1/6-Inch 720p High-Definition (HD) System-On-A-Chip (SOC) Digital Image Sensor

MT9M114 Data Sheet

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

Features

- Superior low-light performance
- Ultra-low-power
- 720p HD video at 30 fps
- Internal master clock generated by on-chip phase-locked loop (PLL) oscillator
- Electronic rolling shutter (ERS), progressive scan
- Integrated image flow processor (IFP) for single-die camera module
- Automatic image correction and enhancement
- Arbitrary image scaling with anti-aliasing
- Two-wire serial interface providing access to registers and microcontroller memory
- Selectable output data format: YCbCr, 565RGB, 555RGB, 444RGB, processed Bayer, BT656, RAW8- and RAW8+2-bit
- Parallel and MIPI data output
- Independently configurable gamma correction
- Adaptive Polynomial lens shading correction
- UVC interface
- Perspective correction
- Multi-camera synchronization

Applications

- Embedded notebook, netbook, and desktop monitor cameras
- Tethered PC cameras
- Game consoles
- Cell phones, mobile devices, and consumer video communications
- Surveillance, medical, and industrial applications

General Description

Aptina 's MT9M114 is a 1/6-inch 1.26 Mp CMOS digital image sensor with an active-pixel array of 1296H x 976V. It includes sophisticated camera functions such as auto exposure control, auto white balance, black level control, flicker avoidance, and defect correction. It is designed for low light performance.The MT9M114 produces extraordinarily clear, sharp digital pictures, making it the perfect choice for a wide range of applications, including mobile phones, PC and notebook cameras, and gaming systems.

Table 1: Key Parameters

Parameter		Typical Value
Optical format		1/6-inch
Active pixels		1296 x 976= 1.26 Mp
Pixel size		1.9 µm x 1.9 µm
Color filter array		RGB Bayer
Shutter type		Electronic rolling shutter (ERS)
Input clock range		6 – 54 MHz
Output pixel clock maximum		96 MHz
Output MIPI data rate maximum		768 Mb/s
Max. Frame Rate		30 fps full res 36.7 fps 720p 75 fps VGA 120 fps QVGA[2]
Responsivity		2.24 V/lux-sec(550 nm)
SNR_{MAX}		37 dB
Dynamic range		70.8 dB
Supply voltage	Digital	1.7 – 1.95V
	Analog	2.5 – 3.1V
	I/O	1.7 – 1.95V or 2.5 – 3.1V
	PLL	2.5 – 3.1V
	PHY	1.7 – 1.95V
Power consumption[1]		135 mW
Operating temperature (ambient) -T_A		–30°C to +70° C
Chief ray angle		27.7°
Active imager size		2.46mm (H) x 1.85mm (V), 3.08mm diagonal
Package options		Bare die, CSP

Notes: 1. Power consumption for typical voltages and 720p output.
2. Reduced FOV

Ordering Information

Table 2: Available Part Numbers

Part Number	Description
MT9M114D00STCZ K24BC1	RGB color die
MT9M114EBLSTCZ	55-pin CSP
MT9M114D00STCZH ES	RGB color demo headboard
MT9M114D00STCZD ES	RGB color demo kit

Table of Contents

List of Figures

List of Tables

Functional Description

Aptina's MT9M114 is a 1/6-inch 1.26 Mp CMOS digital image sensor with an integrated advanced camera system. This camera system features a microcontroller (MCU), a sophisticated image flow processor (IFP), MIPI and parallel output ports (only one output port can be used at a time). 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 IFP will be responsible for processing and enhancing the image.

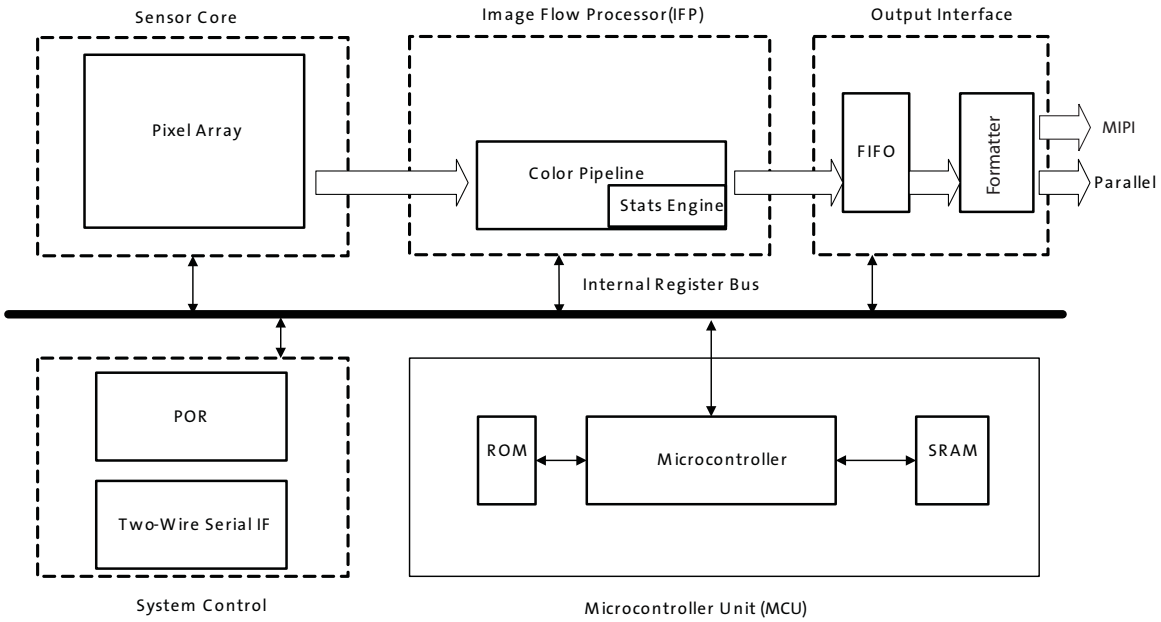
The entire system-on-a-chip (SOC) has superior low-light performance that is particularly suitable for PC camera applications. The MT9M114 features Aptina's breakthrough low-noise CMOS imaging technology that achieves near-CCD image quality (based on signal-to-noise ratio and low-light sensitivity) while maintaining the inherent size, cost, and integration advantages of CMOS.

The Aptina® MT9M114 can be operated in its default mode or programmed for frame size, exposure, gain, and other parameters. The default mode output is a 720p image size at 30 frames per second (fps), assuming a 24MHz input clock. It outputs 8-bit data, using the parallel output port.

Architecture Overview

The MT9M114 combines a 1.26 Mp sensor core with an IFP to form a stand-alone solution for both image acquisition and processing. Both the sensor core and the IFP have internal registers that can be controlled by the user. In normal operation, an integrated microcontroller autonomously controls most aspects of operation. The processed image data is transmitted to the host system either through the parallel or MIPI interface. Figure 1 shows the major functional blocks of the MT9M114.

Figure 1: MT9M114 Block Diagram



Sensor Core

The MT9M114 has a color image sensor with a Bayer color filter arrangement and a 1.2Mp 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 MT9M114 can enhance and optimize the image sensor performance. Built-in optimization algorithms enable the MT9M114 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, vertical perspective correction, aperture correction, and image formatting with cropping and scaling.

Microcontroller Unit (MCU)

The 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 MT9M114 has a phase-locked loop (PLL) oscillator that can generate the internal sensor clock from a common wireless system clock. The PLL adjusts the incoming clock frequency up, allowing the MT9M114 to run at almost any desired resolution and frame rate within the sensor's capabilities. Low-power consumption is a very important requirement.

The MT9M114 provides power-conserving features including a soft standby mode. A two-wire serial interface bus enables read and write access to the MT9M114's internal registers and variables. The internal registers control the sensor core, the color pipeline flow, and the output interface. Variables are located in the microcontroller's RAM memory and are used to configure and control the auto-algorithms and camera control functions.

Output Interface

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

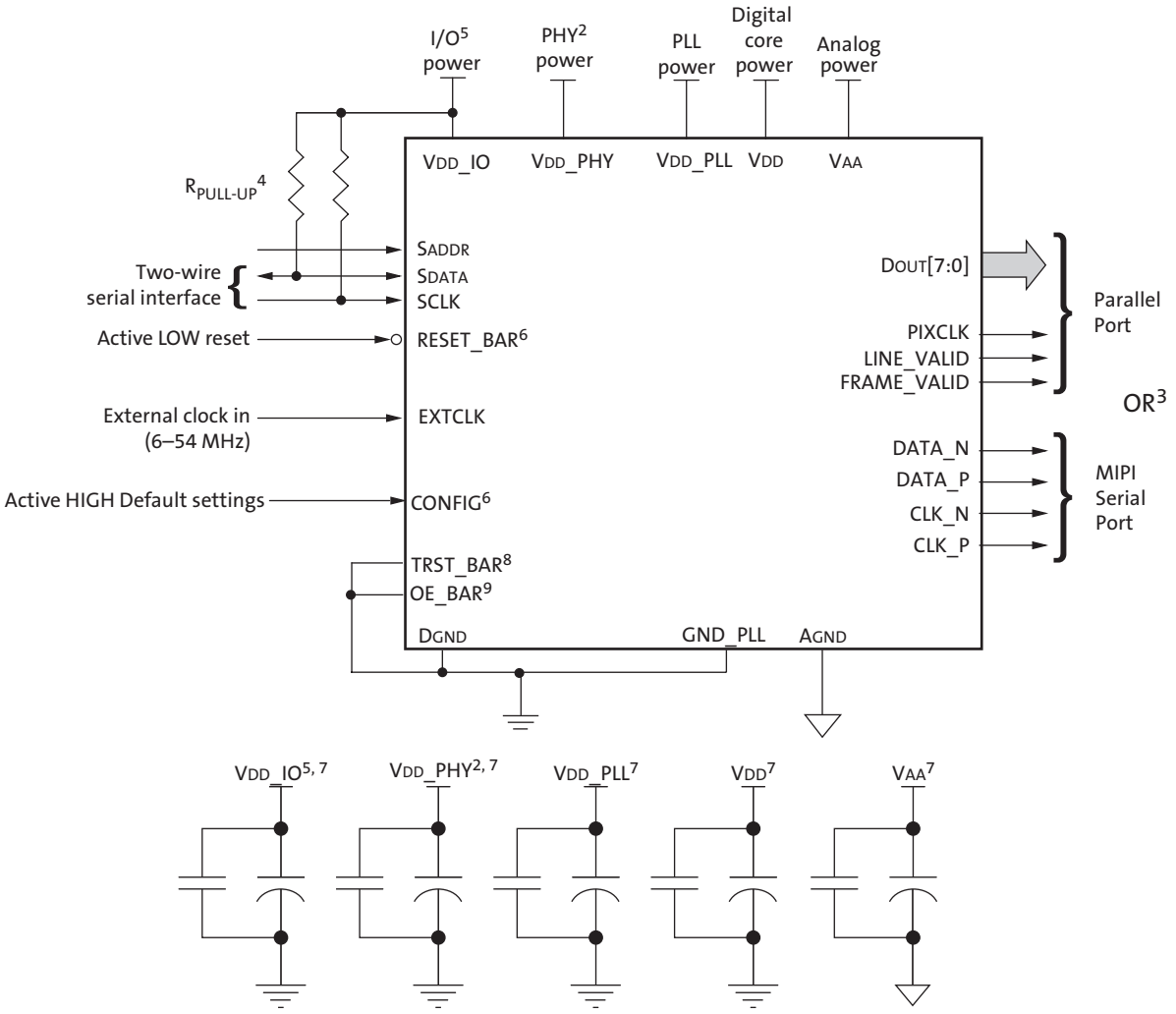
The MT9M114 also includes programmable I/O slew rate to minimize EMI.

System Interfaces

Figure 2 on page 8 shows typical MT9M114 device connections. For low-noise operation, the MT9M114 requires separate power supplies for analog and digital sections of the die. Both power supply rails must be decoupled from ground using capacitors as close as possible to the die. The use of inductance filters is not recommended on the power supplies or output signals.

The MT9M114 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 on page 9 provides the signal descriptions for the MT9M114.

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. Only one of the output modes (serial or parallel) can be used at any time.
4. 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.
5. All inputs must be configured with VDD_IO.
6. RESET_BAR and CONFIG both have internal pull-up resistors and can be left floating.
7. 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.
8. TRST_BAR connects to GND for normal operation.
9. OE_BAR should be connected HIGH when using MIPI interface.

Table 3: Pin Descriptions

Name	Type	Description	Notes
EXTCLK	Input	Input clock signal.	
RESET_BAR	Input/PU	Master reset signal, active LOW. This signal has an internal pull up.	
OE_BAR	Input	Parallel interface enable pad, active LOW.	
SCLK	Input	Two-wire serial interface clock.	
SDATA	I/O	Two-wire serial interface data.	
SADDR	Input	Selects device address for the two-wire serial interface.	
FRAME_VALID (FV)	Output	Identifies rows in the active image. Data can be sampled with PIXCLK when both LV and FV are high (except when BT656 is used).	
LINE_VALID (LV)	Output	Identifies pixels in the active line. Data can be sampled with PIXCLK when both LV and FV are high (except when BT656 is used).	
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.	
DOUT_LSB[1:0]	Output	LSBs when outputting 10-bit image data	
CLK_N	Output	Differential MIPI clock (sub-LVDS, negative).	2
CLK_P	Output	Differential MIPI clock (sub-LVDS, positive).	2
DATA_N	Output	Differential MIPI data (sub-LVDS, negative.	2
DATA_P	Output	Differential MIPI data (sub-LVDS, positive).	2
CONFIG	Input/PU	If on power-up CONFIG = 1 then the part shall go into streaming (default option, PU ensures this will occur). If CONFIG = 0 then the part will go to standby state waiting for host to update.	4
FLASH	Output	Used as a flash signal	2
CHAIN	Output/PU	To synchronize a number of sensors together.	2
TRST_BAR	Input	Must be tied to GND in normal operation.	
VDD	Supply	Digital Power	
DGND	Supply	Digital ground.	1
VDD_IO	Supply	I/O power supply.	
GND_IO	Supply	I/O ground.	
VAA	Supply	Analog power.	
AGND	Supply	Analog ground.	1
VDD_PLL	Supply	PLL Supply	
GND_PLL	Supply	PLL GND	
VDD_PHY	Supply	I/O power supply for the MIPI interface.	3

Notes:
1. AGND and DGND are not connected internally.
2. To be left floating if not using feature. If not using the feature, then there is no need to bond out the relevant pads.
3. Must always be connected even when not using MIPI.
4. When CONFIG = 1 the EXTCLK must be in the range 20-24 MHz.

Decoupling Capacitor Recommendations

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

In order of preference, Aptina recommends:

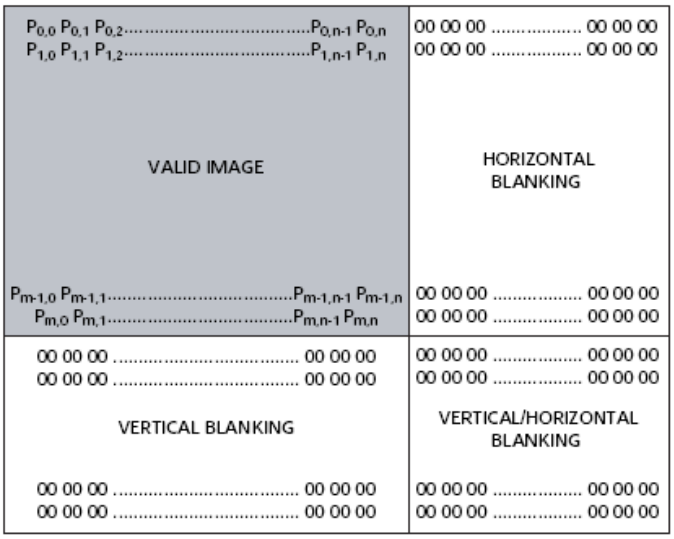
1. Mount 0.1µF and 1µF decoupling capacitors for each power supply as close as possible to the pad and place a 10µF capacitor nearby off-module.
2. If module limitations allow for only six decoupling capacitors for a three-regulator design use a 0.1µF and 1µF capacitor for each of the three regulated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

Output Data Format

The MT9M114 image data is read out in a progressive scan. Valid image data is surrounded by horizontal blanking and vertical blanking, as shown in Figure 3.

LINE_VALID is HIGH in the shaded region of the figure.

