1/6-Inch VGA CMOS Digital Image Sensor

PART NUMBER: MT9V012

Features

- DigitalClarity™ CMOS Imaging Technology
- High frame rate
- Superior low-light performance
- Low dark current
- Simple two-wire serial interface
- Auto black level calibration
- Operating Modes: Snapshot and flash control, high frame rate preview, electronic panning
- Programmable Controls: Gain, frame size/rate, exposure, left–right and top–bottom image reversal, window size, and panning
- Data Interfaces: parallel and low-voltage differential signaling (LVDS)
- Applications
- Cellular Phones
- PC Cameras
- PDAs
- Toys and other battery-powered products

General Description

The Micron® Imaging MT9V012 is an oversize VGA-format CMOS active-pixel digital image sensor with a pixel array of 649H x 489V. It incorporates sophisticated on-chip camera functions such as windowing, mirroring, column and row skip modes, and snapshot mode. It is programmable through a simple two-wire serial interface and has very low power consumption.

The VGA CMOS image sensor features DigitalClarity—Micron's breakthrough low-noise CMOS imaging technology that achieves CCD image quality (based on signal-to-noise ratio and low-light sensitivity) while maintaining the inherent size, cost, and integration advantages of CMOS.

Table 1: Key Performance Parameters

Parameter		Typical Value
Optical Format		1/6-inch VGA (4:3)
Active Imager Size		2.30mm(H) x 1.77mm(V), 2.88mm Diagonal
Active Pixels		640H x 480V
Pixel Size		3.6µm x 3.6µm
Color Filter Array		RGB Bayer Pattern
Shutter Type		Electronic Rolling Shutter (ERS)
Maximum Data Rate/ Master Clock		13.5 MPS/27 MHz
Frame Rate	VGA (640 x 480)	Programmable up to 30 fps
	CIF (352 x 288)	Programmable up to 60 fps
ADC Resolution		10-bit, on-chip
Responsivity		1.0 V/lux-sec (550nm)
Dynamic Range		>71dB
SNR_{MAX}		44dB
Supply Voltage	Analog	2.50V–3.10V (2.80V nominal)
	I/O and Digital	1.70V–1.90V (1.80V nominal) or 2.50V–3.10V (2.80V nominal)
Power Consumption		<55mW at 2.8V, 27 MHz, 30 fps and VGA resolution
Operating Temperature		-30°C to +70°C

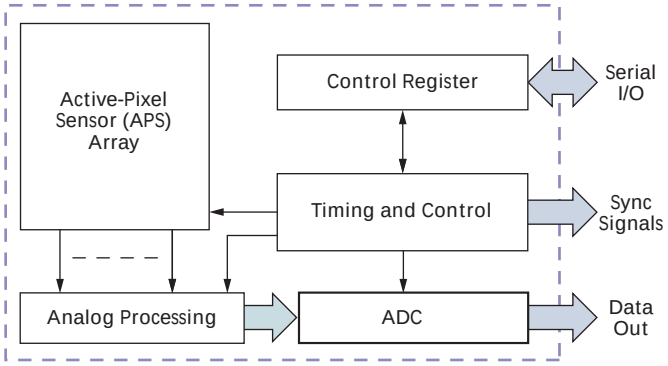
When operated in its default mode, the sensor generates a VGA image at 30 frames per second (fps). An on-chip analog-to-digital converter (ADC) generates a 10-bit value for each pixel. The pixel data is output on a 10-bit output bus and qualified by an output data clock (PIXCLK), together with LINE_VALID and FRAME_VALID signals. A FLASH output strobe is provided to allow an external Xenon or LED light source to synchronize with the sensor exposure time. The sensor can be programmed by the user to control the frame size, exposure, gain setting and other parameters.

Table of Contents

List of Figures

List of Tables

Figure 1: Block Diagram



Functional Overview

The MT9V012 is a progressive-scan sensor that generates a stream of pixel data qualified by LINE_VALID and FRAME_VALID signals. It uses an input master clock of 27 MHz (nominal). The data rate (pixel clock) is one half of the master clock frequency, which means that one pixel is generated every two master clock cycles. Figure 1 shows the sensor block diagram.

The core of the sensor is an 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 that row integrate incident light. The exposure is controlled by varying the time interval between reset and readout. Once a row has been read, the data from the columns is sequenced through an analog signal chain (providing offset correction and gain), and then through an ADC. The output from the ADC is a 10-bit value for each pixel in the array. The ADC output passes through a digital processing signal chain (which provides further offset correction, applies digital gain, and may perform pixel defect correction).

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

The sensor contains a set of 16-bit control and status registers that can be used to control many aspects of the sensor behavior. These registers can be accessed through a two-wire serial interface. In this document, registers are specified either by name (e.g., column start) or by register address (e.g., Reg0x04). Fields within a register are specified by bit or by bit range (e.g., Reg0x20[0] or Reg0x0B[13:0]). Table 6, Register Description, on page 23, describes the control and status registers.

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

The MT9V012 supports two different functional modes of operation:

- Default mode: the sensor generates a VGA-sized image by default, with 10 parallel data outputs per pixel, and separate LINE_VALID, FRAME_VALID, and PIXCLK outputs. All timing control is performed on-chip.
- Serial mode: the sensor generates a VGA-sized image by default. Pixel data, LINE_VALID, and FRAME_VALID are encoded into a single serial data stream that uses a two-signal low-voltage differential signalling (LVDS) interface. All timing control is performed on-chip.

Operating Modes

The functional operating mode of the MT9V012 is controlled by the MODE1 and MODE0 inputs (Table 2). These inputs should be driven to a static logic 1 or static logic 0 level during normal operation.

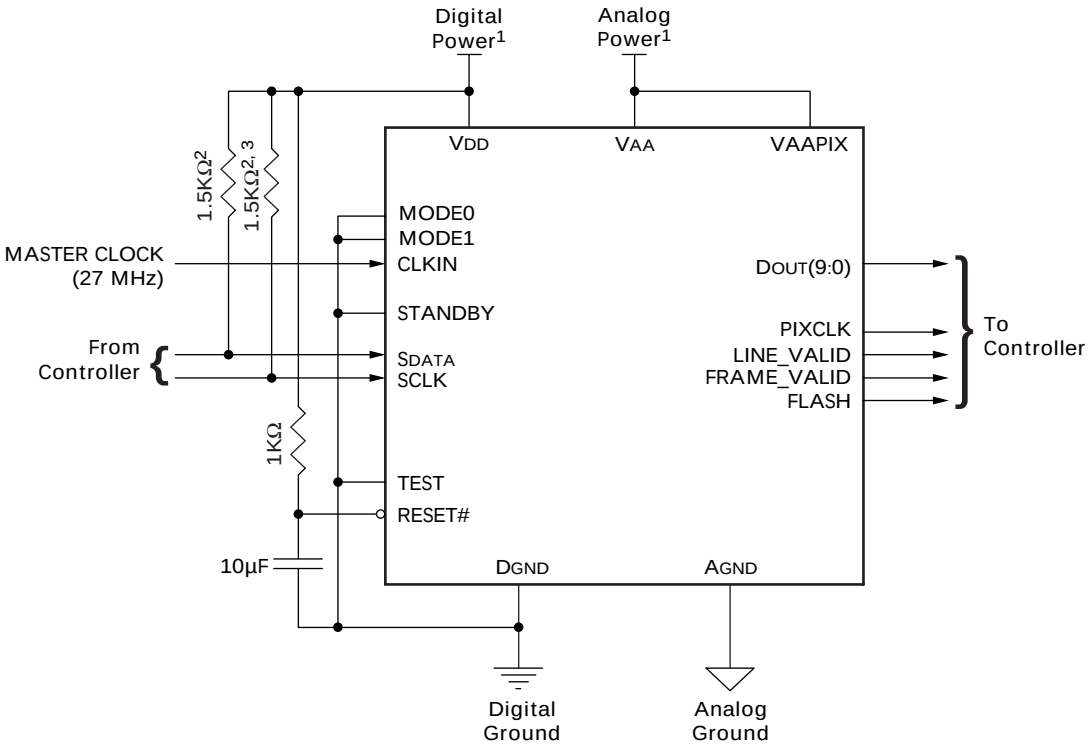
Table 2: Functional Mode Selection

Mode1	Mode0	Description
0	0	Selects default mode
0	1	Selects serial mode
1	0	Not used
1	1	Not used

Default Mode

This section shows a typical configuration schematic for the MT9V012 operating in default mode.

Figure 2: Typical Configuration: Default Mode

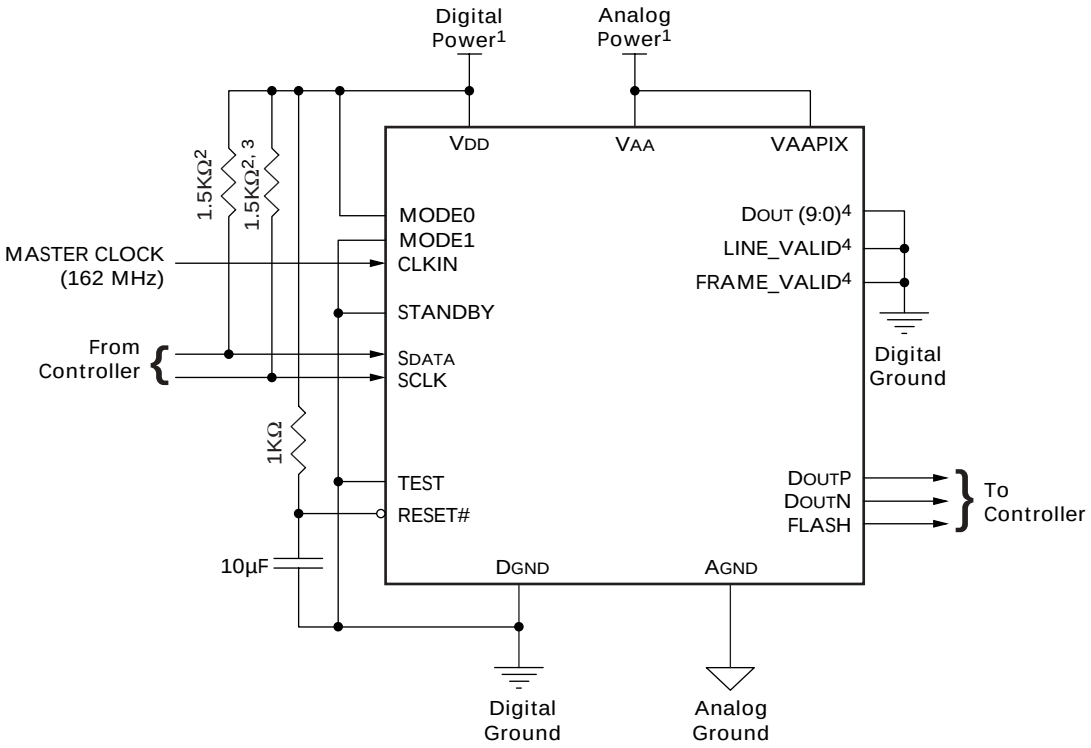


Notes: 1. All power supplies should be adequately decoupled.
2. Resistor value 1.5KΩ is recommended, but may be greater for slower two-wire speed.
3. This pull-up resistor is not required if the controller drives a valid logic level on SCLK at all times.

Serial Mode

This section shows a typical configuration schematic, including the ball diagram and ball description, for the MT9V012 operating in serial mode. This mode operates only at 2.5V to 3.1V VDD range.

Figure 3: Typical Configuration: Serial Mode



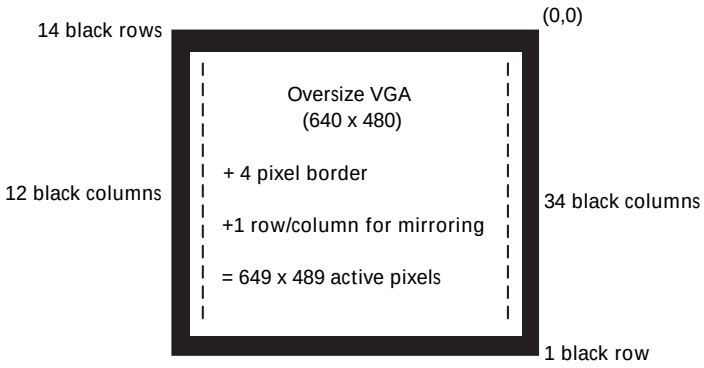
Notes:
1. All power supplies should be adequately decoupled.
2. Resistor value 1.5KΩ is recommended, but may be greater for slower two-wire speed.
3. This pull-up resistor is not required if the controller drives a valid logic level on SCLK at all times.
4. Connect to digital ground directly or through a 10K resistor. Some of these signals have on-die pull-down resistors in this mode and could be left unconnected.

Pixel Array Structure

The MT9V012 pixel array is configured as 695 columns by 504 rows (shown in Figure 4). The first 34 columns and the first 14 rows of pixels are optically black, and are used for the automatic black level adjustment (“Black Level Calibration” on page 40). The last 12 columns and the last row of pixels are also optically black. The optically active pixels are used as follows:

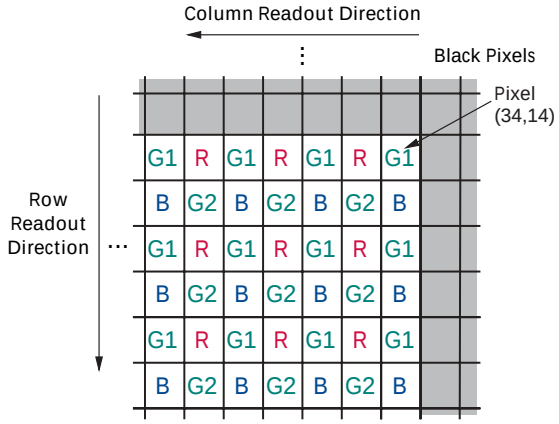
A VGA image (640 columns by 480 rows) is generated, starting at row 18, column 38. A four-pixel boundary of active pixels can be enabled around the image to avoid boundary effects during color interpolation and correction. An additional row and column of active pixels is also provided for use during horizontally- and/or vertically mirrored readout. During mirrored readout, the region of active pixels that is used to generate the image is offset by one pixel in each mirrored direction so that the readout always starts on the same color pixel.

Figure 4: Pixel Array



The MT9V012 uses a Bayer color pattern, as shown in Figure 5. The even-numbered rows contain green and red color pixels; odd-numbered rows contain blue and green color pixels. Even-numbered columns contain green and blue color pixels; odd-numbered columns contain red and green color pixels.

