AP0101 High-Dynamic Range (HDR) Image Signal Processor (ISP)

AP0101 Data Sheet

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

Features

- Up to 1.2Mp (1280x960) Aptina sensor support
- 45 fps at 1.2Mp, 60 fps at 720p
- Optimized for operation with HDR sensors.
- Color and gamma correction
- Auto exposure, auto white balance, 50/60 Hz flicker avoidance
- Adaptive Local Tone Mapping (ALTM)
- Test Pattern Generator
- Two-wire serial programming interface (Slave)
- Interface to low-cost Flash or EPROM through SPI bus (to config and load patches etc)
- High-level host command interface
- Standalone operation supported
- Up to 5 GPIO
- Fail-safe IO
- Multi-Camera synchronization support

Applications

- Surround, rear and front view cameras
- Blind spot / side mirror replacement cameras
- Automotive viewing/processing fusion cameras
- SMPTE296 HDcctv cameras
- Surveillance network IP cameras

Ordering Information

Table 1: Available Part Numbers

Part Number	Description
AP0101PGAD–AS	6.5mm VFBGA Package Part

Table 2: Key Performance Parameters

Parameter		Value
Primary camera interface		Parallel
Primary camera input format		RAW12 Linear/Companded
Output interface		Up to 20-bit Parallel
Output format		YUV422 8-bit,10-bit, and SMPTE296M 10-,12-bit tonemapped Bayer
Maximum resolution		1280x960 (1.2Mp)
Maximum frame rate		45 fps at 1.2Mp, 60 fps at 720p
Maximum output clock frequency		Parallel clock up to 74.25 Mhz
Supply voltage	V_{DDIO_S}	1.8 or 2.8V nominal
	V_{DDIO_H}	2.5 or 3.3V nominal
	V_{DD_REG}	1.8V nominal
	VCC	2.5 or 3.3V nominal
Operating temp.		–40°C to +105°C
Power consumption		140 mW

Table of Contents

List of Figures

List of Tables

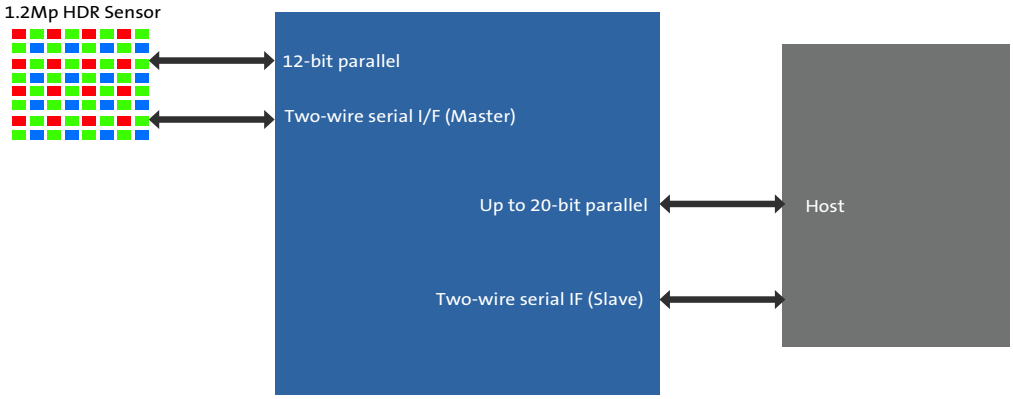
General Description

Aptina's AP0101 is a high-performance, ultra-low power in-line, digital image processor optimized for use with HDR (High Dynamic Range) sensors. The AP0101 provides full auto-functions support (AWB, AE, AFD)and ALTM (Adaptive Local Tone Mapping) to enhance HDR images and advanced noise reduction which enables excellent low-light performance.

Functional Overview

Figure 1 shows the typical configuration of the AP0101 in a camera system. On the host side, a two-wire serial interface is used to control the operation of the AP0101, and image data is transferred using the parallel bus between the AP0101 and the host. The AP0101 interface to the sensor also uses a parallel interface.

