Aptina Confidential and Proprietary

1/2.3-Inch 9Mp CMOS Digital Image Sensor

MT9N001 Data Sheet

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

Features

- DigitalClarity® CMOS imaging technology
- 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 CCP2-compliant sub-low-voltage differential signalling (sub-LVDS) or one/two lane serial mobile industry processor interface (MIPI)
- On-die phase-locked loop (PLL) oscillator
- Bayer pattern down-size scaler
- Integrated position-based color and lens shading correction
- One-time programmable (OTP) memory for storing module information

Applications

- Cellular phones
- Digital still cameras
- PC cameras
- PDAs

General Description

The Aptina MT9N001 is a 1/2.3-inch CMOS active-pixel digital image sensor with a pixel array of 3488H x 2616V including border pixels. 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.

Table 1: Key Performance Parameters

Parameter		Value
Optical format		1/2.3-inch (4:3)
Active imager size		6.104mm(H) x 4.578mm(V) 7.63mm diagonal
Active pixels		3488H x 2616V
Pixel size		1.75 x 1.75µm
Chief ray angle		0°
Color filter array		RGB Bayer pattern
Shutter type		Electronic rolling shutter (ERS) with global reset release (GRR)
Input clock frequency		6–48 MHz
Maximum data rate	Parallel	96 Mp/s at 96 MHz PIXCLK
	CCP2	640 Mbps
	MIPI (two-lane)	1.536 Gbps
Frame rate	Full resolution	Programmable up to 13.2 fps serial, 9.7 fps parallel
	VGA	30 fps VGA binning with 8Mp field-of-view
ADC resolution		12-bit, on-die
Responsivity		0.44 V/lux-sec (550nm)
Dynamic range		65dB
SNR_{MAX}		35dB
Supply voltage	I/O Digital	1.7–1.9V (1.8V nominal) or 2.4–3.1V (2.8V nominal)
	Digital	1.7–1.9V (1.8V nominal)
	Analog	2.6–3.1V (2.8V nominal)
Power Consumption	Full resolution	500mW
	Preview	210mW low power VGA
	Standby	500µW (typical, EXTCLK disabled)
Package		48-pin iLCC (10mm x 10mm) Bare die
Operating temperature		–30°C to +70°C (at junction)

Ordering Information

Table 2: Available Part Numbers

Part Number	Description
MT9N001D00STCC2BBC1	Bare die
MT9N001I12STC	48-pin iLCC

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation All rights reserved.

Products and specifications discussed herein are subject to change by Aptina without notice.

Aptina Confidential and Proprietary

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Table of Contents

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

List of Figures

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

List of Tables

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

General Description

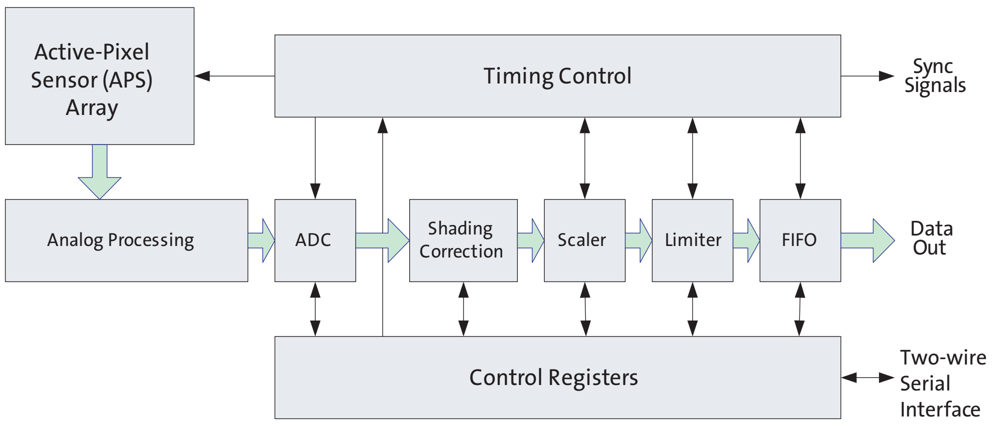
The MT9N001 digital image sensor features DigitalClarity—Aptina's breakthrough low-noise CMOS imaging technology that achieves near-CCD image quality (based on signal-to-noise ratio and low-light sensitivity) while maintaining the inherent size, cost, and integration advantages of CMOS.

When operated in its default mode, the sensor generates a full resolution image at 13.2 frames per second (fps). An on-chip analog-to-digital converter (ADC) generates a 12-bit value for each pixel.

Functional Overview

The MT9N001 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 96 Mp/s, corresponding to a pixel clock rate of 96 MHz. A block diagram of the sensor is shown in Figure 1.

Figure 1: Block Diagram



The core of the sensor is a 9Mp 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.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

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 on page 7 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.
- 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.

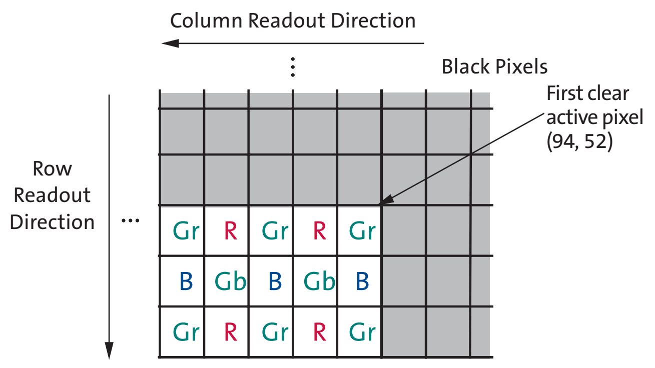
The output FIFO is present to prevent data bursts by keeping the data rate continuous. Programmable slew rates are also available to reduce the effect of electromagnetic interference from the output interface.

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: Pixel Color Pattern Detail (Top Right Corner)



Aptina Confidential and Proprietary

Operating Modes

By default, the MT9N001 powers up with the serial pixel data interface enabled. The sensor can operate either in serial CPP2 or serial MIPI mode, and this mode is preconfigured at the factory. In both cases, the sensor has an SMIA-compatible register interface while the I^{2}C device address is compliant with SMIA or MIPI requirements as appropriate. The reset level on the TEST pin must be tied in a way that is compatible with the configured serial interface of the sensor, for instance TEST = 0 for CCP2 and TEST = 1 for MIPI.

The MT9N001 also supports parallel data out in both CCP2 and MIPI configuration. Typical configurations are shown in Figure 4 on page 11, Figure 5 on page 12, Figure 3 on page 10, and Figure 7 on page 15. These operating modes are described in "Programming Restrictions" on page 24.

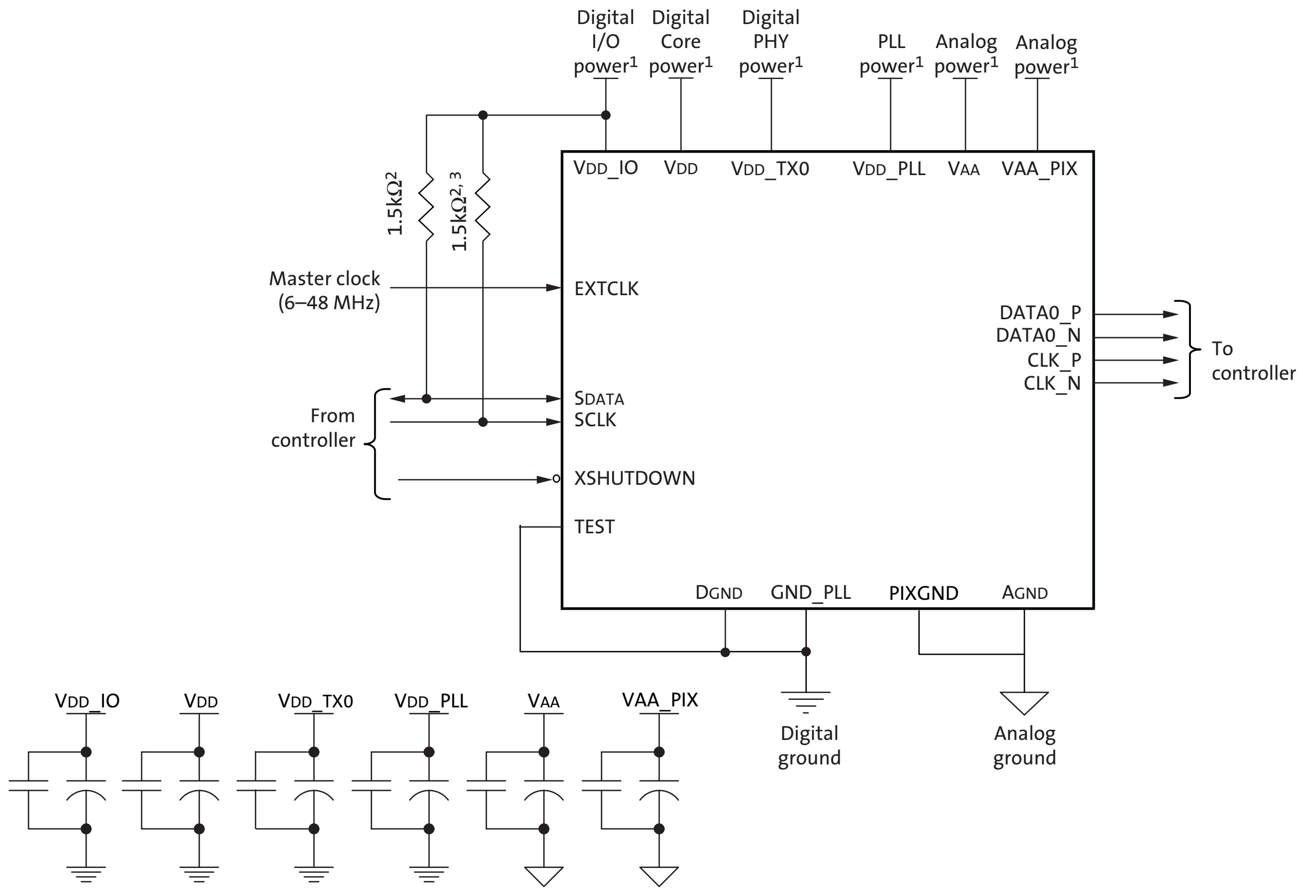
For low-noise operation, the MT9N001 requires separate power supplies for analog and digital. Incoming digital and analog ground conductions should be placed in such a way that coupling between the two are minimized. Both power supply rails should also be routed in such a way that noise coupling between the two supplies and ground is minimized.

Caution **Aptina does not recommend the use of inductance filters on the power supplies or output signals.**

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Figure 3: Typical Configuration: Serial CCP2 Pixel Data Interface

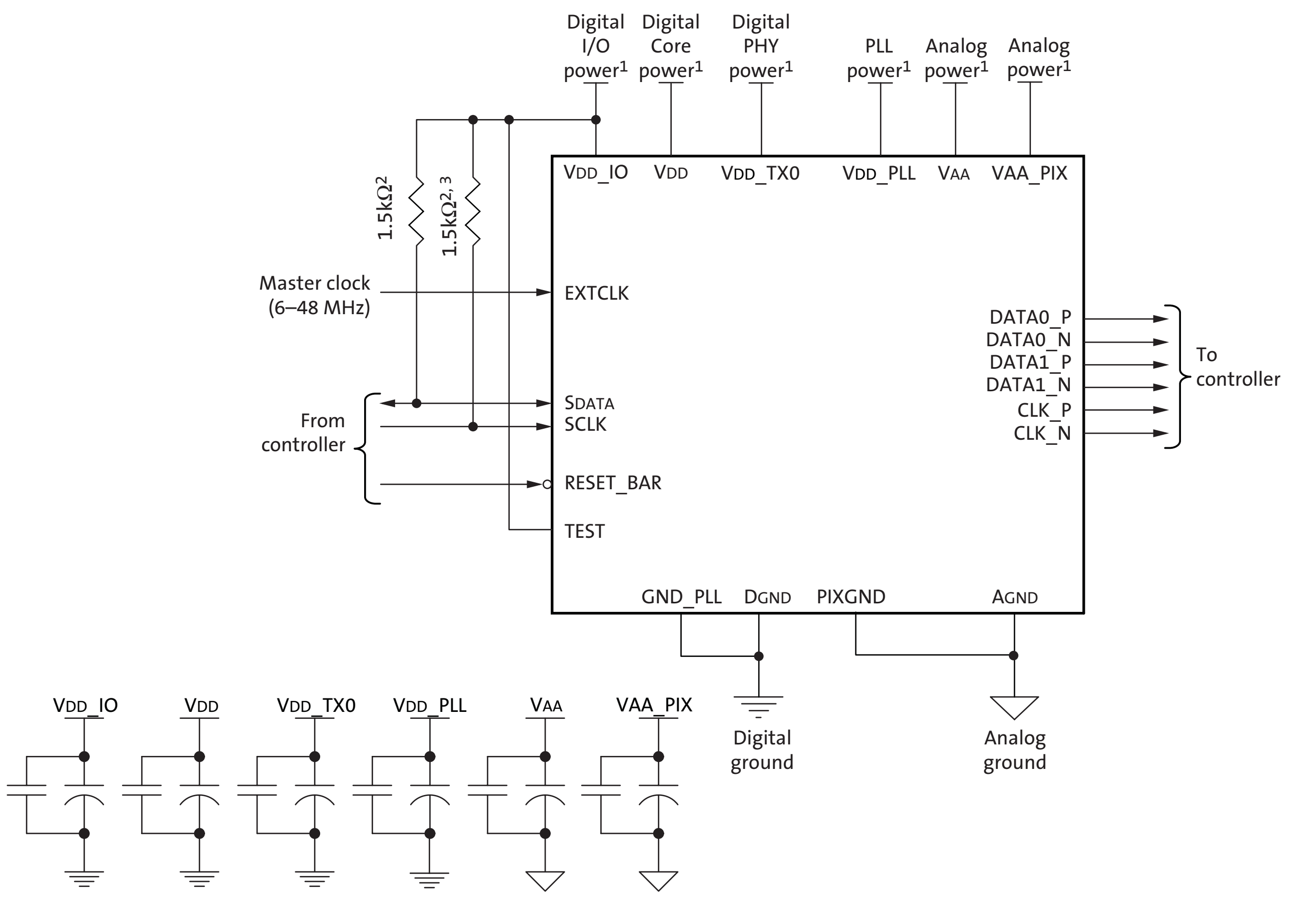


Notes:
1. All power supplies should be adequately decoupled.
2. Aptina recommends a resistor value of 1.5kΩ, but it may be greater for slower two-wire speed.
3. This pull-up resistor is not required if the controller drives a valid logic level on SCLK at all times.
4. VPP, which can be used during the module manufacturing process, is not shown in Figure 3. This pad is left unconnected during normal operation.
5. The parallel interface output pads can be left unconnected if the serial output interface is used.
6. Aptina recommends that 0.1µF and 10µF 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.
7. Aptina recommends that VDD_TX0 be tied to VDD.
8. Aptina recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

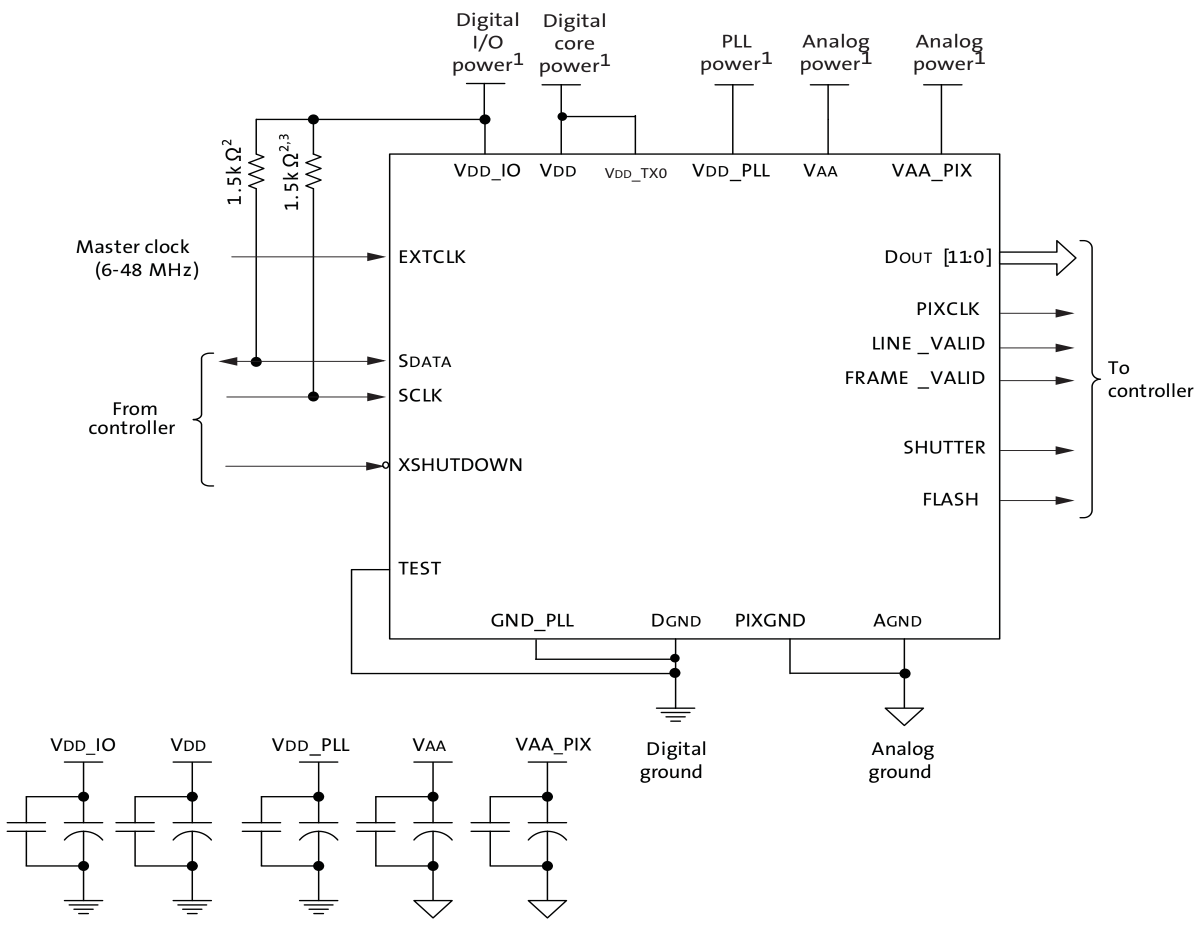
Figure 4: Typical Configuration: Serial Two-Lane MIPI Pixel Data Interface



Notes:
1. All power supplies should be adequately decoupled.
2. Aptina recommends a resistor value of 1.5kΩ, 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. VPP, which can be used during the module manufacturing process, is not shown in Figure 4. This pad is left unconnected during normal operation.
5. The parallel interface output pads can be left unconnected if the serial output interface is used.
6. Aptina recommends that 0.1µF and 10µF 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.
7. Aptina recommends that VDD_TX0 be tied to VDD.
8. Aptina recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

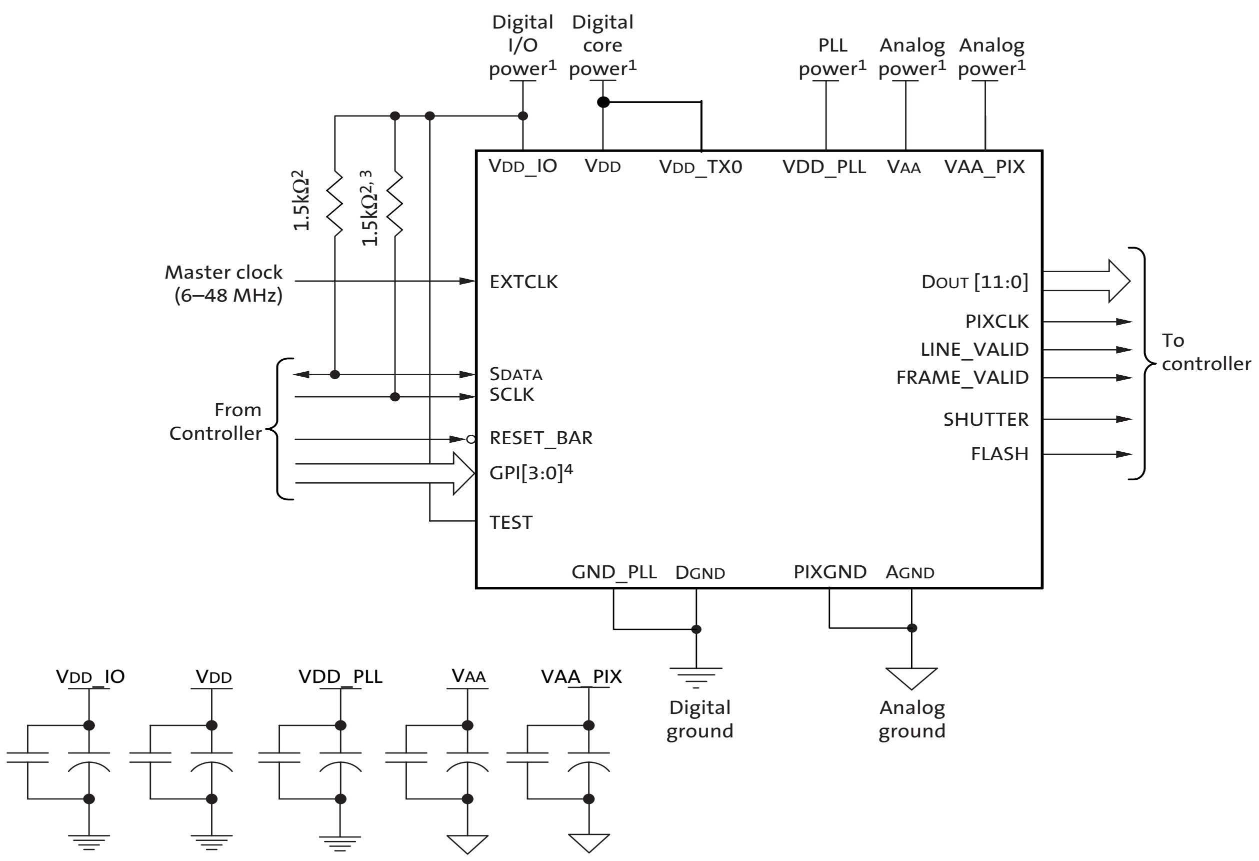
Figure 5: Typical Configuration: CCP2 Parallel Pixel Data Interface



Notes:
1. All power supplies should be adequately decoupled.
2. Aptina recommends a resistor value of 1.5kΩ, 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 5. 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. Aptina recommends that 0.1µF and 10µF 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.
8. Aptina recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.
9. Aptina recommends that VDD_TX0 is tied to VDD when the sensor is using the parallel interface.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Figure 6: Typical Configuration: Parallel MIPI Pixel Data Interface (MIPI)



Notes:
1. All power supplies should be adequately decoupled.
2. Aptina recommends a resistor value of 1.5kΩ, 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 7. 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. Aptina recommends that 0.1µF and 10µF 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.
8. Aptina recommends that analog power planes are placed in a manner such that coupling with the digital power planes is minimized.
9. Aptina recommends that VDD_TX0 is tied to VDD when the sensor is using the parallel interface.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Signal Descriptions

Table 3 provides signal descriptions for MT9N001 die. For pad location and aperture information, refer to the MT9N001 die data sheet.

