1/3-Inch CMOS Digital Image Sensor

MT9M021/MT9M031 Developer Guide

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

MT9M021/MT9M031
Developer Guide

Table of Contents

Introduction 6
Optimal Setting Guidelines 6
Resolution 6
Frame Rate 6
Blanking Control 6
Pixel Data Format 6
Pixel Array Structure 6
Default Readout Order 8
Output Data Format 9
Readout Sequence 9
Parallel Output Data Timing 10
LV and FV 10
LV Format Options 10
High Speed Serial Pixel Interface 11
HiSPi Physical Layer 12
Frame Time 13
Exposure 14
Exposure Indicator 15
Real-Time Context Switching 16
Features 17
Operational Modes 17
Master Mode 17
Trigger Mode 18
Reset 26
Hard Reset of Logic 26
Soft Reset of Logic 26
Output Enable 26
Clocks 26
PLL-Generated Master Clock 26
Spread-Spectrum Clocking 28
Stream/Standby Control 28
Soft Standby 28
Hard Standby 29
Readout Modes 29
Digital Binning 29
Skipping 29
Mirror 32
Column Mirror Image 32
Row Mirror Image 32
Maintaining a Constant Frame Rate 33
Synchronizing Register Writes to Frame Boundaries 33
Restart 34
Temperature Sensor 34
Auto Exposure 35
Auto Exposure Implementation 35
AE Embedded Statistics and Data 37
AE Target Selection 38
Exposure Control System 38
Auto Exposure Control 40
AE Frame Synchronization 45
Gain 46

Digital Gain 46
Column Gain 46
Black Level Correction 46
Row-wise Noise Correction 46
Column Correction 47
Column Correction Triggering 47
Column Triggering on Startup 47
Column Correction Retriggering Due to Mode Change 47
Test Patterns 48
Color Field 48
Vertical Color Bars 48
Walking 1s 49
Two-Wire Serial Register Interface 50
Protocol 50
Start Condition 50
Stop Condition 50
Data Transfer 50
Slave Address/Data Direction Byte 50
Message Byte 51
Acknowledge Bit 51
No-Acknowledge Bit 51
Typical Sequence 51
Single READ from Random Location 52
Single READ from Current Location 52
Sequential READ, Start from Random Location 52
Sequential READ, Start from Current Location 53
Single WRITE to Random Location 53
Sequential WRITE, Start at Random Location 53
Revision History 54

List of Figures

Figure 1: Pixel Array Description . . . 7
Figure 2: Pixel Color Pattern Detail (Top Right Corner) . . . 8
Figure 3: Imaging a Scene . . . 8
Figure 4: Spatial Illustration of Image Readout . . . 9
Figure 5: Default Pixel Output Timing . . . 10
Figure 6: LV Format Options . . . 11
Figure 7: HiSPi Transmitter and Receiver Interface Block Diagram . . . 12
Figure 8: Timing Diagram . . . 12
Figure 9: Line Timing and FRAME_VALID/LINE_VALID Signals . . . 13
Figure 10: Latency For Single Buffered Registers . . . 15
Figure 11: Latency For Double Buffered Registers . . . 15
Figure 12: Master Mode Synchronization Waveform #1 . . . 17
Figure 13: Master Mode Synchronization Waveform #2 . . . 17
Figure 14: Block Diagram . . . 18
Figure 15: Pulsed Trigger Mode . . . 19
Figure 16: Automatic Trigger Mode . . . 19
Figure 17: PLL Block Diagram . . . 26
Figure 18: Pixel Readout (no skipping) . . . 30
Figure 19: Pixel Readout (Row Skip 2X Bayer) . . . 30
Figure 20: Pixel Readout (Row Skip 2X Monochrome) . . . 31
Figure 21: Six Pixels in Normal and Column Mirror Readout Modes . . . 32
Figure 22: Six Rows in Normal and Row Mirror Readout Modes . . . 32
Figure 23: Calculating Temperature Sensor Value . . . 34
Figure 24: AE Block Diagram . . . 36
Figure 25: AE Stats Calculation Block . . . 37
Figure 26: Embedded Statistics Format . . . 38
Figure 27: Exposure Control System . . . 39
Figure 28: Selecting the ROI . . . 41
Figure 29: Automatic Gain Control . . . 42
Figure 30: AE Frame Synchronization . . . 45
Figure 31: Solid Color . . . 48
Figure 32: Vertical Corlor Bars . . . 49
Figure 33: Walking 1s . . . 49
Figure 34: Single READ from Random Location . . . 52
Figure 35: Single READ from Current Location . . . 52
Figure 36: Sequential READ, Start from Random Location . . . 52
Figure 37: Sequential READ, Start from Current Location . . . 53
Figure 38: Single WRITE to Random Location . . . 53
Figure 39: Sequential WRITE, Start at Random Location . . . 53

List of Tables

Table 1: MT9M021/MT9M031 HiSPi Protocol Configuration Settings 11
Table 2: Frame Time (Example Based on 1280 x 960, 45 Frames Per Second) 13
Table 3: Frame Time: Long Integration Time 14
Table 4: Real-Time Context-Switchable Registers 16
Table 5: Exposure Timing 19
Table 6: Snapshot Mode Register Settings 20
Table 7: Example 1 (With Default Setting for Full Resolution) 23
Table 8: Example 2 (With Default Settings for 720p) 24
Table 9: PLL Parameters for the Parallel Interface 27
Table 10: Example PLL Configuration for the Parallel Interface 27
Table 11: PLL Parameters for the Serial Interface 27
Table 12: Example PLL Configurations for the Serial Interface 28
Table 13: Skip Mode Settings 30
Table 14: Exposure Control Variables 40
Table 15: AE Enable Registers 40
Table 16: Auto Exposure Control Registers 44
Table 17: Test Pattern Modes 48

Introduction

This Developer Guide provides detailed descriptions and usage guidelines for various features of the MT9M021/MT9M031 Global Shutter Sensor. Also provided are guidelines for optimal settings for various use cases. For detailed electrical and timing specifications or register descriptions, refer to the MT9M021/MT9M031 Data Sheet and the MT9M021/MT9M031 Register Reference documents, respectively.

Optimal Setting Guidelines

The MT9M021/MT9M031 Global Shutter Sensor has many built-in features and is capable of many resolutions and frame rates. Guidelines for setting resolution and frame rate are provided in this section. Detailed settings for the many features are provided throughout the remainder of this Developer Guide. Window registers are also provided to enable context switching. See the section on Real-Time Context Switching and the register reference guide for more details.

Resolution

Aptina's MT9M021/MT9M031 sensor is capable of a maximum resolution of 1280 x 960 at up to 45fps, or it may be configured to run 720p at 60fps. Registers y_addr_start, x_addr_start, y_addr_end, and x_addr_end are used to specify the image window. The minimum value for x_addr_start is 0 and the maximum value for x_addr_end is 1279. The minimum y_addr_start and maximum y_addr_end are 0 and 975, respectively.

Frame Rate

Achieving the desired frame rate at the proper resolution is a balancing act between row timing and the number of rows in the image. Integration time and the pixel clock frequency are additional factors. The minimum line length is 1650 pixel clocks.

Blanking Control

Horizontal blanking and vertical blanking times are controlled by the Line_length_Pck and Frame_Length_Lines registers, respectively.

- Horizontal blanking is specified in terms of pixel clocks. It is calculated by subtracting the X window size from the Line_Length_Pck register. The minimum horizontal blanking is 370 pixel clocks.
- Vertical blanking is specified in terms of numbers of lines. It is calculated by subtracting the Y window size from the Frame_Length_Lines register. The minimum value for vertical blanking is 30 lines.

The actual imager timing is described in the Frame Time section of this Developer Guide.

Pixel Data Format

Pixel Array Structure

The MT9M021/MT9M031 pixel array is configured as 1412 columns by 1028 rows, (see Figure 1). The dark pixels are optically black and are used internally to monitor black level. Of the right 108 columns, 64 are dark pixels used for row noise correction. Of the top 24 rows of pixels, 12 of the dark rows are used for black level correction. There are

1296 columns by 976 rows of optically active pixels. While the sensor's format is 1280 x 960, the additional active columns and active rows are included for use when horizontal or vertical mirrored readout is enabled, to allow readout to start on the same pixel. The pixel adjustment is always performed for monochrome or color versions. The active area is surrounded with optically transparent dummy pixels to improve image uniformity within the active area. Not all dummy pixels or barrier pixels can be read out.

Figure 1: Pixel Array Description

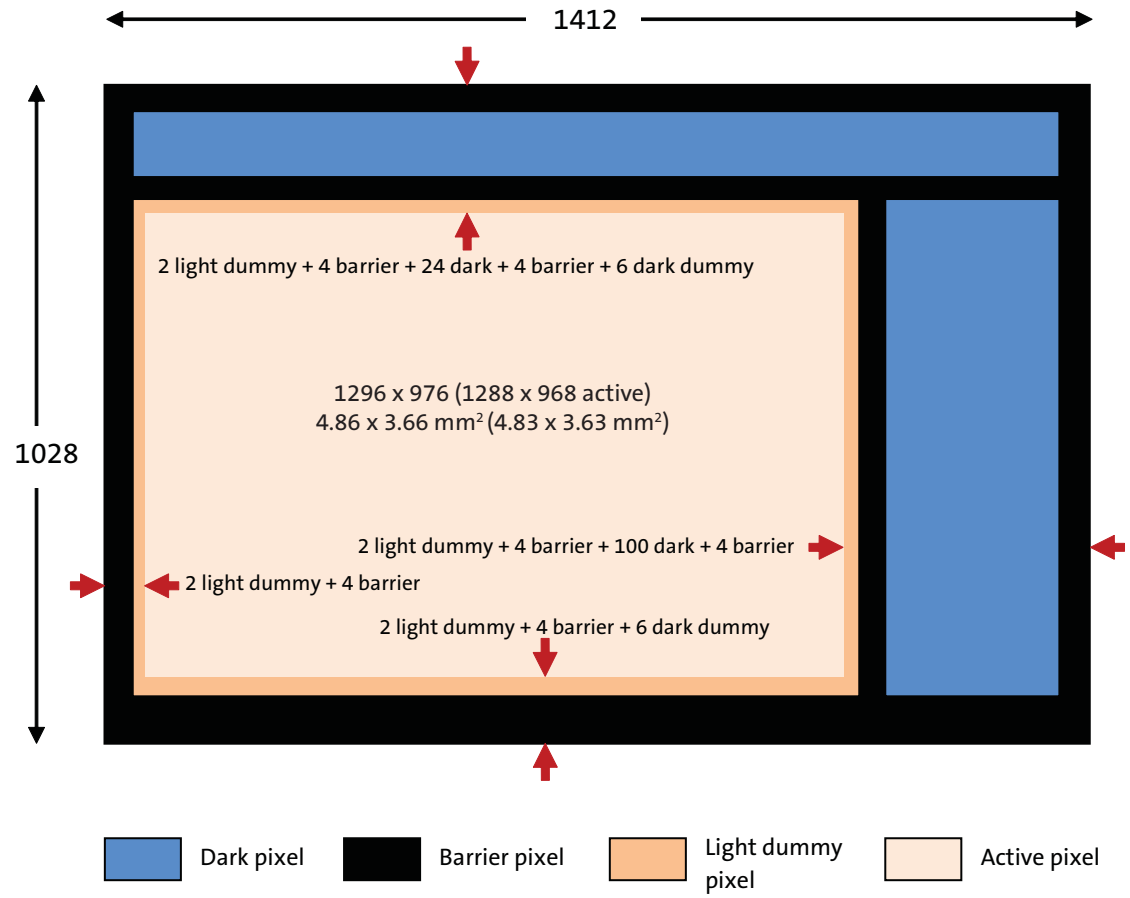
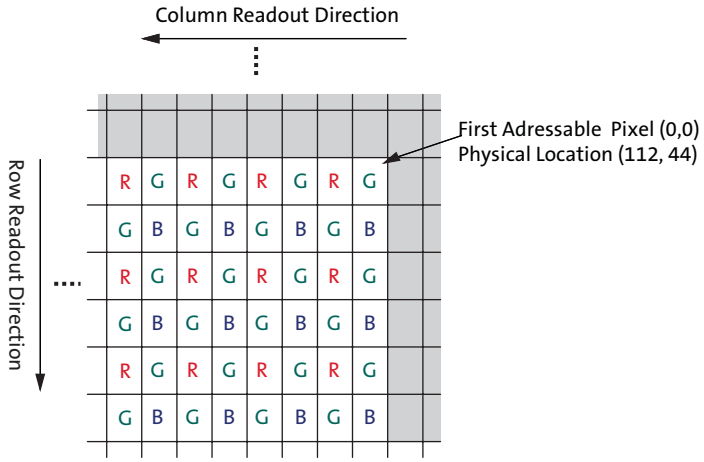


Figure 2: Pixel Color Pattern Detail (Top Right Corner)

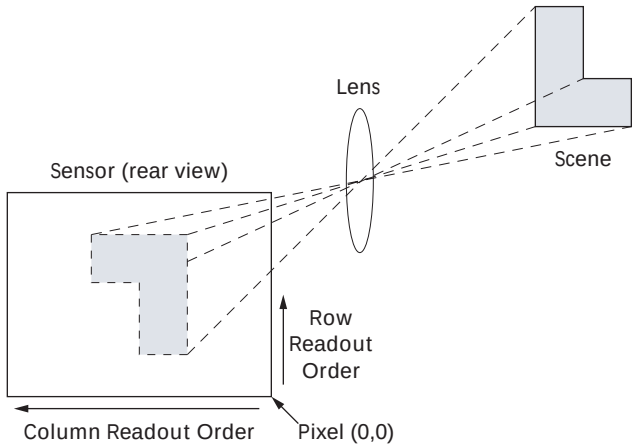


Default Readout Order

By convention, the sensor core pixel array is shown with the first addressable (logical) pixel (0,0) in the top right corner (see Figure 2). This reflects the actual layout of the array on the die. Also, the physical location of the first pixel data read out of the sensor in default condition is that of pixel (112, 44).

When the sensor is imaging, the active surface of the sensor faces the scene as shown in Figure 3. When the image is read out of the sensor, it is read one row at a time, with the rows and columns sequenced as shown in Figure 3 on page 8.