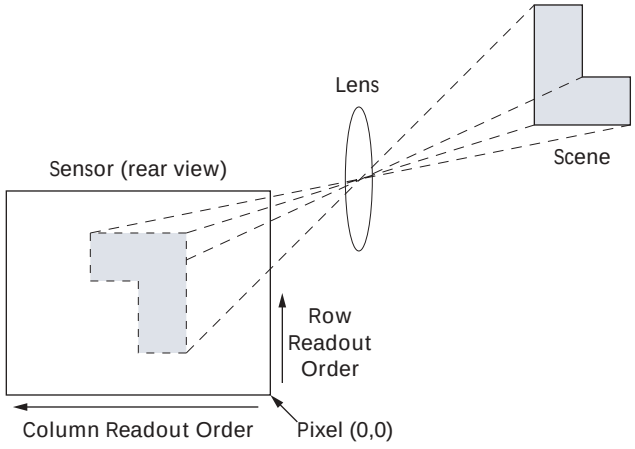
Figure 5: Pixel Color Pattern Detail (Top Right Corner)



Default Readout Order

By convention, the MT9V012 pixel array is shown with pixel (0,0) in the top right-hand corner (see Figure 5). This reflects the actual layout of the array on the die. When the sensor is imaging, the active surface of the sensor faces the scene, as shown in Figure 6. When the image is read out of the sensor, it is read one row at a time, with the rows and columns sequenced as shown in Figure 5. By convention, data from the sensor is shown with the first pixel read out—pixel (34,14) in the case of the MT9V012—in the top left-hand corner (Figure 7).

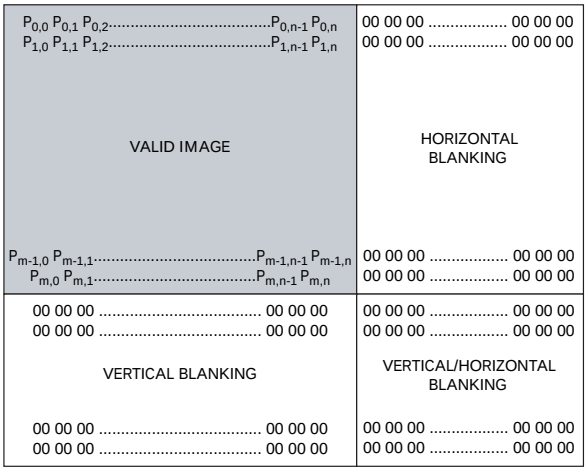
Figure 6: Imaging a Scene



Output Data Format (Default Mode)

The MT9V012 image data is read out in a progressive scan. In default mode, valid image data is surrounded by horizontal blanking and vertical blanking, as shown in Figure 7. The amount of horizontal blanking and vertical blanking is programmable; LINE_VALID is HIGH during the shaded region of the figure. FRAME_VALID timing is described in the next section.

Figure 7: Spatial Illustration of Image Readout



Output Data Timing (Default Mode)

The MT9V012 output data is synchronized with the PIXCLK output. When LINE_VALID is HIGH, one pixel datum is output on the 10-bit DOUT output every PIXCLK period. By default, the PIXCLK signal runs at one-half the frequency of the master clock, CLKIN, and its rising edges occur one-half of a master clock period after transitions on LINE_VALID, FRAME_VALID, and DOUT (see Figure 8). This allows PIXCLK to be used as a clock to sample the data. PIXCLK is continuously enabled, even during the blanking period. The MT9V012 can be programmed to delay the PIXCLK edge relative to the DOUT transitions from 0 to 3.5 master clocks, in steps of one-half of a master clock. This can be achieved by programming the corresponding bits in Reg0x0A. The parameters P, A, and Q in Figure 9 are defined in Table 3 on page 12.

Figure 8: Pixel Data Timing Example

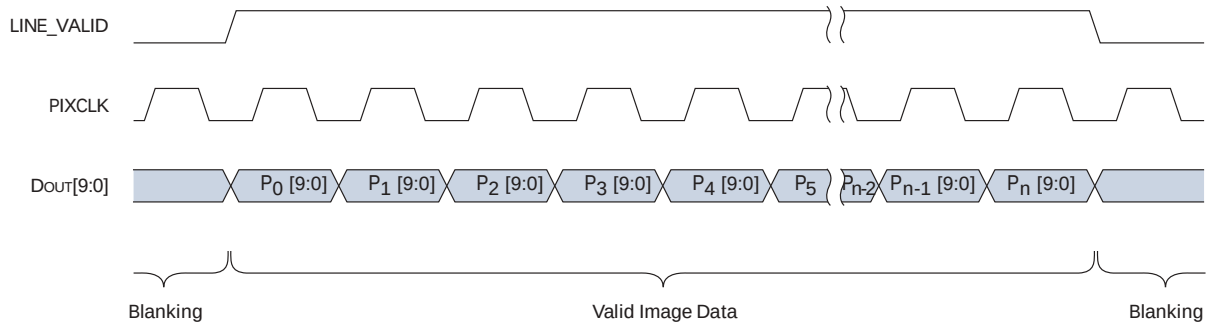


Figure 9: Row Timing and FRAME_VALID/LINE_VALID Signals

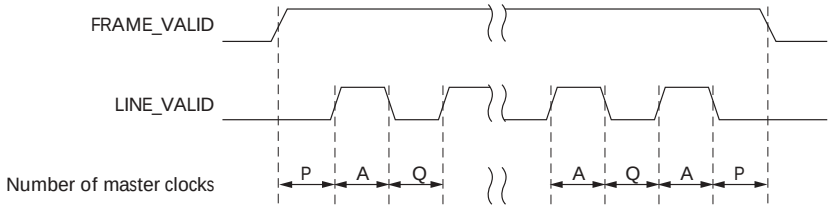


Table 3: Frame Time

Parameter	Name	Equation	Default Timing at 27 MHz
HBLANK_REG	Horizontal Blanking Register	Reg0x07 if Reg0xC8[0] = 0 Reg0x05 if Reg0xC8[0] = 1	0xF4 = 244 pixels
VBLANK_REG	Vertical Blanking Register	Reg0x8 if Reg0xC8[1] = 0 Reg0x6 if Reg0xC8[1] = 1	0x1D = 29 rows
PIXCLK_PERIOD	Pixel Clock Period	Reg0x0A[2:0] * 2	1 pixel clock = 2 master = 37.04ns
S	Skip Factor	For skip 2x mode: S = 2 For skip 4x mode: S = 4 otherwise, S = 1	1
A	Active Data Time	(Reg0x04/S) * PIXCLK_PERIOD	640 pixel clocks = 1,280 master = 47.41µs
P	Frame Start/End Blanking	6 * PIXCLK_PERIOD	6 pixel clocks = 12 master = 0.44µs
Q	Horizontal Blanking	HBLANK_REG * PIXCLK_PERIOD	244 pixel clocks = 488 master = 18.07µs
A + Q	Row Time	((Reg0x04/S) + HBLANK_REG) * PIXCLK_PERIOD	884 pixel clocks = 1,768 master = 65.48µs
V	Vertical Blanking	VBLANK_REG * (A + Q) + (Q - 2*P)	25,868 pixel clocks = 51,736 master = 1.91ms
Nrows * (A+Q)	Frame Valid Time	(Reg0x03/S) * (A + Q) - (Q - 2*P)	424,088 pixel clocks = 848,176 master = 31.41ms
F	Total Frame Time	((Reg0x03/S) + VBLANK_REG) * (A + Q)	449,956 pixel clocks = 899,912 master = 33.33ms

The sensor timing (Table 3) is shown in terms of pixel clock and master clock cycles (see Figure 8 on page 11). The recommended master clock frequency is 27 MHz. The vertical blanking and total frame time equations assume that the number of integration rows (Reg0x09) is less than the number of active rows, plus blanking rows (Reg0x03 + VBLANK_REG). If this is not the case, the number of integration rows must be used instead, to determine the frame time, as shown in Table 4.

Table 4: Frame—Long Integration Time

Parameter	Name	Equation (master clock)	Default Timing
V'	Vertical Blanking (long integration time)	(Reg0x09 – (Reg0x03)/S)) * (A + Q) + (Q - 2*P)	25,868 pixel clocks = 51,736 master = 1.91ms
F'	Total Frame Time (long integration time)	(Reg0x09) * (A + Q)	449,956 pixel clocks = 899,912 master = 33.33ms

Output Data Format (Serial Mode)

The MT9V012 image data is read out in a progressive scan. In serial mode, valid image data is surrounded by horizontal blanking and vertical blanking, as shown in Figure 7 on page 10. However, unlike default mode, serial mode provides pixel data and timing strobes combined into a single serial bit stream. Electrically, this bit stream uses LVDS on the DOUTP and DOUTN output signals.

In serial mode, each pixel is encoded as a 12-bit value by adding a start bit and a stop bit. The sensor CLKIN input runs at the serial bit-rate and is used within the sensor to clock a data serializer circuit; it is divided within the sensor so that most of the circuitry runs at the same rate as in default mode. In serial mode, the pixel rate is fixed at one-half the input frequency; therefore, CLKIN runs at 6 x 27 MHz = 162 MHz.

Output Data Timing (Serial Mode)

The default frame timing in serial mode is identical to the frame timing in default mode. A special three-character "start-of-frame" sequence—0x3FF, 0x0, and 0x3FF—is transmitted to indicate the assertion of FRAME_VALID. Pixel data, LINE_VALID, and FRAME_VALID can be reconstructed externally by detecting the start-of-frame sequence, and using a state machine and counters to identify the active regions of the frame.

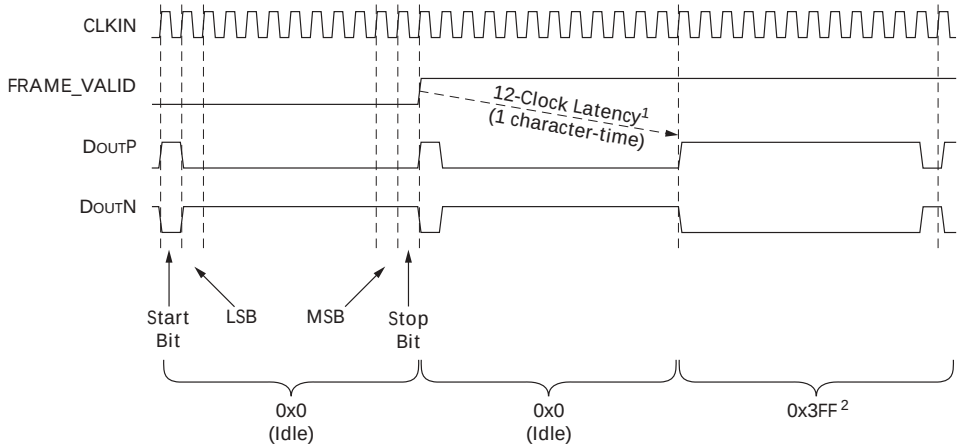
Figure 10 on page 13 shows the beginning of a start-of-frame sequence. It shows the latency introduced in the parallel-to-serial conversion, and the way in which start and stop bits are used to frame 10-bit pixel data. Figure 11 on page 14 shows the serial data stream at the start of a line. In this figure, each 12-bit serial character is represented by its 10-bit payload. In both figures, the LINE_VALID and FRAME_VALID signals are shown for reference only; these signals are not available in serial mode.

The most effective method for detecting the start-of-frame sequence is to look for a continuous sequence of idle (0x0) characters before looking for the 0x3FF, 0x0, 0x3FF sequence. The start-of-frame sequence indicates the start of a frame without ambiguity, as it can never occur as part of a pixel data stream.

In addition, correct operation in serial mode requires that various register settings are left at their default values. For example, Reg0x0A and Reg0x20[15:14].

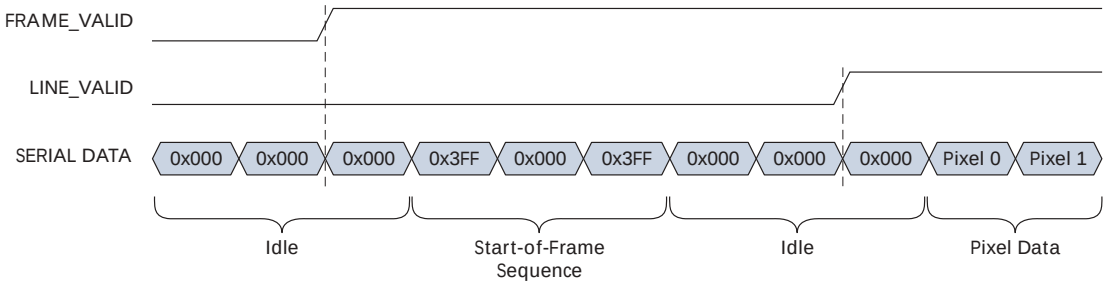
The sensor timing in serial mode is calculated in exactly the same way as for default mode. See "Output Data Timing (Default Mode)" on page 11.

Figure 10: Start of Frame: Serial Mode



Notes: 1. Latency between parallel event and equivalent serial event.
2. First character in start-of-frame sequence.

Figure 11: Start of Line: Serial Mode



Two-Wire Serial Interface

The two-wire serial register interface enables read/write access to control and status registers within the MT9V012.

The interface protocol uses a master/slave model in which a master controls one or more slave devices. The sensor acts as a slave device. The master generates a clock (SCLK) that is an input to the sensor and used to synchronize transfers. Data is transferred between the master and the slave on a bidirectional signal (SDATA). The SDATA signal is pulled up to VDD off-chip by a 1.5KΩ resistor. Either the slave or master device can drive the SDATA line LOW—the interface protocol determines which device is allowed to drive the SDATA line at any given time.

Protocol

The two-wire serial interface defines several different transmission codes, as follows:

- a start bit
- the slave device 8-bit address
- a(an) (no) acknowledge bit
- an 8-bit message
- a stop bit

Sequence

A typical read or write sequence begins by the master sending a start bit. After the start bit, the master sends the 8-bit slave-device address. The last bit of the address determines if the request will be a read or a write, where a “0” indicates a write and a “1” indicates a read. The slave device acknowledges receipt of the address by sending an acknowledge bit back to the master.

If the request was a write, the master then transfers the 8-bit register address to which a write should take place. The slave sends an acknowledge bit to indicate that the register address has been received. The master then transfers the data, 8 bits at a time, with the slave sending an acknowledge bit after each 8 bits. The MT9V012 uses 16-bit data for its internal registers, thus requiring two 8-bit transfers to write to one register. After 16 bits are transferred, the register address is automatically incremented, so that the next 16 bits are written to the next register address. The master stops writing by sending a start or stop bit.

A typical read sequence is executed as follows. The master sends the write mode slave address and 8-bit register address, just as in the write request. The master then sends a start bit and the read mode slave address, and clocks out the register data, 8 bits at a time. The master sends an acknowledge bit after each 8-bit transfer. The register address is auto-incremented after every 16 bits is transferred. The data transfer is stopped when the master sends a no-acknowledge bit.

Bus Idle State

The bus is idle when both the data and clock lines are HIGH. Control of the bus is initiated with a start bit, and the bus is released with a stop bit. Only the master can generate the start and stop bits.

Start Bit

The start bit is defined as a HIGH-to-LOW data line transition while the clock line is HIGH.

Stop Bit

The stop bit is defined as a LOW-to-HIGH data line transition while the clock line is HIGH.

