Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

MT9J003

MT9J003 1/2.3-Inch 10 Mp CMOS Digital Image Sensor

General Description

The ON Semiconductor MT9J003 is a 1/2.3-inch CMOS active-pixel digital imaging sensor with an active pixel array of 3856 (H) x 2764 (V) including border pixels. It can support 10 megapixel (3664 (H) x 2748 (V)) digital still images and a 1080 p (3840 (H) x 2160 (V)) digital video mode. 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 MT9J003 digital image sensor features ON Semiconductor's breakthrough low-noise CMOS imaging technology that achieves near-CCD image quality (based on signal-to-noise ratio and low-light sensitivity) while maintaining the inherent size, cost, and integration advantages of CMOS.

When operated in its default 4:3 still-mode, the sensor generates a full resolution image at 15 frames per second (fps) using the HiSPi serial interface. An on-chip analog-to-digital converter (ADC) generates a 12-bit value for each pixel.

Features

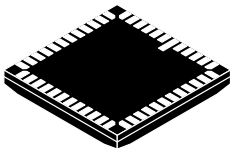
- 1080p Digital Video Mode
- Simple Two-wire Serial Interface
- Auto Black Level Calibration
- Support for External Mechanical Shutter
- Support for External LED or Xenon Flash
- High Frame Rate Preview Mode with Arbitrary Down-size Scaling from Maximum Resolution
- Programmable Controls: Gain, Horizontal and Vertical Blanking, Auto Black Level Offset Correction, Frame Size/rate, Exposure, Left–right and Top–bottom Image Reversal, Window Size, and Panning
- Data Interfaces: Parallel or Four-lane Serial High-speed Pixel Interface (HiSPi) Differential Signaling (Sub-LVDS)
- On-die Phase-locked Loop (PLL) Oscillator
- Bayer Pattern Downsize Scaler
- Integrated Position-based Color and Lens Shading Correction
- One-time Programmable Memory (OTPM) for Storing Module Information

Applications

- Digital Video Cameras
- Digital Still Cameras

ON Semiconductor®

www.onsemi.com



ILCC48 10x10
CASE 847AK

ORDERING INFORMATION

See detailed ordering and shipping information on page 3 of this data sheet.

Table 1. KEY PARAMETERS

Parameter		Value
Optical Format		1/2.3-inch (4:3)
Active Imager Size		6.440 mm (H) x 4.616 mm (V), 7.923 mm Diagonal (Entire Sensor) 6.119 mm (H) x 4.589 mm (V), 7.649 mm Diagonal (Still Mode) 6.413 mm (H) x 3.607 mm (V), 7.358 mm Diagonal (Video Mode)
Active Pixels		3856 (H) x 2764 (V) (Entire Sensor) 3664 (H) x 2748 (V) (4:3, Still Mode) 3840 (H) x 2160 (V) (16:9, Video Mode)
Pixel Size		1.67 x 1.67 μm
Chief Ray Angle		0°, 13.4°
Color Filter Array		RGB Bayer Pattern
Shutter Type		Electronic Rolling Shutter (ERS) with Global Reset Release (GRR)
Maximum Data Rate		96 Mp/s
Maximum Master Clock		60 MHz
Input Clock Frequency		6–48 MHz
Maximum Data Rate	Parallel	80 Mp/s at 80 MHz PIXCLK
	HiSPi (4-lane)	2.8 Gbps
Frame Rate	Still Mode, 4:3 (3664 (H) x 2748 (V)	Programmable up to 15 fps Serial I/F, 7.5 fps Parallel I/F
	Preview Mode VGA	30 fps with Binning 60 fps with Skip2bin2
	1080p Mode (1920 H x 1080 V)	60 fps Using HiSPi I/F 30 fps Using Parallel I/F
ADC Resolution		12-bit, On-die
Responsivity		0.31 V/lux-sec (550 nm)
Dynamic Range		65.2 dB
SNR_{MAX}		34 dB
Supply Voltage	I/O Digital	1.7–1.9 (V) (1.8 (V) Nominal) or 2.4–3.1 (V) (2.8 (V) Nominal)
	Digital	1.7–1.9 (V) (1.8 (V) Nominal)
	Analog	2.4–3.1 (V) (2.8 (V) Nominal)
	SLVS I/O	0.4–0.8 (V) (0.4 or 0.8 (V) Nominal)
Power Consumption	Still Mode at 15 fps w/ Serial I/F	638 mW
	Still Mode at 7.5 fps w/ Parallel I/F	388 mW
	Preview	250 mW Low Power VGA
	Standby	500 μW (Typical, EXTCLK Disabled)
Power Consumption		TBD
Package		48-pin iLCC (10 mm x 10 mm) Bare Die, 48pin Tiny PLCC (12 mm x 12 mm)
Operating Temperature		–30°C to +70°C (at Junction)

ORDERING INFORMATION

Table 2. AVAILABLE PART NUMBERS

Part Number	Product Description	Orderable Product Attribute Description †
MT9J003D00STMUC2CBC1-200	10 MP 1” CIS	Die Sales, 200 μm Thickness
MT9J003I12STCU-DP	10 MP 1/2.3” CIS	Dry Pack with Protective Film
MT9J003I12STCU-DR	10 MP 1/2.3” CIS	Dry Pack without Protective Film
MT9J003I12STCV2-DP	10 MP 1/2.3” CIS	Dry Pack with Protective Film
MT9J003I12STCV2-TP	10 MP 1/2.3” CIS	Tape & Reel with Protective Film
MT9J003I12STMU-DP	10 MP 1/2.3” CIS	Dry Pack with Protective Film

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specification Brochure, BRD8011/D.

FUNCTIONAL OVERVIEW

The MT9J003 is a progressive-scan sensor that generates a stream of pixel data at a constant frame rate. It uses an on-chip, phase-locked loop (PLL) to generate all internal clocks from a single master input clock running between 6 and 48 MHz. The maximum output pixel rate is 80 Mp/s, corresponding to a pixel clock rate of 80 MHz. A block diagram of the sensor is shown in Figure 1.

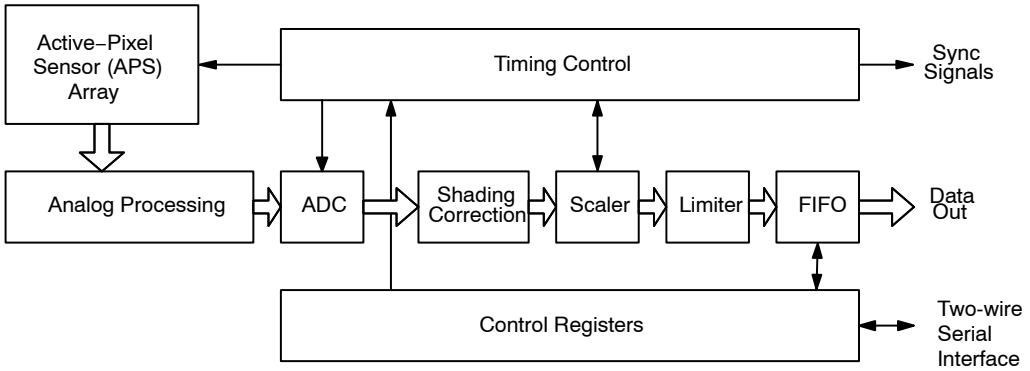


Figure 1. Block Diagram

The core of the sensor is a 10 Mp active-pixel array. The timing and control circuitry sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and reading that row, the pixels in the row integrate incident light. The exposure is controlled by varying the time interval between reset and readout. Once a row has been 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 12-bit value for each pixel in the array. The ADC output passes through a digital processing signal chain (which provides further data path corrections and applies digital gain).

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

The sensor contains a set of control and status registers that can be used to control many aspects of the sensor behavior including the frame size, exposure, and gain setting. These registers can be accessed through a two-wire serial interface.

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

The control registers, timing and control, and digital processing functions shown in Figure 1 are partitioned into three logical parts:

- A sensor core that provides array control and data path corrections. The output of the sensor core is a 12-bit parallel pixel data stream qualified by an output data clock (PIXCLK), together with LINE_VALID (LV) and FRAME_VALID (FV) signals or a 4-lane serial high-speed pixel interface (HiSPi).
- A digital shading correction block to compensate for color/brightness shading introduced by the lens or chief ray angle (CRA) curve mismatch.

- Additional functionality is provided. This includes a horizontal and vertical image scaler, a limiter, a data compressor, an output FIFO, and a serializer.

The output FIFO is present to prevent data bursts by keeping the data rate continuous. Programmable slew rates are also available to reduce the effect of electromagnetic interference from the output interface.

A flash output signal is provided to allow an external xenon or LED light source to synchronize with the sensor exposure time. Additional I/O signals support the provision of an external mechanical shutter.

Pixel Array

The sensor core uses a Bayer color pattern, as shown in Figure 2. The even-numbered rows contain green and red pixels; odd-numbered rows contain blue and green pixels. Even-numbered columns contain green and blue pixels; odd-numbered columns contain red and green pixels.