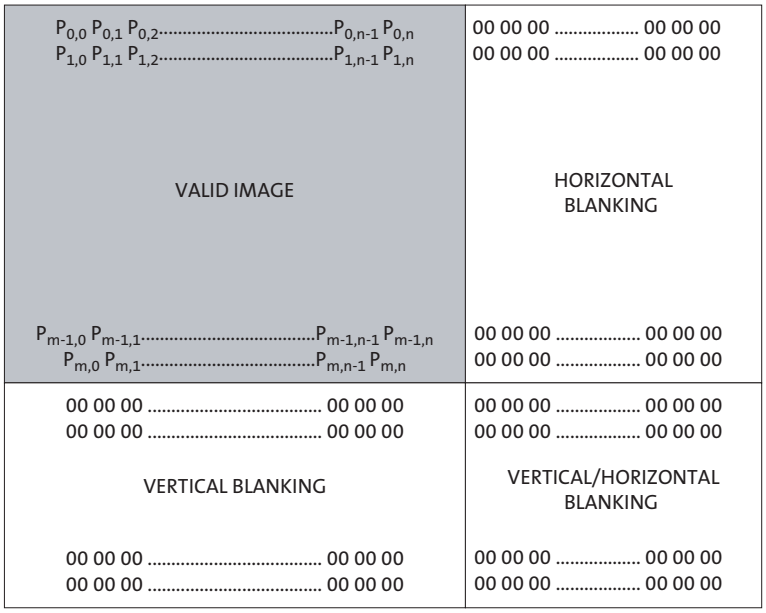
Figure 3: Imaging a Scene



Output Data Format

The MT9M021/MT9M031 image data is read out in a progressive scan. Valid image data is surrounded by horizontal and vertical blanking (see Figure 4). The amount of horizontal row time (in clocks) is programmable through R0x300C. The amount of vertical frame time (in rows) is programmable through R0x300A. Line_Valid (LV) is HIGH during the shaded region of Figure 4. Optional embedded register setup information and histogram statistic information are available in the first two and the last two rows of image data.

Figure 4: Spatial Illustration of Image Readout



Readout Sequence

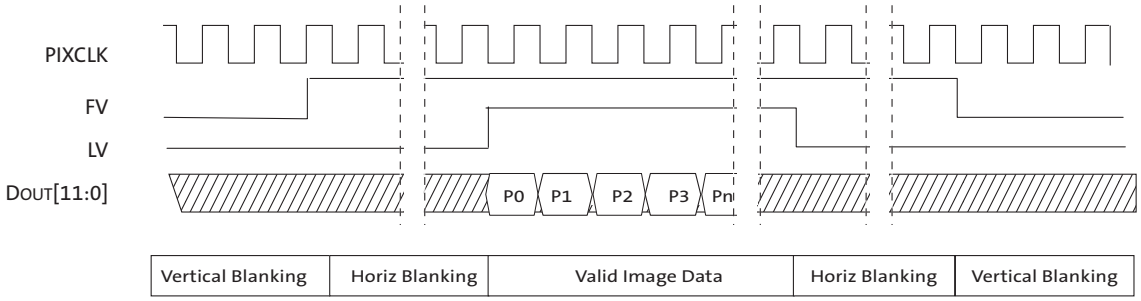
Typically, the readout window is set to a region including only active pixels. The user has the option of reading out dark regions of the array, but if this is done, consideration must be given to how the sensor reads the dark regions for its own purposes.

Parallel Output Data Timing

The output images are divided into frames, which are further divided into lines. By default, the sensor produces 964 rows of 1280 columns each. The FV and LV signals indicate the boundaries between frames and lines, respectively. PIXCLK can be used as a clock to latch the data. One 12-bit pixel datum is output on the DOUT pins for each falling edge of PIXCLK. The launch edge of PIXCLK may be set in register R0x3028. When both FV and LV are asserted, the pixel is valid. PIXCLK cycles that occur when FV is de-asserted are called vertical blanking. PIXCLK cycles that occur when only LV is de-asserted are called horizontal blanking.

To enable the parallel output pins, set R0x301A[7] = 1, and set R0x301A[12] = 1 to disable the HiSPi serializer. The parallel input pins (i.e. TRIGGER, STANDBY, etc) may be enabled by setting R0x301A[8] = 1.

Figure 5: Default Pixel Output Timing



LV and FV

The timing of the FV and LV outputs is closely related to the row time and the frame time.

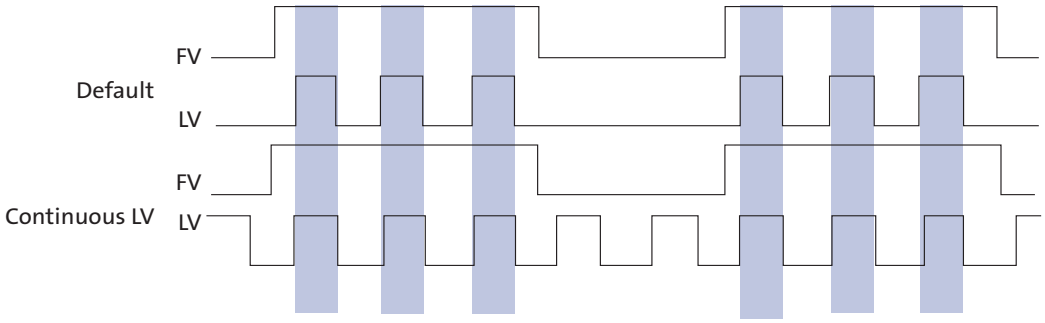
FV will be asserted for an integral number of row times, which will normally be equal to the height of the output image.

LV will be asserted during the valid pixels of each row. The leading edge of LV will be offset from the leading edge of FV by 6 PIXCLKs. Normally, LV will only be asserted if FV is asserted; this is configurable as described below.

LV Format Options

The default situation (R0x306E[1:0] = 0x0) is for LV to be de-asserted when FV is de-asserted. By setting R0x306E[1:0]= 0x1, a continuous LV signal will be output. The formats for reading out four lines and two vertical blanking lines are shown in Figure 6.

Figure 6: LV Format Options



The timing of an entire frame is shown in Figure 9: “Line Timing and FRAME_VALID/LINE_VALID Signals,” on page 13. For detailed timing diagrams and switching parameters, refer to the MT9M021/MT9M031 data sheet.

High Speed Serial Pixel Interface

The MT9M021/MT9M031 also uses Aptina's High-Speed Serial Pixel Interface (HiSPi™). The MT9M021/MT9M031 HiSPi interface supports two protocols, Streaming-SP, and Packetized SP. The streaming protocol conforms to a standard video application where each line of active or intra-frame blanking provided by the sensor is transmitted at the same length. The Packetized SP protocol will transmit only the active data ignoring line-to-line and frame-to-frame blanking data. Refer to Table 1 for HiSPi protocol and lane settings.

Table 1: MT9M021/MT9M031 HiSPi Protocol Configuration Settings

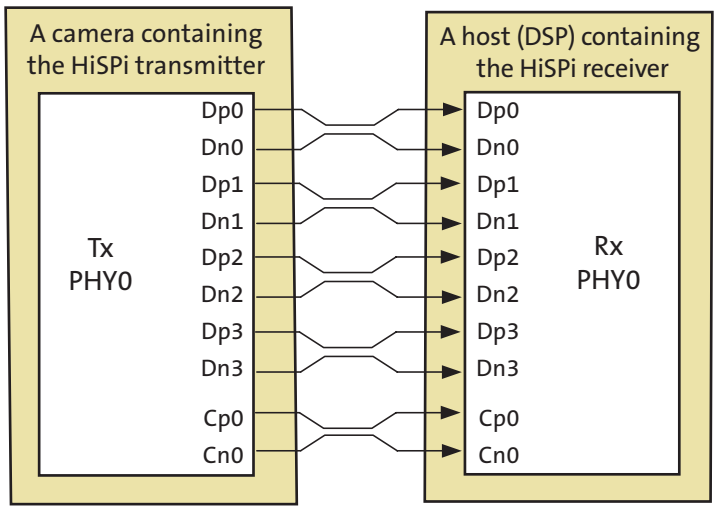
Protocol	R0x31C6
Streaming-SP 2 Lane	0x0004
Streaming-SP 3 Lane	0x000C
Packetized-SP 2 Lane	0x0000
Packetized-SP 3 Lane	0x0008

These protocols are further described in the High-Speed Serial Pixel (HiSPi™) Interface Protocol Specification V1.50.00.

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

To enable the serial interface, set R0x301A[7] = 0, and set R0x301A[12] = 0 to enable the HiSPi serializer. Refer to “Clocks” on page 26 for PLL configuration when using the HiSPi interface.

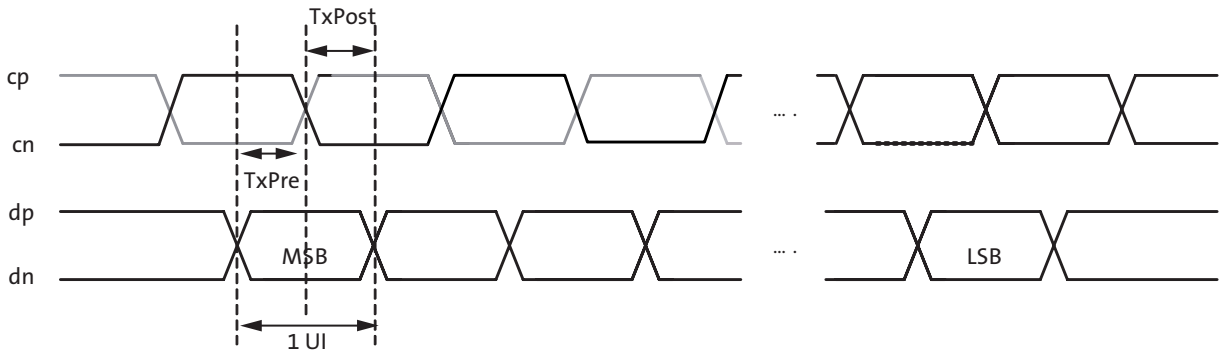
Figure 7: HiSPi Transmitter and Receiver Interface Block Diagram



HiSPi Physical Layer

The HiSPi physical layer is partitioned into blocks of four data lanes and an associated clock lane. Depending on the operating mode and data rate, it can be configured from two to three lanes. Dp3 and Dn3 are not supported by the MT9M021/MT9M031 but pins are connected on the package. The PHY will serialize a 12-bit data word and transmit on both edges of the clock. Figure 8 shows bit transmission. In this example, the word is transmitted in order of MSB to LSB. The receiver latches data at the rising and falling edge of the clock. The MT9M021/MT9M031 supports only the SLVS mode of the HiSPi electrical specification. For detailed timing and electrical specifications for the HiSPi interface, refer to the MT9M021/MT9M031 Datasheet.

Figure 8: Timing Diagram



Frame Time

The pixel clock (PIXCLK) represents the time needed to sample one pixel from the array. The sensor outputs data at the maximum rate of one pixel per PIXCLK. One row time (^{t}ROW) is the period from the first pixel output in a row to the first pixel output in the next row. The row time and frame time are defined by equations in Table 2.

Figure 9: Line Timing and FRAME_VALID/LINE_VALID Signals

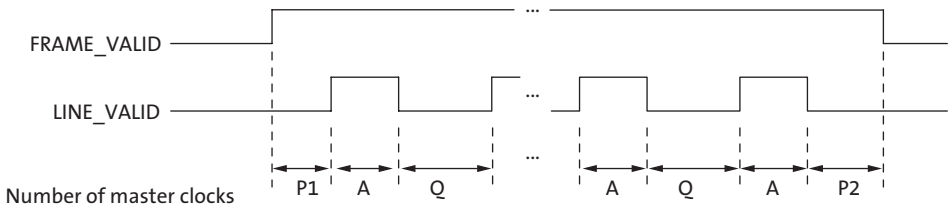


Table 2: Frame Time (Example Based on 1280 x 960, 45 Frames Per Second)

Parameter	Name	Equation	Default Timing at 74.25 MHz
A	Active data time	Context A: R0x3008 - R0x3004 + 1 Context B: R0x308E - R0x308A + 1	1280 pixel clocks = 17.23µs
P1	Frame start blanking	6 (fixed)	6 pixel clocks = 0.08µs
P2	Frame end blanking	6 (fixed)	6 pixel clocks = 0.08µs
Q	Horizontal blanking	R0x300C - A	370 pixel clocks = 4.98µs
A+Q	Row Time (t_{ROW})	R0x300C	1650 pixel clocks = 22.22µs
V	Vertical blanking	Context A: ((R0x300A(R0x3006-R0x3002+1) * (A + Q)) Context B: ((R0x300A(R0x3090-R0x308C+1)) * (A + Q))	49,500 pixel clocks = 666.6µs
Nrows * (A + Q)	Frame valid time	Context A: ((R0x3006-R0x3002+1)*(A+Q))-Q+P1+P2 Context B: ((R0x3090-R0x308C+1)*(A+Q))-Q+P1+P2	1,584,000 pixel clocks = 21.33ms
F	Total frame time	V + (Nrows * (A + Q))	1,633,500 pixel clocks = 22.2ms

Sensor timing is shown in terms of pixel clock cycles (see Figure 5: "Default Pixel Output Timing," on page 10). The recommended pixel clock frequency is 74.25 MHz. The vertical blanking and the total frame time equations assume that the integration time (coarse integration time plus fine integration time) is less than the number of active lines plus the blanking lines:

Coarse Integration Time < Window Height + Vertical Blanking *(EQ 1)*

If this is not the case, the number of integration lines must be used instead to determine the frame time, (see Table 3). In this example, it is assumed that the coarse integration time control is programmed with 2000 rows and the fine shutter width total is zero.

For Master mode, if the integration time registers exceed the total readout time, then the vertical blanking time is internally extended automatically to adjust for the additional integration time required. This extended value is **not** written back to the

frame_length_lines register. The frame_length_lines register can be used to adjust frame-to-frame readout time. This register does not affect the exposure time but it may extend the readout time.

Table 3: Frame Time: Long Integration Time

Parameter	Name	Equation (Number of Pixel Clock Cycles)	Default Timing at 74.25 MHz
F'	Total frame time (long integration time)	Context A: (R0x3012 * (A + Q)) - R0x3014 + P1 + P2 Context B: (R0x3016 * (A + Q)) - R0x3018 + P1 + P2	3,300,012 pixel clocks = 44.44ms

Note: The MT9M021/MT9M031 uses column parallel analog-digital converters; thus short line timing is not possible. The minimum total line time is 1650 columns (horizontal width + horizontal blanking). The minimum horizontal blanking is 370.

Exposure

Total integration time is the result of coarse_integration_time and fine_integration_time registers, and depends also on whether manual or automatic exposure is selected.

The actual total integration time, t_{INT} is defined as:

$$t_{INT} = t_{INTCoarse} + t_{INTFine} \quad \text{(EQ 2)}$$

= (number of lines of integration * line time) + (number of pixels of integration * pixel time)

where:

– Number of Lines of Integration (Auto Exposure Control: Enabled)
When automatic exposure control (AEC) is enabled, the number of lines of integration may vary from frame to frame, with the limits controlled by R0x311E (minimum auto exposure time) and R0x311C (maximum auto exposure time).

– Number of Lines of Integration (Auto Exposure Control: Disabled)
If AEC is disabled, the number of lines of integration equals the value in R0x3012 (context A) or R0x3016 (context B).

– Number of Pixels of Integration (AEC Disabled. Fine integration time is not used for AEC)
Context A: the number of pixels of integration equals the value in R0x3014.
Context B: the number of pixels of integration equals the value in R0x3018.