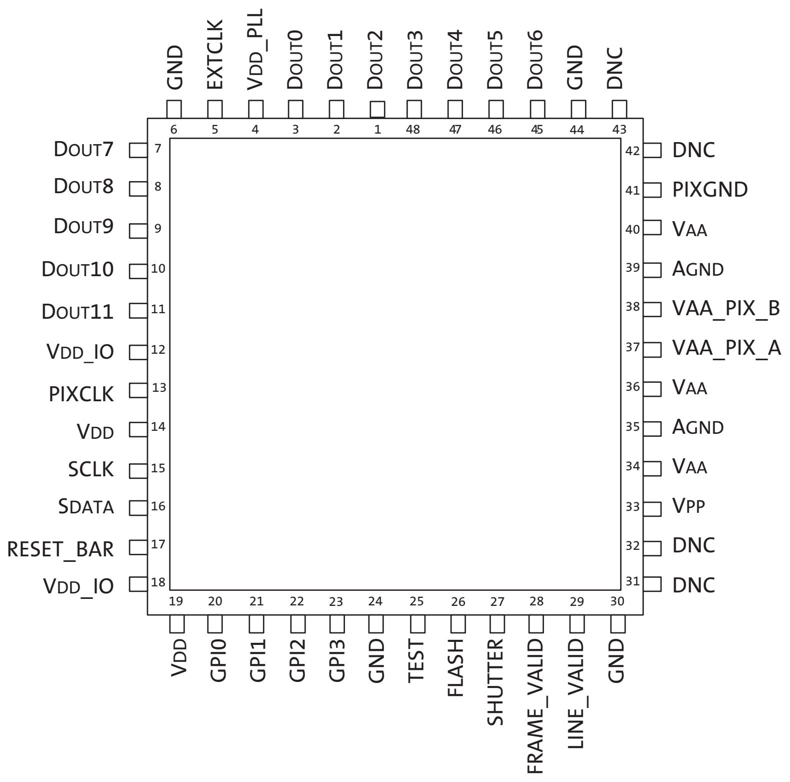
Table 3: Signal Descriptions

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. Connect to DGND for normal operation of the CCP2 configured sensor, or connect to VDD_IO power for the MIPI-configured sensor.
SDATA	I/O	Serial data from READs and WRITEs to control and status registers.
DATA0_P	Output	Differential CCP2/MIPI (sub-LVDS) serial data (positive).
DATA0_N	Output	Differential CCP2/MIPI (sub-LVDS) serial data (negative).
DATA1_P	Output	Differential MIPI (sub-LVDS) serial data 2nd lane (positive). Can be left floating when using one-lane MIPI or CCP2 serial interface.
DATA1_N	Output	Differential MIPI (sub-LVDS) serial data 2nd lane (negative). Can be left floating when using one-lane MIPI or CCP2 serial interface.
CLK_P	Output	Differential CCP2/MIPI (sub-LVDS) serial clock/strobe (positive).
CLK_N	Output	Differential CCP2/MIPI (sub-LVDS) serial clock/strobe (negative).
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 serial data interface. Aptina recommends that VDD_TX0 is tied to VDD when the sensor is used in parallel mode.
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.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Figure 7: 48-Pin iLCC Package Pinout Diagram



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Output Data Format

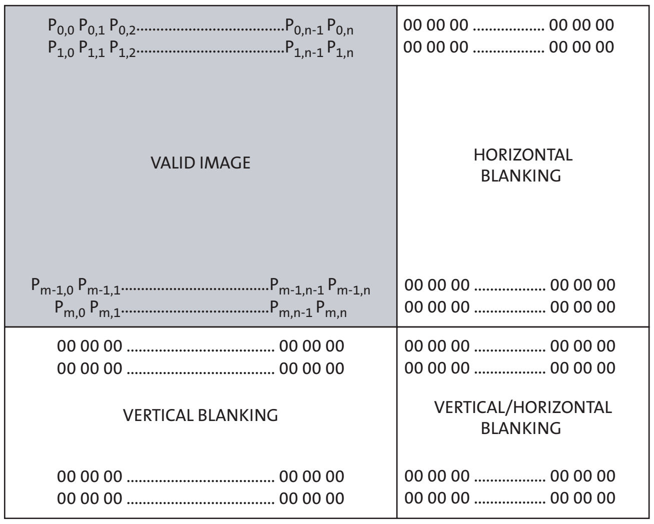
Serial Pixel Data Interface

The MT9N001 supports RAW8, RAW10, and RAW12 image data formats.

Parallel Pixel Data Interface

MT9N001 image data is read out in a progressive scan. Valid image data is surrounded by horizontal blanking and vertical blanking, as shown in Figure 8. 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 8: Spatial Illustration of Image Readout


$P_{0,0}$ $P_{0,1}$ $P_{0,2}$.....$P_{0,n-1}$ $P_{0,n}$
$P_{1,0}$ $P_{1,1}$ $P_{1,2}$.....$P_{1,n-1}$ $P_{1,n}$
VALID IMAGE
$P_{m-1,0}$ $P_{m-1,1}$.....$P_{m-1,n-1}$ $P_{m-1,n}$
$P_{m,0}$ $P_{m,1}$.....$P_{m,n-1}$ $P_{m,n}$
00 00 00 ..... 00 00 00
HORIZONTAL BLANKING
VERTICAL BLANKING
VERTICAL/HORIZONTAL BLANKING

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Output Data Timing (Parallel Pixel Data Interface)

MT9N001 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 9). This allows PIXCLK to be used as a clock to sample the data. PIXCLK is continuously enabled, even during the blanking period. The MT9N001 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 10 are defined in Table 4 on page 18.

Figure 9: Pixel Data Timing Example

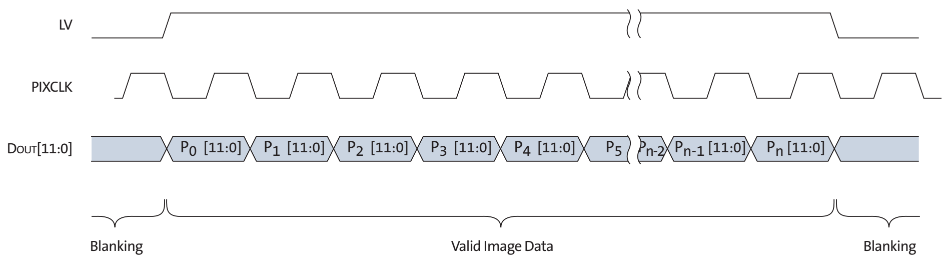
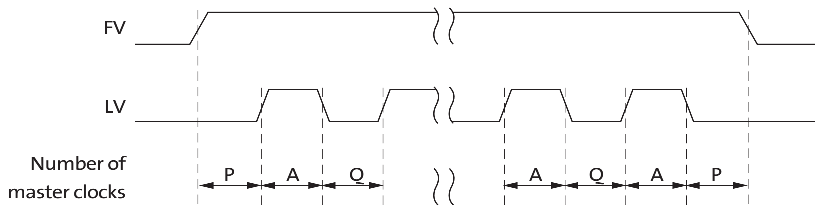


Figure 10: Row Timing and FV/LV Signals



The sensor timing (shown in Table 4) is shown in terms of pixel clock and master clock cycles (see Figure 9 on page 17). The default settings for the on-chip PLL generate 96–192 MHz output pixel clock (op_pix_clk) given a 24 MHz input clock to the MT9N001. Equations for calculating the frame rate are given in “Frame Rate Control” on page 55.

Table 4: Row Timing

Parameter	Name	Equation	Default Timing
PIXCLK_PERIOD	Pixel clock period	R0x3016–7[2:0] / vt_pix_clk_freq_mhz (vt_pix_clk set to 192 MHz)	1 pixel clock = 5.2ns
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) * PIXCLK_PERIOD/S	3488 pixel clocks = 18.166µs
P	Frame start/end blanking	6 * PIXCLK_PERIOD	6 pixel clocks = 31.25ns
Q	Horizontal blanking	(line_length_pck – A) * PIXCLK_PERIOD	1156 pixel clocks = 6.02µs
A + Q	Row time	line_length_pck * PIXCLK_PERIOD	7006 pixel clocks = 36.49µs
N	Number of rows	(y_addr_end - y_addr_start + y_odd_inc) / S	2616 rows
V	Vertical blanking	((frame_length_lines - N) * (A + Q)) + Q – (2 * P)	596654 pixel clocks = 3.11ms
T	Frame valid time	(N * (A + Q)) - Q + (2 * P)	18326552 pixel clocks = 95.45ms
F	Total frame time	line_length_pck * frame_length_lines * PIXCLK_PERIOD	1891240 pixel clocks = 98.50ms

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Two-Wire Serial Register Interface

The two-wire serial interface bus enables read/write access to control and status registers within the MT9N001. This interface is designed to be compatible with the electrical characteristics and transfer protocols of the I^2C specification.

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 1.5kΩ 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 SCLK LOW; the MT9N001 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/Data Direction Byte

Bits [7:1] of this byte represent the device slave address and bit [0] indicates the data transfer direction. A “0” in bit [0] indicates a WRITE, and a “1” indicates a READ. The default slave addresses used by the MT9N001 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

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

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, 8 bits at a time. The master generates an acknowledge bit after each 8-bit transfer. The slave’s internal register address is automatically incremented after every 8 bits are transferred. The data transfer is stopped when the master sends a no-acknowledge bit.

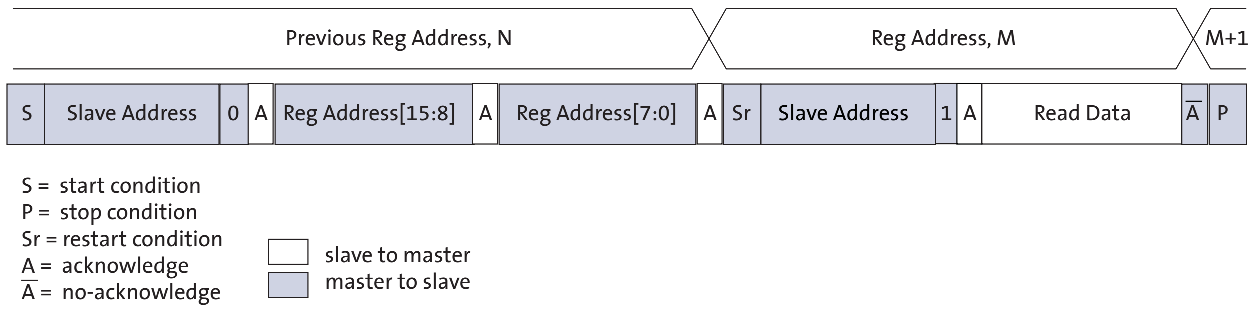
Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Single READ from Random Location

This sequence (Figure 11 on page 21) 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 11 shows how the internal register address maintained by the MT9N001 is loaded and incremented as the sequence proceeds.

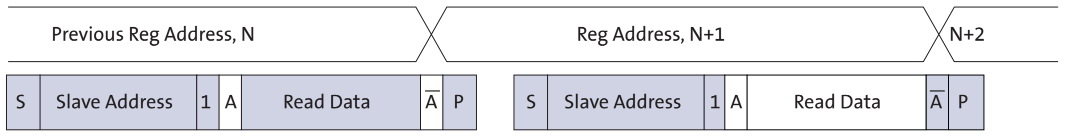
Figure 11: Single READ from Random Location



Single READ from Current Location

This sequence (Figure 12) performs a read using the current value of the MT9N001 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 12: Single READ from Current Location

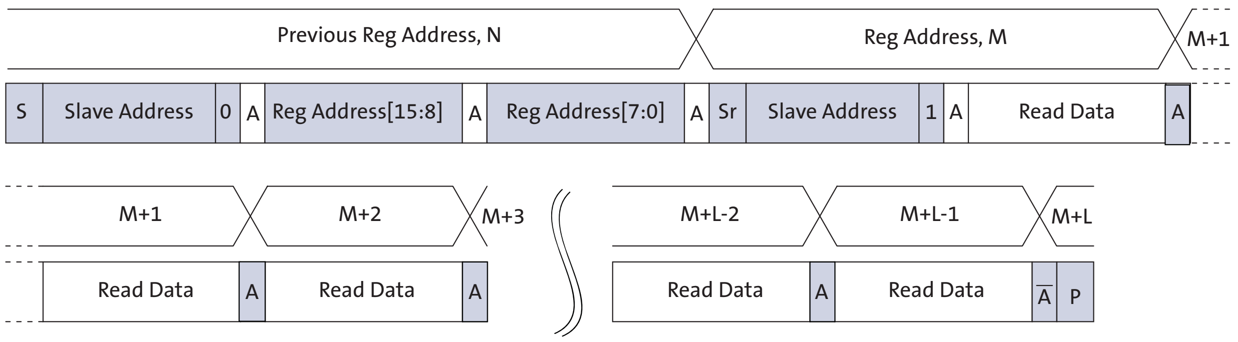


Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Sequential READ, Start from Random Location

This sequence (Figure 13) starts in the same way as the single READ from random location (Figure 11). 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.

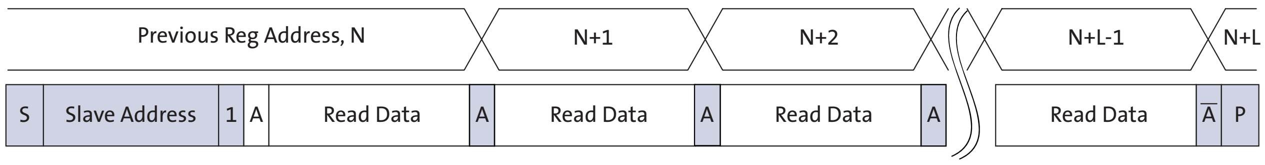
Figure 13: **Sequential READ, Start from Random Location**



Sequential READ, Start from Current Location

This sequence (Figure 14) starts in the same way as the single READ from current location (Figure 12 on page 21). 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.

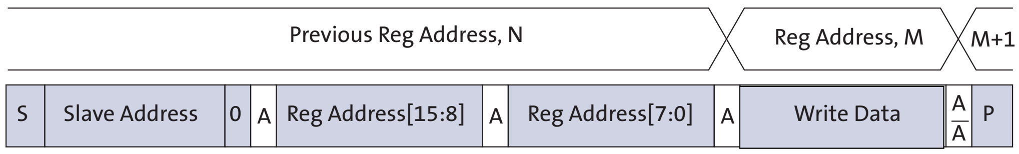
Figure 14: **Sequential READ, Start from Current Location**



Single WRITE to Random Location

This sequence (Figure 15) 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 15: **Single WRITE to Random Location**



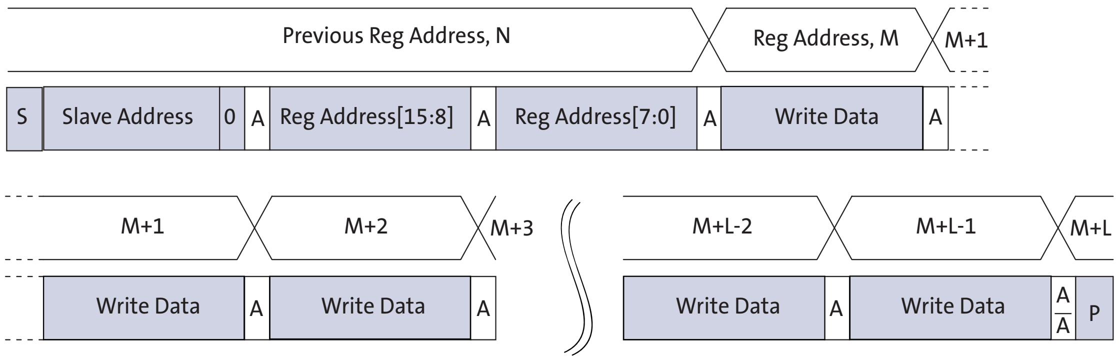
Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Sequential WRITE, Start at Random Location

This sequence (Figure 16) starts in the same way as the single WRITE to random location (Figure 15). 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 16: Sequential WRITE, Start at Random Location



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Programming Restrictions

Table 6 shows a list of programming rules that must be adhered to for correct operation of the MT9N001. It is recommended that these rules are encoded into the device driver stack—either implicitly or explicitly.

Table 5: 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

Table 6: Programming Rules

Parameter	Minimum Value	Maximum Value
coarse_integration_time	coarse_integration_time_min	frame_length_lines - coarse_integration_time_max_margin
fine_integration_time	fine_integration_time_min	line_length_pck - fine_integration_time_max_margin
digital_gain_*	digital_gain_min	digital_gain_max
digital_gain_* is an integer multiple of digital_gain_step_size		
frame_length_lines	min_frame_length_lines	max_frame_length_lines
line_length_pck	min_line_length_pck	max_line_length_pck
	((x_addr_end - x_addr_start + x_odd_inc)/xskip) + min_line_blanking_pck	
frame_length_lines	((y_addr_end - y_addr_start + y_odd_inc)/yskip) + min_frame_blanking_lines	
x_addr_start	x_addr_min	x_addr_max
x_addr_end	x_addr_start	x_addr_max
(x_addr_end - x_addr_start+ x_odd_inc)	must be positive	must be positive
x_addr_start[0]	0	0
x_addr_end[0]	1	1
y_addr_start	y_addr_min	y_addr_max
y_addr_end	y_addr_start	y_addr_max
(y_addr_end - y_addr_start + y_odd_inc)/	must be positive	must be positive
y_addr_start[0]	0	0
y_addr_end[0]	1	1
x_even_inc	min_even_inc	max_even_inc
x_even_inc[0]	1	1
y_even_inc	min_even_inc	max_even_inc
y_even_inc[0]	1	1
x_odd_inc	min_odd_inc	max_odd_inc
x_odd_inc[0]	1	1
y_odd_inc	min_odd_inc	max_odd_inc
y_odd_inc[0]	1	1
scale_m	scaler_m_min	scaler_m_max
scale_n	scaler_n_min	scaler_n_max

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Table 6: Programming Rules (Continued)

Parameter	Minimum Value	Maximum Value
x_output_size	256	2608
x_output_size[0]	0 (this is enforced in hardware: bit[0] is read-only)	0
y_output_size	2	frame_length_lines
y_output_size[0]	0 (this is enforced in hardware: bit[0] is read-only)	0
With subsampling, start and end pixels must be addressed (impact on x/y start/end addresses, function of image orientation bits)		

Output Size Restrictions

The design specification imposes the restriction that an output line shall be a multiple of 32 bits in length. This imposes an additional restriction on the legal values of x_output_size:

- When ccp_data_format[7:0] = 8 (RAW8 data), x_output_size must be a multiple of 4 (x_output_size[1:0] = 0).
- When ccp_data_format[7:0] = 10 (RAW10 data), x_output_size must be a multiple of 16 (x_output_size[3:0] = 0).
- When ccp_data_format[7:0] = 12 (RAW12 data), x_output_size must be a multiple of 8 (x_output_size[3:0] = 0).

