1/11-Inch System-On-A-Chip (SOC) CMOS Digital Image Sensor

MT9V113 Data Sheet

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

Features

- Superior low-light performance
- Ultra-low-power, low-cost
- 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, including four-channel lens shading correction
- Arbitrary image scaling with anti-aliasing
- Two-wire serial interface providing access to registers and microcontroller memory
- Selectable output data format: YCbCr, 565RGB, 555RGB, 444RGB, processed Bayer, RAW8- and RAW10-bit
- Programmable I/O slew rate
- Parallel and serial mobile industry processor interface (MIPI) data output
- Xenon and LED flash support with fast exposure adaptation
- Independently configurable gamma correction

Applications

- Cellular phones
- PC cameras
- PDAs
- Consumer electronics

Table 1: Key Performance Parameters

Parameter		Value
Optical format		1/11-inch (4:3)
Full resolution		640 x 480 pixels (VGA)
Pixel size		2.2 x 2.2µm
Dynamic range		63.5dB
SNR MAX		>34dB
Responsivity		1.15 V/lux-sec
Chief ray angle		27.12° MAX at 100% image height
Color filter array		RGB Bayer pattern
Active pixel array area		1.43mm x 1.07mm
Shutter type		Electronic rolling shutter (ERS)
Input clock frequency		6–48 MHz (PLL-enabled)
Maximum frame rate		30 fps at full resolution
Maximum pixel data output		14 Mp/s
Maximum pixel clock frequency		28 MHz
Supply voltage	Analog	2.50–3.10V
	Digital	1.70–1.95V
	I/O	1.70–1.95V or 2.50–3.10V
	PLL	2.50–3.10V
	PHY[1]	1.70–1.95V
ADC resolution		10-bit, on-die
Typical power consumption[2]		74 mW, operating mode, typical voltages and EXTCLK = 24 MHz
		29µW, standby mode, with typical voltages
Operating temperature		–30°C to +70°C (at junction)

Notes: 1. MIPI Physical layer (PHY).
2. Excluding I/O current

Ordering Information

Table 2: Available Part Numbers

Part Number	Description
MT9V113D00STC	Bare die

Table of Contents

List of Figures

List of Tables

Functional Description

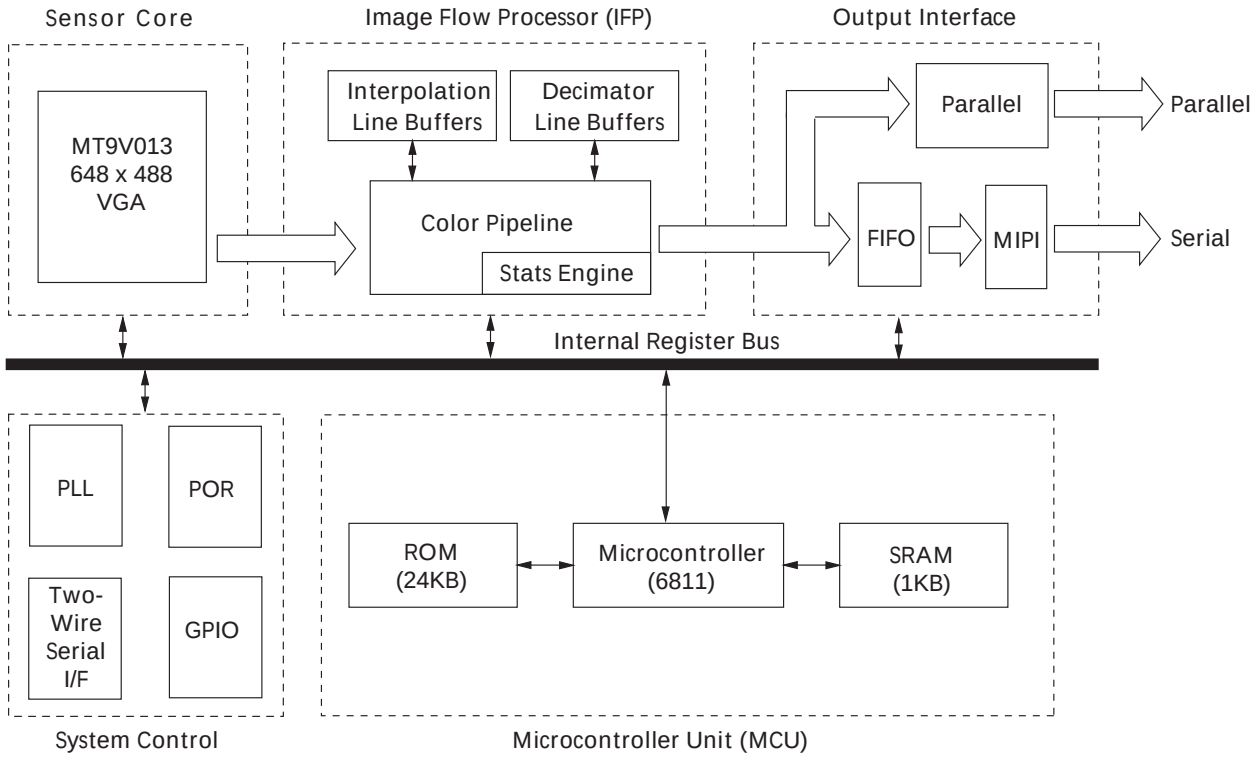
Aptina's MT9V113 is a 1/11-inch VGA CMOS digital image sensor with an integrated advanced camera system. This camera system features a microcontroller (MCU), a sophisticated image flow processor (IFP), and both parallel and serial Mobile Industry Processor Interface (MIPI) ports. The microcontroller 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 648 x 488 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 an ultra-low power operational mode and a superior low-light performance that is particularly suitable for mobile applications. The MT9V113 features Aptina'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.

Architecture Overview

The MT9V113 combines the MT9V013 VGA 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 a parallel bus or a serial data interface. Figure 1 shows the five major functional blocks of the MT9V113.

Figure 1: MT9V113 Block Diagram



Sensor Core

The MT9V113 has a color image sensor with a Bayer color filter arrangement and a VGA active-pixel array with electronic rolling shutter (ERS). The sensor core readout is 10 bits and can be flipped and/or mirrored. The sensor core also supports separate analog and digital gain for all four color channels (R, Gr, Gb, B).

Image Flow Processor (IFP)

The advanced IFP features and flexible programmability of the MT9V113 can enhance and optimize the image sensor performance. Built-in optimization algorithms enable the MT9V113 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, and image formatting with cropping and scaling.

The MT9V113 also includes a hardware sequencer that coordinates all events triggered by the user. The sequencer manages auto white balance (AWB), flicker detection, and auto exposure (AE) for all operating modes including preview, still capture, video, and snapshot with flash.

All modes of operation are individually configurable and organized as two contexts: context A and context B. Each context is defined by sensor image size, frame rate, resolution, and other associated parameters.

Microcontroller Unit (MCU)

The 6811-based MCU communicates with all functional blocks by way of an internal Aptina proprietary bus interface. The MCU firmware configures all the registers in the sensor core and IFP.

System Control

The MT9V113 has a phase-locked loop (PLL) oscillator that can generate the internal sensor clock from the common wireless system clock. The PLL adjusts the incoming clock frequency up, allowing the MT9V113 to run at almost any desired resolution and frame rate within the sensor's capabilities.

Low-power consumption is a very important requirement for all components of wireless devices. The MT9V113 provides power-conserving features, including an internal soft standby mode and a hard standby mode.

A two-wire serial interface bus enables read and write access to the MT9V113'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 for drivers to control the AWB and AE.

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 serial MIPI port. The parallel output port provides 8-bit RGB data or extended 10-bit Bayer data with two general purpose I/O signals.

The general purpose I/Os can be configured to allow the user to output a flash control signal or two least significant data bits during 10-bit parallel output mode.

The MT9V113 also includes programmable I/O slew rate to minimize EMI.

System Interfaces

Figure 2 on page 9 shows typical MT9V113 device connections. For low-noise operation, the MT9V113 requires separate power supplies for analog and digital sections of the die. Both power supply rails should 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.

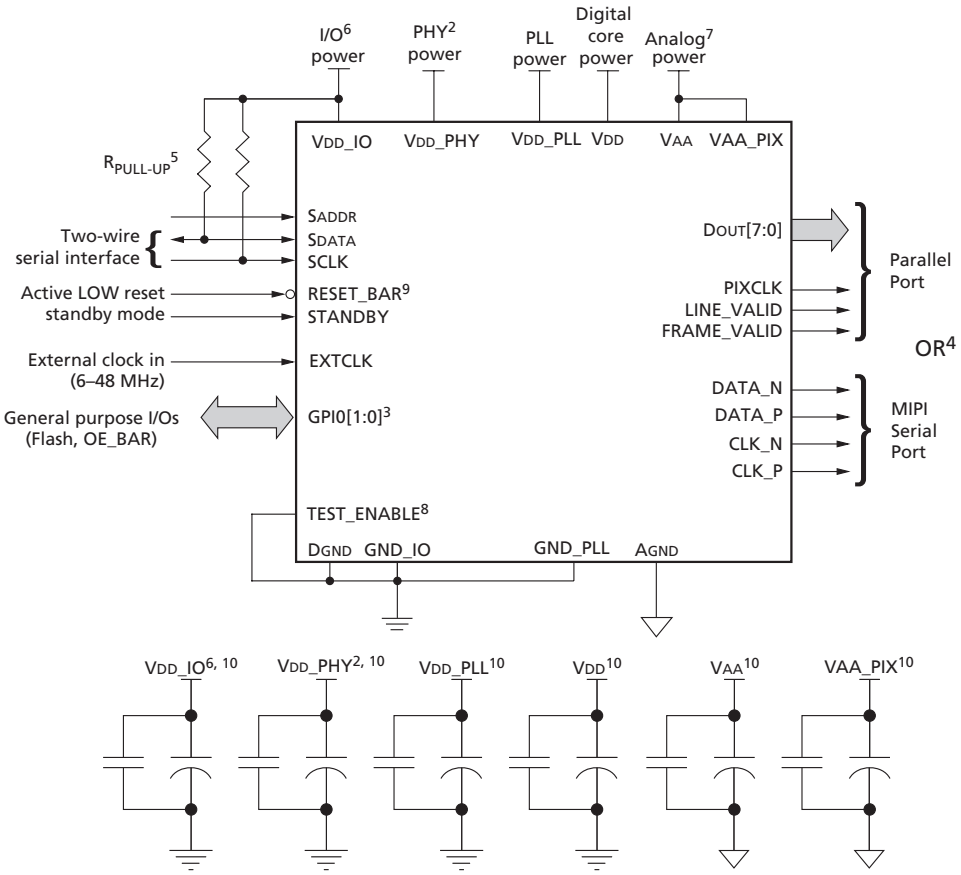
The MT9V113 provides dedicated signals for digital core, PHY, and I/O power domains that can be at different voltages. The PLL and analog circuitry require clean power sources. Table 3 provides the signal descriptions for the MT9V113.