Maximum value for $t_{INTFine}$ is line_length_pck - 750.

Typically, the value of the coarse_integration_time register is limited to the number of lines per frame (which includes vertical blanking lines), such that the frame rate is not affected by the integration time. Figure 10 and Figure 11 on page 15 show the frame latency for changes to single and double buffered registers, respectively. Refer to the Register Reference document to determine register latency.

Figure 10: Latency For Single Buffered Registers

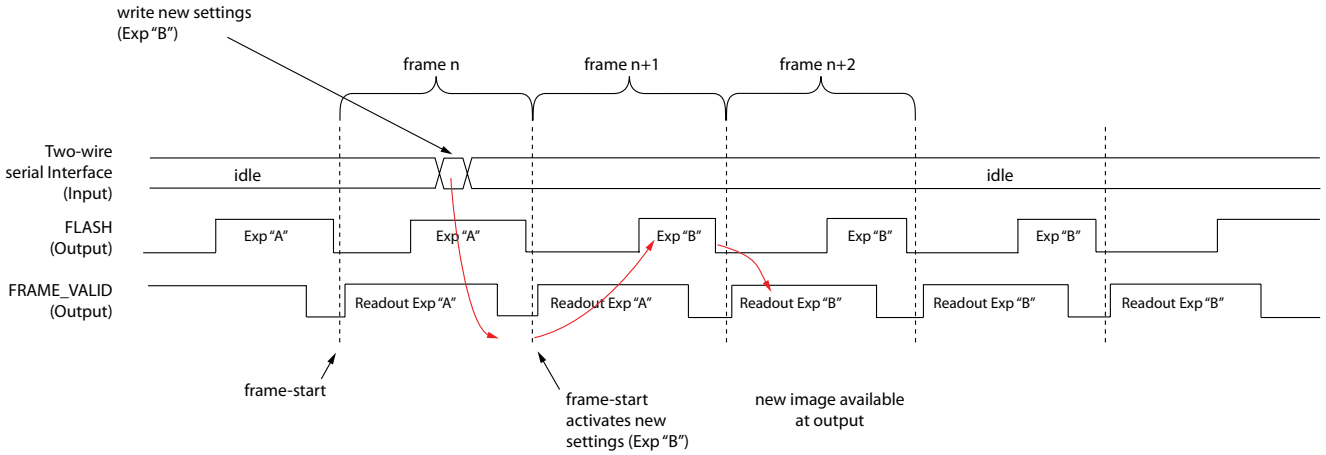
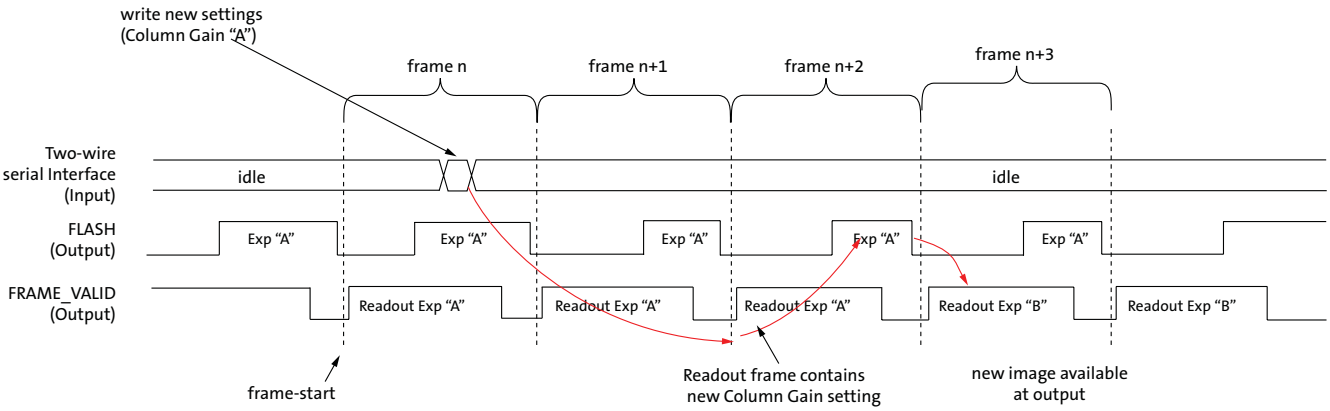


Figure 11: Latency For Double Buffered Registers



Exposure Indicator

The MT9M021/MT9M031 provides an output pin, FLASH, to indicate when the exposure takes place. When R0x3046[8]is set, FLASH is HIGH during exposure. By using R0x3046[7], the polarity of the FLASH pin can be inverted.

Real-Time Context Switching

In the MT9M021/MT9M031, the user may switch between two full register sets (listed in Table 4) by writing to a context switch change bit in R0x30B0[13]. This context switch will change all registers (no shadowing) at the frame start time and have the new values apply to the immediate next exposure and readout time.

Table 4: Real-Time Context-Switchable Registers

Register Description	Register Number	
	Context A	Context B
y_addr_start	R0x3002	R0x308C
x_addr_start	R0x3004	R0x308A
y_addr_end	R0x3006	R0x3090
x_addr_end	R0x3008	R0x308E
coarse_integration_time	R0x3012	R0x3016
fine_integration_time	R0x3014	R0x3018
y_odd_Inc	R0x30A6	R0x30A8
green1_gain (GreenR)	R0x3056	R0x30BC
blue_gain	R0x3058	R0x30BE
red_gain	R0x305A	R0x30C0
green2_gain (GreenB)	R0x305C	R0x30C2
global_gain	R0x305E	R0x30C4
frame_length_lines	R0x300A	R0x30AA
digital_binning	R0x3032[1:0]	R0x3032[5:4]

Features

Note: See the MT9M021/MT9M031 Register Reference for additional details.

Operational Modes

The MT9M021/MT9M031 works in master (video) or trigger (single frame) modes. In master mode, the sensor generates the integration and readout timing. In trigger mode, it accepts an external trigger to start exposure, then generates the exposure and readout timing. The exposure time is programmed through the two-wire serial interface for both modes.

Note: Trigger mode is not compatible with the HiSPi interface.

Master Mode

In master mode, the exposure period occurs simultaneously with the frame readout (see Figures 12 and 13). This makes master mode the fastest mode of operation. When exposure time is greater than the frame length, the number of vertical blanking rows is increased automatically to accommodate the exposure time.

Figure 12: Master Mode Synchronization Waveform #1

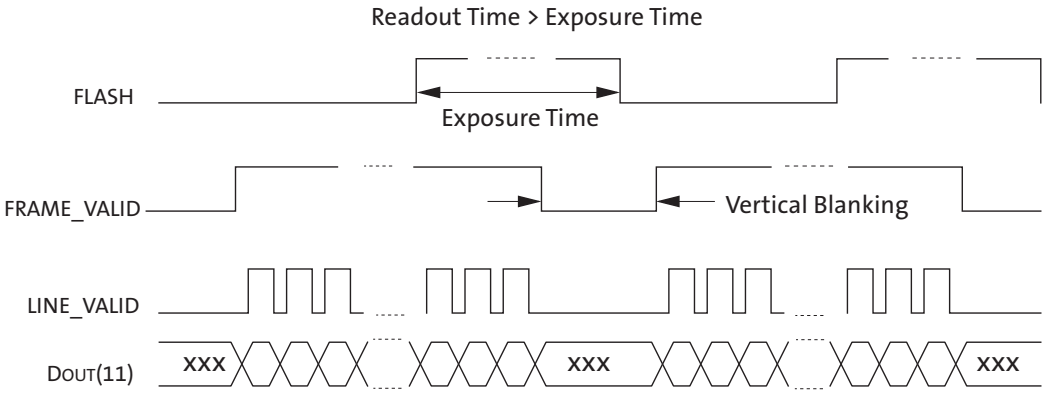
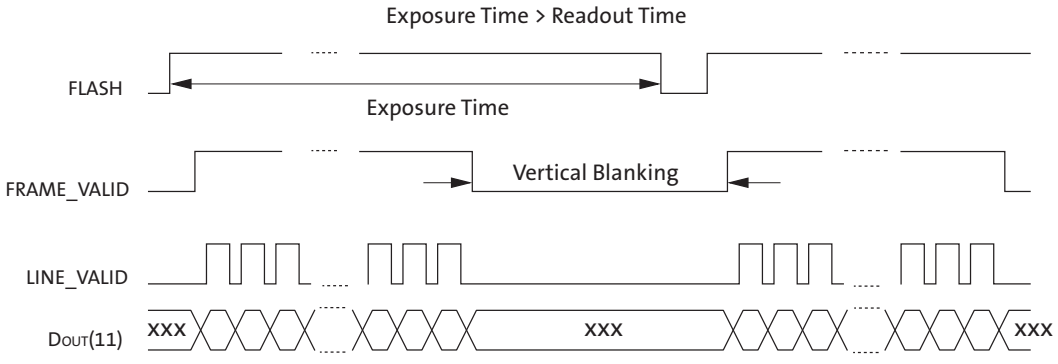


Figure 13: Master Mode Synchronization Waveform #2



Trigger Mode

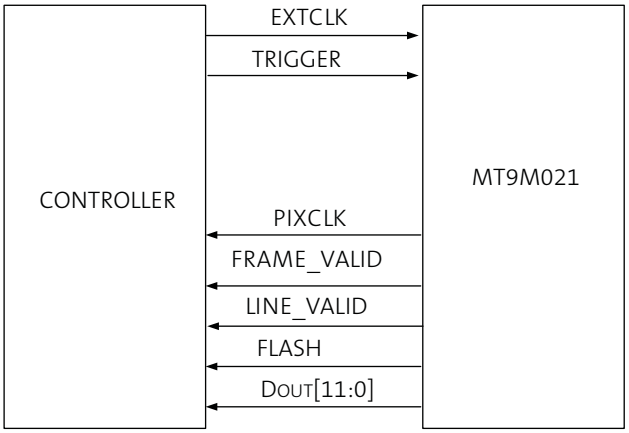
The Aptina™ MT9M021/MT9M031 CMOS image sensor is designed to be able to have the exposure start time synchronized to an external control source. This feature, called trigger mode, coupled with the global shutter mode of operation, is ideal for supporting the demands of machine vision systems, security, and interior and exterior automotive environments. When compared to the master mode of operation, this mode offers a simpler interface.

Operation Details

Many imaging applications commonly require the image sensor to capture an image only after a triggering action has taken place. This triggering action can be the passing of an object on a conveyor belt, the flash of a strobe light, or the press of a button.

The MT9M021/MT9M031 offers the ability to synchronize the start of the image sensor's exposure with this triggering action. This synchronization is controlled on the image sensor through the use of the TRIGGER input signal. Additionally, the image sensor offers the flexibility to program the exposure time remotely. This Developer Guide only addresses the single image sensor (non-stereoscopic) snapshot mode of operation.

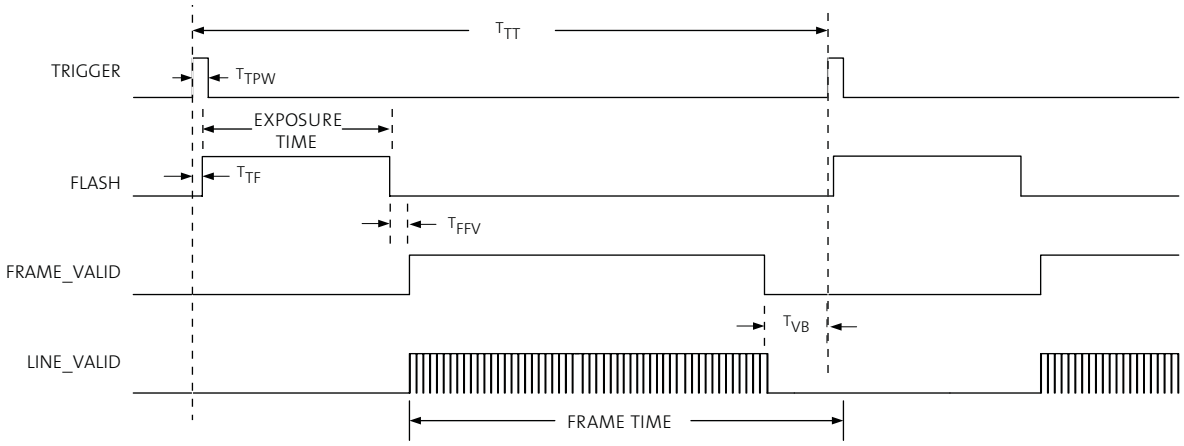
Figure 14: Block Diagram



Trigger Mode Overview

When the image sensor is set to trigger mode (R0x301A, bit 2 = 0), the beginning and duration of the exposure time are controlled. The global shutter feature of the image sensor allows all pixels to be exposed in parallel—all pixels start exposing (integrating charge) simultaneously and stop exposing simultaneously. When exposure stops, the per-pixel integrated charges are digitized and read out of the chip. A new exposure begins only after the readout of all pixels is complete. If the TRIGGER input is left in the asserted state, the sensor will automatically initiate a new frame acquisition sequence upon completion of the current frame.

Figure 15: Pulsed Trigger Mode



Notes:
1. Not drawn to scale.
2. Frame readout shortened for clarity.
3. Progressive scan readout mode shown.

Figure 16: Automatic Trigger Mode

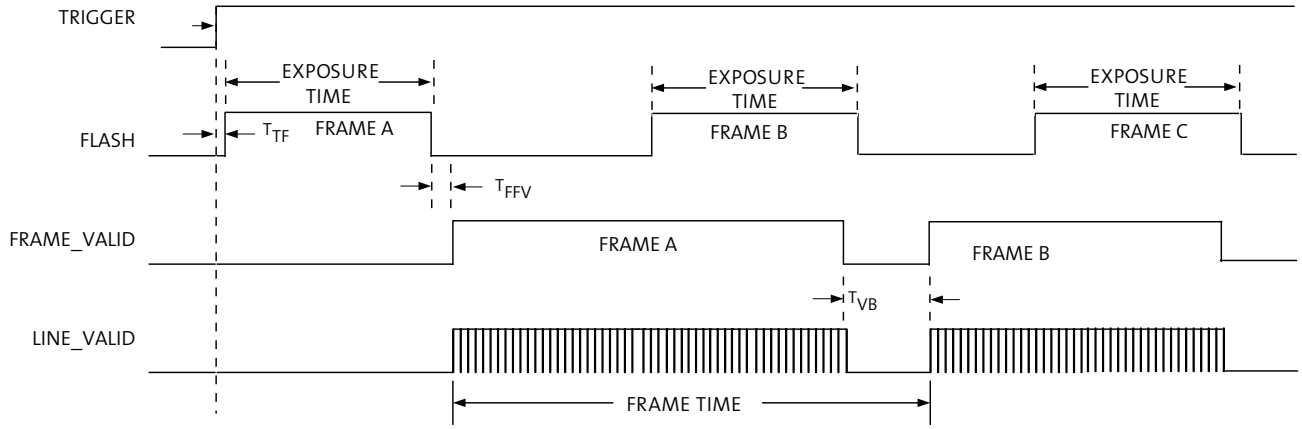


Table 5: Exposure Timing

Symbol	Description	Value
T_{TT}	TRIGGER signal period	T_{TF} + EXPOSURE TIME + T_{FFV} + FRAME TIME
T_{TPW}	TRIGGER signal pulse width	10 row times (MIN)
T_{TF}	TRIGGER to FLASH	8.21 row-times
T_{FFV}	FLASH to FRAME_VALID	18.21 row times
T_{VB}	Vertical blanking time	R0x300A - (R0X3006 - R0x3002 + 1)) * (Active Data Time + Horizontal Blanking Time) (MIN 25 Lines)

Notes:
1. EXTCLK-cycle unit is defined as the reciprocal of the EXTCLK input frequency.
2. See "Row-Time Definition" on page 21 for the row-time unit definition.
3. To change exposure time, change the coarse integration time registers R0x3012 for Context A or R0x3016 for Context B.
4. To change frame rate, change the T_{TT} value.

