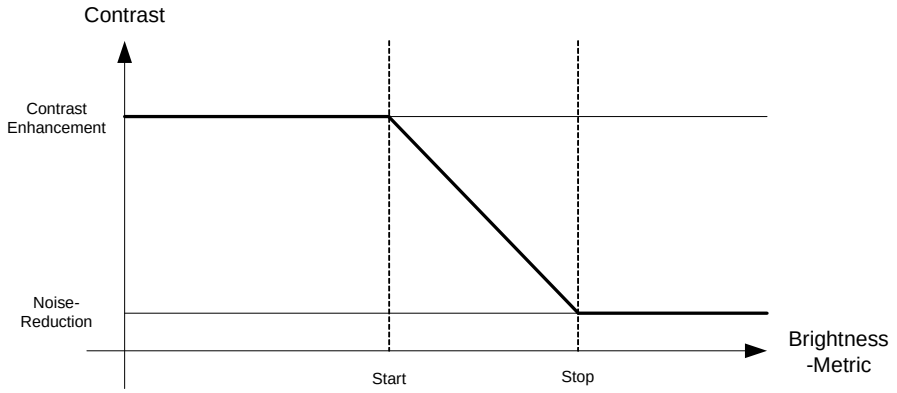
Variable	Name	Type	Default
R0xCC0C VAR(0x13,0x000C)	UVC_CONTRAST_CONTROL	UINT16 UFIXED5	32 (contrast gradient of 1.0)

The AS0260 supports a brightness-dependent contrast control, exposed through the Camcontrol interface. The actual amount of contrast applied (each frame) is dependent upon the estimated brightness of the scene. This allows the AS0260 to adapt to various scene lighting conditions.

The Camcontrol interface provides two contrast settings; one for bright scenes - the 'contrast enhancement' setting, and one for dark scenes - the 'noise reduction' setting. The AS0260 calculates a 'brightness-metric' each frame - this represents the estimated brightness of the scene (note the metric is inverse - the smaller the metric, the brighter the scene). The CAM Control interface supports contrast 'start' and 'stop' controls (specified in brightness-metric units) which indicate the knee points for the brightness-dependent contrast.

As shown in Figure 31, if the brightness-metric is below the start point, the Lowlight algorithm applies the 'contrast enhancement' setting. If the metric is above the stop point, the 'noise-reduction' setting is applied. If the metric is between the two points, the Lowlight algorithm calculates the applied contrast by linear interpolation between the 'contrast enhancement' and 'noise-reduction' settings, proportional to the brightness-metric.

Figure 31: Brightness-Dependent Contrast Control



The UVC contrast variable controls both the 'contrast enhancement' and 'noise-reduction' contrast settings. The 'contrast enhancement' value is set directly by UVC contrast. The 'noise-reduction' value is set proportionally according to the ratio between the CAM 'contrast enhancement' and 'noise-reduction' contrast variable settings.

In all cases, the contrast value represents the gradient of the contrast adjustment curve, measured at the target brightness point (as controlled by UVC_BRIGHTNESS_CONTROL). The AS0260 supports a range of gradients from 0.5 to 2.0; UVC represents this as a contrast range from 16 (0.5) to 64 (2.0). It shows the range of contrasts that can be selected. Note the contrast gradient is measured at the inflection point in the curves - this inflection point is also dependent upon the brightness-metric.

Note that the AS0260 may not support every code within the allowed UVC contrast range; the AS0260 will round to the nearest code.

Note that the automatic contrast curve calculation as supported by the CAM Control interface can be disabled. In this event, attempts to change the UVC contrast will be rejected with EACCES.

The AS0260 does not maintain coherency between UVC_CONTRAST_CONTROL and the CAM Control variable equivalents.

Gain

Variable	Name	Type	Default
R0xCC0E VAR(0x13,0x000E)	UVC_GAIN_CONTROL	UINT16 UFIXED 5	Dependent on gain applied when AE mode is disabled.

The gain control determines the amount of gain applied by the sensor and AS0260 when auto-exposure mode is disabled. If auto-exposure is enabled, any changes will be rejected with EBUSY.

UVC_GAIN_CONTROL will not reflect the current gain applied when auto-exposure mode is enabled. When auto-exposure is disabled with UVC_AUTO_EXPOSURE_MODE, this variable will reflect the active gain at that time.

The permitted range of gains is dependent upon the current CAM Control sensor configuration. Attempts to set a gain outside this range will be accepted, but clamped to the current minimum or maximum. UVC_RESULT_STATUS will be set to ERANGE to indicate a clamp has occurred.

Power Line Frequency Control

Variable	Name	Type	Default
R0xCC03 VAR(0x13,0x0003)	UVC_POWER_LINE_FREQUENCY_CONTROL	UINT8	Dependent upon CAM configuration

The power line frequency control specifies the local power line frequency. This allows the auto-exposure algorithm to limit exposure time to multiples of this frequency, in order to avoid image flicker.

Note the AS0260 does not support the UVC 'Disabled' setting - flicker avoidance cannot be disabled for all lighting levels. However, this value will not be rejected in order to conform to the UVC 1.1 standard. The AS0260 will continue using the last-set value, and the variable will continue to read-back the last-set value.

Note that the UVC_FLICKER_AVOIDANCE_CONFIG configuration variable allows the host to enable an 'outdoor' mode, which permits exposure times that are less than the flicker frequency.

The AS0260 does not maintain coherency between UVC_POWER_LINE_FREQUENCY_CONTROL and the CAM Control variable equivalent. Issue a Refresh command to force coherency.

Hue Control

Variable	Name	Type	Default
R0xCC10 VAR(0x13,0x0010)	UVC_HUE_CONTROL	INT16	Dependent upon CAM configuration

The hue control sets the amount of hue adjustment (rotation) applied by the AS0260. Hue adjustment is global—it affects all pixels in the image.

The AS0260 does not support every code within the permitted range; the AS0260 will round the set value to the nearest supported code.

Saturation Control

Variable	Name	Type	Default
R0xCC12 VAR(0x13,0x0012)	UVC_SATURATION_CONTROL	UINT16 UFIXED7	128 (unity)

The AS0260 supports a brightness-dependent saturation control, exposed through the CAM control interface. The actual amount of saturation applied (each frame) is dependent upon the estimated brightness of the scene. This allows the AS0260 to adapt to various scene lighting conditions. This is very similar to the contrast control adaption.

A lowlight saturation value of zero means no color-correction. A UVC saturation of zero means grey-scale; a monochrome image. In both cases, a value of 128 means 'unity' - the CCM will not be altered. In both cases, a value above 128 results in a 'boosted' CCM.

Sharpness Control

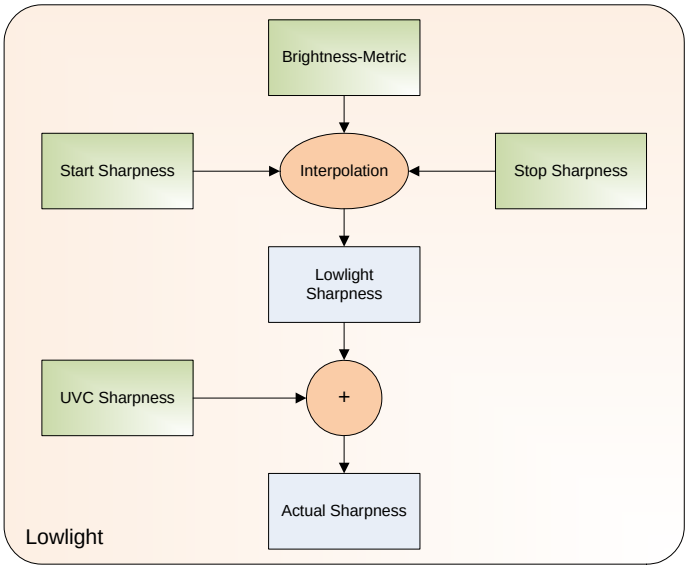
Controls the amount of sharpening adjustment applied to the image by the AS0260.