Figure 3: Spatial Illustration of Image Readout



Power-Up and Power-Down Sequence

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

Figure 4: Power-Up and Power-Down Sequence

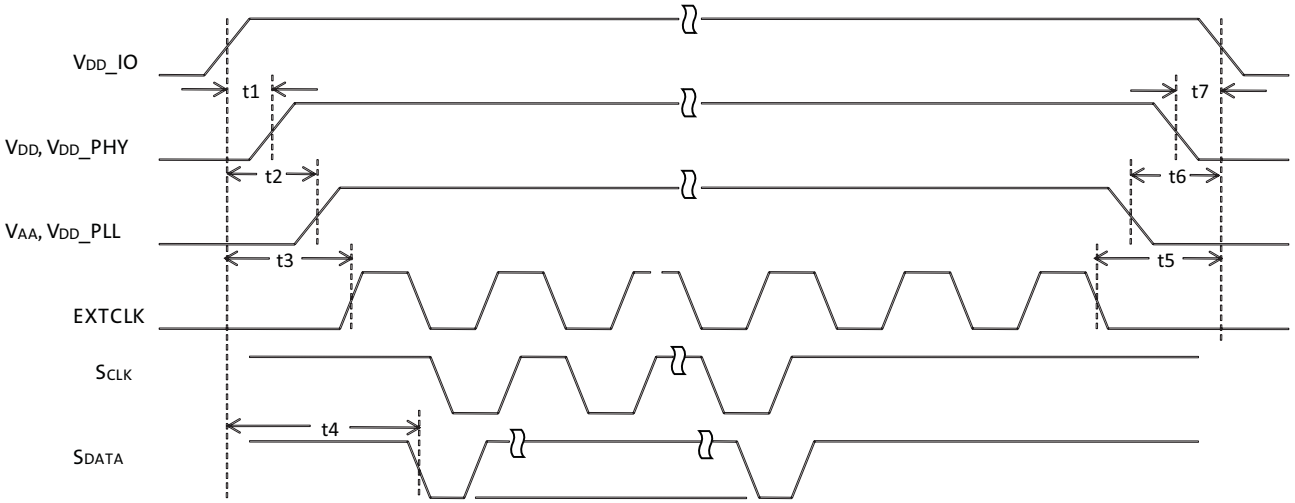


Table 4: Power-Up and Power-Down Signal Timing

Symbol	Parameter	Min	Typ	Max	Unit
t1	Delay from VDD_IO to VDD and VDD_PHY	0	–	50	ms
t2	Delay from VDD_IO to VAA and VDD_PLL	0	–	50	ms
t3	EXTCLK activation	t2	–	–	ms
t4	First serial command[1]	–	t3 + 44	–	ms
t5	EXTCLK cutoff	t6	–	–	ms
t6	Delay from VAA and VDD_PLL to VDD_IO	0	–	50	ms
t7	Delay from VDD and VDD_PHY toVDD_IO	0	–	50	ms

Notes: 1. After power-up or reset, the host should poll the Command register to determine when the device is initialized

Reset

The MT9M114 has 3 types of reset available:

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

The output states during hard reset are shown in Table 5.

Table 5: Status of Output Signals During Hard Reset, Soft Standby, and Power Off

Signal	Reset	Soft Standby (EXTCLK Running)	Soft Standby (Without EXTCLK)	Power Off
DOUT[7:0]	High-Z	High-Z	High-Z	High-Z
PIXCLK	High-Z	High-Z	High-Z	High-Z
LV	High-Z	High-Z	High-Z	High-Z
FV	High-Z	High-Z	High-Z	High-Z
DOUT_LSB[1:0]	High-Z	High-Z	High-Z	High-Z
DATA_N	0	0	0	High-Z
DATA_P	0	0	0	High-Z
CLK_N	0	0	0	High-Z
CLK_P	0	0	0	High-Z
SCLK	Input	Active	Active (Pads are active, but due to no EXTCLK serial comms will not work)	High-Z
SDATA	Input	Active	Active (Pads are active, but due to no EXTCLK serial comms will not work)	High-Z

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 and CONFIG signals have internal pull-up resistors and can be left floating.

Hard Reset

The MT9M114 enters the reset state when the external RESET_BAR signal is asserted LOW, as shown in Figure 5. All the output signals will be in High-Z state. When OE_BAR is in HIGH state, the outputs pins will be High-Z during the internal boot time

Figure 5: Hard Reset Operation

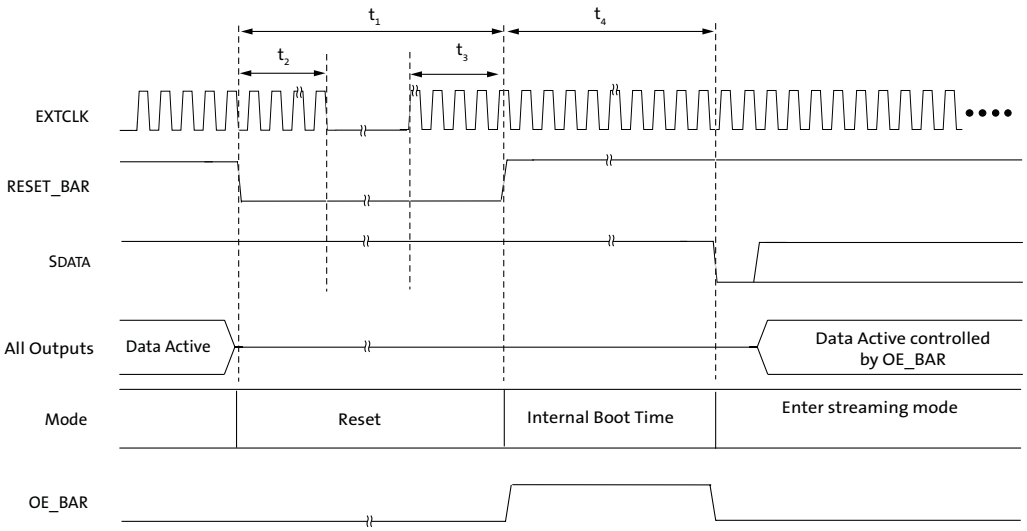


Table 6: Hard Reset

Symbol	Definition	Min	Typ	Max	Unit
t_1	RESET_BAR pulse width	50	–	–	EXTCLK cycles
t_2	Active EXTCLK required after RESET_BAR asserted	10	–	–	
t_3	Active EXTCLK required before RESET_BAR de-asserted	10	–	–	
t_4	Internal boot time[1]	–	44.5	–	ms

Notes: 1. Under the condition of EXTCLK=24MHz and default settings with CONFIG=1

Soft Reset

The host processor can reset the MT9M114 using the two-wire serial interface by writing to SYSCTL 0x001A. SYSCTL 0x001A[0] is used to reset the MT9M114 which is similar to external RESET_BAR signal.

1. Set SYSCTL 0x001A[0] to 0x1 to initiate internal reset cycle.
2. Reset SYSCTL 0x001A[0] to 0x0 for normal operation.
3. Delay of 44.5 ms.

Figure 6: Soft Reset Operation

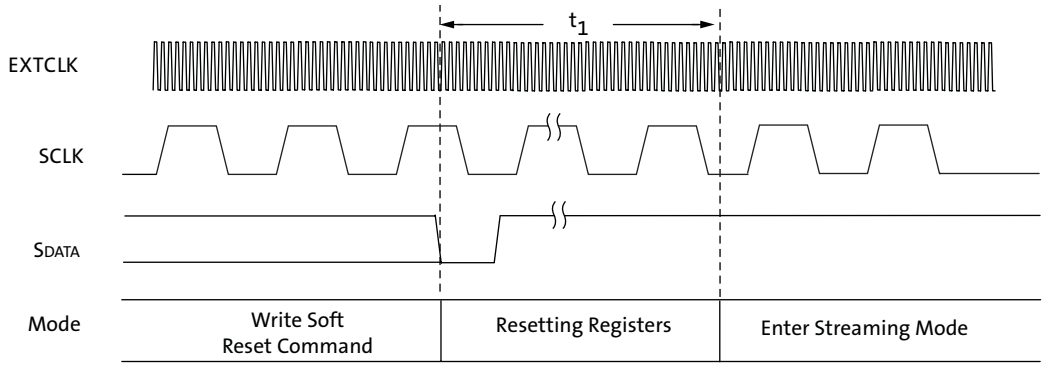


Table 7: Soft Reset Signal Timing

Symbol	Parameter	Min	Typ	Max	Unit
t_1	Soft reset time[1]	–	44.5	–	ms

Notes: 1. Under the condition of EXTCLK=24MHz

Soft Standby Mode

The MT9M114 can enter soft standby mode by receiving a host command through the two-wire serial interface. EXTCLK can be stopped to reduce the power consumption during soft standby mode. However, since two-wire serial interface requires EXTCLK to operate, Aptina recommends that EXTCLK run continuously.

Entering Standby Mode

1. Send the sequence [Enter Standby] to put the MT9M114 into standby.

```
[Enter Standby]
FIELD_WR= SYSMGR_NEXT_STATE, 0x50
// (Optional) First check that the FW is ready to accept a new command
ERROR_IF= COMMAND_REGISTER, HOST_COMMAND_1, !=0, "Set State cmd bit
is already set"
// (Mandatory) Issue the Set State command
// We set the 'OK' bit so we can detect if the command fails
// Note 0x8002 is equivalent to (HOST_COMMAND_OK | HOST_COMMAND_1)
FIELD_WR= COMMAND_REGISTER, 0x8002
```

```
// Wait for the FW to complete the command (clear the HOST_COMMAND_1
bit)

POLL_FIELD= COMMAND_REGISTER, HOST_COMMAND_1, !=0, DELAY=10, TIME-
OUT=100

// Check the 'OK' bit to see if the command was successful

ERROR_IF= COMMAND_REGISTER, HOST_COMMAND_OK, !=1, "Set State cmd
failed",

// Wait for the FW to fully-enter standby (SYSMGR_CURRENT_STATE=0x52)

POLL_FIELD= SYSMGR_CURRENT_STATE,!=0x52,DELAY=50,TIMEOUT=10
```

2. After the part is in standby for 100 EXTCLK cycles the EXTCLK can be turned off.

Exiting Standby Mode

1. Turn EXTCLK on.
2. After 100 EXTCLK cycles send the following sequence entitled [Exit Standby] to bring the MT9M114 out of standby.

```
[Exit Standby]

FIELD_WR= SYSMGR_NEXT_STATE, 0x54

// (Optional) First check that the FW is ready to accept a new command

ERROR_IF= COMMAND_REGISTER, HOST_COMMAND_1, !=0, "Set State cmd bit
is already set"

// (Mandatory) Issue the Set State command

// We set the 'OK' bit so we can detect if the command fails

// Note 0x8002 is equivalent to (HOST_COMMAND_OK | HOST_COMMAND_1)

FIELD_WR= COMMAND_REGISTER, 0x8002

// Wait for the FW to complete the command (clear the HOST_COMMAND_1
bit)

POLL_FIELD= COMMAND_REGISTER, HOST_COMMAND_1, !=0, DELAY=10, TIME-
OUT=100

// Check the 'OK' bit to see if the command was successful

ERROR_IF= COMMAND_REGISTER, HOST_COMMAND_OK, !=1, "Set State cmd
failed",

ERROR_IF= SYSMGR_CURRENT_STATE, !=0x31, "System state is not STREAM-
ING"
```

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 MT9M114 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 MT9M114 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.

MT9M114 output data is synchronized with the PIXCLK output. When LV is HIGH, one pixel value is output on the 8-bit DOUT port every TWO PIXCLK periods as shown in Figure 7. PIXCLK is continuously running, even during the blanking period. (If the user wishes to have PIXCLK turned off during blanking this is possible through a variable setting) PIXCLK phase can be varied by 50 percent, controlled using a register.

Figure 7: Pixel Data Timing Example

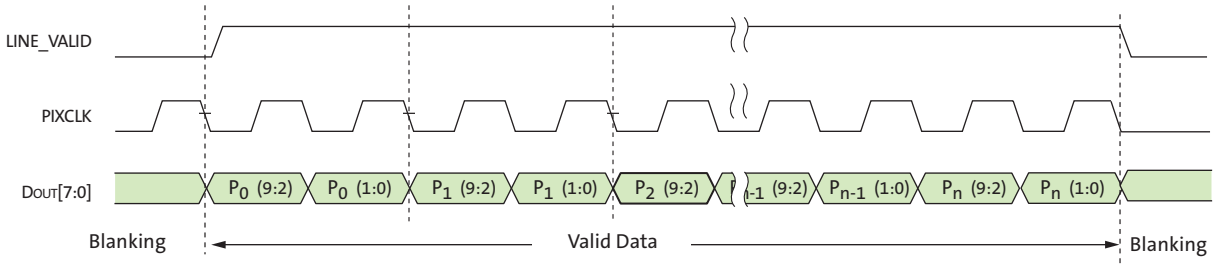
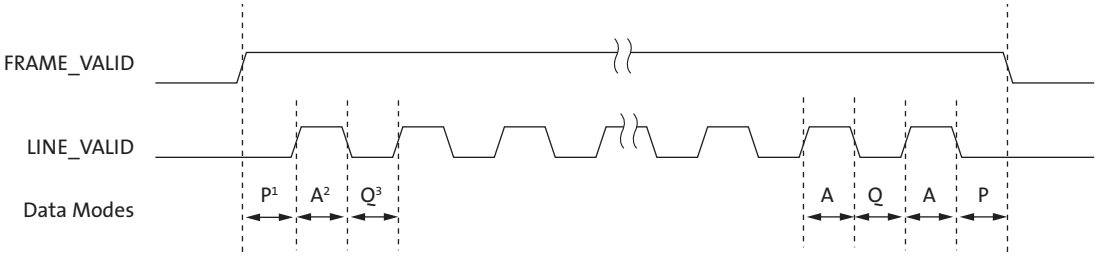


Figure 8: Row Timing, FV, and LV Signals



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

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 768 Mb/s. It supports multiple formats, error checking, and custom short packets. MT9M114 is designed to MIPI D-PHY version v1.0.

When the sensor is in the software standby system state, the MIPI signals (CLK_P, CLK_N, DATA_P, DATA_N) indicate ultra lowpower 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

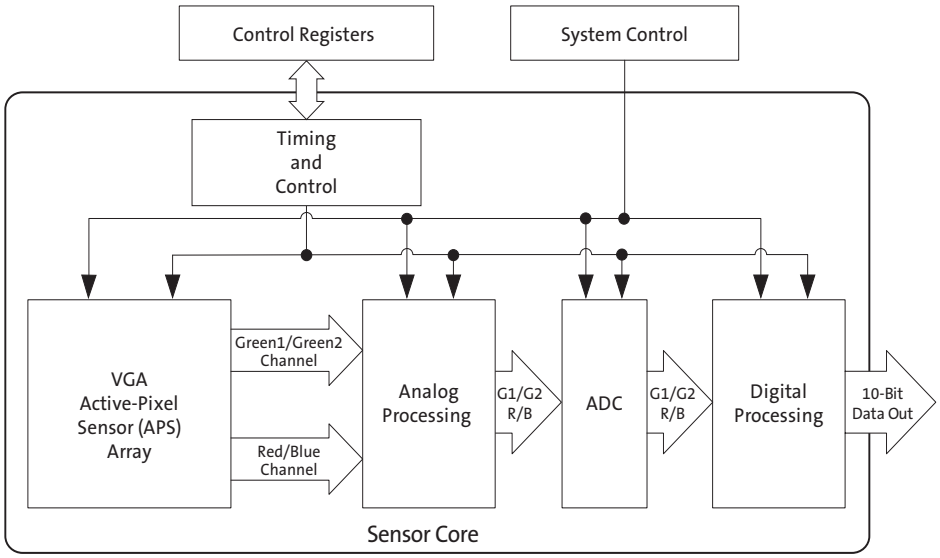
The sensor core of the MT9M114 is a progressive-scan sensor that generates a stream of pixel data at a constant frame rate. Figure 9 shows a block diagram of the sensor core. The timing and control circuitry sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and reading that row, the pixels in the row integrate incident light. The exposure is controlled by varying the time interval between reset and readout. Once a row has been selected, the data from each column is sequenced through an analog signal chain, including offset correction, gain adjustment, and ADC. The final stage of sensor core converts the output of the ADC into 10-bit data for each pixel in the array.