Table 3: Signal Descriptions

Name	Type	Description	Notes
EXTCLK	Input	Input clock signal.	
RESET_BAR	Input	Master reset signal, active LOW.	
STANDBY	Input	Controls sensor's standby mode, active HIGH.	
SCLK	Input	Two-wire serial interface clock.	
SADDR	Input	Selects device address for the two-wire serial interface.	
SDATA	I/O	Two-wire serial interface data.	
GPIO[1:0]	I/O	General purpose digital I/O or DOUT[1:0] for 10-bit data output mode.	
CLK_N	Output	Differential MIPI clock (sub-LVDS, negative) (must leave floating if not used).	
CLK_P	Output	Differential MIPI clock (sub-LVDS, positive) (must leave floating if not used).	
DATA_N	Output	Differential MIPI data (sub-LVDS, negative) (must leave floating if not used).	
DATA_P	Output	Differential MIPI data (sub-LVDS, positive) (must leave floating if not used).	
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.	
VDD	Supply	Digital power.	
DGND	Supply	Digital ground.	1
VDD_IO	Supply	I/O power supply.	
GND_IO	Supply	I/O ground.	
VDD_PLL	Supply	PLL power.	
GND_PLL	Supply	PLL ground.	2
VAA	Supply	Analog power.	
VAA_PIX	Supply	Pixel array power.	
AGND	Supply	Analog ground.	1
VDD_PHY	Supply	I/O power supply for the MIPI interface.	3
TEST_ENABLE	Input	Connect to DGND or leave floating for normal operation.	

Notes:
1. AGND and DGND are not connected internally.
2. GND_PLL is not connected to either AGND or DGND internally.
3. If the MIPI interface is not used, connect VDD_PHY to VDD. Do not leave the power signal floating.

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 following signals must be left floating: DATA_P, DATA_N, CLK_P, and CLK_N. The VDD_PHY power signal must always be connected to the 1.8V supply.
3. The function and direction of GPIO signals are programmable. GPIO signals do not have internal pull-up or pull-down resistors. An external source needs to drive GPIO signals when GPIO signals are configured as inputs.
4. Only one of the output modes (serial or parallel) can be used at any time.
5. 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.
6. All inputs must be configured with VDD_IO.
7. VAA and VAA_PIX must be tied together.
8. TEST_ENABLE has an internal pull-down resistor and can be left floating.
9. RESET_BAR has an internal pull-up resistor and can be left floating.
10. Aptina 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.

Power-On Reset

The MT9V113 includes a power-on reset feature that initiates a reset upon power-up. Even though this feature is included on the device, it is advised that the user still manually assert a hard 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 after hard reset are shown in Table 4 on page 11.

A soft reset sequence to the sensor has the same 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.

Standby

The MT9V113 supports two different standby modes:

- Hard standby mode
- Soft standby mode

Soft standby disables the sensor core and most of the digital logic. It can be initiated by writing to an internal register by host system. During soft standby the two-wire serial interface is still active and the sensor can be programmed through register commands. All register settings and RAM content will be preserved. It can be performed in any sequencer state after all AE, AWB, histogram, and flicker calculations are finished.

Soft standby is executed after the completion of the current line by default. It is possible to synchronize the execution of standby with the end of frame through a register.

Hard standby mode uses the STANDBY signal to ensure the lowest power consumption. When using this mode all registers are set to default values, although the SOC will write over some of the values. The two-wire serial interface is not active and the sensor must be started by de-asserting STANDBY.

All output signal status during standby are shown in Table 4 on page 11.

Table 4: Status of Output Signals During Reset and Standby

Signal	Reset	Post-Reset	Standby
DOUT[7:0]	High-Z	High-Z	High-Z
PIXCLK	High-Z	High-Z	High-Z
LV	High-Z	High-Z	High-Z
FV	High-Z	High-Z	High-Z
DATA_N	0	0	0
DATA_P	0	0	0
CLK_N	0	0	0
CLK_P	0	0	0
GPIO[1:0]	High-Z	High-Z	High-Z

General Purpose I/O Interface

The MT9V113 provides two general-purpose I/O signals, GPIO[1] and GPIO[0].

In input mode, both GPIO signals can be monitored by the host processor. The GPIO[0] can be programmed as the output enable control signal for DOUT [7:0], LV, FV, and PIXCLK. Output enable signal is active LOW. When it is driven to logic 1 by host, all the outputs are in High-Z state. Each signal has separate direction control bits.

GPIO[1] can be programmed as the flash STROBE output signal. Both GPIO[1] and GPIO[0] can also be programmed to provide the least significant image data bits (DOUT[1:0]) during 10-bit output mode (DOUT[9:0]).

Module ID

MT9V113 provides 3 bits of module ID to be read by the host processor. The 3 least significant bits (LSBs) of the parallel output (DOUT[2], DOUT[1], and DOUT[0]) can be read to retrieve the 3-bit module ID.

The 3 bits require external pull-up or pull-down resistors. The module ID can be read from an internal register when parallel outputs are disabled. After the parallel signals are enabled, the module ID bits will be forced to “000” values internally.

Image Data Output Interface

The user can select either the 8-bit parallel or serial MIPI output to transmit the sensor image data to host system. Only one of the output modes can be used at any time.

The MT9V113 has an output FIFO to retain a constant pixel output clock independent from the data output rate variations due to scaling factor.

Parallel Port

The MT9V113 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.

MT9V113 output data is synchronized with the PIXCLK output. When LV is HIGH, one pixel value is output on the 8-bit DOUT port every PIXCLK period as shown in Figure 3. PIXCLK is continuously running, even during the blanking period. The MT9V113 can be programmed to delay the PIXCLK edge relative to the DOUT transitions. This can be achieved by programming a register. PIXCLK phase can be programmable by user.

Figure 3: Pixel Data Timing Example

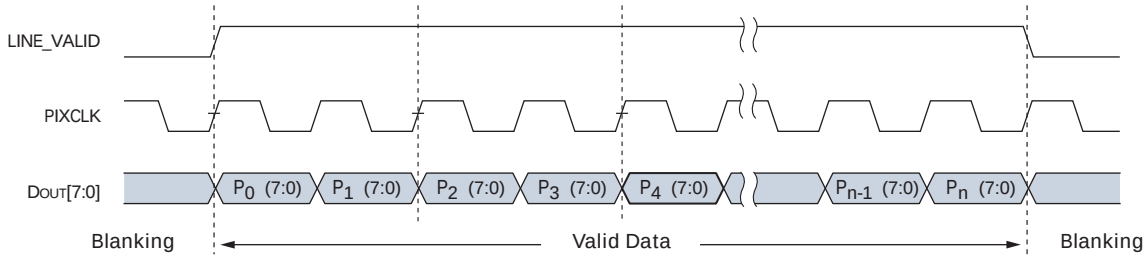
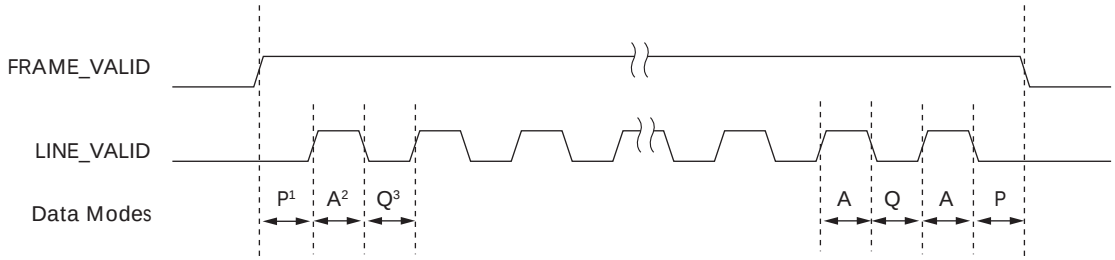


Figure 4: Row Timing, FV, and LV Signals



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

Serial Port

This section describes how frames of pixel data are represented on the high-speed MIPI serial interface. The MIPI output transmitter implements a serial differential sub-LVDS transmitter capable of up to 224 Mb/s. 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) indicate ultra low-power state (ULPS) corresponding to (nominal) 0V levels being driven on CLK_P, CLK_N, DATA_P, and DATA_N. 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

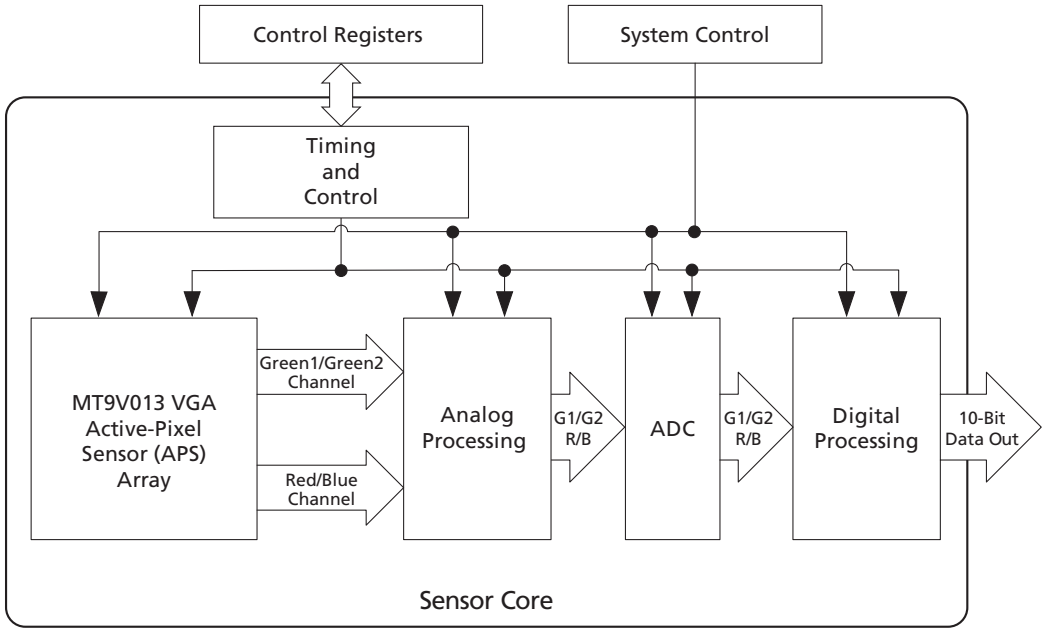
The sensor core of the MT9V113 is a progressive-scan sensor that generates a stream of pixel data at a constant frame rate. Figure 5 shows a block diagram of the sensor core. It includes the MT9V013 VGA active-pixel array. The timing and control circuitry sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and reading that row, the pixels in the row integrate incident light. The exposure is controlled by varying the time interval between reset and readout. Once a row has been 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.