Figure 1: AP0101 Connectivity



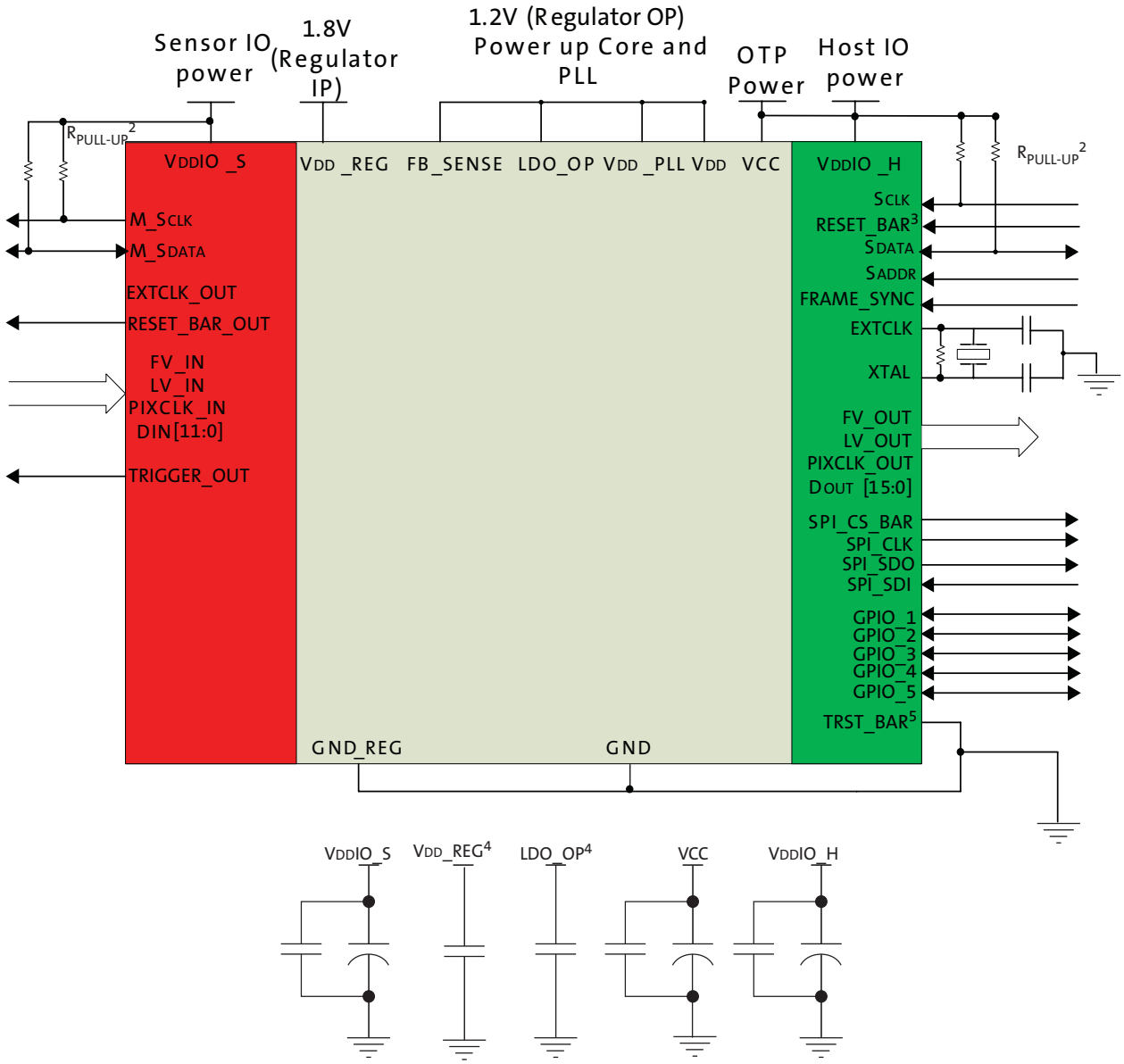
System Interfaces

Figure 2 on page 8 shows typical AP0101 device connections.

All power supply rails must be decoupled from ground using capacitors as close as possible to the package.

The AP0101 signals to the sensor and host interfaces can be a different supply voltage levels to optimize power consumption and maximize flexibility. Table 1 on page 9 provides the signal descriptions for the AP0101.

Figure 2: Typical Configuration



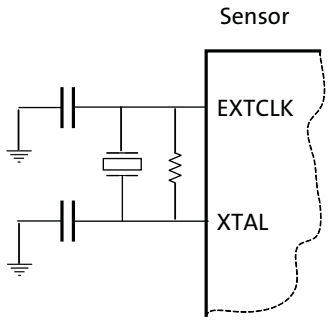
Note:
1. This typical configuration shows only one scenario out of multiple possible variations for this sensor.
2. Aptina recommends a 1.5kΩ resistor value for the two-wire serial interface RPULL-UP; however, greater values may be used for slower transmission speed.
3. RESET_BAR has an internal pull-up resistor and can be left floating if not used.
4. The decoupling capacitors for the regulator input and output should have a value of 1.0uF. The capacitors should be ceramic and need to have X5R or X7R dielectric..
5. TRST_BAR connects to GND for normal operation.
6. Aptina recommends that 0.1µF and 1µF decoupling capacitors for each power supply are mounted as close as possible to the pin. Actual values and numbers may vary depending on layout and design consideration

Crystal Usage

As an alternative to using an external oscillator, a crystal may be connected between EXTCLK and XTAL. Two small loading capacitors and a feedback resistor should be added, as shown in Figure 3.

Aptina does not recommend using the crystal option for applications above 85°C. A crystal oscillator with temperature compensation is recommended for applications that require this.

Figure 3: Using a Crystal Instead of an External Oscillator



Note: The value of load cap is Xtal-dependent. Aptina recommends Xtal with small load cap.

Table 3: Pin Descriptions

Name	Type	Description
EXTCLK	Input	Master input clock. This can either be a square-wave generated from an oscillator (in which case the XTAL input must be left unconnected) or direct connection to a crystal.
XTAL	Output	If EXTCLK is connected to one pin of a crystal, the other pin of the crystal is connected to XTAL pin; otherwise this signal must be left unconnected.
RESET_BAR	Input/PU	Master reset signal, active LOW. This signal has an internal pull up.
SCLK	Input	Two-wire serial interface clock (host interface).
SDATA	I/O	Two-wire serial interface data (host interface).
SADDR	Input	Selects device address for the two-wireslave serial interface. When connected to GND the device ID is 0x90. When wired to VDDIO_H, a device ID of 0xBA is selected.
FRAME_SYNC	Input	This input can be used to set the output timing of the AP0101. The input buffer associated with this input is permanently enabled. This signal should be connected to GND if not used.
STANDBY	Input	Standby mode control, active HIGH.
SPI_SCLK	Output	Clock output for interfacing to an external SPI flash or EEPROM memory.
SPI_SDI	Input	Data in from SPI flash or EEPROM memory. When no SPI device is fitted, this signal is used to determine whether the AP0101 should auto-configure: 0: Do not auto-configure; Two-wire interface will be used to configure the device (host-config mode) 1: Auto-configure. This signal has an internal pull-up resistor.
SPI_SDO	Output	Data out to SPI flash or EEPROM memory.
SPI_CS_BAR	Output	Chip select out to SPI flash or EEPROM memory.
FV_OUT	Output	Host frame valid output (synchronous to PIXCLK_OUT)
LV_OUT	Output	Host line valid output (synchronous to PIXCLK_OUT)