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

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

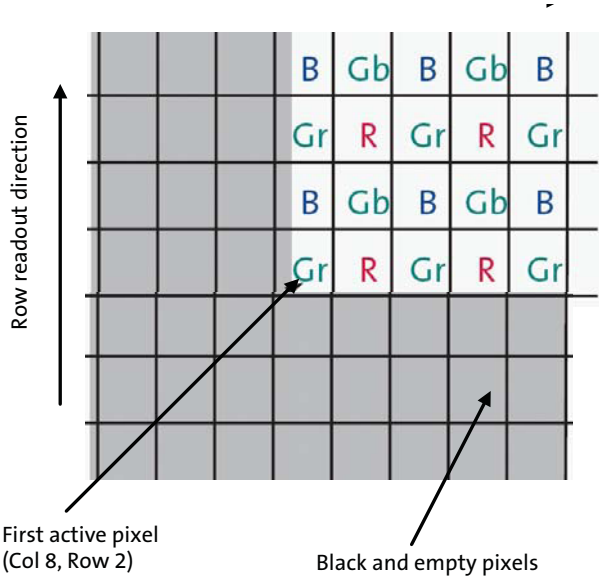
The output from the sensor core is a Bayer pattern; alternate rows are a sequence of either green and red pixels or blue and green pixels. The offset and gain stages of the analog signal chain provide per-color control of the pixel data.

Figure 9: Sensor Core Block Diagram



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

Figure 10: Pixel Color Pattern Detail (Bottom Left Corner)

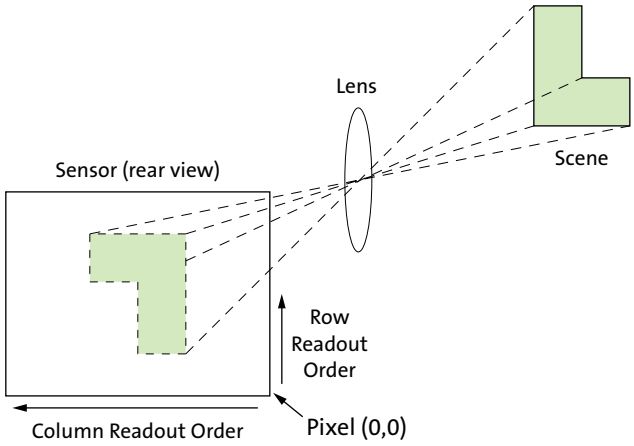


For the MT9M114 the first active pixel is defined as the first pixel that would be used as part of the demosaic border.

When the sensor is operating in a system, the active surface of the sensor faces the scene as shown in Figure 11 on page 20.

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

Figure 11: Imaging a Scene



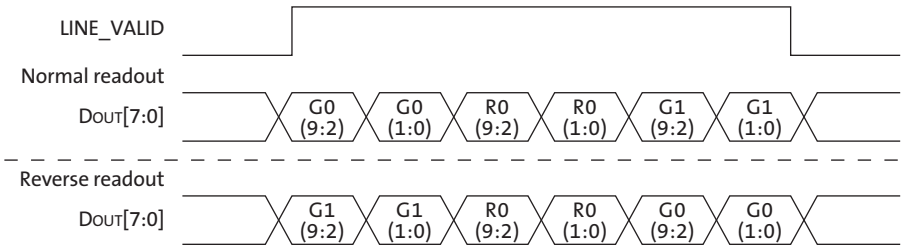
The sensor core supports different readout options to modify the image before it is sent to the IFP. The readout can be limited to a specific window size of the original pixel array.

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

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

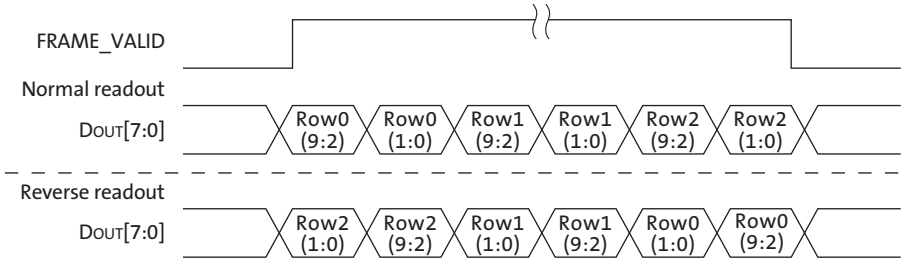
When the sensor is configured to mirror the image horizontally, the order of pixel readout within a row is reversed, so that readout starts from the last column address and ends at the first column address. Figure 12 shows a sequence of 3 pixels being read out with normal readout and reverse readout.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

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



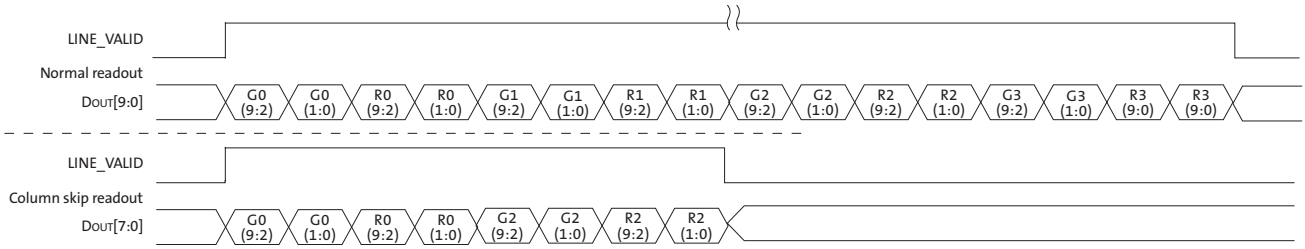
When the sensor is configured to flip the image vertically, the order in which pixel rows are read out is reversed, so that row readout starts from the last row address and ends at the first row address. Figure 13 on page 21 shows a sequence of 3 rows being read out with normal readout and reverse readout. This change in sensor core output is corrected by the IFP.

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



The MT9M114 sensor core supports subsampling with skipping to increase the frame rate. The proper image output size and cropped size must be programmed before enabling subsampling mode. Figure 14 shows the readout with 2X skipping.

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



Pixel Readouts

The following diagrams show a sequence of data being read out with no skipping. The effect of the different subsampling on the pixel array readout is shown in Figure 15 through Figure 20 on page 25.

Figure 15: Pixel Readout (No Skipping)

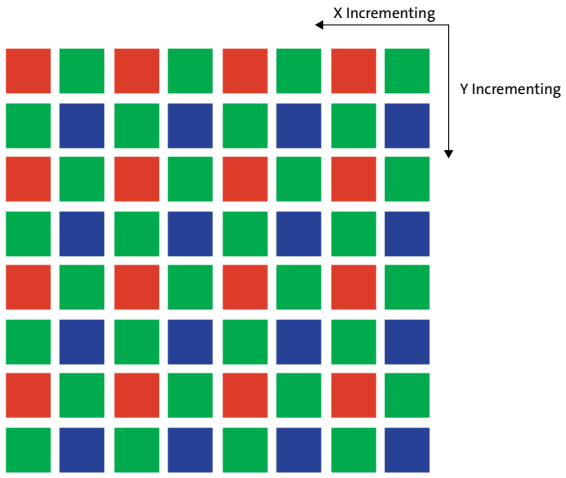


Figure 16: Pixel Readout (Column Skipping)

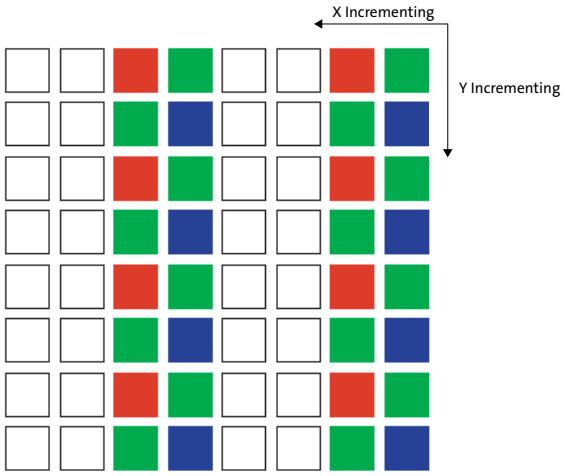


Figure 17: Pixel Readout (Row Skipping)

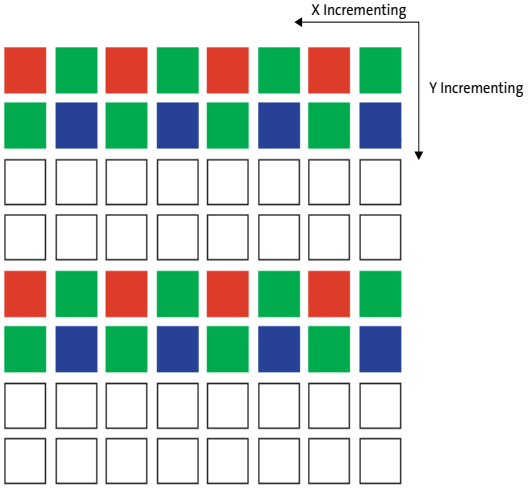
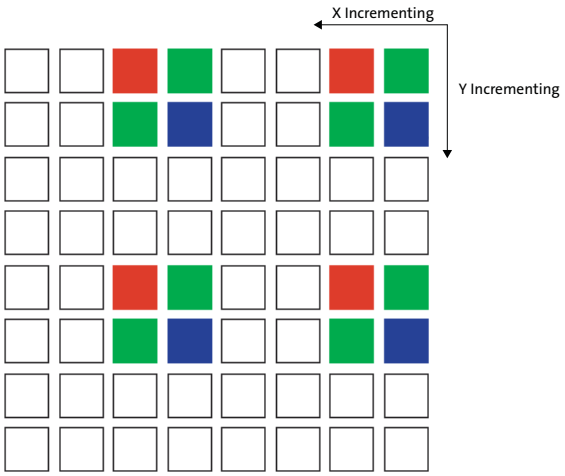


Figure 18: Pixel Readout (Column and Row Skipping)



Binning and Summing

The MT9M114 sensor core supports binning and summing. Binning has many of the same characteristics as subsampling but it gathers image data from all pixels in the active window (rather than a subset of them).

Pixel binning will sample pixels and average the value together in the analog domain. Summing will add the charge or voltage values of the neighboring pixels together. (Σe- means "charge summing", Σv- means "voltage summing", and avg means "digital averaging (post ADC). The advantage of using summing is that the pixel data is added together and up to 4X increase in responsivity is achieved.

Figure 19: Pixel Binning and Summing

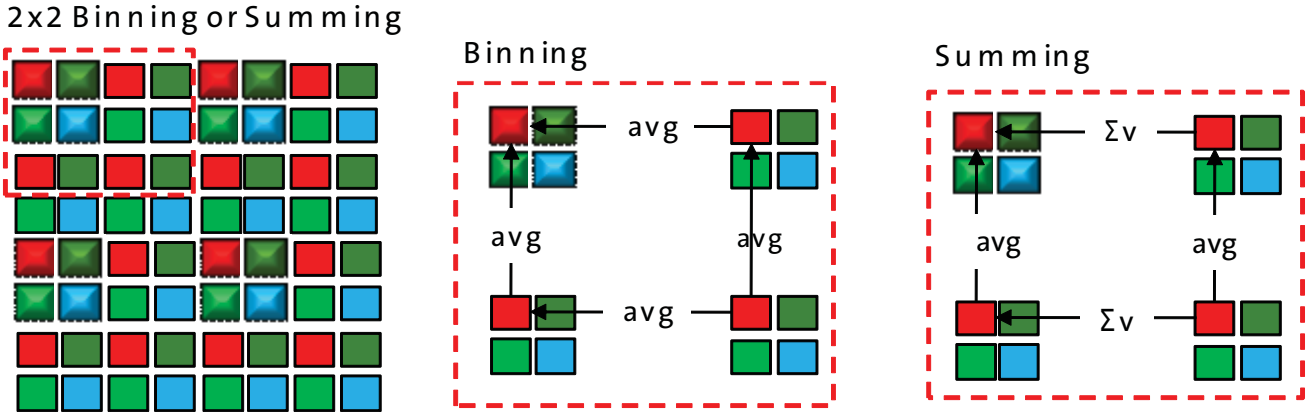


Figure 20: Pixel Readout (Column and Row Binning)

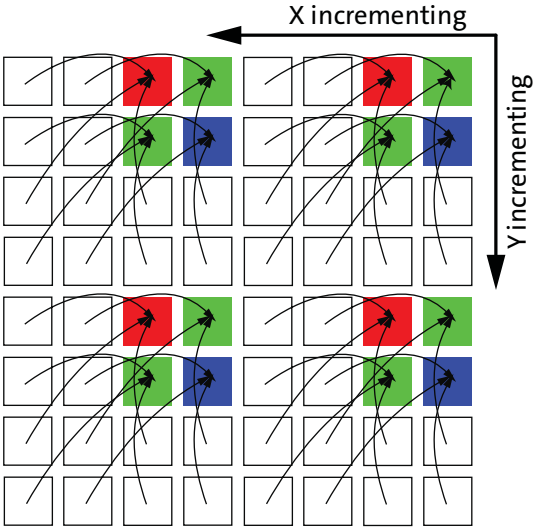
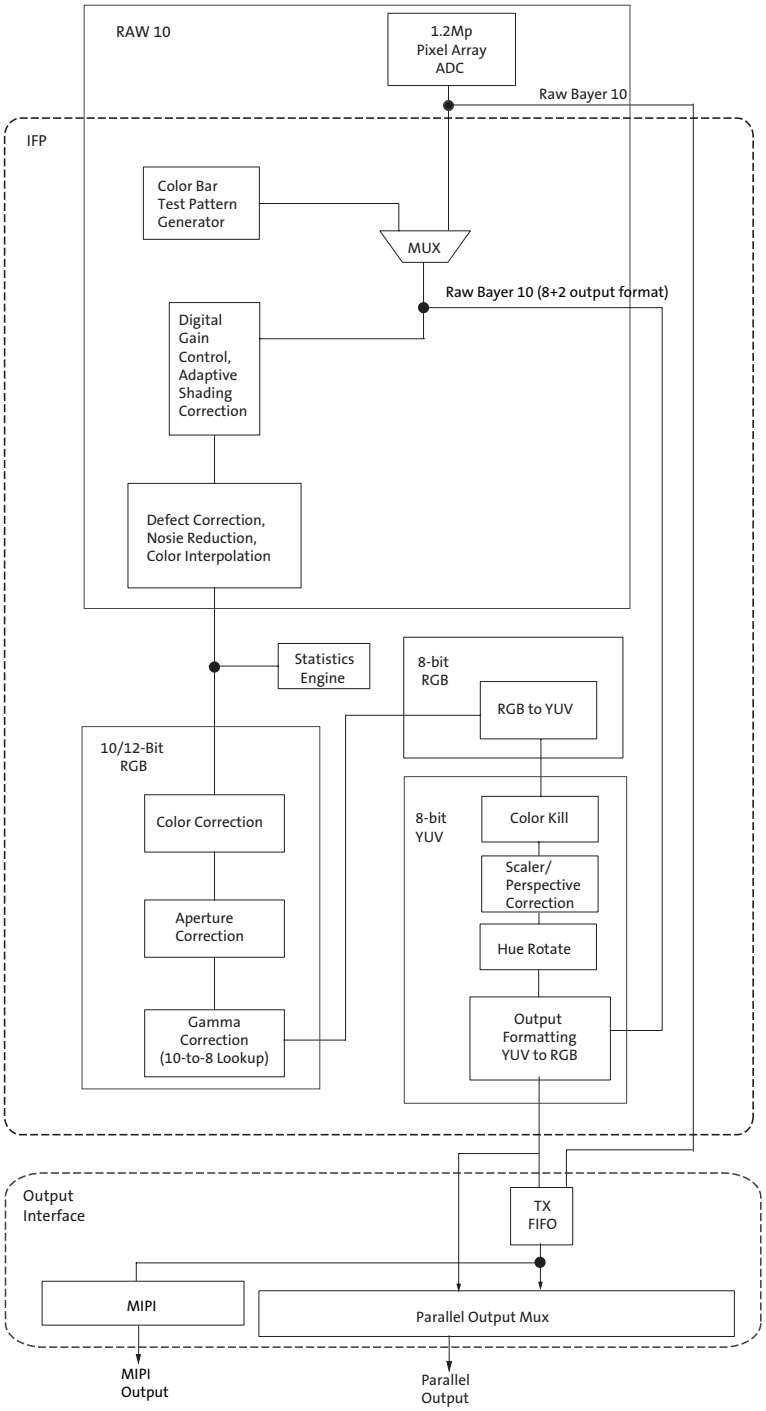


Image Flow Processor

Image control processing in the MT9M114 is implemented in the IFP hardware logic. For normal operation, the microcontroller automatically adjusts the operational parameters of the IFP. Figure 21 shows the image data processing flow within the IFP.