Variable	Name	Type	Default
R0xCC14 VAR(0x13,0x0014)	UVC_SHARPNESS_CONTROL	INT16	0 (no sharpening adjustment)

The AS0260 supports a brightness-dependent sharpness control, exposed through the CAM control interface. The actual amount of sharpening applied (each frame) is dependent upon the estimated brightness of the scene. This allows the AS0260 to adapt to various scene lighting conditions. This is very similar to the contrast control adaption - see section 6.7.

The CAM control interface provides two sharpness controls; one for brighter scenes, and one for dark scenes. These correspond to the brightness-metric knee points. If the brightness-metric is below the start point, the Lowlight algorithm applies the 'bright' sharpness. If the metric is above the stop point, the 'dark' setting is applied. If the metric is between the two points, the Lowlight algorithm calculates the sharpness by linear interpolation between the 'bright' and 'dark' settings, proportional to the brightness-metric. The result of the Lowlight calculation is termed the 'lowlight' sharpness.

Figure 32: UVC Sharpness Control Flow



The UVC sharpness control is implemented as a relative adjustment to the 'lowlight' sharpness, as shown in Figure 2. For example, Table 15 shows how the UVC sharpness control affects a nominal lowlight sharpness of three. The actual sharpness value used is limited to the range 0 to +7. To ensure no sharpening is applied, set UVC sharpness to -7.

Table 15: UVC Adjustment of Lowlight Sharpness

Lowlight Sharpness	UVC Sharpness	Actual Sharpness
3	0	+3
3	+3	+6
3	+7	+7
3	-3	0
3	-7	0

The UVC sharpness control is effectively a clamp on the adaptive sharpness range that the SOC can apply. The clamp ranges are shown in Table 16.

Table 16: UVC Sharpness vs. Adaptive Sharpness Range

UVC sharpness setting	Adaptive sharpness range
-7	0
-6	0 to 1
-5	0 to 2
-4	0 to 3
-3	0 to 4
-2	0 to 5
-1	0 to 6
0	0 to 7

Table 16: UVC Sharpness vs. Adaptive Sharpness Range

UVC sharpness setting	Adaptive sharpness range
1	1 to 7
2	2 to 7
3	3 to 7
4	4 to 7
5	5 to 7
6	6 to 7
7	7

Gamma Control

Controls the amount of gamma correction applied by the AS0260

Variable	Name	Type	Default
R0xCC16 VAR(0x13,0x0016)	UVC_GAMMA_CONTROL	UINT16	Dependent upon CAM configuration

The gamma control adjusts the amount of gamma correction applied by the SOC. The value of the control is expressed in units multiplied by 100. Note that the gamma value represents the 'display' gamma - this is the gamma of the final display terminal. The AS0260 applies the reciprocal of the display gamma.

Note that the automatic gamma curve calculation as supported by the Camcontrol interface can be disabled. In this event, attempts to change the UVC gamma will be rejected with EACCES.

The AS0260 does not maintain coherency between UVC_GAMMA_CONTROL and the Camcontrol variable equivalent. Issue a Refresh command to force coherency.

White Balance Temperature Control

Controls the white balance temperature adjustment applied by the AS0260 (when auto-white-balance is disabled)

Variable	Name	Type	Default
R0xCC18 VAR(0x13,0x0018)	UVC_WHITE_BALANCE_TEMPERATURE_CONTROL	UINT16	Dependent on color temperature when AWB mode is disabled

The white balance temperature control sets the white-balance temperature applied by the AS0260 when the auto-white-balance (AWB) algorithm is disabled. Attempts to set this control when AWB is enabled will be rejected with EBUSY.

The white-balance temperature is used by the AS0260 to calculate the 'ideal' color-correction matrix and to calculate the ratios of red and blue gains to apply to white-balance the scene. When AWB is enabled, the AS0260 calculates the white-balance temperature itself. UVC_WHITE_BALANCE_TEMPERATURE_AUTO_CONTROL is used to disable AWB.

The permitted range of color temperatures is dependent upon the current Camcontrol configuration. Attempts to set a color temperature outside this range will be accepted, but clamped to the current minimum or maximum. UVC_RESULT_STATUS will be set to ERANGE to indicate a clamp has occurred.

The AS0260 does not maintain coherency between UVC_WHITE_BALANCE_TEMPERATURE_CONTROL and the Camcontrol variable equivalent. Issue a Refresh command to force coherency.

Note: The UVC Control interface does not support white balance component control. However, this feature is supported by the CamControl interface.

White Balance Temperature Auto Control

Variable	Name	Type	Default
R0xCC01 VAR(0x13,0x0001)	UVC_WHITE_BALANCE_TEMPERATURE_AUTO_CONTROL	UINT8	Dependent upon CAM configuration

The white balance temperature auto control enables or disables the AS0260 auto-white-balance (AWB) algorithm.

When the host switches from auto-white-balance mode to manual white-balance mode, the AS0260 will update the value of UVC_WHITE_BALANCE_TEMPERATURE_CONTROL to reflect the current setting (the previous contents of this variable will be lost).

Frame Interval

Controls the frame-rate (when auto-exposure mode is disabled)

Variable	Name	Type	Default
R0xCC1C VAR(0x13,0x001C)	UVC_FRAME_INTERVAL_CONTROL	UINT32	The frame-interval when AE is disabled

The frame interval control determines the frame-rate when auto-exposure mode is disabled. If auto-exposure mode is enabled, any change will be rejected with EBUSY.

The permitted range of frame-rates is dependent upon the current Camcontrol configuration (pixel clock speed and sensor configuration). Attempts to set a frame-interval that is outside this range will be accepted but clamped to the permitted minimum or maximum. UVC_RESULT_STATUS will be set to ERANGE to indicate a clamp has occurred.

UVC Configuration and Status

The following subsections detail the UVC configuration and status variables. These do not form part of the UVC standard. They are provided to allow the implementation (or meaning) of selected UVC controls to be configured.

Manual Exposure Configuration

Variable	Name	Type	Default
R0xCC20 VAR(0x13,0x0020)	UVC_MANUAL_EXPOSURE_CONFIG	BITFIELD8	0x0 (fixed frame rate, no flicker avoidance)

The UVC_FRAME_INTERVAL_CONTROL variable (frame-interval) allows the host to set the frame-rate, and the UVC_EXPOSURE_TIME_ABSOLUTE_CONTROL (exposure time) variable allows the host to directly control the exposure time (both provided auto-exposure mode is disabled). However, in some cases it may be appropriate for the exposure time to be limited:

- Such that exposure cannot exceed the current frame interval
- Such that image flicker is avoided by constraining exposure to multiples of the power-line frequency period

If DISABLE_FIXED_FRAME_RATE (bit 0) is clear, any change in exposure time will be rejected if it exceeds the current value of frame-interval. If DISABLE_FIXED_-FRAME_RATE is set, the AS0260 will accept any value of exposure time (subject to its permitted range, and the state of bit 1). Note also that if DISABLE_FIXED_FRAME_RATE is clear, if frame-interval is reduced to a value below the current exposure time value, the AS0260 will automatically reduce the exposure time.

If ENABLE_FLICKER_AVOIDANCE (bit 1) is set, any change in exposure time will be rounded-down to the nearest next multiple of the power-line frequency period. No error will be reported. If ENABLE_FLICKER_AVOIDANCE is clear, the AS0260 will accept any value of exposure time (subject to its permitted range).

Note: The default configuration is to restrict exposure time such that it will not exceed the frame-interval - this conforms to the UVC 1.1 specification.

Flicker Avoidance Configuration

Configures the AS0260 flicker-avoidance algorithm.

Variable	Name	Type	Default
0xCC21 VAR(0x13,0x0021)	UVC_FLICKER_AVOIDANCE_CONFIG	BITFIELD8	Dependent upon CAM configuration

The flicker-avoidance algorithm can operate in two modes:

- Flicker-avoidance: exposure time is restricted to multiples of the flicker period, regardless of the scene brightness.
- Flicker-avoidance with outdoor override: exposure time is restricted to multiples of the flicker period, unless the scene brightness is typical for an 'outdoor' scene (where power line frequency flicker artefacts are not expected). In these brighter scenes, AE can choose the most appropriate exposure time.