This restriction only applies when the serial pixel data path is in use. It can be met by rounding up x_output_size to an appropriate multiple. Any extra pixels in the output image as a result of this rounding contain undefined pixel data but are guaranteed not to cause false synchronization on the serial data stream.

When the parallel pixel data path is in use, the only restriction on x_output_size is that it must be even (x_output_size[0] = 0), and this restriction is enforced in hardware.

When the serial pixel data path is in use, there is an additional restriction that x_output_size must be small enough such that the output row time (set by x_output_size, the framing and CRC overhead of 12 bytes and the output clock rate) must be less than the row time of the video array (set by line_length_pck and the video timing clock rate).

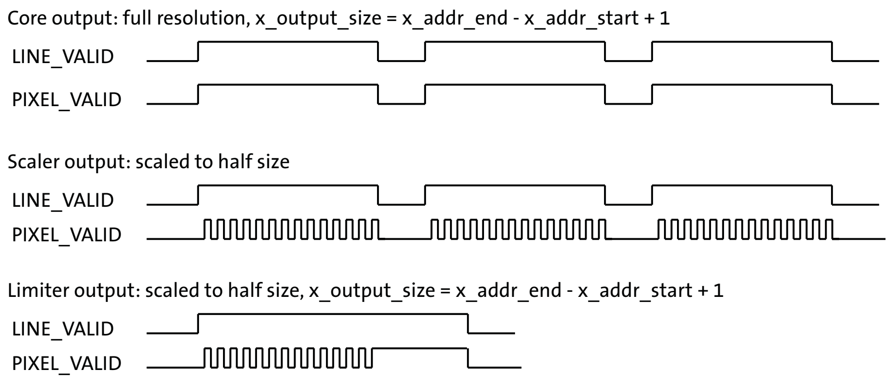
Effect of Scaler on Legal Range of Output Sizes

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 MT9N001 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 17 on page 26.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Figure 17: Effect of Limiter on the Data Path


Core output: full resolution, x_output_size = x_addr_end - x_addr_start + 1
LINE_VALID
PIXEL_VALID
Scaler output: scaled to half size
LINE_VALID
PIXEL_VALID
Limiter output: scaled to half size, x_output_size = x_addr_end - x_addr_start + 1
LINE_VALID
PIXEL_VALID

In Figure 17, three different stages in the data path (see “Digital Data Path” on page 70) 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 signalled by transitions in PIXEL_VALID. Overall, PIXEL_VALID is asserted for $(N/2)$ pixel times per row.

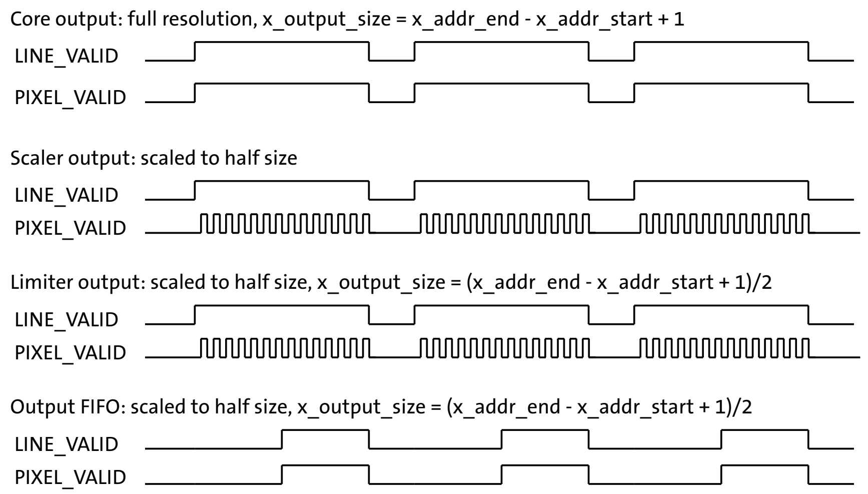
The third stage is the output of the limiter when the x_output_size is still set to match the active array size. Because the scaler has reduced the amount of valid pixel data without reducing the row time, the limiter attempts to pad the row with $(N/2)$ additional pixels. If this has the effect of extending LV across the whole of the horizontal blanking time, the MT9N001 will cease to generate output frames.

A correct configuration is shown in Figure 18 on page 27, 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 18 on page 27 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.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Figure 18: **Timing of Data Path**


Core output: full resolution, x_output_size = x_addr_end - x_addr_start + 1
LINE_VALID
PIXEL_VALID
Scaler output: scaled to half size
LINE_VALID
PIXEL_VALID
Limiter output: scaled to half size, x_output_size = (x_addr_end - x_addr_start + 1)/2
LINE_VALID
PIXEL_VALID
Output FIFO: scaled to half size, x_output_size = (x_addr_end - x_addr_start + 1)/2
LINE_VALID
PIXEL_VALID

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).**

Aptina Confidential and Proprietary

Changing Registers While Streaming

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

- ccp_channel_identifier
- ccp_data_format
- ccp_signaling_mode
- 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" on page 60.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

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" on page 19.

Default Power-Up State

The MT9N001 provides interfaces for pixel data through the CCP2 high speed serial interface described by the SMIA specification or the MIPI serial interface and a parallel data interface.

At power-up and after a hard or soft reset, the reset state of the MT9N001 is to enable the SMIA CCP2 high speed serial interface for a CCP2-configured sensor, and CSI-2 high speed serial interface for a MIPI-configured sensor.

The CCP2 and MIPI serial interfaces share pins, and only one can be enabled at a time. This is done at the factory.

Serial Pixel Data Interface

The serial pixel data interface uses these output-only signal pairs:

- DATA0_P
- DATA0_N
- DATA1_P
- DATA1_N
- CLK_P
- CLK_N

The signal pairs are driven differentially using sub-LVDS switching levels. The serial pixel data interface is enabled by default at power up and after reset. DATA0_P and DATA0_N are the data pair for the CCP2 or one-lane MIPI serial interface, while DATA1_P and DATA1_N are the second data pair for two-lane serial MIPI.

The DATA0_P, DATA0_N, DATA1_P, DATA1_N, CLK_P, and CLK_N pads are turned off if the SMIA serial disable bit is asserted (R0x301A–B[12] = 1) or when the sensor is in the soft standby state.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

In data/clock mode, the clock remains HIGH when no data is being transmitted. In data/strobe mode before frame start, clock is LOW and data is HIGH.

When the serial pixel data interface is used, the LV, FV, PIXCLK, and DOUT[11:0] signals can be left unconnected.

R0x0112-3 (ccp_data_format) The following data formats are setting supported:

- 0x0A0A – sensor supports RAW10 uncompressed data format. A sensor with a 12-bit ADC can support this mode by discarding all but the upper 10 bits of a pixel value.
- 0x0C0C – sensor supports RAW12 uncompressed data format
- 0x0808 – sensor supports RAW8 uncompressed data format. A sensor with a 12-bit or 10-bit ADC can support this mode by discarding all but the upper 8 bits of a pixel value.
- 0x0A08 – sensor supports RAW8 data format in which an adaptive compression algorithm is used to perform 10-bit to 8-bit compression on the upper 10 bits of each pixel value.
- 0x0C08 – sensor supports RAW8 data format in which an adaptive compression algorithm is used to perform 12-bit to 8-bit compression on the upper 12 bits of each pixel value.

Also, the ccp_serial_format register (R0x31AE) register controls which serial interface is in use when the serial interface is enabled (reset_register[12] = 0). The following serial formats are supported:

- 0x0101 – sensor supports one-lane CCP2 operation.
- 0x0201 – sensor supports one-lane MIPI operation.
- 0x0202 – sensor supports two-lane MIPI operation.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

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 8 shows the recommended settings.

When the parallel pixel data interface is in use, the DATA0_P, DATA0_N, DATA1_P, DATA1_N, CLK_P, and CLK_N 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 7. Selection of a pin to use for the OE_N function is described in "General Purpose Inputs" on page 35.

Table 7: 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 8.

Table 8: 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.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

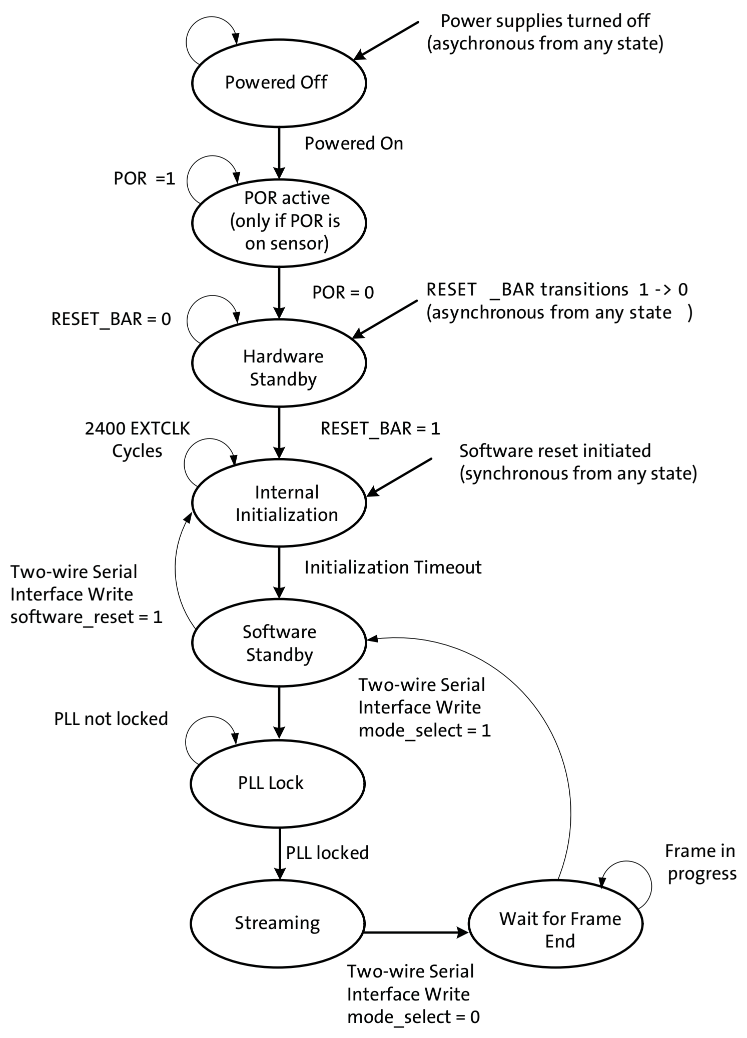
Aptina Confidential and Proprietary

System States

The system states of the MT9N001 are represented as a state diagram in Figure 19 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 9 on page 33.

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 9.

Figure 19: MT9N001 System States



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Table 9: RESET_BAR and PLL in System States

State	EXTCLKs	PLL
Powered off	x	VCO powered down
POR active	x	
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		

Notes: 1. VCO = voltage-controlled oscillator.

Power-On Reset Sequence

When power is applied to the MT9N001, 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. The RESET_BAR signal is functionally equivalent to the SMIA-specified XSHUTDOWN signal.

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

While RESET_BAR is asserted (or the internal power-on reset circuit is active) the MT9N001 is in its lowest-powered, powered-up state; the internal PLL is disabled, the CCP2 serializer is disabled and internal clocks are gated off.

When the sensor leaves the hardware standby state it performs an internal initialization sequence that takes 2400 EXTCLK cycles (tentative). After this, it enters a low-power software standby state. While the initialization sequence is in progress, the MT9N001 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 200µs(typical), 1ms (maximum).

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Soft Reset Sequence

The MT9N001 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 10 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 10: 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.	
LINE_VALID	Output	High-Z. Can be left disconnected/floating.	
FRAME_VALID	Output		
DOUT[11:0]	Output		
PIXCLK	Output		
SCLK	Input	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.	
FLASH	Output	High-Z.	Logic 0.
SHUTTER	Output	High-Z.	Logic 0.
DATA0_P	Output	CCP2: High Z MIPI: Ultra Low-Power State (ULPS), represented as an LP-00 state on the input (both wires at 0V).	
DATA0_N	Output		
DATA1_P	Output		
DATA1_N	Output		
CLK_P	Output		
CLK_N	Output		
GPI[3:0]	Input	Powered down. Can be left disconnected/floating.	
TEST	Input	Enabled. Must be driven to a logic 0 for a serial CCP2-configured sensor, or 1 for a serial MIPI-configured sensor.	

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

General Purpose Inputs

The MT9N001 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" on page 31)
- Trigger (see the sections below)
- Standby functions
- SADDR selection (see "Serial Register Interface" on page 29)

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

Streaming/Standby Control

The MT9N001 can be switched between its soft standby and streaming states under pin or register control, as shown in Table 11. Selection of a pin to use for the STANDBY function is described in "General Purpose Inputs" on page 35. The state diagram for transitions between soft standby and streaming states is shown in Figure 19 on page 32.

Table 11: Streaming/STANDBY

STANDBY	Streaming R0x301A–B[2]	Description
Disabled	0	Soft standby
Disabled	1	Streaming
X	0	Soft standby
0	1	Streaming
1	X	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 12. Selection of a pin to use for the TRIGGER function is described in "General Purpose Inputs" on page 35.

Table 12: Trigger Control

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

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

PLL

The sensor contains a PLL for timing generation and control. The PLL contains a prescaler to divide the input clock applied on EXTCLK, a VCO to multiply the prescaler output, and a set of dividers to generate the output clocks. The clocking structure is shown in Figure 20.

Figure 20: Clocking Structure

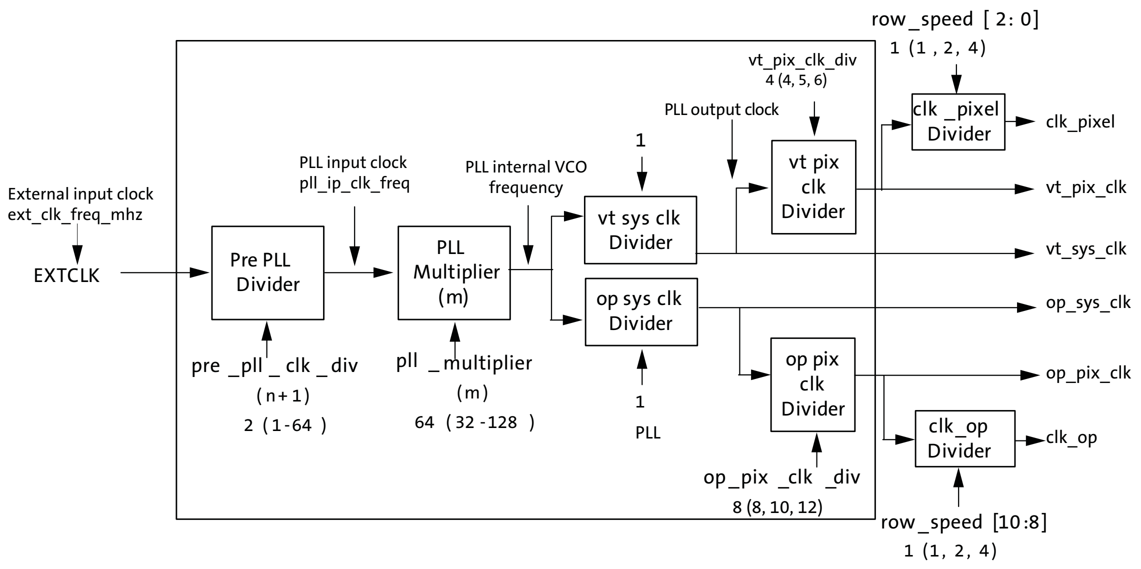


Figure 20 shows the different clocks and the register names. It also shows the default setting for each divider/multiplier control register, and the range of legal values for each divider/multiplier control register. The vt and op sys clk Divider is hardwired in the design.

From the diagram, the clock frequencies can be calculated as follows:

Internal pixel clock used to readout the pixel array:

(EQ 1)

$$clk\_pixel\_freq\_mhz = \frac{ext\_clk\_freq\_mhz \text{ x } pll\_multiplier}{pre\_pll\_clk\_div \text{ x } vt\_pix\_clk\_div \text{ x } row\_speed\ [2:0]} = \frac{24 \text{ MHz x } 64}{2\text{x}4\text{x}1} = 192 \text{ MHz}$$

External pixel clock used to output the data:

(EQ 2)

$$clk\_op\_freq\_mhz = \frac{ext\_clk\_freq\_mhz \text{ x } pll\_multiplier}{pre\_pll\_clk\_div \text{ x } op\_pix\_clk\_div \text{ x } row\_speed\ [10:8]} = \frac{24 \text{ MHz x } 64}{2\text{x}8\text{x}1} = 96 \text{ MHz}$$

Internal master clock:

(EQ 3)

$$vt\_pix\_clk\_freq\_mhz/2$$

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

The parameter limit register space contains registers that declare the minimum and maximum allowable values for:

- The frequency allowable on each clock.
- The divisors that are used to control each clock.

The following factors determine what are valid values, or combinations of valid values, for the divider/multiplier control registers:

- The minimum/maximum frequency limits for the associated clock must be met.

  pll_ip_clk_freq must be in the range 2–24 MHz. Higher frequencies are preferred.
  PLL internal VCO frequency must be in the range 384–768 MHz.

- The minimum/maximum value for the divider/multiplier must be met.

  Range for m: 32–128.
  Range for n: 0–63. Range for (n + 1): 1–64.

- The op_pix_clk must never run faster than the vt_pix_clk to ensure that the output data stream is contiguous.
- Given the maximum programmed line length, the minimum blanking time, the maximum image width, the available PLL divisor/multiplier values, and the requirement that the output line time (including the necessary blanking) must be output in a time equal to or less than the time defined by line_length_pck.

Although the PLL VCO input frequency range is advertised as 6–48 MHz, superior performance is obtained by keeping the VCO input frequency as high as possible.

The usage of the output clocks is shown below:

- clk_pixel is used by the sensor core to control the timing of the pixel array. The sensor core produces one 12-bit pixel each vt_pix_clk period. The line length (line_length_pck) and fine integration time (fine_integration_time) are controlled in increments of the clk_pixel period.
- clk_op is used to load parallel pixel data from the output FIFO. The output FIFO generates one pixel each op_pix_clk period.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Programming the PLL Divisors

The PLL divisors should be programmed while the MT9N001 is in the software standby state. After programming the divisors, it is necessary to wait for the VCO lock time before enabling the PLL. The PLL is enabled by entering the streaming state.

An external timer will need to delay the entrance of the streaming mode by 1 millisecond so that the PLL can lock.

The effect of programming the PLL divisors while the MT9N001 is in the streaming state is undefined.

Influence of ccp_data_format

R0x0112–3 (ccp_data_format) controls whether the pixel data interface will generate 12, 10, or 8 bits per pixel.

When the pixel data interface is generating 8 bits per-pixel, op_pix_clk_div must be programmed with the value 8. When the pixel data interface is generating 10 bits per pixel, op_pix_clk_div must be programmed with the value 10.

Clock Control

The MT9N001 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 when required.

When the MT9N001 enters a low-power state, almost all of the internal clocks are stopped. The only exception is that a small amount of logic is clocked so that the two-wire serial interface continues to respond to READ and WRITE requests.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Features

Scaler

The MT9N001 sensor includes scaling capabilities. This allows the user to generate full field-of-view, low resolution images. Scaling is advantageous because it uses all pixel values to calculate the output image which helps to avoid aliasing. It is also more convenient than binning because the scale factor varies smoothly and the user is not limited to certain ratios of size resolution.

The scaling factor is programmable in 1/16 steps.

$$ScaleFactor = \frac{scale\_n}{scale\_m} = \frac{16}{scale\_m} \quad \text{(EQ 4)}$$

scale_n is fixed at 16.
scale_m is adjustable with R0x0404 (scale_m)
Legal values for m are 16 through 128. The user has the ability to scale from 1:1 (*scale_m* = 16) to 1:8 (*scale_m* = 128).

Shading Correction (SC)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. There are also other factors causing color plane nonuniformity in images captured by image sensors. The cumulative result of all these factors is known as image shading. The MT9N001 has an embedded shading correction module that can be programmed to counter the shading effects on each individual Red, GreenB, GreenR, and Blue color signal.