Figure 21: Image Flow Processor

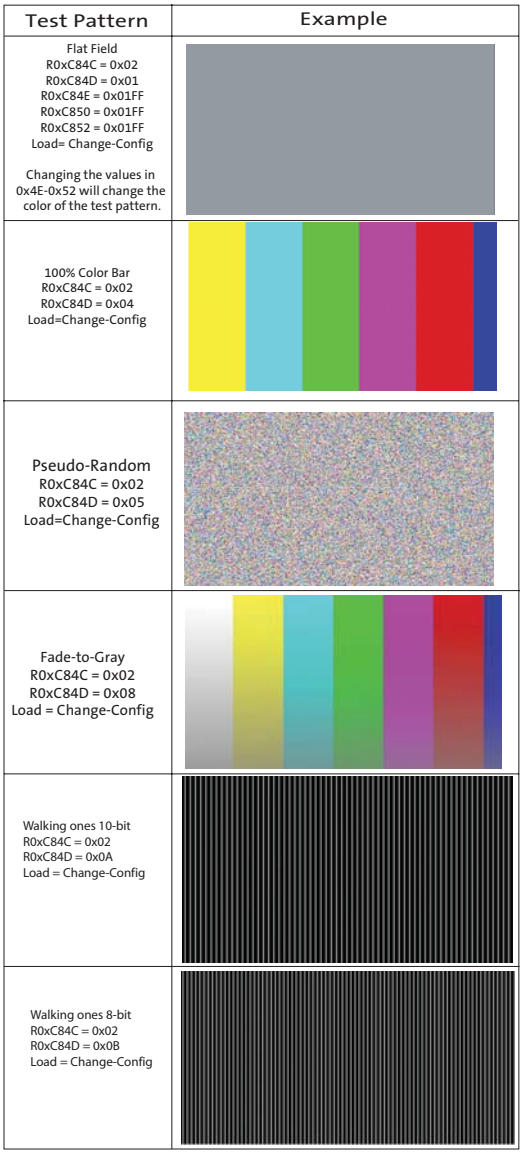


For normal operation of the MT9M114, streams of raw image data from the sensor core are continuously fed into the color pipeline. The MT9M114 features an automatic color bar test pattern generation function to emulate sensor images as shown in Figure 22: “Color Bar Test Pattern,” on page 27. The color bar test pattern is fed to the IFP for testing the image pipeline without sensor operation.

Color bar test pattern generation can be selected by programming variables. To select enter test pattern mode R0xC84C = 0x02; to exit this mode R0xC84C must be set to 0x00.

A Change-Config command needs to be issued when switching to CAM mode to enable test pattern as well as when exiting.

Figure 22: Color Bar Test Pattern



Digital Gain

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

Adaptive PGA (APGA)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. There are also other factors causing fixed pattern signal gradients in images captured by image sensors. The cumulative result of all these factors is known as image shading. The MT9M114 has an embedded shading correction module that can be programmed to counter the shading effects on each individual R, Gb, Gr, and B color signal.

In some cases, different illuminants can introduce different color shading response. The APGA feature on the MT9M114 will compensate for the dependency of the lens shading of the illuminant. The MT9M114 will allow for up to three different illuminants to be compensated.

Color Interpolation and Edge Detection

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

Color Correction and Aperture Correction

To achieve good color fidelity of the IFP output, interpolated RGB values of all pixels are subjected to color correction. The IFP multiplies each vector of three pixel colors by a 3 x 3 color correction matrix. The three components of the resulting color vector are all sums of three 10-bit numbers. Since such sums can have up to 12 significant bits, the bit width of the image data stream is widened to 12 bits per color (36 bits per pixel). The color correction matrix can either be programmed by the user or automatically selected by the AWB algorithm implemented in the IFP.

Traditionally this would have been based off two sets of CCM, one for Warm light like Tungsten and the other for Daylight (the part would interpolate between the two matrixes). This is not an optimal solution for cameras used in a Cool White Fluorescent (CWF) environment, for example when using a webcam. A better solution is to provide three CCMs, which would include a matrix for CWF (interpolation now between three matrixes). The MT9M114 offers this feature which will give the user improved color fidelity when under CWF type lighting.

Color correction should ideally produce output colors that are independent of the spectral sensitivity and color crosstalk characteristics of the image sensor. The optimal values of the color correction matrix elements depend on those sensor characteristics and on the spectrum of light incident on the sensor. The color correction settings can be adjusted using variables.

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

Gamma Correction

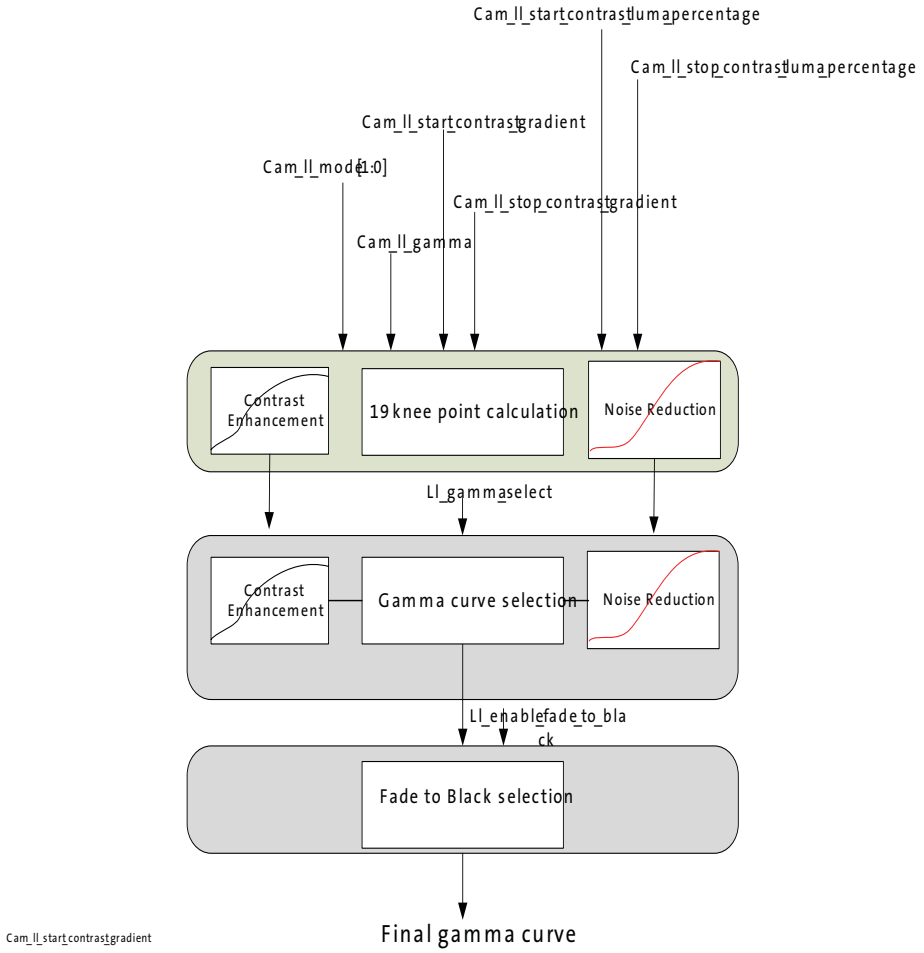
The gamma correction curve (as shown in Figure 23) is implemented as a piecewise linear function with 19 knee points, taking 12-bit arguments and mapping them to 8-bit output. The abscissas of the knee points are fixed at 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. The 8-bit ordinates are programmable through variables.

The MT9M114 IFP includes a block for gamma correction that has the capability to adjust its shape, based on brightness, to enhance the performance under certain lighting conditions.

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

The MT9M114 has the ability to calculate the 19 point knee points based on a small number of variable inputs from the host, another option is for the host to program one or both of the 19 knee points. The diagram below shows how the gamma feature interacts in MT9M114.

Figure 23: Gamma Interaction



Gamma Knee Point Calculation

The MT9M114 allows for the 19 knee point curves to be programmed based off a small number of variables. The table below shows the variables which are required.

Table 8: Variables Required for Gamma Knee Point Calculation

Variable	Name	Function
VAR(0x12,0x0124) or (R0xC924)	cam_ll_llmode	0x00: User will program 19 knee point gamma curves 0x01: MT9M114 will calculate 19 knee point for contrast curve (first curve or table). 0x02: MT9M114 will calculate 19 knee point for noise reduction curve (second curve or table). 0x03: MT9M114 will calculate both 19 knee point curves.
VAR(0x12,0x013C) or (R0xC93C)	cam_ll_start_contrast_bm	Interpolation start point for first curve
VAR(0x12,0x013E) or (R0xC93E)	cam_ll_stop_contrast_bm	Interpolation stop point for second curve
VAR(0x12,0x0140) or (R0xC940)	cam_ll_gamma	The value of the gamma curve, this is applied to both 19 knee point curves. The default is 220, this equates to a gamma of 2.2.

Table 8: Variables Required for Gamma Knee Point Calculation

Variable	Name	Function
VAR(0x12,0x0142) or (R0xC942)	cam_ll_start_contrast_gradient	The value of the contrast gradient which would be used for the first curve
VAR(0x12,0x0143) or (R0xC943)	cam_ll_stop_contrast_gradient	The value of the contrast gradient which would be used for the second curve
VAR(0x12,0x0144) or (R0xC944)	cam_ll_start_contrast_luma_percentage	The percentage of target luma for the inflexion point in the first curve
VAR(0x12,0x0145) or (R0xC945)	cam_ll_start_contrast_luma_percentage	The percentage of target luma for the inflexion point second curve
VAR(0x12,0x0156) or (R0xC956)	cam_ll_inv_brightness_metric	Measure of scene brightness, reference points for cam_ll_start_contrast_bm and cam_ll_stop_contrast_bm

The concept of how the variables cam_ll_XX_contrast_gradient and cam_ll_XX_contrast_luma_percentage interact to produce a curve is shown below.

Figure 24: Automatic Gamma Curve

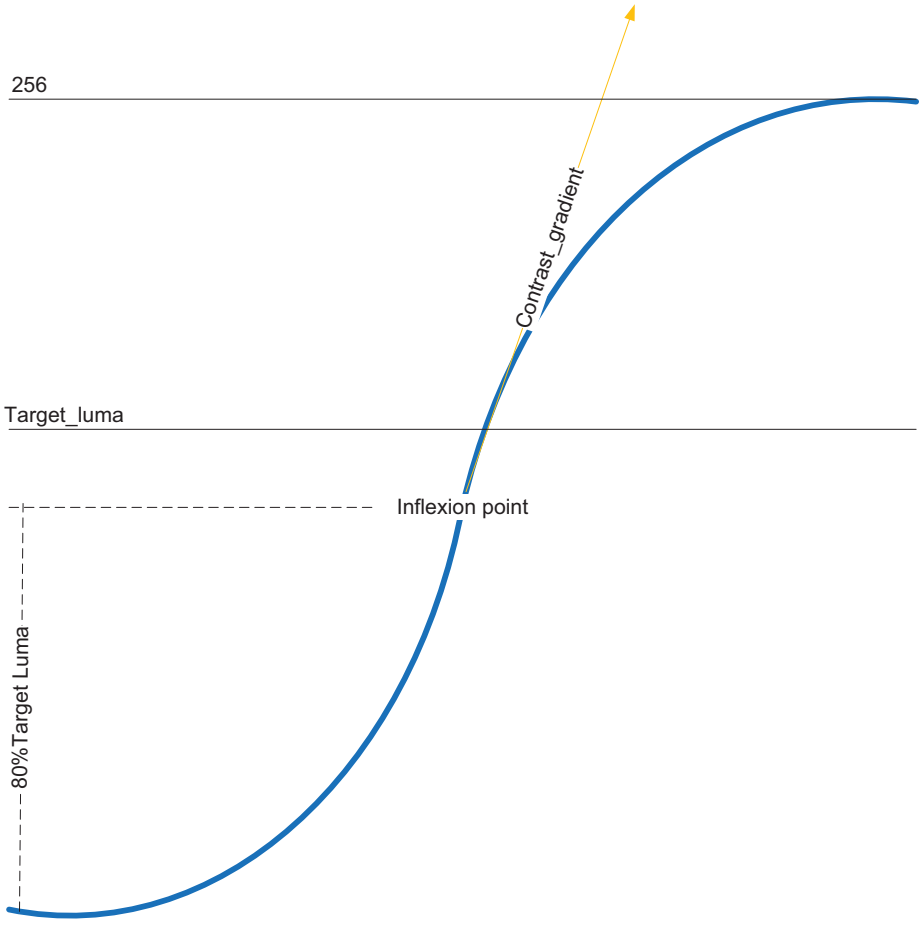
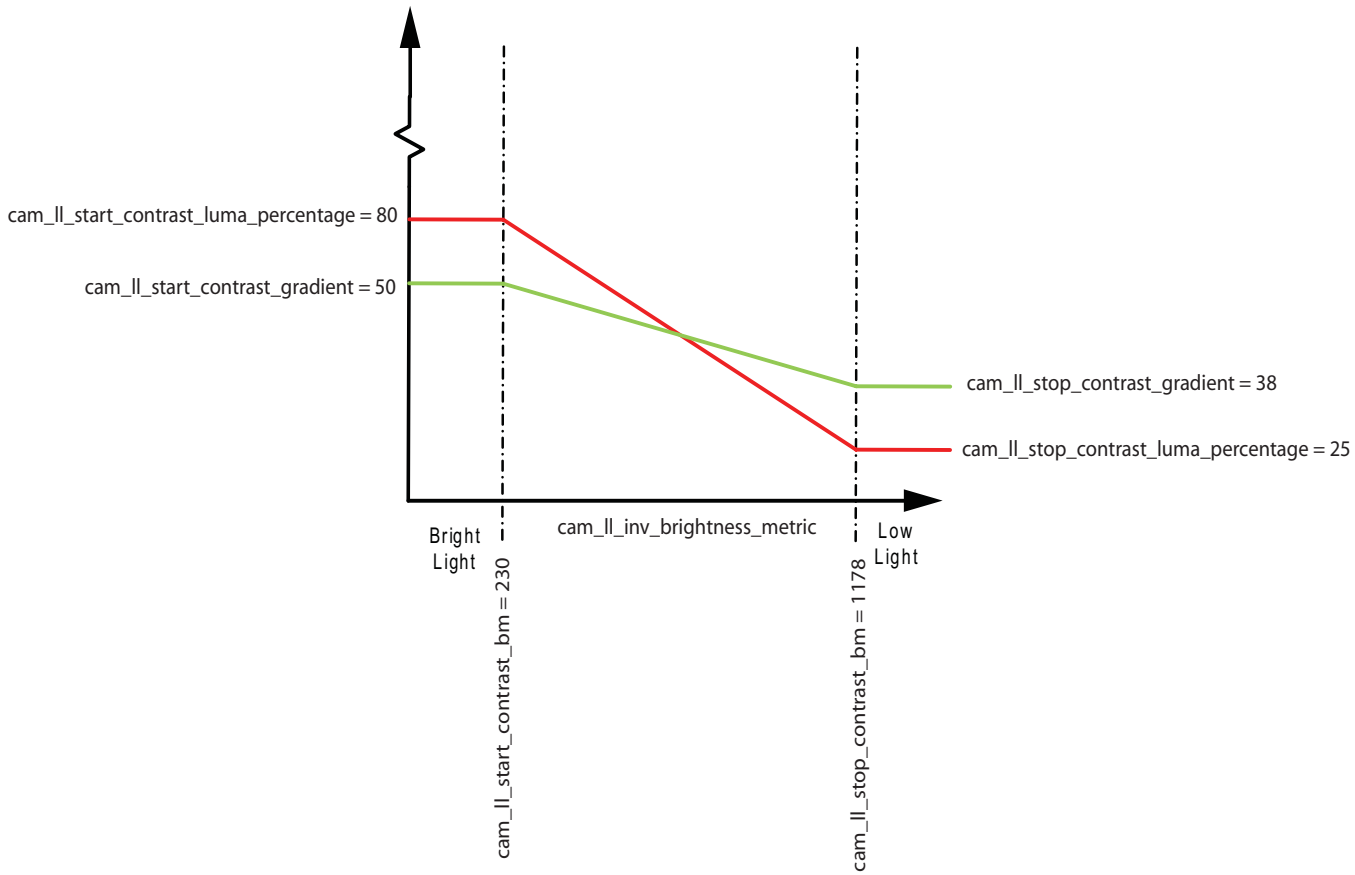


Figure 25 shows the interaction of the variables and cam_ll_inv_brightness_metric.