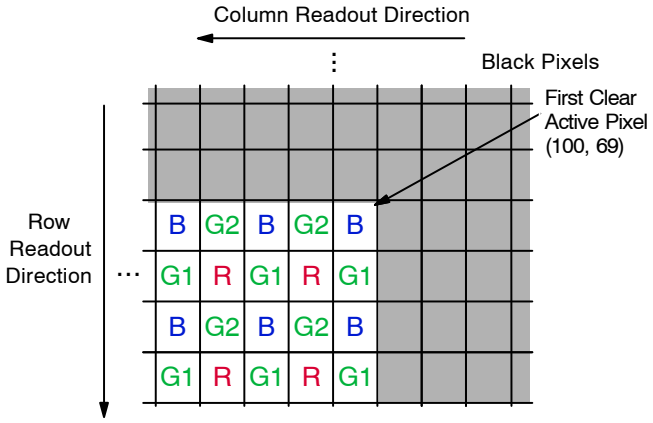


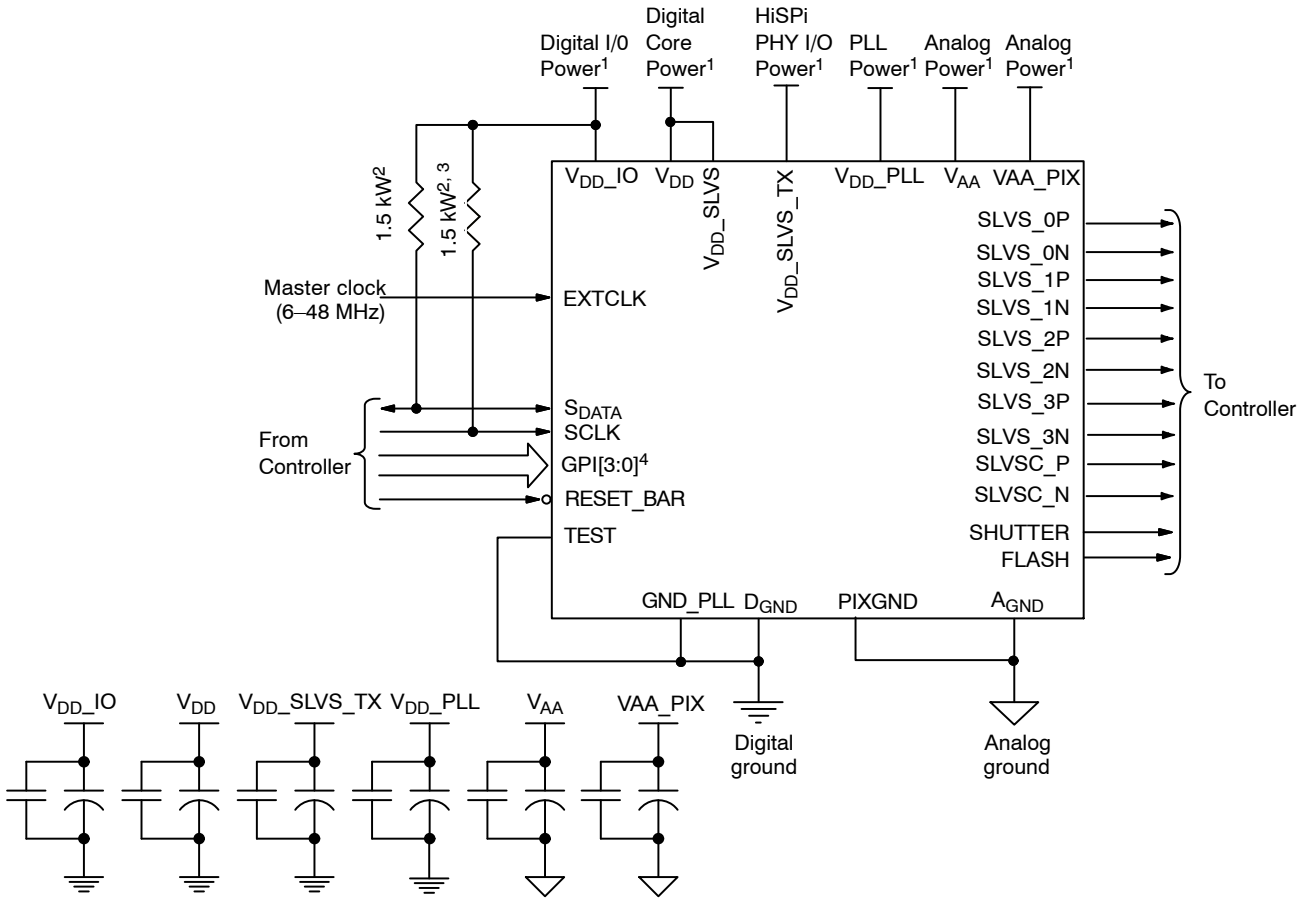
Figure 2. Block Diagram

OPERATING MODES

By default, the MT9J003 powers up with the serial pixel data interface enabled. The sensor can operate in serial HiSPi or parallel mode

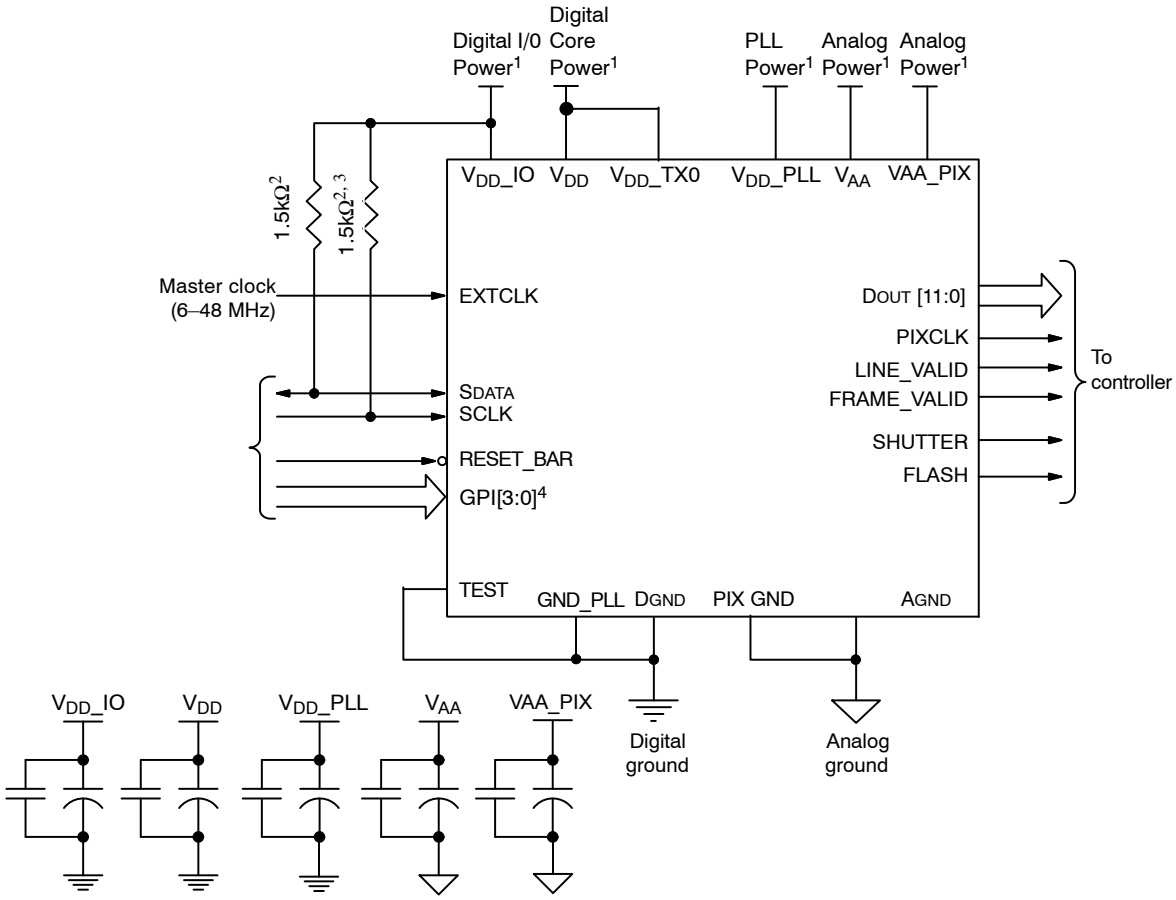
For low-noise operation, the MT9J003 requires separate power supplies for analog and digital power. Incoming digital and analog ground conductors should be placed in such a way that coupling between the two are minimized. Both power supply rails should also be routed in such a way that noise coupling between the two supplies and ground is minimized.

CAUTION: ON Semiconductor does not recommend the use of inductance filters on the power supplies or output signals.



Notes:
1. All power supplies should be adequately decoupled.
2. ON Semiconductor recommends a resistor value of 1.5 kΩ, but it 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. The GPI pins can be statically pulled HIGH or LOW to be used as module IDs, or they can be programmed to perform special functions (TRIGGER, OE_N, SADDR, STANDBY) to be dynamically controlled.
5. VPP, which can be used during the module manufacturing process, is not shown in Figure 3. This pad is left unconnected during normal operation.
6. The parallel interface output pads can be left unconnected if the serial output interface is used.
7. ON Semiconductor recommends that 0.1 µF and 10 µF decoupling capacitors for each power supply are mounted as close as possible to the pad. Actual values and results may vary depending on layout and design considerations. Check the MT9J003 demo headboard schematics for circuit recommendations
8. ON Semiconductor recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.
9. The signal path between the HiSPi serial transmitter and receiver should be adequately designed to minimize any trans-impedance mismatch and/or reflections on the data path.

Figure 3. Typical Configuration: Serial Four-Lane HiSPi Interface



Notes:
1. All power supplies should be adequately decoupled.
2. ON Semiconductor recommends a resistor value of 1.5 kΩ, but it 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. The GPI pins can be statically pulled HIGH or LOW to be used as module IDs, or they can be programmed to perform special functions (TRIGGER, OE_N, SADDR, STANDBY) to be dynamically controlled.
5. VPP, which can be used during the module manufacturing process, is not shown in Figure 4. This pad is left unconnected during normal operation.
6. The serial interface output pads can be left unconnected if the parallel output interface is used.
7. ON Semiconductor recommends that 0.1 µF and 10 µF decoupling capacitors for each power supply are mounted as close as possible to the pad. Actual values and results may vary depending on layout and design considerations. Check the MT9J003 demo headboard schematics for circuit recommendations.
8. ON Semiconductor recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.
9. ON Semiconductor recommends that VDD_TX0 is tied to VDD when the sensor is using the parallel interface.

Figure 4. Typical Configuration: Parallel Pixel Data Interface

SIGNAL DESCRIPTIONS

Table 3 provides signal descriptions for MT9J003 die. For pad location and aperture information, refer to the MT9J003 die data sheet.