Slave Address

The 8-bit address of a two-wire serial interface device consists of 7 bits of address and 1 bit of direction. A “0” in the LSB of the address indicates write mode, and a “1” indicates read mode. The default slave addresses used by the MT9V012 are 0xBA (write address) and 0xBB (read address). Reg0x0D[10] can be used to select the alternate slave addresses 0x90 (write address) and 0x91 (read address).

Writes to Reg0x0D[10] are inhibited when STANDBY is asserted (all other writes proceed normally). This allows two sensors to co-exist as slaves on this interface, but they must be addressed independently. Enable this capability as follows:

- After RESET# is negated, both sensors will use the default slave address. Reads or writes on the serial register interface to the default slave address will be decoded by both sensors simultaneously.
- After reset, assert STANDBY to one sensor and negate STANDBY to the other sensor. Perform a write to Reg0x0D with bit 10 set. The sensor with STANDBY asserted will ignore the write to bit 10 and will continue to decode at the default slave address. The sensor with STANDBY negated will have its Reg0x0D[10] set and will respond to the alternate slave address for all subsequent READ and WRITE operations.

Data Bit Transfer

One data bit is transferred during each clock pulse. The serial interface clock pulse is provided by the master. The data must be stable during the high period of the two-wire serial interface clock—it can only change when the serial clock is LOW. Data is transferred 8 bits at a time, followed by an acknowledge bit.

Acknowledge Bit

The master generates the acknowledge clock pulse. The transmitter (which is the master when writing, or the slave when reading) releases the data line, and the receiver indicates an acknowledge bit by driving the data line LOW during the acknowledge clock pulse.

No-Acknowledge Bit

The no-acknowledge bit is generated when the data line is not driven LOW by the receiver during the acknowledge clock pulse. A no-acknowledge bit is used to terminate a read sequence.

Page Register

The MT9V012 two-wire serial interface and its associated protocols support an address space of 256 16-bit locations. This address space can be extended by a 3-bit page prefix, and controlled through accesses to Reg0xF0.

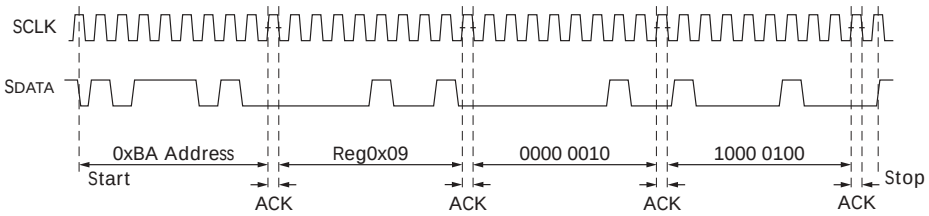
The paging mechanism is intended to allow access to other sets of registers when the sensor is embedded as part of a more complex integrated sub-system (for example, in an SOC). All of the registers within the MT9V012 are accessible on page 0 (the default page).

Two-Wire Serial Interface Sample Write and Read Sequences

16-Bit Write Sequence

A typical write sequence for writing 16 bits to a register is shown in Figure 12. A start bit given by the master starts the sequence, followed by the write address. The image sensor then sends an acknowledge bit and expects the register address to come first, followed by the 16-bit data. After each 8-bit transfer, the image sensor sends an acknowledge bit. All 16 bits must be written before the register is updated. After 16 bits are transferred, the register address is automatically incremented so that the next 16 bits are written to the next register. The master stops writing by sending a start or stop bit.

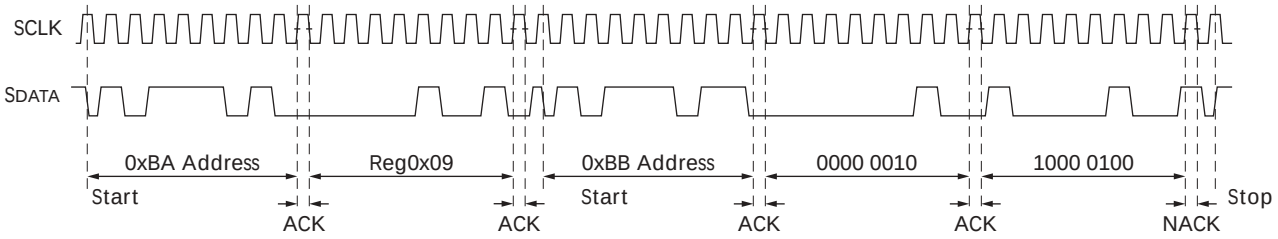
Figure 12: Timing Diagram Showing a Write to Reg0x09, Value 0x0284



16-Bit Read Sequence

A typical read sequence is shown in Figure 13. First the master writes the register address, as in a write sequence. Then a start bit and the read address specify that a read is about to happen from the register. The master clocks out the register data 8 bits at a time. The master sends an acknowledge bit after each 8-bit transfer. The register address should be incremented after every 16 bits is transferred. The data transfer is stopped when the master sends a no-acknowledge bit.

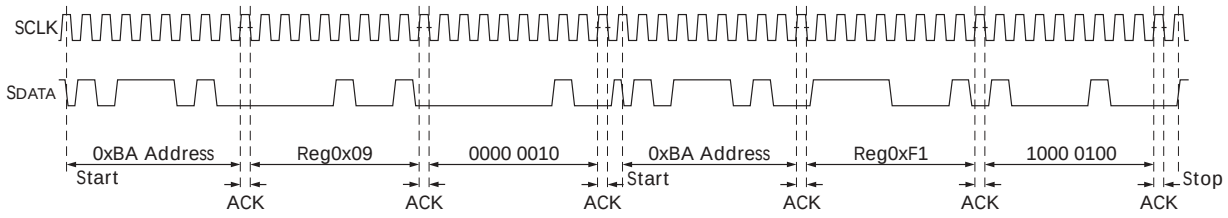
Figure 13: Timing Diagram Showing a Read from Reg0x09; Returned Value 0x0284



8-Bit Write Sequence

To be able to write one byte at a time to the register, a special register address is added. The 8-bit write is done by writing the upper 8 bits to the desired register, then writing the lower 8 bits to the special register address (Reg0xF1). The register is not updated until all 16 bits have been written. It is not possible to update just half of a register. Figure 14 shows a typical sequence for 8-bit writes. The second byte is written to the special register (Reg0xF1).

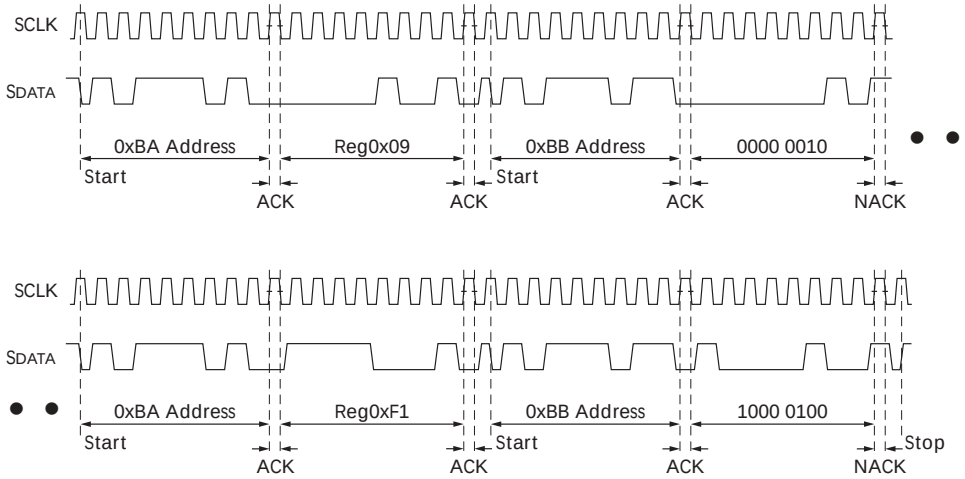
Figure 14: Timing Diagram Showing a Write to Reg0x09, Value 0x0284



8-Bit Read Sequence

To read one byte at a time, the same special register address is used for the lower byte. The upper 8 bits are read from the desired register. By following this with a read from the special register (Reg0xF1), the lower 8 bits are accessed (Figure 15). The master sets the no-acknowledge bits.

Figure 15: Timing Diagram Showing a Read from Reg0x09; Returned Value 0x0284



Registers

The MT9V012 provides a register address space of 256 locations.

Register Map

Table 5 shows the locations used within the address space. Locations that are not shown in the table are reserved for future use; they should not be read from or written to. The effect of reading from or writing to "Reserved" registers is UNDEFINED and may include the possibility of causing permanent electrical damage to the sensor.

Table 5: Register List and Default Value

Register Number (Hex)	Description	Data Format (Binary)	Default Value (Hex)
0x00	Chip Version	0001 0010 0010 0010 (LSB)	0x1222
0x01	Row Start	0000 000d dddd dddd	0x000E
0x02	Column Start	0000 00dd dddd dddd	0x0026
0x03	Row Width	0000 000d dddd dddd	0x01E0
0x04	Column Width	0000 00dd dddd dddd	0x0280
0x05	Horizontal Blanking B	00dd dddd dddd dddd	0x00F4
0x06	Vertical Blanking B	0ddd dddd dddd dddd	0x001D
0x07	Horizontal Blanking A	00dd dddd dddd dddd	0x0234
0x08	Vertical Blanking A	0ddd dddd dddd dddd	0x010D
0x09	Shutter Width	dddd dddd dddd dddd	0x01FD
0x0A	Row Speed	ddd0 000d dddd 0ddd	0x0011
0x0B	Extra Delay	00dd dddd dddd dddd	0x0000
0x0C	Shutter Delay	00dd dddd dddd dddd	0x0000
0x0D	Reset	d000 0ddd dddd dd0d	0x0008
0x20	Read Mode B	dd00 00dd dddd dddd	0x0400
0x21	Read Mode A	0000 0000 0000 dd00	0x040C
0x22	Dark Col/Rows	0000 00dd dddd dddd	0x012B
0x23	Flash	00dd dddd dddd dddd	0x0608
0x24	Extra Reset	0d00 0000 0000 0000	0x4000
0x2B	Green1 Gain	0000 dddd dddd dddd	0x0020
0x2C	Blue Gain	0000 dddd dddd dddd	0x0040
0x2D	Red Gain	0000 dddd dddd dddd	0x0020
0x2E	Green2 Gain	0000 dddd dddd dddd	0x0020
0x2F	Global Gain	0000 dddd dddd dddd	0x0020
0x30	Row Noise	dddd dddd dddd dddd	0x042A
0x31	Reserved	—	0x1C00
0x32	Reserved	—	0x002A
0x33	Reserved	—	0x0341
0x34	Reserved	—	0xC009
0x35	Reserved	—	0x2022
0x36	Reserved	—	0xF0F0
0x37	Reserved	—	0x0000
0x3B	Reserved	—	0x0021
0x3C	Reserved	—	0x1A20
0x3D	Reserved	—	0x201E
0x3E	Reserved	—	0x2020

Table 5: Register List and Default Value (continued)

Register Number (Hex)	Description	Data Format (Binary)	Default Value (Hex)
0x3F	Reserved	—	0x1020
0x40	Reserved	—	0x2000
0x41	Reserved	—	0x00D7
0x42	Reserved	—	0x0777
0x58	Reserved	—	0x0000
0x59	Black Rows	0000 0000 dddd dddd	0x000C
0x5A	Reserved	—	0xE00A
0x5B	Dark G1 Average	0000 0000 0??? ????	
0x5C	Dark B Average	0000 0000 0??? ????	
0x5D	Dark R Average	0000 0000 0??? ????	
0x5E	Dark G2 Average	0000 0000 0??? ????	
0x5F	Calib Threshold	0ddd dddd 0ddd dddd	0x231D
0x60	Calib Control	d000 000d dddd dddd	0x0080
0x61	Calib Green1	0000 000d dddd dddd	0x0000
0x62	Calib Blue	0000 000d dddd dddd	0x0000
0x63	Calib Red	0000 000d dddd dddd	0x0000
0x64	Calib Green2	0000 000d dddd dddd	0x0000
0x65	Reserved	—	0x0000
0x70	Reserved	—	0x7B0A
0x71	Reserved	—	0x7B0A
0x72	Reserved	—	0x190E
0x73	Reserved	—	0x750F
0x74	Reserved	—	0x5732
0x75	Reserved	—	0x5634
0x76	Reserved	—	0x7335
0x77	Reserved	—	0x3012
0x78	Reserved	—	0x3012
0x79	Reserved	—	0x7506
0x7A	Reserved	—	0x770A
0x7B	Reserved	—	0x7809
0x7C	Reserved	—	0x7D06
0x7D	Reserved	—	0x3110
0x7E	Reserved	—	0x007E
0x7F	Reserved	—	0x7C01
0x80	Reserved	—	0x5904
0x81	Reserved	—	0x5904
0x82	Reserved	—	0x570A
0x83	Reserved	—	0x2D0B
0x84	Reserved	—	0x580B
0x85	Reserved	—	0x480E
0x86	Reserved	—	0x5B02
0x87	Reserved	—	0x005C
0xC8	Context Control	0000 0000 d000 dddd	0x000B
0xF0	Page Map	0000 0000 0000 0ddd	0x0000
0xF1	Bytewise Address	0000 0000 0000 0000	0x0000

Table 5: Register List and Default Value (continued)

Register Number (Hex)	Description	Data Format (Binary)	Default Value (Hex)
0xF5	Reserved	—	0x03FF
0xF6	Reserved	—	0x01FF
0xF7	Reserved	—	0x0000
0xF8	Reserved	—	0x0000
0xF9	Reserved	—	0x0000
0xFA	Reserved	—	0x0000
0xFB	Reserved	—	0x0000
0xFC	Reserved	—	0x0000
0xFD	Reserved	—	0x0000
0xFF	Chip Version	0001 0010 0010 0010	0x1222

Note: 1 = always 1
0 = always 0
d = programmable
? = read-only

Register Description

Table 6 provides a detailed description of the registers. Bit fields that are not identified in the table are read-only "0."

Double-Buffered Registers

Some sensor settings cannot be changed during frame readout. For example, changing Reg0x03 (row width) partway through frame readout would result in inconsistent LINE_VALID behavior. To avoid this, the MT9V012 double buffers many registers by implementing a "pending" and a "live" version. Reads and writes access the pending register. The live register controls the sensor operation.

The values in the pending registers are transferred to the live registers at a fixed point in the frame timing, called "frame start." Frame start is defined as the point at which the first dark row is read out. By default, this occurs 6 row times before FRAME_VALID goes HIGH. Reg0x22 enables the dark rows to be shown in the image, but this has no effect on the position of frame start.

In Table 6, the "Sync'd" column shows which registers or register fields are double-buffered in this way.

Reg0x0D[15] can be used to inhibit transfers from the pending to the live registers. This control bit should be used when the user wants to make many register changes and have them all take effect simultaneously.

Bad Frames

A bad frame is a frame where all rows do not have the same integration time, or where offsets to the pixel values have changed during the frame.