Figure 25: Gamma Reference Variables Against Brightness Metric



Aptina would recommend that cam_ll_start_contrast_bm is set at 100 lux and cam_ll_stop_contrast_bm is set at 20 lux, but due to the flexibility of the MT9M114 it is at the discretion of the user.

Aptina recommends setting cam_ll_llmode = 0x03 as this will allow the MT9M114 to calculate both of the 19 knee point curves based on the user inputs, otherwise the user will have to program both of the 19-knee-point curves.

Gamma Curve Selection

The MT9M114 allows the user to select between the two-curve interpolation or either of the curves

Table 9: Gamma curve Selection

Variable	Name	Function
VAR(0x0F,0x0007) or (R0xBC07)	ll_gamma_select	0x00= Auto curve select. The curves will interpolate based on settings of cam_ll_start_contrast_bm and cam_ll_stop_contrast_bm 0x01 = Contrast curve is only used 0x02 = Noise reduction curve is only used

Fade to Black Selection

The final stage of the gamma flow is the enabling and use of Fade-to-Black. The MT9M114 IFP allows for the image to fade to black under extreme low-light conditions. This feature enables users to optimize the performance of the sensor under low-light conditions. It minimizes the perception of noise and artifacts while the available illumination is diminishing.

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

Table 10: Fade-to-Black Selection

Variable	Name	Function
VAR(0x0F,0x0007) or (R0xBC07)	ll_mode	When bit 3=1, this will enable the Fade-to-Black feature
VAR(0x12,0x014A) or (R0xC94A)	cam_ll_start_fade_to_black_luma	Starting point for Fade-to-Black to begin
VAR(0x12,0x014C) or (R0xC94C)	cam_ll_stop_fade_to_black_luma	End point for Fade-to-Black, after this point the image will be black
VAR(0x0F,0x003A) or (R0xBC3A)	ll_average_luma_fade_to_black	Measure of scene brightness, reference points for cam_ll_start_fade_to_black_luma and cam_ll_stop_fade_to_black_luma

Aptina would recommend that cam_ll_start_fade_to_black_luma is set at 3 lux and cam_ll_stop_fade_to_black_luma is set at 1 lux, but due to the flexibility of the MT9M114 it is at the discretion of the user.

Image Scaling and Cropping

To ensure that the size of images output by the MT9M114 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 the selected width (the output widths should be in multiples of 4) 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. The height and width definitions for the output window must be equal to or smaller than the cropped image. The image cropping and scaler module can be used together to implement a digital zoom and pan.

Hue Rotate

The MT9M114 has integrated hue rotate. This feature will help for improving the color image quality and give customers the flexibility for fine color adjustment and special color effects.

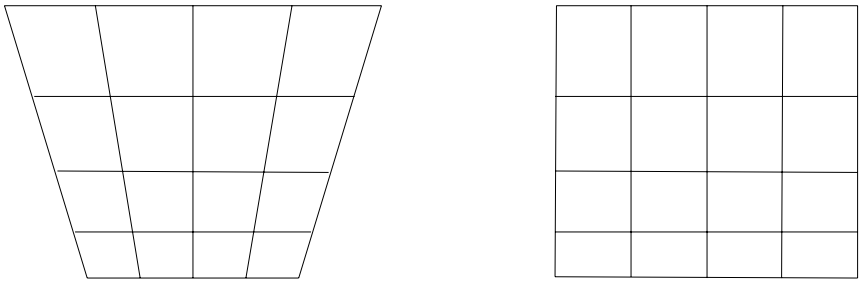
Table 11: Hue Control

Variable	Name	Function
VAR(0x12,0x73) or R0xC873	Hue Angle	Adjusts the global hue angle adjustment. 0xEA = –22° 0x00 = 0° 0x16 = +22°

Vertical Perspective Correction

The MT9M114 has vertical perspective correction (VPC) also known as the Tilt Connection; this allows the user to correct (within limits) for an off-horizontal axis camera.

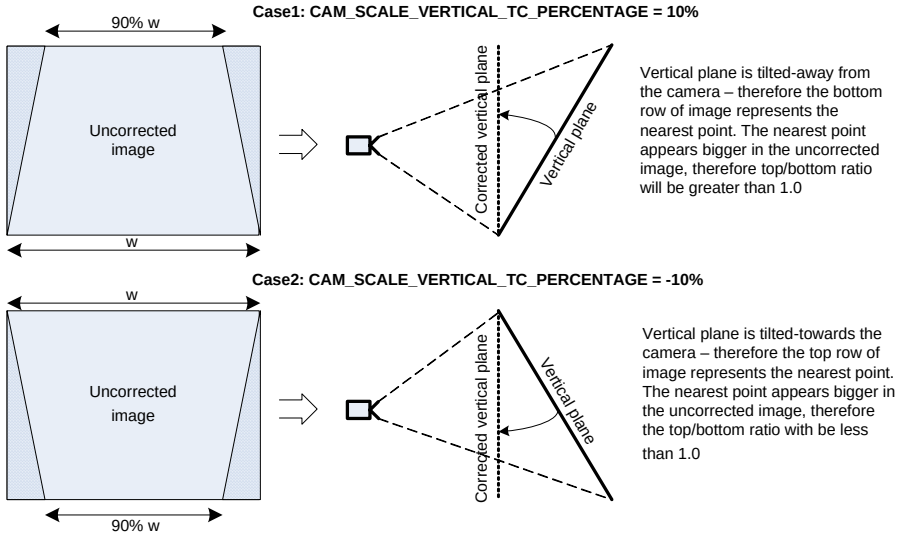
Original Image *VPC Corrected Image*



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

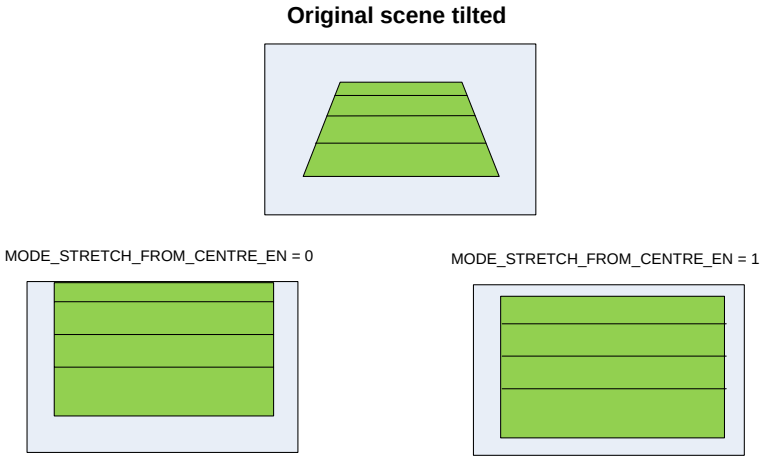
Variable	Name	Function
VAR(0x12,0x005E) or (R0xC85E)	cam_scale_vertical_tc_mode	When set, the vertical stretching factor is applied to the centre of the image, so top/ bottom lines are cropped. When clear, the crop occurs in the top or bottom of the scene dependent on the percentage value (cam_scale_vertical_tc_percentage).
VAR(0x12,0x0060) or (R0xC860)	cam_scale_vertical_tc_percentage	The amount of tilt (perspective) correction to be applied. If negative, this value represents % of FOV reduction with the bottom line unaffected. If positive, this value represents % of FOV reduction with the top line unaffected.
VAR(0x12,0x0062) or (R0xC862)	cam_scale_vertical_tc_stretch_factor	Ratio of vertical stretching against the percentage applied. Vertical stretching = stretch factor x percentage/2.

The effect of using cam_scale_vertical_tc_percentage can be seen below.



Cam_scale_vertical_tc_percentage defines how much tilt needs to be corrected for in percentage terms.

The effect of using cam_scale_vertical_tc_mode can be seen below.



Camera Control and Auto Functions

Auto Exposure

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

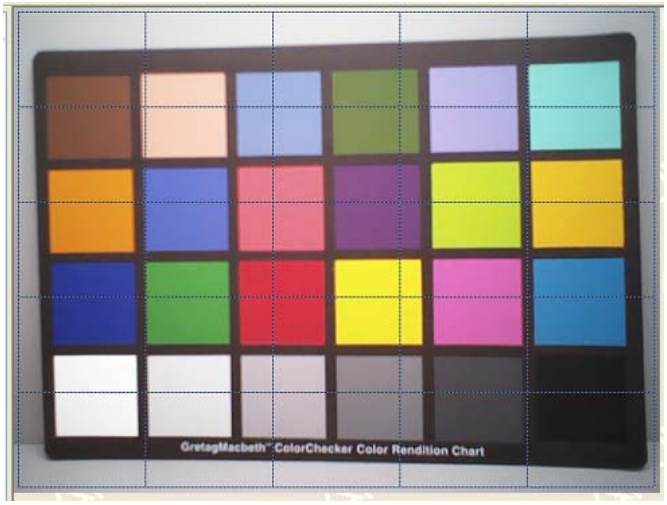
Auto exposure is implemented by a firmware driver that analyzes image statistics collected by the exposure measurement engine, makes a decision, and programs the sensor core and color pipeline to achieve the desired exposure. The measurement engine subdivides the image into 25 windows organized as a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

Four auto exposure algorithm modes are available:

- Average brightness tracking (ABT) or Average Y (ae_rule_algo VAR= 9, 0x0004, 0x0000 or REG= 0xA404, 0x0000)
  The average brightness tracking AE uses a constant average tracking algorithm where a target brightness value is compared to a current brightness value, and the gain and integration time are adjusted accordingly to meet the target requirement.
- Weighted Average Brightness (ae_rule_algo VAR= 9, 0x0004, 0x0001 or REG= 0xA404, 0x0001)
  Each of the 25 windows can be assigned a weight relative to other window weights, which can be changed independently of each other. For example, the weights can be set to allow the center of the image to be weighted higher than the periphery. See Figure 26.

Figure 26: 5 x 5 Grid

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



- Adaptive Weighted AE for highlights (ae_rule_algo VAR= 9, 0x0004, 0x0002 or REG= 0xA404, 0x0002)- The scene will be exposed based on the brightness of each window, and will adapt to correctly expose the highlights (brighter windows). This would correctly expose the foreground of an image when the background is dark.
- Adaptive Weighted AE for lowlights(ae_rule_algo VAR= 9, 0x0004, 0x0003 or REG= 0xA404, 0x0003)- The scene will be exposed based on the brightness of each window, and will adapt to correctly expose the lowlights. This would correctly expose the foreground of an image when the background is brighter.

Sample images below show the benefits of the different AE modes.

Light Background

Average Brightness Tracking or Average Y

Weighted Average Brightnes s(centre)

Adaptive weighted based on zone luma (highlights)

Adaptive weighted based on zone luma (lowlights)

Note: This mode is intended to expose the background vs. the foreground

In the use case above the Adaptive weighted for lowlights exposes the face slightly better when compared to the Weighted Average Brightness.

However, if the foreground subject is moved off-center:

Weighted Average Brightness (centre) (lowlights)

Adaptive weighted based on zone luma

This shows the advantage of using the Adaptive Weighted AE for lowlights(ae_rule_algo = 0x03); when the face moves off center it still is exposed correctly.

Dark Background

Average Brightness Tracking or Average Y

Weighted Average Brightnes s(centre)

Adaptive weighted based on zone luma (highlights)

Adaptive weighted based on zone luma (lowlights)

Note: This mode is correctly exposing the background of the image, hence you can see the shadows.

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

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 the small changes. While the default settings are adequate in most situations, the user can program target brightness, measurement window, and other parameters described above.

The driver changes AE parameters (integration time, gains, and so on) to drive 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 MT9M114 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 MT9M114 AWB displays the current AWB position in color temperature, the range of which will be defined when programming the CCM matrixes.

Flicker Avoidance

Flicker occurs when the integration time is not an integer multiple of the period of the light intensity. The MT9M114 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 MT9M114 supports an indoor AE mode, that will ensure flicker-free operation (VAR8= 18, 0x0078[0]=0x1 o REG= 0xC878[0]= 0x1). The MT9M114 will calculate all flicker parameters based on the sensor settings which are programmed in the Cam Control variables. This means the user only needs to select if 50- or 60-Hz flicker needs to be avoided (VAR 0x12, 0x008B or R0xC88B = 50 for 50-Hz flicker avoidance and 60 for 60-Hz avoidance).

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 MT9M114 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.

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

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

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

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

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

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

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

Uncompressed YUV/RGB Data Ordering

The MT9M114 supports swapping YCbCr mode, as illustrated in Table 12.

Table 12: 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 13. 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 13: RGB Ordering in Default Mode

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

Uncompressed Raw Bayer Bypass Output

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

1. Using both DOUT[7:0] and DOUT_LSB[1:0].
2. Using only DOUT[7:0] with a special 8 + 2 data format, shown in Table 14.
3. Using the MIPI interface.

Table 14: 2-Byte Bayer Format

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

UVC Interface

The MT9M114 supports a set of UVC (USB Video Class) controls in order to simplify the integration of the MT9M114 with a host's USB bridge (or ISP) device.

The MT9M114 firmware includes a 'UVC Control' component that augments the CamControl variables. The UVC Control component sits above the CamControl interface (in terms of abstraction) and acts as a 'virtual host'. The intention is that CamControl and all other components are unaware of the UVC Control component.

UVC Control exposes a 'UVC control' page of shared variables to the host. This page contains variables compliant with the UVC 1.1 specification (where possible). The variables on this page are named to match the UVC specification, and have matching data sizes, units and ranges as required. Each UVC variable is 'virtual' - it does not control any MT9M114 function directly.

MT9M114 therefore provides a 'dual-personality' host interface:

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

The secondary UVC Control interface, which simplifies integration of MT9M114 into a PC-Cam application.

More details on this topic can be found in the Developer Guide.

Table 15: Summary of UVC Commands

Variable	Name
R0xCC00 VAR(0x13,0x0000)	UVC_AE_MODE_CONTROL
R0xCC01 VAR(0x13,0x0001)	UVC_WHITE_BALANCE_TEMPERATURE_AUTO_CONTROL
R0xCC02 VAR(0x13,0x0002)	UVC_AE_PRIORITY_CONTROL
R0xCC03 VAR(0x13,0x0003)	UVC_POWER_LINE_FREQUENCY_CONTROL
R0xCC04 VAR(0x13,0x0004)	UVC_EXPOSURE_TIME_ABSOLUTE_CONTROL
R0xCC08 VAR(0x13,0x0008)	UVC_BACKLIGHT_COMPENSATION_CONTROL
R0xCC0A VAR(0x13,0x000A)	UVC_BRIGHTNESS_CONTROL
R0xCC0C VAR(0x13,0x000C)	UVC_CONTRAST_CONTROL
R0xCC0E VAR(0x13,0x000E)	UVC_GAIN_CONTROL UINT16
R0xCC10 VAR(0x13,0x0010)	UVC_HUE_CONTROL
R0xCC12 VAR(0x13,0x0012)	UVC_SATURATION_CONTROL UINT16
R0xCC14 VAR(0x13,0x0014)	UVC_SHARPNESS_CONTROL
R0xCC16 VAR(0x13,0x0016)	UVC_GAMMA_CONTROL

Table 15: Summary of UVC Commands (continued)

Variable	Name
R0xCC18 VAR(0x13,0x0018)	UVC_WHITE_BALANCE_TEMPERATURE_CONTROL
R0xCC1C VAR(0x13,0x001C)	UVC_FRAME_INTERVAL_CONTROL
R0xCC20 VAR(0x13,0x0020)	UVC_MANUAL_EXPOSURE_CONFIG
R0xCC21 VAR(0x13,0x0021)	UVC_FLICKER_AVOIDANCE_CONFIG

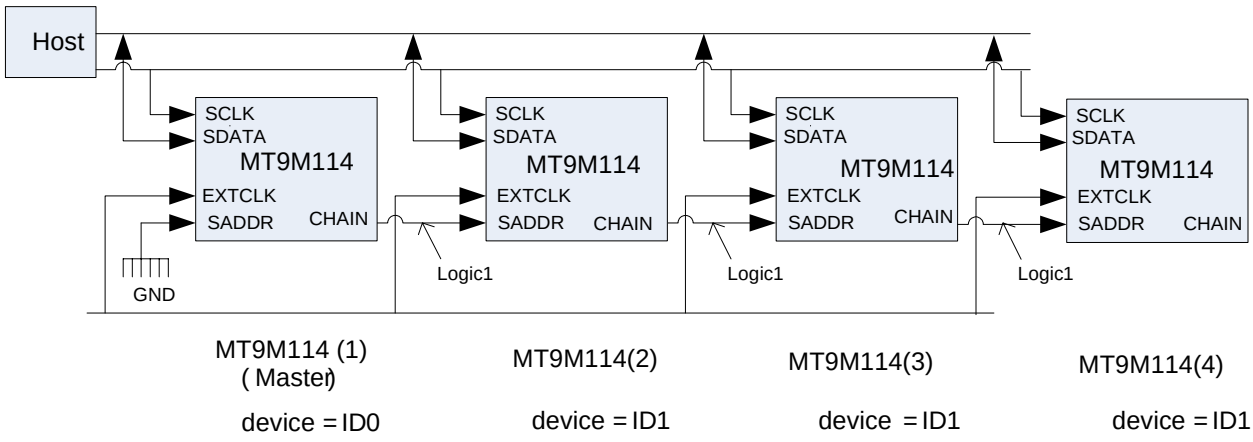
Multi-Camera Sync

The MT9M114 supports more than one device to be connected in a "daisy-chain" type configuration. One of the devices will act as the master and the remainder will be slaves.