Table 3: Pin Descriptions

Name	Type	Description
PIXCLK_OUT	Output	Host pixel clock output.
DOUT[15:0]	Output	Host pixel data output (synchronous to H_PIXCLK) DOUT[15:0]. Note 20-bit output (SMPTE) also uses GPIO[5:2].
GPIO [5:1]	I/O	General purpose digital I/O. Note: 20-bit output (SMPTE) also uses GPIO[5:2]
TRST_BAR	Input	Must be tied to GND in normal operation.
EXT_CLK_OUT	Output	Clock to external sensor.
RESET_BAR_OUT	Output	Reset signal to external signal.
M_SCLK	I/O	Two-wire serial interface clock (Master).
M_SDATA	I/O	Two-wire serial interface clock (Master).
FV_IN	Input	Sensor frame valid input.
LV_IN	Input	Sensor line valid input.
PIXCLK_IN	Input	Sensor pixel clock input.
DIN[11:0]	Input	Sensor pixel data input DIN[11:0]
TRIGGER_OUT	Output	Trigger signal for external sensor.
VDDIO_S	Supply	Sensor I/O power supply.
GND	Supply	Ground for sensor IO, host IO, PLL , VCC and VDD.
VDD_REG	Supply	Input to on-chip 1.8V to 1.2V regulator.
LDO_OP	Output	Output from on chip 1.8V to 1.2V regulator.
FB_SENSE	Output	On-chip regulator sense signal.
GND_REG	Supply	Ground for on-chip regulator
VDD_PLL	Supply	PLL supply.
VDD	Supply	Core supply.
VCC	Supply	OTPM power supply.
VDDIO_H	Supply	Host I/O power supply.

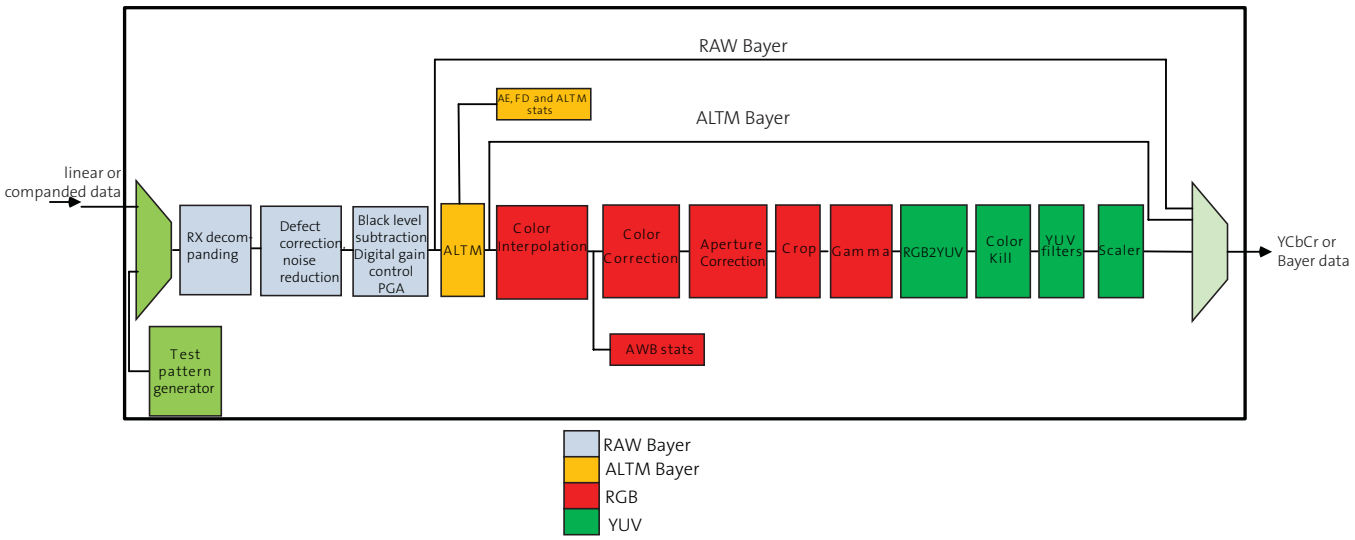
Multi-Camera Synchronization Support

The AP0101 supports multi-camera synchronization via the FRAME_SYNC pin. The host (or controlling entity) 'broadcasts' a sync-pulse to all cameras within the system that triggers capture. The AP0101 will propagate the sync-pulse to the TRIGGER_OUT pin to the sensor's TRIGGER pin.

Image Flow Processor

Image and color processing in the AP0101 is implemented as an image flow processor (IFP) coded in hardware logic. During normal operation, the embedded microcontroller will automatically adjust the operation parameters. The IFP is broken down into different sections, as outlined in Figure 4 on page 9.

Figure 4: Image Flow Processor Block Diagram



Defect Correction

After data decompanding the image stream processing starts with defect correction.

To obtain defect free images, the pixels marked defective during sensor readout and the pixels determined defective by the defect correction algorithms are replaced with values derived from the non-defective neighboring pixels. This image processing technique is called defect correction.

AdaCD (Adaptive Color Difference)

Automotive applications require good performance in extremely low light, even at high temperature conditions. In these stringent conditions the image sensor is prone to higher noise levels, and so efficient noise reduction techniques are required to circumvent this sensor limitation and deliver a high quality image to the user.

The AdaCD Noise Reduction Filter is able to adapt its noise filtering process to local image structure and noise level, removing most objectionable color noise while preserving edge details.

Black Level Subtraction and Digital Gain

After Noise reduction the pixel data goes through black level subtraction and multiplication of all pixel values by a programmable digital gain. Independent color channel digital gain can be adjusted with registers. Black level subtract (to compensate for sensor data pedestal) is a single value applied to all color channels. If the black level subtraction produces a negative result for a particular pixel, the value of this pixel is set to 0.