The flicker avoidance configuration variable allows the host to select between these two modes. Note that if the outdoor mode is disabled, bright scenes may result in an overexposed image, as the minimum exposure time may not prevent saturation.

Multi-Camera Sync

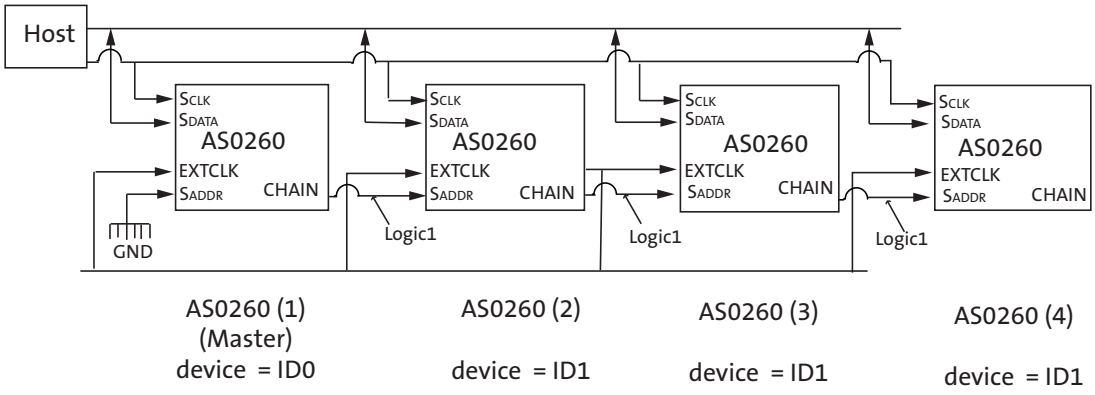
The AS0260 supports more than one device to be connected in a “daisy-chain” type configuration. One of the devices will act as the master and the remainder will be slaves.

A typical connection diagram is shown inFigure 33. All of the AS0260 that are to communicate are:

- Connected in a daisy-chain using SADDR as an input and CHAIN as an output.
- Clocked from a common clock source
- Controlled from a single master, presumed to be under software control of a host system.

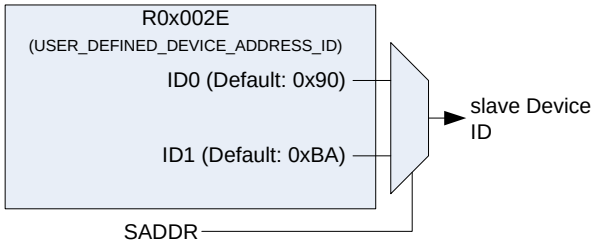
When only two AS0260 image sensors need to be synchronized, as is the case in a 3D camera application, the AS0260 offers an additional feature of synchronized auto exposure and auto white balance. In this mode, the slave device mimics the exposure and white balance settings of the master device.

Figure 33: Multi-Camera Connection



SADDR is normally used as a static input that selects between two slave device addresses (See Figure 34. In order to implement the multi-sync function this input now has additional functionality that does not interfere with its use as device address selection.

Figure 34: Normal Use of SADDR



There is a single register to control this function, named CHAIN_CONTROL (R0x31FC). This register is controlled by the host. The register field assignment is shown in Table 17.

Table 17: CHAIN_CONTROL Register

Bit	Name	Default	Description
15	chain_enable	0	0: multi-camera daisy-chain communication function is disabled. 1: multi-camera daisy-chain communication function is enabled. The result of toggling this bit while the sensor is streaming is UNDEFINED.
14	sync_enable	0	0: multi_sync function is disabled. 1: multi-sync function is enabled. The result of toggling this bit while the sensor is streaming is UNDEFINED.
13	master	0	0: this node is not the master. 1: this node is the master. The result of toggling this bit while the sensor is streaming is UNDEFINED.
12	RESERVED		
11:8	position	0	A unique value assigned to each device in the daisy-chain. The device furthest from the master is assigned a position value of 0. The next device is assigned a position value of 1. For N devices in a daisy-chain, the master is assigned a position value of N-1. The result of toggling this bit while the sensor is streaming is UNDEFINED.
7:0	RESERVED		

Configuration

Before the multi-sync function can be used, each AS0260 in the daisy-chain must be configured. This process is performed by the host with no involvement from AS0260 firmware. Configuration involves assigning a unique slave address to each AS0260 and configuring the CHAIN_CONTROL register on each AS0260.

After reset (before configuration) the master AS0260 has its SADDR input wired to '0' and all other AS0260 in the daisy-chain have their SADDR inputs driven to '1'. Therefore, AS0260 Master will respond to slave address ID0 (associated with SADDR = 0) and all the other AS0260 in the daisy-chain will respond simultaneously to slave address ID1. Each AS0260 has its CHAIN pin configured as an input. This situation is shown in Figure 33. The host configures each AS0260 in sequence, starting with the master and ending with the farthest slave in the daisy-chain:

- AS0260(1) Master: The host uses slave address ID0 (associated with SADDR = 0) and therefore accesses registers on AS0260(1) (the master). It writes to register (R0x002E) to change the slave addresses associated with ID0 and ID1 on this device to a single, new, unique value; call it ID-AS0260(1). It then writes (using AS0260(1) to register PAD_CONTROL (R0x0032) to configure CHAIN as an output. Finally, it writes (using AS0260(1)) to the CHAIN_CONTROL register to set chain_enable =1, sync_enable=1, master=1 and position = N – 1 (where there are N devices in the daisy-chain). The effect of enabling TMS as an output is to drive the TMS output low.
- AS0260(2): This AS0260 now has SADDR=0 and so will respond to slave address ID0. The host configures this in the same way as AS0260(1) with the exceptions that it assigns ID-AS0260(2), sets master=0 and position = N-2 (where there are N devices in the daisy-chain). As before, the effect of enabling CHAIN as an output is to drive the CHAIN output low.
- AS0260(3): As for AS0260(2): assign ID-AS0260(3), master=0, position = N-3
- AS0260(4): As for AS0260(2): assign ID-AS0260(4), master=0, position = N-4

Theory Of Operation

When multiple AS0260 devices have been connected and configured as described above, the multi-sync function operates as follows:

When the master device is placed in streaming mode (as the result of a mode change initiated by the host) it generates an event on its CHAIN output. It then delays its own streaming until the last of the slave devices has received an event signal.

When a slave device is placed in streaming mode (as the result of a mode change initiated by the host) it delays streaming until it has received an event on its SADDR input.

Each slave in the daisy-chain propagates events received on its input. Each slave uses its local value of "position" to delay its respond to an event. This allows an event propagated down the daisy-chain to be acted upon simultaneously by all devices in the daisy-chain.

Using Multi-Sync

The host can use the normal mechanism to configure the AS0260 and set them streaming. It can do this in any order provided that it sets the master streaming last.

It is desirable (but not essential) for the master to be taken out of streaming mode first (by using a host command).

At the time that the AS0260 are placed in streaming mode, all AS0260 must have the same integration time The recommended mechanism is:

1) Boot each device into standby by enabling 'host-config' mode.

2) Reconfigure each device.

3) Wake each device and commence streaming using the Leave Standby command.

The AS0260 need not maintain the same integration time once they are streaming.

All the AS0260 must be operated with the same configuration (image size, output format, PLL bypassed and frame timing). Any time that the configuration is to be changed, all AS0260 must be taken out of streaming mode (using host command), reconfigured, then placed back in streaming mode (master last). This will allow the output data to remain in synchronisation.