The Correction Function

Color-dependent solutions are calibrated using the sensor, lens system and an image of an evenly illuminated, featureless gray calibration field. From the resulting image, register values for the color correction function (coefficients) can be derived.

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

$$Pcorrected(row, col) = Psensor(row,col) * f(row,col) \quad \text{(EQ 5)}$$

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

Each function includes a set of color-dependent coefficients defined by registers R0x3600–3726. The function's origin is the center point of the function used in the calculation of the coefficients. Using an origin near the central point of symmetry of the sensor response provides the best results. The center point of the function is determined by ORIGIN_C (R0x3782) and ORIGIN_R (R0x3784) and can be used to counter an offset in the system lens from the center of the sensor array.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

One-Time Programmable (OTP) Memory

The MT9N001 has a two-byte OTP memory that can be utilized during module manufacturing to store specific information about the module. This feature provides system integrators and module manufacturers the ability to label and distinguish various module types based on lens, IR-cut filter, or other properties.

During the programming process, a dedicated pin for high voltage needs to be provided to perform the anti-fusing operation. This voltage (VPP) would need to be 8.5V ±3%. Instantaneous VPP cannot exceed 9V at any time. The completion of the programming process will be communicated by a register through the two-wire serial interface.

Because this programming pin needs to sustain a higher voltage than other input/output pins, having a dedicated high voltage pin (VPP) minimizes the design risk. If the module manufacturing process can probe the sensor at the die or PCB level (that is, supply all the power rails, clocks, two-wire serial interface signals), then this dedicated high voltage pin does not need to be assigned to the module connector pinout. However, if the VPP pin needs to be bonded out as a pin on the module, the trace for VPP needs to carry a maximum of 1mA is needed for programming only. This pin should be left floating once the module is integrated to a design. If the VPP pin does not need to be bonded-out as a pin on the module, it should be left floating inside the module.

The programming of the OTP memory requires the sensor to be fully powered and remain in software standby with its clock input applied. The information will be programmed through the use of the two-wire serial interface, and once the data is written to an internal register, the programming host machine will apply a high voltage to the programming pin, and send a program command to initiate the anti-fusing process. After the sensor has finished programming the OTP memory, a status bit will be set to indicate the end of the programming cycle, and the host machine can poll the setting of the status bit through the two-wire serial interface. Only one programming cycle for the 16-bit word can be performed.

Reading the OTP memory data requires the sensor to be fully powered and operational with its clock input applied. The data can be read through a register from the two-wire serial interface.

The steps below describe the process to program and verify the programmed data in the OTP memory:

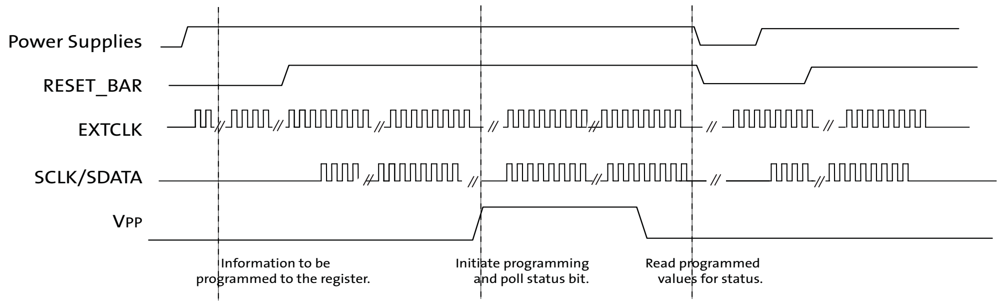
1. Apply power to all the power rails of the sensor (VDD, VDD_IO, VAA, VAA_PIX, VDD_PLL, and VDD_TX0).
   1a. Set VAA to 3.1V during OTP memory programming phase.
   1b. VPP needs to be floated during this phase.
   1c. Other supplies at nominal.
2. Provide 24 MHz EXTCLK clock input. The PLL settings are discussed at the end of the document.
3. Perform the proper reset sequence to the sensor.
4. Place the sensor in soft standby (sensor default state upon power-up) or ensure the streaming is turned OFF when the part is in active mode.
5. VPP ramps to 8.5V in preparation to program. Power supply (VPP) slew rate should be slower than 1V/µs.
6. Set R0x3052 to the value 0x045C.
7. Set R0x3054 to the value 0XEA99.
8. Write the 16-bit word data by programming R0x304C.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

9. Initiate the OTP memory programming process by setting R0x304A[0] to the value 0x0001.
10. Check R0x304A [2] = 1, until bit is set to "1" to check for program completion.
11. Repeat steps 9 and 10 two more times.
12. Remove high voltage and float VPP pin.
13. Power down the sensor.
14. Apply nominal power to all the power rails of the sensor VDD, VDD_IO, VAA, VAA_PIX and VDD_PLL). VPP must be floated.
15. Set EXTCLK to normal or customer-defined operating frequency.
16. Perform the proper reset sequence to the sensor.
17. Initiate the OTP memory reading process by setting R0x304A[4] to the value 0x0010.
18. Poll the register bit R0x304A[6] until bit set to "1" to check for read completion.
19. Read the 16-bit word data from the R0x304E.

Figure 21: Sequence for Programming the MT9N001


Power Supplies
RESET_BAR
EXTCLK
SCLK/SDATA
VPP
Information to be programmed to the register.
Initiate programming and poll status bit.
Read programmed values for status.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Image Acquisition Modes

The MT9N001 supports two image acquisition modes:

1. Electronic rolling shutter (ERS) mode
   This is the normal mode of operation. When the MT9N001 is streaming; it generates frames at a fixed rate, and each frame is integrated (exposed) using the ERS. When the ERS is in use, timing and control logic within the sensor sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and subsequently reading that row, the pixels in the row integrate incident light. The integration (exposure) time is controlled by varying the time between row reset and row readout. For each row in a frame, the time between row reset and row readout is fixed, leading to a uniform integration time across the frame. When the integration time is changed (by using the two-wire serial interface to change register settings), the timing and control logic controls the transition from old to new integration time in such a way that the stream of output frames from the MT9N001 switches cleanly from the old integration time to the new while only generating frames with uniform integration. See "Changes to Integration Time" on page 20.
2. Global reset mode
   This mode can be used to acquire a single image at the current resolution. In this mode, the end point of the pixel integration time is controlled by an external electromechanical shutter, and the MT9N001 provides control signals to interface to that shutter. The operation of this mode is described in detail in "Global Reset" on page 60.

The benefit of using an external electromechanical shutter is that it eliminates the visual artifacts associated with ERS operation. Visual artifacts arise in ERS operation, particularly at low frame rates, because an ERS image effectively integrates each row of the pixel array at a different point in time.

Window Control

The sequencing of the pixel array is controlled by the x_addr_start, y_addr_start, x_addr_end, and y_addr_end registers. For both parallel and serial interfaces, the output image size is controlled by the x_output_size and y_output_size registers.

Pixel Border

The default settings of the sensor provide a 3488H x 2616V image. A border of up to 8 pixels (4 in binning) on each edge can be enabled by reprogramming the x_addr_start, y_addr_start, x_addr_end, y_addr_end, x_output_size, and y_output_size registers accordingly.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

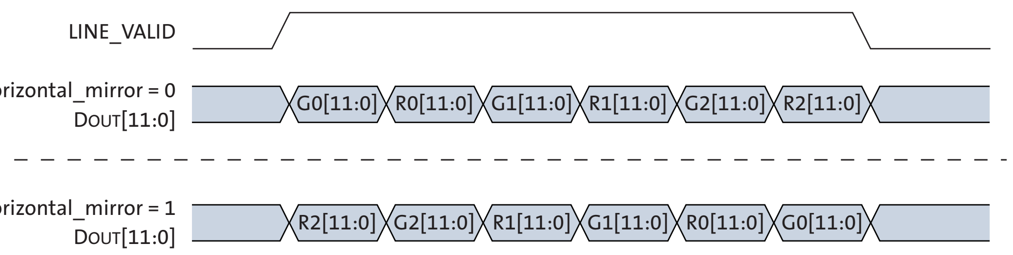
Aptina Confidential and Proprietary

Readout Modes

Horizontal Mirror

When the horizontal_mirror bit is set in the image_orientation register, the order of pixel readout within a row is reversed, so that readout starts from x_addr_end and ends at x_addr_start. Figure 22 shows a sequence of 6 pixels being read out with horizontal_mirror = 0 and horizontal_mirror = 1. Changing horizontal_mirror causes the Bayer order of the output image to change; the new Bayer order is reflected in the value of the pixel_order register.

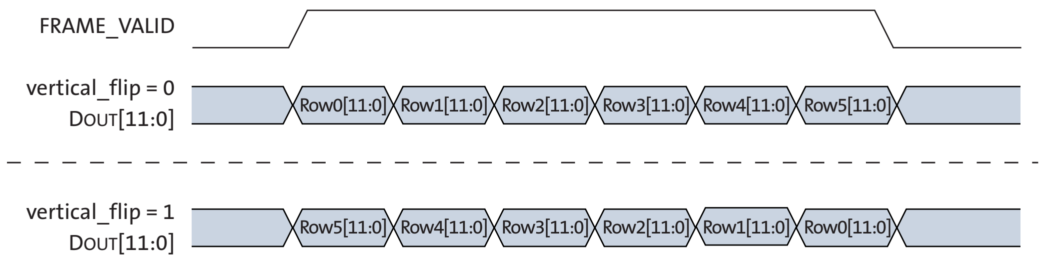
Figure 22: Effect of horizontal_mirror on Readout Order



Vertical Flip

When the vertical_flip bit is set in the image_orientation register, the order in which pixel rows are read out is reversed, so that row readout starts from y_addr_end and ends at y_addr_start. Figure 23 shows a sequence of 6 rows being read out with vertical_flip = 0 and vertical_flip = 1. Changing vertical_flip causes the Bayer order of the output image to change; the new Bayer order is reflected in the value of the pixel_order register.

Figure 23: Effect of vertical_flip on Readout Order



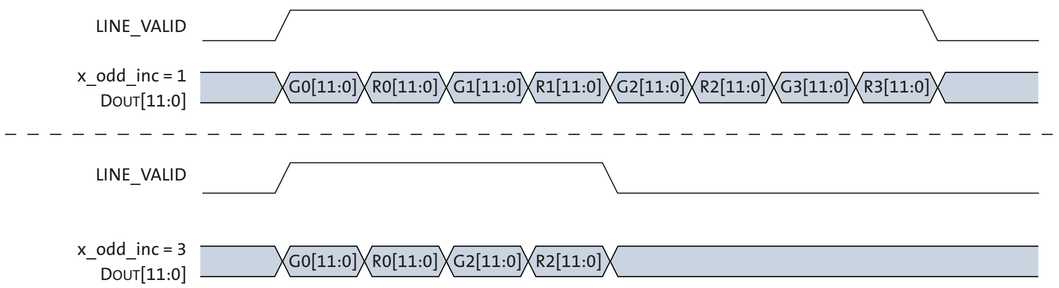
Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Subsampling

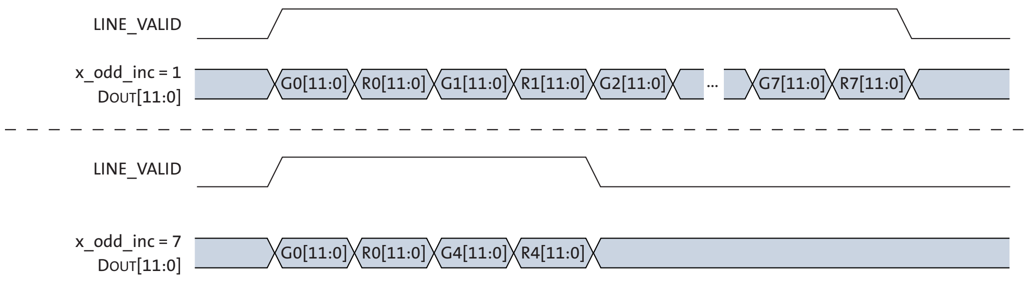
The MT9N001 supports subsampling. Subsampling reduces the amount of data processed by the analog signal chain in the MT9N001, thereby allowing the frame rate to be increased. Subsampling is enabled by setting x_odd_inc and/or y_odd_inc. Values of 1, 3, and 7 horizontal and 1, 3, 7, 15, 31, and 63 vertical can be supported. Setting both of these variables to 3 reduces the amount of row and column data processed and is equivalent to the 2 x 2 skipping readout mode provided by the MT9N001. Figure 24 shows a sequence of 8 columns being read out with x_odd_inc = 3 and y_odd_inc = 1.

Figure 24: Effect of x_odd_inc = 3 on Readout Sequence



A 1/16 reduction in resolution is achieved by setting both x_odd_inc and y_odd_inc to 7. This is equivalent to 4 x 4 skipping readout mode provided by the MT9N001. Figure 25 shows a sequence of 16 columns being read out with x_odd_inc = 7 and y_odd_inc = 1.

Figure 25: Effect of x_odd_inc = 7 on Readout Sequence



The effect of the different subsampling settings on the pixel array readout is shown in Figure 26 on page 45, Figure 27 on page 45, and Figure 28 on page 46.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Figure 26: Pixel Readout (No Subsampling)

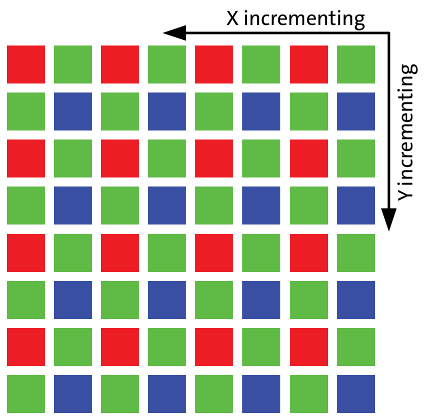
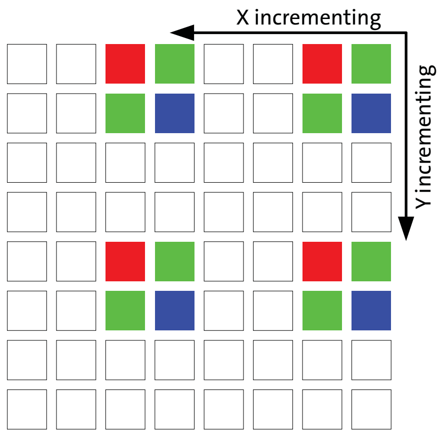


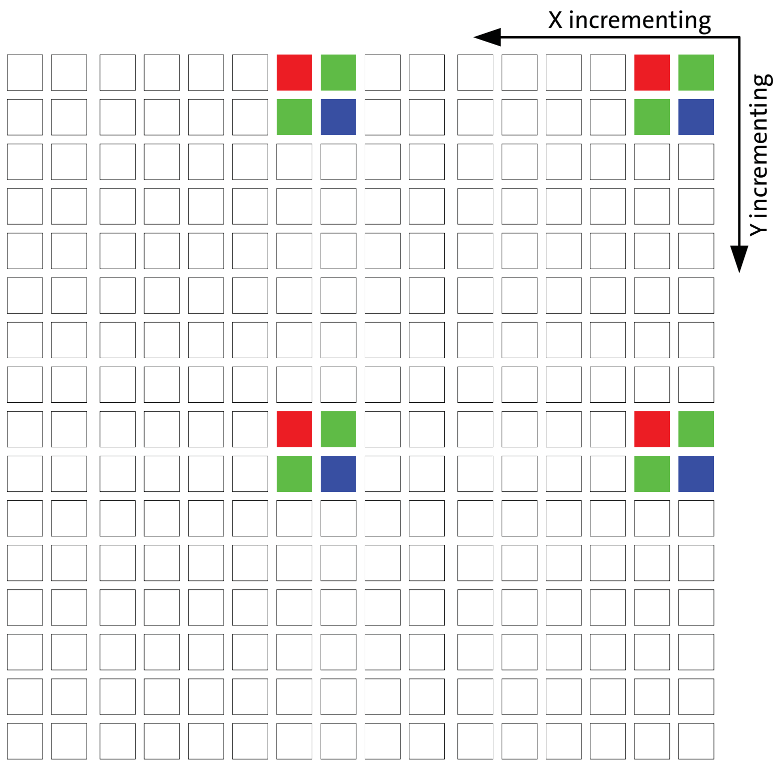
Figure 27: Pixel Readout (x_odd_inc = 3, y_odd_inc = 3)



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Figure 28: Pixel Readout (x_odd_inc = 7, y_odd_inc = 7)



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Figure 29: Pixel Readout (x_odd_inc = 7, y_odd_inc = 15)

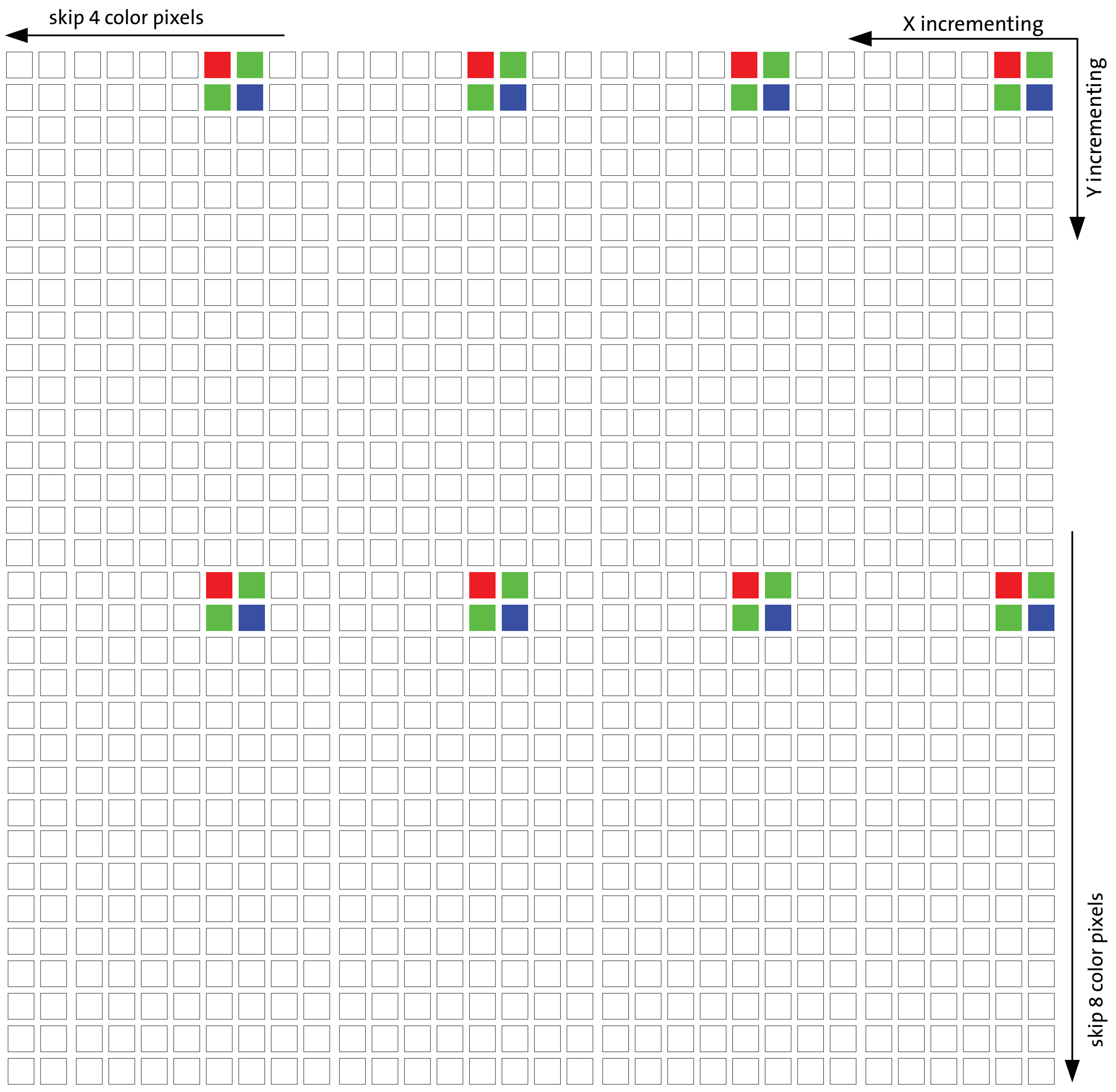
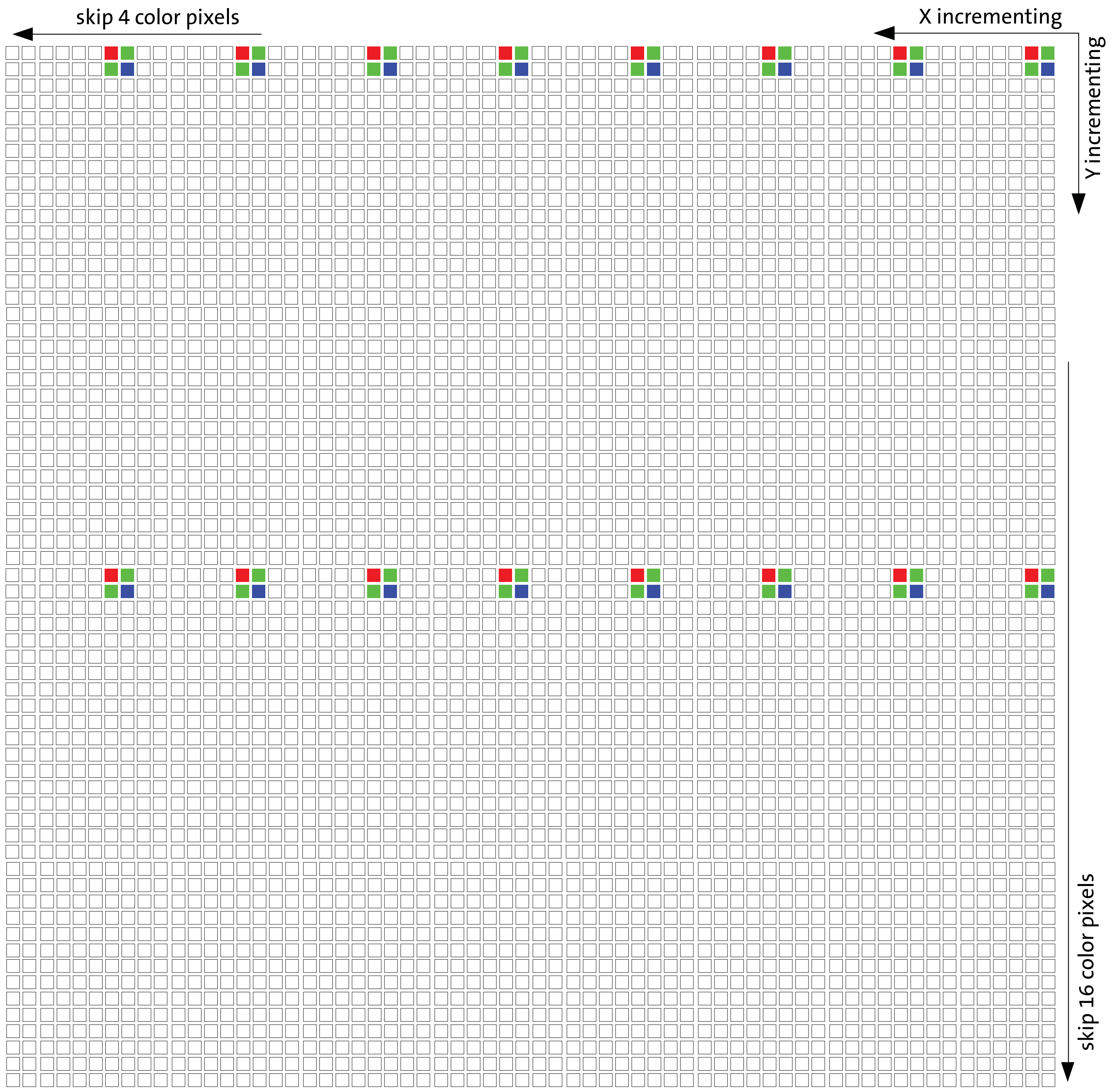