Positional Gain Adjustments (PGA)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. There are also other factors causing fixed pattern signal gradients in images captured by image sensors. The cumulative result of all these factors is known as image shading. The AP0101 has an embedded shading correction module that can be programmed to counter the shading effects on each individual R, Gb, Gr, and B color signal.

The Correction Function

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

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

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

Adaptive Local Tone Mapping (ALTM)

Real world scenes often have very high dynamic range (HDR) that far exceeds the electrical dynamic range of the imager. Dynamic range is defined as the luminance ratio between the brightest and the darkest object in a scene. In recent years many technologies have been developed to capture the full dynamic range of real world scenes. For example, the multiple exposure method is a widely adopted method for capturing high dynamic range images, which combines a series of low dynamic range images of the same scene taken under different exposure times into a single HDR image.

Even though the new digital imaging technology enables the capture of the full dynamic range, low dynamic range display devices are the limiting factor. Today's typical LCD monitor has contrast ratio around 1,000:1; however, it is not atypical for an HDR image having contrast ratio around 250,000:1. Therefore, in order to reproduce HDR images on a low dynamic range display device, the captured high dynamic range must be compressed to the available range of the display device. This is commonly called tone mapping.

Tone mapping methods can be classified into global tone mapping and local tone mapping. Global tone mapping methods apply the same mapping function to all pixels. While global tone mapping methods provide computationally simple and easy to use solutions, they often cause loss of contrast and detail. A local tone mapping is thus necessary in addition to global tone mapping for the reproduction of visually more appealing images that also reveal scene details that are important for automotive safety and surveillance applications. Local tone mapping methods use a spatially vary mapping function determined by the neighborhood of a pixel, which allows it to increase the local contrast and the visibility of some details of the image. Local methods usually yield more pleasing results because they exploit the fact that human vision is more sensitive to local contrast.

Aptina's ALTM solution significantly improves the performance over global tone mapping. ALTM is directly applied to the Bayer domain to compress the dynamic range from 20-bit to 12-bit. This allows the regular color pipeline to be used for HDR image rendering.

Color Interpolation

In the raw data stream fed by the sensor core to the IFP, each pixel is represented by a 20- or 12-bit integer number, 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 ALTM 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 pads 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 register 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. The color correction matrix can be either programmed by the user or automatically selected by the auto white balance (AWB) algorithm implemented in the IFP. Color correction should ideally produce output colors that are corrected for 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 variables can be adjusted through register settings.

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 register settings.

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.

Gamma Correction

The gamma correction curve is implemented as a piecewise linear function with 33 points, taking 12-bit arguments and mapping them to 10-bit output. The abscissas of the points can be fixed or bottom weighted.

The 10-bit ordinates are programmable through variables.

The AP0101 has the ability to calculate the 33 point curves based on a small number of variable inputs from the host, another option is for the host to program one or both of the 33 point curves.

Camera Control and Auto Functions

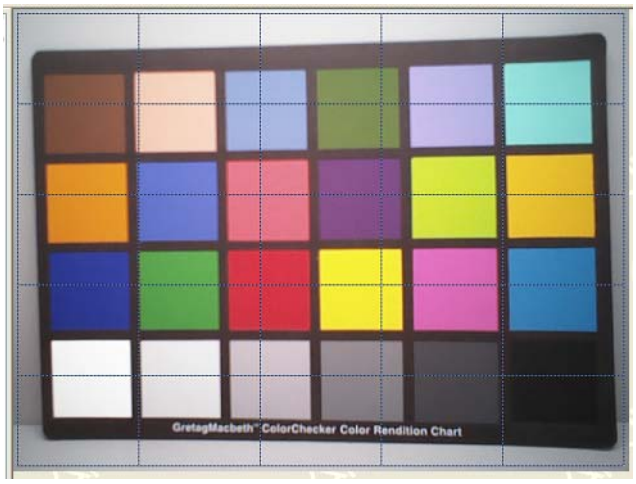
Auto Exposure

The auto exposure algorithm optimizing scene exposure to minimize clipping and saturation in critical area of the image. This is achieved 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 and color pipeline to achieve the desired exposure. The measurement engine subdivides the image into 25 windows organized as a 5 x 5 grid.

Figure 5: 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



AE Track 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 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 scene brightness to the programmable target.

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

Auto White Balance

The AP0101 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 AP0101 AWB displays the current AWB position in color temperature, the range of which will be defined when programming the CCM matrixes.

The region of interest can be controlled through the combination of an inclusion window and an exclusion window.

Flicker Avoidance

Flicker occurs when the integration time is not an integer multiple of the period of the light intensity. The AP0101 can be programmed to avoid flicker for 50 or 60 Hertz. For integration times below the light intensity period (10ms for 50Hz environment), flicker cannot be avoided. The AP0101 supports an indoor AE mode, that will ensure flicker-free operation.

Output Formatting

Uncompressed YCbCr Data Ordering

The AP0101 supports swapping YCbCr mode, as illustrated in Table 4.