Register Settings

The TRIGGER mode of operation requires that register R0x301A (Reset Register) bit 2 be set to "0". Setting this register to "1" will switch the sensor back to the master mode of operation. The general purpose I/O (GPI) pins must also be enabled as shown in Table 6.

Table 6: Snapshot Mode Register Settings

Register	Register Name	Bit	Bit Name	Bit Description	Value in Dec (Hex)
R0x301A	Reset Register	2	Stream	0 = Trigger mode 1 = master mode	0
R0x301A	Reset Register	8	GPI_EN	0 = GPI input buffers disabled 1 =GPI input buffers enabled	1
R0x301A	Reset Register	11	forced_pll_on	0 = PLL powered down in standby 1 =PLL always powered	1

Start of Exposure

The start of exposure is controlled by the TRIGGER input on the image sensor. Normally, TRIGGER is held in a LOW state. To start exposing, this signal is changed to a HIGH state. This HIGH state is then sampled on the rising edge of the master clock (EXTCLK) of the image sensor. TRIGGER must be held HIGH for greater than 10 row times.

Duration of Exposure

The duration of the exposure is set by the value stored in R0x3012, which represents an equivalent number of row-times (see "Row-Time Definition" on page 21) to the actual exposure time. If the exposure time is to be set to approximately 2.22ms and default settings are being used (where one row-time equals 22.22µs), a value of "100" is entered in R0x3012 (2.22ms / 22.22µs = 100). In this mode, only whole number row-time increments are allowed—no fractional time increments can be achieved. It may be possible to adjust the number of horizontal active or blanking pixels to bring the desired exposure time to a whole number row-time increment.

The minimum exposure time supported by the MT9M021/MT9M031 image sensor in snapshot mode is 3 row-times.

The exposure time using the default power up settings of the sensor can be determined as follows:

$$exposure\_time = coarse\_integration\_time \times row\_time \quad \text{(EQ 3)}$$

$$exposure\_time = (179\ rows) \times (22.22\ \mu s) = 3.98ms \quad \text{(EQ 4)}$$

A restriction exists on coarse integration time when using trigger mode to ensure correct output of the FLASH signal. The following values for coarse integration time should be avoided:

$$coarse\_integration\_time = frame\_length\_lines - (active\_rows + col\_correction\_rows\ (8) + delta\_dark\_rows\ (6) + embedded\_stats\ (2) + 7) \quad (EQ\ 5)$$

As an example, if frame_length_lines is 990 and the number of active rows is 960 (964 minus 4 embedded stats and data rows), then an integration time of:

$$CIT = 990 - (960 + 8 + 6 + 2 + 7) \quad (EQ\ 6)$$

$$= 7 \quad (EQ\ 7)$$

should be avoided to ensure proper exposure in trigger mode. This restriction does not apply to master mode operation.

Row-Time Definition

One row-time is equal to the sum (R0x300C) of the number of active pixels (columns) and the number of horizontal blanking pixels divided by the pixel readout rate:

$$row\_time = \frac{active\_pixels + horizontal\_blank\_pixels}{PIXCLK\_frequency} \quad (EQ\ 8)$$

$$row\_time_{default\_settings} = \frac{line\_length\_pck(R0x300C)}{PIXCLK\_frequency} = \frac{1650}{74.25MHz} = 22.22\,\mu s \quad (EQ\ 9)$$

Note: Proper operation of the image sensor requires that the sum (R0x300C)of the active pixels and the horizontal blanking pixels must be a value greater than or equal to 1650.

Exposure and Data Synchronization Outputs

The MT9M021/MT9M031 image sensor offers an output synchronization signal (FLASH) that can be used to control the flash of a light source. The timing of this signal in snapshot mode is similar to the other exposure modes. The signal is normally held in a LOW state. FLASH changes to a HIGH state when the image sensor is exposing (integrating charge). FLASH returns to the normal LOW state once the exposure (set by register R0x3012 (Context A), or register R0x3016 (Context B)) has timed out.

To indicate a valid frame of video data is being output from the image sensor, FRAME_VALID switches to a HIGH state. The change of state occurs slightly over 18 row-times after the exposure time ends. FRAME_VALID returns to a LOW state after the active rows have been read out. The number of active rows plus vertical blanking is stored in the FRAME_LENGTH_LINES register (R0x300A) (default value is 990).

During the valid video frame state, LINE_VALID switches to a HIGH state to indicate a valid row of video data is being presented. LINE_VALID returns to a LOW state after a set number of PIXCLK cycles. This set number of master clock cycles equals the number of pixels stored in the line_length_pck register (R0x300C). The default value is 1650 pixel clock cycles.

TRIGGER Input Restrictions

In snapshot mode, the MT9M021/MT9M031 operates as a sequential readout image sensor, i.e. the image sensor first exposes all the pixels in a frame and then, after completing the exposure, reads them out. A new exposure can start can only after this readout has been completed. A second snapshot exists if the trigger pin is held high constantly. The first frame after trigger is asserted is a sequential frame. A second exposure occurs during the first frame readout time and then that exposure is readout on the next frame. All subsequent frames after the second frame readout will have the exposure N+1 and readout N occurring within the same frame.

The minimum time between two successive TRIGGER input pulses (shown as T_{T2T} in Figure 15 on page 19) is calculated from the exposure time and the frame time. The exposure time is described in "Exposure and Data Synchronization Outputs" on page 21. The frame time may be calculated from two variables: the row-time, and the number of rows-per-frame. The number of rows-per-frame is equal to the sum (R0x300A) of the number of active rows and the number of vertical blanking rows:

$$rows\_per\_frame = active\ rows + vertical\ blanking\ rows \quad \text{(EQ 10)}$$

$$rows\_per\_frame_{default\_settings} = frame\_length\_lines\ (R0x300A) = 990\ rows \quad \text{(EQ 11)}$$

The frame time is equal to the product of the number of rows-per-frame and the row-time.

$$frame\_time = rows\_per\_frame\ \ x\ (row\_time) \quad \text{(EQ 12)}$$

$$frame\_time_{default\_settings} = (990\ rows)x\left(\frac{22.22\mu s}{row}\right) = 21.99ms \quad \text{(EQ 13)}$$

The minimum time between two successive TRIGGER pulses equals the sum of the frame time, the exposure time, T_{TF} and T_{FFV}:

$$T_{TT} = (frame\_time) + (exposure\_time) + (T_{TF} + T_{FFV}) \quad \text{(EQ 14)}$$

Further, the maximum allowable frame rate may be calculated from these same three variables. The maximum frame rate is the reciprocal of the sum of the frame time, the exposure time, trigger to flash, and flash to frame valid times:.

$$frame\_rate = \frac{1}{frame\_time + exposure\_time + T_{TF} + T_{FFV}} \quad \text{(EQ 15)}$$

The TRIGGER pulse period should correspond to the desired frame rate. For individual (asynchronous) trigger pulses, the TRIGGER signal should be asserted no sooner than FRAME_VALID is deasserted, and the minimum of 30 rows of vertical blanking has elapsed.

Example Frame Rate Calculations

Two examples follow on performing frame rate calculations for the MT9M021/ MT9M031 image sensors, shown in Table 7 below and Table 8 on page 24.

Example 1 shows maximum allowable frame rate or case where trigger pin is only asserted once per frame:

Table 7: Example 1 (With Default Setting for Full Resolution)

Image Sensor Condition	Setting Description	Register [Bits] = Value
Exposure mode	Trigger	R0x301A[2] = 0 R0x301A[8] = 1 R0x301A[11] = 1
PIXCLK frequency	74.25 MHz	N/A
Window height	964[1]	R0x3006 = 963 R0x3002 = 4
Window width	1280	R0x3008 = 1281 R0x3004 = 2
line_length_pck	1650	R0x300C = 1650
frame_length_lines	990	R0x300A = 990
Integration time	179	R0x3012 = 179

Notes: 1. Window height is 964 rows with embedded stats and data enabled.

Step 1: Calculate the row-time.

$$row\_time = \frac{line\_length\_pck}{PIXCLK\_frequency} \quad \text{(EQ 16)}$$

$$row\_time = \frac{1650}{74.25\ MHz} = 22.22\,\mu s \quad \text{(EQ 17)}$$

Step 2: Calculate the rows-per-frame read out.

$$rows\_per\_frame = frame\_length\_lines \quad \text{(EQ 18)}$$

$$rows\_per\_frame = 990\ \ rows \quad \text{(EQ 19)}$$

Step 3: Calculate the frame time.

$$frame\_time = (rows\_per\_frame)x(row\_time) \quad \text{(EQ 20)}$$

$$frame\_time = (990\ rows)x\left(\frac{22.22\,\mu s}{row}\right) = 21.99ms \quad \text{(EQ 21)}$$

Step 4: Calculate the actual exposure time.

$$exposure\_time = (integration\_time) \times (row\_time)$$
$$exposure\_time = (179\ rows) \times 22.22\,\mu s = 3.97\,ms \quad \text{(EQ 22)}$$

Step 5: Calculate $T_{TF} + T_{FFV}$.

$$T_{TF} + T_{FFV} = (8.21\ rows + 18.21\ rows) \times row\_time$$
$$= 26.42\ rows \times 22.22\,\mu s \quad \text{(EQ 23)}$$
$$= 587\,\mu s \quad \text{(EQ 24)}$$

Step 6: Calculate the maximum allowable frame rate.

$$frame\_rate = \frac{1}{frame\_time + exposure\_time + T_{TF} + T_{FFV}}$$
$$frame\_rate = \frac{1}{21.99ms + 3.97ms + 587\,\mu s} \quad \text{(EQ 25)}$$
$$frame\_rate = = \frac{1}{26.55ms}$$
$$frame\_rate = = 37.67Hz$$

Table 8: Example 2 (With Default Settings for 720p)

Image Sensor Condition	Setting Description	Register[Bits] = Value
Exposure mode	Trigger	R0x301A[2] = 0 R0x301A[8] = 1 R0x301A[11] = 1
PIXCLK frequency	74.25 MHz	N/A
Window height	720	R0x3006 = 723 R0x3002 = 4
Window width	1280	R0x3008 = 1281 R0x3004 = 2
line_length_pck	1650	R0x300C = 1650
frame_length_lines	748	R0x300A = 748
Integration time	179	R0x3012 = 179

Step 1: Calculate the row-time.

$$row\_time = \frac{line\_length\_pck}{PIXCLK\_frequency} \quad \text{(EQ 26)}$$

$$row\_time = \frac{1650}{74.25\ MHz} = 22.22\,\mu s \quad \text{(EQ 27)}$$

Step 2: Calculate the rows-per-frame read out.

$$rows\_per\_frame = frame\_length\_lines \quad \text{(EQ 28)}$$

$$row\_per\_frame = 748\ rows \quad \text{(EQ 29)}$$

Step 3: Calculate the frame time.

$$frame\_time = (rows\_per\_frame)x(row\_time) \quad \text{(EQ 30)}$$

$$frame\_time = (748\ rows)x\left(\frac{22.22\mu s}{row}\right) = 16.62ms \quad \text{(EQ 31)}$$

Step 4: Calculate the actual exposure time.

$$exposure\_time = (integration\_time)x(row\_time) \quad \text{(EQ 32)}$$

$$exposure\_time = (179\ rows \times 22.22\mu s) = 2.97ms \quad \text{(EQ 33)}$$

Step 5: Calculate $T_{TF} + T_{FFV}$.

$$\begin{aligned} T_{TF} + T_{FFV} &= (8.21\ rows + 18.21\ rows) \times row\_time \\ &= 26.42\ rows \times 22.22\mu s \\ &= 587\mu s \end{aligned} \quad \text{(EQ 34)}$$

(EQ 35)

Step 6: Calculate the maximum allowable frame rate.

$$\begin{aligned} frame\_rate &= \frac{1}{frame\_time + exposure\_time + T_{TF} + T_{FFV}} \\ frame\_rate &= \frac{1}{16.62ms + 2.97ms + 587\mu s} \\ frame\_rate &= = \frac{1}{20.18\ ms} \\ frame\_rate &= = 49.56Hz \end{aligned} \quad \text{(EQ 36)}$$

Reset

The MT9M021/MT9M031 may be reset by using RESET_BAR (active LOW) or the reset register.

Hard Reset of Logic

The RESET_BAR pin can be connected to an external RC circuit for simplicity. The recommended RC circuit uses a 10kΩ resistor and a 0.1µF capacitor. The rise time for the RC circuit is 1ms maximum. Registers written via the two-wire interface will not be preserved following a hard reset.

Soft Reset of Logic

Soft reset of logic is controlled by bit 0 of the R0x301A Reset register. This bit is a self-resetting bit and also returns to "0" during two-wire serial interface reads. Registers written via the two wire interface will not be preserved following a soft reset.

Output Enable

The MT9M021/MT9M031's outputs can be tri-stated with the OE_BAR pin. Before the external pin can be used to control output enable, set register R0x301A[6] = 0 to disable the output drivers. Then set R0x301A[8] = 1 to enable the input pins (OE_BAR, TRIGGER, and STANDBY). Driving OE_BAR low will enable the output drivers, while driving it high will tri-state the parallel output pins. The parallel outputs can also be tri-stated by setting R0x301A[7] = 0.