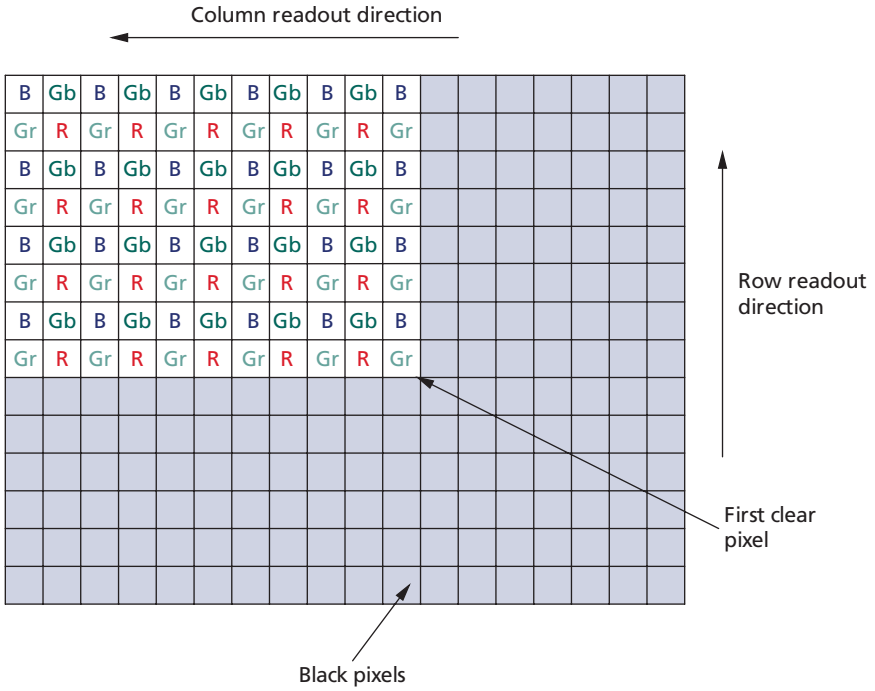
The output from the sensor core is a Bayer pattern; alternate rows are a sequence of either green and red pixels or blue and green pixels. The offset and gain stages of the analog signal chain provide per-color control of the pixel data.

Figure 5: Sensor Core Block Diagram



The sensor core uses a Bayer color pattern, as shown in Figure 6. 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.

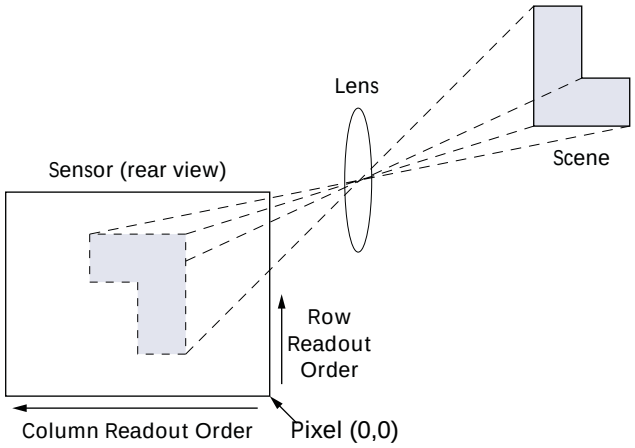
Figure 6: Pixel Color Pattern Detail (Bottom Right Corner)



The MT9V113 sensor core pixel array is shown with pixel (0,0) in the bottom right corner, which reflects the actual layout of the array on the die. Figure 7 on page 16 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 7: Imaging a Scene



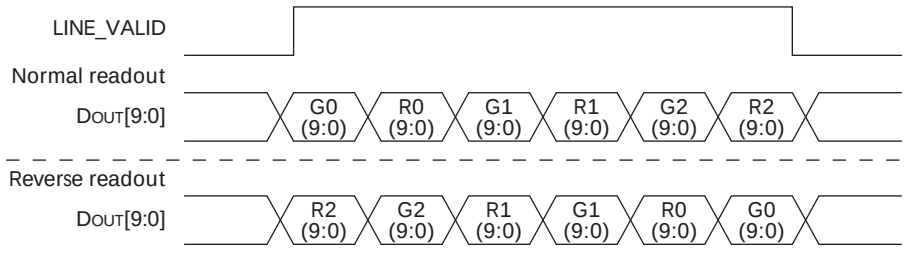
The sensor core supports different readout options to modify the image before it is sent to the IFP. The readout can be limited to a specific window size of the original pixel array.

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 registers. The edge pixels in the 648 x 488 array are present to avoid edge effects and should not be included in the visible window.

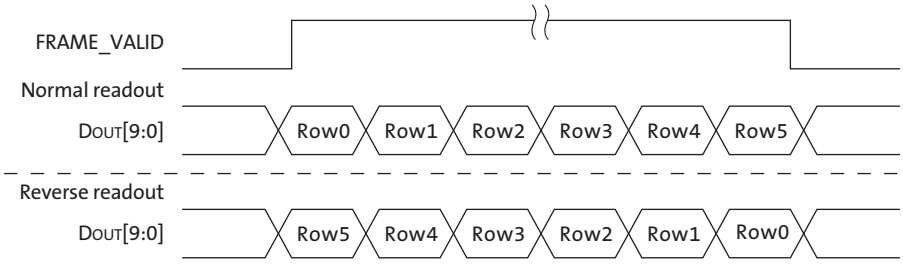
When the sensor is configured to mirror the image horizontally, the order of pixel readout within a row is reversed, so that readout starts from the last column address and ends at the first column address. Figure 8 shows a sequence of 6 pixels being read out with normal readout and reverse readout. This change in sensor core output is corrected by the IFP.

Figure 8: Six 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 9 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 9: Six Rows in Normal and Row Mirror Readout Mode



The MT9V113 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 10 shows the readout with 2X skipping.

Figure 10: Eight Pixels in Normal and Column Skip 2X Readout Mode

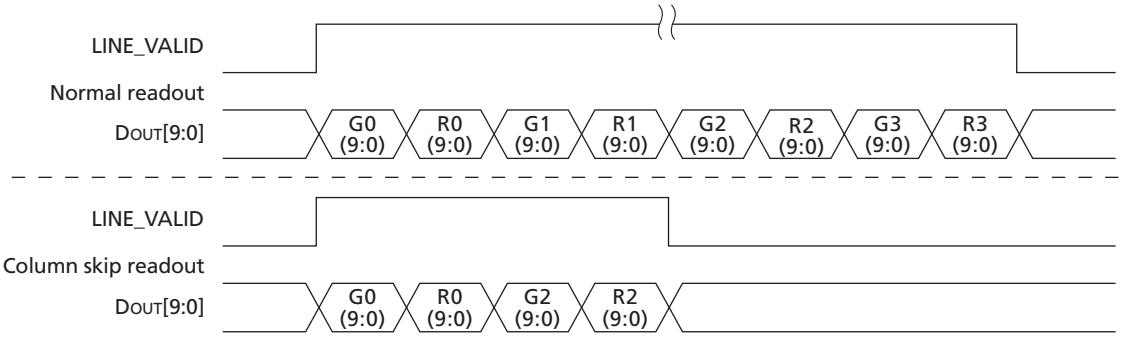


Figure 11 on page 18 through Figure 14 on page 19 show the different skipping modes supported in MT9V113.

Figure 11: Pixel Readout (no skipping)

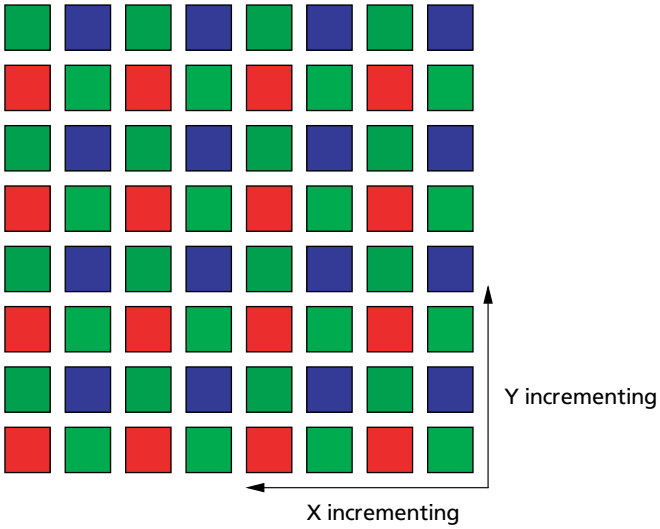


Figure 12: Pixel Readout (x-direction skipping)

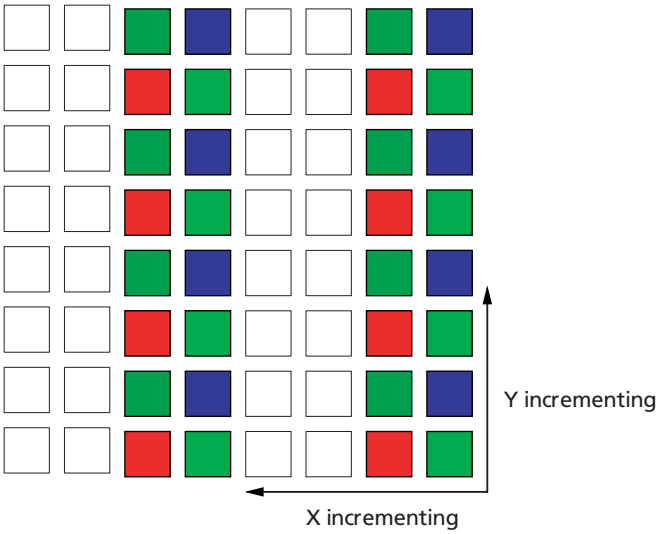


Figure 13: Pixel Readout (y-direction skipping)

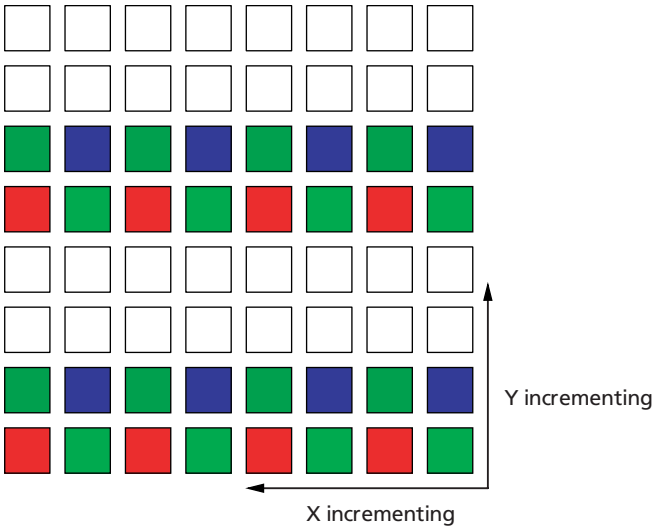


Figure 14: Pixel Readout (x- and y-direction skipping)