Table 3. SIGNAL DESCRIPTIONS

Pad Name	Pad Type	Description
EXTCLK	Input	Master Clock Input, 6–48 MHz
RESET_BAR (XSHUTDOWN)	Input	Asynchronous active LOW reset. When asserted, data output stops and all internal registers are restored to their factory default settings
SCLK	Input	Serial clock for access to control and status registers
GPI[3:0]	Input	General purpose inputs. After reset, these pads are powered-down by default; this means that it is not necessary to bond to these pads. Any of these pads can be configured to provide hardware control of the standby, output enable, SADDR select, and shutter trigger functions. Can be left floating if not used
TEST	Input	Enable manufacturing test modes. It should not be left floating. It can be tied to ground or VDD_IO when used in parallel or HiSPi. It should be connected to DGND for normal operation of the CCP2 configured sensor, or connected to VDD_IO power for the MIPI-configured sensor
SDATA	I/O	Serial data from READs and WRITEs to control and status registers
LINE_VALID	Output	LINE_VALID (LV) output. Qualified by PIXCLK
FRAME_VALID	Output	FRAME_VALID (FV) output. Qualified by PIXCLK
DOUT[11:0]	Output	Parallel pixel data output. Qualified by PIXCLK
PIXCLK	Output	Pixel clock. Used to qualify the LV, FV, and DOUT[11:0] outputs
FLASH	Output	Flash output. Synchronization pulse for external light source. Can be left floating if not used
SHUTTER	Output	Control for external mechanical shutter. Can be left floating if not used
VPP	Supply	Power supply used to program one-time programmable (OTP) memory. Disconnect pad when not programming or when feature is not used
VDD_TX0	Supply	PHY power supply. Digital power supply for the MIPI or CCP2 serial data interface. ON Semiconductor recommends that VDD_TX0 is always tied to VDD when using an unpackaged sensor
VDD_SLVS	Supply	HiSPi power supply for data and clock output. This should be tied to VDD
VDD_SLVS_TX	Supply	Digital Power Supply for the HiSPi I/O
VAA	Supply	Analog Power Supply
VAA_PIX	Supply	Analog Power Supply for the Pixel Array
AGND	Supply	Analog Ground
VDD	Supply	Digital Power Supply
VDD_IO	Supply	I/O Power Supply
DGND	Supply	Common Ground for Digital and I/O
VDD_PLL	Supply	PLL Power Supply
GND_PLL	Supply	PLL Ground
PIXGND	Supply	Pixel Ground
SLVS_0P	Output	Lane 1 Differential HiSPi (LVDS) Serial Data (positive). Qualified by the SLVS Serial Clock
SLVS_0N	Output	Lane 1 Differential HiSPi (LVDS) Serial Data (negative). Qualified by the SLVS Serial Clock
SLVS_1P	Output	Lane 2 Differential HiSPi (LVDS) Serial Data (positive). Qualified by the SLVS Serial Clock
SLVS_1N	Output	Lane 2 Differential HiSPi (LVDS) Serial Data (negative). Qualified by the SLVS Serial Clock
SLVS_2P	Output	Lane 3 Differential HiSPi (LVDS) Serial Data (positive). Qualified by the SLVS Serial Clock

Table 3. SIGNAL DESCRIPTIONS (continued)

Pad Name	Pad Type	Description
SLVS_2N	Output	Lane 3 Differential HiSPi (LVDS) Serial Data (negative). Qualified by the SLVS Serial Clock
SLVS_3P	Output	Lane 4 Differential HiSPi (LVDS) Serial Data (positive). Qualified by the SLVS Serial Clock
SLVS_3N	Output	Lane 4 Differential HiSPi (LVDS) Serial Data (negative). Qualified by the SLVS Serial Clock
SLVS_CP	Output	Differential HiSPi (LVDS) Serial Clock (positive). Qualified by the SLVS Serial Clock
SLVS_CN	Output	Differential HiSPi (LVDS) Serial Clock (positive). Qualified by the SLVS Serial Clock

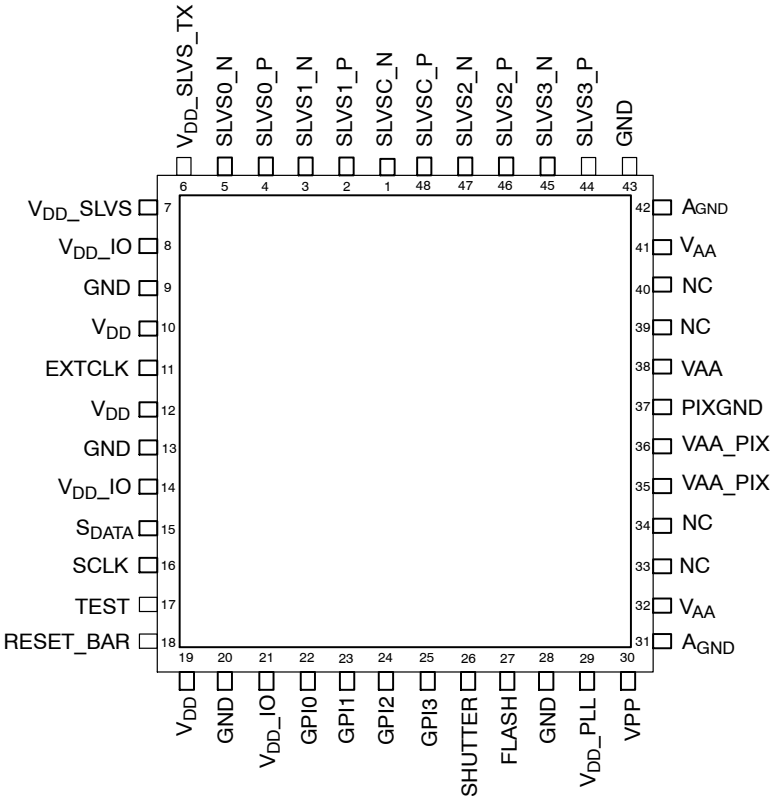


Figure 5. HiSPi Package Pinout Diagram

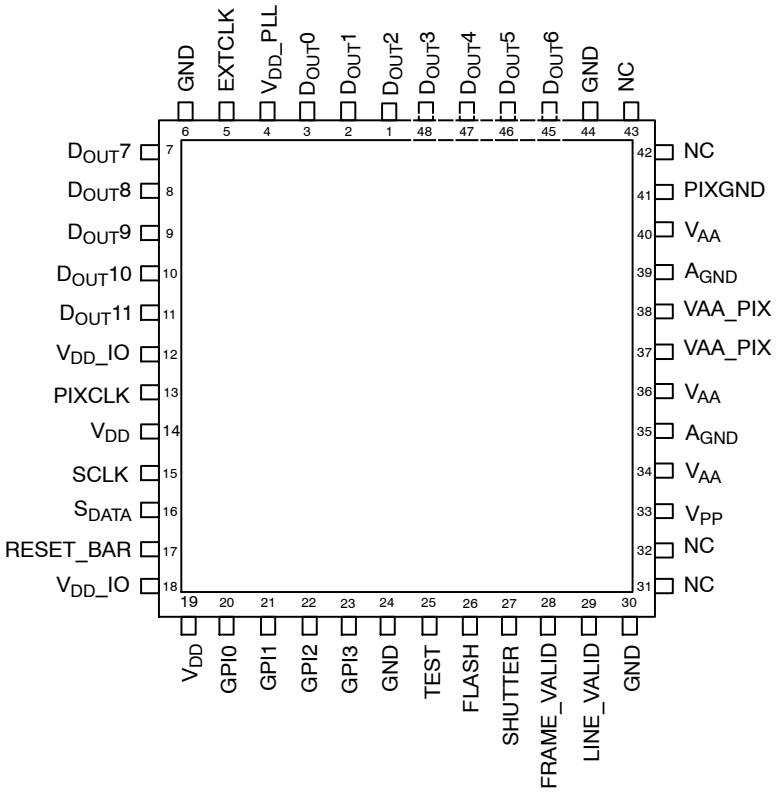


Figure 6. 48−Pin iLCC Parallel Package Pinout Diagram

OUTPUT DATA FORMAT

Serial Pixel Data Interface

The MT9J003 supports RAW8, RAW10, and RAW12 image data formats over a serial interface. The sensor supports a 1 and 2-lane MIPI as well as the HiSPi interface. These interfaces are not described in the data sheet.

High Speed Serial Pixel Interface

The High Speed Serial Pixel (HiSPi) interface uses four data and one clock low voltage differential signaling (LVDS) outputs.

- SLVS_CP, SLVS_CN
- SLVS_[0:3]P, SLVS_[0:3]N

The HiSPi interface supports two protocols, streaming and packetized. The streaming protocol conforms to a standard video application where each line of active or intra-frame blanking provided by the sensor is transmitted at the same length. The packetized protocol will transmit only the active data ignoring line-to-line and frame-to-frame blanking data.

The HiSPi interface building block is a unidirectional differential serial interface with four data and one double data rate (DDR) clock lanes. One clock for every four serial data lanes is provided for phase alignment across multiple lanes. Figure 7 shows the configuration between the HiSPi transmitter and the receiver.