Many changes to the sensor register settings can cause a bad frame. For example, when Reg0x03 (row width) is changed, the new register value does not affect sensor behavior until the next frame start. However, the frame that would be read out at that frame start will have been integrated using the old row width, so reading it out using the new row width would result in a frame with an incorrect integration time.

By default, most bad frames are masked: LINE_VALID and FRAME_VALID are inhibited for these frames so that the vertical blanking time between frames is extended by the frame time.

In Table 6, the “Bad Frame” column shows where changing a register or register field will cause a bad frame. The following notation is used:

- N—No. Changing the register value will not produce a bad frame.
- Y—Yes. Changing the register value might produce a bad frame.
- YM—Yes; but the bad frame will be masked out unless the “show bad frames” feature (Reg0x0D[8])is enabled.

Changes to Integration Time

If the integration time (Reg0x09) is changed while FRAME_VALID is asserted for frame *N*, the first frame output using the new integration time is frame *(N + 2)*. The sequence is as follows:

1. During frame *N*, the new integration time is held in the Reg0x09 pending register.
2. At the start of frame (*N + 1*), the new integration time is transferred to the Reg0x09 live register. Integration for each row of frame (*N + 1*) has been completed using the old integration time.
3. The earliest time that a row can start integrating using the new integration time is immediately after that row has been read for frame (*N + 1*). The actual time that rows start integrating using the new integration time is dependent upon the new value of the integration time.
4. When frame (*N + 1*) is read out, it will have been integrated using the new integration time.

If the integration time is changed (Reg0x09 written) on successive frames, each value written will be applied for a single frame; the latency between writing a value and it affecting the frame readout remains at two frames.

Changes to Gain Settings

Usually, when the gain settings (Reg0x2B, Reg0x2C, Reg0x2D, Reg0x2E, and Reg0x2F) are changed, the gain is updated on the next frame start. When the integration time and the gain are changed at the same time, the gain update is held off by one frame so that the first frame output with the new integration time also has the new gain applied.

Table 6: Register Description

Bit	Bit Description		Default (hex)	Sync'd to Frame Start	Bad Frame
0x00/0xFF (0/255) Chip Version					
15:0	Chip Version	Chip version. Read-only.	1222	N	N
0x01 (1) Row Start					
8:0	Row Start	The first row to be read out (not counting any dark rows that may be read). To move the image window, set this register to the starting "Y" value. Setting a value less than eight is not recommended, as the dark rows should be read using Reg0x22.	12	Y	YM
0x02 (2) Column Start					
9:0	Column Start	The first column to be read out (not counting dark columns that may be read). To move the image window, set this register to the starting "X" value. Setting a value below Reg0x18 is not recommended, as readout of dark columns should be controlled by Reg0x22.	26	Y	YM
0x03 (3) Row Width					
8:0	Row Width	Number of rows in the image to be read out (not counting any dark rows or border rows that may be read). The minimum supported value is 2.	1E0	Y	YM
0x04 (4) Column Width					
9:0	Column Width	Number of columns in the image to be read out (not counting any dark columns or border columns that may be read). The minimum supported value is 9.	280	Y	YM
0x05 (5) Horizontal Blanking B					
10:0	Horizontal Blanking B	Number of blank columns in a row when context B is selected (Reg0xC8[0] = 1). The extra columns are added at the beginning of a row. See "Minimum Horizontal Blanking" on page 35.	F4	Y	YM
0x06 (6) Vertical Blanking B					
14:0	Vertical Blanking B	Number of blank rows in a frame when context B is selected (Reg0xC8[1] = 1). The minimum supported value is (6 + Reg0x22[2:0]). The actual vertical blanking time can be controlled by the shutter width (Reg0x9). See "Output Data Timing (Default Mode)" on page 11.	1D	Y	N
0x07 (7) Horizontal Blanking A					
10:0	Horizontal Blanking A	Number of blank columns in a row when context A is selected (Reg0xC8[0] = 0). The extra columns are added at the beginning of a row. See "Minimum Horizontal Blanking" on page 35.	234	Y	YM
0x08 (8) Vertical Blanking A					
14:0	Vertical Blanking A	Number of blank rows in a frame when context A is chosen (Reg0xC8[1] = 1). The minimum supported value is (6 + Reg0x22[2:0]). The actual vertical blanking time can be controlled by the shutter width (Reg0x9). See "Output Data Timing (Default Mode)" on page 11.	10D	Y	N
0x09 (9) Shutter Width					
15:0	Shutter Width	Integration time in number of rows. The integration time is also influenced by the shutter delay (Reg0x0C) and the overhead time.	1FD	Y	N

Table 6: Register Description (continued)

Bit	Bit Description		Default (hex)	Sync'd to Frame Start	Bad Frame
0x0A (10) Row Speed					
2:	Pixel Clock Speed	A programmed value of *N* gives a pixel clock period of (2 x *N*) master clocks. A value of 0 is treated like (and reads back as) a value of 1.	1	Y	YM
3	Reserved	Read-only.	0	N	N
7:4	Delay Pixel Clock	Number of half-master-clock-cycle increments to delay the rising edge of PIXCLK relative to transitions on FRAME_VALID, LINE_VALID, and DOUT.	1	N	N
8	Invert Pixel Clock	Invert PIXCLK. When clear, FRAME_VALID, LINE_VALID, and DOUT are set up relative to the delayed rising edge of PIXCLK. When set, FRAME_VALID, LINE_VALID, and DOUT are set up relative to the delayed falling edge of PIXCLK.	0	N	N
13	Reserved	Reserved. Do not change from default value.	0		
15:14	Reserved	Reserved. Do not change from default value.	0		
0x0B (11) Extra Delay					
13:0	Extra Delay	Extra blanking inserted between frames. A programmed value of *N* increases the vertical blanking time by *N* pixel clock periods. Can be used to get a more exact frame rate. May affect the integration times of parts of the image when the integration time is less than 1 frame.	0	Y	
0x0C (12) Shutter Delay					
10:0	Shutter Delay	The amount of time from the end of the sampling sequence to the beginning of the pixel reset sequence. This register should normally be set to zero. A non-zero value will only have a visible effect on the image when the integration time (Reg0x09) is small. A programmed value of *N* reduces the integration time by *N* master clock periods (*N* / 2) pixel clock periods). Legal values for this register are shown in "Maximum Shutter Delay" on page 37.	0	Y	N
0x0D (13) Reset					
0	Reset	Setting this bit puts the sensor into reset; the frame being generated will be truncated, and the signal interface will go to an idle state. All internal registers (except for this bit) will go to the default power-up state. Clearing this bit resumes normal operation.	0	N	YM
1	Restart	Setting this bit causes the sensor to truncate the current frame and start resetting the first row. The delay before the first valid frame is read out equals the integration time. This bit is write-1, but always reads back as "0."	0	N	YM
2	Standby	Setting this bit places the sensor in a low-power state. See "Power Saving Modes" on page 42.	0	N	YM
3	Reserved	This read/write bit has no function.	0	N	N
4	Output Disable	Setting this bit puts the signal interface into High-Z. See "Output Enable Control" on page 41.	0	N	N
5	Reserved	Reserved.	0		
6	Drive outputs	By default, asserting STANDBY causes the signal interface to enter High-Z. Setting this bit stops STANDBY from contributing to output-enable control. See "Output Enable Control" on page 41.	0	N	N

Table 6: Register Description (continued)

Bit	Bit Description		Default (hex)	Sync'd to Frame Start	Bad Frame
7	Inhibit Standby	By default, asserting STANDBY places the sensor in a low-power state. Setting this bit stops STANDBY from affecting entry to or exit from the low-power state. See "Power Saving Modes" on page 42.	0	N	N
8	Show Bad Frames	By default, the sensor only shows good frames. When this bit is set, all frames are output.	0	N	Y
9	Restart Bad Frames	When this bit is set, a restart is forced any time a bad frame is detected. This can shorten the delay when waiting for a good frame, since the delay for masking out a bad frame will be the integration time rather than the full frame time.	0	N	N
10	Toggle SADDR	By default, the sensor serial bus will respond to addresses 0xBA/0xBB. When this bit is set, the sensor serial bus will respond to addresses 0x90/0x91. Writes to this bit are ignored when STANDBY is asserted. See "Slave Address" on page 16.	0	N	N
12	Reserved	Reserved.	0	N	
13	Reserved	Reserved.	0	N	
15	Synchronize Changes	By default, the update for many registers is synchronized to frame start. Setting this bit inhibits this update; register changes will remain pending until this bit is returned to "0." When this bit is returned to "0," all pending register updates will be made on the next frame start.	0	N	N
0x20 (32) Read Mode - Context B					
0	Mirror Rows	Rows read out from bottom to top (upside down). When set, row readout starts from row (Row Start + Row Size) and continues down to (Row Start + 1). When clear, readout starts at row start and continues to (Row Start + Row Size - 1). This ensures that the starting color is maintained.	0	Y	YM
1	Mirror Columns	Columns read out from right to left (mirrored). When set, column readout starts from column (Col Start + Col Size) and continues down to (Col Start + 1). When clear, readout starts at col start and continues to (Col Start + Col Size - 1). This ensures that the starting color is maintained.	0	Y	YM
2	Row Skip 2x—Context B	When read mode for context B is selected (Reg0xC8[3] = 1): 1 = read out two rows, skip two rows (i.e. row 8, row 9, row 12, row 13...). 0 = normal readout.	0	Y	YM
3	Column Skip 2x—Context B	When read mode for context B is selected (Reg0xC8[3] = 1): 1 = read out two columns, skip two columns (as with rows). 0 = normal readout.	0	Y	YM
4	Row Skip 4x	1 = read out two rows, skip six rows (i.e. row 8, row 9, row 16, row 17...). 0 = normal readout.	0	Y	YM
5	Column Skip 4x	1 = read out two columns, skip six columns (as with rows). 0 = normal readout.	0	Y	YM

Table 6: Register Description (continued)

Bit	Bit Description		Default (hex)	Sync'd to Frame Start	Bad Frame
7:6	Zoom	In zoom mode, the pixel data rate is slowed by a factor of either two or four, and either 1 or 3 additional blank rows are added between output rows. This is designed to give the controller logic time to repeat data to fill in a window—that is either two or four times larger—with repeated data. The pixel clock speed is not affected by this operation, and the output data for each pixel is valid for either two or four pixel clocks. In zoom 2x mode, every row is followed by a blank row (with its own LINE_VALID, but all data bits = 0) of equal time. In zoom 4x mode, every row is followed by three blank rows. The combination of this register and an appropriate change to the window-sizing registers enables the user to zoom to a region of interest without affecting the frame rate. 00 = no zoom (default). X1 = zoom 2x. 10 = zoom 4x.	0	Y	YM
8	Over-Sized	When this bit is set, a four-pixel border will be output around the active image array, independent of readout mode (skip, zoom, mirror, etc.). Setting this bit adds 8 to the number of rows and columns in the frame.	0	Y	YM
9	Show Border	This bit indicates whether to show the border enabled by bit 8. X0 = normal behavior, no border. 01 = border is enabled but not shown; vertical blanking is increased by 8 rows, horizontal blanking is increased by 8 pixels. 11 = border is enabled and shown; FRAME_VALID time is extended by 8 rows, LINE_VALID is extended by 8 pixels. See "Pixel Border" on page 32.	0	N	Y
10	Use 1 ADC—Context B	The MT9V012 operates with a single ADC, so this read-only bit always reads 1.	1	Y	YM
14	Continuous LINE_VALID	1 = Continuous LINE_VALID (continue producing LINE_VALID during vertical blanking). 0 = Normal LINE_VALID (default, no LINE_VALID during vertical blanking).	0	N	N
15	XOR LINE_VALID	1 = LINE_VALID = continuous LINE_VALID XOR FRAME_VALID. 0 = LINE_VALID determined by bit 9. Ineffective if continuous LINE_VALID is set.	0	N	N
0x21 (33) Read Mode - Context A					
2	Row Skip 2x—Context A	When read mode for context A is selected (Reg0xC8[3] = 0): 1 = read out two rows, skip two rows (i.e. row 8, row 9, row 12, row 13...). 0 = normal readout.	1	Y	YM
3	Column Skip 2x—Context A	When read mode for context A is selected (Reg0xC8[3] = 0): 1 = read out 2 columns, skip 2 columns (as with rows). 0 = normal readout.	1	Y	YM
10	1 ADC Mode—Context A	The MT9V012 operates with a single ADC, so this read-only bit always reads 1.	1	Y	N

Table 6: Register Description (continued)

Bit	Bit Description		Default (hex)	Sync'd to Frame Start	Bad Frame
0x22 (34) Show Control					
2:0	Number Dark Rows	A value of *n* causes (*n* + 1) dark rows to be read out at the beginning of each frame when dark-row readout is enabled (bit 3).	7	N	Y
3	Reserved	This read/write bit has no function.	1	N	Y
6:4	Dark Start Address	The start address for the dark rows. Must be set so all dark-row readouts fall in the address space 7:0.	0	N	N
7	Show Dark and Extra Rows	When set, the programmed dark rows will be output before the active window. In addition, two additional rows will be output before the active window and two additional rows will be output after the active window. FRAME_VALID will thus be asserted earlier than normal. This has no effect on integration time or frame rate.	0	N	N
8	Read Dark Columns	Enables dark column readout, columns 21:2, for use in the row-wise noise correction algorithm. When disabled, an arbitrary number of dark columns can be read out by including them in the active image. Enabling the dark columns increases the minimum value for horizontal blanking but does not affect the row time.	1	N	Y
9	Show Dark Columns	When set, the 20 dark columns will be output before the active pixels in a line. There is an idle period of two pixels between dark column readout and active image readout. Therefore, when set, LINE_VALID will be asserted twenty-two pixel times earlier than normal, and the horizontal blanking time will be decreased by the same amount.	0	N	N
0x23 (35) Flash Control					
7:0	Xenon Count	Length of flash pulse when Xenon flash is enabled. The value specifies the length in units of 1,024 x PIXCLK cycle increments. When the Xenon count is set to its maximum value (0xFF), the flash pulse will automatically be truncated prior to the readout of the first row, giving the longest pulse possible.	8	N	N
8	LED Flash	Enable LED flash. When set, the FLASH output will assert prior to the start of the resetting of a frame and will remain asserted until the end of the frame readout.	0	Y	Y??
9	Every Frame	1 = Flash should be enabled every frame. 0 = Flash should be enabled for 1 frame only.	1	N	N
10	End of Reset	1 = In Xenon mode, the flash is triggered after resetting a frame. 0 = In Xenon mode, the flash is triggered after a frame readout.	1	N	N
12:11	Frame Delay	Flash pulse delay measured in frames.	0	N	N
13	Xenon Flash	Enable Xenon flash. When set, the FLASH output will assert for the programmed period (bits [7:0]) during vertical blanking. This is achieved by keeping the integration time equal to one frame, and the pulse width less than the vertical blanking time.	0	Y	N??
14	Triggered	Indicates that the FLASH output was asserted for the current frame. Read-only.	0		

Table 6: Register Description (continued)