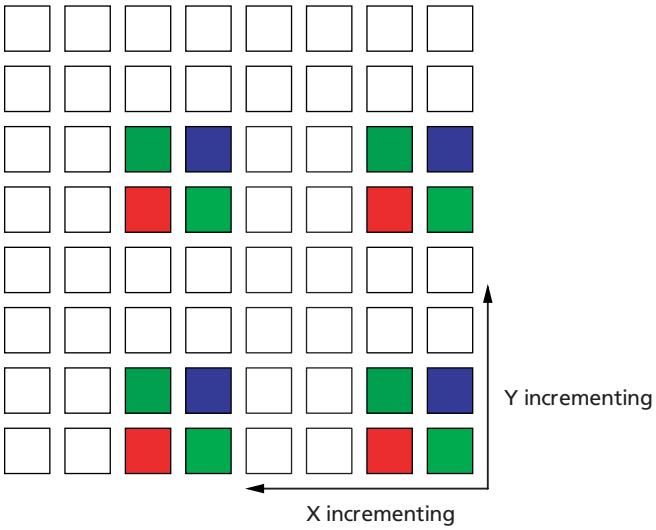
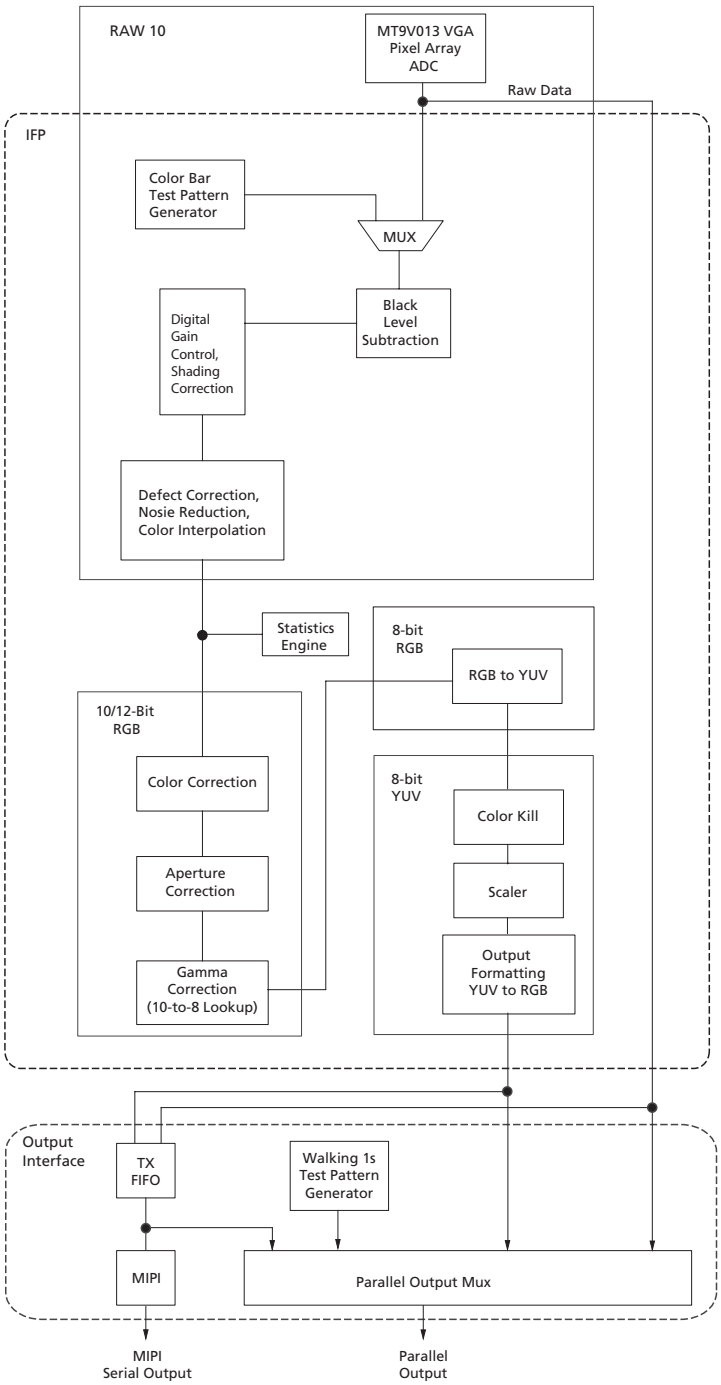


Image Flow Processor

Image control processing in the MT9V113 is implemented in the IFP hardware logic. The IFP registers can be programmed by the host processor. For normal operation, the 6811 microcontroller automatically adjusts the operational parameters of the IFP. Figure 15 shows the image data processing flow within the IFP.

Figure 15: Image Flow Processor



For normal operation of the MT9V113, streams of raw image data from the sensor core are continuously fed into the color pipeline. The MT9V113 features an automatic color bar test pattern generation function to emulate sensor images as shown in Figure 16. 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 a register. Disabling the MCU is recommended during color bar test pattern testing.

Figure 16: Color Bar Test Pattern

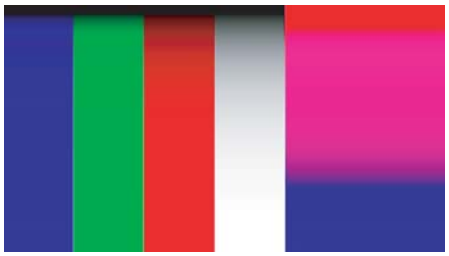
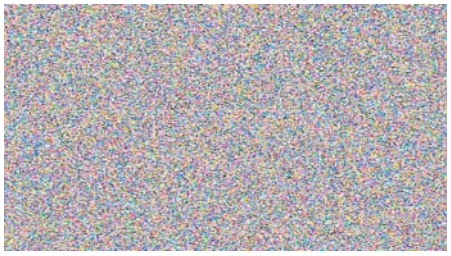
Test Pattern	Example
Flat Field	
Vertical Ramp	
Color Bar	
Pseudo-Random	

Image Corrections

Image stream processing starts with black level subtraction and multiplication of all pixel values by a programmable digital gain. Both operations can be independently set to separate values for each color channel (R, Gr, Gb, B). Independent color channel digital gain can be adjusted with registers. Independent color channel black level adjustments can also be made. The value of this pixel is set to "0" if the black level subtraction produces a negative result for a particular pixel.

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 MT9V113 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 IFP performs continuous defect correction that can mask pixel array defects such as high dark-current (hot) pixels and pixels that are darker or brighter than their neighbors due to photoresponse nonuniformity. The module is edge-aware with exposure that is based on configurable thresholds. The thresholds are changed continuously based on the brightness of the current scene. Enabling and disabling noise reduction, and setting thresholds can be defined through register settings.

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 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. 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 variables can be adjusted through register settings.

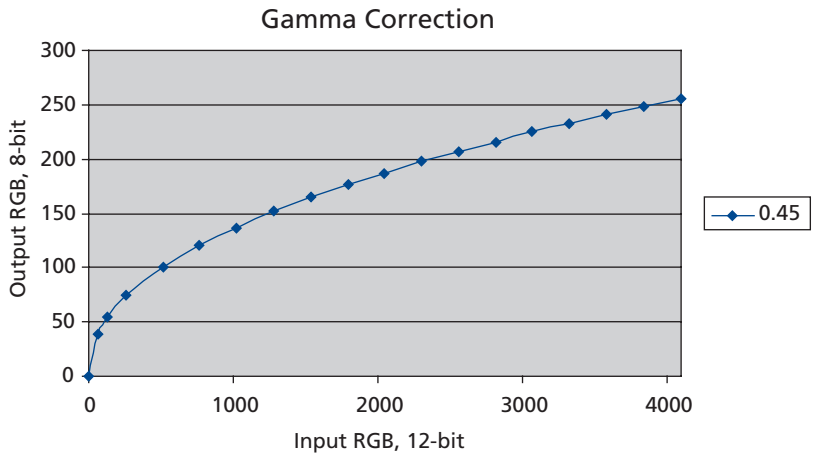
To increase image sharpness, a programmable 2D aperture correction (sharpening filter) is applied to color-corrected image data. The gain and threshold for 2D correction can be defined through register settings.

Gamma Correction

The gamma correction curve (as shown in Figure 17) 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 IFP registers.

The MT9V113 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 brighter lighting condition, the other one corresponding to a darker lighting condition. At power-up, the IFP loads the two tables with default values. The final gamma correction table used depends on the brightness of the scene and can take the form of either uploaded tables or an interpolated version of the two tables. A single (non-adjusting) table for all conditions can also be used.

Figure 17: Gamma Correction Curve



Special effects like negative image, sepia, or B/W can be applied to the data stream at this point. These effects can be enabled and selected by registers.

To remove high- or low-light color artifacts, a color kill circuit is included. It affects only pixels whose luminance exceeds a certain preprogrammed threshold. The U and V values of those pixels are attenuated proportionally to the difference between their luminance and the threshold.

Image Scaling and Cropping

To ensure that the size of images output by the MT9V113 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 for 4X, 2X, and 1X can be achieved. The location of the cropped window is configurable so that panning is also supported. A separate cropped window is defined for

context A and context B. In both contexts, 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.

Camera Control and Auto Functions

The MT9V113 can operate in several modes including preview, still capture (snapshot), and video. All modes of operation are individually configurable and are organized as two contexts—context A and context B. Context switching can be accomplished by sending a command through the two-wire serial interface.

Context Switching

Context A is primarily intended for use in the preview mode. During this mode, the sensor usually outputs low resolution images (QVGA: 320 x 240) at a relatively high frame rate.

Context B can be configured for the still capture or video mode. For still capture mode, the user typically specifies the desired output image size. For video mode, the user can select a different image size and a fixed frame rate. If the flash is enabled, the user can also change the frame rate in still capture mode.

To take a snapshot, the user first sends a command to change the context from A to B. A typical sequence of events after this command are:

1. The camera may turn on its LED flash if it has one and is required to use it. With the flash on, the camera exposure and white balance are automatically adjusted to the changed illumination of the scene.
2. The camera captures one or more frames of desired size. A camera equipped with a xenon flash generates strobes while capturing images. When image capture is completed, the camera automatically returns to context A and resumes running in preview mode.

To start video capture, the user must program relevant context B settings, such as capture mode, image size, and frame rate. To start video capture, the user first sends a context switching command. Upon receiving it, the MT9V113 switches to the modified context B settings, while continuing to output YUV-encoded image data. AE adjusts automatically and provides smooth continuous image outputs. To exit the video capture mode, the user must send another context change command to switch back to context A.

The MT9V113 supports both xenon and LED flash through one of the GPIO signals. The timing of the flash STROBE signal with the default settings is shown in Figure 18, Figure 19, and Figure 20 on page 25. The flash can be programmed to fire only once or be delayed by few frames. For the xenon flash, the duration of the flash time can be also programmed.

The flash output can be inverted by programming a register.