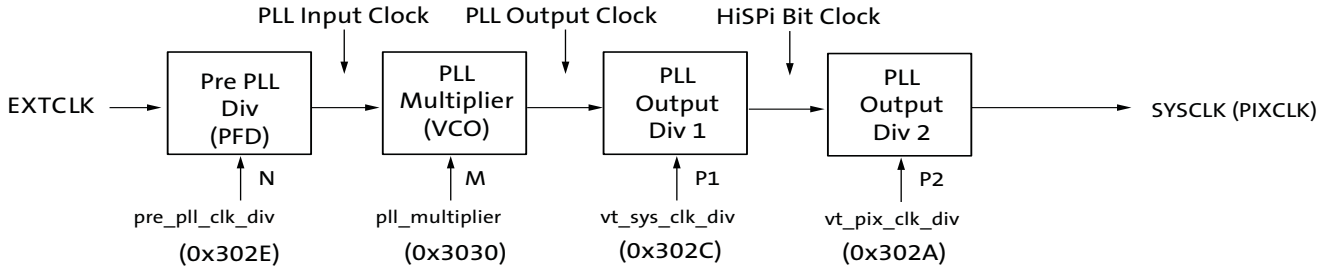
Clocks

PLL-Generated Master Clock

The PLL contains a prescaler to divide the input clock applied on EXTCLK, a VCO to multiply the prescaler output, and two divider stages to generate the output clock. The clocking structure is shown in Figure 17. The MT9M021/MT9M031 default power up state for the PLL dividers is for a pixel clock of 74.25 MHz that is generated from a 20.25 MHz EXTCLK. PLL control registers can be programmed to generate frequencies other than the default power up state. Refer to Table 9 on page 27 for PLL parameters for the parallel interface, and Table 11 on page 27 for the HISPi interface. Example PLL configurations are also provided using an EXTCLK of 27MHz. This is different from the default power up state.

Note: The PLL control registers must be programmed while the sensor is in the software Standby state. The effect of programming the PLL divisors while the sensor is in the streaming state is undefined.

Figure 17: PLL Block Diagram



The PLL is enabled by default on the MT9M021/MT9M031. To configure and use the PLL:

1. Bring the MT9M021/MT9M031 up as normal; make sure that ^{f}EXTCLK is between 6 and 50 MHz and ensure the sensor is in software standby (R0x301A[2]= 0). PLL control registers must be set in software standby.
2. Set pll_multiplier, pre_pll_clk_div, vt_sys_clk_div, and vt_pix_clk_div based on the desired input (f_{EXTCLK}) and output (f_{PIXCLK}) frequencies. Determine the M, N, P1, and P2 values to achieve the desired f_{PIXCLK} using the formula:

$$f_{PIXCLK} = (f_{EXTCLK} \times M) / (N \times P1 \times P2) \quad \textit{(EQ 37)}$$

where

M = pll_multiplier
N = pre_pll_clk_div
P1 = vt_sys_clk_div
P2 = vt_pix_clk_div

3. Wait 1ms to ensure that the VCO has locked.
4. Set R0x301A[2]=1 to enable streaming and to switch from EXTCLK to the PLL-generated clock.

Table 9: PLL Parameters for the Parallel Interface

Parameter	Symbol	Min	Max	Unit
External Clock	EXTCLK	6	50	MHz
VCO Clock	FVCO	384	768	MHz
Output Clock	PIXCLK		74.25	Mpixel/s

Table 10: Example PLL Configuration for the Parallel Interface

Parameter	Register Value	Output
FVCO		594 MHz
pre_pll_clk_div (N)	2	
pll_multiplier (M)	44	
vt_sys_clk_div	1	
vt_pix_clk_div	8	
PIXCLK		74.25 MPixel/s
Output pixel rate		74.25 MPixel/s

Table 11: PLL Parameters for the Serial Interface

Parameter	Symbol	Min	Max	Unit
External Clock	EXTCLK	6	50	MHz
VCO Clock	FVCO	384	768	MHz
Readout Clock	PIXCLK	74.25		Mpixel/s
Output Serial Data Rate Per Lane	FSERIAL	280 (HiSPi)	700 (HiSPi)	Mbps
Output Serial Clock Speed Per Lane	FSERIAL_CLK	140 (HiSPi)	350(HiSPi)	MHz

Table 12: Example PLL Configurations for the Serial Interface

Parameter	Register Value		Units
	3-lane	2-lane	
FVCO	742.5	445.5	MHz
pre_pll_clk_div	2	2	
pll_multiplier	55	33	
vt_sys_clk_div	2	1	
vt_pix_clk_div	5	6	
HiSPi Bit Clock	371.25	445.5	MHz
HiSPi Differential Clock	185.63	222.75	MHz
PIXCLK	74.25	74.25	Mpixel/s

Notes:
1. The PLL can be bypassed at any time (sensor will run directly off EXTCLK) by setting R0x30B0[14]=1. However, only the parallel data interface is supported with the PLL bypassed. The PLL is always bypassed in software standby mode.
2. The following restrictions apply to the PLL tuning parameters:
   $32 \le M \le 255$
   $1 \le N \le 63$
   $1 \le P1 \le 16$
   $4 \le P2 \le 16$
3. The VCO frequency, defined as $f_{VCO} = f_{EXTCLK} \times M/N$ must be within 384–768 MHz.
4. If using HiSPi output mode, use the following settings for P2 (vt_pix_clk_div).
   4a. If 12-bit mode (3 lanes): set P2 (R0x302A) = 5
   4b. If 12-bit mode (2 lanes): set P2 (R0x302A) = 6

The user can utilize the Register Wizard tool accompanying DevWare to generate PLL settings given a supplied input clock and desired output frequency.

Spread-Spectrum Clocking

To facilitate improved EMI performance, the external clock input allows for spread spectrum sources, with no impact on image quality. Limits of the spread spectrum input clock are:

- 5% maximum clock modulation
- 35 kHz maximum modulation frequency
- Accepts triangle wave modulation, as well as sine or modified triangle modulations.

Stream/Standby Control

The sensor supports two standby modes: Hard Standby and Soft Standby. In both modes, external clock can be optionally disabled to further minimize power consumption. If this is done, then the power-up sequence described in the MT9M021/ MT9M031 data sheet must be followed.

Soft Standby

Soft Standby is a low power state that is controlled through register R0x301A[2]. Depending on the value of R0x301A[4], the sensor will go to standby after completion of the current frame readout (default behavior) or after the completion of the current row readout. When the sensor comes back from Soft Standby, previously written register settings are still maintained. Soft standby will not occur if the TRIGGER pin is held high. A specific sequence needs to be followed to enter and exit from Soft Standby.

Entering Soft Standby:
1. Set R0x301A[12] = 1 if serial mode was used
2. Set R0x301A[2] = 0 and drive the TRIGGER pin LOW.
3. External clock can be turned off to further minimize power consumption (Optional)

Exiting Soft Standby:
1. Enable external clock if it was turned off
2. R0x301A[2] = 1 or drive the TRIGGER pin HIGH.
3. R0x301A[12] = 0 if serial mode is used

Hard Standby

Hard Standby puts the sensor in a lower power state.

A specific sequence needs to be followed to enter and exit from Hard Standby.

Entering Hard Standby:
1. R0x301A[8] = 1
2. R0x301A[12] = 1 if serial mode was used
3. Assert STANDBY pin
4. External clock can be turned off to further minimize power consumption (optional)

Exiting Hard Standby:
1. Enable external clock if it was turned off
2. De-assert STANDBY pin
3. Set R0x301A[8] = 0 (unless other inputs are used such as trigger, output_en, etc.)

Readout Modes

By default, the resolution of the output image is the full width and height of the FOV as defined above. The output resolution can be reduced by digital binning.

Digital Binning

All of the pixels in the FOV contribute to the output image in digital binning mode. This can result in a more pleasing output image with reduced artifacts. It also improves low-light performance. For RGB and monochrome mode, the digital binning factor is set by the register DIGITAL_BINNING (R0x3032). For Context A, use bits [1:0], for Context B, use bits [5:4]. Available settings are: 00 = No binning; 01 = Horizontal binning; 10 = Horizontal and vertical binning. For RGB mode, resampling must be enabled by setting bit 4 of register 0x306E. For monochrome operation, R0x30B0[7] must be set to 1. Enabling horizontal or vertical binning mode does not affect the sensor frame rate.

Skipping

Skipping reduces resolution by using only selected rows from the FOV in the output image. In skip mode, entire rows of pixels are not sampled, resulting in a lower resolution output image. A skip 2X mode skips one Bayer pair of pixels for every pair output. Skipping is set by R0x30A6 (context A) and R0x30A8 (context B). The maximum supported skip is 64 rows. Both Bayer and monochrome skip modes are supported. Refer to Table 13 on page 30 for supported skip factors.

Table 13: Skip Mode Settings

Skip Factor	R0x30A6 (R0x30A8)
No Skip	0x0001
2	0x0003
4	0x0007
8	0x000F
16	0x001F
32	0x003F
64	0x007F

Figure 18: Pixel Readout (no skipping)

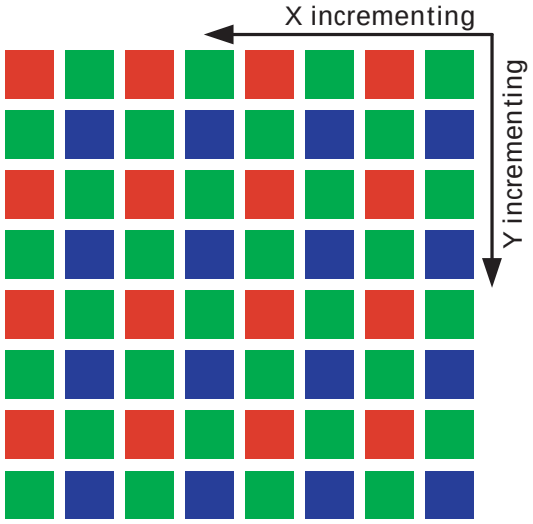


Figure 19: Pixel Readout (Row Skip 2X Bayer)

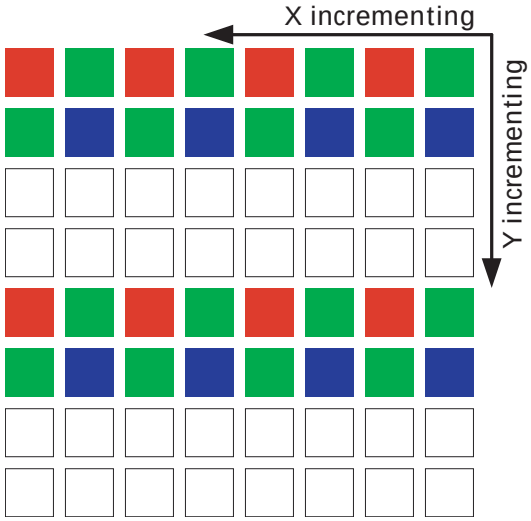
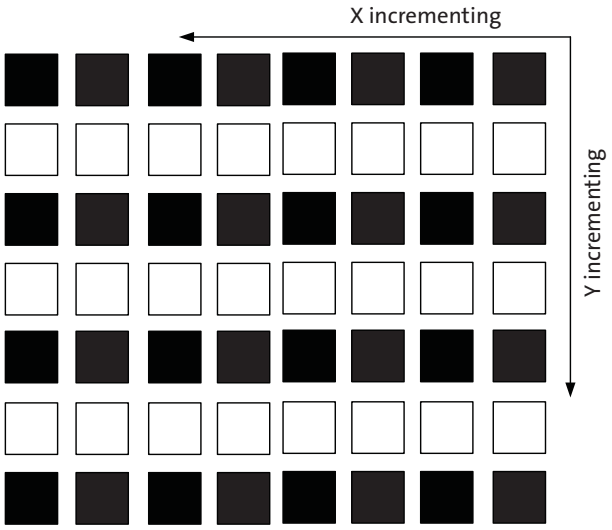


Figure 20: Pixel Readout (Row Skip 2X Monochrome)



Mirror

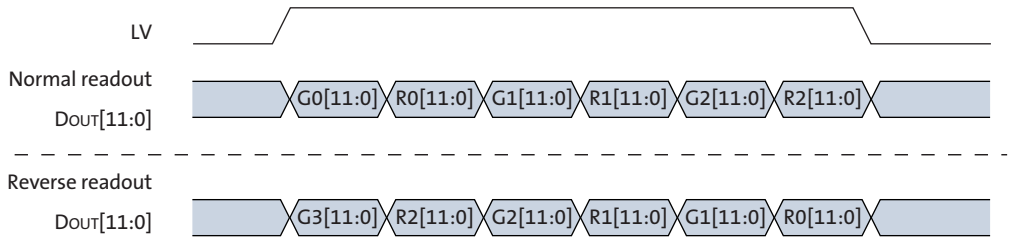
Column Mirror Image

By setting R0x3040[14] = 1, the readout order of the columns is reversed, as shown in Figure 21. The starting color, and therefore the Bayer pattern, is preserved when mirroring the columns.

When using horizontal mirror mode, the user must retrigger column correction. Refer to the column correction section to see the procedure for column correction retriggering.

Bayer resampling must be enabled, by setting bit 4 of register 0 x 306E[4] = 1.

Figure 21: Six Pixels in Normal and Column Mirror Readout Modes

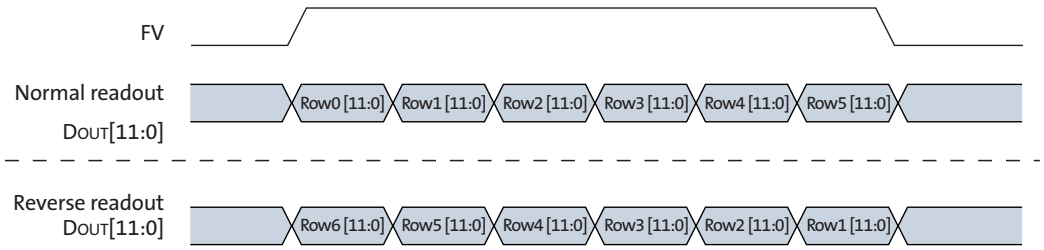


Row Mirror Image

By setting R0x3040[15] = 1, the readout order of the rows is reversed as shown in Figure 22. The starting Bayer color pixel is maintained in this mode by a 1-pixel shift in the imaging array. When using horizontal mirror mode, the user must retrigger column correction. Refer to the column correction section to see the procedure for column correction retriggering.

Bayer resampling must be enabled, by setting bit 4 of register 0 x 306E[4] = 1.

Figure 22: Six Rows in Normal and Row Mirror Readout Modes



Maintaining a Constant Frame Rate

Maintaining a constant frame rate while continuing to have the ability to adjust certain parameters is the desired scenario. This is not always possible, however, because register updates are synchronized to the read pointer, and the shutter pointer for a frame is usually active during the readout of the previous frame. Therefore, any register changes that could affect the row time or the set of rows sampled causes the shutter pointer to start over at the beginning of the next frame.

By default, the following register fields cause a "bubble" in the output rate (that is, the vertical blank increases for one frame) if they are written in video mode, even if the new value would not change the resulting frame rate. The following list shows only a few examples of such registers; a full listing can be seen in the MT9M021/MT9M031 Register Reference.

- X_Addr_Start
- X_Addr_End
- Y_Addr_Start
- Y_Addr_End
- Frame_Length_Lines
- Line_Length_Pclk
- Coarse_Integration_Time
- Fine_Integration_Time
- Read_Mode

The size of this bubble is (Integration_Time × ^{t}ROW), calculating the row time according to the new settings.