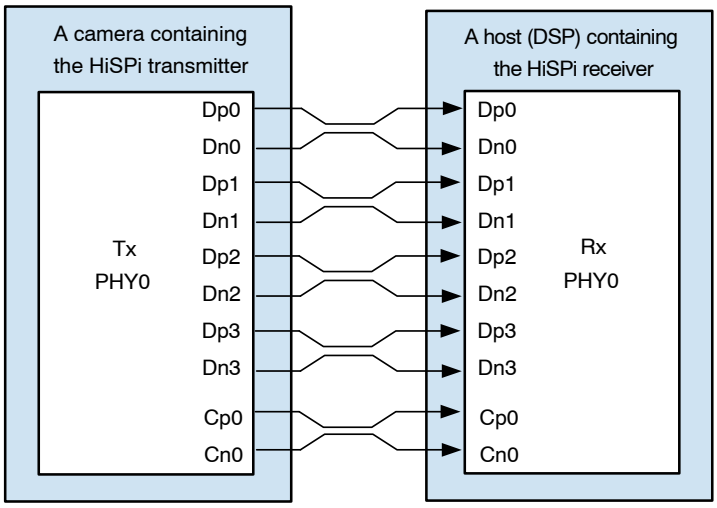


Figure 7. HiSPi Transmitter and Receiver Interface Block Diagram

HiSPi Physical Layer

The HiSPi physical layer is partitioned into blocks of four data lanes and an associated clock lane. Any reference to the PHY in the remainder of this document is referring to this minimum building block.

The PHY will serialize a 10-, 12-, 14- or 16-bit data word and transmit each bit of data centered on a rising edge of the clock, the second on the following edge of clock. Figure 8 shows bit transmission. In this example, the word is transmitted in order of MSB to LSB. The receiver latches data at the rising and falling edge of the clock.

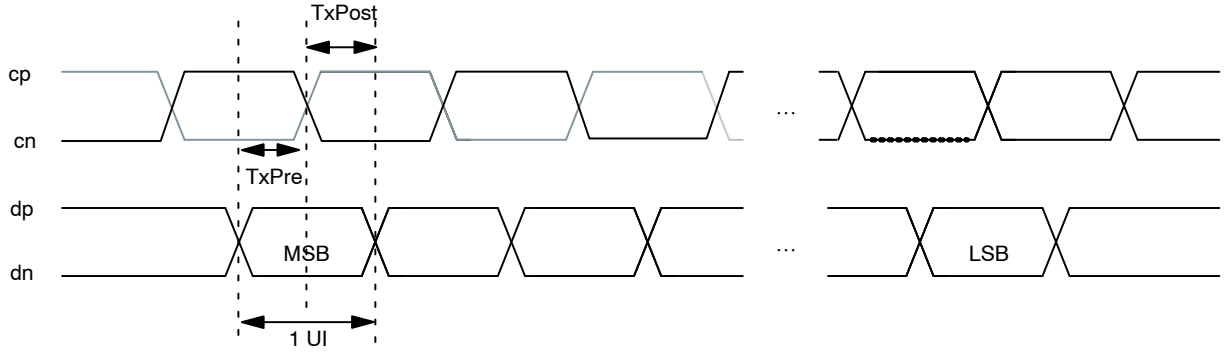


Figure 8. Timing Diagram

DLL Timing Adjustment

The specification includes a DLL to compensate for differences in group delay for each data lane. The DLL is connected to the clock lane and each data lane, which acts as a control master for the output delay buffers. Once the DLL has gained phase lock, each lane can be delayed in 1/8 unit interval (UI) steps. This additional delay allows the user to increase the setup or hold time at the receiver circuits and can be used to compensate for skew introduced in PCB design.

If the DLL timing adjustment is not required, the data and clock lane delay settings should be set to a default code of 0x000 to reduce jitter, skew, and power dissipation.

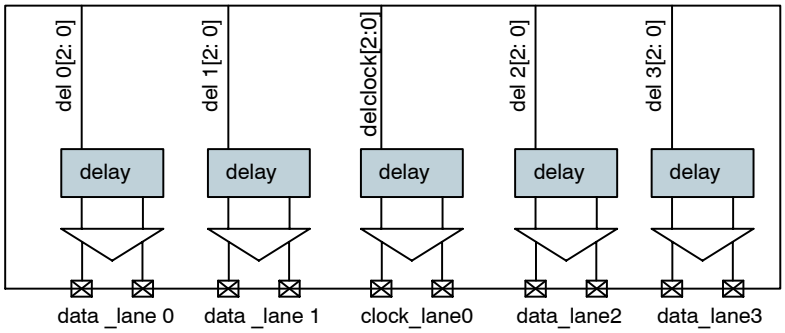


Figure 9. Block Diagram of DLL Timing Adjustment

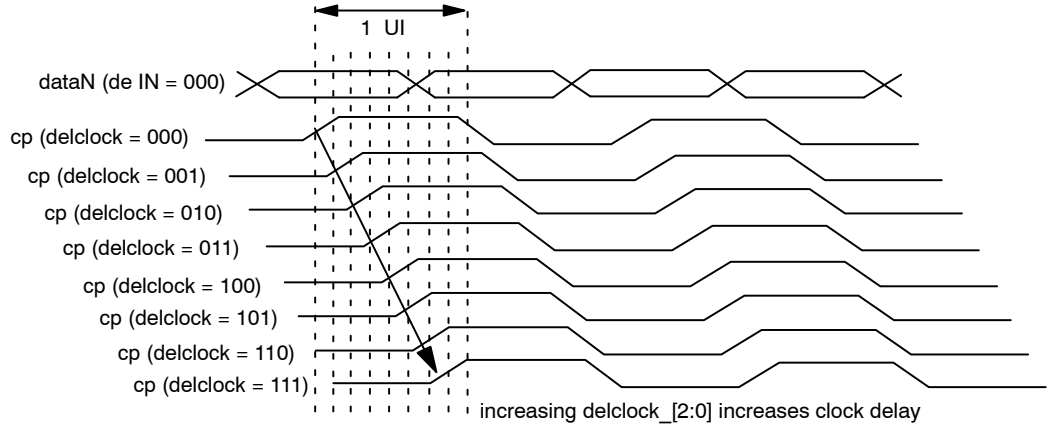


Figure 10. Delaying the clock_lane with Respect to data_lane

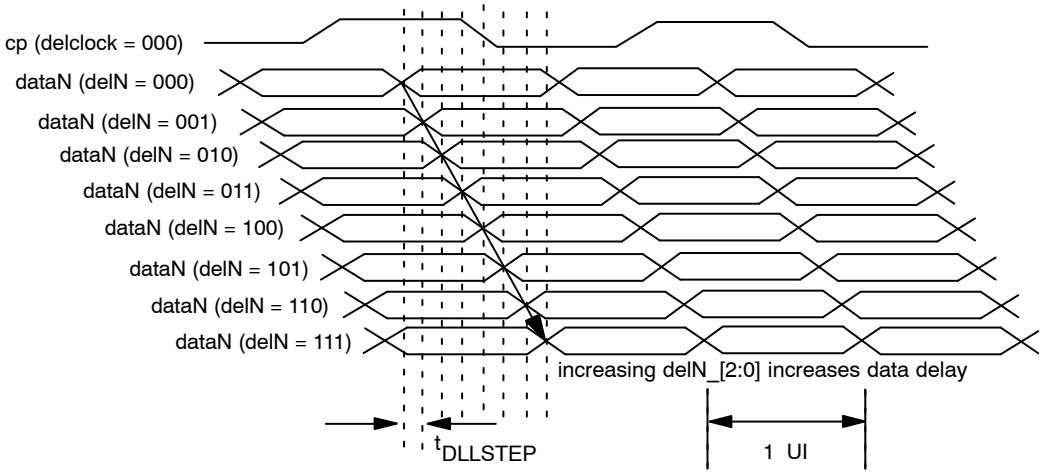


Figure 11. Delaying data_lane with Respect to the clock_lane

HiSPi Streaming Mode Protocol Layer

The protocol layer is positioned between the output data path of the sensor and the physical layer. The main functions of the protocol layer are generating sync codes, formatting pixel data, inserting horizontal/vertical blanking codes, and distributing pixel data over defined data lanes.

The HiSPi interface can only be configured when the sensor is in standby. This includes configuring the interface to transmit across 1, 2, or all 4 data lanes.

Protocol Fundamentals

Referring to Figure 12, it can be seen that a SYNC code is inserted in the serial data stream prior to each line of image data. The streaming protocol will insert a SYNC code to transmit each active data line and vertical blanking lines.

The packetized protocol will transmit a SYNC code to note the start and end of each row. The packetized protocol uses sync a “Start of Frame” (SOF) sync code at the start of a frame and a “Start of Line” (SOL) sync code at the start of a line within the frame. The protocol will also transmit an “End of Frame” (EOF) at the end of a frame and an “End of Line” (EOL) sync code at the end of a row within the frame

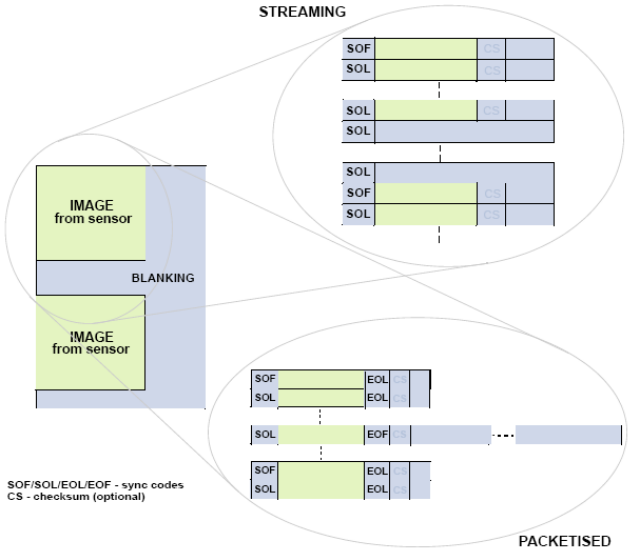


Figure 12. Steaming vs. Packetized Transmission

Parallel Pixel Data Interface

MT9J003 image data is read out in a progressive scan. Valid image data is surrounded by horizontal blanking and vertical blanking, as shown in Figure 13. The amount of horizontal blanking and vertical blanking is programmable; LV is HIGH during the shaded region of the figure. FV timing is described in the "Output Data Timing (Parallel Pixel Data Interface)".