Figure 18: Xenon Flash Enabled

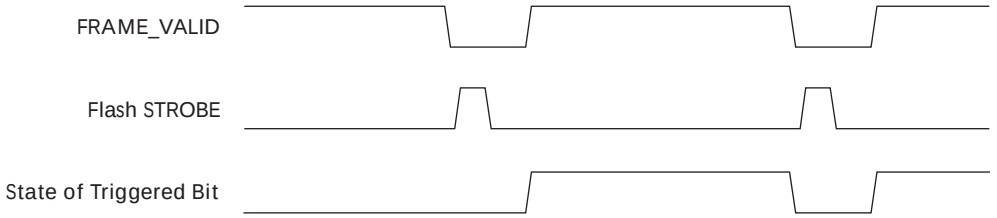


Figure 19: LED Flash Enabled

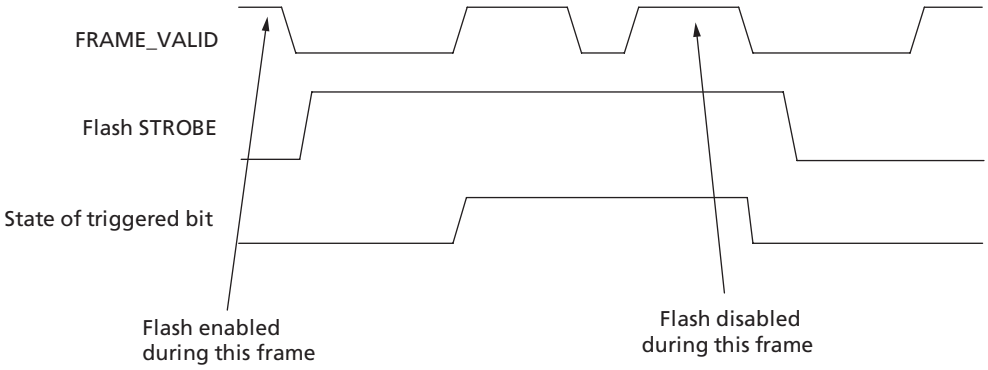
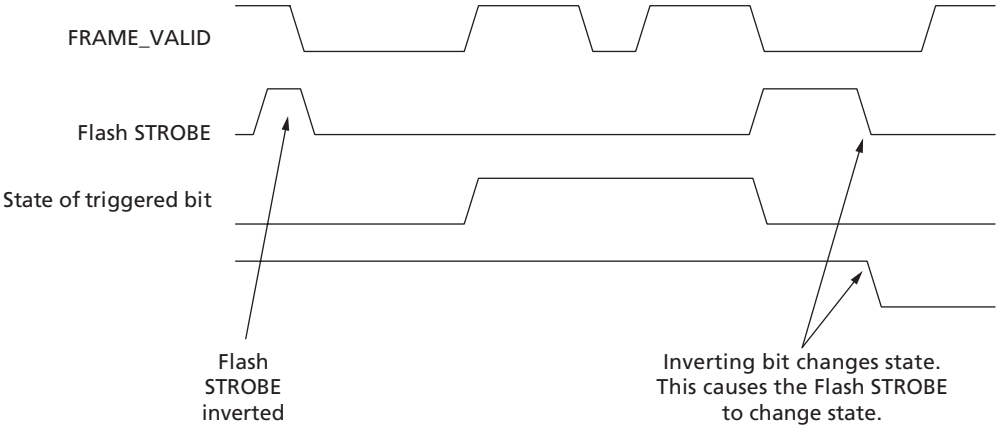


Figure 20: LED Flash Using Inverted STROBE



Auto Exposure

The AE 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.

AE is implemented by a firmware driver that analyzes image statistics collected by the exposure measurement engine, and then programs the sensor core and color pipeline to achieve the desired exposure.

To achieve the required amount of exposure, the AE driver adjusts the sensor integration time, gains, ADC reference, and IFP digital gains. In addition, the overall brightness of the scene can be adjusted by the user.

The MT9V113 includes several AE algorithms, two of which are:

1. Average brightness tracking (ABT)
   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.
2. Dynamic range tracking (DRT)
   The dynamic range tracking AE examines the data from the statistics engine and produces a corrected brightness target so that overexposed or underexposed scene can be avoided. This also allows a manual control of the image brightness.

Both algorithms are available in preview and capture modes.

Additionally, a scene-evaluative AE algorithm is available for use in snapshot mode. The algorithm performs scene analysis and classification with respect to its brightness, contrast, and composure and then decides to increase, decrease, or keep the original exposure target. It makes the most difference for backlight and bright outdoor conditions.

Auto White Balance

The MT9V113 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, digital, and sensor core analog gains. 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.

Flicker Detection

Flicker occurs when the integration time is not an integer multiple of the period of the light intensity. The automatic flicker detection block does not compensate for the flicker, but rather avoids it by detecting the flicker frequency and adjusting the integration time. For integration times below the light intensity period (10ms for 50Hz environment), flicker cannot be avoided.

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 MT9V113 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:

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

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

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

Uncompressed YUV/RGB Data Ordering

The MT9V113 supports swapping YCbCr mode, as illustrated in Table 5.

Table 5: 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 6. 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 6: RGB Ordering in Default Mode

Mode (Swap Disabled)	Byte	$D_7D_6D_5D_4D_3D_2D_1D_0$
565RGB	Odd	$R_7R_6R_5R_4R_3G_7G_6G_5$
	Even	$G_4G_3G_2B_7B_6B_5B_4B_3$
555RGB	Odd	$0\ R_7R_6R_5R_4R_3G_7G_6$
	Even	$G_4G_3G_2B_7B_6B_5B_4B_3$
444xRGB	Odd	$R_7R_6R_5R_4G_7G_6G_5G_4$
	Even	$B_7B_6B_5B_4\ 0\ 0\ 0\ 0$
x444RGB	Odd	$0\ 0\ 0\ 0\ R_7R_6R_5R_4$
	Even	$G_7G_6G_5G_4B_7B_6B_5B_4$

Uncompressed 10-Bit Bypass Output

Raw 10-bit Bayer data from the sensor core can be output in bypass mode in two ways:

1. Using both DOUT[7:0] and GPIO[1:0]. The GPIO signals are the lowest 2 bits of data.
2. Using only DOUT[7:0] with a special 8 + 2 data format, shown in Table 7.

Table 7: 2-Byte RGB Format

Byte	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	$0\ 0\ 0\ 0\ 0\ 0\ D_1D_0$

Table 8: Data Formats Supported by MIPI Interface

Data Format	Data Type
YUV 422 8-bit	0x1E
565RGB	0x22
555RGB	0x21
444RGB	0x20
RAW8	0x2A

Table 8: Data Formats Supported by MIPI Interface (continued)

Data Format	Data Type
RAW10	0x2B
RAW12	0x2C
RAW14	0x12
User-defined byte-based data (including compressed data)	0x30 0x31 0x32 0x33

Note: Data will be packed as RAW8 if the data type specified does not match any of the above data types.

Register and Variable Description

There are two methods to change internal registers and RAM variables of MT9V113:

1. The MCU firmware driver
2. The two-wire serial interface through the external host device

The firmware for the MT9V113 is implemented in multiple drivers that are responsible for different parts of operation, as shown in Figure 21.

The sequencer is responsible for coordinating all events triggered by the user.

Figure 21: Software Architecture

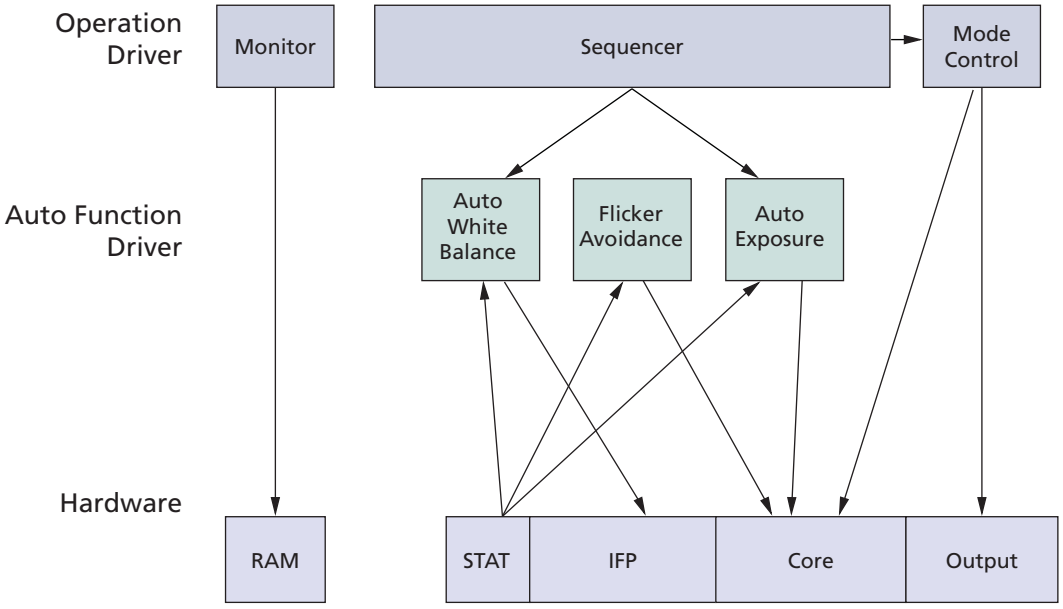


Table 9: Summary of MT9V113 Registers and Variables

Registers			
	Address	**Map**	**Functions**
	0x3000–0x31FE	Core	Sensor core
	0x3200–0x33FE	SOC1	IFP, Parallel Output
	0x3400–0x37BE	SOC2	MIPI, PGA Coefficient
	0x0000–0x0050	SYSCTL	PLL, power, standby
	0x0982–0x099E	XDMA	RAM address pointer
	0x1040–0x10FD	GPIO	GPIO
Variables			
ID Number	**Offset Address**	**Map**	**Function**
1	0x0000–0x00FF	Sequencer	IFP operation
2	0x0000–0x00FF	Auto Exposure	Auto Exposure
3	0x0000–0x00FF	Auto White Balance	Auto White Balance
4	0x0000–0x00FF	Anti-Flicker	Anti-Flicker
7	0x0000–0x00FF	Mode	Context Switch, Image Size
11	0x0000–0x00FF	Histogram	Gamma

Note: For details on registers and variables, see the MT9V113 Register and Variable Reference.

How to Access Registers and Variables

Registers and variables are accessed in different ways.

Registers

All the registers shown in Table 9 can be accessed by the two-wire serial interface with 16-bit addresses and 16-bit data; however the GPIO registers are treated as variables for writing to and reading from as explained in the “Variables” section.

“Two-Wire Serial Interface” on page 32 describes the interface protocol of the two-wire serial interface in more detail. The 6811 microcontroller can also access all the registers through physical address as shown in Table 9.

Variables

Variables are located in the microcontroller RAM memory. Each driver, such as auto exposure, auto white balance, and flicker detection, has a unique driver ID (0...31) number and a set of variables associated with that driver. Each variable associated with that driver is uniquely identified by its offset.

All driver variables and GPIO registers can be accessed through the mcu_variable_address and mcu_variable_data registers in the XDMA register map. The variables are accessed using a logical address. This address, which is set in the mcu_variable_address register, consists of a 5-bit driver ID number and a 8-bit variable offset.

Reserved

All the reserved bits should not be changed. The user should write the original values back when changing the registers.

Two-Wire Serial Interface

The two-wire serial interface bus enables read and write access to control and status registers within the MT9V113. This interface is designed to be compatible with the MIPI Alliance Standard for Camera Serial Interface 2 (CSI-2) 1.0, which uses the electrical characteristics and transfer protocols of the two-wire serial interface specification.

The interface protocol uses a master/slave model in which a master controls one or more slave devices. The MT9V113 always operates in slave mode. The host (master) generates a clock (SCLK) that is an input to the MT9V113 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.