A typical connection diagram is shown inFigure 27. All of the MT9M114 that are to communicate are:

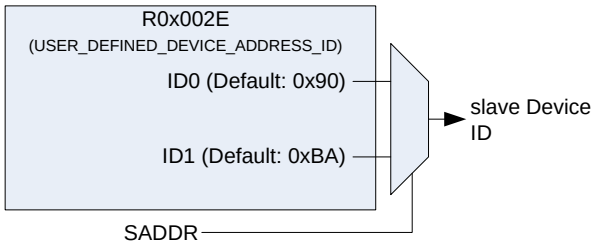
- Connected in a daisy-chain using SADDR as an input and CHAIN as an output.
- Clocked from a common clock source
- Controlled from a single master, presumed to be under software control of a host system.

Figure 27: Multi-Camera Connection



SADDR is normally used as a static input that selects between two slave device addresses (See Figure 28. In order to implement the multi-sync function this input now has additional functionality that does not interfere with its use as device address selection.

Figure 28: Normal Use of SADDR



There is a single register to control this function, named CHAIN_CONTROL (R0x31FC). This register is controlled by the host. The register field assignment is shown in Table 16.

Table 16: CHAIN_CONTROL Register

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

Configuration

Before the multi-sync function can be used, each MT9M114 in the daisy-chain must be configured. This process is performed by the host with no involvement from MT9M114 firmware. Configuration involves assigning a unique slave address to each MT9M114 and configuring the CHAIN_CONTROL register on each MT9M114.

After reset (before configuration) the master MT9M114 has its SADDR input wired to '0' and all other MT9M114 in the daisy-chain have their SADDR inputs driven to '1'. Therefore, MT9M114 Master will respond to slave address ID0 (associated with SADDR = 0) and all the other MT9M114 in the daisy-chain will respond simultaneously to slave address ID1. Each MT9M114 has its CHAIN pin configured as an input. This situation is shown in Figure 27. The host configures each MT9M114 in sequence, starting with the master and ending with the farthest slave in the daisy-chain:

- MT9M114(1) Master: The host uses slave address ID0 (associated with SADDR = 0) and therefore accesses registers on MT9M114(1) (the master). It writes to register (R0x002E) to change the slave addresses associated with ID0 and ID1 on this device to a single, new, unique value; call it ID-MT9M114(1). It then writes (using MT9M114(1) to register PAD_CONTROL (R0x0032) to configure CHAIN as an output. Finally, it writes (using MT9M114(1)) to the CHAIN_CONTROL register to set chain_enable =1,

sync_enable=1, master=1 and position = N – 1 (where there are N devices in the daisy-chain). The effect of enabling TMS as an output is to drive the TMS output low.

- MT9M114(2): This MT9M114 now has SADDR=0 and so will respond to slave address ID0. The host configures this in the same way as MT9M114(1) with the exceptions that it assigns ID-MT9M114(2), sets master=0 and position = N-2 (where there are N devices in the daisy-chain). As before, the effect of enabling CHAIN as an output is to drive the CHAIN output low.
- MT9M114(3): As for MT9M114(2): assign ID-MT9M114(3), master=0, position = N-3
- MT9M114(4): As for MT9M114(2): assign ID-MT9M114(4), master=0, position = N-4

Theory Of Operation

When multiple MT9M114 devices have been connected and configured as described above, the multi-sync function operates as follows:

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

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

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

Using Multi-Sync

The host can use the normal mechanism to configure the MT9M114 and set them streaming. It can do this in any order provided that it sets the master streaming last.

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

At the time that the MT9M114 are placed in streaming mode, all MT9M114 must have the same integration time The recommended mechanism is:

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

2) Reconfigure each device.

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

The MT9M114 need not maintain the same integration time once they are streaming.

All the MT9M114 must be operated with the same configuration (image size, output format, PLL bypassed and frame timing). Any time that the configuration is to be changed, all MT9M114 must be taken out of streaming mode (using host command), reconfigured, then placed back in streaming mode (master last). This will allow the output data to remain in synchronisation.

Clocking

The multi-sync mechanism requires that all MT9M114 devices in the daisy-chain are operated synchronously on the same input clock. This constraint is imposed in order to allow the event codes to be propagated synchronously from the master through to each slave.

Once this constraint has been met, the MT9M114 devices are required to operate in exact synchronisation (such that a PIXCLK, FRAME_VALID and LINE_VALID out of one MT9M114 is valid for all MT9M114 in the daisy-chain). In this case, the MT9M114 internal PLL must be bypassed (and the MT9M114 must be using parallel output data).

Hardware Functions

Two-Wire Serial Interface

The two-wire serial interface bus enables read and write access to control and status registers and variables within the MT9M114.

The interface protocol uses a master/slave model in which a master controls one or more slave devices. The MT9M114 always operates in slave mode. The host (master) generates a clock (SCLK) that is an input to the MT9M114 and is used to synchronize transfers. Data is transferred between the master and the slave on a bidirectional signal (SDATA).

The host should always ensure that the following relationship is adhered to.

SCLK ≤ (PIXEL CLOCK/22)

Protocol

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

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

The bus is idle when both SCLK and SDATA are HIGH. Control of the bus is initiated with a start condition, and the bus is released with a stop condition. Only the master can generate the start and stop conditions.

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

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

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

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

Slave Address

Bits [7:1] of this byte represent the device slave address and bit [0] indicates the data transfer direction. A “0” in bit [0] indicates a WRITE, and a “1” indicates a READ. If the SADDR signal is driven LOW, then addresses used by the MT9M114 are R0x090 (write address) and R0x091 (read address). If the SADDR signal is driven HIGH, then addresses used by the MT9M114 are R0x0BA (write address) and R0x0BB (read address).

Message Byte

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

Acknowledge Bit

Each 8-bit data transfer is followed by an acknowledge bit or a no-acknowledge bit in the SCLK clock period following the data transfer. The transmitter (which is the master when writing, or the slave when reading) releases SDATA. The receiver indicates an acknowledge bit by driving SDATA LOW. As for data transfers, SDATA can change when SCLK is LOW and must be stable while SCLK is HIGH.

No-Acknowledge Bit

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

Stop Condition

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

Typical Serial Transfer

A typical read or write sequence begins by the master generating a start condition on the bus. After the start condition, the master sends the 8-bit slave address/data direction byte. The last bit indicates whether the request is for a read or a write, where a "0" indicates a write and a "1" indicates a read. If the address matches the address of the slave device, the slave device acknowledges receipt of the address by generating an acknowledge bit on the bus.

If the request was a write, the master then transfers the 16-bit register address to which a write should take place. This transfer takes place as two 8-bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master then transfers the data as an 8-bit sequence; the slave sends acknowledge bit at the end of the sequence. After 8 bits have been transferred, the slave's internal register address is automatically incremented, so that the next 8 bits are written to the next register address. The master stops writing by generating a (re)start or stop condition.

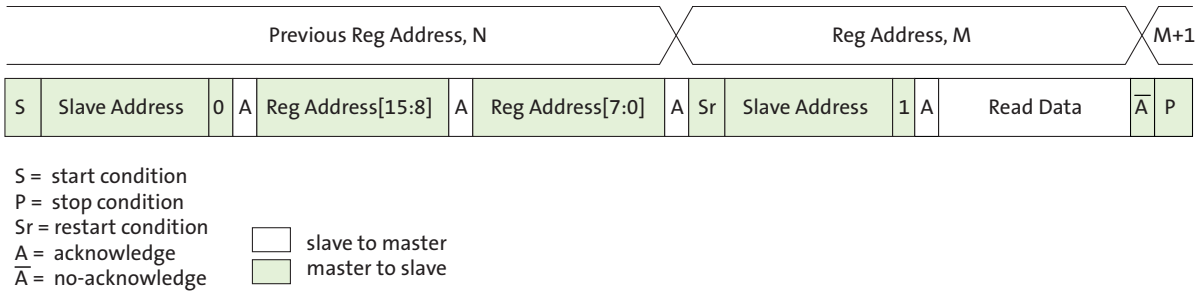
If the request was a read, the master sends the 8-bit write slave address/data direction byte and 16-bit register address, just as in the write request. The master then generates a (re)start condition and the 8-bit read slave address/data direction byte, and clocks out the register data, 8 bits at a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

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

Single Read from Random Location

This sequence (see Figure 29) starts with a dummy write to the 16-bit address that is to be used for the read. The master terminates the write by generating a restart condition. The master then sends the 8-bit read slave address/data direction byte and clocks out one byte of register data. The master terminates the read by generating a no-acknowledge bit followed by a stop condition. Figure 29 shows how the internal register address maintained by the MT9M114 is loaded and incremented as the sequence proceeds.

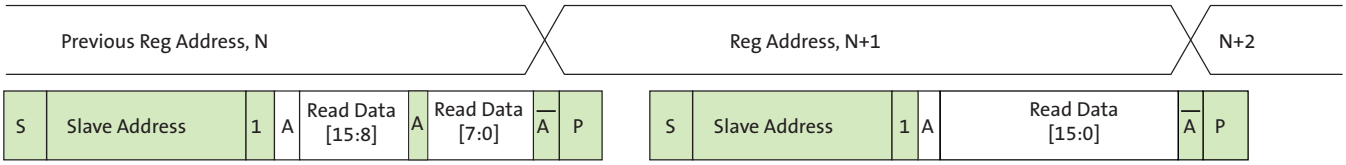
Figure 29: Single Read from Random Location



Single Read from Current Location

This sequence (Figure 30) performs a read using the current value of the MT9M114 internal register address. The master terminates the read by generating a no-acknowledge bit followed by a stop condition. The figure shows two independent read sequences.

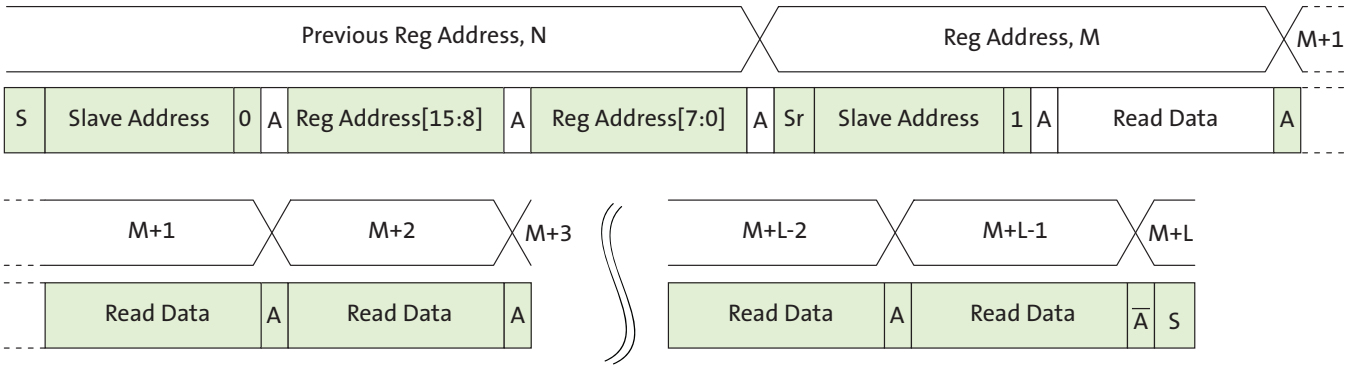
Figure 30: Single Read from Current Location



Sequential Read, Start from Random Location

This sequence (Figure 31) starts in the same way as the single read from random location (Figure 29). 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.

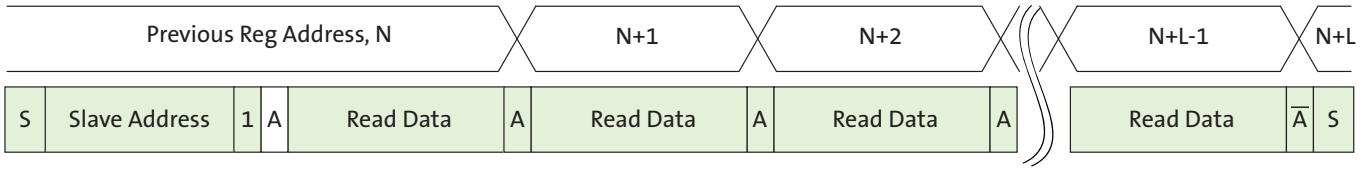
Figure 31: Sequential Read, Start from Random Location



Sequential Read, Start from Current Location

This sequence (Figure 32) starts in the same way as the single read from current location (Figure 30). 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.

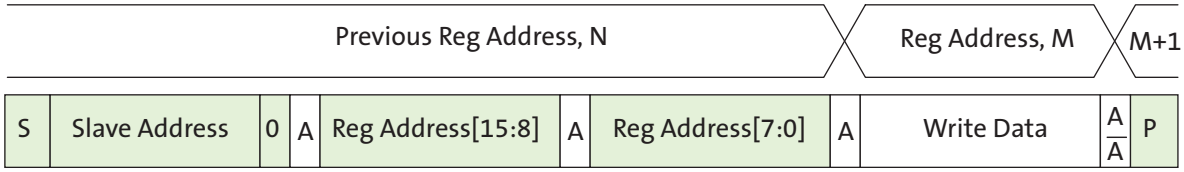
Figure 32: Sequential Read, Start from Current Location



Single Write to Random Location

This sequence (Figure 33) begins with the master generating a start condition. The slave address/data direction byte signals a write and is followed by the high then low bytes of the register address that is to be written. The master follows this with the byte of write data. The write is terminated by the master generating a stop condition.

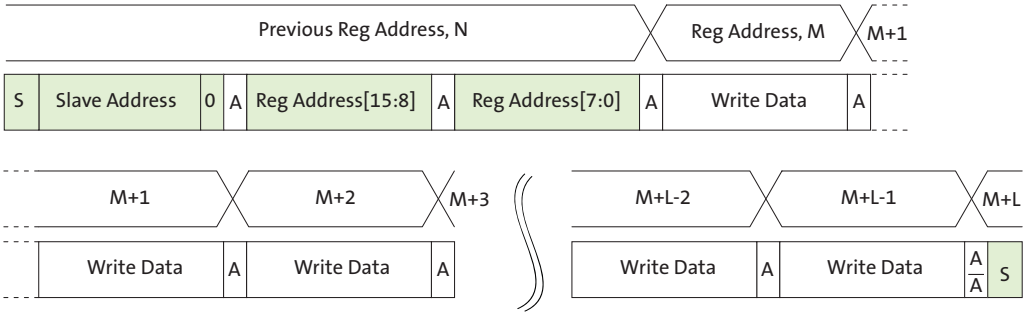
Figure 33: Single Write to Random Location



Sequential Write, Start at Random Location

This sequence (Figure 34) starts in the same way as the single write to random location (Figure 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 writes until L bytes have been written. The write is terminated by the master generating a stop condition.

Figure 34: Sequential Write, Start at Random Location



Patch RAM

MT9M114 has Patch Ram, this allows for issues to be fixed without changing silicon version and also allows for new features to be added to the silicon.

The patch would come in the format of Load and Apply sections, the user needs to implement both sections. Below includes detail of what can be achieved when in different host states.