Table 4: YCbCr Output Data Ordering

Mode	Data Sequence			
Default (no swap	Cbi	Yi	Cri	Yi+1
Swapped CrCb	Cri	Yi	Cbi	Yi+1
Swapped YC	Yi	Cbi	Yi+1	Cri
Swapped CrCb, YC	Yi	Cri	Yi+1	Cbi

The data ordering for the YCbCr ouput modes for AP0101 are shown in Table 5:

Table 5: YCbCr Output Modes

Mode	Byte	Pixel i	Pixel i+1	Notes
YCbCr_422_8_8	Odd (DOUT [7:0])	Cbi	Cri	Data range of 0-255 (Y=16-235 and C=16-240)
	Even (DOUT [7:0])	Yi	Yi+1	
YCbCr_422_10_10	Odd (DOUT [9:0])	Cbi	Cri	Data range of 0-1023 (Y=64-940 and C=64-960)
	Even (DOUT [9:0])	Yi	Yi+1	
YCbCr_422_16	Single (DOUT [15:0])	Cbi_Yi	Cri_Yi+1	Data range of 0-255 (Y=16-235 and C=16-240)
SMPTE	Single (DOUT [15:0], GPIO[5:2])	Cbi_Yi	Cri_Yi+1	Data range of 4-1019 (Y=64-940 and C=64-960)

The data ordering for the ALTM Bayer ouput modes for AP0101 are shown in Table 6

Table 6: ALTM Bayer Output Modes

Mode	Byte	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
ALTM_Bayer_10	Single	0	0	0	0	0	0	D9	D98	D7	D6	D5	D4	D3	D2	D1	D0
ALTM_Bayer_12	Single	0	0	0	0	D11	D10	D9	D8	D97	D6	D5	D4	D3	D2	D1	D0

Table 6 and 7 show LSB aligned data, it is possible using register setting to obtain MSB aligned data.

The data ordering for the Bayer output modes for AP0101 are shown in Table 7.

Table 7: Bayer Output Modes

Mode	Byte	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Notes
Bayer_12	Single	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	RAW Bayer data
Bayer_20	Odd	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	20-bit decompanded data
	Even	D3	D2	D1	D0	0	0	0	0	0	0	0	0	0	0	0	0	

Slave Two-Wire Serial Interface

The two-wire slave serial interface bus enables read/write access to control and status registers within the AP0101.

The interface protocol uses a master/slave model in which a master controls one or more slave devices.

Protocol

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

- a start or restart condition
- a slave address/data direction byte
- a 16-bit register address
- an acknowledge or a no-acknowledge bit
- data bytes
- 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.

The SADDR pin is used to select between two different addresses in case of conflict with another device. If SADDR is LOW, the slave address is 0x90; if SADDR is HIGH, the slave address is 0xBA. See Table 8 below.

Table 8: Two-Wire Interface ID Address Switching

SADDR	Two-Wire Interface Address ID
0	0x90
1	0xBA

Start Condition

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

At the end of a transfer, the master can generate a start condition without previously generating a stop condition; this is known as a “repeated start” or “restart” condition.

Data Transfer

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

Slave Address/Data Direction Byte

Bits [7:1] of this byte represent the device slave address and bit [0] indicates the data transfer direction. A “0” in bit [0] indicates a write, and a “1” indicates a read. The default slave addresses used by the AP0101 are 0x90 (write address) and 0x91 (read address). Alternate slave addresses of 0xBA (write address) and 0xBB (read address) can be selected by asserting the SADDR input signal.

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 Operation

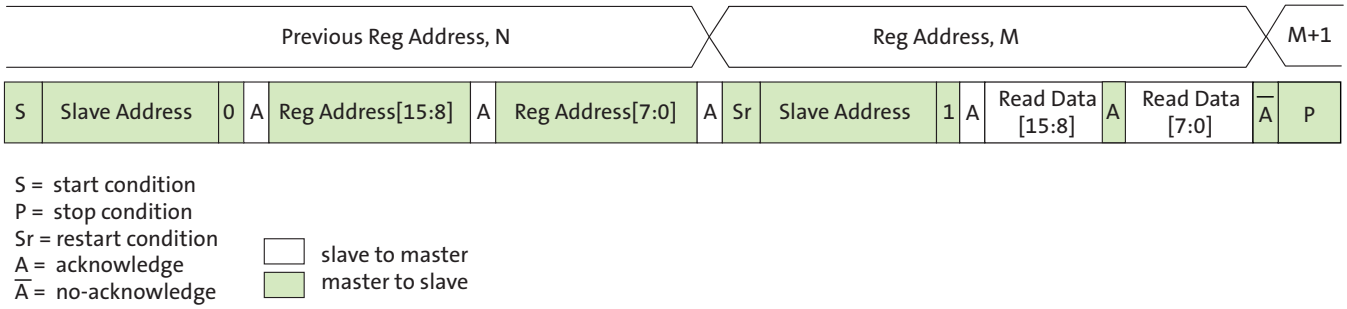
A typical READ or WRITE sequence begins by the master generating a start condition on the bus. After the start condition, the master sends the 8-bit slave address/data direction byte. The last bit indicates whether the request is for a READ or a WRITE, where a “0” indicates a WRITE and a “1” indicates 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 will 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 will then transfer the 16-bit data, 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 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 data transfer is stopped when the master sends a no-acknowledge bit.

Single READ from Random Location

Figure 6 shows the typical READ cycle of the host to the AP0101. The first two bytes sent by the host are an internal 16-bit register address. The following 2-byte READ cycle sends the contents of the registers to host.

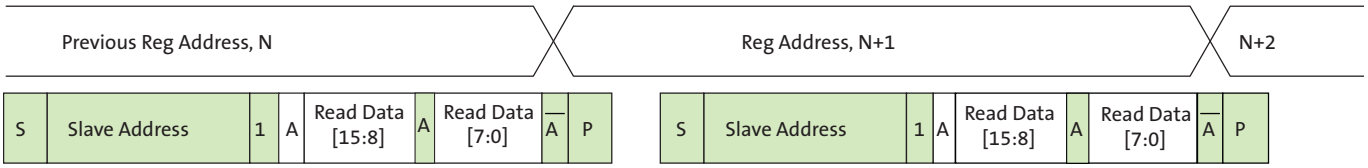
Figure 6: Single READ from Random Location



Single READ from Current Location

Figure 7 shows the single READ cycle without writing the address. The internal address will use the previous address value written to the register.