Bit	Bit Description		Default (hex)	Sync'd to Frame Start	Bad Frame
15	Flash	Reflects the current state of the FLASH output. Read-only.	0		
0x24 (36) Extra Reset					
13:0	Reserved	Read-only.	0	N	N
14	Next Row Reset	When set, and the integration time is less than one frame time; row (*n* + 1) is reset immediately prior to resetting row (*n*). This is intended to prevent blooming across rows under conditions of very high illumination.	1	N	N
15	Reserved	Read-only.	0	N	N
0x2B (43) Green1 Gain					
6:0	Initial Gain	Initial gain = bits [6:0] * 0.03125.	20	Y	N
8:7	Analog Gain	Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain (each bit gives 2x gain).	0	Y	N
11:9	Digital Gain	Total gain = (bit [9]+ 1) * (bit [10] + 1) * (bit [11]+ 1) * analog gain (each bit gives 2x gain).	0	Y	N
0x2C (44) Blue Gain					
6:0	Initial Gain	Initial gain = bits [6:0] * 0.03125.	40	Y	N
8:7	Analog Gain	Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain (each bit gives 2x gain).	0	Y	N
11:9	Digital Gain	Total gain = (bit [9] + 1) * (bit [10] + 1) * (bit [11]+ 1) * analog gain (each bit gives 2x gain).	0	Y	N
0x2D (45) Red Gain					
6:0	Initial Gain	Initial gain = bits [6:0] * 0.03125.	20	Y	N
8:7	Analog Gain	Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain (each bit gives 2x gain).	0	Y	N
11:9	Digital Gain	Total gain = (bit [9] + 1) * (bit [10] + 1) * (bit [11]+ 1) * analog gain (each bit gives 2x gain).	0	Y	N
0x2E (46) Green2 Gain					
6:0	Initial Gain	Initial gain = bits [6:0] * 0.03125.	20	Y	N
8:7	Analog Gain	Analog gain = (bit [8] + 1) * (bit [7] + 1) * initial gain (each bit gives 2x gain).	0	Y	N
11:9	Digital Gain	Total gain = (bit [9] + 1) * (bit [10] + 1) * (bit [11]+ 1) * analog gain (each bit gives 2x gain).	0	Y	N
0x2F (47) Global Gain					
11:0	Global Gain	This register can be used to set all four gains at once. When read, it will return the value stored in Reg0x2B.	20	Y	N
0x30 (48) Row Noise					
9:0	Row Noise Constant	Constant, used in the row noise cancellation algorithm. It should be set to the dark level targeted by the black level algorithm, plus the noise expected between the averaged values of dark columns. At default, the constant is set to 42 LSB.	2A	N	Y
10	Enable Correction	1 = Enable row noise cancellation algorithm. When this bit is set, the average value of the dark columns readout will be used as a correction for the whole row. The dark average will be subtracted from each pixel on the row, then a constant will be added (bits 9:0). 0 = normal operation.	1	N	Y

Table 6: Register Description (continued)

Bit	Bit Description		Default (hex)	Sync'd to Frame Start	Bad Frame
11	Use Black Level Average	1 = Use black level frame average from the dark rows in the row noise correction algorithm for low gains. Note that this frame average was taken before the last adjustment of the offset DAC for that frame, so it might be slightly off. 0 = Use the mean of black level programmed threshold in the row noise correction algorithm for low gains.	0	N	Y
14:12	Gain Threshold	When the upper analog gain bits are equal to or larger than this threshold, the dark column average is used in the row noise correction algorithm. Otherwise, the value to subtract is decided by bit 11. This check is done independently for each color, and is a means to turn off the black level algorithm for lower gains, if desired.	0	N	N
15	Frame-Wise Digital Correction	By default, the row noise is calculated and applied individually for each color of each row. When this bit is set, the row noise is calculated and applied for each color of each of the first two rows (2 pairs of values) and the same values are applied to each subsequent row, so that new values are calculated and applied once per frame.	0	N	N
0x5B (91) Green1 Frame Average					
6:0	Green1 Frame Average	The frame averaged green1 black level that is used in the black level calibration algorithm. Read-only.	0	N	N
0x5C (92) Blue Frame Average					
6:0	Blue Frame Average	The frame averaged blue black level that is used in the black level calibration algorithm. Read-only.	0	N	N
0x5D (93) Red Frame Average					
6:0	Red Frame Average	The frame averaged red black level that is used in the black level calibration algorithm. Read-only.	0	N	N
0x5E (94) Green2 Frame Average					
6:0	Green2 Frame Average	The frame averaged green2 black level that is used in the black level calibration algorithm. Read-only.	0	N	N
0x5F (95) Calib Threshold					
6:0	Lower Threshold	Lower threshold for targeted black level in ADC LSBs.	1D	N	N
14:8	Upper Threshold	Upper threshold for targeted black level in ADC LSBs.	23	N	N
0x60 (96) Calibration Control					
0	Manual Override	Manual override of black level correction. 1 = Override automatic black level correction with programmed values. (Reg0x61 - 0x64). 0 = normal operation (default).	0	N	Y
1	Same Green	When this bit is set, the same calibration value will be used for all green pixels: Calib green2 = calib green1.	0	N	Y
2	Same Red/Blue	When this bit is set, the same calibration value will be used for red and blue pixels: calib blue = calib red.	0	N	Y
3	Reserved		0		
4	Step Size Forced to One	When set, the step size will be forced to "1" for the rapid-sweep algorithm. Default operation (0) is to start at a higher step size when in rapid sweep mode, to converge to the correct value faster.	0	N	N

Table 6: Register Description (continued)

Bit	Bit Description		Default (hex)	Sync'd to Frame Start	Bad Frame
7:5	Frames to Average Over	Two to the power of this value decide how many frames to average over when the black level algorithm is in the averaging mode. In this mode, the running frame average will be calculated from the following formula: Running frame average = Old running frame average - (old running frame average)/2^n + (new frame average)/2^n.	4	N	N
8	Sweep Mode	When set, the calibration value will be increased by one every frame, and all channels will be the same. This can be used to get a ramp input to the ADC from the calibration DACs.	0	N	N
12	Recalculate	When set, the rapid sweep mode will be triggered if enabled, and the running frame average will be reset to the current frame average. This bit is write - 1 but always reads back as "0."	0	Y	N
15	Disable Rapid Sweep Mode	Disables the rapid sweep mode in the black level algorithm. The averaging mode will still be enabled.	0	Y	N
0x61 (97) Green1 Calibration Value					
8:0	Green1 Calibration Value	Analog calibration offset for green1 pixels, represented as a two's complement signed 8-bit value (if [8] is clear, the offset is positive and the magnitude is given by [7:0]. If [8] is set, the offset is negative and the magnitude is given by Not ([7:0]) + 1). If Reg0x60[0] = 0, this register is read-only and returns the current value computed by the black level calibration algorithm. If Reg0x60[0] = 1, this register is read/write, and can be used to set the calibration offset manually. Green1 pixels share rows with red pixels.	0	N	Y
0x62 (98) Blue Calibration Value					
8:0	Blue Calibration Value	Analog calibration offset for blue pixels, represented as a two's complement signed 8-bit value (if [8] is clear, the offset is positive and the magnitude is given by [7:0]. If [8] is set, the offset is negative and the magnitude is given by Not ([7:0]) + 1). If Reg0x60[0] = 0, this register is read-only and returns the current value computed by the black level calibration algorithm. If Reg0x60[0] = 1, this register is read/write and can be used to set the calibration offset manually.	0	N	Y
0x63 (99) Red Calibration Value					
8:0	Red Calibration Value	Analog calibration offset for red pixels, represented as a two's complement signed 8-bit value (if [8] is clear, the offset is positive and the magnitude is given by [7:0]. If [8] is set, the offset is negative and the magnitude is given by Not ([7:0]) + 1). If Reg0x60[0] = 0, this register is read-only and returns the current value computed by the black level calibration algorithm. If Reg0x60[0] = 1, this register is read/write and can be used to set the calibration offset manually.	0	N	Y

Table 6: Register Description (continued)

Bit	Bit Description		Default (hex)	Sync'd to Frame Start	Bad Frame
0x64 (100) Green2 Calibration Value					
8:0	Green2 Calibration Value	Analog calibration offset for green2 pixels, represented as a two's complement signed 8-bit value (if [8] is clear, the offset is positive and the magnitude is given by [7:0]. If [8] is set, the offset is negative and the magnitude is given by Not ([7:0]) + 1). If Reg0x60[0] = 0, this register is read-only and returns the current value computed by the black level calibration algorithm. If Reg0x60[0] = 1, this register is read/write and can be used to set the calibration offset manually. Green2 pixels share rows with blue pixels.	0	N	Y
0xC8 (200) Context Control					
0	Horizontal Blanking Select	1 = Use horizontal blanking context B, Reg0x05. 0 = Use horizontal blanking context A, Reg0x07.	1	Y	YM
1	Vertical Blanking Select	1 = Use vertical blanking context B, Reg0x06. 0 = Use vertical blanking context A, Reg0x08.	1	Y	YM
2	LED Flash Enable	Enable LED flash. Same physical register as Reg0x23[8].	0	Y	N
3	Read Mode Select	1 = Use read mode context B, Reg0x20. 0 = Use read mode context A, Reg0x21. Note that bits only found in read mode, context B register, will always be taken from that register.	1	Y	YM
7	Xenon Flash Enable	Enable Xenon flash. Same physical register as Reg0x23[13].	0	Y	N
15	Restart	Setting this bit will cause the sensor to abandon the current frame and start resetting the first row. Same physical register as Reg0x0D[1].	0	N	YM

Feature Description

Window Control

Window Start

The row and column start address of the displayed image can be set by Reg0x01 (row start) and Reg0x02 (column start).

Window Size

The size of the displayed image can be set by Reg0x03 (row width) and Reg0x04 (column width). The default image size is 640 columns and 480 rows (VGA).

Pixel Border

When Reg0x20[9:8] are both set, a four-pixel border will be added around the specified image. This border can be used as extra pixels for image processing algorithms. The border is independent of the readout mode, which means that even in skip modes, a four-pixel border will be output in the image. When enabled, the row and column widths will be eight larger than the values programmed in Reg0x03 and Reg0x04. If the border is enabled but not shown in the image (Rex0x20[9:8] = 01), the horizontal blanking and vertical blanking values will be eight larger than the values programmed into the blanking registers.

Context Switching

Reg0xC8 is designed to enable easy switching between sensor modes. Some key registers and bits in the sensor have two physical register locations, called contexts. Bits 0, 1, and 3 of Reg0xC8 control which context register context is currently in use. A "1" in a bit will select context B, while a "0" will select context A for this parameter. The select bits can be used in any combination, but by default are set up to make switching between preview mode and full-resolution mode easy:

Context B (Default Context)

Reg0xC8	= 0x000B	(Context B)
Reg0x05	= 0x00F4	(Horizontal blanking, context B)
Reg0x06	= 0x001D	(Vertical blanking, context B)
Reg0x20	= 0x0400	(1 ADC, no column or row skip)

Description: Full-resolution VGA image at 30 fps

Context A (Alternate Context)

Reg0xC8	= 0x0000	(Context A)
Reg0x07	= 0x0234	(Horizontal blanking, context A)
Reg0x08	= 0x010D	(Vertical blanking, context A)
Reg0x21	= 0x040C	(1 ADC, column and row skip)

Description: Half-resolution QVGA image at 30 fps

The horizontal blanking and vertical blanking values for the two contexts are chosen so that row time will be preserved between contexts. This ensures that changing contexts does not affect integration time. A few more control bits are also available through the context register (Reg0xC8) so that flash and restarting the sensor can be done simultaneously with changing contexts. See Table 6, Register Description, on page 23 for more information.

Readout Modes

Column Mirror Image

By setting Reg0x20[1] = 1, the readout order of the columns will be reversed, as shown in Figure 16. The starting color is preserved when mirroring the columns.

Row Mirror Image

By setting Reg0x20[0] = 1, the readout order of the rows will be reversed as shown in Figure 17. The starting color is preserved when mirroring the rows.

Figure 16: Six Pixels in Normal and Column Mirror Readout Modes

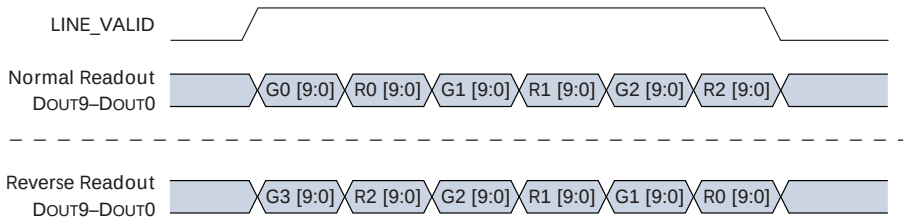
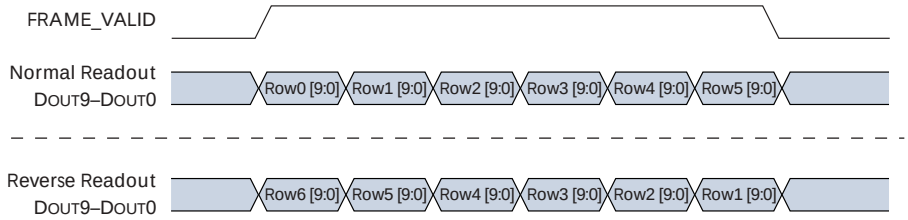


Figure 17: Six Rows in Normal and Row Mirror Readout Modes



Column and Row Skip

By setting Reg0x20[2] = 1 (Reg0x21 in context A), only half of the columns set will be read out. An example is shown in Figure 18. Only columns with bit 1 equal to “0” will be read out (xxxxxxx0x). The row skip works in the same way and will only read out rows with bit 1 equal to “0.” Row skip mode is enabled by setting Reg0x20[4]. For both row and column skips, the number of rows or columns read out will be half of what is set in Reg0x03 or Reg0x04, respectively.

The sensor can also be programmed to only read out a quarter of the specified window size by setting Reg0x20[5:4], as shown in Figure 19. In all cases, the row and column sequencing ensures that the Bayer pattern is preserved.