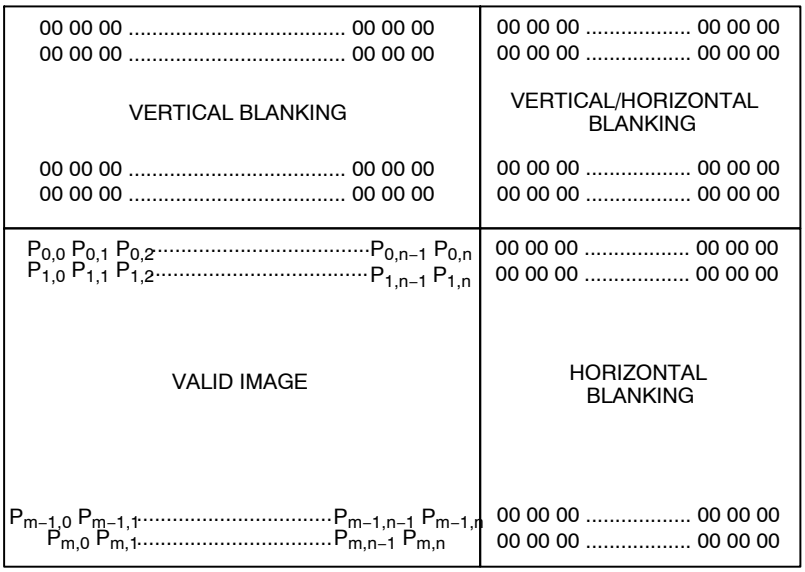


Figure 13. Spatial Illustration of Image Readout

Output Data Timing (Parallel Pixel Data Interface)

MT9J003 output data is synchronized with the PIXCLK output. When LV is HIGH, one pixel value is output on the 12-bit DOUT output every PIXCLK period. The pixel clock frequency can be determined based on the sensor's master input clock and internal PLL configuration. The rising edges on the PIXCLK signal occurs one-half of a pixel clock period after transitions on LV, FV, and DOUT (see Figure 14). This allows PIXCLK to be used as a clock to sample the data. PIXCLK is continuously enabled, even during the blanking period. The MT9J003 can be programmed to delay the PIXCLK edge relative to the DOUT transitions. This can be achieved by programming the corresponding bits in the row_speed register. The parameters P, A, and Q in Figure 15 are defined in Table 4.

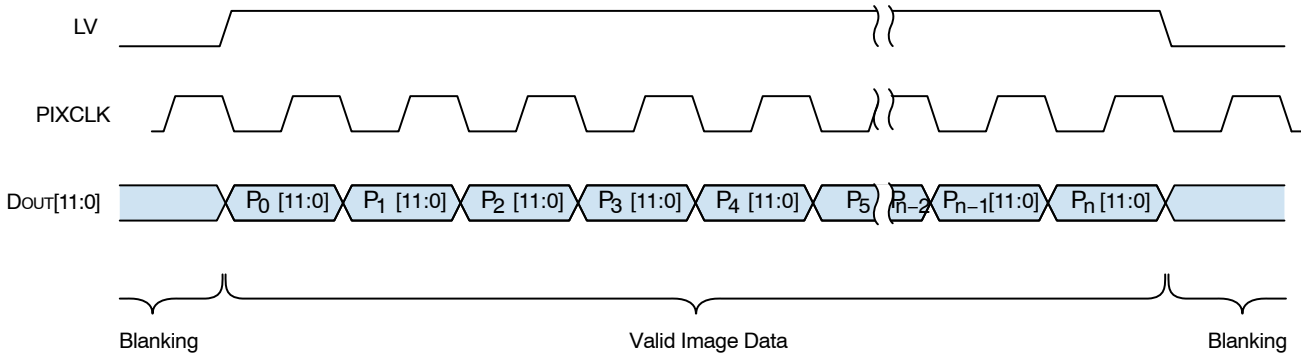


Figure 14. Pixel Data Timing Example

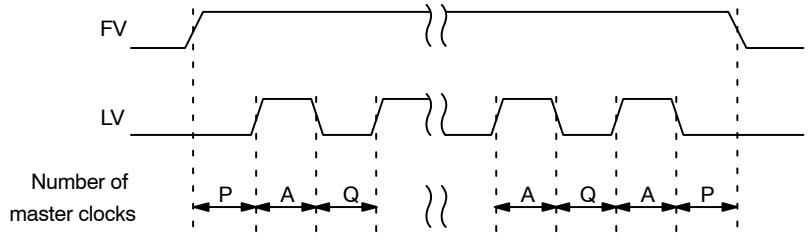


Figure 15. Row Timing and FV/LV Signals

The sensor timing (shown in Table 4) is shown in terms of pixel clock and master clock cycles (see Figure 14). The default settings for the on-chip PLL generate a pixel array clock (vt_pix_clk) of 160 MHz and an output clock (op_pix_clk) of 40 MHz given a 20 MHz input clock to the MT9J003. Equations for calculating the frame rate are given in “Frame Rate Control”.

Table 4. ROW TIMING WITH HiSPi INTERFACE

Parameter	Name	Equation	Default Timing
PIXCLK_ PERIOD	Pixel Clock Period	1/vt_pix_clk_freq_mhz	1 Pixel Clock = 6.25 ns
S	Skip (Subsampling) Factor	For x_odd_inc = y_odd_inc = 3, S = 2. For x_odd_inc = y_odd_inc = 7, S = 4. otherwise, S = 1 For y_odd_inc = 3, S = 2 For y_odd_inc = 7, S = 4 For y_odd_inc = 15, S = 8 For y_odd_inc = 31, S = 16 For y_odd_inc = 63, S = 32	1
A	Active Data Time	(x_addr_end – x_addr_start + x_odd_inc) * 0.5 * PIXCLK_PERIOD/S = 3775 – 112 + 12	1832 Pixel Clock = 11.45 µs
P	Frame Start/end Blanking	6 * PIXCLK_PERIOD	6 Pixel Clock = 37.5 ns
Q	Horizontal Blanking	(line_length_pck – A) * PIXCLK_PERIOD = 3694 – 1832	1862 Pixel Clock = 11.63 µs
A + Q	Row Time	line_length_pck * PIXCLK_PERIOD	3694 Pixel Clock = 23.09 µs
N	Number of Rows	(y_addr_end – y_addr_start + y_odd_inc) / S = (2755 – 8 + 1)/1	2748 Rows
V	Vertical Blanking	((frame_length_lines – N) * (A + Q)) + Q – (2 * P) = (2891 – 2748) * 3694 + 1862 – 12	530092 Pixel Clock = 3.31 ms
T	Frame Valid Time	(N * (A + Q)) – Q + (2 * P) = 2748*3694 – 1862 + 12	10149262 Pixel Clock = 63.42 ms
F	Total Frame Time	line_length_pck * frame_length_lines * PIXCLK_PERIOD = 2891 * 3694	10679354 Pixel Clock = 66.75 ms

Table 5. ROW TIMING WITH PARALLEL INTERFACE

Parameter	Name	Equation	Default Timing
PIXCLK_PERIOD	Pixel Clock Period	1/vt_pix_clk_freq_mhz	1 Pixel Clock = 6.25 ns
S	Skip (Subsampling) Factor	For x_odd_inc = y_odd_inc = 3, S = 2. For x_odd_inc = y_odd_inc = 7, S = 4. otherwise, S = 1 For y_odd_inc = 3, S = 2 For y_odd_inc = 7, S = 4 For y_odd_inc = 15, S = 8 For y_odd_inc = 31, S = 16 For y_odd_inc = 63, S = 32	1
A	Active Data Time	(x_addr_end – x_addr_start + x_odd_inc) * 0.5 * PIXCLK_PERIOD/S = (3775 – 112+1)/2	1832 Pixel Clocks = 11.45 µs
P	Frame Start/end Blanking	6 * PIXCLK_PERIOD	6 Pixel Clocks = 75 ns
Q	Array Horizontal Blanking	(line_length_pck – A) * PIXCLK_PERIOD = 7358 – 1832	5526 Pixel Clocks = 34.5 µs External horizontal blanking is 30 pixel clocks or 187 ns.
A + Q	Row Time Limited by Output Interface Speed	x_output_size * clk_pixel/clk_op + 30 = 3664 * 160 MHz/80 MHz + 30	7358 Pixel Clocks = 46.1 µs
N	Number of Rows	(y_addr_end – y_addr_start + y_odd_inc) / S = (2755 – 8 + 1)/1	2748 rows
V	Vertical Blanking	((frame_length_lines – N) * (A + Q)) + Q – (2 * P) = (2891 – 2748)*7358 + 1862 – 12	1054044 Pixel Clocks = 6.59 ms
T	Frame Valid Time	(N * (A + Q)) – Q + (2 * P) = 2748 * 7358 – 1862 + 12	20217934 Pixel Clocks = 126.36 ms
F	Total Frame Time	line_length_pck * frame_length_lines * PIXCLK_PERIOD = 2891 * 37358	21271978 Pixel Clocks = 132.95 ms

Table 6. ROW TIMING WITH PARALLEL INTERFACE USING LOW POWER MODE

Parameter	Name	Equation	Default Timing
PIXCLK_PERIOD	Pixel Clock Period	1/vt_pix_clk_freq_mhz	1 Pixel Clock = 12.5 ns
S	Skip (Subsampling) Factor	For x_odd_inc = y_odd_inc = 3, S = 2. For x_odd_inc = y_odd_inc = 7, S = 4. otherwise, S = 1 For y_odd_inc = 3, S = 2 For y_odd_inc = 7, S = 4 For y_odd_inc = 15, S = 8 For y_odd_inc = 31, S = 16 For y_odd_inc = 63, S = 32	1
A	Active Data Time	(x_addr_end – x_addr_start + x_odd_inc) * 0.5 * PIXCLK_PERIOD/S = (3775–112+1)/2	1832 Pixel Clocks = 22.9 µs
P	Frame Start/end Blanking	6 * PIXCLK_PERIOD	6 Pixel Clocks = 75 ns

Table 6. ROW TIMING WITH PARALLEL INTERFACE USING LOW POWER MODE (continued)

Parameter	Name	Equation	Default Timing
Q	Array Horizontal Blanking	(line_length_pck – A) * PIXCLK_PERIOD = 3694 – 1832	1862 Pixel Clocks = 23.2 µs External horizontal blanking is 30 pixel clocks or 375 ns.
A + Q	Row Time Limited by Output Interface Speed	x_output_size * clk_pixel/clk_op + 30 = 3664 * 80 MHz/80 MHz + 30	3694 Pixel Clocks = 46.1 µs
N	Number of Rows	(y_addr_end – y_addr_start + y_odd_inc) / S = (2755 – 8 + 1)/1	2748 Rows
V	Vertical Blanking	((frame_length_lines – N) * (A + Q)) + Q – (2 * P) = (2891 – 2748) * 7358 + 1862 – 12	530092 Pixel Clocks = 6.63 ms
T	Frame Valid Time	(N * (A + Q)) – Q + (2 * P) = 2748 * 3694 – 1862 + 12	10149262 Pixel Clocks = 126.86 ms
F	Total Frame Time	line_length_pck * frame_length_lines * PIXCLK_PERIOD = 2891 * 3694	10679354 Pixel Clocks = 133.5 ms

Frame Rates at Common Resolutions

Table 7 shows examples of register settings to achieve common resolutions and their frame rates.