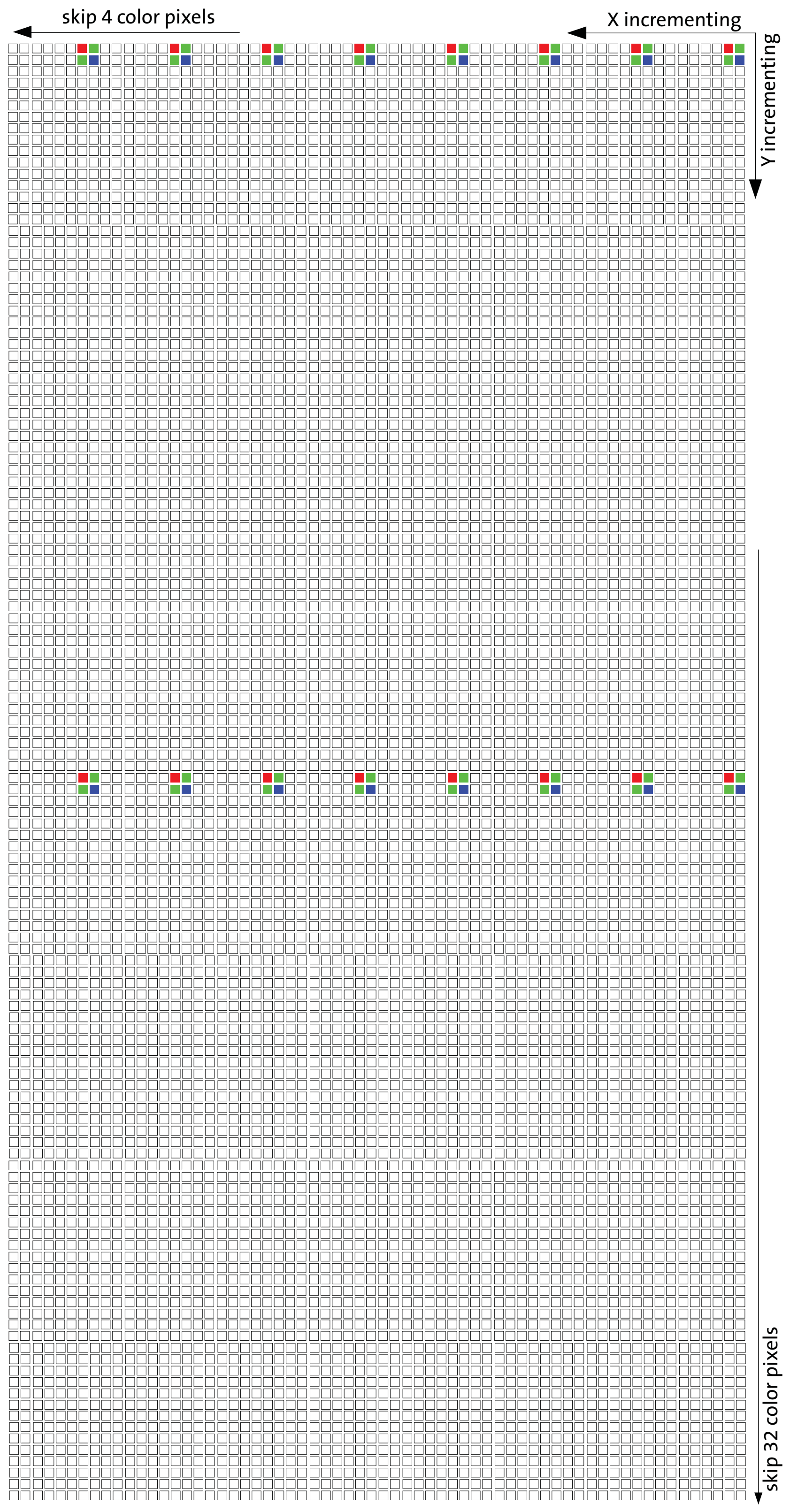
Figure 30: **Pixel Readout (x_odd_inc = 7, y_odd_inc = 31)**



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Figure 31: Pixel Readout (x_odd_inc = 7, y_odd_inc = 63)



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Programming Restrictions when Subsampling

When subsampling is enabled as a viewfinder mode and the sensor is switched back and forth between full resolution and subsampling, Aptina recommends that line_length_pck be kept constant between the two modes. This allows the same integration times to be used in each mode.

When subsampling is enabled, it may be necessary to adjust the x_addr_end, x_addr_start, and y_addr_end settings: the values for these registers are required to correspond with rows/columns that form part of the subsampling sequence. The adjustment should be made in accordance with these rules:

x_skip_factor = (x_odd_inc + 1) / 2
y_skip_factor = (y_odd_inc + 1) / 2

- for 2X skip, x_addr_start is multiple of 2 * 4 = 8
- for 4X skip, x_addr_start is multiple of 4 * 4 = 16

The number of columns/rows read out with subsampling can be found from the equation below:

columns/rows = (addr_end - addr_start + odd_inc) / skip_factor (EQ 6)

Example:

The sensor is set up to give out a full resolution 3488 x 2616 image:

[full resolution starting address with (8,8)]

```
REG=0x0104, 1            //GROUPED_PARAMETER_HOLD
REG=0x0382, 1            //X_ODD_INC
REG=0x0386, 1            //Y_ODD_INC
REG=0x0344, 8            //X_ADDR_START
REG=0x0346, 8            //Y_ADDR_START
REG=0x0348, 3495         //X_ADDR_END
REG=0x034A, 2623         //Y_ADDR_END
REG=0x034C, 3488         //X_OUTPUT_SIZE
REG=0x034E, 2616         //Y_OUTPUT_SIZE
REG=0x0104, 0            //GROUPED_PARAMETER_HOLD
```

To halve the resolution in each direction (1744 x 1308), the registers need to be reprogrammed as follows:

[2 x 2 skipping starting address with (8,8)]

```
REG=0x0104, 1            //GROUPED_PARAMETER_HOLD
REG=0x0382, 3            //X_ODD_INC
REG=0x0386, 3            //Y_ODD_INC
REG=0x0344, 8            //X_ADDR_START
REG=0x0346, 8            //Y_ADDR_START
REG=0x0348, 3493         //X_ADDR_END
REG=0x034A, 2621         //Y_ADDR_END
REG=0x034C, 1744         //X_OUTPUT_SIZE
REG=0x034E, 1308         //Y_OUTPUT_SIZE
REG=0x0104, 0            //GROUPED_PARAMETER_HOLD
```

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

To quarter the resolution in each direction (872 x 654) the registers need to be reprogrammed as follows:

[4 x 4 skipping starting address with (16,16)]

```
REG=0x0104, 1          //GROUPED_PARAMETER_HOLD
REG=0x0382, 7          //X_ODD_INC
REG=0x0386, 7          //Y_ODD_INC
REG=0x0344, 16         //X_ADDR_START
REG=0x0346, 16         //Y_ADDR_START
REG=0x0348, 3497       //X_ADDR_END
REG=0x034A, 2625       //Y_ADDR_END
REG=0x034C, 872        //X_OUTPUT_SIZE
REG=0x034E, 654        //Y_OUTPUT_SIZE
REG=0x0104, 0          //GROUPED_PARAMETER_HOLD
```

Table 13 shows the row or column address sequencing for normal and subsampled readout. In the 2X skip case, there are two possible subsampling sequences (because the subsampling sequence only reads half of the pixels) depending upon the alignment of the start address. Similarly, there will be four possible subsampling sequences in the 4X skip case (though only the first two are shown in Table 13).

Table 13: Row Address Sequencing During Subsampling

odd_inc = 1—Normal	odd_inc = 3, 2X Skip	odd_inc = 7, 4X Skip
start = 0	start = 0	start = 0
0	0	0
1	1	1
2		
3		
4	4	
5	5	
6		
7		
8	8	8
9	9	9
10		
11		
12	12	
13	13	
14		
15		

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

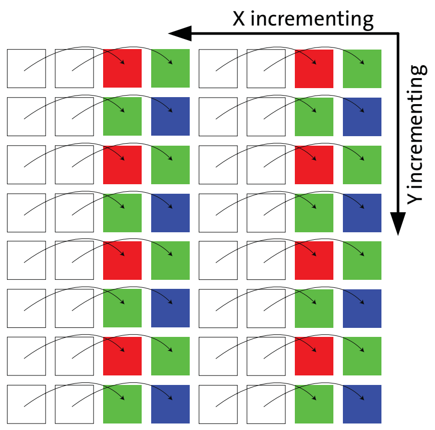
Binning

The MT9N001 supports 2 x 1 and 2 x 2 analog binning (column binning, also called x-binning, and row/column binning, also called xy-binning). Binning has many of the same characteristics as subsampling, but because it gathers image data from all pixels in the active window (rather than a subset of them), it achieves superior image quality and avoids the aliasing artifacts that can be a characteristic side effect of subsampling.

Binning is enabled by selecting the appropriate subsampling settings (odd_inc = 3 and y_odd_inc = 1 for x-binning, x_odd_inc = 3 and y_odd_inc = 3 for xy-binning) and setting the appropriate binning bit in read_mode (R0x3040–1). As with subsampling, x_addr_end and y_addr_end may require adjustment when binning is enabled. It is the first of the two columns/rows binned together that should be the end column/row in binning, so the requirements to the end address are exactly the same as in non-binning subsampling mode. The effect of the different subsampling settings is shown in Figure 32 and Figure 33 on page 53.

Binning can also be enabled when the 4X subsampling mode is enabled (x_odd_inc = 7 and y_odd_inc = 1 for x-binning, x_odd_inc = 7 and y_odd_inc = 7 for xy-binning). In this mode, however, not all pixels will be used so this is not a 4X binning implementation. An implementation providing a combination of skip2 and bin2 is used to achieve 4X subsampling with better image quality. The effect of this subsampling mode is shown in Figure 34 on page 53.

Figure 32: Pixel Readout (x_odd_inc = 3, y_odd_inc = 1, x_bin = 1)



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Figure 33: Pixel Readout (x_odd_inc = 3, y_odd_inc = 3, xy_bin = 1)

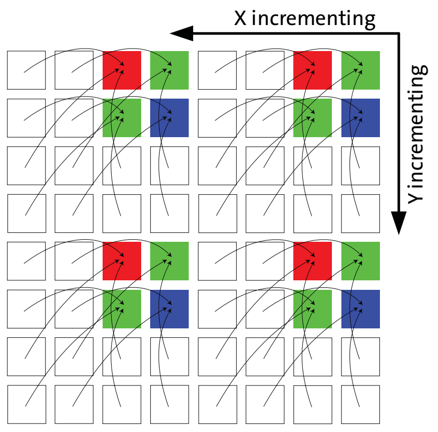
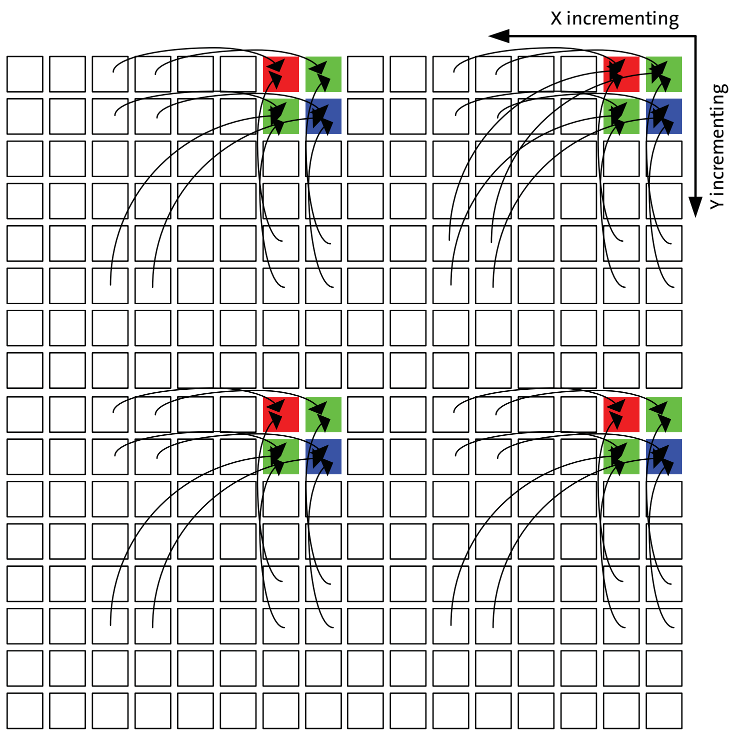


Figure 34: Pixel Readout (x_odd_inc = 7, y_odd_inc = 7, xy_bin = 1)



Binning address sequencing is a bit more complicated than during subsampling only, because of the implementation of the binning itself.

For a given column *n*, there is only one other column, n_bin, that it can be binned with, because of physical limitations in the column readout circuitry. The possible address sequences are shown in Table 14.

Summing can also be implemented in binning mode where pixel values are added together as opposed to being averaged in regular binning mode. Vertical summing can be enabled with the MT9N001. The Aptina-recommended combination is to enable Y summing using register 0x3040[13], and X binning using register 0x3040[11].

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Table 14: Column Address Sequencing During Binning

odd_inc = 1—Normal	odd_inc = 3, 2X Bin
x_addr_start = 0	x_addr_start = 0
0	0/2
1	1/3
2	
3	
4	4/6
5	5/7
6	
7	
8	8/10
9	9/11
10	
11	
12	12/14
13	13/15
14	
15	

There are no physical limitations on what can be binned together in the row direction. A given row *n* will always be binned with row n + 2 in 2X subsampling mode and with row n + 4 in 4X subsampling mode. Therefore, which rows get binned together depends upon the alignment of y_addr_start. The possible sequences are shown in Table 15.

Table 15: Row Address Sequencing During Binning

odd_inc = 1—Normal	odd_inc = 3, 2X Bin	odd_inc = 7, 2X Skip + 2X Bin
x_addr_start = 0	x_addr_start = 0	x_addr_start = 0
0	0/2	0/4
1	1/3	1/5
2		
3		
4	4/6	
5	5/7	
6		
7		
8	8/10	8/12
9	9/11	9/13
10		
11		
12	12/14	
13	13/15	
14		
15		

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Programming Restrictions when Binning

Binning requires different sequencing of the pixel array and imposes different timing limits on the operation of the sensor. In particular, xy-binning requires two read operations from the pixel array for each line of output data, which has the effect of increasing the minimum line blanking time. The SMIA specification cannot accommodate this variation because its parameter limit registers are defined as being static.

As a result, when xy-binning is enabled, some of the programming limits declared in the parameter limit registers are no longer valid. In addition, the default values for some of the manufacturer-specific registers need to be reprogrammed. See section "Minimum Frame Time" on page 56, section "Minimum Row Time" on page 56, and section "Fine Integration Time Limits" on page 57.

Table 16: Readout Modes

Readout Modes	x_odd_inc, y_odd_inc	xy_bin
2X skip	3	0
2X bin	3	1
4X skip	7	0
2X skip and 2X bin	7	1

Frame Rate Control

The formulas for calculating the frame rate of the MT9N001 are shown below.

The line length is programmed directly in pixel clock periods through register line_length_pck. For a specific window size, the minimum line length can be found from in Equation 7:

$$minimum\ line\_length\_pck = \left(\frac{x\_addr\_end - x\_addr\_start\ + 1}{subsampling\ factor} + min\_line\_blanking\_pck\right) \tag{EQ 7}$$

Note that line_length_pck also needs to meet the minimum line length requirement set in register min_line_length_pck. The row time can either be limited by the time it takes to sample and reset the pixel array for each row, or by the time it takes to sample and read out a row. Values for min_line_blanking_pck are provided in “Minimum Row Time” on page 56.

The frame length is programmed directly in number of lines in the register frame_line_length. For a specific window size, the minimum frame length is shown in Equation 8:

$$minimum\ frame\_length\_lines = \left(\frac{y\_addr\_end - y\_addr\_start\ + 1}{subsampling\ factor} + min\_frame\_blanking\_lines\right) \tag{EQ 8}$$

The frame rate can be calculated from these variables and the pixel clock speed as shown in Equation 9:

$$frame\ rate = \frac{vt\_pixel\_clock\_mhz\ x\ 1\ x\ 10^6}{line\_length\_pck\_x\ frame\_length\_lines} \tag{EQ 9}$$

If coarse_integration_time is set larger than frame_length_lines the frame size will be expanded to coarse_integration_time + 1.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Minimum Row Time

The minimum row time and blanking values with default register settings are shown in Table 17.

Table 17: Minimum Row Time and Blanking Numbers

Register	No Row Binning			Row Binning		
row_speed[2:0]	1	2	4	1	2	4
min_line_blanking_pck	0x06E4	0x0442	0x02F1	0x0C48	0x06CC	0x040E
min_line_length_pck	0x0910	0x0588	0x0438	0x01200	0x0900	0x0550

In addition, enough time must be given to the output FIFO so it can output all data at the set frequency within one row time.

There are therefore three checks that must all be met when programming line_length_pck:

1. line_length_pck≥ min_line_length_pck in Table 17.
2. line_length_pck ≥ (x_addr_end - x_addr_start + x_odd_inc)/((1+x_odd_inc)/2) + min_line_blanking_pck in Table 17.
3. The row time must allow the FIFO to output all data during each row line_length_pck ≥ (x_output_size * 2 + 0x005E) * "vt_pix_clk period" / "op_pix_clk period".

Minimum Frame Time

The minimum number of rows in the image is 2, so min_frame_length_lines will always equal (min_frame_blanking_lines + 2).