Figure 18: Eight Pixels in Normal and Column Skip 2X Readout Modes

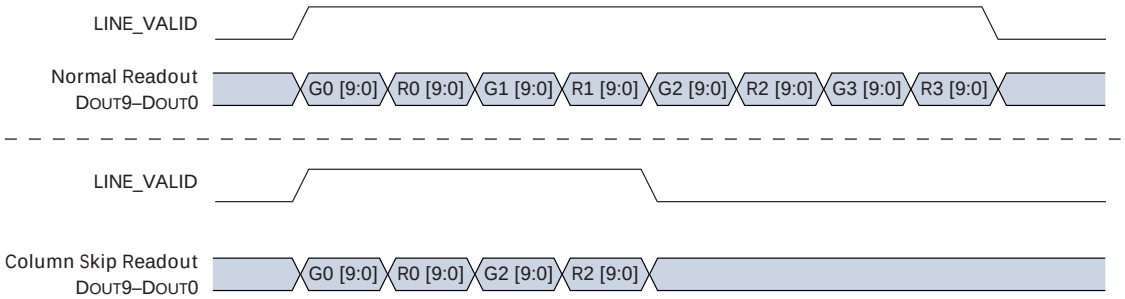
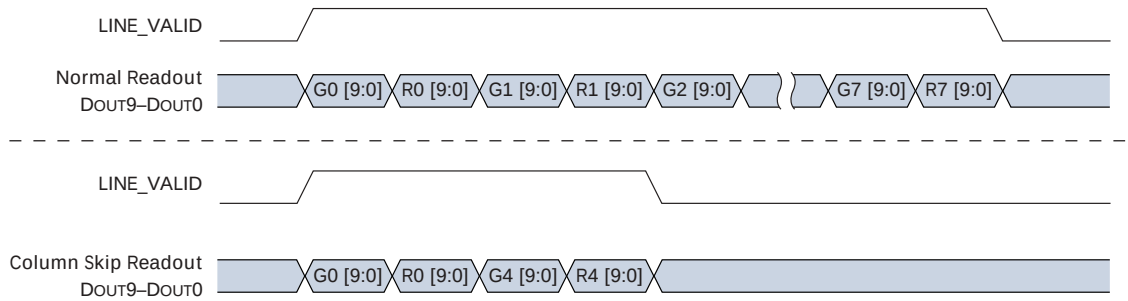


Figure 19: Sixteen Pixels in Normal and Column Skip 4X Readout Modes

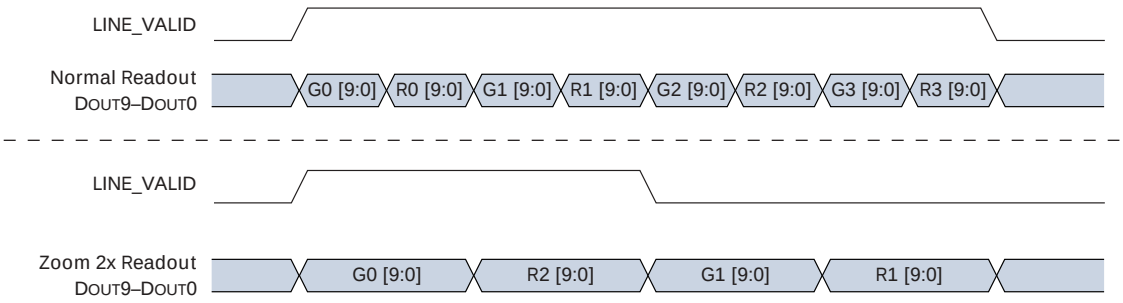


Digital Zoom

Reg0x20[7:6] allows a digital zoom of 2x or 4x to be applied. In zoom mode, the pixel data rate is slowed by a factor of either two or four, and either one or three additional blank rows are added between output rows. This is designed to give the controller logic time to repeat data to fill in a window that is either two or four times larger with repeated data.

The pixel clock speed is not affected by this operation, and therefore the output data for each pixel is valid for either two or four pixel clocks. In 2x zoom mode, every row is followed by a blank row (with its own LINE_VALID, but all data bits = 0) of equal time. In 4s zoom mode, every row is followed by three blank rows. In the zoom modes, Reg0x03 and Reg0x04 will still specify the window size out of the sensor including the extra blanking, so the active image read out will, in effect, be half or one-quarter of the output image.

Figure 20: Eight Pixels in Normal and Zoom 2X Readout Modes



Frame Rate Control

For a given window size, the blanking registers (Reg0x05-Reg0x08), along with the row speed register (Reg0x0A), can be used to set a particular frame rate.

The frame timing equations (Table 3 and Table 4 on page 12) can be rearranged to express the horizontal blanking or vertical blanking values as a function of the frame rate:

HBLANK_REG = master clock freq / (frame rate * ((Reg0x03/S) + VBLANK_REG) * PIXCLK_PERIOD) - (Reg0x04/S)
VBLANK_REG = master clock freq / (frame rate * ((Reg0x04/S) + HBLANK_REG) * PIXCLK_PERIOD) - (Reg0x03/S)

The HBLANK_REG value allows the frame rate to be adjusted with a minimum resolution of one PIXCLK_PERIOD multiplied by the total number of rows (displayed plus blanking). If finer resolution is required, Reg0x0b (extra delay) can be used. Reg0x0b allows the frame time to be changed in increments of 2 x (master clock period).

Minimum Horizontal Blanking

The minimum horizontal blanking value is constrained by the time used for sampling a row of pixels and the overhead in the row readout. This can be expressed in an equation as:

HBLANK(MIN) = startup overhead + sampling time + extra cb time + dark col time
= 31+ done_sample/2 + 16 + 22 * read_dark_cols
= 47 + done_sample/2 + 22 * read_dark_cols

where:

done_sample = Reg0x7E rounded up to nearest even number
read_dark_cols = Reg0x22, bit 8

with default settings:

HBLANK(MIN) = 47+ (126)/2 + 22
= 132 PIXCLK periods

To get an aggressive minimum value for the horizontal blanking, the larger of Reg0x79[15:8] and Reg0x76[15:8] can be substituted for the Reg0x7E value in the above equation. With default settings, this gives a minimum HBLANK time of 127.

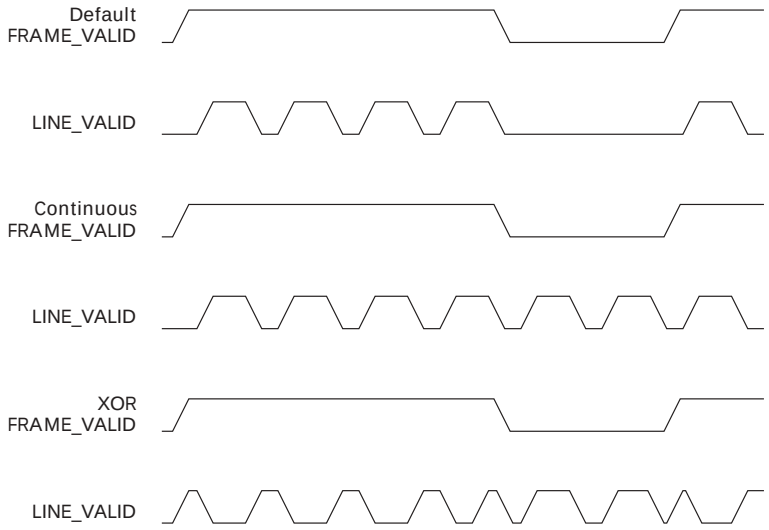
Valid Data Signals Options

LINE_VALID Signal

By setting bits 9 and 10 of Reg0x20, the LINE_VALID signal is programmed for three different output formats. The formats shown below illustrate reading out 4 rows and 2 vertical blanking rows (Figure 21).

In the last format, the LINE_VALID signal is the XOR between the continuous LINE_VALID signal and the FRAME_VALID signal.

Figure 21: LINE_VALID Formats



Integration Time

Integration time is controlled by Reg0x09 (shutter width, in multiples of the row time) and Reg0x0C (shutter delay, in PIXCLK_PERIOD/2). Reg0x0C is used to control sub-row integration times and will only have a visible effect for small values of Reg0x09. The total integration time, ^{t}INT, is shown in the equation below:

^{t}INT = Reg0x09 * Row Time - Integration Overhead - Shutter Delay

where:

Row Time = (Reg0x04 + HBLANK_REG) * PIXCLK_PERIOD master clock periods from Table 3 on page 12

Overhead Time = 182 master clock periods

Shutter Delay = Reg0x0C/2 * PIXCLK_PERIOD master clock periods

with default settings:

^{t}INT = (509 * 884 * 2) - 182 - 0
= 899,730 master clock periods

In this equation, the integration overhead corresponds to the delay between the row reset sequence and the row sample (read) sequence.

Typically, the value of Reg0x09 is limited to the number of rows per frame (which includes vertical blanking rows), such that the frame rate is not affected by the integration time. If Reg0x09 is increased beyond the total number of rows per frame, the sensor will add additional blanking rows as needed. A second constraint is that ^{t}INT must be

adjusted to avoid banding in the image caused by light flicker. This means that ^{t}INT must be a multiple of 1/120 of a second under 60Hz flicker, and a multiple of 1/100 of a second under 50Hz flicker.

Maximum Shutter Delay

The shutter delay can be used to reduce the integration time. A programmed value of *N* reduces the integration time by *N* master clock periods. The maximum shutter delay is set by the row time and the sample time, as shown in the equation below:

max shutter delay = Row Time - Shutter Overhead

where:

Row Time = (Reg0x04 + HBLANK_REG) * PIXCLK_PERIOD from Table 3 on page 12

Shutter Overhead = 331 master clock periods

with default settings:

max shutter delay = (884 * 2) - 331
= 1437 master clock periods

If the value in this register exceeds the maximum value given by this equation, the sensor may not generate an image.

Flash Strobe

The MT9V012 supports both Xenon and LED flash through the flash output ball. The timing of the flash ball with the default settings is shown in Figures 22, 23, and 24. Reg0x23 allows the timing of the flash to be changed. The flash can be programmed to fire only once; be delayed by a few frames when asserted; and (for Xenon flash) the flash duration can be programmed.

Enabling the LED flash will cause one bad frame, where several of the rows only have the flash on for part of their integration time. This can be avoided by forcing a restart (write Reg0x0D[1] = 1) immediately after enabling the flash; the first bad frame will then be masked out as shown in Figure 24. Read-only bit Reg0x23[14] is set during frames that are correctly integrated; the state of this bit is shown below.

Figure 22: Xenon Flash Enabled

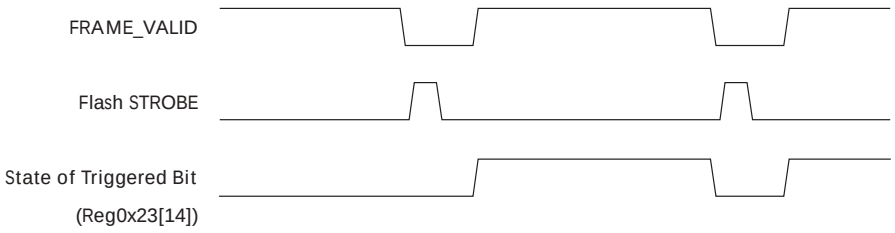
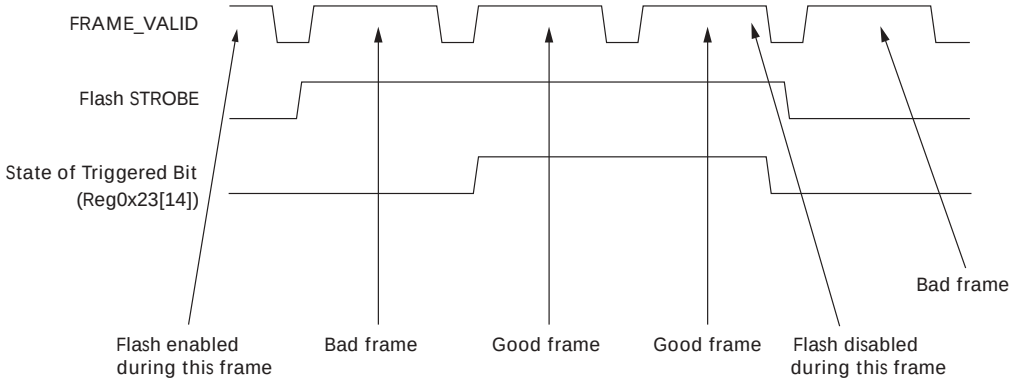
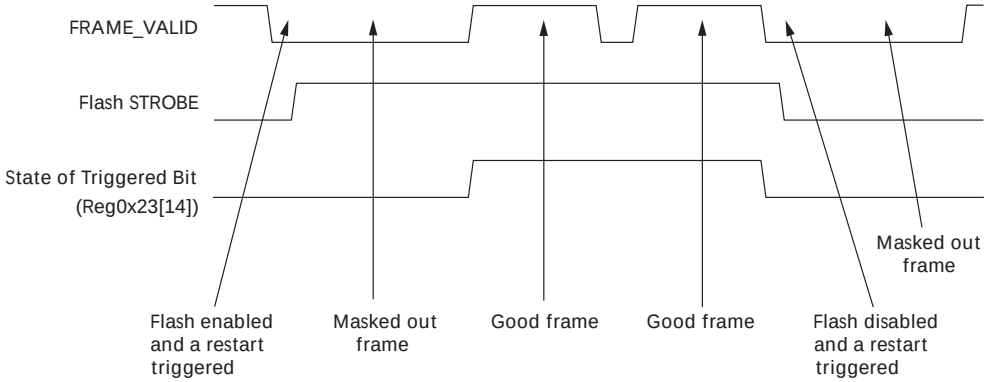


Figure 23: LED Flash Enabled



Note: Integration time = number of rows in a frame.

Figure 24: LED Flash Enabled Following Forced Restart

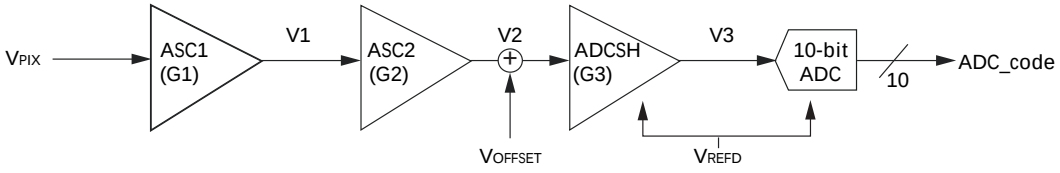


Note: Integration time = number of rows in a frame.

Analog Signal Path

A block diagram of the analog signal path is shown in Figure 25. The analog signal path consists of two gain stages (ASC1, ASC2), a sample-and-hold (ADCSH) stage with black level calibration capability (V_{OFFSET}), and a 10-bit ADC.

Figure 25: Analog Signal Path



Stage-by-Stage Transfer Functions

Transfer functions proceed, stage by stage, as described below:

Let VPIX be the input of the signal path:	VPIX = pixel output voltage = signal path input voltage,	
The output voltage of ASC 1st stage is:	V1 = -1 * G1 *VPIX	(1)
The output voltage of ASC 2nd stage is:	V2 = -1 * G2 * V1	(2)
The output voltage of ADC Sample-and-Hold stage is:	V3 = 2 * G3 * V2 - VREFD + VOFFSET	(3)
and the ADC output code is:	ADC output code = 511 * (1 + (V3 / VrEFD))	(4)
From (1) to (4), the ADC output code can also be written as:	ADC code = (1022/VREFD) * [G1 * G2 * G3 * VPIX + (Voffset/(2 * G3))]	(5)

Where G1, G2, and G3 are the gain settings, VOFFSET is the offset (calibration) voltage, and VREFD is the reference voltage of the ADC. The gain setting G3 is applied to the signal but is not applied to VOFFSET. The parameters VREFD, G1, G2, G3, and VOFFSET are described below.