Clocking

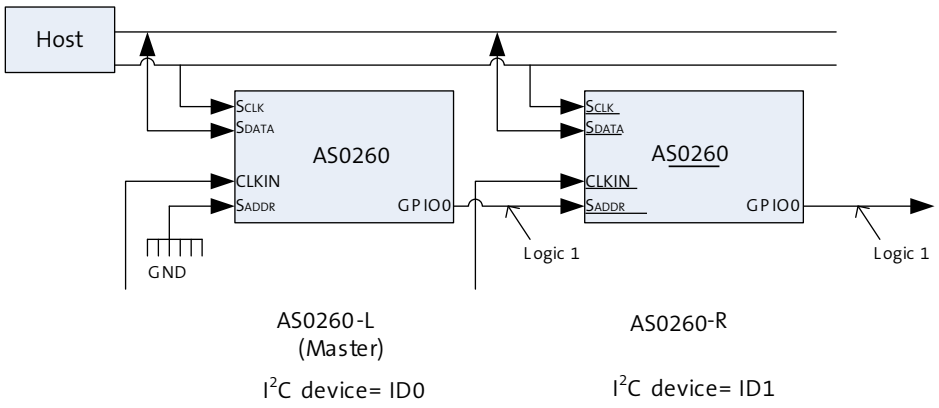
The multi-sync mechanism requires that all AS0260 devices in the daisy-chain are operated synchronously on the same input clock. This constraint is imposed in order to allow the event codes to be propagated synchronously from the master through to each slave.

Once this constraint has been met, the AS0260 devices are required to operate in exact synchronisation (such that a PIXCLK, FRAME_VALID and LINE_VALID out of one AS0260 is valid for all AS0260 in the daisy-chain). In this case, the AS0260 internal PLL must be bypassed (and the AS0260 must be using parallel output data).

AptiSync2 (Auto-Sync)

An additional control is available to synchronize the auto exposure and auto white balance functions of two image sensors. No additional hardware connections are needed to support this control.

Figure 35: AptiSync2 Hardware Connections



This auto-sync mode is enabled through the following control register. When enabled the master device's GPIO/CHAIN pin and slave device's SADDR pin are used for inter-sensor communication (UDI).

Table 18: AUTOSYNC_MODE Register

Bit	Name	Default	Description
2	Enable	0	0: Auto-sync function is disabled. 1: Auto-sync communication function is enabled. The result of toggling this bit while the sensor is streaming is UNDEFINED.
1	Slave	0	0: Device is master 1: Device is slave The result of toggling this bit while the sensor is streaming is UNDEFINED.
0	UDI	0	0: GPIO/CHAIN function assigned to GPIO/CHAIN pin 1: UDI function assigned to GPIO/CHAIN pin The result of toggling this bit while the sensor is streaming is UNDEFINED.

Hardware Functions

Two-Wire Serial Interface

The two-wire serial interface bus enables read and write access to control and status registers and variables within the AS0260.

The interface protocol uses a master/slave model in which a master controls one or more slave devices. The AS0260 always operates in slave mode. The host (master) generates a clock (SCLK) that is an input to the AS0260 and is used to synchronize transfers. Data is transferred between the master and the slave on a bidirectional signal (SDATA).

Protocol

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

1. a (repeated) start condition
2. a slave address/data direction byte
3. a 16-bit register address (8-bit addresses are not supported)
4. an (a no) acknowledge bit
5. a 16-bit data transfer (8-bit data transfers are not supported)
6. 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.

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.

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

Data is transferred serially, 8 bits at a time, with the most significant bit (MSB) transmitted first. Each byte of data is followed by an acknowledge bit or a no-acknowledge bit. This data transfer mechanism is used for the slave address/data direction byte and for message bytes. One data bit is transferred during each SCLK clock period. SDATA can change when SCLK is LOW and must be stable while SCLK is HIGH.

Slave Address

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. If the SADDR signal is driven LOW, then addresses used by the AS0260 are R0x090 (write address) and R0x091 (read address). If the SADDR signal is driven HIGH, then addresses used by the AS0260 are R0x0BA (write address) and R0x0BB (read address).

Message Byte

Message bytes are used for sending register addresses and register write data to the slave device and for retrieving register read data. The protocol used is outside the scope of the two-wire serial interface specification.

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.

Stop Condition

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

Typical Serial Transfer

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 a 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 acknowledge bit at the end of the sequence. After 8 bits have been transferred, the slave’s internal register address is automatically incremented, so that the next 8 bits are written to the next register address. The master stops writing by generating a (re)start or stop condition.

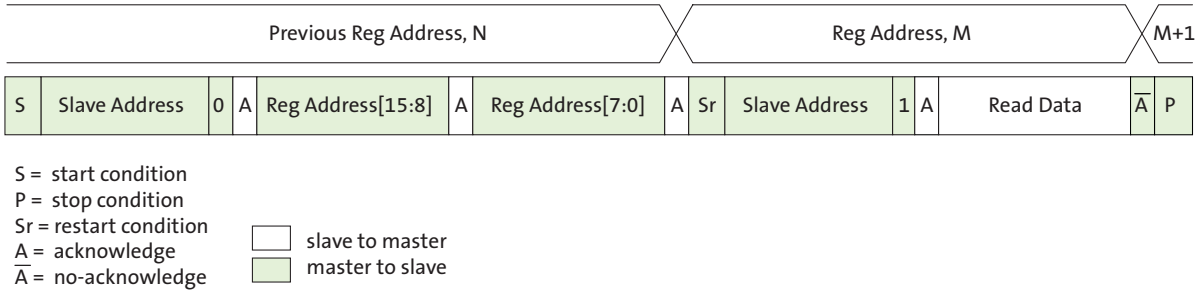
If the request was a read, the master sends the 8-bit write slave address/data direction byte and 16-bit register address, just as in the write request. The master then generates a (re)start condition and the 8-bit read slave address/data direction byte, and clocks out the register data, 8 bits at a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

Note: If a customer is using direct memory writes (XDMA), AND the first write ends on an odd address boundary AND the second write starts on an even address boundary AND the first write is not terminated by a STOP, the write data can become corrupted. To avoid this, ensure that a serial write is terminated by a STOP.

Single Read from Random Location

This sequence (see Figure 36) 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 36 shows how the internal register address maintained by the AS0260 is loaded and incremented as the sequence proceeds.

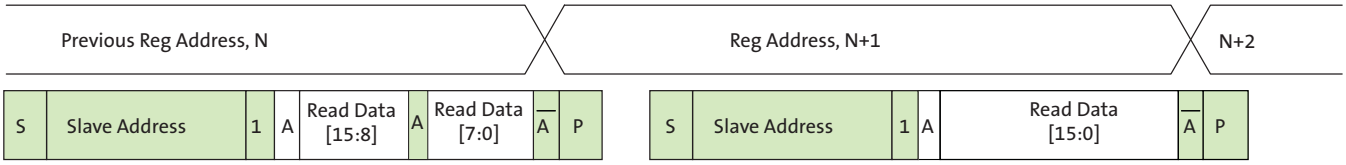
Figure 36: Single Read from Random Location



Single Read from Current Location

This sequence (Figure 37) performs a read using the current value of the AS0260 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.

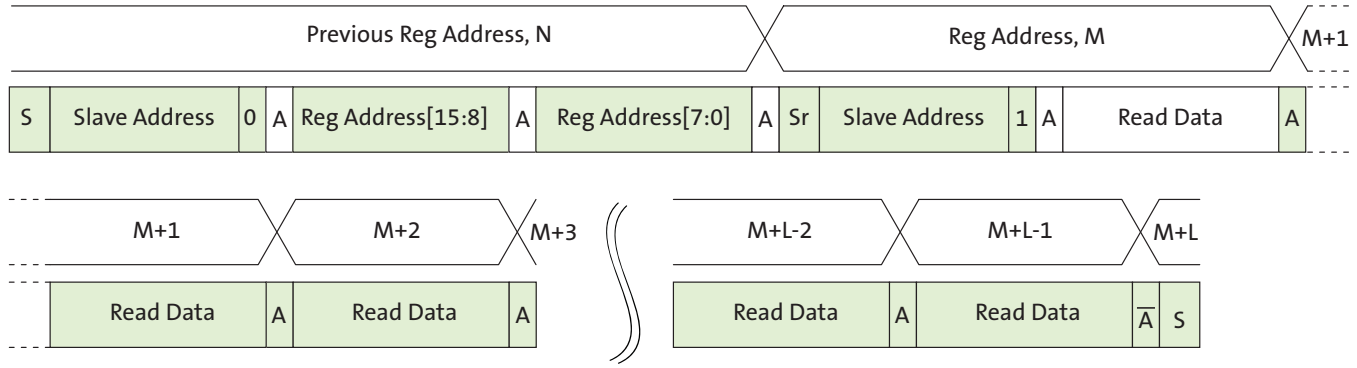
Figure 37: Single Read from Current Location



Sequential Read, Start from Random Location

This sequence (Figure 38) starts in the same way as the single read from random location (Figure 36). 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.