The Coarse_Integration_Time and Fine_Integration_Time fields may be written to without causing a bubble in the output rate under certain circumstances. Because the shutter sequence for the next frame often is active during the output of the current frame, this would not be possible without special provisions in the hardware. Writes to these registers take effect two frames after the frame they are written, which allows the integration time to increase without interrupting the output or producing a corrupt frame (as long as the change in integration time does not affect the frame time).

Synchronizing Register Writes to Frame Boundaries

Changes to most register fields that affect the size or brightness of an image take effect on the frame after the one during which they are written. These fields are noted as "synchronized to frame boundaries" in the MT9M021/MT9M031 Register Reference. To ensure that a register update takes effect on the next frame, the write operation must be completed after the leading edge of FV and before the trailing edge of FV.

As a special case, in trigger mode, register writes that occur after FV but before the next trigger will take effect immediately on the next frame, as if there had been a Restart. However, if the trigger for the next frame occurs during FV, register writes take effect as with video mode.

Fields not identified as being frame-synchronized are updated immediately after the register write is completed. The effect of these registers on the next frame can be difficult to predict if they affect the shutter pointer.

Restart

To restart the MT9M021/MT9M031 at any time during the operation of the sensor, write a "1" to the Restart register (R0x301A[1] = 1). This has two effects: first, the current frame is interrupted immediately. Second, any writes to frame-synchronized registers and the shutter width registers take effect immediately, and a new frame starts. The current row completes before the new frame is started, so the time between issuing the Restart and the beginning of the next frame is a maximum of ^{t}ROW.

Temperature Sensor

The MT9M021/MT9M031 sensor has a built-in PTAT-based (Proportional To Absolute Temperature) temperature sensor, accessible through registers, that is capable of measuring die junction temperature. The temperature sensor can be enabled by writing R0x30B4[0]=1 and R0x30B4[4]=1. After this, the temperature sensor output value can be read from R0x30B2[10:0].

The value read out from the temperature sensor register is an ADC output value that needs to be converted downstream to a final temperature value in degrees Celsius. Since the PTAT device characteristic response is quite linear in the temperature range of operation required, a simple linear function as in Equation 38 can be used to convert the ADC output value to the final temperature in degrees Celsius.

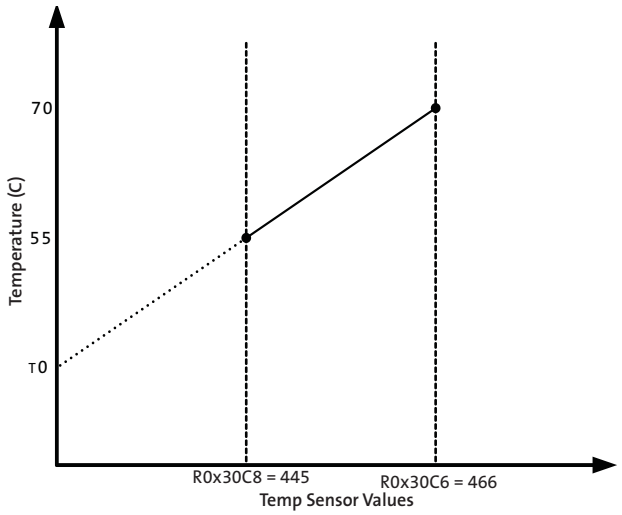
$$\text{Temperature} = \text{slope} \times \text{R0x30B2[10:0]} + \text{T0} \quad \text{(EQ 38)}$$

For this conversion, a minimum of 2 known points are needed to construct the line formula by identifying the slope and y-intercept "T0". These calibration values can be read from registers R0x30C6 and R0x30C8 which correspond to values read at 70°C and 55°C respectively. Once read, the slope and y-intercept values can be calculated and used in the above equation.

Example: What is the temperature in degrees Celsius when R0x30B2 = 0x1A2 (418)?

For this particular sensor, the 70°C calibration data reads R0x30C6 = 0x01D2 (466), and the 50°C calibration data reads R0x30C8 = 0x01BD (445). From these values, the correct temperature reading can be found as follows:

Figure 23: Calculating Temperature Sensor Value



slope = (70 - 55) / (466 - 445) = (15/21) = 0.714

From here, the intercept T0 can be found:

55 = (0.714) x (445) + T0
T0 = -262.73

Now, the temperature corresponding to a register reading of 0x1A2 can be determined:

Temperature = (0.714) x (418) - 262.73
Temperature = 35.7°C

For more information on the temperature sensor registers, refer to the MT9M021/MT9M031 Register Reference.

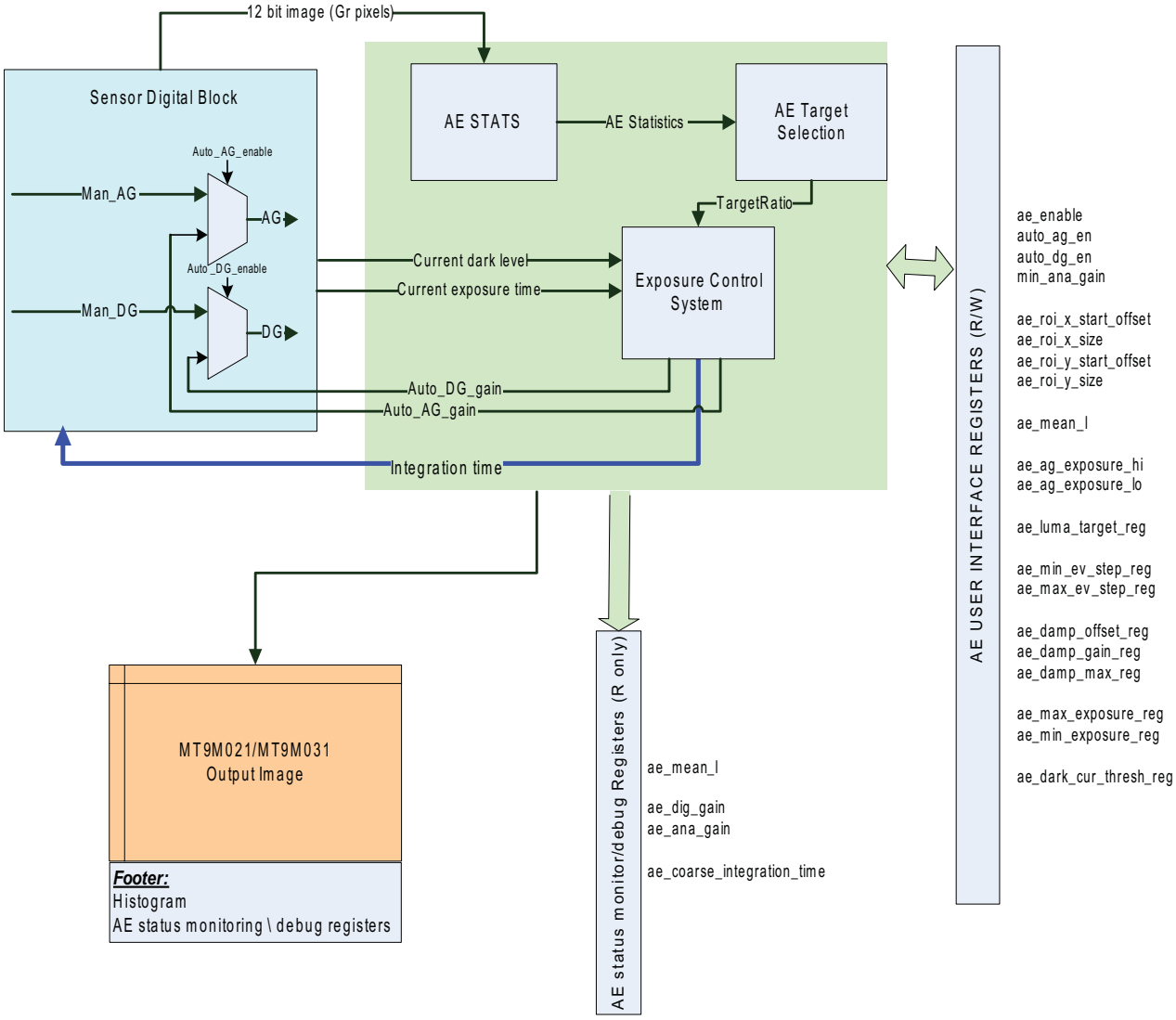
Auto Exposure

The integrated automatic exposure control (AEC) is responsible for ensuring that optimal settings of exposure and gain are computed and updated every other frame. AEC can be enabled or disabled by R0x3100[0]. When AEC is disabled (R0x3100[0] = 0), the sensor uses the manual exposure value in the coarse and fine integration time registers and the manual gain value in the gain registers. When AEC is enabled (R0x3100[0]=1), the target luma value is set by AE_LUMA_TARGET_REG (R0x3102). For MT9M021/MT9M031, this target luma has a default value of 0x0500 or about half scale. The luma target maximum auto exposure value is limited by R0x311C; the minimum auto exposure is limited by R0x311E. These values are in units of line-times. The minimum value for register 0x311E is 1 line. The exposure control measures current scene luminosity by accumulating a histogram of Gr pixel values while reading out a frame. It then compares the current luminosity to the desired output luminosity. Finally, the appropriate adjustments are made to the exposure time and gain.

Auto Exposure Implementation

The MT9M021/MT9M031 Auto Exposure control is implemented as three main blocks - AE Stats Calculation, AE Target Selection, and an Exposure Control System. See Figure 1 for details.

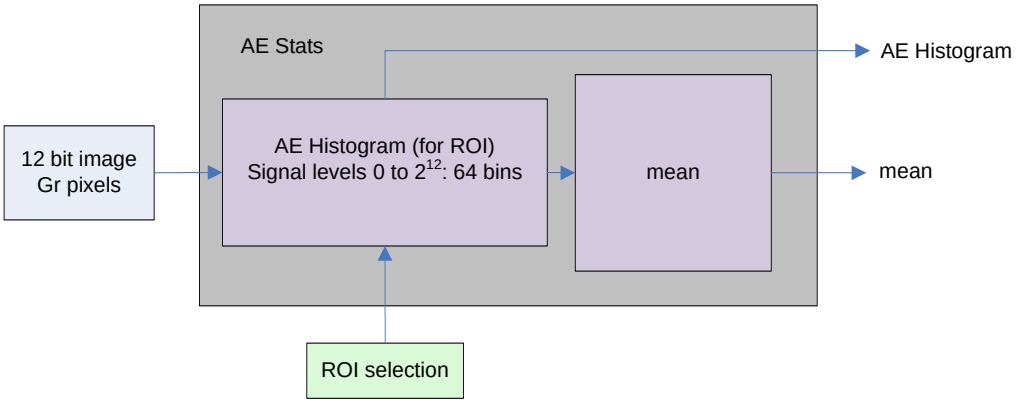
Figure 24: AE Block Diagram



AE Embedded Statistics and Data

The AE Stats Calculation block (Figure 2) takes the user specified Region of Interest (ROI) and creates a histogram of 64 evenly distributed bins from 0 to the maximum code value of 212 based on Gr pixels. If no ROI is specified, statistics are gathered from the full output frame. From this histogram, all relevant auto exposure statistics are generated:

Figure 25: AE Stats Calculation Block



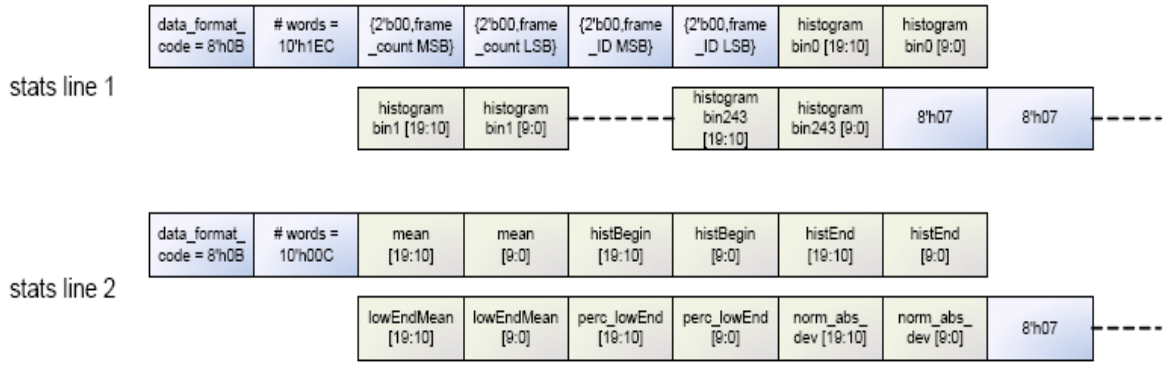
1.AE Histogram: 64 evenly spaced bins for digital code values 0 to 212, If a ROI is specified, the histogram is populated only with Gr pixels which lie in the ROI.

2.mean: Mean code value of AE Histogram

The generation of statistics for use by off-chip AE algorithms must be enabled by setting register R0x3064[7] = 1. Embedded statistics will not be output if this register is not set. Embedded data may also be enabled by setting register R0x3064[8] = 1, but is not necessary for statistics generation or auto exposure control.

All the statistics data (including histogram data) is embedded in the two rows immediately following the image. The embedded statistics are output as shown in Figure 26. The first line contains histogram data. Only histogram data for bins 0 to 63 are relevant - higher bins will output all zeros. The second line contains statistics based on the histogram for the current frame. The only relevant statistic for MT9M021/MT9M031 auto exposure is the mean. If the on-chip auto exposure is not used, it is recommended that auto exposure algorithms be developed based on the histogram data found in line 1.

Figure 26: Embedded Statistics Format



The embedded data contains the configuration of the image being displayed, and is found in the first two rows of the image. This includes all register settings used to capture the current frame. The registers embedded in these rows are as follows:

Line 1: Registers R0x3000 to R0x312F

Line 2: Registers R0x3136 to R0x31BF, R0x31D0 to R0x31FF

Only values for registers found in the Register Reference document are relevant. The format of the embedded register data transmission is as follows. In parallel mode, since the pixel word depth is 12 bits/pixel, the sensor's 16-bit register data will be transferred over 2 pixels where the register data will be broken up into 8msb and 8 LSB. The alignment of the 8-bit data will be on the 8 MSB bits of the 12-bit pixel word. For example, if a register value of 0x1234 is to be transmitted, it will be transmitted over two 12-bit pixels as follows: 0x120, 0x340. The 10-bit histogram data is not broken up and is output msb aligned over the 12 bit parallel interface and padded with zeros.