VREFD

The VREFD parameters are as follows:

The ADC reference voltage VREFD IS:	VREFD = VREF_HI - VREF_LO	(6)
where	VREF_HI = 55.56mV * (Reg0x41[7:4] + 23)	(7)
using default register values:	VREF_HI = 55.56mV * (13 + 23) = 2.000V	
and	VREF_LO = 55.56mV * (Reg0x41[3:0] + 11)	(8)
using default register values:	VREF_LO = 55.56mV * (7 +11) = 1.000V	
so	VREFD = 55.56mV * (Reg0x41[7:4] - Reg0x41[3:0] + 12)	(9)
using default register values	VREFD = 2.000 - 1.000 = 1.000V	

Gain Settings: G1, G2, G3

The gains for green1, blue, red, and green2 pixels are set by registers Reg0x2B, Reg0x2C, Reg0x2D, and Reg0x2E, respectively. Gain can also be set globally by Reg0x2F. The analog gain is set by bits [8:0] of the corresponding register as follows:

G1 = bit[7] + 1	(10)
G2 = bit[6:0] / 32	(11)
G3 = bit[8] + 1	(12)

Digital gain is set by bits [11:9] of the same registers.

Offset Voltage: VOFFSET

The offset voltage provides a constant offset to the ADC to fully utilize the ADC input dynamic range. The offset voltages for green1, blue, red, and green2 pixels are manually set by registers Reg0x61, Reg0x62, Reg0x63, and Reg0x64, respectively. Note that the offset voltages can also be set automatically by the black level calibration loop.

For a given color, the offset voltage, VOFFSET, is determined by:

	VOFFSET = 0.250V * offset_gain * offset_sign * offset_code[7:0]/255	(13)
where:	"offset_sign" is determined by bit[8] as:	
	if bit[8] = 0, offset_sign = +1	(14)
	if bit[8] = 1, offset_sign = -1	(15)
	"offset_code" is the decimal value of bit[7:0]	

"Offset_gain" is determined by the 2-bit code from Reg0x5A[1:0], as shown in Table 7. These step sizes are not exact; increasing the stage0 ADC gain from two to four will decrease the step size significance; decreasing the ADC VREFD will increase the step size significance.

Table 7: Offset Gain

Reg0x5A[1:0]	offset_gain
00	No calibration voltage is applied.
01	1 calibration LSB is equal to 0.5 ADC LSB.
10	1 calibration LSB is equal to 1.0 ADC LSB.
11	1 calibration LSB is equal to 2.0 ADC LSB.

Recommended Gain Settings

The analog gain circuitry in the MT9V012 provides signal gains from 1 through 15.875. The minimum gain of 1 (gain registers set to 0x20) corresponds to the setting where the pixel signal is guaranteed to saturate the ADC under all specified operating conditions. Any reduction of the gain below this value may cause the sensor to saturate, under certain conditions, at ADC output values less than the maximum. It is recommended that this guideline be followed at all times.

Since bits [8:7] of the gain registers are multiplicative factors for the gain settings, there are alternative ways of achieving certain gains. Some settings offer superior noise performance to others, despite providing the same overall gain. Table 8 lists the recommended gain settings.

Table 8: Recommended Gain Settings

Desired Gain	Recommended Gain Register Setting
1–1.969	0x020–0x03F
2–7.938	0x0A0–0x0FF
8–15.875	0x1C0–0x1FF

Black Level Calibration

The MT9V012 performs automatic on-chip black level calibration. The calibration algorithm operates on each of the four colors of the pixel array independently. At the start of each frame, four dark rows are read (two rows of each color pair) and the average value of a pixel color is calculated. The pixels are averaged as if they were light-sensitive and passed through the appropriate color gain.

Each average is digitally low-pass filtered over many frames to remove temporal noise and random instabilities associated with the measurement.

For each color, the filtered average is compared to a minimum acceptable level—low threshold, and a maximum acceptable level—high threshold. If the average is lower than the minimum acceptable level, the offset correction voltage for that color is increased by one offset LSB (at default, 1 LSB offset = 1mV). If it is above the maximum level, the offset correction voltage is decreased by 1 LSB (1mV). To avoid oscillation of the black level from below to above the region, the thresholds should be programmed so the difference is at least two times the offset DAC step size.

Whenever the gain is changed, or if the black level recalculation bit, reset bit, or restart bit is set, a rapid sweep algorithm will be triggered and the running digitally filtered average is reset to the first frame average of the dark pixels. Digital filtering over many frames is then restarted.

After changes to the sensor configuration, large shifts in the black level calibration can result. To quickly adapt to this shift, a rapid sweep of the black level during the dark-row readout is performed on the first frame after certain changes to the sensor registers. Any changes to the registers noted above will cause this recalculation. The data from this sweep allows the sensor to choose an accurate new starting point for the running average. In the rapid sweep mode, fewer pixels of each color are averaged and used to set the new applied offset. Rapid sweep mode allows several offset values to be tried during the black row readout. The rapid sweep mode can be disabled or triggered manually using control bits in Reg0x60.

The current black level calibration values can be read back from Reg0x61, Reg0x62, Reg0x63, and Reg0x64. Each value is a 9-bit signed value for one of the four colors of the Bayer pattern. In normal operation, these values are calculated at the beginning of each frame. However, if Reg0x60[0] is set to "1," these registers can be written to, overriding the automatic black level calculation. This feature can be used in conjunction with readout of the black rows (Reg0x22[7]) if the user would like to use an external black level calibration circuit.

The formula for the offset correction voltage is shown in "Offset Voltage: VOFFSET" on page 39.

Row-Wise Noise Cancellation

In addition to the automatic black level calibration applied to the whole frame, a row-wise noise cancellation algorithm is applied digitally to the ADC output value.

Pixel value = ADC value - dark column average + Reg0x30[9:0]

Digital Signal Path

Data from the ADC is processed digitally to generate the pixel output data.

Output Enable Control

When the sensor is configured to operate in default mode, the DOUT, FRAME_VALID, LINE_VALID, PIXCLK, and FLASH outputs can be placed in High-Z under hardware or software control, as shown in Table 9.

Table 9: Output Enable Control

Standby	Reg0x0D[4] (output_dis)	Reg0x0D[6] (drive_SIGNALS)	Output State
0	0 (default)	0 (default)	driven
1	0 (default)	0 (default)	High-Z
0	1	0 (default)	High-Z
1	1	0 (default)	High-Z
"Don't Care"	0 (default)	1	driven
"Don't Care"	1	1	High-Z

In all cases, the transition between driven and high impedance states occurs asynchronously. Output enable control is provided as a mechanism to allow multiple sensors to share a single set of interface signals with a host controller.

When the sensor is configured to operate in serial mode, output enable control is not available: The DOUT, FRAME_VALID, LINE_VALID and PIXCLK signals are always placed in High-Z, and the FLASH signal is always driven.

There is no benefit in placing the signals in High-Z while the sensor is in its low-power standby state. Therefore, in single-sensor applications that use STANDBY to enter and leave the standby state, programming Reg0x0D[6] = 1 is recommended.

Power Saving Modes

The sensor can be put into a low power standby state by either of these mechanisms:

- Asserting STANDBY (provided that Reg0x0D[7] = 0)
- Setting Reg0x0D[2] = 1 by performing a register write through the serial register interface

The two methods are equivalent and have the same effect:

- The source of standby is synchronized and latched. Once latched, the full standby sequence is completed even if the source of standby is removed
- The readout of the current row is completed
- Internal clocks are gated off
- The analog signal chain and associated current and voltage sources are placed in a low-power state

The standby state is maintained for as long as the standby source remains asserted. The state of the signal interface while in standby state is shown in Table 10.

Table 10: Signal State During Standby

Signal	State
LINE_VALID	0
FRAME_VALID	0
LINE_VALID	0
PIXCLK	1
FLASH	0
DOUT(9:0)	0

Output enable control can be used to place the signal interface in High-Z (“Output Enable Control” on page 41).

While in standby, the state of the internal registers is maintained and the sensor continues to respond to accesses through its serial register interface. An even lower power standby state can be achieved by stopping the input clock (CLKIN) while in standby. If the input clock is stopped, the sensor will not respond to accesses through its serial register interface.

Exit from standby must be through the same mechanism as entry to standby. When the standby source is negated:

1. The internal clocks are restarted
2. The analog circuitry is restored to its normal operating state
3. The timing and control circuitry performs a restart, equivalent to writing Reg0x0D[1] = 1

After this sequence has completed, normal operation is resumed. If the input clock has been stopped during standby it must be restarted before leaving standby.

When the sensor is configured in serial mode, the LVDS output remains driven when the sensor is in the standby state.

Floating Inputs

Many MT9V012 signals use bidirectional pads. There are three reasons for this:

- The signal associated with the pad is bidirectional in normal use (the only signal in this category is SDATA)
- The pad is used for output signals in some modes and input signals in other modes (e.g., LINE_VALID)

- The pad is normally used as an output, but is used as an input during manufacturing test modes (e.g., DOUT(9:0)

Standard design practice dictates that signal inputs should not be allowed to float for long periods of time. This leads to two areas where the design application should be reviewed:

1. When using the output enable control (“Output Enable Control” on page 41), ensure that all MT9V012 bidirectional pads that enter a high- impedance state are driven to a valid logic level.
2. When operating the MT9V012 in serial mode ensure that unused bidirectional pads are terminated correctly.

Some of the MT9V012 pads incorporate internal pull-up or pull-down resistors so that they will achieve valid logic levels when left unconnected (See Table 11). These on-chip resistors are only designed to pull an unconnected pad to a valid logic level and should not be relied upon to provide a pull-up or pull-down to any external node.

Table 11: On-chip Pull-ups/Pull-downs

Signal	Behavior
MODE1	On-chip pull-down resistor is enabled permanently.
MODE0	On-chip pull-up resistor is enabled permanently.
DOUT(9:0)	On-chip pull-down resistor is enabled when the sensor is configured to operate in serial mode.
LINE_VALID	On-chip pull-down resistor is enabled when the sensor is configured to operate in serial mode.
FLASH	On-chip pull-down resistor is enabled permanently.
PIXCLK	On-chip pull-down resistor is enabled permanently.

Dark Row/Column Display

Optically black rows from 503 through 500 and optically black columns from 694 through 692 are unused and are not normally visible in the displayed image. They can be included in the displayed image by adjusting the window start and/or window size registers (“Window Control” on page 32).

Optically black rows from 7 through 0 are used to provide data for black level calibration (“Black Level Calibration” on page 40) and are not normally visible in the displayed image. Setting Reg0x22[7] = 1 makes these rows visible in the displayed image. This is achieved by asserting FRAME_VALID earlier than normal, and keeping it asserted longer, so that the following rows are displayed:

- The optically black rows at the start of the pixel array (controlled by Reg0x22[2:0])
- Two rows before the visible rows
- The visible rows (controlled by Reg0x01, Reg0x03, and Reg0x20)
- Two rows after the visible rows

The effect of setting Reg0x22[7] = 1 is that the image is bigger (has more rows) than is programmed by Reg0x03.

Optically-black columns from 23 through 0 are used to provide data for row-wise noise cancellations (“Row-Wise Noise Cancellation” on page 41), and are not normally visible in the displayed image. There are two methods for making them visible in the displayed image:

1. Set Reg0x22[8] = 0 (to disable readout of the dark columns); set Reg0x30[10] = 0 (to disable row-wise correction); then adjust Reg0x02

2. Set Reg0x22[9] = 1

When Reg0x22[9] = 1, LINE_VALID is asserted 22 pixel clocks earlier than normal. Data from columns from 21 through 2 (20 columns) is followed by two pixel clocks of undefined data, then by data from the visible columns (controlled by Reg0x02, Reg0x04, and Reg0x20).

Clock Control

The MT9V012 uses an aggressive clock-gating methodology to reduce power consumption: the clocked logic is divided into a number of separate domains, each of which is only clocked as required. Reg0x65 can be used to bypass the clock gating, so that clocks to individual domains run continuously.

When the MT9V012 enters a low-power state, almost all of the internal clocks are stopped. The only exception is that a small amount of logic (approximately 10 flip-flops) is clocked so that accesses to the two-wire serial interface continue to function correctly. See "Power Saving Modes" on page 42 for further details.

Electrical Specifications

Table 12: DC Electrical Characteristics: 2.50V–3.10V

V_{DD} = 2.50V–3.10V; V_{AA} = VAAPIX = 2.50V–3.10V; T_A = Ambient = 25°C

Symbol	Definition	Condition	Min	Typ	Max	Units
V_{IH}	Input High Voltage		V_{PWR} - 0.3		V_{PWR} + 0.3	V
V_{IL}	Input Low Voltage		-0.3		0.8	V
I_{IN}	Input Leakage Current	No Pull-up Resistor; V_{IN} = V_{DD} or DGND	-15		15	µA
V_{OH}	Output High Voltage		V_{PWR}-0.2			V
V_{OL}	Output Low Voltage				0.2	V
I_{OZ}	Tri-state Output Leakage Current				15	µA
I_{PWR}A	Analog Quiescent Supply Current	Default settings	TBD	25.76		mA
I_{PWR}P	Pixel Array Quiescent Supply Current	Default settings	TBD	1.39		mA
I_{PWR}IOD	Digital and I/O Quiescent Supply Current, Default Mode	CLKIN = 27 MHz; default settings	TBD	2.5		mA
I_{PWRS}IOD	Digital and I/O Quiescent Supply Current, Serial Mode	CLKIN = 27 MHz; default settings	TBD	2.5		mA
I_{PWR}A Standby	Analog Standby Supply Current	STANDBY = V_{DD}[1]	0.0	TBD	TBD	µA
I_{PWR}P Standby	Pixel Array Standby Supply Current	STANDBY = V_{DD}[1]	0.0			µA
I_{PWR}IOD Standby	Digital and I/O Standby Supply Current, Default Mode	STANDBY = V_{DD}[1], CLKIN = 0 MHz	0.0	TBD	TBD	µA
I_{PWRS}IOD Standby	Digital and I/O Standby Supply Current, Serial Mode	STANDBY = V_{DD}[1], CLKIN = 0 MHz	0.0	TBD	TBD	µA
I_{PWR}IOD Standby ClkOn	Digital and I/O Standby Supply Current with Clock On, Default Mode	STANDBY = V_{DD}, CLKIN = 27 MHz	0.0	TBD	TBD	µA
I_{PWRS}IOD Standby ClkOn	Digital and I/O Standby Supply Current with Clock On, Serial Mode	STANDBY = V_{DD}, CLKIN = 27 MHz	0.0	TBD	TBD	µA

Notes: 1. To place the chip in standby mode, first raise STANDBY to V_{DD}, wait until FRAME_VALID and LINE_VALID are de-asserted, then wait two master clock cycles before turning off the master clock.