MT9V113 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 MT9V113 are R0x078 (write address) and R0x079 (read address). If the SADDR signal is driven HIGH, then addresses used by the MT9V113 are R0x07A (write address) and R0x07B (read address).

Messages

Message bytes are used for sending MT9V113 internal register addresses and data. The host should always use 16-bit address (two bytes) and 16-bit data to access internal registers. Refer to READ and WRITE cycles in Figure 22 on page 33 through Figure 26 on page 35.

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. For data transfers, SDATA can change when SCLK is LOW and must be stable while SCLK is HIGH.

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

Typical Operation

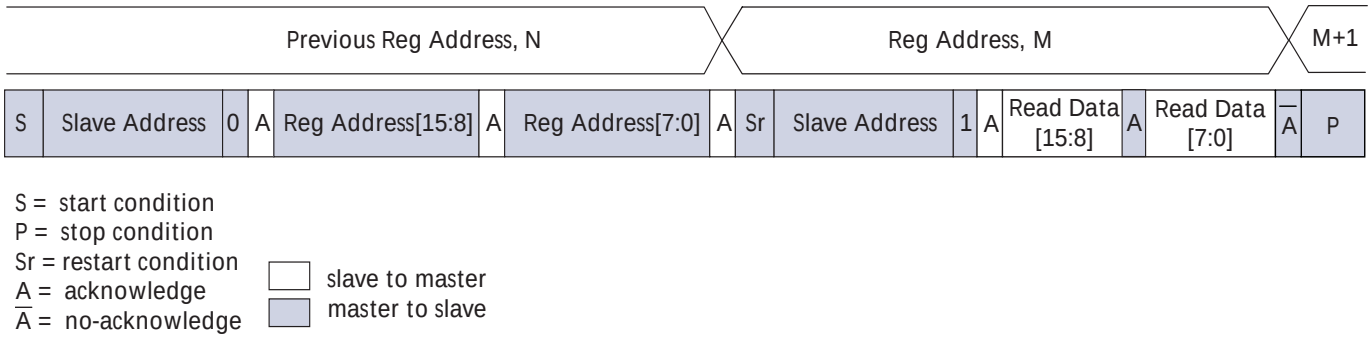
A typical READ or WRITE sequence begins by the master generating a start condition on the bus. After the start condition, the master sends the 8-bit slave address/data direction byte. The last bit indicates whether the request is for a READ or a WRITE, where a "0" indicates a WRITE and a "1" indicates 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 22 shows the typical READ cycle of the host to MT9V113. The first 2 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.

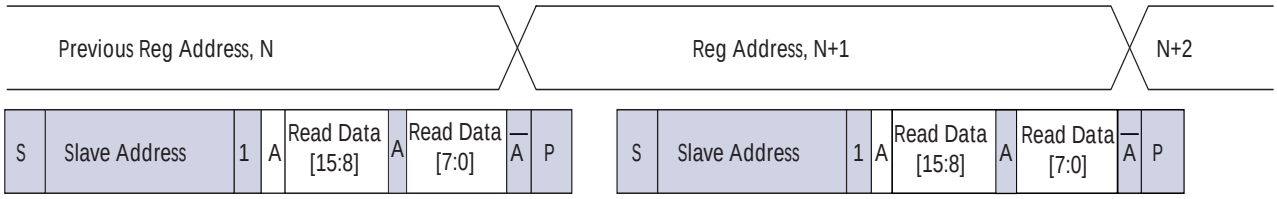
Figure 22: Single READ from Random Location



Single READ from Current Location

Figure 23 shows the single READ cycle without writing the address. The internal address will use the previous address value written to the register.

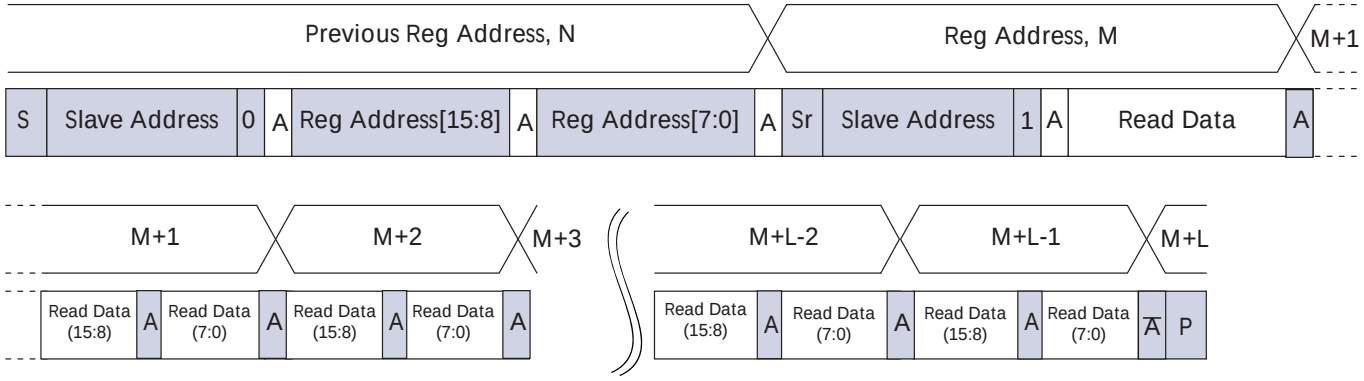
Figure 23: Single Read from Current Location



Sequential READ, Start from Random Location

This sequence (Figure 24) starts in the same way as the single READ from random location (Figure 22 on page 33). 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.

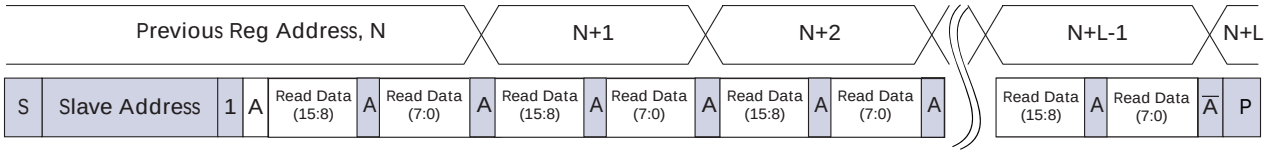
Figure 24: Sequential READ, Start from Random Location



Sequential READ, Start from Current Location

This sequence (Figure 25) starts in the same way as the single READ from current location (Figure 23). 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.

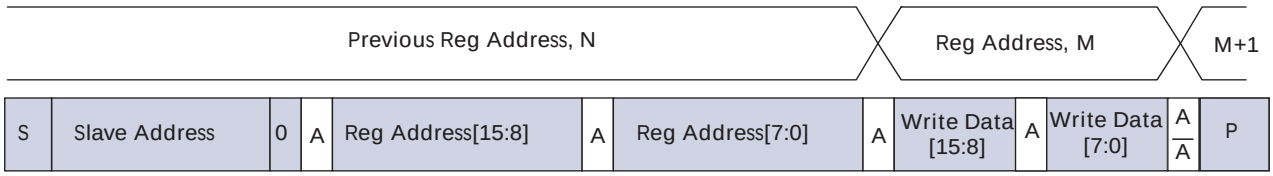
Figure 25: Sequential READ, Start from Current Location



Single Write to Random Location

Figure 26 shows the typical WRITE cycle from the host to the MT9V113. 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.

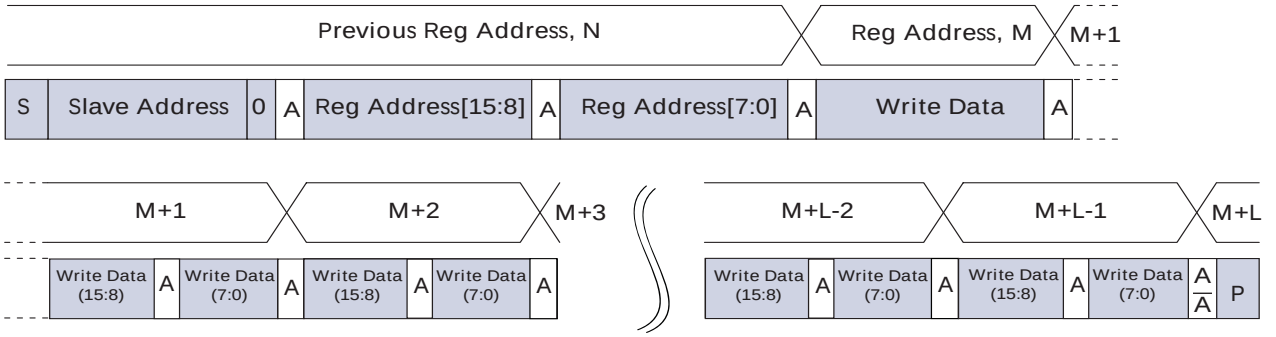
Figure 26: Single WRITE to Random Location



Sequential WRITE, Start at Random Location

This sequence (Figure 27) starts in the same way as the single WRITE to random location (Figure 26). 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 27: Sequential WRITE, Start at Random Location



Spectral Characteristics

Figure 28: Chief Ray Angle (CRA) vs. Image Height

CRA vs. Image Height Plot

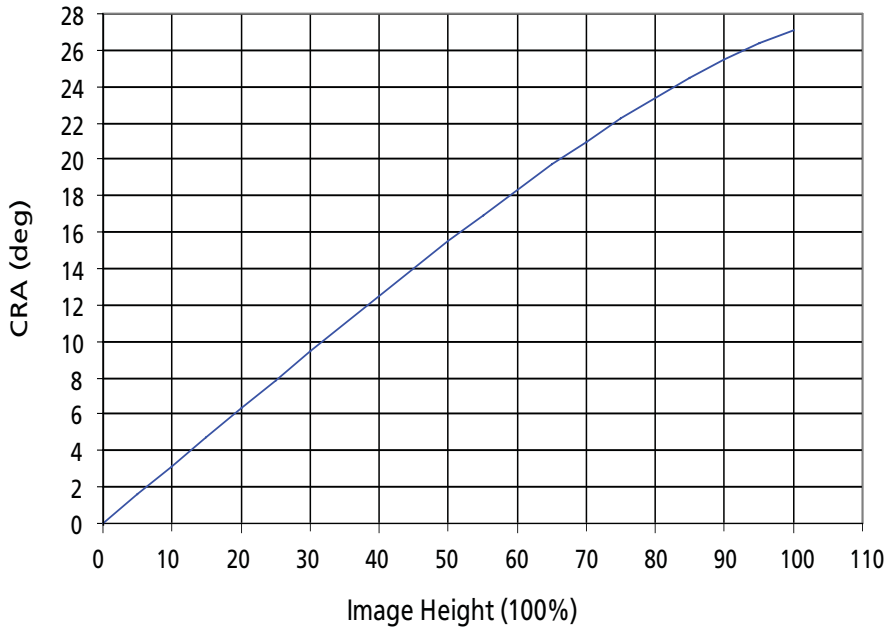
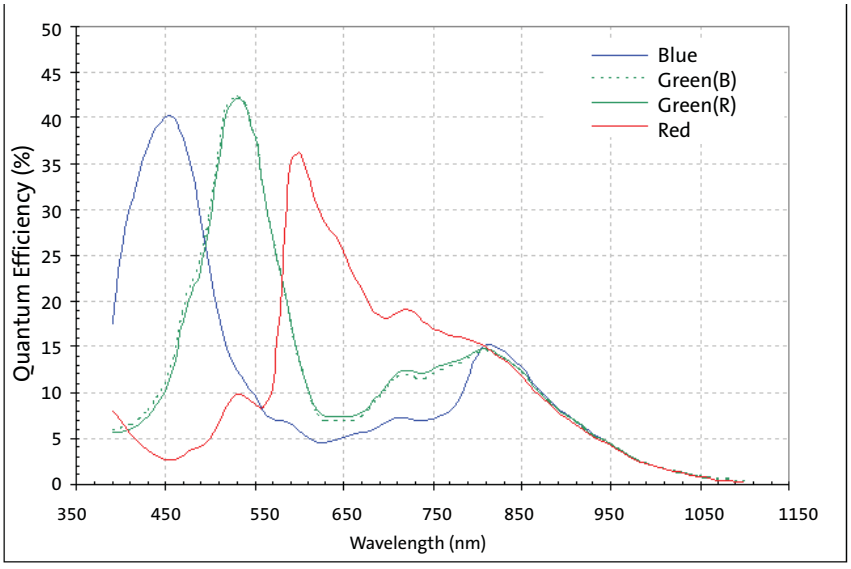