AE Target Selection

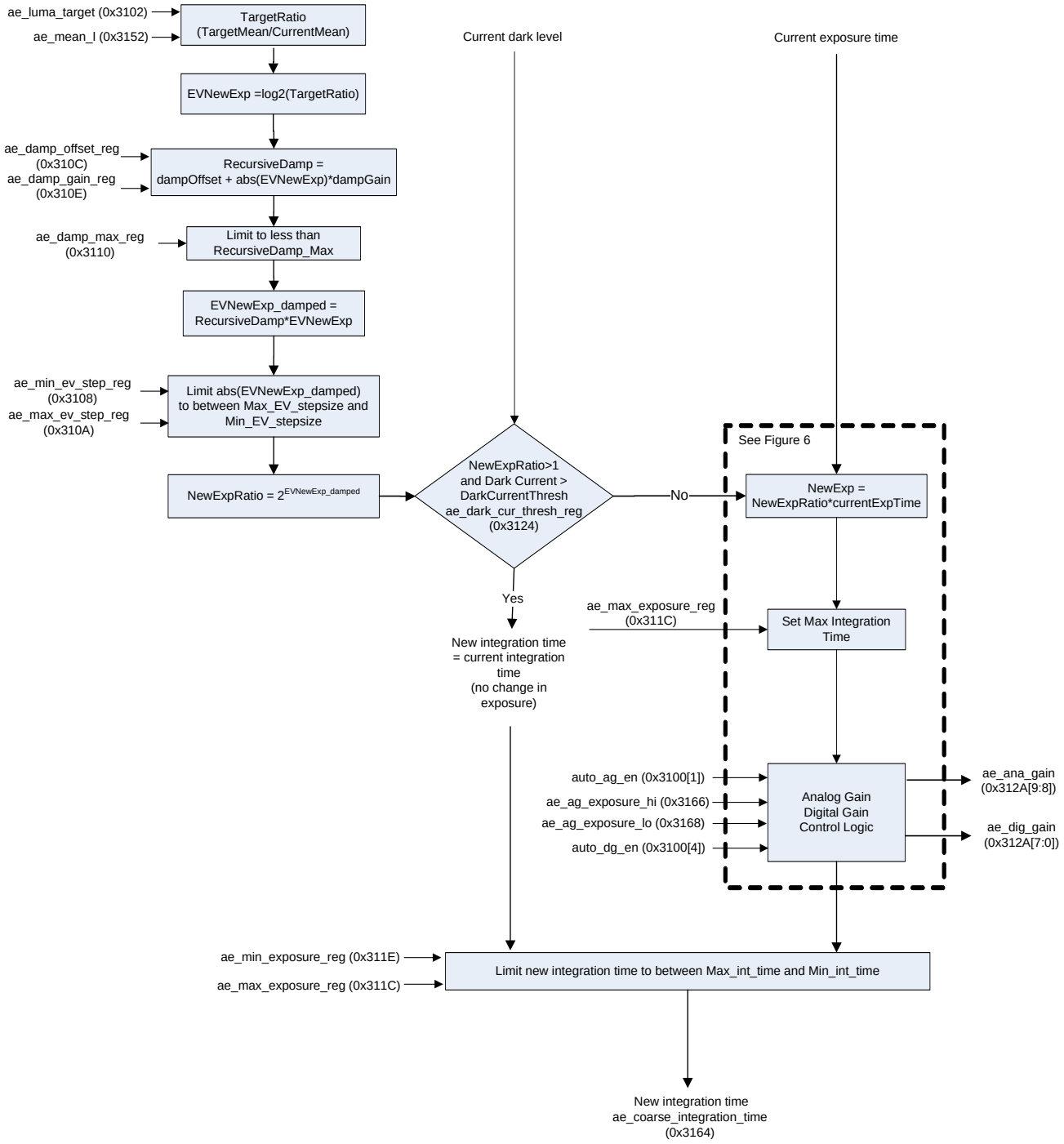
The Exposure Target Selection block determines a ratio based on the mean value of the generated histogram of the current frame, and the target mean value as specified by the user (R0x3102). This ratio allows the Control System to determine how much and in what direction to adjust the exposure relative to the current exposure value.

The mean target ratio (TargetRatio) is the exposure change, expressed as a ratio, to move the current image mean (CurrentMean) to a user-specified mean target (TargetMean). See the “Exposure Control System” section and Figure 27 for more information.

Exposure Control System

The Exposure Control System outputs the new integration time along with a damping factor to prevent too rapid of a response. If enabled, analog and digital gains will be selected as well. The Control System will also monitor the dark current. If the Exposure Target Selection block indicates that the exposure should be increased, but the dark current exceeds a user specified threshold, the Control System will maintain the current integration time. The automatic digital and analog gains and exposure limits enclosed by the dashed line in Figure 27 is illustrated in more detail in Figure 6 and described in “Controlling Auto Exposure” on page 40.

Figure 27: Exposure Control System



Variables found in Figure 4 are described in Table 14 on page 40.

Table 14: Exposure Control Variables

Internal Value	Description
EVNewExp	Target ratio translated into EV units (stops). Can be positive or negative. In EV units, >0 means exposure is increasing, <0 means exposure is decreasing.
RecursiveDamp	Damping factor. Should be >0 and <1 for desirable AE operation. If less than 0, AE will step further from the target; if greater than 1, AE will overstep the target.
EVNewExp_damped	New exposure step in EV units. Can be positive or negative.
NewExpRatio	New exposure step as ratio. Should be positive. As ratio, >1 means exposure is increasing, <1 means exposure is decreasing.
NewExp	New exposure expressed as rows of integration or possibly msec (depends on rest of system).

Auto Exposure Control

Enabling Auto Exposure

Several registers are used to enable various features of the automatic exposure control. The auto exposure block is enabled or disabled by register R0x3100[0]. By default, the AEC will only modify the coarse integration time to reach the target exposure. If enabled, analog and digital gains may be adjusted as well. Analog gain adjustment is enabled by setting auto_ag_en (R0x3100[1] = 1), and digital gain adjustment is enabled by setting auto_dg_en (R0x3100[4] = 1). A minimum column gain (1x, 2x, 4x, 8x), min_ana_gain, may be defined in register R0x3100[6:5]. Digital gain may be adjusted from 1x to 7.97x. A summary of AEC enable registers is listed in Table 15.

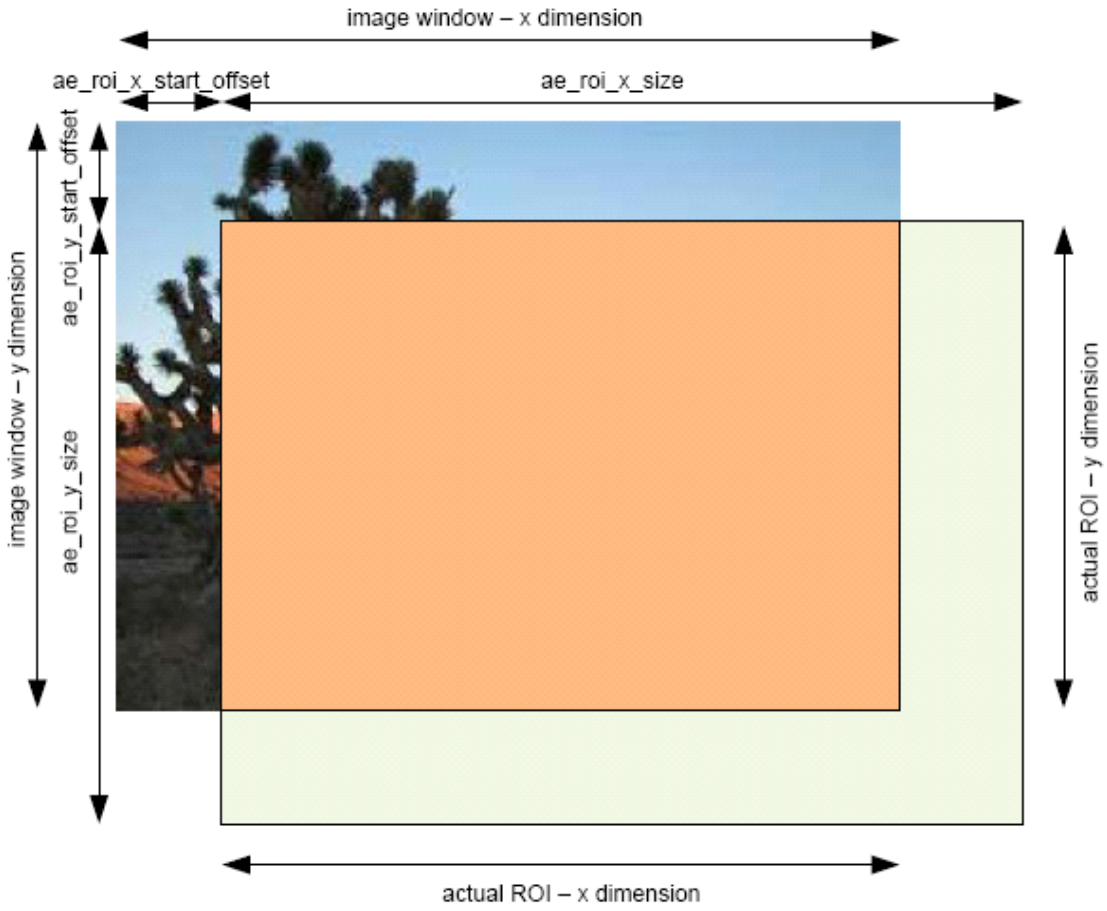
Table 15: AE Enable Registers

Register	Name	Function
0x3100[0]	ae_enable	0: On-chip AE disabled 1: On-chip AE enabled
0x3100[1]	auto_ag_en	0: AE will not control analog gain 1: AE will control analog gain
0x3100[4]	auto_dg_en	0: AE will not control digital gain 1: AE will control digitalgain
0x3100[6:5]	min_ana_gain	Minimum analog gain to ge used by AE 00: 1x (default) 01: 2x 10: 4x 11: 8x

Controlling Auto Exposure

The histogram is generated and statistics calculated based on the Gr pixels within a user specified region of interest. The ROI is specified by four programmable register values - ae_roi_x_start_offset, ae_roi_y_start_offset, ae_roi_x_size and ae_roi_y_size. The ae_roi_x_start_offset and ae_roi_y_start_offset values define the starting coordinate of the ROI with respect to the image window that is output and the ae_roi_x_size and ae_roi_y_size values define the dimensions of the ROI. Each value must be an even number. If the requested ROI extends 'beyond' the image window then it will be restricted in size such that the final pixel of the ROI will be the final pixel of the image window, as illustrated in Figure 28

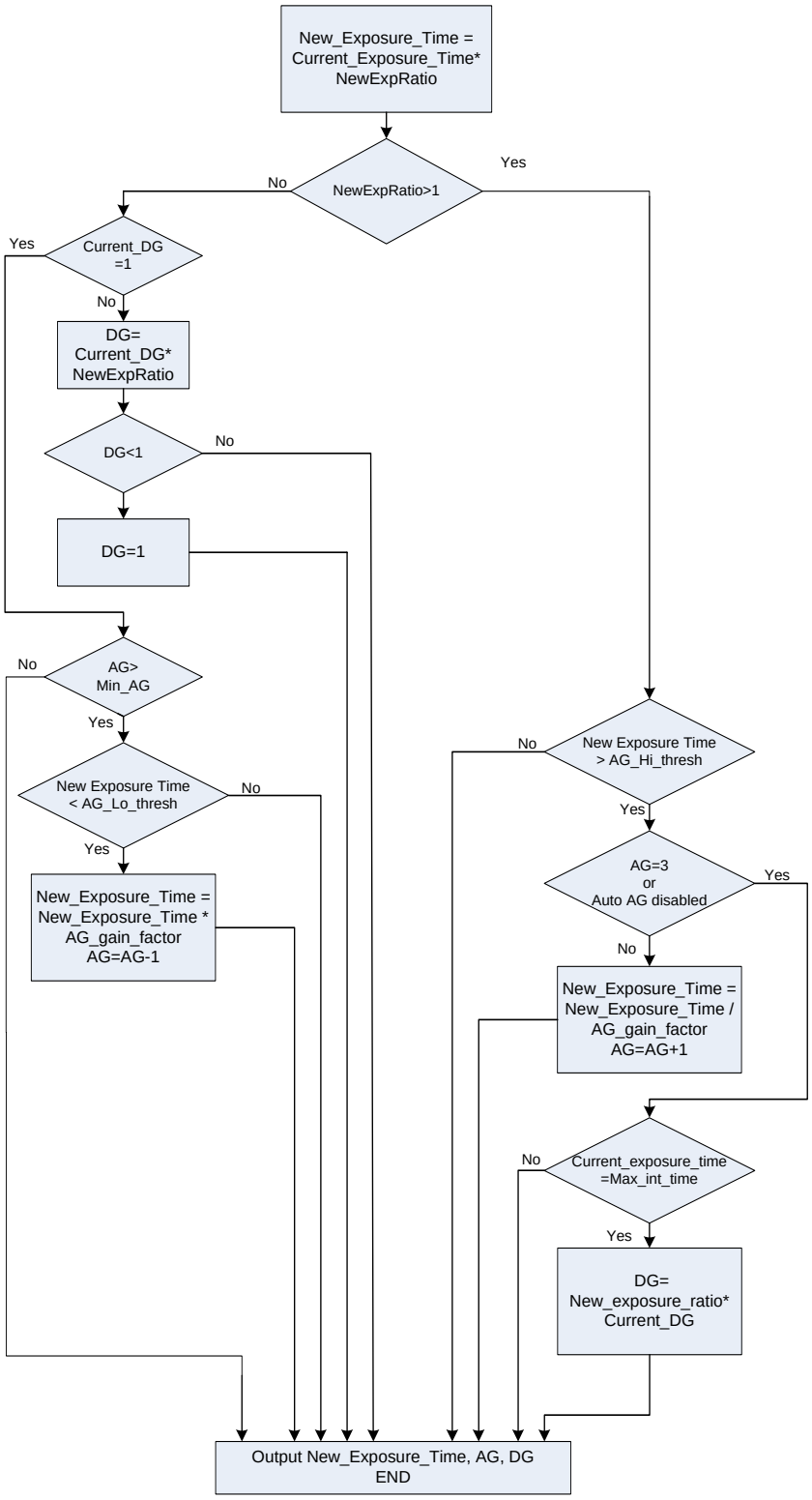
Figure 28: Selecting the ROI



The target luma value may be set in the ae_luma_target_reg register. The AE Target Selection block will use this value to determine the target ratio provided to the Exposure Control System as illustrated in Figure 4. The exposure range can be limited by setting values for ae_max_exposure_reg and ae_min_exposure_reg. The integration time fed back to the Sensor Digital Block (see Figure 1) will not fall outside of this specified range.

To extend the exposure range, the AE logic can also automatically adjust analog gain and digital gain. The controls for enabling automatic analog and digital gain selection may be found in Table 15 on page 40. The control flow chart is shown in Figure 29 and is an expanded view of the portion of Figure 4 that is enclosed by the dashed line.

Figure 29: Automatic Gain Control



If auto_ag_en is set, the automatic adjustment of analog gains may be restricted based on integration time. By setting a value for ae_ag_exposure_hi, the analog gain will not be increased until the integration time set by this register is reached. Similarly, the analog gain will not be decreased unless the integration time is reduced below the value set in ae_ag_exposure_lo. To avoid oscillation, the ae_ag_exposure_lo setting should be lower than the ae_ag_exposure_hi setting.