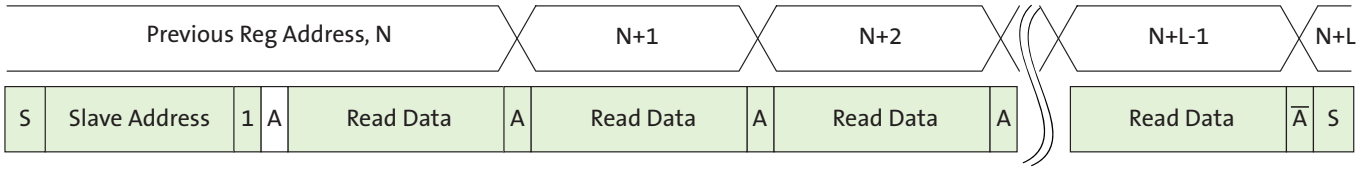
Figure 38: Sequential Read, Start from Random Location



Sequential Read, Start from Current Location

This sequence (Figure 39) starts in the same way as the single read from current 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.

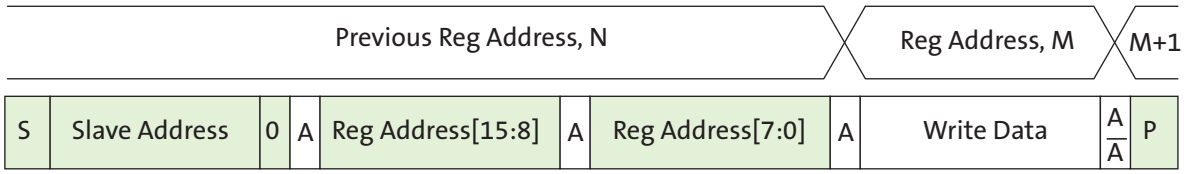
Figure 39: Sequential Read, Start from Current Location



Single Write to Random Location

This sequence (Figure 40) 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.

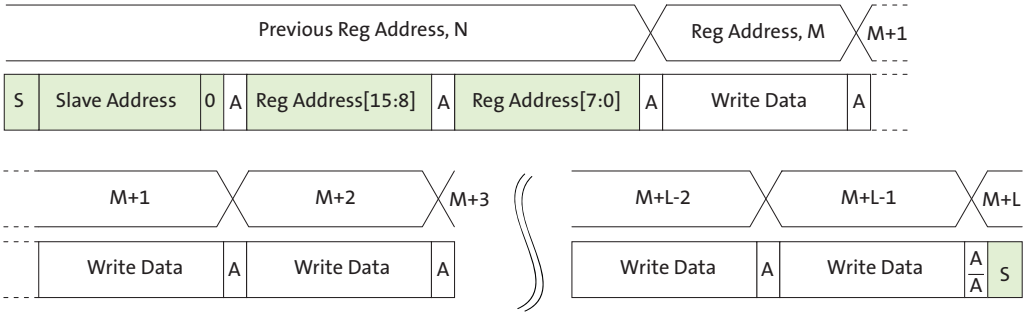
Figure 40: Single Write to Random Location



Sequential Write, Start at Random Location

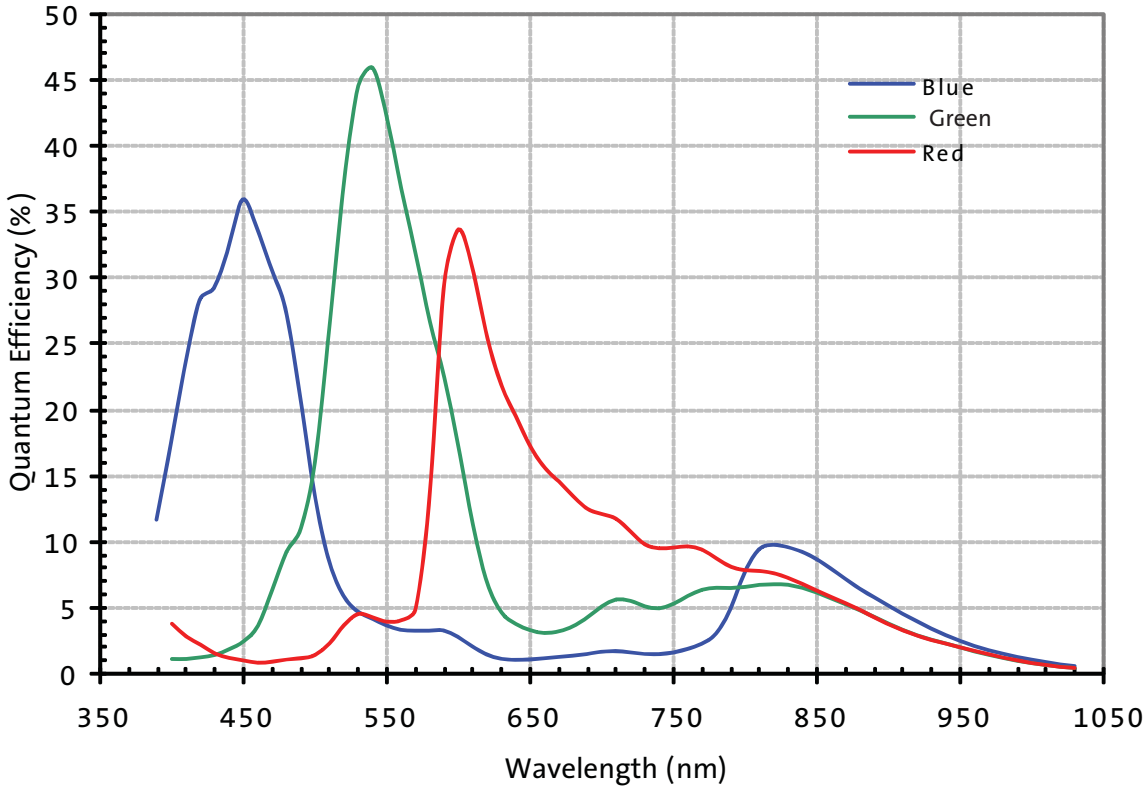
This sequence (Figure 41) starts in the same way as the single write to random location (Figure 40). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit, and continues to perform byte writes until L bytes have been written. The write is terminated by the master generating a stop condition.