Image Height		CRA
(%)	(mm)	(deg)
0	0	0
5	0.044	1.59
10	0.088	3.17
15	0.132	4.75
20	0.176	6.33
25	0.220	7.90
30	0.264	9.47
35	0.308	11.01
40	0.352	12.53
45	0.396	14.04
50	0.440	15.52
55	0.484	16.96
60	0.528	18.35
65	0.572	19.70
70	0.616	21.00
75	0.660	22.23
80	0.704	23.40
85	0.748	24.48
90	0.792	25.48
95	0.836	26.34
100	0.880	27.10

Figure 29: Quantum Efficiency



Electrical Specifications

Caution **Stresses above those listed in Table 10 may cause permanent damage to the device.**

Table 10: Absolute Maximum Ratings

Symbol	Parameter	Rating		Unit
		Min	Max	
V_{DD}	Core digital voltage	–0.3	2.4	V
V_{DD_IO}	I/O digital voltage	–0.3	4.0	V
V_{AA}	Analog voltage	–0.3	4.0	V
VAA_PIX	Pixel supply voltage	–0.3	4.0	V
$V_{DD\_PLL\_Max}$	PLL supply voltage	–0.3	4.0	V
V_{DD_PHY}	PHY supply voltage	–0.3	2.4	V
V_{IN}	Input voltage	–0.3	VDD_IO + 0.3	V
T_{OP}	Operating temperature (measure at junction)	–30	70	°C
T_{STG}[1]	Storage temperature	–40	85	°C

Note: 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 11: Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units
V_{DD}	Core digital voltage	1.7	1.8	1.95	V
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
VAA_PIX	Pixel supply voltage	2.5	2.8	3.1	V
V_{DD_PLL}	PLL supply voltage	2.5	2.8	3.1	V
V_{DD_PHY}	I/O MIPI interface supply voltage	1.7	1.8	1.95	V
T_J	Operating temperature (at junction)	–30	55	70	°C

Table 12: DC Electrical Characteristics

Symbol	Parameter	Condition	Min	Max	Unit
V_{IH}	Input HIGH voltage		0.7 * V_{DD_IO}	V_{DD_IO} + 0.5	V
V_{IL}	Input LOWvoltage		–0.3	0.3 * V_{DD_IO}	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	1.7	–	V
		V_{DD_IO} = 1.8V, I_{OH} = 4mA	1.6	–	V
		V_{DD_IO} = 1.8V, I_{OH} = 8mA	1.4	–	V
		V_{DD_IO} = 2.8V, I_{OH} = 2mA	2.7	–	V
		V_{DD_IO} = 2.8V, I_{OH} = 4mA	2.6	–	V
		V_{DD_IO} = 2.8V, I_{OH} = 8mA	2.5	–	V
V_{OL}	Output LOW voltage	V_{DD_IO} = 1.8V, I_{OH} = 2mA	–	0.1	V
		V_{DD_IO} = 1.8V, I_{OH} = 4mA	–	0.2	V
		V_{DD_IO} = 1.8V, I_{OH} = 8mA	–	0.4	V
		V_{DD_IO} = 2.8V, I_{OH} = 2mA	–	0.1	V
		V_{DD_IO} = 2.8V, I_{OH} = 4mA	–	0.2	V
		V_{DD_IO} = 2.8V, I_{OH} = 8mA	–	0.4	V

Table 13: Operating/Standby Current Consumption

^{f}EXTCLK = 48 MHz; ^{f}PIXCLK = 28 MHz; voltages = Typ or Max; V_{DD}_MIPI = V_{DD}; T_J = Typ or Max; excludes V_{DD}_IO current

Symbol	Parameter	Condition	Typ	Max	Unit
I_{DD}	Digital operating current	Context A	11	16	mA
I_{AA}	Analog operating current		10	12	mA
I_{AA_PIX}	Pixel supply current		0.5	0.6	mA
I_{DD_PLL}	PLL supply current		9	13	mA
	Total supply current		30	42	mA
	Total power consumption		74	111	mW
I_{DD}	Digital operating current	ContextB	11	16	mA
I_{AA}	Analog operating current		10	12	mA
I_{AA_PIX}	Pixel supply current		0.5	0.6	mA
I_{DD_PLL}	PLL supply current		9	13	mA
	Total supply current		30	42	mA
	Total power consumption		74	111	mW
Hard standby	Total standby current when asserting the STANDBY signal		12	15	µA
	Standby power		29	41	µW
Soft standby (clock on)	Total standby current	^{f}EXTCLK = 48 MHz, Soft standby mode as defined in Developer Guide	6	7.5	mA
	Standby power		11	14	mW
Soft standby (clock off)	Total standby current	Soft standby mode as defined in Developer Guide	12	15	µA
	Standby power		29	41	µW

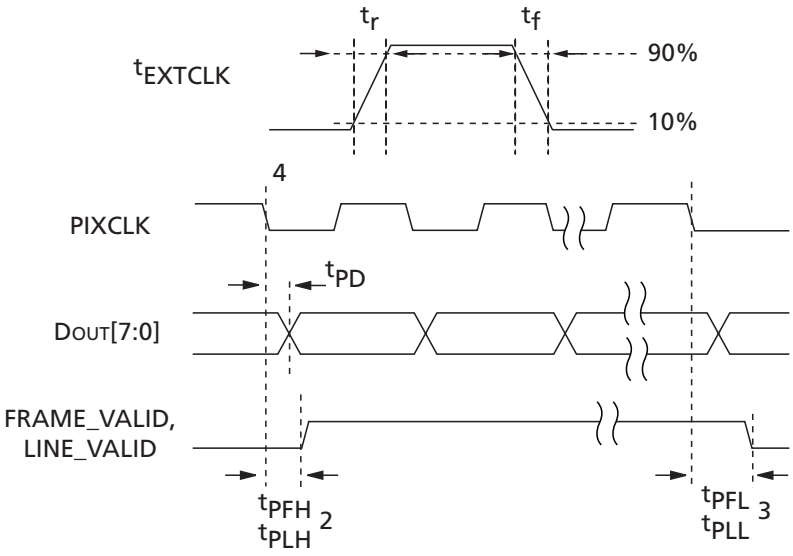
Table 14: AC Electrical Characteristics

^{f}EXTCLK = 6–48 MHz; V_{DD} = 1.8V; V_{DD_IO} = 1.8V–2.8V; V_{AA} = 2.8V; V_{AA_PIX} = 2.8V; V_{DD_PLL} = 2.8V; C_{LOAD} = 50pF

Symbol	Parameter	Conditions		Min	Typ	Max	Unit	Note
^{f}EXTCLK	External clock frequency	PLL enabled		6	14	48	MHz	
^{t}R	Input clock rise time			–	–	5	ns	
^{t}F	Input clock fall time				–	5	ns	
	Clock duty cycle			45	–	55	%	
^{t}JITTER	Input clock jitter (peak-to-peak jitter)				–	1	ns	
Output signal slew	Rise and fall time of parallel output signals (PIXCLK, FV, LV, DOUT) with slew rate programmed to 7. See SYSCTL register 0x001E.	V_{DD_IO} = 2.8V Input clock = 48 Mhz	C_{LOAD} = 30pf	–	3	–	ns	
			C_{LOAD} = 50pf	–	4	–	ns	
	Rise and fall time of parallel output signals (PIXCLK, FV, LV, DOUT) with slew rate programmed to 4. See SYSCTL register 0x001E.	V_{DD_IO} = 2.8V Input clock = 48 Mhz	C_{LOAD} = 30pf	–	4	–	ns	
			C_{LOAD} = 50pf	–	5		ns	
	Rise and fall time of parallel output signals (PIXCLK, FV, LV, DOUT) with slew rate programmed to 0. See SYSCTL register 0x001E.	V_{DD_IO} = 2.8V Input clock = 48 Mhz	C_{LOAD} = 30pf	–	9	–	ns	
			C_{LOAD} = 50pf	–	11	–	ns	
^{f}PIXCLK	PIXCLK frequency			–	–	28	MHz	1
^{t}PIXCLK_JITTER	Pixel clock jitter (output jitter, peak-to-peak)			1.3	2.1	3.7	ns	
^{t}PD	PIXCLK to data valid	Input clock = 48 MHz, C_{LOAD} = 50pF		–2	–	2	ns	
^{t}PFH	PIXCLK to FV HIGH			–	–	20	ns	
^{t}PLH	PIXCLK to LV HIGH			–	–	20	ns	
^{t}PFL	PIXCLK to FV LOW			–	–	20	ns	
^{t}PLL	PIXCLK to LV LOW			–	–	20	ns	
C_{IN}	Input pin capacitance				7	–	pF	

Note: PIXCLK output signal can be inverted internally by programming register.