Table 18: Minimum Frame Time and Blanking Numbers

Register	
min_frame_blanking_lines	0x008F
min_frame_length_lines	0x0091

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Integration Time

The integration (exposure) time of the MT9N001 is controlled by the fine_integration_time and coarse_integration_time registers.

The limits for the fine integration time are defined by:

$$fine\_integration\_time\_min \le fine\_integration\_time \le (line\_length\_pck - fine\_integration\_time\_max\_margin \quad \text{(EQ 10)}$$

The limits for the coarse integration time are defined by:

$$coarse\_integration\_time\_min \le coarse\_integration\_time \quad \text{(EQ 11)}$$

The actual integration time is given by:

$$integration\_time = \frac{((coarse\_integration\_time * line\_length\_pck) + fine\_integration\_time)}{(vt\_pix\_clk\_freq\_mhz * 10^6)} \quad \text{(EQ 12)}$$

It is required that:

$$coarse\_integration\_time <= (frame\_length\_lines - coarse\_integration\_time\_max\_margin) \quad \text{(EQ 13)}$$

If this limit is exceeded, the frame time will automatically be extended to (*coarse_integration_time* + *coarse_integartion_time_max_margin*) to accommodate the larger integration time.

Fine Integration Time Limits

The limits for the fine_integration_time can be found from fine_integration_time_min and fine_integration_time_max_margin. Values for different mode combinations are shown in Table 19.

Table 19: fine_integration_time Limits

Register	No Row Binning			Row Binning		
row_speed[2:0]	1	2	4	1	2	4
fine_integration_time_min	0x056A	0x0C26	0x00C2	0x0B1A	0x059E	0x0178
fine_integration_time_max_margin	0x03A6	0x01C2	0x0182	0x06E6	0x0362	0x0308

fine_correction

For the fine_integration_time limits, the fine_correction constant will change with the pixel clock speed and binning mode. These values are shown in Table 20.

Table 20: fine_correction Values

Register	No Row Binning			Row Binning		
row_speed[2:0]	1	2	4	1	2	4
fine_correction	0x0100	0x007A	0x0037	0x0238	0x0116	0x0085

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Low Power Mode

The MT9N001 sensor supports a low power mode, which can be entered by programming register bit read_mode[9]. Setting this bit will do the following:

- Double the value of pc_speed[2:0] internally. This means halving the internal pixel clock frequency.
- Lower currents in the analog domain. This can be done by setting a low power bit in the static control register. The current will be halved where appropriate in the analog domain.

Note that enabling the low power mode will not put the sensor in subsampling mode. This will have to be programmed separately as described earlier in this document. Low power is independent of the readout mode and can also be enabled in full resolution mode. Because the pixel clock speed is halved, the frame rates that can be achieved with low power mode are lower than in full power mode.

Because only internal pixel clock speeds of 1, 2, and 4 are supported, low power mode combined with pc_speed[2:0] = 4 is an illegal combination.

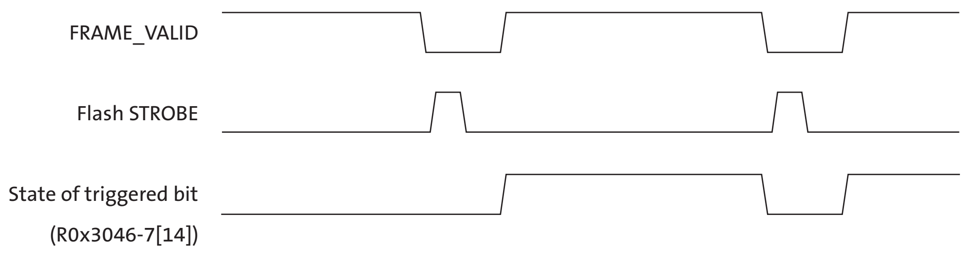
Any limitations related to changing the internal pixel clock speed will also apply to low power mode, because it automatically changes the pixel clock speed. Therefore, the limiter registers need to be reprogrammed to match the new internal pixel clock frequency.

Flash Control

The MT9N001 supports both xenon and LED flash through the FLASH output signal. The timing of the FLASH signal with the default settings is shown in Figure 35, and in Figure 36 and Figure 37 on page 59. The flash and flash_count registers allow the timing of the flash to be changed. The flash can be programmed to fire only once, 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 either by first enabling mask bad frames (write reset_register[9] = 1) before the enabling the flash or by forcing a restart (write reset_register[1] = 1) immediately after enabling the flash; the first bad frame will then be masked out, as shown in Figure 37 on page 59. Read-only bit flash[14] is set during frames that are correctly integrated; the state of this bit is shown in Figures 35, 36, and Figure 37.

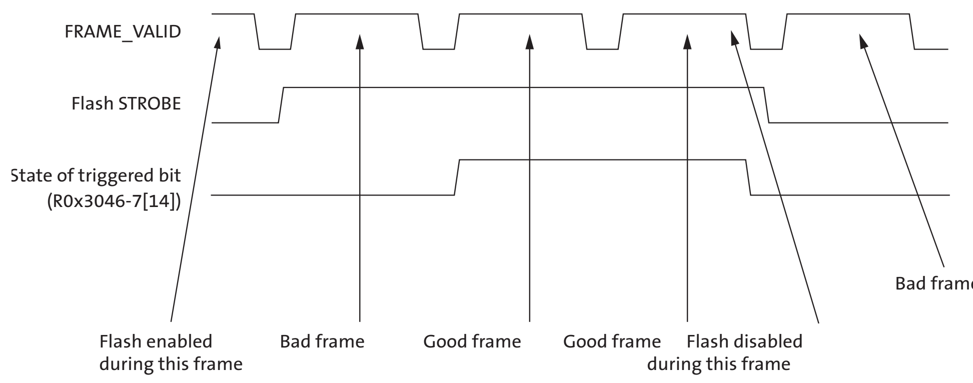
Figure 35: Xenon Flash Enabled



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

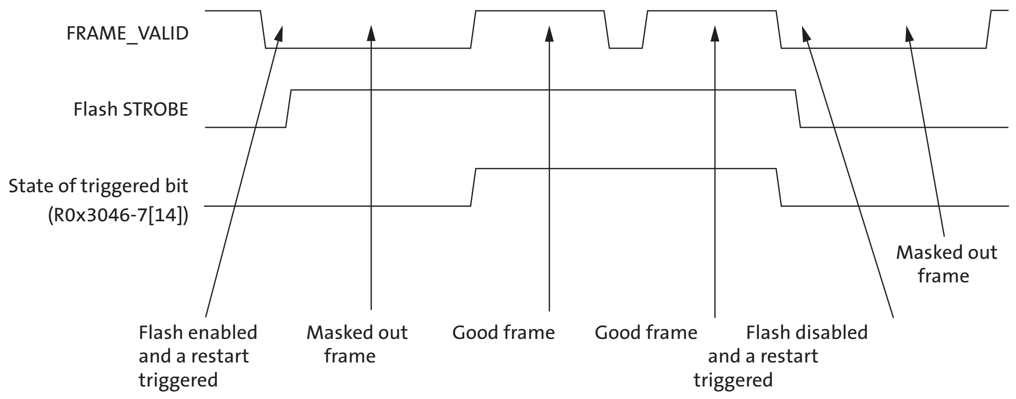
Aptina Confidential and Proprietary

Figure 36: LED Flash Enabled


FRAME_VALID
Flash STROBE
State of triggered bit (R0x3046-7[14])
Bad frame
Flash enabled during this frame
Bad frame
Good frame
Good frame
Flash disabled during this frame

Notes:
1. Integration time = number of rows in a frame.
2. Bad frames will be masked during LED flash operation when mask bad frames bit field is set (R0x301A[9] = 1).
3. An option to invert the flash output signal through R0x3046[7] is also available.

Figure 37: LED Flash Enabled Following Forced Restart


FRAME_VALID
Flash STROBE
State of triggered bit (R0x3046-7[14])
Masked out frame
Flash enabled and a restart triggered
Masked out frame
Good frame
Good frame
Flash disabled and a restart triggered

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Global Reset

Global reset mode allows the integration time of the MT9N001 to be controlled by an external electromechanical shutter. Global reset mode is generally used in conjunction with ERS mode. The ERS mode is used to provide viewfinder information, the sensor is switched into global reset mode to capture a single frame, and the sensor is then returned to ERS mode to restore viewfinder operation.

Global reset mode is designed for use in conjunction with the parallel pixel data interface. The SMIA specification does not define a global reset mode and only provides for operation in ERS mode. The MT9N001 does support the use of global reset mode in conjunction with the SMIA data path, but there are additional restrictions on its use.

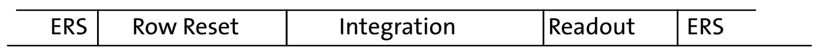
Overview of Global Reset Sequence

The basic elements of the global reset sequence are:

1. By default, the sensor operates in ERS mode and the SHUTTER output signal is LOW. The electromechanical shutter must be open to allow light to fall on the pixel array. Integration time is controlled by the coarse_integration_time and fine_integration_time registers.
2. A global reset sequence is triggered.
3. All of the rows of the pixel array are placed in reset.
4. All of the rows of the pixel array are taken out of reset simultaneously. All rows start to integrate incident light. The electromechanical shutter may be open or closed at this time.
5. If the electromechanical shutter has been closed, it is opened.
6. After the desired integration time (controlled internally or externally to the MT9N001), the electromechanical shutter is closed.
7. A single output frame is generated by the sensor with the usual LV, FV, PIXCLK, and DOUT timing. As soon as the output frame has completed (FV de-asserts), the electromechanical shutter may be opened again.
8. The sensor automatically resumes operation in ERS mode.

This sequence is shown in Figure 38. The following sections expand to show how the timing of this sequence is controlled.

Figure 38: Overview of Global Reset Sequence



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Entering and Leaving the Global Reset Sequence

A global reset sequence can be triggered by a register write to global_seq_trigger[0] (global trigger, to transition this bit from a 0 to a 1) or by a rising edge on a suitably-configured GPI input (see "Trigger Control" on page 35).

When a global reset sequence is triggered, the sensor waits for the end of the current row. When LV de-asserts for that row, FV is de-asserted 6 PIXCLK periods later, potentially truncating the frame that was in progress.

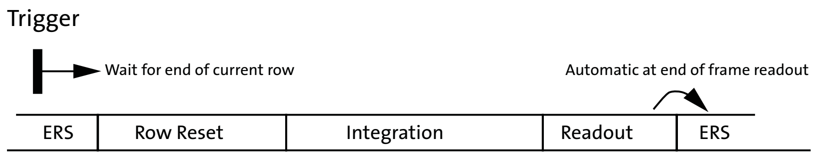
The global reset sequence completes with a frame readout. At the end of this readout phase, the sensor automatically resumes operation in ERS mode. The first frame integrated with ERS will be generated after a delay of approximately:

*((13 + coarse_integration_time) * line_length_pck).*

This sequence is shown in Figure 39.

While operating in ERS mode, double-buffered registers ("Double-Buffered Registers" on page 19) are updated at the start of each frame in the usual way. During the global reset sequence, double-buffered registers are updated just before the start of the readout phase.

Figure 39: Entering and Leaving a Global Reset Sequence


Trigger
Wait for end of current row
Automatic at end of frame readout
ERS
Row Reset
Integration
Readout
ERS

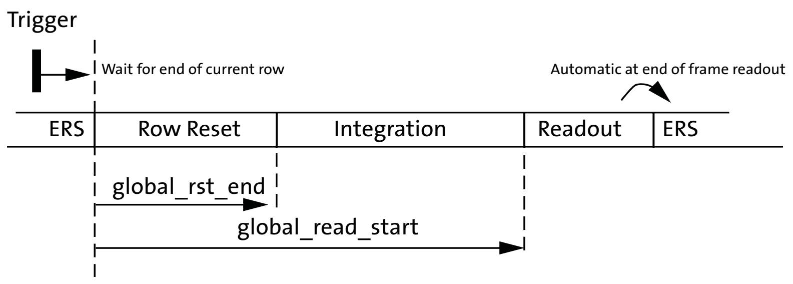
Programmable Settings

The registers global_rst_end and global_read_start allow the duration of the row reset phase and the integration phase to be controlled, as shown in Figure 40. The duration of the readout phase is determined by the active image size.

The recommended setting for global_rst_end is 0x074C. This allows sufficient time for all rows of the pixel array to be set to the correct reset voltage level. The row reset phase takes a finite amount of time due to the capacitance of the pixel array and the capability of the internal voltage booster circuit that is used to generate the reset voltage level.

As soon as the global_rst_end count has expired, all rows in the pixel array are simultaneously taken out of reset and the pixel array begins to integrate incident light.

Figure 40: Controlling the Reset and Integration Phases of the Global Reset Sequence



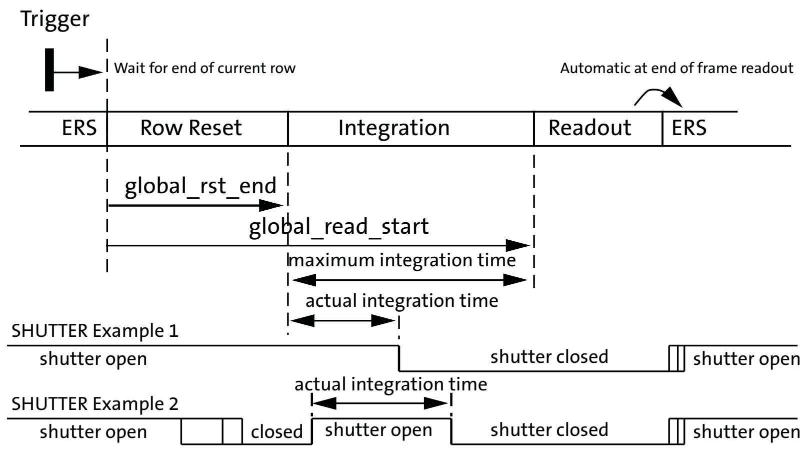
Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Control of the Electromechanical Shutter

Figure 41 shows two different ways in which a shutter can be controlled during the global reset sequence. In both cases, the maximum integration time is set by the difference between global_read_start and global_rst_end. In shutter example 1, the shutter is open during the initial ERS sequence and during the row reset phase. The shutter closes during the integration phase. The pixel array is integrating incident light from the start of the integration phase to the point at which the shutter closes. Finally, the shutter opens again after the end of the readout phase. In shutter example 2, the shutter is open during the initial ERS sequence and closes sometime during the row reset phase. The shutter both opens and closes during the integration phase. The pixel array is integrating incident light for the part of the integration phase during which the shutter is open. As for the previous example, the shutter opens again after the end of the readout phase.

Figure 41: Control of the Electromechanical Shutter



It is essential that the shutter remains closed during the entire row readout phase (that is, until FV has de-asserted for the frame readout); otherwise, some rows of data will be corrupted (over-integrated).

It is essential that the shutter closes before the end of the integration phase. If the row readout phase is allowed to start before the shutter closes, each row in turn will be integrated for one row-time longer than the previous row.

After FV de-asserts to signal the completion of the readout phase, there is a time delay of approximately *10 * line_length_pck* before the sensor starts to integrate light-sensitive rows for the next ERS frame. It is essential that the shutter be opened at some point in this time window; otherwise, the first ERS frame will not be uniformly integrated.

The MT9N001 provides a SHUTTER output signal to control (or help the host system control) the electromechanical shutter. The timing of the SHUTTER output is shown in Figure 42 on page 63. SHUTTER is de-asserted by default. The point at which it asserts is controlled by the programming of global_shutter_start. At the end of the global reset readout phase, SHUTTER de-asserts approximately *2 * line_length_pck* after the de-assertion of FV.

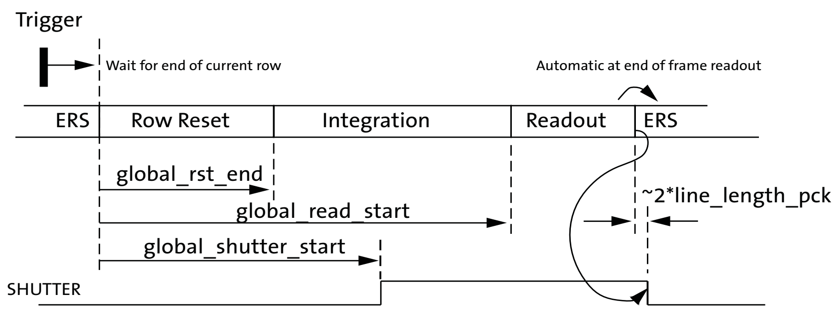
This programming restriction must be met for correct operation:

- global_read_start > global_shutter_start.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

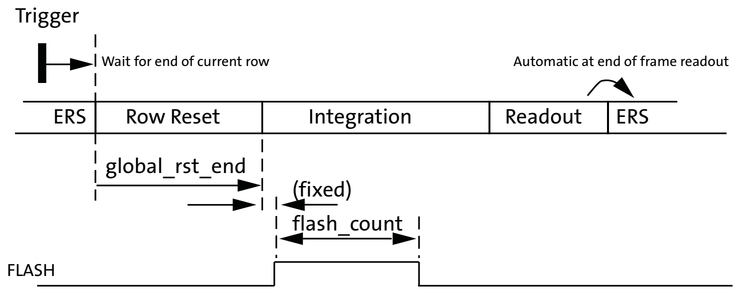
Figure 42: Controlling the SHUTTER Output



Using FLASH with Global Reset

If global_seq_trigger[2] = 1 (global flash enabled) when a global reset sequence is triggered, the FLASH output signal will be pulsed during the integration phase of the global reset sequence. The FLASH output will assert a fixed number of cycles after the start of the integration phase and will remain asserted for a time that is controlled by the value of the flash_count register, as shown in Figure 43.

Figure 43: Using FLASH with Global Reset



External Control of Integration Time

If global_seq_trigger[1] = 1 (global bulb enabled) when a global reset sequence is triggered, the end of the integration phase is controlled by the level of trigger (global_seq_trigger[0] or the associated GPI input). This allows the integration time to be controlled directly by an input to the sensor and allows integration times that are longer than can be accommodated by the programming limits 174msec by the global_read_start register.

This operation corresponds to the shutter "B" setting on a traditional camera, where "B" originally stood for "Bulb" (the shutter setting used for synchronization with a magnesium foil flash bulb) and was later considered to stand for "Brief" (an exposure that was longer than the shutter could automatically accommodate).

When the trigger is de-asserted to end integration, the integration phase is extended by a further time given by *global_read_start – global_shutter_start*. Usually this means that global_read_start should be set to *global_shutter_start + 1*.

The operation of this mode is shown in Figure 44 on page 64. The figure shows the global reset sequence being triggered by the GPI2 input, but it could be triggered by any of the GPI inputs or by the setting and subsequence clearing of the global_seq_trigger[0] under software control.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

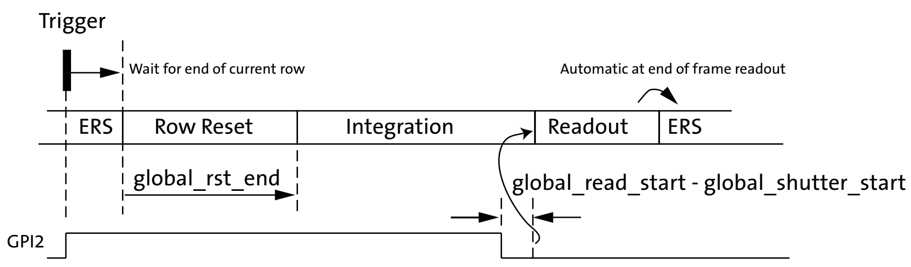
The integration time of the GRR sequence is defined as:

$$Integration = \frac{(global\_read\_start - global\_rst\_end)\ x\ 512}{vt\_pix\_clk\_freq\_mhz} \quad \text{(EQ 14)}$$

These programming restrictions must be met for correct operation of bulb exposures:

- global_read_start > global_shutter_start
- global_shutter_start > global_rst_end
- global_shutter_start must be smaller than the exposure time (that is, this counter must expire before the trigger is de-asserted)

Figure 44: Global Reset Bulb


Trigger
Wait for end of current row
Automatic at end of frame readout
ERS
Row Reset
Integration
Readout
ERS
global_rst_end
global_read_start - global_shutter_start
GPI2

Retriggering the Global Reset Sequence

The trigger for the global reset sequence is edge-sensitive; the global reset sequence cannot be retriggered until the global trigger bit (in the global_seq_trigger register) has been returned to “0,” and the GPI (if any) associated with the trigger function has been de-asserted.

The earliest time that the global reset sequence can be retriggered is the point at which the SHUTTER output de-asserts; this occurs approximately *2 * line_length_pck* after the negation of FV for the global reset readout phase.