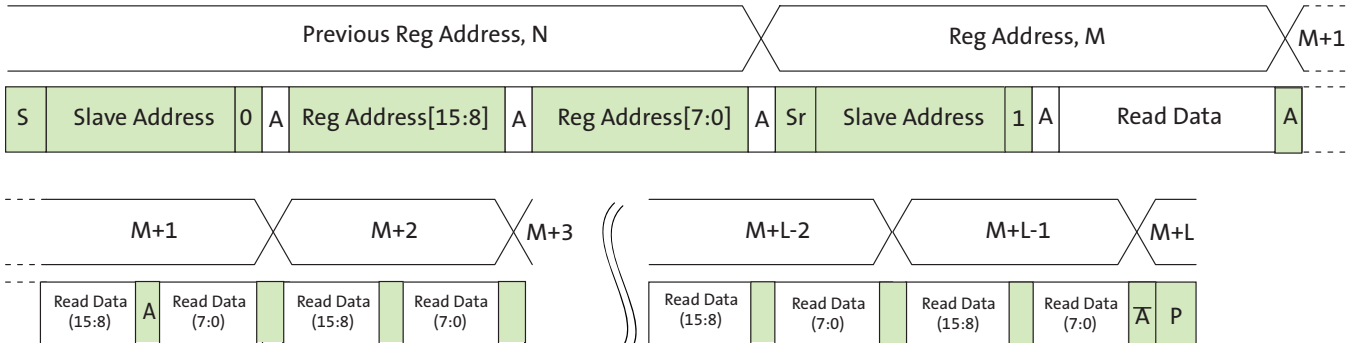
Figure 7: Single Read from Current Location



Sequential READ, Start from Random Location

This sequence (Figure 8) starts in the same way as the single READ from random location (Figure 6 on page 17). 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.

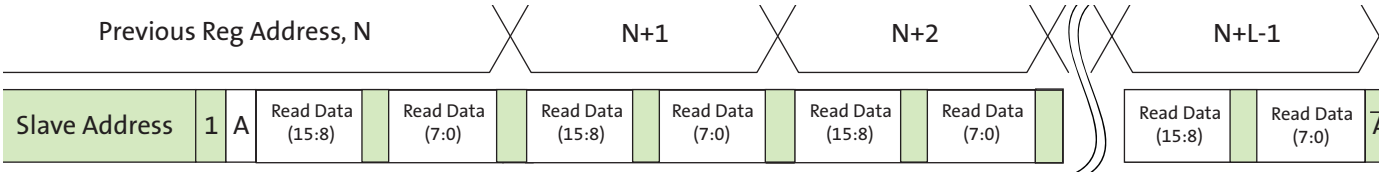
Figure 8: Sequential READ, Start from Random Location



Sequential READ, Start from Current Location

This sequence (Figure 9) starts in the same way as the single READ from current location (Figure 7). 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.

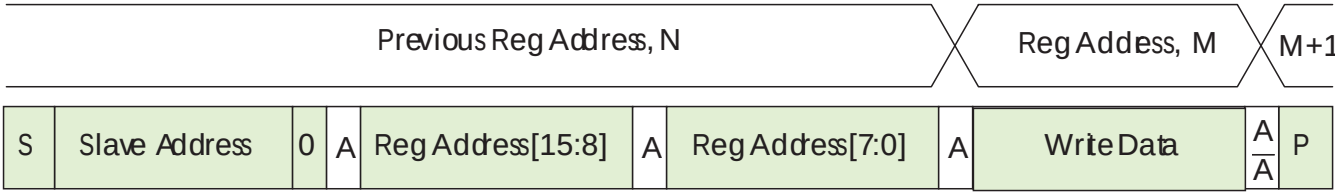
Figure 9: Sequential READ, Start from Current Location



Single Write to Random Location

Figure 10 shows the typical WRITE cycle from the host to the AP0101.The first 2 bytes indicate a 16-bit address of the internal registers with most-significant byte first. The following 2 bytes indicate the 16-bit data.

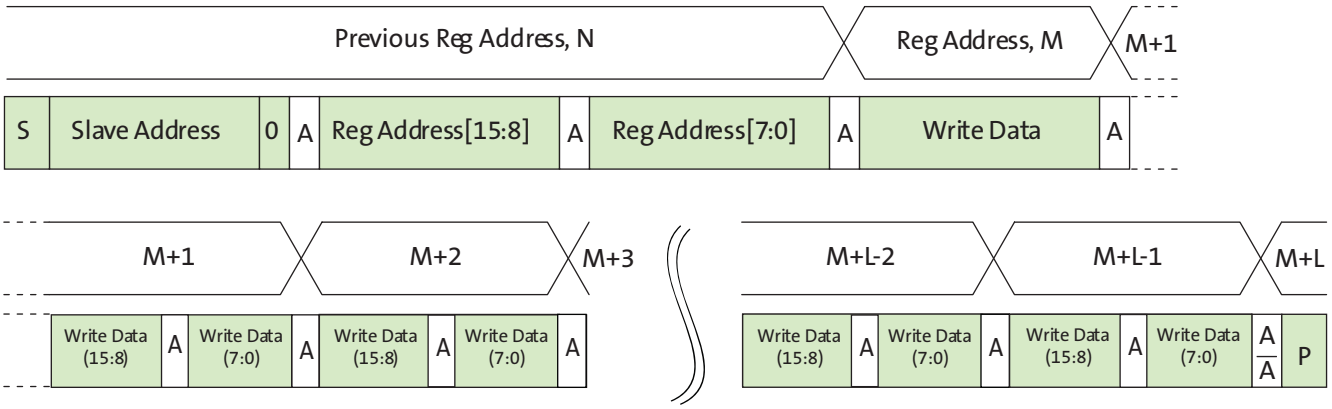
Figure 10: Single WRITE to Random Location



Sequential WRITE, Start at Random Location

This sequence (Figure 11) starts in the same way as the single WRITE to random location (Figure 10). 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 11: Sequential WRITE, Start at Random Location



Device Configuration

After power is applied and the device is out of reset by de-asserting the RESET_BAR pin, it will enter a boot sequence to configure its operating mode. There are essentially four configuration modes: OTPM Config, Flash/EEPROM Config, Auto Config, and Host Config. Figure 12: “Power-Up Sequence – Configuration Options Flow Chart,” on page 20 contains more details on the configuration options.