Table 7. REGISTER SETTINGS FOR COMMON RESOLUTIONS

Resolution	Interface	Frame Rate	Subsampling Mode	x_addr_start	x_addr_end	y_addr_start	y_addr_end
3664x2748 (Full Resolution)	HiSPi	14.7 fps	N/A	112	3775	8	2755
	Parallel	7.5 fps					
1920x1080 (1080p HDTV)	HiSPi	59.94 fps	2 x 2 Summing	32	3873	296	2453
	Parallel	29.97 fps					
1280x720 (720p HDTV)	HiSPi and Parallel	59.94 fps	2 x 2 Summing	32	3873	296	2453
1408x792 + 10% EIS (720p HDTV + 10% EIS)	HiSPi and Parallel	59.94 fps	2 x 2 Summing	624	3437	304	1885
640x480 (Low Power Monitor)	HiSPi and Parallel	29.97 fps	Sum2Skip2	112	3769	8	2753

TWO-WIRE SERIAL REGISTER INTERFACE

The two-wire serial interface bus enables read/write access to control and status registers within the MT9J003.

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 is used to synchronize transfers. Data is transferred between the master and the slave on a bidirectional signal (SDATA). SDATA is pulled up to VDD off-chip by a 1.5 kΩ resistor. Either the slave or master device can drive SDATA LOW-the interface protocol determines which device is allowed to drive SDATA at any given time.

The protocols described in the two-wire serial interface specification allow the slave device to drive SCLKLOW; the MT9J003 uses SCLK as an input only and therefore never drives it LOW.

Protocol

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

1. a (repeated) start condition
2. a slave address/data direction byte
3. an (a no-) acknowledge bit
4. a message byte
5. a stop condition

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

Start Condition

A start condition is defined as a HIGH-to-LOW transition on SDATA while SCLK is HIGH. At the end of a transfer, the master can generate a start condition without previously generating a stop condition; this is known as a "repeated start" or "restart" condition.

Stop Condition

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

Data Transfer

Data is transferred serially, 8 bits at a time, with the MSB transmitted first. Each byte of data is followed by an acknowledge bit or a no-acknowledge bit. This data transfer mechanism is used for the slave address/data direction byte and for message bytes.

One data bit is transferred during each SCLK clock period. SDATA can change when SCLK is LOW and must be stable while SCLK is HIGH.

Slave Address

Bits [7:1] of this byte represent the device slave address and bit [0] indicates the data transfer direction. A "0" in bit [0] indicates a WRITE, and a "1" indicates a READ. The default slave addresses used by the MT9J003 for the MIPI configured sensor are 0x6C (write address) and 0x6D (read address) in accordance with the MIPI specification. Alternate slave addresses of 0x6E (write address) and 0x6F (read address) can be selected by enabling and asserting the SADDR signal through the GPI pad. But for the CCP2 configured sensor, the default slave addresses used are 0x20 (write address) and 0x21 (read address) in accordance with the SMIA specification. Also, alternate slave addresses of 0x30 (write address) and 0x31 (read address) can be selected by enabling and asserting the SADDR signal through the GPI pad.

An alternate slave address can also be programmed through R0x31FC.

Message Byte

Message bytes are used for sending register addresses and register write data to the slave device and for retrieving register read data.

Acknowledge Bit

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

No-Acknowledge Bit

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

Typical Sequence

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

If the request was a WRITE, the master then transfers the 16-bit register address to which the WRITE should take place. This transfer takes place as two 8-bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master then transfers the data as an 8-bit sequence; the slave sends an acknowledge bit at the end of the sequence. The master stops writing by generating a (re)start or stop condition.

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

Single READ From Random Location

This sequence (Figure 16) starts with a dummy WRITE to the 16-bit address that is to be used for the READ. The master terminates the WRITE by generating a restart condition. The master then sends the 8-bit read slave address/data direction byte and clocks out one byte of register data. The master terminates the READ by generating a no-acknowledge bit followed by a stop condition. Figure 16 shows how the internal register address maintained by the MT9J003 is loaded and incremented as the sequence proceeds.

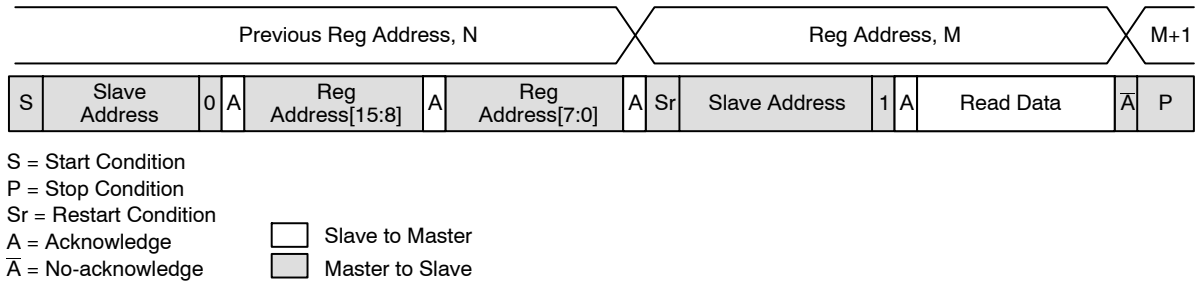


Figure 16. Single READ from Random Location

Single READ From Current Location

This sequence (Figure 17) performs a read using the current value of the MT9J003 internal register address. The master terminates the READ by generating a no-acknowledge bit followed by a stop condition. The figure shows two independent READ sequences.

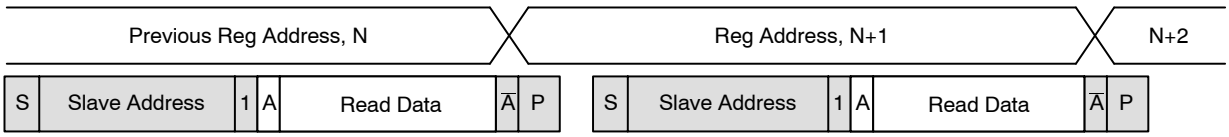


Figure 17. Single READ from Current Location

Sequential READ, Start From Random Location

This sequence (Figure 18) starts in the same way as the single READ from random location (Figure 16). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte READs until "L" bytes have been read.

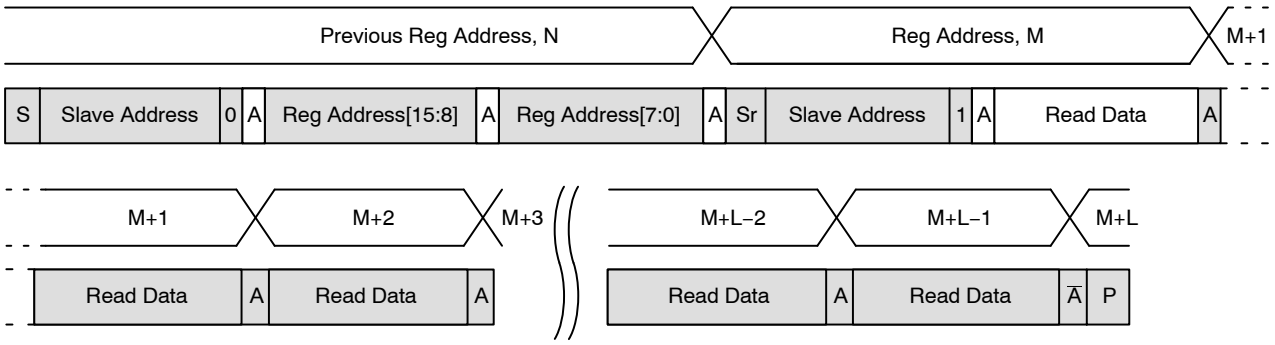


Figure 18. Sequential READ, Start from Random Location

Sequential READ, Start From Current Location

This sequence (Figure 19) starts in the same way as the single READ from current location (Figure 17). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte READs until "L" bytes have been read.

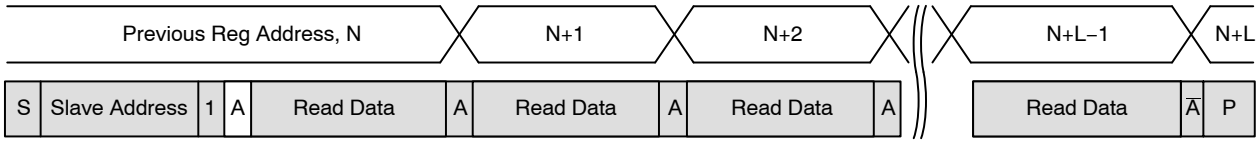


Figure 19. Sequential READ, Start from Current Location

Single WRITE to Random Location

This sequence (Figure 20) begins with the master generating a start condition. The slave address/data direction byte signals a WRITE and is followed by the HIGH then LOW bytes of the register address that is to be written. The master follows this with the byte of write data. The WRITE is terminated by the master generating a stop condition.