Figure 41: Sequential Write, Start at Random Location



Spectral Characteristics

Figure 42: Quantum Efficiency vs. Wavelength



Chief Ray Angle

Table 19: Chief Ray Angle Characteristics

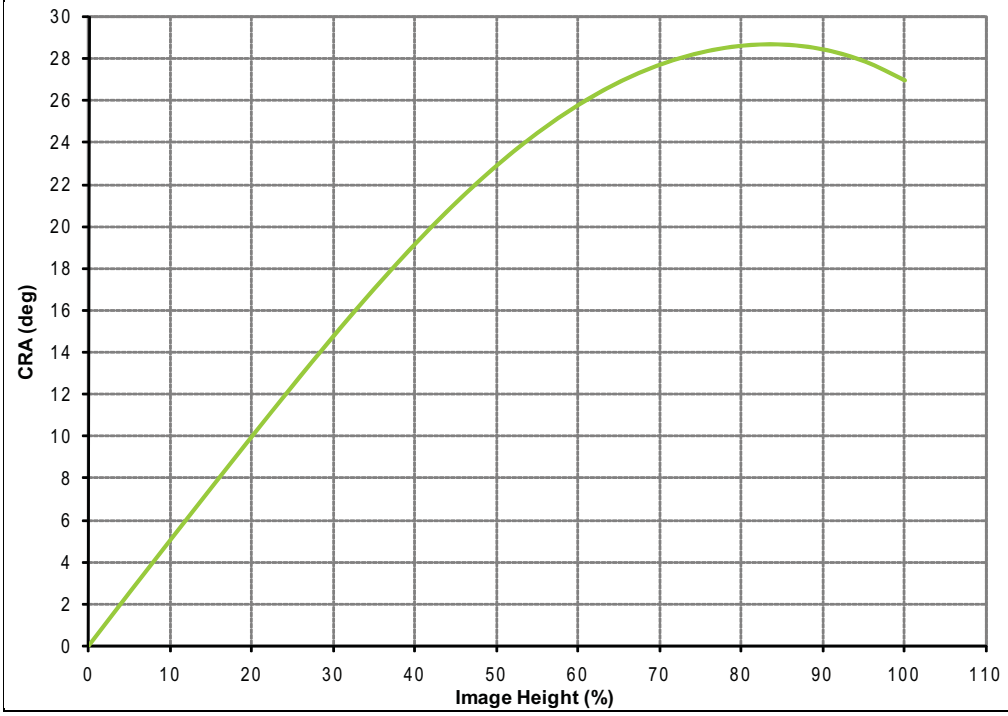


Image Height		CRA
(%)	(mm)	(deg)
0	0	0
5	0.077	2.54
10	0.154	5.04
15	0.231	7.52
20	0.308	9.98
25	0.386	12.40
30	0.463	14.76
35	0.540	17.02
40	0.617	19.15
45	0.694	21.12
50	0.771	22.89
55	0.848	24.45
60	0.925	25.78
65	1.002	26.87
70	1.079	27.70
75	1.157	28.28
80	1.234	28.61
85	1.311	28.66
90	1.388	28.43
95	1.465	27.88
100	1.542	26.95

CSP Package Details

AS0260 sensor is also available in chip scale package (CSP) and this section provides the relevant CSP package details necessary for the optical design of camera system.

Table 20: Package Dimension

Parameter	Symbol	Millimeters			Inches		
		Nominal	Min	Max	Nominal	Min	Max
Package Body Dimension X	A	6.005	5.97955	6.02955	0.23640	0.23542	0.23738
Package Body Dimension Y	B	4.158	4.13255	4.18255	0.16368	0.16270	0.16467
Package Height	C	0.710	0.655	0.765	0.02795	0.02579	0.03012
Cavity wall height	C4	0.0410	0.0370	0.0450	0.00161	0.00146	0.00177
Cavity wall + epoxy thickness glass to the wafer bonding top point)	C5	0.0435	0.0385	0.0485	0.00171	0.00152	0.00191
Glass Thickness	C3	0.400	0.390	0.410	0.01575	0.01535	0.01614
Package Body Thickness	C2	0.570	0.535	0.605	0.02244	0.02106	0.02382
Ball Height	C1	0.140	0.110	0.170	0.00551	0.00433	0.00669
Ball Diameter	D	0.280	0.250	0.310	0.01102	0.00984	0.01220
Total Ball Count		54					
Ball Count X axis	N1	9					
Ball Count Y axis	N2	6					
UBM	U	0.310	0.300	0.320	0.0122	0.01181	0.01260
Pins Pitch X axis	J1	0.620	0.610	0.630	0.02441	0.02402	0.02480
Pins Pitch Y axis	J2	0.620	0.610	0.630	0.02441	0.02402	0.02480
BGA ball center to package center offset in X-direction	X	0	-0.025	0.025	0	-0.00098	0.00098
BGA ball center to package center offset in Y-direction	Y	0	-0.025	0.025	0	-0.00098	0.00098
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	0.522	0.492	0.552	0.02056	0.01938	0.02174
Edge to Ball Center Distance along Y	S2	0.529	0.499	0.559	0.02082	0.01964	0.02200

Figure 43: CSP Mechanical Drawing

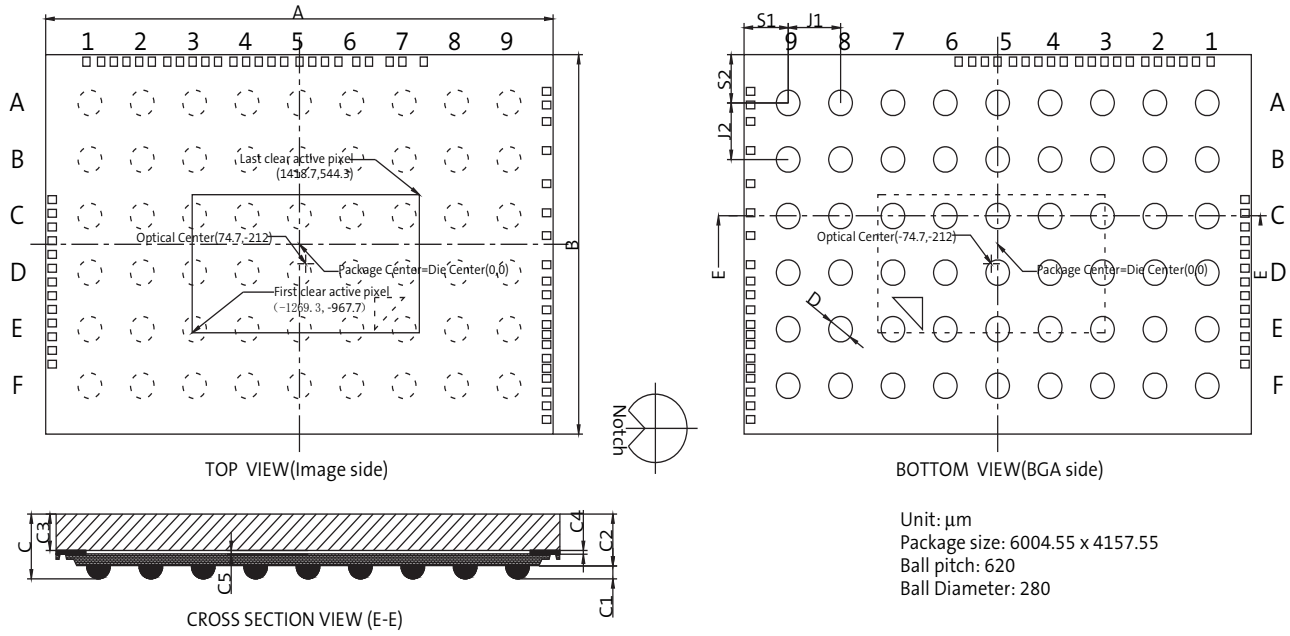


Table 21: Ball Matrix

	1	2	3	4	5	6	7	8	9
A	DGND	DOUT5	DGND	DOUT0	GPIO1	LINE_VALID	REG_IN0	DGND	DGND
B	DGND	DOUTT6	DOUT4	VDD_IO	GPIO0	DGND	REG_OUT	CLK_P	CLK_N
C	GPIO2	DOUT7	VDD_IO	DOUT3	DOUT1	FRAME_VALID	REG_FB	DATA_2N	DATA_2P
D	VDD_IO	PIXCLK	VDD	DOUT2	DGND	VDD_IO	VDD	DATA_N	DATA_P
E	SDATA	EXTCLK	DGND	DGND	SHUTDOWN	AGND	DGND	VDD_PHY	VPP
F	TRST_BAR	Reserved	SCLK	RESET_BAR	SADDR	AGND	AGND	VAA_PIX	VAA

Notes: 1. TRST_BAR connects to DGND for normal operation.
2. Reserved pins must be left floating.