The AP0101 firmware supports a System Configuration phase at start-up. This consists of five modes of execution:

1. OTPM config
2. Flash detection, then one of:
3. Flash Config
4. Auto Config
5. Host Config

The System Configuration phase is entered immediately after the firmware initializes following power-up or reset. After OTPM config, the firmware then enters the Flash Detection mode.

The Flash Detection mode attempts to detect the presence of an SPI Flash or EEPROM device:

- If no device is detected, the firmware then samples the SPI_SDI pin state to determine the next mode:
  - If SPI_SDI == 0 then it enters the Host-Config mode.
  - If SPI_SDI == 1 then it enters the Auto-Config mode.
- If a device is detected, the firmware switches to the Flash-Config mode.

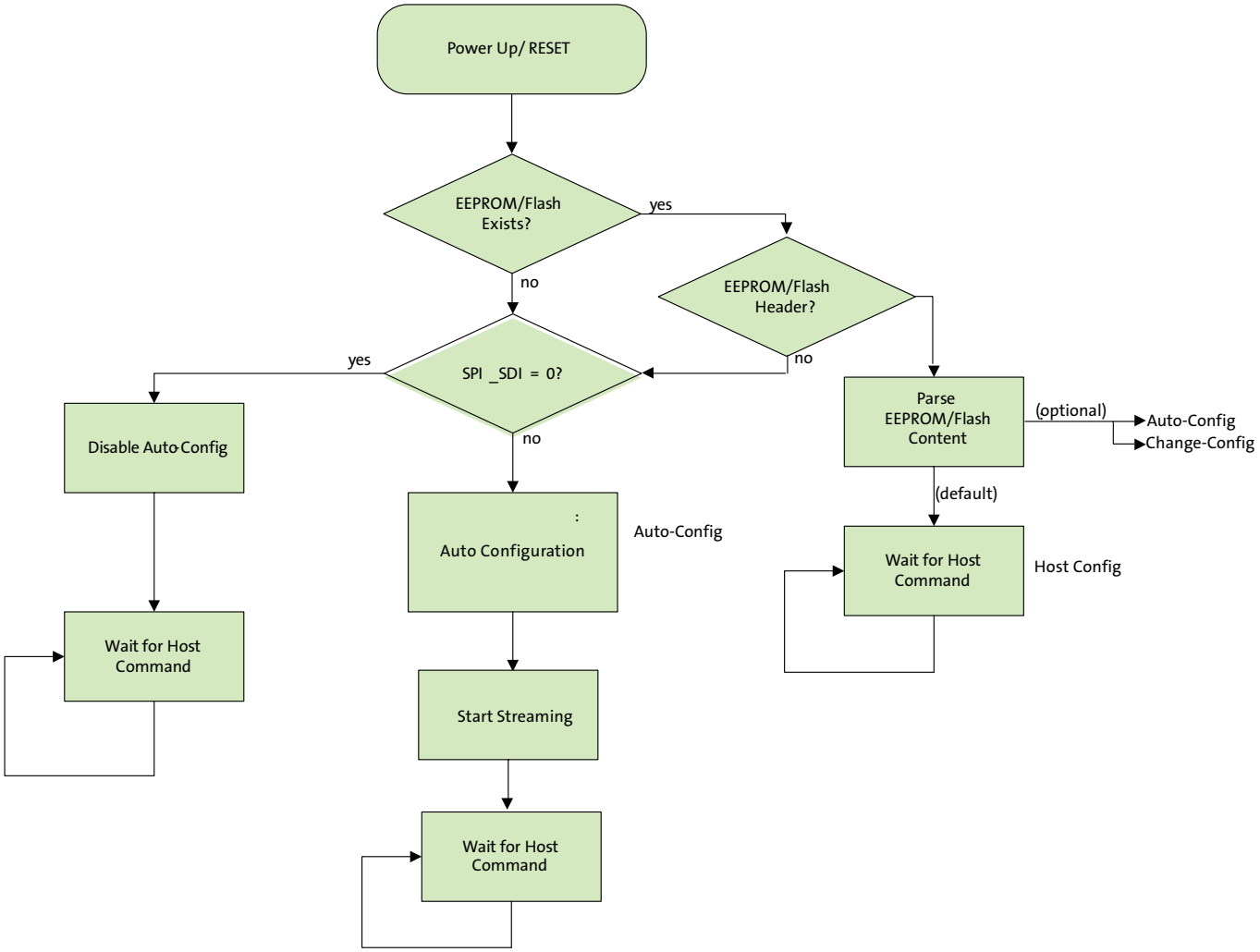
In the Flash-Config phase, the firmware interrogates the device to determine if it contains valid configuration records:

- If no records are detected, then the firmware enters the Auto-Config mode.
- If records are detected, the firmware processes them. By default, when all Flash records are processed the firmware switches to the Host-Config mode. However, the records encoded into the Flash can optionally be used to instruct the firmware to proceed to auto-config, or to start streaming (via a Change-Config).

In the Host-Config mode, the firmware performs no configuration, and remains idle waiting for configuration and commands from the host. The System Configuration phase is effectively complete and the AP0101 will take no actions until the host issues commands.