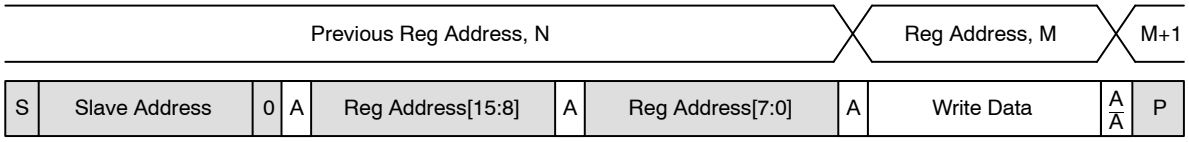


Figure 20. Single WRITE to Random Location

Sequential WRITE, Start at Random Location

This sequence (Figure 21) starts in the same way as the single WRITE to random location (Figure 20). Instead of generating a no-acknowledge bit after the first byte of data has been transferred, the master generates an acknowledge bit and continues to perform byte WRITEs until "L" bytes have been written. The WRITE is terminated by the master generating a stop condition.

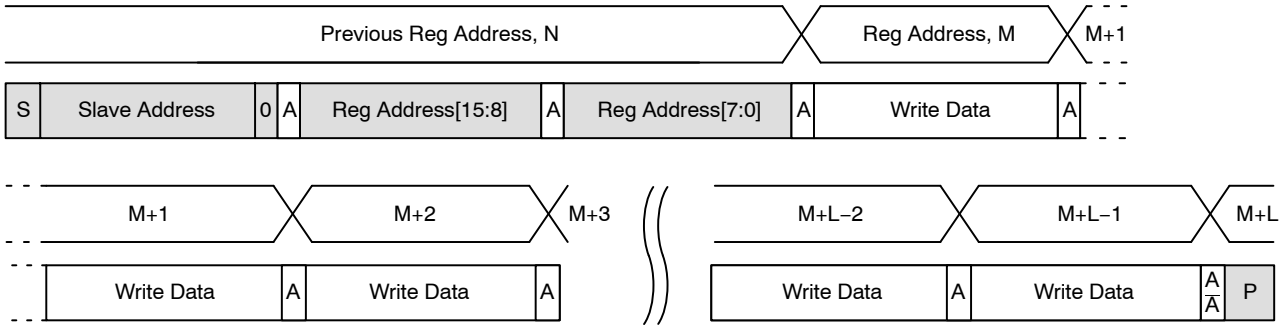


Figure 21. Sequential WRITE, Start at Random Location

PROGRAMMING RESTRICTIONS

The following sections list programming rules that must be adhered to for correct operation of the MT9J003.

Table 8. DEFINITIONS FOR PROGRAMMING RULES

Name	Definition
xskip	xskip = 1 if x_odd_inc = 1; xskip = 2 if x_odd_inc = 3; xskip = 4 if x_odd_inc = 7
yskip	yskip = 1 if y_odd_inc = 1; yskip = 2 if y_odd_inc = 3; yskip = 4 if y_odd_inc = 7; yskip = 8 if y_odd_inc = 15; yskip = 16 if y_odd_inc = 31; yskip = 32 if y_odd_inc = 63

X Address Restrictions

The minimum column address available for the sensor is 24. The maximum value is 3879.

Effect of Scaler on Legal Range of Output Sizes

When the scaler is enabled, it is necessary to adjust the values of x_output_size and y_output_size to match the image size generated by the scaler. The MT9J003 will operate incorrectly if the x_output_size and y_output_size are significantly larger than the output image. To understand the reason for this, consider the situation where the sensor is operating at full resolution and the scaler is enabled with a scaling factor of 32 (half the number of pixels in each direction). This situation is shown in Figure 22.

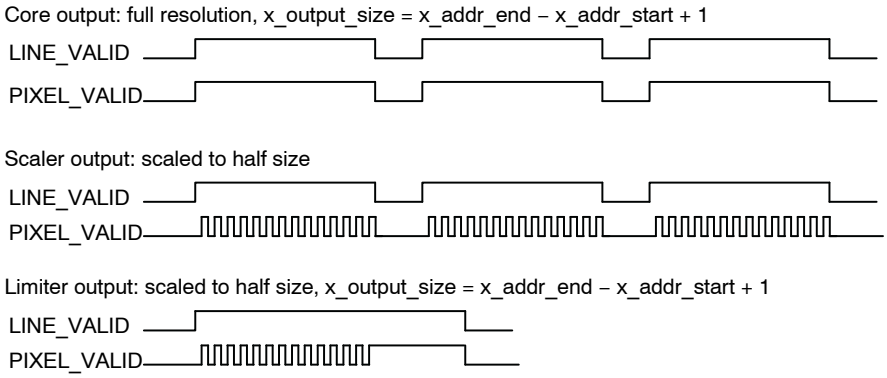


Figure 22. Effect of Limiter on the Data Path

In Figure 22, three different stages in the data path (see "Timing Specifications") are shown. The first stage is the output of the sensor core. The core is running at full resolution and x_output_size is set to match the active array size. The LV signal is asserted once per row and remains asserted for *N* pixel times. The PIXEL_VALID signal toggles with the same timing as LV, indicating that all pixels in the row are valid.

The second stage is the output of the scaler, when the scaler is set to reduce the image size by one-half in each dimension. The effect of the scaler is to combine groups of pixels. Therefore, the row time remains the same, but only half the pixels out of the scaler are valid. This is signaled by transitions in PIXEL_VALID. Overall, PIXEL_VALID is asserted for (*N*/2) pixel times per row.

The third stage is the output of the limiter when the x_output_size is still set to match the active array size. Because the scaler has reduced the amount of valid pixel data without reducing the row time, the limiter attempts to pad the row with (*N*/2) additional pixels. If this has the effect of extending LV across the whole of the horizontal blanking time, the MT9J003 will cease to generate output frames.

A correct configuration is shown in Figure 23, in addition to showing the x_output_size reduced to match the output size of the scaler. In this configuration, the output of the limiter does not extend LV.

Figure 23 also shows the effect of the output FIFO, which forms the final stage in the data path. The output FIFO merges the intermittent pixel data back into a contiguous stream. Although not shown in this example, the output FIFO is also capable of operating with an output clock that is at a different frequency from its input clock.

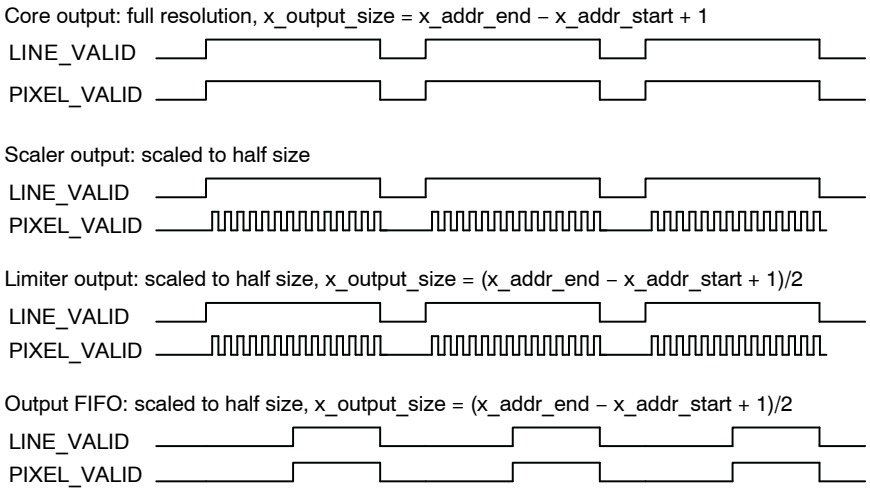


Figure 23. Timing of Data Path

Output Data Timing

The output FIFO acts as a boundary between two clock domains. Data is written to the FIFO in the VT (video timing) clock domain. Data is read out of the FIFO in the OP (output) clock domain.

When the scaler is disabled, the data rate in the VT clock domain is constant and uniform during the active period of each pixel array row readout. When the scaler is enabled, the data rate in the VT clock domain becomes intermittent, corresponding to the data reduction performed by the scaler.

A key constraint when configuring the clock for the output FIFO is that the frame rate out of the FIFO must exactly match the frame rate into the FIFO. When the scaler is disabled, this constraint can be met by imposing the rule that the row time on the serial data stream must be greater than or equal to the row time at the pixel array. The row time on the serial data stream is calculated from the x_output_size and the data_format (8, 10, or 12 bits per pixel), and must include the time taken in the serial data stream for start of frame/row, end of row/frame and checksum symbols.

CAUTION: If this constraint is not met, the FIFO will either underrun or overrun. FIFO underrun or overrun is a fatal error condition that is signalled through the data path_status register (R0x306A).

Changing Registers While Streaming

The following registers should only be reprogrammed while the sensor is in software standby:

- vt_pix_clk_div
- vt_sys_clk_div
- pre_pll_clk_div
- pll_multiplier
- op_pix_clk_div
- op_sys_clk_div

Programming Restrictions When Using Global Reset

Interactions between the registers that control the global reset imposes some programming restrictions on the way in which they are used; these are discussed in ”Global Reset”.

CONTROL OF THE SIGNAL INTERFACE

This section describes the operation of the signal interface in all functional modes.

Serial Register Interface

The serial register interface uses these signals:

- SCLK
- SDATA
- SADDR (through the GPI pad)

SCLK is an input-only signal and must always be driven to a valid logic level for correct operation; if the driving device can place this signal in High-Z, an external pull-up resistor should be connected on this signal.

SDATA is a bidirectional signal. An external pull-up resistor should be connected on this signal.