Electrical Specifications

Caution **Stresses above those listed in Table 22 may cause permanent damage to the device.**

Table 22: Absolute Maximum Ratings

Symbol	Parameter	Rating		Unit
		Min	Max	
$V_{DD\_IO\_MAX}$	I/O digital voltage	–0.3	4.0	V
V_{AA_MAX}	Analog voltage	–0.3	4.0	V
$V_{AA\_PIX\_MAX}$	Analog pixel supply voltage	–0.3	4.0	V
REG_IN0_MAX	Digital supply voltage	–0.3	2.4	V
$V_{DD\_PHY\_MAX}$	PHY supply voltage	–0.3	2.4	V
V_{IN}	DC input voltage	–0.3	V_{DD_IO} + 0.3	V
I_{IN}	Transient input current (0.5 sec. duration)	–	150	mA
T_{OP}	Operating temperature (measure at junction)	–30	70	°C
T_{STG} [1]	Storage temperature	–40	85	°C

Notes: 1. This is a stress rating only, and functional operation of the device at these or any other conditions above those indicated in the product specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Recommended Operating Conditions

Table 23: Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units
V_{DD_IO}	I/O digital voltage	2.5	2.8	3.1	V
		1.7	1.8	1.95	V
V_{AA}	Analog voltage	2.5	2.8	3.1	V
V_{AA_PIX}	Pixel supply voltage	2.5	2.8	3.1	V
REG_IN0	Digital supply voltage	1.7	1.8	1.95	V
V_{DD_PHY}	PHY supply voltage	1.7	1.8	1.95	V
T_J	Operating temperature (at junction)	–30	55	70	°C

Table 24: DC Electrical Characteristics

Symbol	Parameter	Condition	Min	Max	Unit
V_{IH}	Input HIGH voltage		V_{DD_IO} * 0.7	V_{DD_IO} + 0.3	V
V_{IL}	Input LOW voltage		–0.3	V_{DD_IO} * 0.3	V
I_{IN}	Input leakage current	V_{IN} = 0V or V_{IN} = V_{DD_IO}		10	µA
V_{OH}	Output HIGH voltage	V_{DD_IO} = 1.8V, I_{OH} = 2mA	V_{DD_IO} – 0.3		V
V_{OL}	Output LOW voltage	V_{DD_IO} = 1.8V, I_{OH} = 2mA	–	0.4	V

Table 25: Operating Current Consumption (Parallel)

Default Setup Conditions: f_{EXTCLK}= 24 Mhz, f_{PIXCLK} = 96 Mhz, V_{AA} = V_{AA_PIX} = V_{DD_IO} = 2.8V, V_{DD_PHY} = REG_IN0 = 1.8V, Tj = 25°C unless otherwise stated

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{AA}	Analog supply voltage		2.5	2.8	3.1	V
V_{AA_PIX}	Pixel supply voltage		2.5	2.8	3.1	V
V_{DD_PHY}	PHY supply voltage		1.7	1.8	1.95	V
REG_IN0	Digital supply voltage		1.7	1.8	1.95	V
V_{DD_IO}	Digital IO supply voltage	V_{DD_IO} = 2.8V	2.5	2.8	3.1	V
		V_{DD_IO} = 1.8V	1.7	1.8	1.95	V
I_{AA}	Analog supply current	1080p Full resolution 30 fps				mA
		720p, 30 fps				mA
		VGA, 60 fps				mA
I_{AA_PIX}	Pixel supply current	1080p Full resolution 30 fps				mA
		720p, 30 fps				mA
		VGA, 60 fps				mA
I_{REG_IN0}	Digital supply current	1080p Full resolution 30 fps				mA
		720p, 30 fps				mA
		VGA, 60 fps				mA
I_{DD_PHY}	PHY supply current	1080p Full resolution 30 fps				mA
		720p, 30 fps				mA
		VGA, 60 fps				mA
Total power consumption		1080p Full resolution 30 fps		255		mW
		720p, 30 fps		215		mW
		VGA, 60 fps				mW

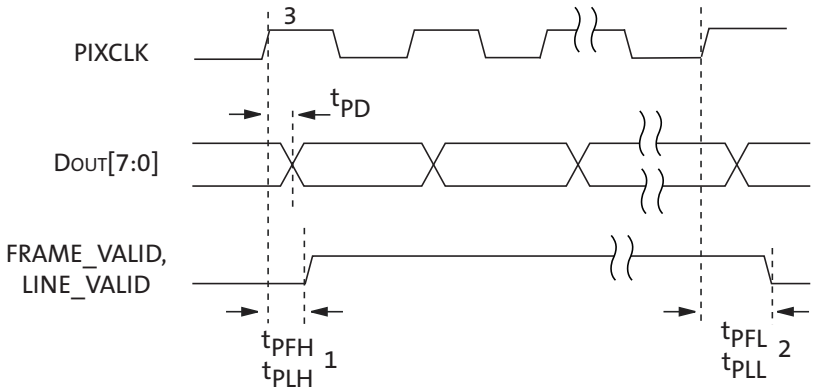
Table 26: AC Electrical Characteristics

EXTCLK = 24 MHz; REG_IN0 = V_{DD}_PHY = 1.8V; V_{DD}_IO = V_{AA} = V_{AA}_PIX = 2.8V; T_J = 25°C unless otherwise stated

Symbol	Parameter	Conditions	Min	Typ	Max	Unit	Notes
f_{EXTCLK}	External clock frequency		6		54	MHz	1
DEXTCLK	External input clock duty cycle		40	50	60	%	
t_{JITTER}	External input clock jitter		–	500	–	ps	
f_{PIXCLK}	Pixel clock frequency		6		96	MHz	
$t_{RPIXCLK}$	Pixel clock rise time	C_{LOAD} = 25pf	–			ns	
$t_{FPIXCLK}$	Pixel clock fall time	C_{LOAD} = 25pf	–			ns	
t_{PD}	PIXCLK to data valid		–			ns	
t_{PFH}	PIXCLK to FV HIGH		–			ns	
t_{PFL}	PIXCLK to FV LOW		–			ns	
t_{PLH}	PIXCLK to LV HIGH		–			ns	
t_{PLL}	PIXCLK to LV LOW		–			ns	
PIXCLK slew rate							
	Programmable Slew = 7	V_{DD}_IO = 2.8V, C_{LOAD} = 25pf	–		–	V/ns	
		V_{DD}_IO = 1.8V, C_{LOAD} = 25pf	–		–	V/ns	
	Programmable Slew = 4	V_{DD}_IO = 2.8V, C_{LOAD} = 25pf	–		–	V/ns	
		V_{DD}_IO = 1.8V, C_{LOAD} = 25pf	–		–	V/ns	
	Programmable Slew = 0	V_{DD}_IO = 2.8V, C_{LOAD} = 25pf	–		–	V/ns	
		V_{DD}_IO = 1.8V, C_{LOAD} = 25pf	–		–	V/ns	
Output slew rate							
	Programmable Slew = 7	V_{DD}_IO = 2.8V, C_{LOAD} = 25pf	–		–	V/ns	
		V_{DD}_IO = 1.8V, C_{LOAD} = 25pf	–		–	V/ns	
	Programmable Slew = 4	V_{DD}_IO = 2.8V, C_{LOAD} = 25pf	–		–	V/ns	
		V_{DD}_IO = 1.8V, C_{LOAD} = 25pf	–		–	V/ns	
	Programmable Slew = 0	V_{DD}_IO = 2.8V, C_{LOAD} = 25pf	–		–	V/ns	
		V_{DD}_IO = 2.8V, C_{LOAD} = 25pf	–		–	V/ns	

Notes: 1. V_{IH}/V_{IL} restrictions apply.

Figure 44: Parallel Pixel Bus Timing Diagram



Notes: 1. FRAME_VALID leads LINE_VALID by 6 PIXCLKs.
2. FRAME_VALID trails LINE_VALID by 6 PIXCLKs.