STANDBY- LOAD PATCHES ONLY

STREAMING- LOAD AND APPLY PATCHES

SUSPEND-LOAD AND APPLY PATCHES

Chief Ray Angle

Table 17: Chief Ray Angle

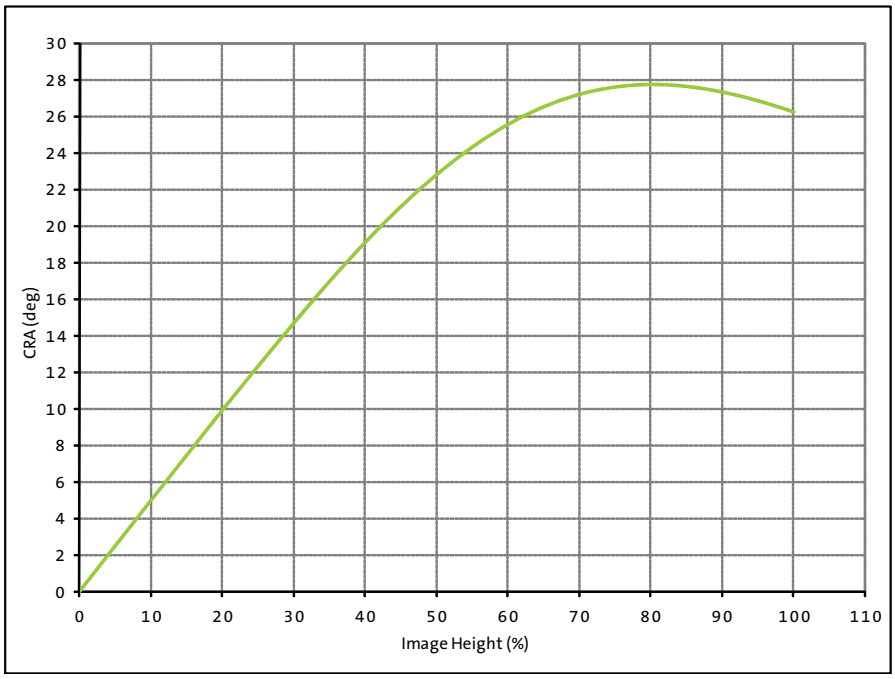
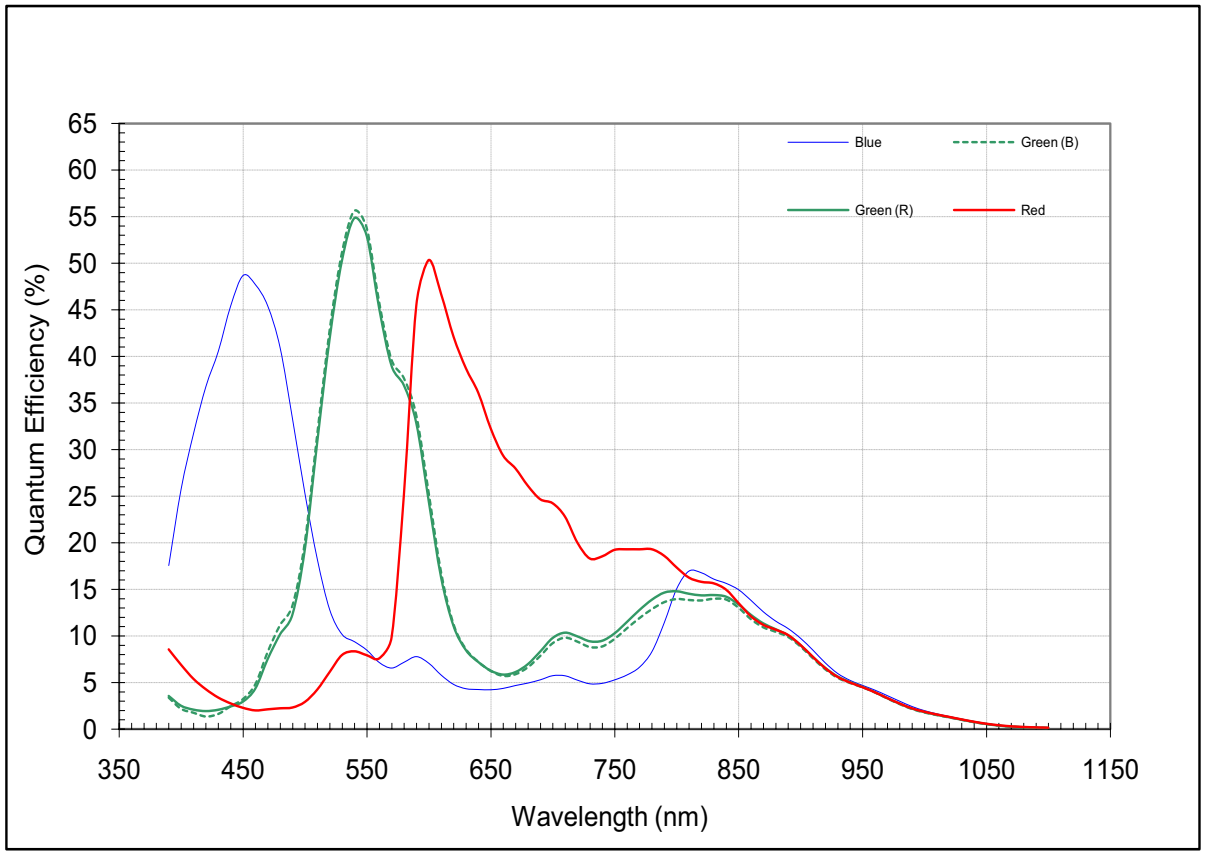


Image Height		CRA
(%)	(mm)	(deg)
0	0	0
5	0.076	2.51
10	0.152	4.98
15	0.228	7.44
20	0.304	9.89
25	0.380	12.32
30	0.456	14.68
35	0.532	16.94
40	0.608	19.08
45	0.684	21.03
50	0.760	22.78
55	0.836	24.28
60	0.912	25.52
65	0.988	26.49
70	1.064	27.16
75	1.140	27.57
80	1.216	27.70
85	1.292	27.60
90	1.368	27.29
95	1.444	26.81
100	1.520	26.20

Figure 35: Typical Quantum Efficiency



Note: This is measured on bare die.

Electrical Specifications

Caution **Stresses above those listed in Table 18 may cause permanent damage to the device.**

Table 18: Absolute Maximum Ratings

Symbol	Parameter	Rating		Unit
		Min	Max	
V_{DD}_MAX	Core digital voltage	–0.3	2.4	V
V_{DD}_IO_MAX	I/O digital voltage	–0.3	4.0	V
V_{AA}_MAX	Analog voltage	–0.3	4.0	V
V_{DD}__PLL_MAX	PLL supply voltage	–0.3	4.0	V
V_{DD}_PHY_MAX	PHY supply voltage	–0.3	2.4	V
V_{IN}	DC input voltage	–0.3	VDD_IO + 0.3	V
I_{IN}	Transient input current (0.5 sec. duration)	–	150	mA
T_{OP}	Operating temperature (measure at junction)	–30	75	°C
T_{STG}[1]	Storage temperature	–40	85	°C

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

Recommended Operating Conditions

Table 19: 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
V_{DD_PLL}	PLL supply voltage	2.5	2.8	3.1	V
V_{DD_PHY}	PHY supply voltage	1.7	1.8	1.95	V
T_J	Operating temperature (at junction)	–30	55	70	°C

Table 20: DC Electrical Characteristics

Symbol	Parameter	Condition	Min	Max	Unit	Notes
V_{IH}	Input HIGH voltage		V_{DD_IO} * 0.7	–	V	1
V_{IL}	Input LOW voltage		–	V_{DD_IO} * 0.3	V	1
I_{IN}	Input leakage current	V_{IN} = 0V or V_{IN} = V_{DD_IO}		10	µA	2
V_{OH}	Output HIGH voltage	I_{OH} = 2 mA	V_{DD_IO}* 0.75		V	
V_{OL}	Output LOW voltage	I_{OH} = 2 mA	–	V_{DD_IO} * 0.25	V	

Notes:
1. V_{IL} and V_{IH} have min/max limitations specified by absolute ratings.
2. Excludes CONFIG and RESET_BAR as they have an internal pull-up resistor.

Table 21: Operating Current Consumption

Default Setup Conditions: f_{EXTCLK} = 24 MHz, f_{PIXCLK} = 96 MHz, V_{AA} = V_{DD}_IO = V_{DD}_PLL = 2.8V, V_{DD} = V_{DD}_PHY = 1.8V, Tj = 70°C unless otherwise stated, PN9 enabled, specified under MIPI and Parallel output conditions .

Symbol	Conditions	Min	Typ	Max	Unit
V_{DD}		1.7	1.8	1.95	V
V_{AA}		2.5	2.8	3.1	V
V_{DD}_PHY		1.7	1.8	1.95	V
V_{DD}_PLL		2.5	2.8	3.1	V
V_{DD}_IO	V_{DD}_IO = 2.8V	2.5	2.8	3.1	V
	V_{DD}_IO = 1.8V	1.7	1.8	1.95	V
I_{DD}	Full resolution 30 fps, parallel		39	50	mA
	720p, 30 fps, parallel		33	45	mA
	VGA binned, 60 fps, parallel		25	40	mA
	Full resolution, 30fps, MIPI		39	50	mA
	720p, 30 fps, MIPI		33	45	mA
	VGA binned, 60 fps, MIPI		25	40	mA
I_{AA}	Full resolution, 30 fps, parallel		19	35	mA
	720p, 30 fps, parallel		19	35	mA
	VGA binned, 60 fps, parallel		19	35	mA
	Full resolution, 30 fps, MIPI		19	35	mA
	720p, 30 fps, MIPI		19	35	mA
	VGA binned, 60 fps, MIPI		19	35	mA
I_{DD}_PLL	Full resolution, 30 fps, parallel		8	20	mA
	720p, 30 fps, parallel		8	20	mA
	VGA, 60 fps, parallel		8	20	mA
	Full resolution, 30f ps, MIPI		25	40	mA
	720p, 30 fps, MIPI		25	40	mA
	VGA binned, 60 fps, MIPI		25	40	mA
I_{DD}_PHY	Full resolution, 30 fps, parallel		0.02	0.5	mA
	720p, 30fps, parallel		0.02	0.5	mA
	VGA binned, 60 fps, parallel		0.02	0.5	mA
	Full resolution, 30 fps, MIPI		0.18	1	mA
	720p, 30 fps, MIPI		0.18	1	mA
	VGA binned, 60 fps, MIPI		0.18	1	mA
Total power consumption[1]	Full resolution, 30 fps, parallel		146		mW
	720p, 30fps, parallel		135		mW
	VGA binned, 60 fps, parallel		121		mW
	Full resolution, 30 fps, MIPI		194		mW
	720p, 30 fps, MIPI		183		mW
	VGA binned, 60 fps, MIPI		169		mW

Notes: 1. Total power excludes V_{DD}_IO current.

Table 22: Standby Current Consumption (Parallel and MIPI)

Default Setup Conditions: f_{EXTCLK}=24 MHz, f_{PIXCLK}=96 MHz, V_{AA} = V_{DD}_IO = V_{DD}_PLL = 2.8V, V_{DD} = V_{DD}_PHY = 1.8V, T_J = 70°C unless otherwise stated

		Typical	Max	Unit
Soft Standby (CLK ON)	Total standby current in parallel and MIPI mode	1.4	3	mA
	Total power consumption in parallel and MIPI mode	2.5		mW
Soft Standby (CLK OFF)	Total standby current in parallel and MIPI mode	80	500	µA
	Total power consumption in parallel and MIPI mode	150		µW

Note: All power measurements exclude IO current.

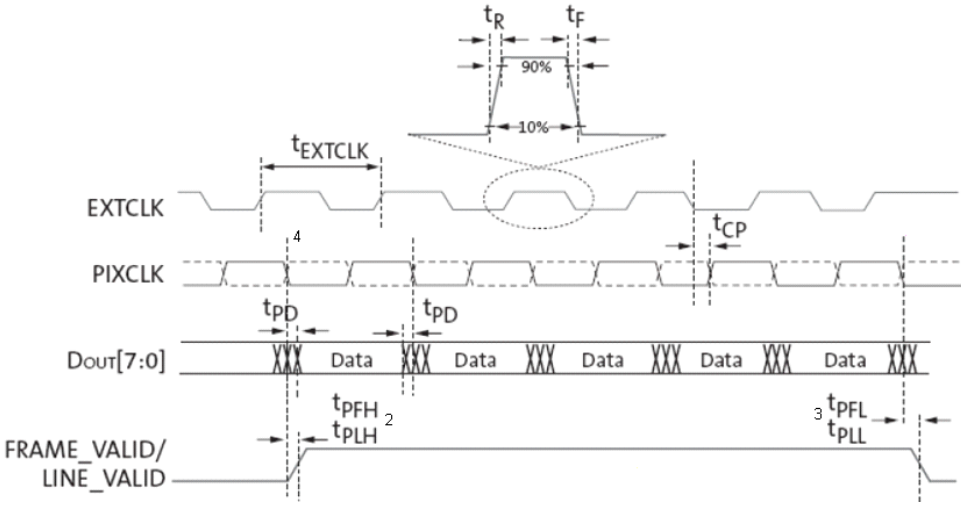
Table 23: AC Electrical Characteristics

EXTCLK = 6–54 MHz; V_{DD} = V_{DD_PHY} = 1.8V; V_{DD_IO} = V_{AA} = V_{DD_PLL} = 2.8V; T_J = 25°C unless otherwise stated

Symbol	Parameter	Conditions	Min	Typ	Max	Unit	Notes
f_{EXTCLK}	External clock frequency		6		54	MHz	1
D_{EXTCLK}	External input clock duty cycle		40	50	60	%	
t_{JITTER}	External input clock jitter		–	500	–	ps	2
t_{PD}	PIXCLK to data valid		–	2	5	ns	
t_{PFH}	PIXCLK to FV HIGH		–	2	5	ns	
t_{PLH}	PIXCLK to LV HIGH		–	2	5	ns	
t_{PFL}	PIXCLK to FV LOW		–	2	5	ns	
t_{PLL}	PIXCLK to LV LOW		–	2	5	ns	
t_{CP}	EXTCLK TO PIXCLK propagation delay	t_{PIXCLK} = PICXCLK period		0.1 x t_{PIXCLK}		ns	
PIXCLK slew rate							
	Slew = 4	V_{DD_IO} = 2.8V, PLL bypass, 6 MHz EXTCLK, C_{LOAD} = 35 pF	–	0.647	–	V/ns	
		V_{DD_IO} = 1.8V, PLL bypass, 6 MHz EXTCLK, C_{LOAD} = 35 pF	–	0.27	–	V/ns	
Output slew rate							
	Slew = 4	V_{DD_IO} = 2.8V, PLL bypass, 6 MHz EXTCLK, C_{LOAD} = 35 pF	–	0.229	–	V/ns	
		V_{DD_IO} = 1.8V, PLL bypass, 6 MHz EXTCLK, C_{LOAD} = 35 pF	–	0.112	–	V/ns	

Notes:
1. V_{IH}/V_{IL} restrictions apply.
2. Based on lab measurements. Could vary with noisier system-level electronics.