Figure 30: Parallel Pixel Bus Timing Diagram



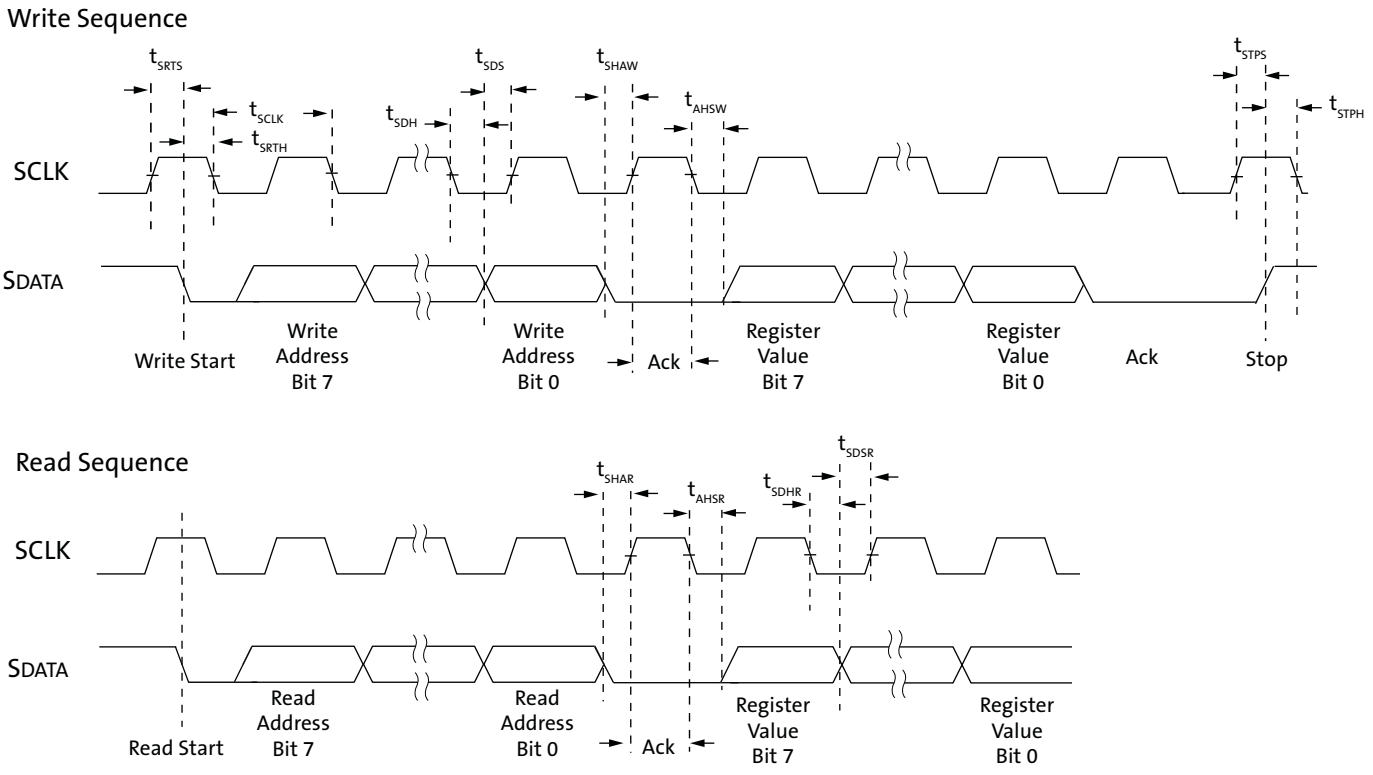
Notes:
1. PLL disabled.
2. FRAME_VALID leads LINE_VALID by 6 PIXCLKs.
3. FRAME_VALID trails LINE_VALID by 6 PIXCLKs.
4. DOUT[7:0], FRAME_VALID, and LINE_VALID are shown with respect to the falling edge of PIXCLK. This feature is programmable and DOUT[7:0], FRAME_VALID, and LINE_VALID can be synchronized to the rising edge of PIXCLK.
5. Propagation delay is measured from 50% of rising and falling edges.

Table 15: Two-Wire Serial Interface Timing Data

^{f}EXTCLK = 14 MHz; V_{DD} = 1.8V; V_{DD_IO} = 1.8V; V_{AA} = 2.8V; V_{AA_PIX} = 2.8V; V_{DD_PLL} = 2.8V; V_{DD_PHY} = NA; T_J = 70°C; C_{LOAD} = 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		33	50	50	%
^{t}r	SCLK/SDATA rise time		–	–	300	ns
^{t}SRTS	Start setup time	Master write to slave	600	–	–	ns
^{t}SRTH	Start hold time	Master write to slave	300	–	–	ns
^{t}SDH	SDATA hold	Master write to slave	5	–	900	ns
^{t}SDS	SDATA setup	Master write to slave	100	–	–	ns
^{t}SHAW	SDATA hold to ack	Master write to slave	150	–	–	ns
^{t}AHSW	Ack hold to SDATA	Master write to 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 read from slave	300	–	–	ns
^{t}AHSR	Ack hold to SDATA	Master read from slave	300	–	–	ns
^{t}SDHR	SDATA hold	Master read from slave	300	–	650	ns
^{t}SDSR	SDATA setup	Master read from slave	300	–	–	ns

Figure 31: Two-Wire Serial Bus Timing Parameters



MIPI AC and DC Electrical Characteristics

Table 16: MIPI High-Speed Transmitter DC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
V_{OD}	HS transmit differential voltage	–	175	–	mV
CMTX	HS transmit static common mode voltage	–	200	–	mV
ΔV_{OD}	VOD mismatch when output is Differential-1 or Differential-0	–	4	–	mV
Δ VCMTX(1,0)	VCMTX mismatch when output is Differential-1 or Differential-0	–	1	–	mV
V_{OHHS}	HS output HIGH voltage	–	170	–	mV
Z_{OS}	Single-ended output impedance	–	50	–	Ω
ΔZ_{OS}	Single-ended output impedance mismatch	–	3	–	%

Table 17: MIPI High-Speed Transmitter AC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
	Data bit rate	–	–	224	Mb/s
t_{rise}	20–80% rise time	–	380	–	ps
t_{fall}	20–80% fall time	–	380	–	ps
VCMTX(LF)	Common-level variations between 50–450 MHz	–	15	–	mV peak

Table 18: MIPI Low-Power Transmitter DC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
V_{OL}	Thevenin output low level	–	8	–	mV
V_{OH}	Thevenin output high level	–	1.2	–	V
Z_{OLP}	Output impedance of LP transmitter	–	105	–	Ω

Table 19: MIPI Low-Power Transmitter AC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
t_{rise}	15–85% rise time	–	6	–	ns
t_{fall}	15–85% fall time	–	12	–	ns
Slew	Slew rate (C_{LOAD} 5–20pf)	–	270	–	mV/ns
Slew	Slew rate (C_{LOAD} 20–70pf)	–	170	–	mV/ns

Revision History

Rev. L 10/25/11

- Updated two-wire serial interface timing data in Table 15 on page 42
- Removed Map Table and Description Table columns from Table 9 on page 31 and added note to refer to MT9V113 Register and Variable Reference

Rev. K 4/2/10

- Deleted section on Package Dimensions

Rev. J 2/11/10

- Removed registers and transferred to register reference

Rev. I 6/19/09

- Updated to Aptina Imaging Corporation template
- Updated fourth paragraph of "Standby" on page 10
- Added section on Package Dimensions

Rev. H 7/31/2008

- Added consumer electronics to "Applications" on page 1
- Updated typical power consumption in operating mode from 70mW to 74mW in Table 1, "Key Performance Parameters," on page 1
- Updated Figure 7 on page 20 to change G1 to Gr and G2 to Gb
- Changed column heading of Table 10 on page 36 through Table 21 on page 46 from "Register Description" to "Register Name"
- Added note to "Register Summary" on page 36 explaining correspondence of Green1 and Green2 to GreenR and GreenB

Rev.G 7/18/2008

- Updated power consumption in Table 1, "Key Performance Parameters," on page 1
- Added R99 and R100 to Table 17, "Driver ID = 3: Auto White Balance Variables," on page 42 and Table 29, "Driver ID = 3: Auto White Balance Variable Descriptions," on page 73
- Changed typical T_J from 25 to 55 in Table 11 on page 37
- Consolidated Tables 36, "Typical Operating Current (VDD_IO 1.7 – 1.9V)")through Table 41, "Hardware Standby Current with Clock Off (I/O Voltage Up to 3.1V)" into Table 12, "DC Electrical Characteristics," on page 38 and Table 13, "Operating/ Standby Current Consumption," on page 39
- Rounded off numbers in Table 13, "Operating/Standby Current Consumption," on page 39
- Rounded off ^{f}PIXCLK in Table 14 on page 40 to one decimal place
- Deleted Aptina logo on last page and left-justified legal disclaimer (to comply with new template)

Rev. F, Production 03/18/2008

- Added Figure 29: "Quantum Efficiency," on page 36
- Added Table 39, "Operating Current (with I/O Voltage Up to 3.1V)," on page 97
- Added Table 40, "Software Standby Current with Clock On (I/O Voltage Up to 3.1V)," on page 98
- Added Table 41, "Hardware Standby Current with Clock Off (I/O Voltage Up to 3.1V)," on page 99

- Deleted Max value for ^{t}JITTER and changed unit form ps to ns; added row for ^{t}PIXCLK_JITTER in Table 14, “AC Electrical Characteristics,” on page 40

Rev. E, Preliminary 11/13/2007

- Added new section: “Image Data Output Interface” on page 12
- Updated Table 1, “Key Performance Parameters,” on page 1
- Added Figure 24 and Figure 25 on page 34
- Added Figure 27 on page 35
- Updated Table 20, “Driver ID = 11: Histogram Variables,” on page 10
- Added Figure 28: “Chief Ray Angle (CRA) vs. Image Height,” on page 36
- Changed Table 36 and Table 37 on page 95 from maximum to typical
- Updated Figure 31 on page 42 and Table 15 on page 42
- Updated Table 11, “SOC1 Register Map,” on page 36
- Updated Table 17, “Driver ID = 3: Auto White Balance Variables,” on page 42
- Updated Table 20, “Driver ID = 11: Histogram Variables,” on page 10
- Updated Table 32, “Driver ID = 11: Histogram Variable Descriptions,” on page 86

Rev. D, Preliminary 5/7/2007

- Updated document from Advance to Preliminary.
- Updated data in Table 1 on page 1.
- Changed Table 36 on page 94 to maximum operating current.
- Changed Table 37 on page 95 to maximum standby current with clock on.
- Changed Table 38 on page 96 to maximum standby current with clock off.
- Changed Table 36 on page 94 to dc electrical characteristics.
- Added Table 14 on page 40: AC electrical characteristics.
- Added Table 15 on page 42: Two-wire serial bus characteristics.
- Added Table 16 on page 43 through Table 19 on page 43: MIPI AC and DC characteristics.
- Updated Figure 30 on page 96 and Figure 31 on page 42: Two-wire serial interface.
- Updated Table 23 on page 48, SOC1 register R0x321C[7]: function of bit.
- Updated Table 25 on page 59, SYSCTL register R0x0022: register description and default value and default value for SYSCTL register R0x0016.
- Updated Table 13 on page 39, default value for SYSCTL register R0x0016 .
- Updated Table 26 on page 63: Description of XDMA register R0x098C.
- Updates to other register tables throughout the document.

Rev. C, Advance 3/21/2007

- Updated Figure 2 on page 9, Figure 7 on page 16, Figure 15 on page 20, and Figure 16 on page 21 to make figures more readable.
- Minor editing to make document consistent.
- Updated Figure 1 on page 6 and Figure 15 on page 20.
- Register tables added (summary and detailed) Table 10 on page 36 through Table 33 on page 90.

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

Rev. B, Advance 1/16/2007

- Updated Figure 28: Chief Ray Angle (CRA) vs. Image Height on page 36.
- Updated Chief ray angle in Table 1, “Key Performance Parameters,” on page 1.

Rev. A, Advance 12/15/2006

- Initial release.