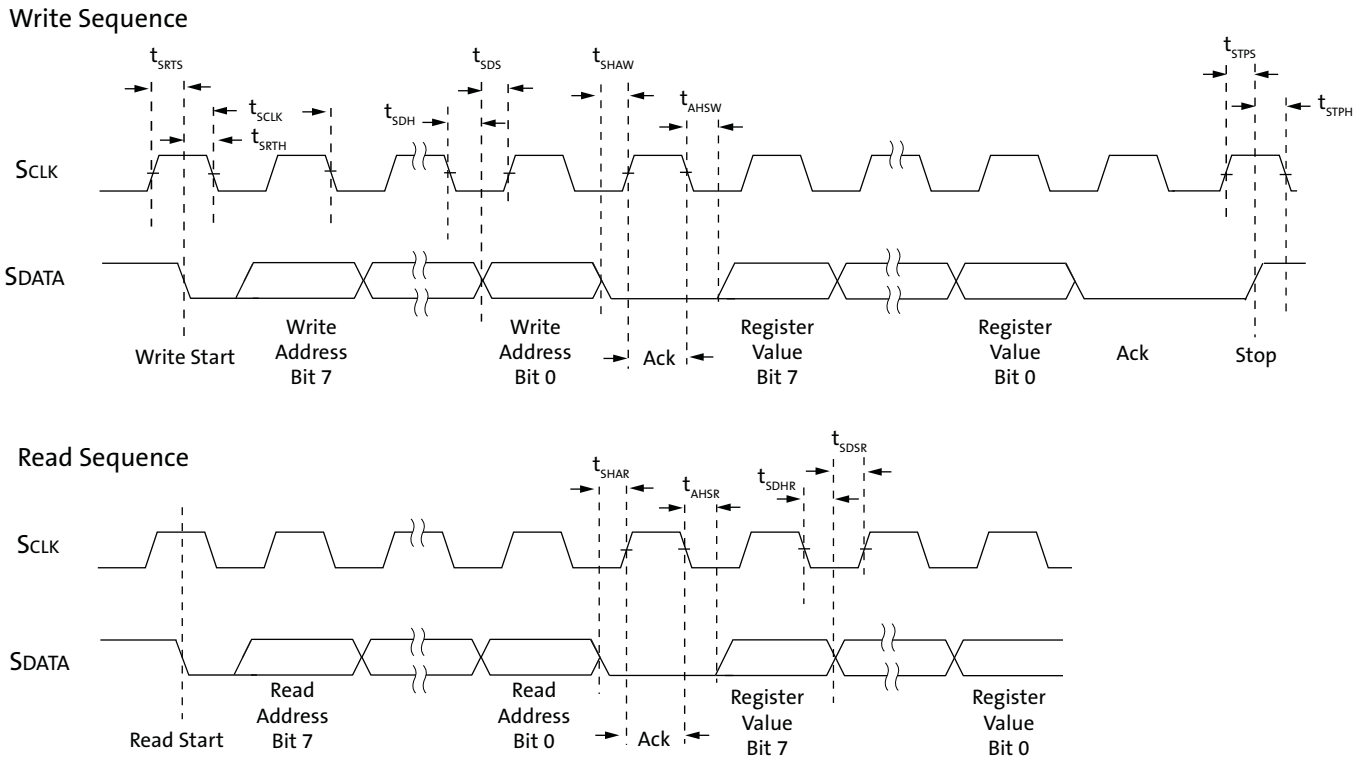
3. DOUT[7:0], FRAME_VALID, and LINE_VALID are shown with respect to the rising edge of PIXCLK. This feature is programmable and DOUT[7:0], FRAME_VALID, and LINE_VALID can be synchronized to the falling edge of PIXCLK.

Table 27: Two-Wire Serial Interface Timing Data

^{f}EXTCLK = 24 MHz; REG_IN0 = 1.8V; VDD_IO = 1.8V; VAA = 2.8V; VAA_PIX = 2.8V; T_J = 70°C; CLOAD = 68.5pF

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{SCLK}	Serial interface input clock frequency		100	–	400	kHz
t_{SCLK}	Serial interface input clock period		2.5	–	10	µs
	SCLK duty cycle		45	50	55	%
t_r	SCLK/SDATA rise time		–	–	300	ns
t_{SRTS}	Start setup time	Master write to slave	600	–	–	
t_{SRTH}	Start hold time	Master write to slave	300	–	–	ns
t_{SDH}	SDATA hold	Master write to slave	300	–	–	ns
t_{SDS}	SDATA setup	Master write to slave	300	–	–	ns
t_{SHAW}	SDATA hold to ack	Master read from slave	150	–	–	ns
t_{AHSW}	Ack hold to SDATA	Master read from slave	150	–	–	ns
t_{STPS}	Stop setup time	Master write to slave	300	–	–	ns
t_{STPH}	Stop hold time	Master write to slave	600	–	–	ns
t_{SHAR}	SDATA hold to ack	Master write to slave	300	–	–	ns
t_{AHSR}	Ack hold to SDATA	Master write to slave	300	–	–	ns
t_{SDHR}	SDATA hold	Master read from slave	300	–	–	ns
t_{SDSR}	SDATA setup	Master read from slave	350	–	–	ns

Figure 45: Two-Wire Serial Bus Timing Parameters



MIPI AC and DC Electrical Characteristics

Table 28: MIPI High-Speed Transmitter DC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
V_{OD}	HS transmit differential voltage	140	–	270	mV
VCMTX	HS transmit static common mode voltage	150	–	250	mV
ΔV_{OD}	VOD mismatch when output is Differential-1 or Differential-0	–	–	13	mV
Δ VCMTX(1,0)	VCMTX mismatch when output is Differential-1 or Differential-0	–	–	5	mV
V_{OHHS}	HS output HIGH voltage	–	–	360	mV
Z_{OS}	Single-ended output impedance	40	–	62.5	Ω
ΔZ_{OS}	Single-ended output impedance mismatch	–	–	17	%

Table 29: MIPI High-Speed Transmitter AC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
	Data bit rate	–	–	768	Mb/s
t_{rise}	20–80% rise time	150	–	500	ps
t_{fall}	20–80% fall time	150	–	500	ps

Table 30: MIPI Low-Power Transmitter DC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
V_{OL}	Thevenin output low level	–	–	55	mV
V_{OH}	Thevenin output high level	–	1.15	–	V
Z_{OLP}	Output impedance of LP transmitter	110	–	–	Ω

Table 31: MIPI Low-Power Transmitter AC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
t_{rise}	15–85% rise time	–	–	25	ns
t_{fall}	15–85% fall time	–	–	25	ns
Slew	Slew rate (C_{LOAD} 5–20pf)	–	–	200	mV/ns
Slew	Slew rate (C_{LOAD} 20–70pf)	–	–	150	mV/ns

MIPI Specification Reference

The AS0260 design and this documentation is based on the following reference documents:

- MIPI Alliance Standard for CSI-2 version 1.0
- MIPI Alliance Standard for D-PHY version 1.0

Revision History

Rev. G 5/4/15

- Updated “Ordering Information” on page 2
- Removed Confidential marking

Rev. F 3/23/15

- Updated to ON Semiconductor template

Rev. E 2/14/13

- Updating revision history to reflect change from 60fps to 30fps for 720p frame rate in Table 1, “Key Parameters,” on page 1

Rev. D 9/25/12

- Updated Table 1, “Key Parameters,” on page 1
- Updated Table 2, “Available Part Numbers,” on page 1
- Updated Table 1, “Pin Descriptions,” on page 9
- Updated Table 25, “Operating Current Consumption (Parallel),” on page 76
- Updated Table 31, “Operating Current Consumption (MIPI),” on page 76
- Updated Table 32, “Non-Operating Current Consumption,” on page 77
- Updated Table 26, “AC Electrical Characteristics,” on page 77
- Updated Table 37, “MIPI Low-Power Transmitter DC Characteristics,” on page 81

Rev. C 4/5/12

- Updated Table 2, “Available Part Numbers,” on page 1
- Updated Figure 2: “Typical Configuration,” on page 8
- Updated Table 1, “Pin Descriptions,” on page 9
- Updated Table 4, “Power-Up Signal Timing,” on page 10
- Updated Figure 4: “Power-Up Sequence,” on page 11
- Added “Power-Down Sequence” on page 13, including Figure 5, Power-Down Sequence and Table 5, Power-Down Signal Timing
- Replaced “Power-On Reset” with “Power-On Reset” on page 11
- Added “Soft Standby with State Retention” on page 17
- Deleted “One-Time Programmable Memory”
- Updated Table 31, “Operating Current Consumption (MIPI),” on page 76
- Added Table 31, “Operating Current Consumption (MIPI),” on page 76
- Added Table 32, “Non-Operating Current Consumption,” on page 77

Rev. B 1/31/12

- Updated Figure 2: “Typical Configuration,” on page 8
- Updated Table 1, “Pin Descriptions,” on page 9
- Updated Figure 51: “CSP Mechanical Drawing,” on page 74
- Updated Table 26, “Ball Matrix,” on page 74
- Updated Table 25, “Operating Current Consumption (Parallel),” on page 76

Rev. A 8/15/11

- Initial release