Using Global Reset with SMIA Data Path

When a global reset sequence is triggered, it usually results in the frame in progress being truncated (at the end of the current output line). The SMIA data path limiter function (see Figure 47 on page 70) attempts to extend (pad) all frames to the programmed value of y_output_size. If this padding is still in progress when the global reset readout phase starts, the SMIA data path will not detect the start of the frame correctly. Therefore, to use global reset with the serial data path, this timing scenario must be avoided. One possible way of doing this would be to synchronize (under software control) the assertion of trigger to an end-of-frame marker on the serial data stream.

At the end of the readout phase of the global reset sequence, the sensor automatically resumes operation in ERS mode.

The frame that is read out of the sensor during the global reset readout phase has exactly the same format as any other frame out of the serial pixel data interface, including the addition of two lines of embedded data. The values of the coarse_integration_time and fine_integration_time registers within the embedded data match the programmed values of those registers and do *not* reflect the integration time used during the global reset sequence.

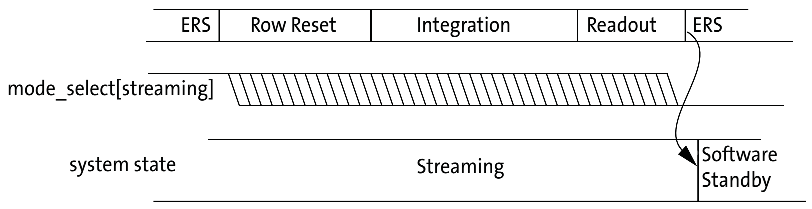
Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Global Reset and Soft Standby

If the mode_select[stream] bit is cleared while a global reset sequence is in progress, the MT9N001 will remain in streaming state until the global reset sequence (including frame readout) has completed, as shown in Figure 45.

Figure 45: Entering Soft Standby During a Global Reset Sequence



Analog Gain

The MT9N001 provides two mechanisms for setting the analog gain. The first uses the SMIA gain model. The second uses the traditional Aptina Imaging gain model. The following sections describe both models, the mapping between the models, and the operation of the per-color and global gain control.

Using Per-color or Global Gain Control

The read-only analogue_gain_capability register returns a value of “1,” indicating that the MT9N001 provides per-color gain control. However, the MT9N001 also provides the option of global gain control. Per-color and global gain control can be used interchangeably. A WRITE to a global gain register is aliased as a WRITE of the same data to the four associated color-dependent gain registers. A READ from a global gain register is aliased to a READ of the associated greenB/greenR gain register.

The read/write gain_mode register required by SMIA has no defined function. In the MT9N001, this register has no side effects on the operation of the gain; per-color and global gain control can be used interchangeably regardless of the state of the gain_mode register.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

SMIA Gain Model

The SMIA gain model uses these registers to set the analog gain:

- analogue_gain_code_global
- analogue_gain_code_greenR
- analogue_gain_code_red
- analogue_gain_code_blue
- analogue_gain_code_greenB

The SMIA gain model requires a uniform step size between all gain settings. The analog gain is given by:

$$gain = \frac{analogue\_gain\_m0 \times analogue\_gain\_code}{analogue\_gain\_c1} = \frac{analogue\_gain\_code\_<color>}{8} \quad \text{(EQ 15)}$$

Aptina Imaging Gain Model

The Aptina Imaging gain model uses these registers to set the analog gain:

- global_gain
- greenR_gain
- red_gain
- blue_gain
- greenB_gain

This provides a 7-bit, a 3X, and a 2X gain state. As a result, the step size varies depending upon if the 2X and 3X gain stages are enabled. The analog gain is given by:

$$gain = (2*<color>\_gain[9]+<color>\_gain[8]-<color>\_gain[9]*<color>\_gain[8]+1)\ x\ (<color>\_gain[7]+1) x \frac{<color>\_gain[6:0]}{32} \quad \text{(EQ 16)}$$

As a result of the 2X and 3X gain stages, many of the possible gain settings can be achieved in many different ways. The recommended gain sequence is shown in Table 21.

Table 21: Recommended Gain Stages

Desired Gain	Recommended Gain Register Setting
1–1.96875	0x1020–0x103F
2–7.938	0x10A0–0x10FF
8–15.875	0x11C0–0x11FF
15.9735–23.8125	0x12D5–0x12FF

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Gain Code Mapping

The Aptina Imaging gain model maps directly to the underlying structure of the gain stages in the analog signal chain. When the SMIA gain model is used, gain codes are translated into equivalent settings in the Aptina Imaging gain model.

When the SMIA gain model is in use and values have been written to the analogue_gain_code_<color> registers, the associated value in the Aptina Imaging gain model can be read from the associated <color>_gain register. In cases where there is more than one possible mapping, the 2X and 3X gain stages are enabled to provide the mapping with the lowest noise.

When the Aptina Imaging gain model is in use and values have been written to the gain_<color> undefined. The reason for this is that many of the gain codes available in the Aptina Imaging gain model have no corresponding value in the SMIA gain model.

The result of this is that the two gain models can be used interchangeably, but gains written through one set of registers should be read back through the same set of registers.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Sensor Core Digital Data Path

Test Patterns

The MT9N001 supports a number of test patterns to facilitate system debug. Test patterns are enabled using test_pattern_mode (R0x0600–1). The test patterns are listed in Table 22.

Table 22: Test Patterns

test_pattern_mode	Description
0	Normal operation: no test pattern
1	Solid color
2	100% color bars
3	Fade-to-gray color bars
4	PN9 link integrity pattern (only on sensors with serial interface)
256	Walking 1s (12-bit value)
257	Walking 1s (10-bit value)
258	Walking 1s (8-bit value)

Test patterns 0–3 replace pixel data in the output image (the embedded data rows are still present). Test pattern 4 replaces all data in the output image (the embedded data rows are omitted and test pattern data replaces the pixel data).

For all of the test patterns, the MT9N001 registers must be set appropriately to control the frame rate and output timing. This includes:

- All clock divisors
- x_addr_start
- x_addr_end
- y_addr_start
- y_addr_end
- frame_length_lines
- line_length_pck
- x_output_size
- y_output_size

Test Cursors

The MT9N001 supports one horizontal and one vertical cursor, allowing a crosshair to be superimposed on the image or on test patterns 1–3. The position and width of each cursor are programmable in R0x31E8–R0x31EE. Both even and odd cursor positions and widths are supported.

Each cursor can be inhibited by setting its width to “0.” The programmed cursor position corresponds to the x and y addresses of the pixel array. For example, setting horizontal_cursor_position to the same value as y_addr_start would result in a horizontal cursor being drawn starting on the first row of the image. The cursors are opaque (they replace data from the imaged scene or test pattern). The color of each cursor is set by the values of the Bayer components in the test_data_red, test_data_greenR, test_data_blue and test_data_greenB registers. As a consequence, the cursors are the same color as test pattern 1 and are therefore invisible when test pattern 1 is selected.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

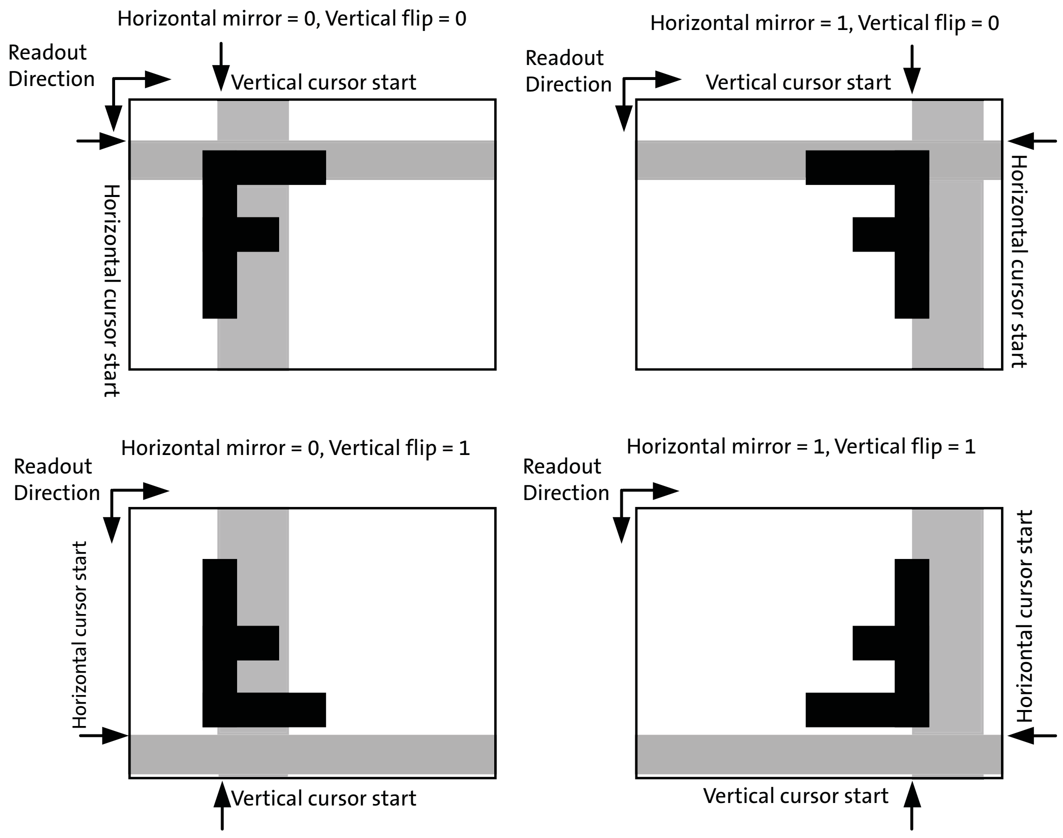
Aptina Confidential and Proprietary

When vertical_cursor_position = 0x0FFF, the vertical cursor operates in an automatic mode in which its position advances every frame. In this mode the cursor starts at the column associated with x_addr_start = 0 and advances by a step-size of 8 columns each frame, until it reaches the column associated with x_addr_start = 2040, after which it wraps (256 steps). The width and color of the cursor in this automatic mode are controlled in the usual way.

The effect of enabling the test cursors when the image_orientation register is non-zero is not defined by the design specification. The behavior of the MT9N001 is shown in Figure 46 on page 69 and the test cursors are shown as translucent, for clarity. In practice, they are opaque (they overlay the imaged scene). The manner in which the test cursors are affected by the value of image_orientation can be understood from these implementation details:

- The test cursors are inserted last in the data path, the cursor is applied with out any sensor corrections.
- The drawing of a cursor starts when the pixel array row or column address is within the address range of cursor start to cursor start + width.
- The cursor is independent of image orientation.

Figure 46: Test Cursor Behavior with image_orientation



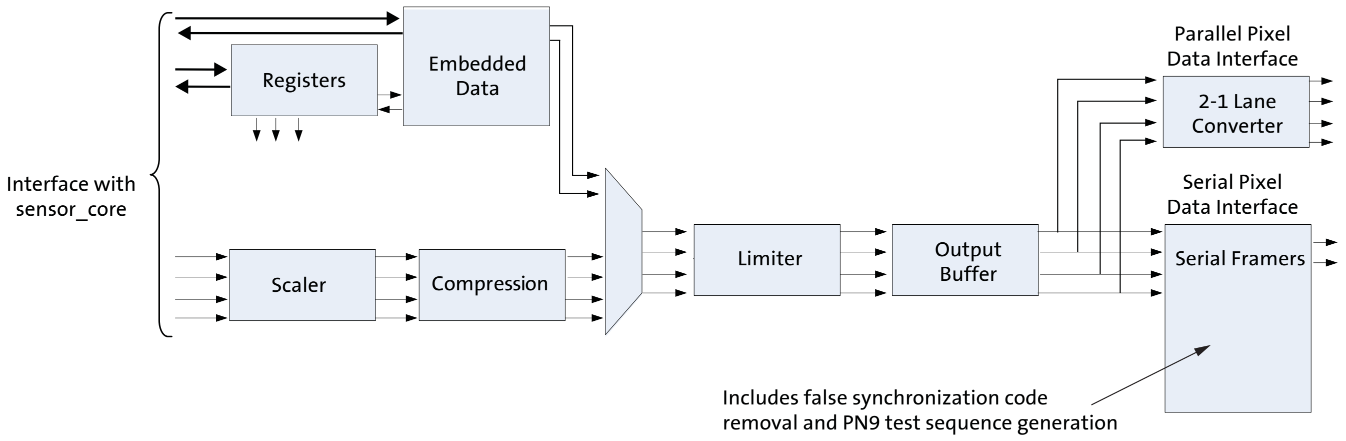
Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Digital Data Path

The digital data path after the sensor core is shown in Figure 47.

Figure 47: Data Path


Interface with sensor_core
Registers
Embedded Data
Scaler
Compression
Limiter
Output Buffer
Parallel Pixel Data Interface
2-1 Lane Converter
Serial Pixel Data Interface
Serial Framers
Includes false synchronization code removal and PN9 test sequence generation

Embedded Data Format and Control

When the serial pixel data path is selected, the first two rows of the output image contain register values that are appropriate for the image. The 12-bit format places the data byte in bits [11:4] and sets bits [3:0] to a constant value of 0101. Some register values are dynamic and may change from frame to frame. Additional information on the format of the embedded data can be located in the SMIA functional specification.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Timing Specifications

Power-Up Sequence

The recommended power-up sequence for the MT9N001 is shown in Figure 48. The available power supplies (VDD_IO, VDD, VDD_TX0, VDD_PLL, VAA, VAA_PIX) can be turned on at the same time or have the separation specified below.

1. Turn on VDD_IO power supply.
2. After 0–500ms, turn on VDD and VDD_TX0 power supply.
3. After 0–500ms, turn on VDD_PLL and VAA/VAA_PIX power supplies.
4. After the last power supply is stable, enable EXTCLK.
5. Assert RESET_BAR for at least 1ms.
6. Wait 2400 EXTCLKs (tentative) for internal initialization into software standby.
7. Configure PLL, output, and image settings to desired values.
8. Set mode_select = 1 (R0x0100).
9. Wait 1ms for the PLL to lock before streaming state is reached.

Figure 48: Power-Up Sequence

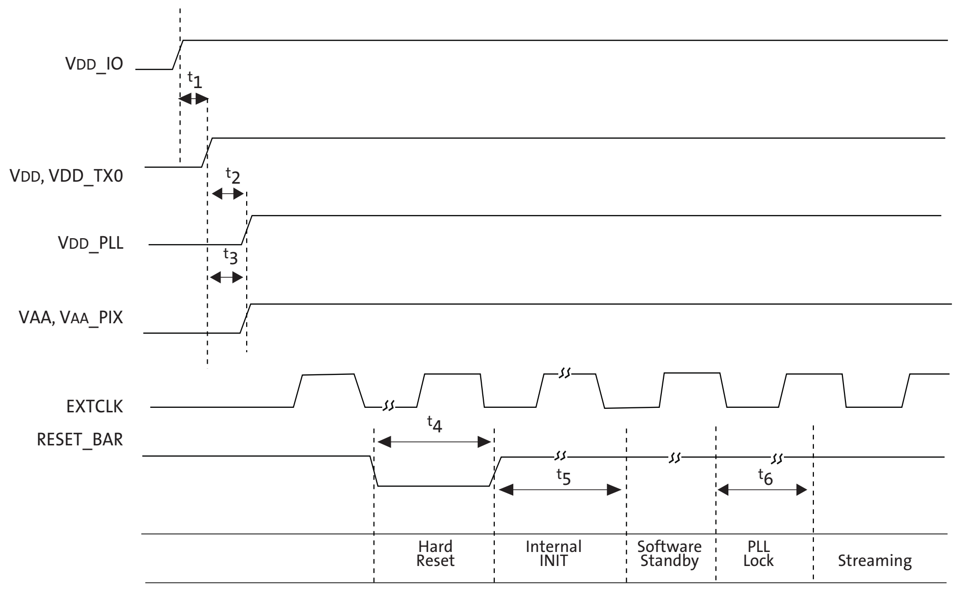


Table 23: Power-Up Sequence

Definition	Symbol	Min	Typ	Max	Unit
VDD_IO to VDD time	t_1	0	–	500	ms
VDD to VDD_PLL time	t_2	0	–	500	ms
VDD to VAA/VAA_PIX time	t_3	0	–	500	ms
Active hard reset	t_4	1	–	–	ms
Internal initialization	t_5	2400	–	–	EXTCLKs
PLL lock time	t_6	1	–	–	ms

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Power-Down Sequence

The recommended power-down sequence for the MT9N001 is shown in Figure 49. The available power supplies (VDD_IO, VDD, VDD_TX0, VDD_PLL, VAA, VAA_PIX) can be turned off at the same time or have the separation specified below.

1. Disable streaming if output is active by setting mode_select = 0 (R0x0100).
2. The soft standby state is reached after the current row or frame, depending on configuration, has ended.
3. After 0–500ms, turn off VDD_IO power supply.
4. After 0–500ms, turn off VDD and VDD_TX0 power supply.
5. Turn off the VAA/VAA_PIX and VDD_PLL power supplies.
6. Assert hard reset by setting RESET_BAR to a logic “0.”

Figure 49: Power-Down Sequence

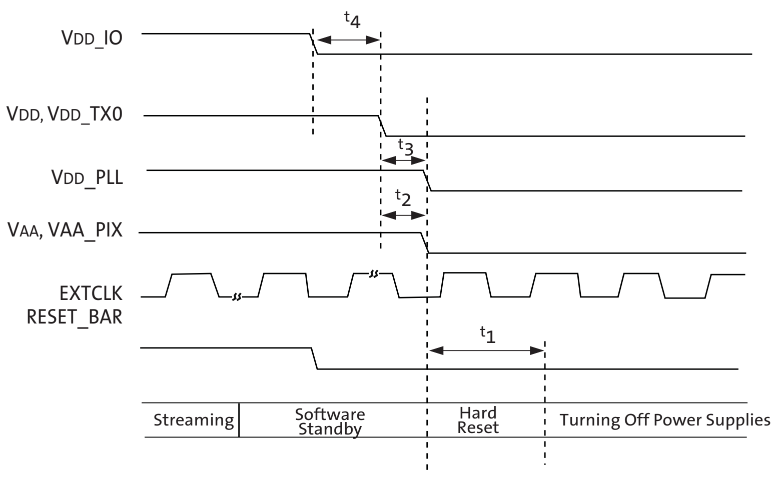


Table 24: Power-Down Sequence

Definition	Symbol	Min	Typ	Max	Unit
Hard reset	t_1	1	–	–	ms
VDD/VAA/VAA_PIX to VDD time	t_2	0	–	500	ms
VDD_PLL to VDD time	t_3	0	–	500	ms
VDD to VDD_IO time	t_4	0	–	500	ms

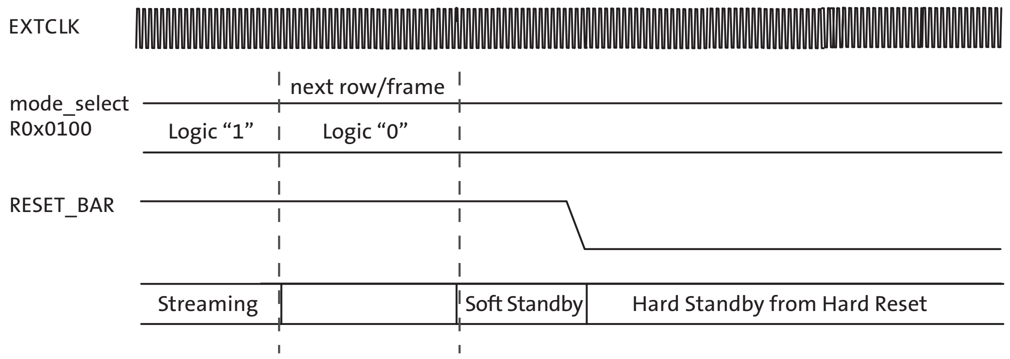
Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Hard Standby and Hard Reset

The hard standby state is reached by the assertion of the RESET_BAR pad (hard reset). Register values are not retained by this action, and will be returned to their default values once hard reset is completed. The minimum power consumption is achieved by the hard standby state. The details of the sequence are described below and shown in Figure 50.

1. Disable streaming if output is active by setting mode_select = 0 (R0x0100).
2. The soft standby state is reached after the current row or frame, depending on configuration, has ended.
3. Assert RESET_BAR (active LOW) to reset the sensor.
4. The sensor remains in hard standby state if RESET_BAR remains in the logic “0” state.

Figure 50: Hard Standby and Hard Reset



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Soft Standby and Soft Reset

The MT9N001 can reduce power consumption by switching to the soft standby state when the output is not needed. Register values are retained in the soft standby state. Once this state is reached, soft reset can be enabled optionally to return all register values back to the default. The details of the sequence are described below and shown in Figure 51.