SADDR is a signal, which can be optionally enabled and controlled by a GPI pad, to select an alternate slave address. These slave addresses can also be programmed through R0x31FC.

This interface is described in detail in ”Two-Wire Serial Register Interface”.

Parallel Pixel Data Interface

The parallel pixel data interface uses these output-only signals:

- FV
- LV
- PIXCLK
- DOUT[11:0]

The parallel pixel data interface is disabled by default at power up and after reset. It can be enabled by programming R0x301A. Table 10 shows the recommended settings.

When the parallel pixel data interface is in use, the serial data output signals can be left unconnected. Set reset_register[12] to disable the serializer while in parallel output mode.

Output Enable Control

When the parallel pixel data interface is enabled, its signals can be switched asynchronously between the driven and High-Z under pin or register control, as shown in Table 9. Selection of a pin to use for the OE_N function is described in ”General Purpose Inputs”.

Table 9. OUTPUT ENABLE CONTROL

OE_N Pin	Drive Signals R0x301A–B[6]	Description
Disabled	0	Interface High-Z
Disabled	1	Interface Driven
1	0	Interface High-Z
X	1	Interface Driven
0	X	Interface Driven

Configuration of the Pixel Data Interface

Fields in R0x301A are used to configure the operation of the pixel data interface. The supported combinations are shown in Table 10.

Table 10. CONFIGURATION OF THE PIXEL DATA INTERFACE

Serializer Disable R0x301 A–B[12]	Parallel Enable R0x301A–B[7]	Standby End-of-Frame R0x301A–B[4]	Description
0	0	1	Power up default. Serial pixel data interface and its clocks are enabled. Transitions to soft standby are synchronized to the end of frames on the serial pixel data interface
1	1	0	Parallel pixel data interface, sensor core data output. Serial pixel data interface and its clocks disabled to save power. Transitions to soft standby are synchronized to the end of the current row readout on the parallel pixel data interface
1	1	1	Parallel pixel data interface, sensor core data output. Serial pixel data interface and its clocks disabled to save power. Transitions to soft standby are synchronized to the end of frames in the parallel pixel data interface

System States

The system states of the MT9J003 are represented as a state diagram in Figure 24 and described in subsequent sections. The effect of RESET_BAR on the system state and the configuration of the PLL in the different states are shown in Table 11.

The sensor's operation is broken down into three separate states: hardware standby, software standby, and streaming. The transition between these states might take a certain amount of clock cycles as outlined in Table 11.

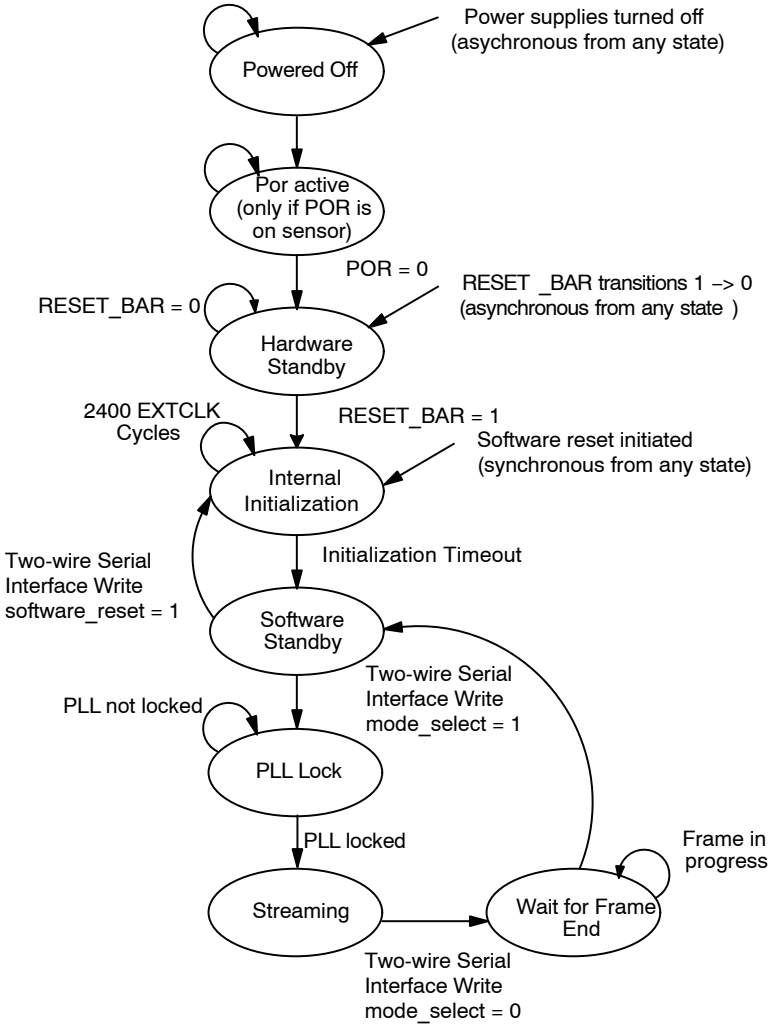


Figure 24. MT9J003 System States

Table 11. RESET_BAR AND PLL IN SYSTEM STATES

State	EXTCLKs	PLL
Powered Off	–	VCO Powered Down (Note 1)
POR Active	–	
Hardware Standby	0	
Internal Initialization	1	
Software Standby		
PLL Lock		VCO Powering Up and Locking, PLL Output Bypassed
Streaming		VCO Running, PLL Output Active
Wait for Frame End		

1. VCO = voltage-controlled oscillator.

Power-On Reset Sequence

When power is applied to the MT9J003, it enters a low-power hardware standby state. Exit from this state is controlled by the later of two events:

1. The negation of the RESET_BAR input.
2. A timeout of the internal power-on reset circuit.

It is possible to hold RESET_BAR permanently de-asserted and rely upon the internal power-on reset circuit.

When RESET_BAR is asserted it asynchronously resets the sensor, truncating any frame that is in progress.

When the sensor leaves the hardware standby state it performs an internal initialization sequence that takes 2400 EXTCLK cycles. After this, it enters a low-power software standby state. While the initialization sequence is in progress, the MT9J003 will not respond to READ transactions on its two-wire serial interface. Therefore, a method to determine when the initialization sequence has completed is to poll a sensor register; for example, R0x0000. While the initialization sequence is in progress, the sensor will not respond to its device address and READs from the sensor will result in a NACK on the two-wire serial interface bus. When the sequence has completed, READs will return the operational value for the register (0x2800 if R0x0000 is read).

When the sensor leaves software standby mode and enables the VCO, an internal delay will keep the PLL disconnected for up to 1ms so that the PLL can lock. The VCO lock time is 200 µs(typical), 1 ms (maximum).

Soft Reset Sequence

The MT9J003 can be reset under software control by writing “1” to software_reset (R0x0103). A software reset asynchronously resets the sensor, truncating any frame that is in progress. The sensor starts the internal initialization sequence, while the PLL and analog blocks are turned off. At this point, the behavior is exactly the same as for the power-on reset sequence.

Signal State During Reset

Table 12 shows the state of the signal interface during hardware standby (RESET_BAR asserted) and the default state during software standby. After exit from hardware standby and before any registers within the sensor have been changed from their default power-up values.

Table 12. SIGNAL STATE DURING RESET

Pad Name	Pad Type	Hardware Standby	Software Standby
EXTCLK	Input	Enabled. Must be driven to a valid logic level	
RESET_BAR (XSHUTDOWN)	Input	Enabled. Must be driven to a valid logic level	
GPI[3:0]		Powered down. Can be left disconnected/floating	
TEST		Enabled. Must be driven to a logic 0 for a serial CCP2-configured sensor, or 1 for a serial MIPI-configured sensor	
SCLK		Enabled. Must be pulled up or driven to a valid logic level	
SDATA	I/O	Enabled as an input. Must be pulled up or driven to a valid logic level	
LINE_VALID	Output	High-Z. Can be left disconnected or floating	
FRAME_VALID			
DOUT[11:0]			
PIXCLK			
SLVS0_P			
SLVS0_N			
SLVS1_P			
SLVS1_N			
SLVS2_P			
SLVS2_N			
SLVS3_P			
SLVS3_N			
CLK_P			
CLK_N			
FLASH		High-Z.	Logic 0.
SHUTTER			

General Purpose Inputs

The MT9J003 provides four general purpose inputs. After reset, the input pads associated with these signals are powered down by default, allowing the pads to be left disconnected/floating.

The general purpose inputs are enabled by setting reset_register[8] (R0x301A). Once enabled, all four inputs must be driven to valid logic levels by external signals. The state of the general purpose inputs can be read through gpi_status[3:0] (R0x3026).

In addition, each of the following functions can be associated with none, one, or more of the general purpose inputs so that the function can be directly controlled by a hardware input:

- Output enable (see “Output Enable Control”)
- Trigger (see the sections below)
- Standby functions
- SADDR selection (see “Serial Register Interface”)

The gpi_status register is used to associate a function with a general purpose input.

Streaming/Standby Control

The MT9J003 can be switched between its soft standby and streaming states under pin or register control, as shown in Table 13. Selection of a pin to use for the STANDBY function is described in “General Purpose Inputs”. The state diagram for transitions between soft standby and streaming states is shown in Figure 24.

Table 13. STREAMING/STANDBY

Standby	Streaming R0x301A–B[2]	Description
Disabled	0	Soft Standby
Disabled	1	Streaming
–	0	Soft Standby
0	1	Streaming
1	–	Soft Standby

Trigger Control

When the global reset feature is in use, the trigger for the sequence can be initiated either under pin or register control, as shown in Table 14. Selection of a pin to use for the TRIGGER function is described in “General Purpose Inputs”.

Table 14. TRIGGER CONTROL

Trigger	Global Trigger R0x3160–1[0]	Description
Disabled	0	Idle
Disabled	1	Trigger
0	0	Idle
–	1	Trigger
1	–	Trigger