In the Stsrt Streaming mode, the firmware performs a 'Change-Config' operation. This applies the current configuration settings to the AP0101, and commences streaming. This completes the System Configuration phase.

Figure 12: Power-Up Sequence – Configuration Options Flow Chart



Supported SPI Devices

Table 9 lists supported EEPROM/Flash devices. Devices not compatible will require a firmware patch. Contact Aptina for additional support.

Table 9: SPI Flash Devices

Type	Density	Manufacturer	Device	Speed (MHz)	Standard	Temp Range (°C)	Supported
Flash	1 MB	ST	M25P10-AVMB3	50		–40 to +125	Yes
Flash	8 MB	Atmel	AT26DF081A	70	JEDEC/Device ID	–20 to +85	Yes
EEPROM	1MB	ST	M95M01-R	5		–40 to +130	Yes
EEPROM	8KB	Microchip	25LC080	2		–40 to +125	Yes

Caution **Stresses greater than those listed in Table 10 may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.**

Table 10: Absolute Maximum Ratings

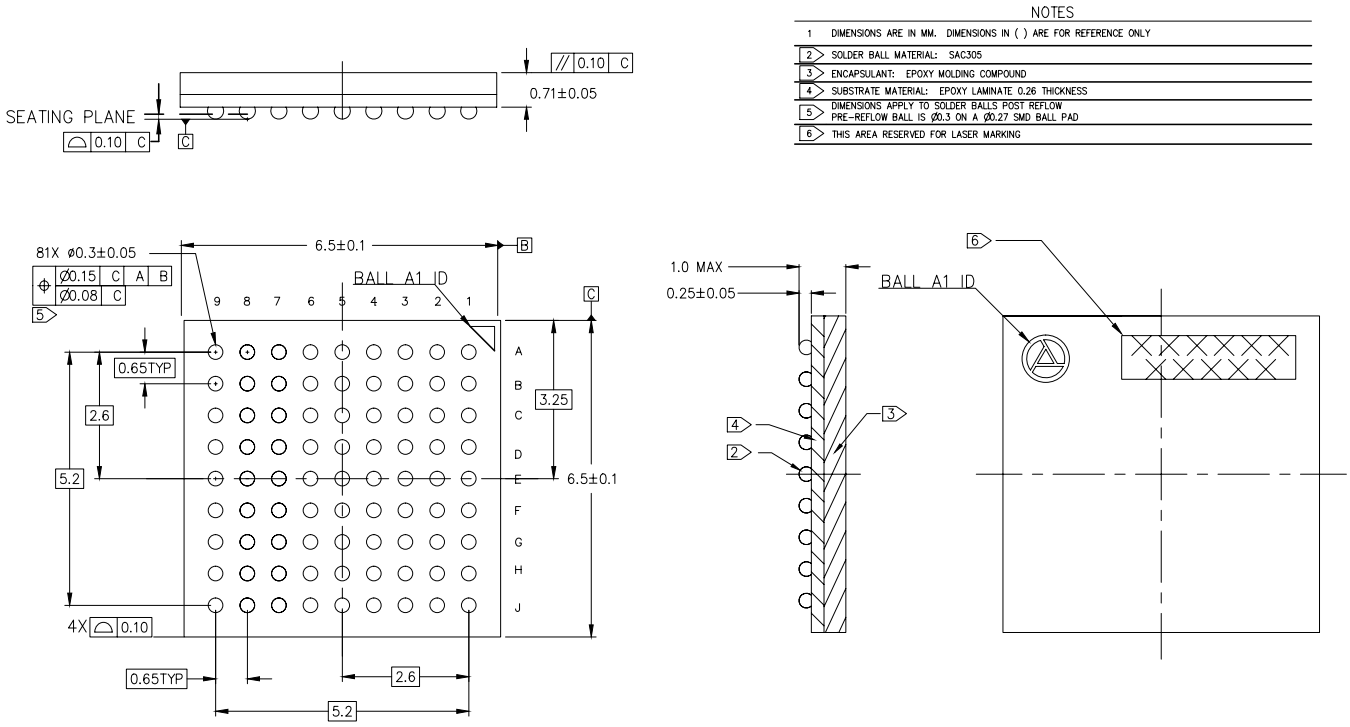
Symbol	Parameter	Rating		Unit
		Min	Max	
V_{DD}	Digital power (1.2V)	-0.3	2.4	V
V_{DDIO_H}	Host I/O power (2.8v)	-0.3	4	V
V_{DDIO_S}	Sensor I/O power (2.8V)	-0.3	4	V
V_{CC}	OTPM power (2.8v)	-0.3	4	V
V_{DD_PLL}	PLL power (1.2V)	-0.3	4	V
V_{IN}	DC Input Voltage	-0.3	$V_{DDIO\_}$*+0.3	V
V_{OUT}	DC Output Voltage	-0.3	$V_{DDIO\_}$*+0.3	V
T_{STG}	Storage temperature	-50	150	°C

Table 11: Electrical Characteristics and Operating Conditions

Parameter	Condition	Min	Typ	Max	Unit
Supply input to on-chip regulator (V_{DD_REG})		1.62	1.8	3.6	V
Host IO voltage (V_{DDDIO_H})		2.25	2.5/3.3	3.6	V
Sensor IO voltage		1.62	1.8/2.8	3.6	V
OTPM power supply (V_{CC})		2.25	2.5/3.3	3.6	V
Functional operating temperature		-40		105	°C
Storage temperature		-55		150	°C

Package and Die Options

Figure 13: Package Diagram



Revision History

Rev. A 6/20/11

- Initial release

10 Eunos Road 8 13-40, Singapore Post Center, Singapore 408600 prodmktg@aptina.com www.aptina.com