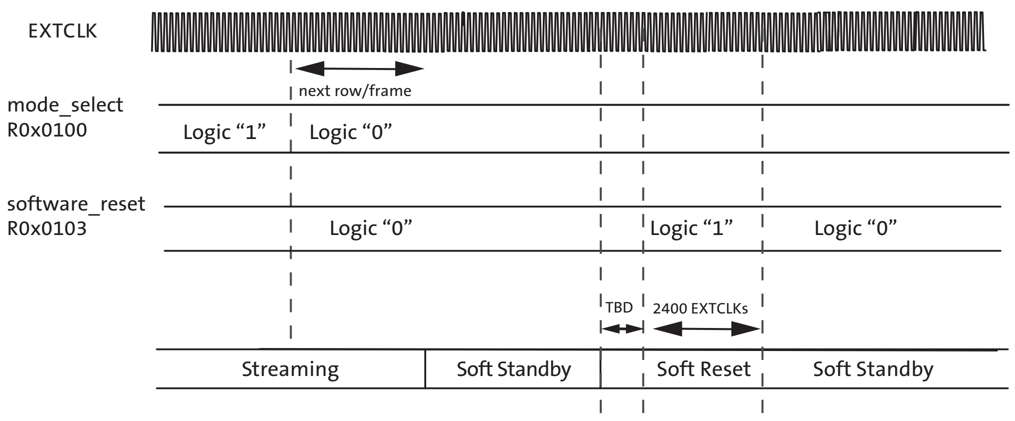
Soft Standby

1. Disable streaming if output is active by setting mode_select = 0 (R0x0100).
2. The soft standby state is reached after the current row or frame, depending on configuration, has ended.

Soft Reset

1. Follow the soft standby sequence listed above.
2. Set software_reset = 1 (R0x0103) to start the internal initialization sequence.
3. After 2400 EXTCLKs (tentative), the internal initialization sequence is completed and the current state returns to soft standby automatically. All registers, including software_reset, returns to their default values.

Figure 51: Soft Standby and Soft Reset

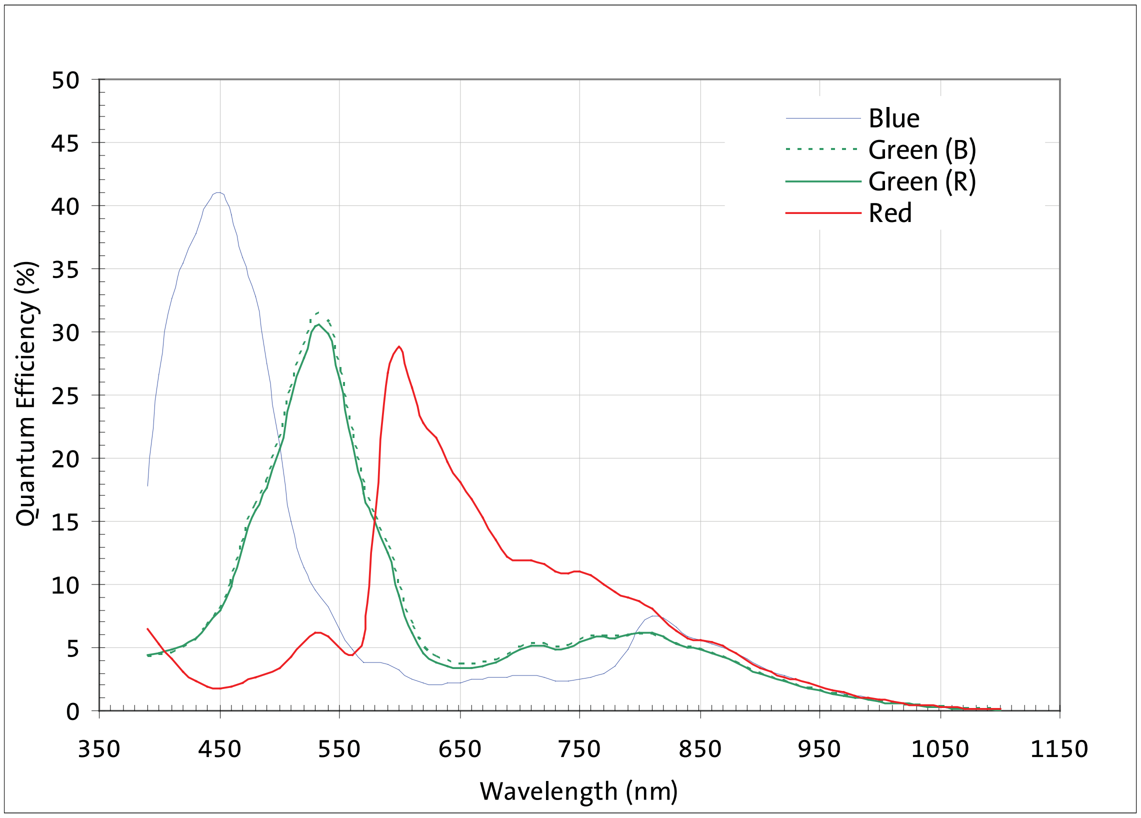


Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Spectral Characteristics

Figure 52: Quantum Efficiency



Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Electrical Characteristics

Table 25: DC Electrical Definitions and Characteristics

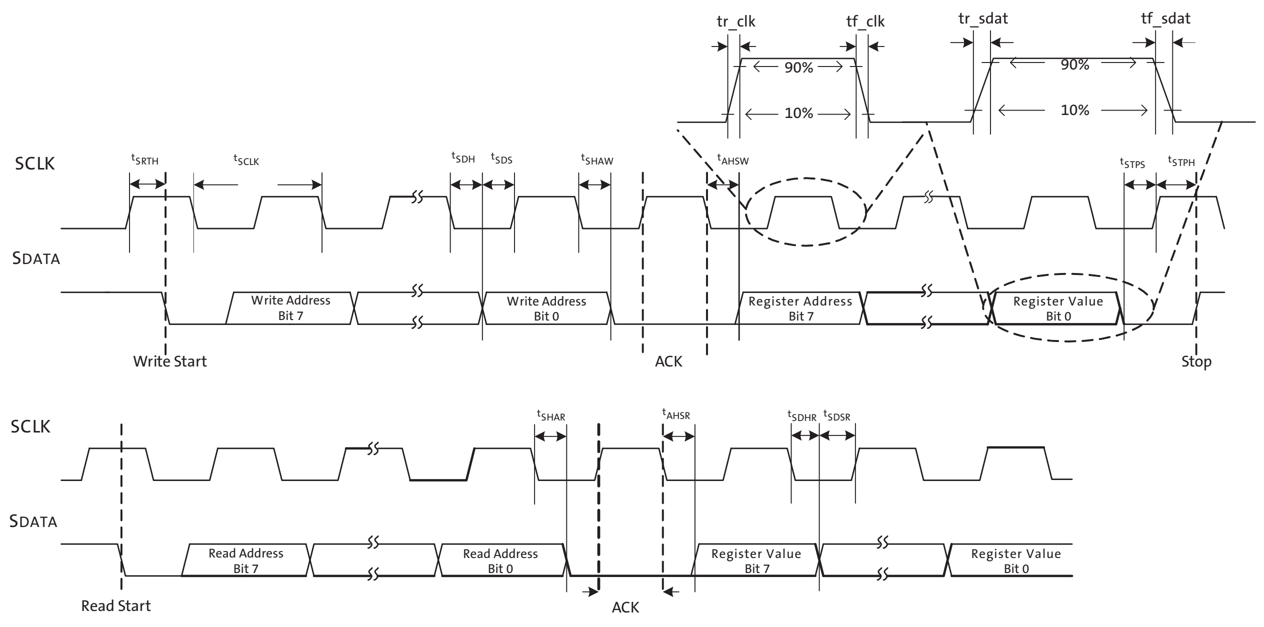
^{f}EXTCLK = 24 MHz; VDD = 1.8V; VDD_IO = 1.8V; VAA = 2.8V; VAA_PIX = 2.8V; VDD_PLL = 2.8V; Output load = 68.5pF

Definition	Condition	Symbol	Min	Typ	Max	Unit
TJ = 25°C						
Core digital voltage		VDD	1.7	1.8	1.9	V
I/O digital voltage	Parallel pixel data interface	VDD_IO	1.7	1.8	1.9	V
Analog voltage		VAA	2.6	2.8	3.1	V
Pixel supply voltage		VAA_PIX	2.6	2.8	3.1	V
PLL supply voltage		VDD_PLL	2.4	2.8	3.1	V
Digital operating current	Streaming, full resolution		75	80	85	mA
I/O digital operating current	Streaming, full resolution		50	60	70	mA
Analog operating current	Streaming, full resolution		135	160	185	mA
Pixel supply current	Streaming, full resolution		2	4	5	mA
PLL supply current	Streaming, full resolution		14	16	18	mA
Hard standby (clock on)	Analog and pixel		100	125	150	µA
	Digital and PLL		50	85	120	µA
Soft standby (clock on)	Analog and pixel		100	125	150	µA
	Digital and PLL		50	85	120	µA
TJ = 60°C						
Core digital voltage		VDD	1.7	1.8	1.9	V
I/O digital voltage	Parallel pixel data interface	VDD_IO	1.7	1.8	1.9	V
Analog voltage		VAA	2.6	1.8	1.9	V
Pixel supply voltage		VAA_PIX	2.6	2.8	3.1	V
PLL supply voltage		VDD_PLL	2.4	2.8	3.1	V
Digital operating current	Streaming, full resolution		70	75	80	mA
I/O digital operating current	Streaming, full resolution		45	55	65	mA
Analog operating current	Streaming, full resolution		130	155	180	mA
Pixel supply current	Streaming, full resolution		2	4	5	mA
PLL supply current	Streaming, full resolution		14	16	18	mA
Hard standby (clock on)	Analog and pixel		90	100	110	µA
	Digital and PLL		85	120	155	µA
Soft standby (clock on)	Analog and pixel		90	100	110	µA
	Digital and PLL		85	120	155	µA

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Figure 53: Two-WIre Serial Bus Timing Parameters



Notes: 1. Read sequence: For an 8-bit READ, read waveforms start after WRITE command and register address are issued.

Table 26: Two-Wire Serial Register Interface Electrical Characteristics

^{f}EXTCLK = 24 MHz; VDD = 1.8V; VDD_IO = 1.8V; VAA = 2.8V; VAA_PIX = 2.8V; VDD_PLL = 2.8V;
Output load = 68.5pF; T_J = 25°C

Definition	Condition	Symbol	Nom	Unit
Input LOW voltage		VIL	0	V
Input leakage current	No pull up resistor VIN = VDD_IO or DGND	IIN	TBD	µA
Output LOW voltage	At specified 3mA	VOL	0	V
Output LOW current	At V_0 drop = 0.4mA	IOL	TBD	mA
Input pad capacitance		CIN	TBD	pF
Load capacitance		CLOAD	TBD	pF

Table 27: I/O Timing

Symbol	Definition	Conditions	Min	Typ	Max	Units
^{f}EXTCLK	Input clock frequency	PLL enabled	6	24	48	MHz
^{t}EXTCLK	Input clock period	PLL enabled	166	41	20	ns
^{t}R	Input clock rise time		0.1	–	1	V/ns
^{t}F	Input clock fall time		0.1	–	1	V/ns
	Clock duty cycle		45	50	55	%
^{t}JITTER	Input clock jitter		–	–	0.3	ns
Output pin slew	Fastest	CLOAD = 15pF	–	0.7	–	V/ns
^{f}PIXCLK	PIXCLK frequency	Default	–	96	–	MHz
^{t}PD	PIXCLK to data valid	Default	–	–	3	ns

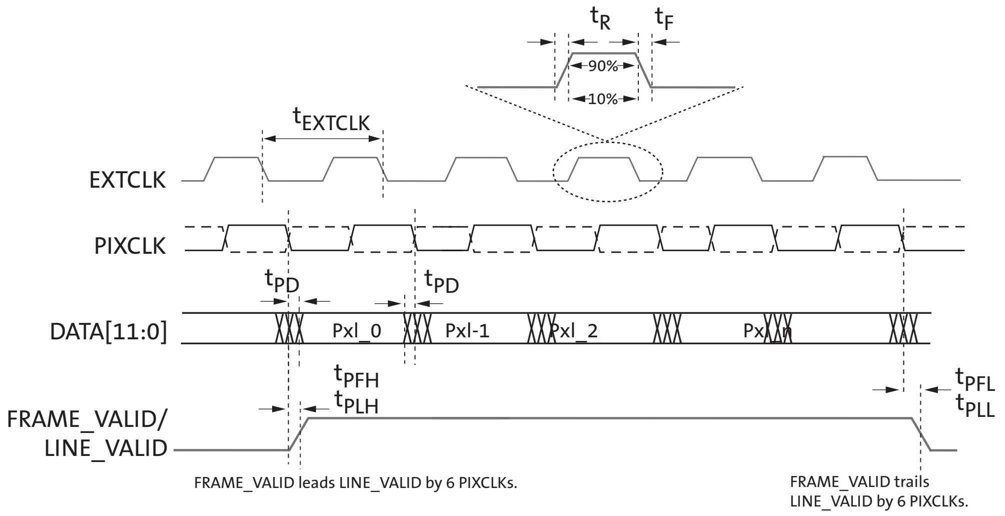
Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Table 27: I/O Timing (Continued)

Symbol	Definition	Conditions	Min	Typ	Max	Units
t_{PFH}	PIXCLK to FRAME_VALID HIGH	Default	–	–	3	ns
t_{PLH}	PIXCLK to LINE_VALID HIGH	Default	–	–	3	ns
t_{PFL}	PIXCLK to FRAME_VALID LOW	Default	–	–	3	ns
t_{PLL}	PIXCLK to LINE_VALID LOW	Default	–	–	3	ns

Figure 54: I/O Timing Diagram

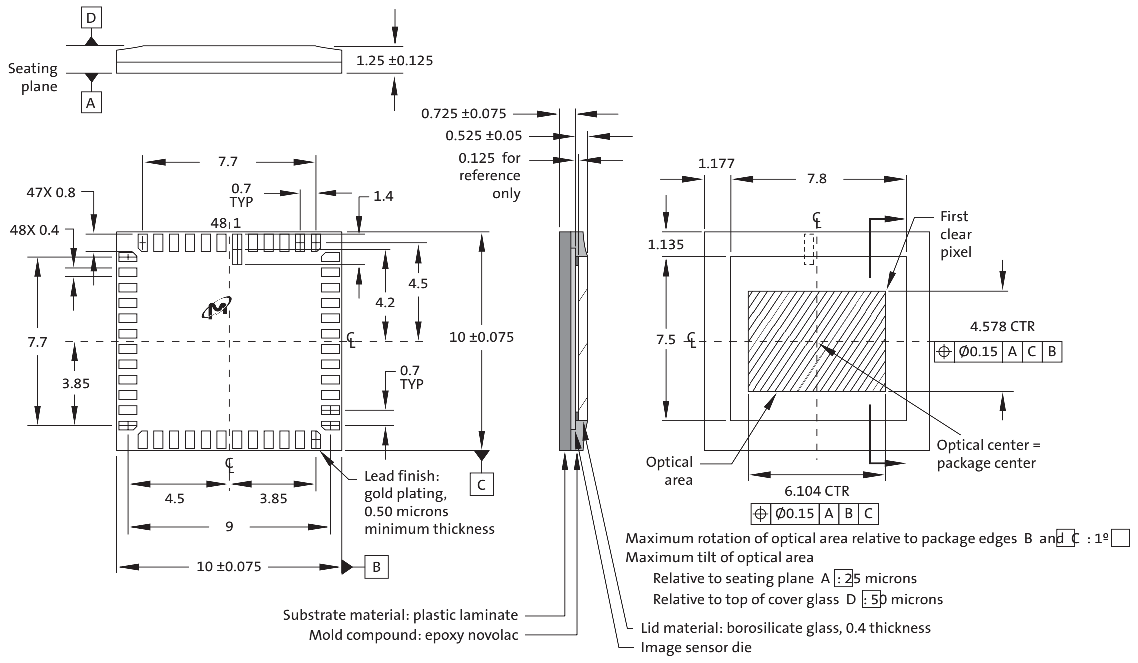


Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Package Dimensions

Figure 55: 48-Pin iLCC Package Outline Drawing


D
Seating plane
A
1.25 ±0.125
0.725 ±0.075
0.525 ±0.05
0.125 for reference only
7.7
0.7 TYP
1.4
47X 0.8
48X 0.4
48 1
4.5
4.2
7.7
3.85
10 ±0.075
0.7 TYP
4.5
3.85
9
10 ±0.075
B
C
Lead finish: gold plating, 0.50 microns minimum thickness
1.177
7.8
1.135
First clear pixel
7.5
4.578 CTR
Ø0.15 A C B
Optical area
Optical center = package center
6.104 CTR
Ø0.15 A B C
Maximum rotation of optical area relative to package edges B and C : 1º
Maximum tilt of optical area
Relative to seating plane A : 25 microns
Relative to top of cover glass D : 50 microns
Substrate material: plastic laminate
Mold compound: epoxy novolac
Lid material: borosilicate glass, 0.4 thickness
Image sensor die

Notes: 1. All dimensions in millimeters.

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

SMIA and MIPI Specification Reference

The sensor design and this documentation is based on the following reference documents:

1. SMIA Specifications:
   - Functional Specification:
     SMIA 1.0 Part 1: Functional Specification (Version 1.0 dated 30 June 2004)
     SMIA 1.0 Part 1: Functional Specification ECR0001 (Version 1.0 dated 11 Feb 2005)
   - Electrical Specification:
     SMIA 1.0 Part 2: CCP2 Specification (Version 1.0 dated 30 June 2004)
     SMIA 1.0 Part 2: CCP2 Specification ECR0001 (Version 1.0 dated 11 Feb 2005)
2. MIPI Specifications:
   - MIPI Alliance Standard for CSI-2 version 1.0
   - MIPI Alliance Standard for D-PHY version 0.81

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

Aptina Confidential and Proprietary

Revision History

Rev. D **1/20/10**

- Updated to Aptina template
- Moved register tables to a separate document (MT9N001 Register Reference)

Rev. C, Production **04/10/08**

- Update "Features" on page 1
- Update "Applications" on page 1
- Update Table 1, "Key Performance Parameters," on page 1
- Update "Operating Modes" on page 9
- Update Figure 5: "Typical Configuration: CCP2 Parallel Pixel Data Interface," on page 12
- Update Figure 7: "48-Pin iLCC Package Pinout Diagram," on page 15
- Update Table 3, "Signal Descriptions," on page 14
- Update Figure 7: "48-Pin iLCC Package Pinout Diagram," on page 15
- Update Table 12, "Register Description—Manufacturer-Specific," on page 40
- Add "Scaler" on page 39
- Update "Binning" on page 52
- Update "Power-Down Sequence" on page 72
- Update "Spectral Characteristics" on page 75
- Update "Quantum Efficiency" on page 75
- Update "Electrical Characteristics" on page 76

Rev. B, Preliminary **10/07**

- Update "Features" on page 1
- Update Table 1, "Key Performance Parameters," on page 1
- Update "Operating Modes" on page 9
- Update Figure 3: "Typical Configuration: Serial CCP2 Pixel Data Interface," on page 10
- Update Figure 4: "Typical Configuration: Serial Two-Lane MIPI Pixel Data Interface," on page 11
- Update Figure 5: "Typical Configuration: CCP2 Parallel Pixel Data Interface," on page 12
- Update Figure 7: "48-Pin iLCC Package Pinout Diagram," on page 15
- Add Table 7, "Register List and Default—SMIA Configuration," on page 21
- Add Table 8, "Register List and Default Values—SMIA Parameter Limits," on page 24
- Add Table 9, "Register List and Default Values—Manufacturer-Specific," on page 26
- Add See "Register Descriptions" on page 31
- Add Table 10, "Register Description—SMIA Configuration," on page 31
- Add Table 11, "Register Description—SMIA Parameter Limits," on page 36
- Add Table 12, "Register Description—Manufacturer-Specific," on page 40
- Add "Programming Restrictions" on page 24
- Update Table 5, "Definitions for Programming Rules," on page 24
- Update "Programming Restrictions when Subsampling" on page 50
- Update Table 17, "Minimum Row Time and Blanking Numbers," on page 56
- Update Table 19, "fine_integration_time Limits," on page 57
- Update Table 20, "fine_correction Values," on page 57
- Update "Low Power Mode" on page 58
- Add "Package Dimensions" on page 79

Aptina reserves the right to change products or specifications without notice.
©2007 Aptina Imaging Corporation. All rights reserved.

- Add Figure 55: “48-Pin iLCC Package Outline Drawing,” on page 79

Rev. A, Advance 08/13/07

- Initial release

10 Eunos Road 8 13-40, Singapore Post Center, Singapore 408600 prodmktg@aptina.com www.aptina.com
Aptina, Aptina Imaging, DigitalClarity, Osmium, and the Aptina logo are the property of Aptina Imaging Corporation
All other trademarks are the property of their respective owners.
This data sheet contains minimum and maximum limits specified over the power supply and temperature range set forth herein. Although considered final, these specifications are subject to change, as further product development and data characterization sometimes occur.