Table 13: DC Electrical Characteristics: 1.70V–1.90V

V_{DD} = 1.70V–1.90V; V_{AA} = VAAPIX = 2.50V - 3.10V; T_A = Ambient = 25°C

Symbol	Definition	Condition	Min	Typ	Max	Units
V_{IH}	Input High Voltage		V_{PWR} - 0.3		V_{PWR} + 0.3	V
V_{IL}	Input Low Voltage		-0.3		0.8	V
I_{IN}	Input Leakage Current	No Pull-up Resistor; V_{IN} = V_{DD} or DGND	-15		15	µA
V_{OH}	Output High Voltage		V_{PWR} - 0.2			V
V_{OL}	Output Low Voltage				0.2	V
I_{OZ}	Tri-state Output Leakage Current				15	µA
I_{PWR}A	Analog Quiescent Supply Current	Default settings	TBD	25.76		mA
I_{PWR}P	Pixel Array Quiescent Supply Current	Default settings	TBD	1.39		mA
I_{PWR}IOD	Digital and I/O Quiescent Supply Current, Default Mode	CLKIN = 27 MHz; default settings	TBD	2.5		mA
I_{PWR}A Standby	Analog Standby Supply Current	STANDBY = V_{DD}[1]	0.0	TBD	TBD	µA
I_{PWR}P Standby	Pixel Array Standby Supply Current	STANDBY = V_{DD}[1]	0.0			µA
I_{PWR}IOD Standby	Digital and I/O Standby Supply Current, Default Mode	STANDBY = V_{DD}[1], CLKIN = 0 MHz	0.0	TBD	TBD	µA
I_{PWR}IOD Standby ClkOn	Digital and I/O Standby Supply Current with Clock On, Default Mode	STANDBY = V_{DD}, CLKIN = 27 MHz	0.0	TBD	TBD	µA

Notes: 1. To place the chip in standby mode, first raise STANDBY to V_{DD}, wait until FRAME_VALID and LINE_VALID are de-asserted, then wait 2 master clock cycles before turning off the master clock.

Table 14: AC Electrical Characteristics: 2.50V–3.10V

V_{PWR} = 2.50V–3.10V; T_A = Ambient = 25°C

Symbol	Definition	Condition	Min	Typ	Max	Units
^{f}CLKIN	Input Clock Frequency Clock		1		27	MHz
	Duty Cycle					MIN/MAX
^{t}R	Input Clock Rise Time			TBD		ns
^{t}F	Input Clock Fall Time			TBD		ns
^{t}PLHP	CLKIN to PIXCLK propagation delay LOW-to-HIGH	CLOAD = 10pF				ns
^{t}PHLP	CLKIN to PIXCLK propagation delay HIGH-to-LOW	CLOAD = 10pF				ns
^{t}PLHD	CLKIN to DOUT(9:0)propagation delay LOW-to-HIGH	CLOAD = 10pF				ns
^{t}PHLD	CLKIN to DOUT(9:0) propagation delay HIGH-to-LOW	CLOAD = 10pF				ns
^{t}OH	Data Hold Time					ns
^{t}PLHF,L	CLKIN to FRAME_VALID and LINE_VALID propagation LOW-to-HIGH	CLOAD = 10pF		TBD		ns
^{t}PHLF,L	CLKIN to FRAME_VALID and LINE_VALID propagation HIGH-to-LOW			TBD		ns
^{t}ETSU	VRR_N, ESR_N, HPA_N input setup to rising edge of CLKIN			TBD		ns
^{t}ETH	VRR_N, ESR_N, HPA_N input hold from rising edge of CLKIN			TBD		ns
^{t}PHLF	CLKIN to flash propagation delay LOW-to-HIGH			TBD		ns
^{t}PHLF	CLKIN to flash propagation delay HIGH-to-LOW			TBD		ns
^{t}OED	STANDBY LOW to FRAME_VALID, LINE_VALID, PIXCLK, DOUT(9:0), flash driven	TBD		TBD		ns
^{t}OEZ	STANDBY HIGH to FRAME_VALID, LINE_VALID, PIXCLK, DOUT(9:0), flash floating	TBD		TBD		ns

Table 15: AC Electrical Characteristics: 1.70V–1.90V

V_{PWR} = 1.70V–1.90V; T_A = Ambient = 25°C

Symbol	Definition	Condition	Min	Typ	Max	Units
^{f}CLKIN	Input Clock Frequency Clock		1		27	MHz
	Duty Cycle					MIN/MAX
^{t}R	Input Clock Rise Time			TBD		ns
^{t}F	Input Clock Fall Time			TBD		ns
^{t}PLHP	CLKIN to PIXCLK propagation delay LOW-to-HIGH	CLOAD = 10pF				ns
^{t}PHLP	CLKIN to PIXCLK propagation delay HIGH-to-LOW	CLOAD = 10pF				ns
^{t}PLHD	CLKIN to DOUT(9:0)propagation delay LOW-to-HIGH	CLOAD = 10pF				ns
^{t}PHLD	CLKIN to DOUT(9:0) propagation delay HIGH-to-LOW	CLOAD = 10pF				ns
^{t}OH	Data Hold Time from rising edge of CLKIN					ns
^{t}PLHF,L	CLKIN to FRAME_VALID and LINE_VALID propagation LOW-to-HIGH	CLOAD = 10pF		TBD		ns
^{t}PHLF,L	CLKIN to FRAME_VALID and LINE_VALID propagation HIGH-to-LOW			TBD		ns
^{t}ETSU	VRR_N, ESR_N, HPA_N input setup to rising edge of CLKIN			TBD		ns
^{t}ETH	VRR_N, ESR_N, HPA_N input hold from rising edge of CLKIN			TBD		ns
^{t}PHLF	CLKIN to flash propagation delay LOW-to-HIGH			TBD		ns
^{t}PHLF	CLKIN to flash propagation delay HIGH-to-LOW			TBD		ns
^{t}OED	STANDBY LOW to FRAME_VALID, LINE_VALID, PIXCLK, DOUT(9:0), flash driven	TBD		TBD		ns
^{t}OEZ	STANDBY HIGH to FRAME_VALID, LINE_VALID, PIXCLK, DOUT(9:0), flash floating	TBD		TBD		ns

Propagation Delay for FRAME_VALID and LINE_VALID

The LINE_VALID and FRAME_VALID signals change on the rising edge of the master input clock, as shown in Figure 26.

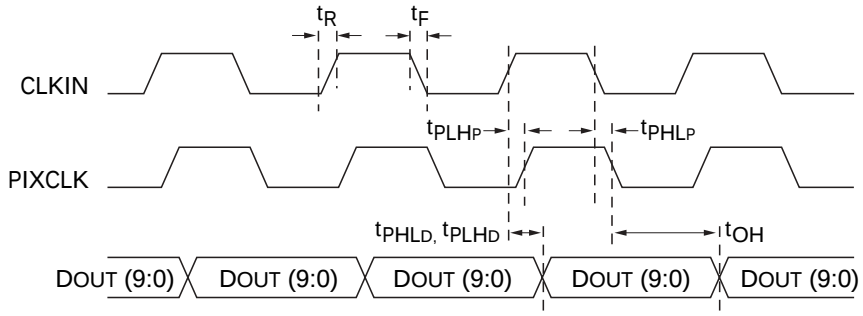
Figure 26: Propagation Delay for FRAME_VALID and LINE_VALID

Propagation Delay for PIXCLK and DOUT

The DOUT signals change on the rising edge of the master input clock, as shown in Figure 27. LINE_VALID asserts at the same time as the first valid pixel data, at the start of a line, and remains asserted until the end of the final valid pixel data for the line (see Figure 8 on page 11).

The timing and behavior of PIXCLK depend upon the Reg0x0A settings.

Figure 27: Propagation Delays for PIXCLK and DOUT Signals



Two-wire Serial Bus Timing

The two-wire serial bus operation requires certain minimum master clock cycles between transitions. These are specified in the following diagrams in master clock cycles.

Figure 28: Serial Host Interface Start Condition Timing

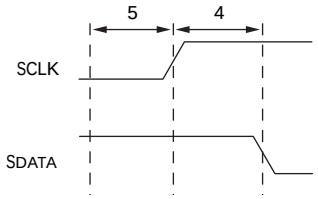
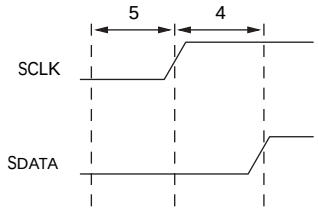
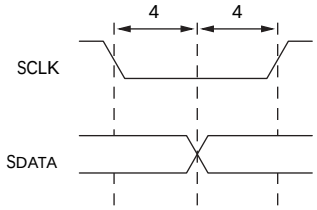


Figure 29: Serial Host Interface Stop Condition Timing



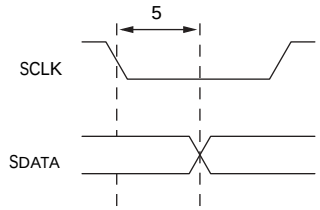
Note: All timing are in units of master clock cycle.

Figure 30: Serial Host Interface Data Timing for Write



Note: SDATA is driven by an off-chip transmitter.

Figure 31: Serial Host Interface Data Timing for Read



Note: SDATA is driven LOW by the sensor, or allowed to be pulled HIGH by a pull-up resistor off-chip.

Figure 32: Acknowledge Signal Timing After an 8-Bit Write to the Sensor

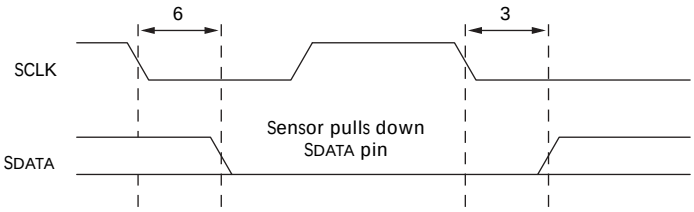
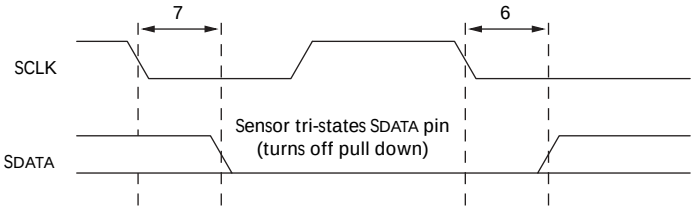
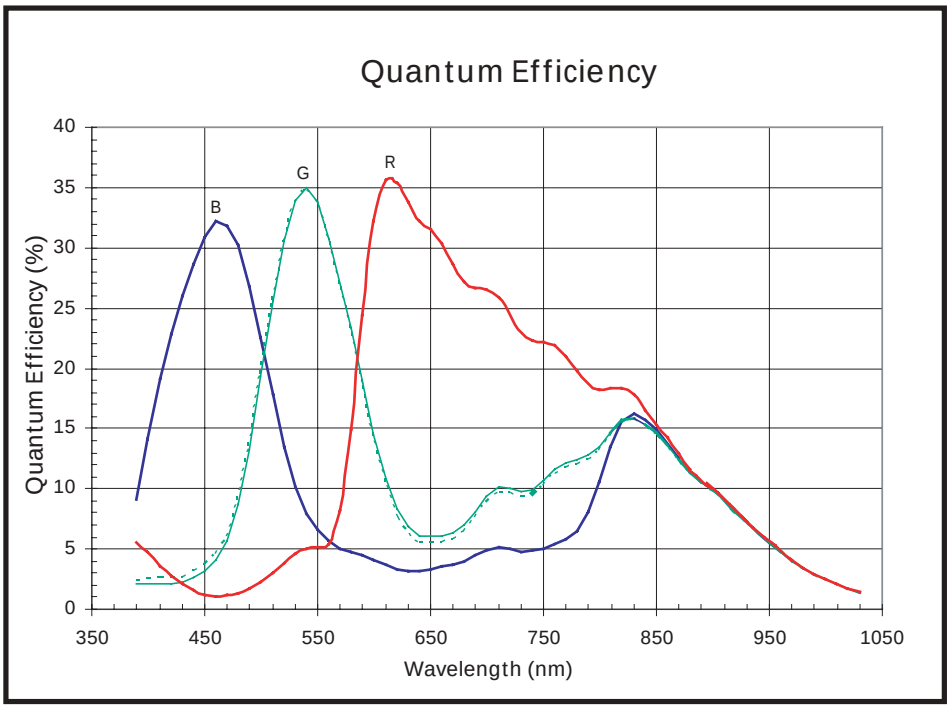


Figure 33: Acknowledge Signal Timing After an 8-Bit Read from the Sensor



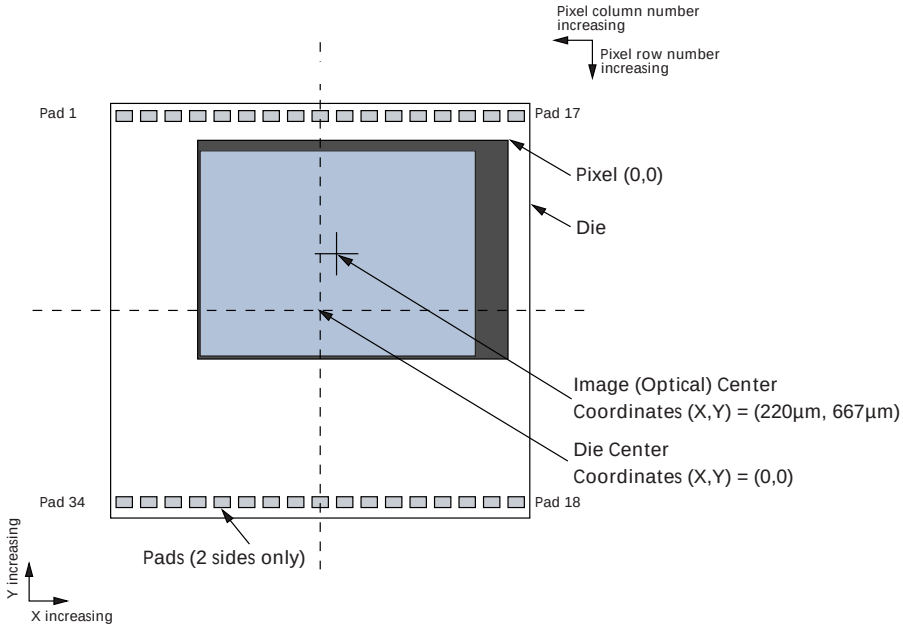
Note: After a read, the master receiver must drive SDATA LOW to acknowledge receipt of data bits. When read sequence is complete, the master must generate a no acknowledge by leaving SDATA to float HIGH. On the following cycle, a start or stop bit may be used.

Figure 34: Typical Spectral Characteristics



Mechanical Specifications

Figure 35: Image Center Offset



Notes: 1. Figure not to scale.

Data Sheet Designation

Preliminary: This data sheet contains initial characterization limits that are subject to change upon full characterization of production devices.

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900
prodmktg@micron.com www.micron.com Customer Comment Line: 800-932-4992

Revision History

Rev B, Preliminary 2/05

- Updated Table 1
- Removed Figure 3, Figure 5, and Figure 38
- Removed Table 3 and Table 4

Rev A, Preliminary 11/04

- Initial release