Figure 36: Parallel Pixel Bus Timing Diagram



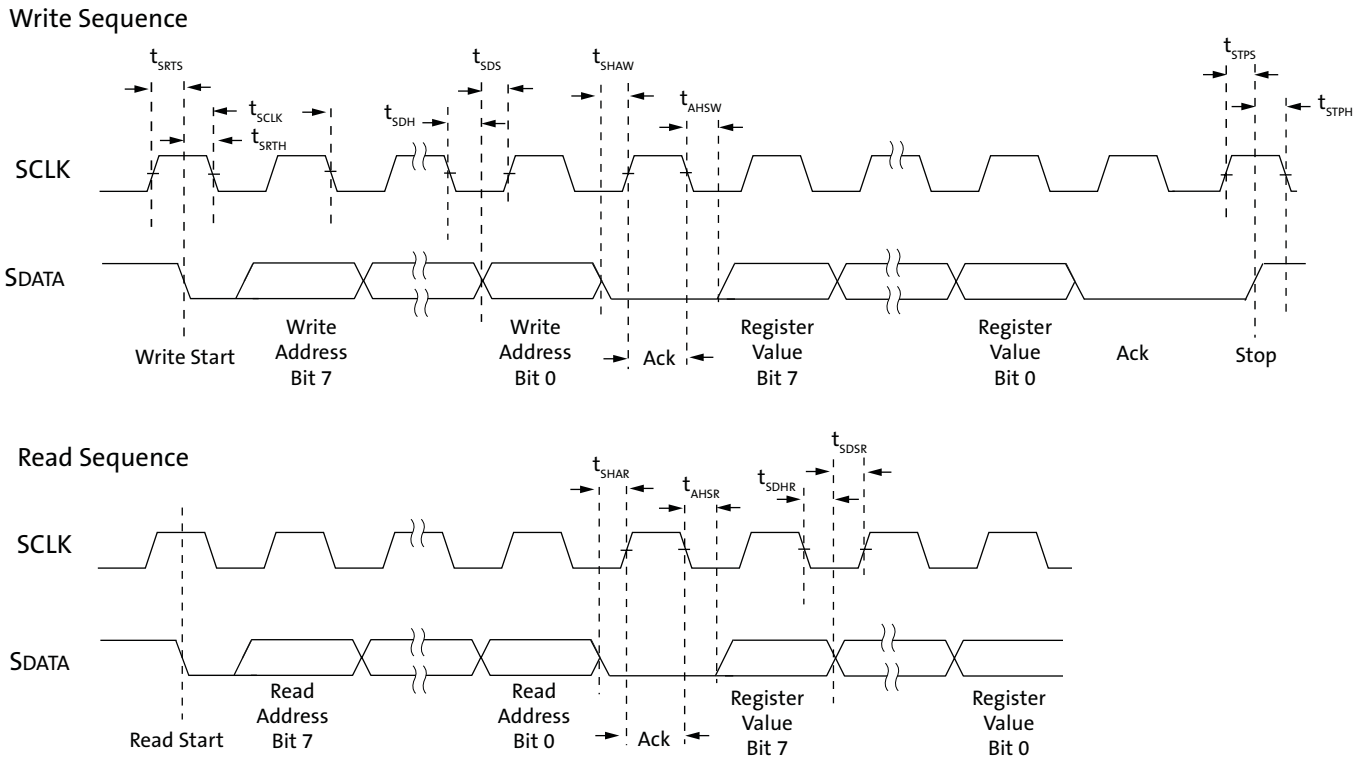
Notes:
1. FRAME_VALID leads LINE_VALID by 6 PIXCLKs.
2. FRAME_VALID trails LINE_VALID by 6 PIXCLKs.
3. Dout[7:0], FRAME_VALID, and LINE_VALID are shown with respect to the falling edge of PIXCLK. This feature is programmable and Dout[7:0], FRAME_VALID, and LINE_VALID can be synchronized to the rising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

Table 24: Two-Wire Serial Interface Timing Data

^{f}EXTCLK = 50 MHz; V_{DD} = 1.8V; V_{DD_IO} = 1.8V; V_{AA} = 2.8V; 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		10	–	2.5	µs
	SCLK duty cycle		45	50	55	%
^{t}r	SCLK/SDATA rise time		–	–	300	ns
^{t}SRTS	Start setup time	Master write to slave	600	–	–	
^{t}SRTH	Start hold time	Master write to slave	300	–	–	ns
^{t}SDH	SDATA hold	Master write to slave	300	–	650	ns
^{t}SDS	SDATA setup	Master write to slave	300	–	–	ns
^{t}SHAW	SDATA hold to ack	Master read from slave	150	–	–	ns
^{t}AHSW	Ack hold to SDATA	Master read from slave	150	–	–	ns
^{t}STPS	Stop setup time	Master write to slave	300	–	–	ns
^{t}STPH	Stop hold time	Master write to slave	600	–	–	ns
^{t}SHAR	SDATA hold to ack	Master write to slave	300	–	–	ns
^{t}AHSR	Ack hold to SDATA	Master write to slave	300	–	–	ns
^{t}SDHR	SDATA hold	Master read from slave	300	–	650	ns
^{t}SDSR	SDATA setup	Master read from slave	350	–	–	ns

Figure 37: Two-Wire Serial Bus Timing Parameters



MIPI AC and DC Electrical Characteristics

Table 25: MIPI High-Speed Transmitter DC Specifications

Symbol	Parameter	Min	Nom	Max	Units
V_{CMTX}	HS transmit static common-mode voltage	150	200	250	mV
$\lvert\Delta V_{CMTX}(1,0)\rvert$	VCMTX mismatch when output is Differential-1 or Differential-0			5	mV
$\lvert V_{OD}\rvert$	HS transmit differential voltage	140	200	270	mV
$\lvert\Delta V_{OD}\rvert$	VOD mismatch when output is Differential-1 or Differential-0			10	mV
V_{OHHS}	HS output high voltage			360	mV
Z_{OS}	Single-ended output impedance	40	50	62.5	Ω
ΔZ_{OS}	Single-ended output impedance mismatch			10	%

Table 26: MIPI High-Speed Transmitter AC Specifications

Parameter	Description	Min	Nom	Max	Units
	Data bit rate			768	Mb/s
t_R and t_F	20%-80% rise time and fall time			0.3	UI
		150			ps

Table 27: MIPI Low-Power Transmitter DC Characteristics

Parameter	Description	Min	Nom	Max	Units
V_{OL}	Thevenin output high level	1.1	1.2	1.3	V
V_{OH}	Thevenin output low level	-50		50	mV
Z_{OLP}	Output impedance of LP transmitter	110			Ω

Table 28: MIPI Low-Power Transmitter AC Characteristics

Symbol	Parameter		Min	Nom	Max	Units
TRLP/TFLP	15%-85% rise time and fall time				25	ns
TREOT	30%-85% rise time and fall time				35	ns
TLP-PULSE-TX	Pulse width of the LP exclusive-OR clock	First LP exclusive-OR clock pulse after Stop state or last pulse before Stop state	40			ns
		All other pulses	20			ns
TLP-PER-TX	Period of the LP exclusive-OR clock		90			ns
δV/δtSR	Slew rate @ C_{LOAD} = 70pF				150	mV/ns
	Slew rate @ C_{LOAD} = 0 to 70pF (Rising Edge Only)		30			mV/ns
	Slew rate @ C_{LOAD} = 0 to 70pF (Rising Edge Only)		30 – 0.075 * (VO,INST-700)			mV/ns
C_{LOAD}	Load capacitance		0		70	pF

Table 29: Clock Signal Specification

Symbol	Parameter	Min	Typ	Max	Units
UIINST	UI instantaneous			12.5	ns

Table 30: Data-Clock Timing Specifications

Symbol	Parameter	Min	Typ	Max	Units
TSKEW	Data to Clock Skew (measured at transmitter)	-0.15		0.15	UIINST

Package Dimensions

Table 31: Package Dimensions

Parameter	Symbol	Nominal	Min	Max	Nominal	Min	Max
		Millimeters			Inches		
Package Body Dimension X	A	4.64815	4.62315	4.67315	0.18300	0.18201	0.18398
Package Body Dimension Y	B	3.84775	3.82275	3.87275	0.15149	0.15050	0.15247
Package Height	C	0.690	0.645	0.735	0.02717	0.02539	0.02894
Cavity height (glass to pixel distance)	C4	0.041	0.037	0.045	0.00161	0.00146	0.00177
Glass Thickness	C3	0.400	0.390	0.410	0.01575	0.01535	0.01614
Package Body Thickness	C2	0.570	0.535	0.605	0.02244	0.02106	0.02382
Ball Height	C1	0.120	0.100	0.140	0.00472	0.00394	0.00551
Ball Diameter	D	0.230	0.200	0.260	0.0090	0.00787	0.01023
Total Ball Count	N	55					
Ball Count X axis	N1	8					
Ball Count Yaxis	N2	7					
UBM	U	0.240	0.230	0.250	0.009449	0.00906	0.00984
Pins Pitch X axis	J1	0.520	0.510	0.530	0.020472	0.02008	0.02087
Pins Pitch Y axis	J2	0.520	0.510	0.530	0.020472	0.02008	0.02087
BGA ball center to package center offset in X-direction	X	0	–0.025	0.025	0	–0.00098	0.00098
BGA ball center to package center offset in Y-direction	Y	0	–0.025	0.025	0	–0.00098	0.00098
Edge to Ball Center Distance along X	S1	0.504	0.474	0.534	0.01985	0.01866	0.02103
Edge to Ball Center Distance along Y	S2	0.364	0.334	0.394	0.01433	0.01314	0.01551

Note: Package center = die center.

Figure 38: Package Mechanical Drawing

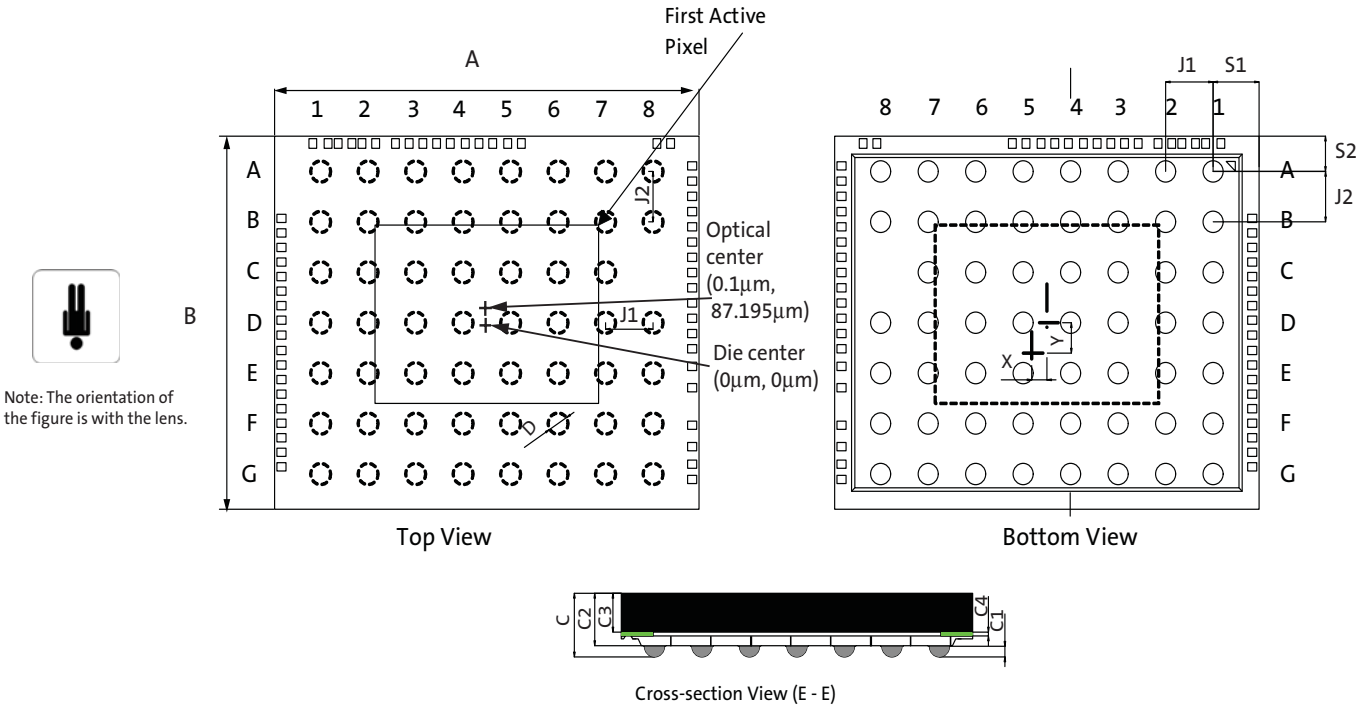


Table 32: Ball Matrix

	1	2	3	4	5	6	7	8
A	VAA	Reserved[1]	DOUT[6]	DOUT[4]	DOUT[2]	VDD	DOUT[1]	VDD
B	GND	VAA	VDD_IO	DOUT[5]	DOUT[3]	GND	DOUT[0]	VDD_IO
C	VDD	OE_BAR	AGND	GND	VDD_IO	FV	LV	
D	CONFIG	SCLK	SDATA	DOUT[7]	Reserved[1]	DOUT_LSB1	GND	VDD
E	VDD_IO	CHAIN	Reserved[1]	SADDR	RESET_ BAR	DOUT_LSB0	GND	VDD_PHY
F	EXTCLK	PIXCLK	GND	TRST_BAR	DATA_N	DATA_P	CLK_P	CLK_N
G	VDD	FLASH	VDD	PGND[2]	PGND[2]	VDD_PLL	GND_PLL	GND_PLL

Notes:
1. Do not use.
2. To be used for EMI shielding.

Revision History

Rev. E 4/13/11

- Corrected typo in Table 31, "Package Dimensions," on page 62 (changed Ball Diameter nominal, millimeters to 0.230)

Rev. D 3/11/11

- Updated to Production
- Updated Figure 2: "Typical Configuration," on page 8 and added note 9
- Updated Table 3, "Pin Descriptions," on page 9
- Updated Table 4, "Power-Up and Power-Down Signal Timing," on page 11
- Updated "Serial Port" on page 17
- Added 3rd paragraph to "Two-Wire Serial Interface" on page 47
- Added "Patch RAM" on page 51
- Updated Table 22, "Standby Current Consumption (Parallel and MIPI)," on page 57
- Updated Table 25, "MIPI High-Speed Transmitter DC Specifications," on page 60
- Updated Table 26, "MIPI High-Speed Transmitter AC Specifications," on page 60
- Updated Table 27, "MIPI Low-Power Transmitter DC Characteristics," on page 60
- Updated Table 28, "MIPI Low-Power Transmitter AC Characteristics," on page 60
- Added Table 29, "Clock Signal Specification," on page 60
- Added Table 30, "Data-Clock Timing Specifications," on page 61
- Updated Table 31, "Package Dimensions," on page 62

Rev. C 12/16/10

- Applied new Aptina template
- Updated Table 1, "Key Parameters," on page 1
- Updated Table 2, "Available Part Numbers," on page 1
- Updated Figure 4: "Power-Up and Power-Down Sequence," on page 11
- Updated Table 4, "Power-Up and Power-Down Signal Timing," on page 11
- Updated Table 5, "Status of Output Signals During Hard Reset, Soft Standby, and Power Off," on page 12
- Updated "Uncompressed Raw Bayer Bypass Output" on page 42
- Updated "UVC Interface" on page 43
- Updated Table 21, "Operating Current Consumption," on page 56
- Updated Table 22, "Standby Current Consumption (Parallel and MIPI)," on page 57
- Updated Table 23, "AC Electrical Characteristics," on page 58
- Updated Figure 36: "Parallel Pixel Bus Timing Diagram," on page 58
- Updated Table 31, "Package Dimensions," on page 62

Rev. B, Preliminary 9/9/10

- Added Figure 35: "Typical Quantum Efficiency," on page 53
- Removed Exposure Control section
- Updated"Auto Exposure" on page 37
- Removed old Figure 20
- Updated Figure 13: "Three Rows in Normal and Row Mirror Readout Mode," on page 21
- Add Figure 35: "Typical Quantum Efficiency," on page 53
- Updated Table , "," on page 60
- Updated Table 23, "AC Electrical Characteristics," on page 58

- Updated Figure 2: “Typical Configuration,” on page 8
- Updated Table 25, “MIPI High-Speed Transmitter DC Characteristics,” on page 60
- Updated Table 23, “AC Electrical Characteristics,” on page 58
- Added Table 24, “Standby Current Consumption (Parallel and MIPI),” on page 69
- Replaced Table 21, “Operating Current Consumption,” on page 56
- Updated Table 20, “DC Electrical Characteristics,” on page 55
- Added section “Soft Standby Mode” on page 14
- Updated Table 6, “Hard Reset,” on page 13 and Table 7, “Soft Reset Signal Timing,” on page 14
- Added CSP information
- Updated Table 1, “Key Parameters,” on page 1
- Updated Table 2, “Available Part Numbers,” on page 1
- Updated Table 6, “Hard Reset,” on page 13 notes
- Updated “Soft Reset” on page 14
- Updated Table 7, “Soft Reset Signal Timing,” on page 14
- Added “Serial Port” on page 17
- Updated “Binning and Summing” on page 25
- Added Figure 19: “Pixel Binning and Summing,” on page 25
- Replaced Figure 24: “Automatic Gamma Curve,” on page 31
- Removed old figures 26, 27, and 28
- Replaced Table 17, “Chief Ray Angle,” on page 52
- Updated Table 23, “AC Electrical Characteristics,” on page 58
- Added “Package Dimensions” on page 62

Rev. A 9/29/09

- Initial release

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