The integration time and analog gain selected by the exposure control system may be found in the ae_coarse_integration_time (R0x3164) and ae_ana_gain(R0x312A[9:8]) registers, respectively. The minimum analog gain to be selected may be set in the min_ana_gain (R0x3100[6:5]) register, and can be 1x, 2x, 4x, or 8x. If auto_dg_en (R0x3100[4]) is set, the digital gain selected by the exposure control system can be read from register ae_dig_gain (R0x312A[7:0]). The digital gain can vary from 1 to 7.97. The minimum step is 1/32.

The step size of the AE control may be configured. Both a minimum and maximum step size may be set in units of EV (exposure value) steps in registers 0x3108 and 0x310A, respectively. The step size represents the minimum or maximum value that the AE Target Selection will use for the next exposure value. It does not represent the incremental change from frame to frame. The selected new exposure value will be clipped to the minimum EV step if it is less than the value specified in R0x3108. Because the minimum step size in EV units is typically a small number less than one, it should be scaled by 256 before setting the register value.

Changes in exposure are smoothed based on damping parameters. A maximum damping value may be specified in R0x3110. Additional damping controls include ae_damp_gain_reg and ae_damp_offset_reg. These can be thought of as a coarse and fine damping control, respectively.

At high temperature, the sensor may have high dark current which will increase with longer exposures. To avoid increasing the exposure when there is excessive dark current, AE has a dark current check. The sensor supplies the current dark current level to AE and if the dark current is greater than the user-specified (R0x3124) darkCurrentThresh, AE does not increase exposure.

```
If (NewExpRatio > 1) & (DarkCurrent > DarkCurrentThresh)
NewExpRatio = 1;  //Do not increase exposure
End
```

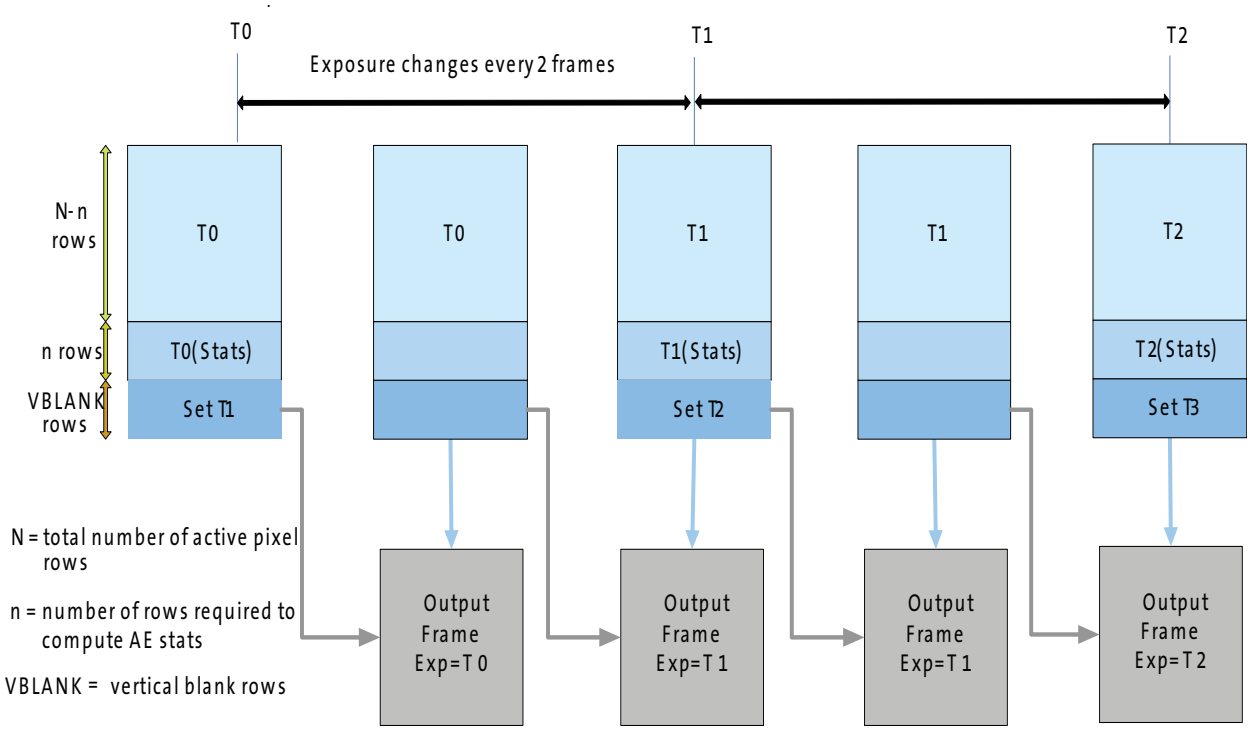
Table 16: Auto Exposure Control Registers

Register	Name	Function
0x3140	ae_roi_x_start_offset	Number of pixels into each row before the ROI starts
0x3142	ae_roi_y_start_offset	Number of rows into each frame before the ROI starts
0x3144	ae_roi_x_size	Number of columns in the ROI
0x3146	ae_roi_y_size	Number of rows in the ROI
0x3102	ae_luma_target_reg	Average luma target value to be reached by the auto exposure
0x3108	ae_min_ev_step_reg	Minimum exposure value step size. Since min_ev_step sizes are small (typically less than 1), they are multiplied by 256 and then the value is written to this register.
0x310A	ae_max_ev_step_reg	Maximum exposure value step size. Since this value is always greater than 1 there is no need to multiply by 256 as in the case of min_EV_stepsize.
0x310C	ae_damp_offset_reg	Adjusts step size and settling speed.
0x310E	ae_damp_gain_reg	Adjusts step size and settling speed.
0x3110	ae_damp_max_reg	Max value allowed for damping (multipled by 256 since internal value is typically <1). For most applications, the value of damping should be <1, otherwise AE will overshoot the target. For applications with fast settling required, it may be desirable to allow damping >1. Default value: 0.875 * 256 = 0x00E0
0x311C	ae_max_exposure_reg	Maximum integration (exposure) time in rows to be used by AE.
0x311E	ae_min_exposure_reg	Minimum integration (exposure) time in rows to be used by AE.
0x3166	ae_ag_exposure_hi	At this integration time, the analog gain is increased (when AE is enabled to control analog gain).
0x3168	ae_ag_exposure_lo	At this integration time, the analog gain is reduced (when AE is enabled to control analog gain).
0x3124	ae_dark_cur_thresh_reg	The dark current level that stops AE from increasing integration time. Note that increased integration time would increase dark current as well and signal level (SNR) would drop because photo diode well capacity is limited.

AE Frame Synchronization

A delay is incurred between the time when a frame with the newly updated AE value applied is seen by the AE module and when it reaches the sensor core logic (which sets the exposure times for the sensor). This delay is associated with the Delay Buffers and Sensor Data Path delays. The AE module will perform its calculations during the vertical blanking time and the new exposure value will be seen by the sensor core logic after the next frame has started. Therefore the result is that the third frame after the current frame will reflect the new exposure time. Figure 30 illustrates how the exposure changes every two frames.

Figure 30: AE Frame Synchronization



Gain

Digital Gain

Digital gain can be controlled globally by R0x305E (Context A) or R0x30C4 (Context B). There are also registers that allow individual control over each Bayer color channel:

GreenR	R0x3056
GreenB	R0x305C
Red	R0x305A
Blue	R0x3058

The format for digital gain setting is xxx.yyyyy where 0b00100000 represents a 1x gain setting and 0b00110000 represents a 1.5x gain setting. The step size for yyyyy is 0.03125 while the step size for xxx is 1. Therefore to set a gain of 2.09375 one would set digital gain to 01000011. The maximum digital gain is 7.97x.

Column Gain

The MT9M021/MT9M031 has a column parallel architecture and therefore has an analog gain stage per column. The column (analog) gain can be set to 1x, 2x, 4x or 8x. This can be set in R0x30B0[5:4](Context A) or R0x30B0[9:8] (Context B).

Black Level Correction

Black level correction is handled automatically by the image sensor. No adjustments are provided except to enable or disable this feature. Setting R0x30EA[15] disables the automatic black level correction. Default setting is for automatic black level calibration to be enabled.

The automatic black level correction measures the average value of pixels from a set of optically black lines in the image sensor. The pixels are averaged as if they were light-sensitive and passed through the appropriate gain. This line average is then digitally low-pass filtered over many frames to remove temporal noise and random instabilities associated with this measurement. The new filtered average is then compared to a minimum acceptable level, low threshold, and a maximum acceptable level, high threshold. If the average is lower than the minimum acceptable level, the offset correction value is increased by a predetermined amount. If it is above the maximum level, the offset correction value is decreased by a predetermined amount. The high and low thresholds have been calculated to avoid oscillation of the black level from below to above the targeted black level.

Row-wise Noise Correction

Row (Line)-wise Noise Correction is handled automatically by the image sensor. No adjustments are provided except to enable or disable this feature. Clearing R0x3044[10] disables the row noise correction. Default setting is for row noise correction to be enabled.

Row-wise noise correction is performed by calculating an average from a set of optically black pixels at the start of each line and then applying each average to all the active pixels of the line.

Column Correction

The MT9M021/MT9M031 uses a column parallel readout architecture to achieve fast frame rates. Without any corrections, the consequence of this architecture is that different column signal paths have slightly different offsets that might show up on the final image as structured fixed pattern noise.

The MT9M021/MT9M031 has column correction circuitry that measures this offset and removes it from the image before output. This is done by sampling dark rows containing tied pixels and measuring an offset coefficient per column to be corrected later in the signal path.

Column correction can be enabled/disabled via R0x30D4[15]. Additionally, the number of rows used for this offset coefficient measurement is set in R0x30D4[3:0]. By default this register is set to 0x7, which means that 8 rows are used. This is the recommended value. Other control features regarding column correction can be viewed in the MT9M021/MT9M031 Register reference. Any changes to column correction settings need to be done when the sensor streaming is disabled and the appropriate triggering sequence must be followed as described below.

Column Correction Triggering

Column correction requires a special procedure to trigger depending on which state the sensor is in.

Column Triggering on Startup

When streaming the sensor for the first time after powerup, a special sequence needs to be followed to make sure that the column correction coefficients are internally calculated properly.

1. Follow proper power up sequence for power supplies and clocks.
2. Apply sequencer settings.
3. Apply frame timing and PLL settings as required by application.
4. Set analog and digital gains to 1x.
5. Enable column correction and settings (R0x30D4 = 0xE007).
6. Enable streaming (R0x301A[2]=1)or drive the TRIGGER pin HIGH.
7. Wait 8 frames to settle.
8. Disable streaming (R0x301A[2]=0) or drive the TRIGGER pin LOW.

After this, the sensor has calculated the proper column correction coefficients and the sensor is ready for streaming. Any other settings (including gain, integration time, etc.) can be done afterwards without affecting column correction.

Column Correction Retriggering Due to Mode Change

Since column offsets are sensitive to changes in the analog signal path, such changes require column correction circuitry to be retriggered for the new path. Examples of such mode changes include: horizontal mirror, vertical flip, changes to column correction settings.

When such changes take place, the following sequence needs to take place:

1. Disable streaming (R0x301A[2]=0) and drive the TRIGGER pin LOW.
2. Disable column correction (0x30D4[15]= 0).
3. Enable streaming (R0x301A[2]=1) or drive the TRIGGER pin HIGH.
4. Wait one frame time, or more.

5. Disable streaming (R0x301A[2]= 0)and drive the TRIGGER pin LOW.
6. Enable column correction (0x30D4[15]= 0).
7. Enable Streaming (R0x301A[2]=1) or drive the TRIGGER pin HIGH.
8. Wait 8 frames to settle.

Note: The above steps are not needed if the sensor is being reset (soft or hard reset) upon the mode change.

Test Patterns

The MT9M021/MT9M031 has the capability of injecting a number of test patterns into the top of the datapath to debug the digital logic. With one of the test patterns activated, any of the datapath functions can be enabled to exercise it in a deterministic fashion. Test patterns are selected by test_pattern_mode register (R0x3070). Only one of the test patterns can be enabled at a given point in time by setting the test_pattern_mode register according to Table 17. When test patterns are enabled the active area will receive the value specified by the selected test pattern and the dark pixels will receive the value in test_pattern_green (R0x3074 and R0x3078) for green pixels, test_pattern_blue (R0x3076) for blue pixels, and test_pattern_red (R0x3072) for red pixels.

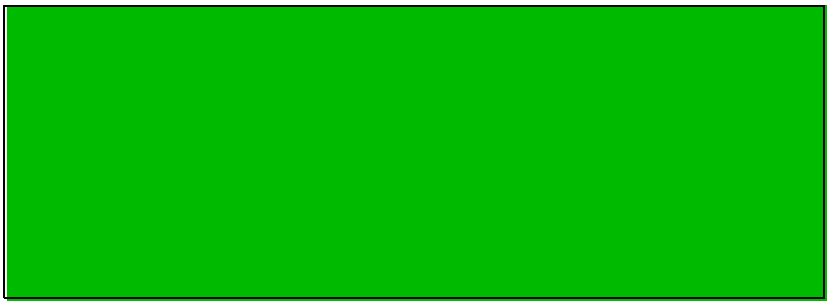
Table 17: Test Pattern Modes

Test_Pattern_Mode	Test Pattern Output
0	No test pattern (normal operation)
1	Solid color test pattern
2	100% color bar test pattern
3	Fade-to-grey color bar test pattern
256	Walking 1s test pattern (12-bit)

Color Field

When the color field mode is selected, the value for each pixel is determined by its color. Green pixels will receive the value in test_pattern_green, red pixels will receive the value in test_pattern_red, and blue pixels will receive the value in test_pattern_blue. See Figure 31 for a solid green pattern with Gr = Gb = 3072.

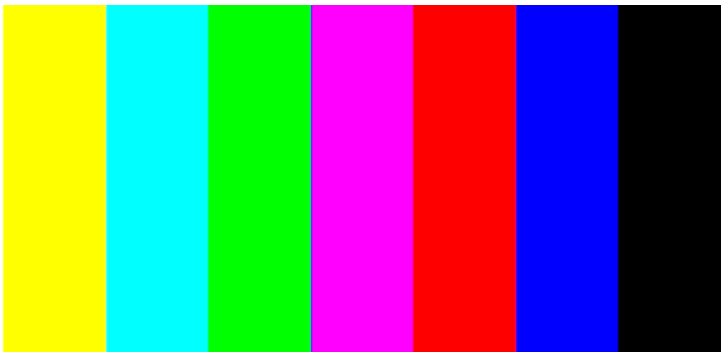
Figure 31: Solid Color



Vertical Color Bars

When the vertical color bars mode is selected, a typical color bar pattern will be sent through the digital pipeline. See Figure 32:

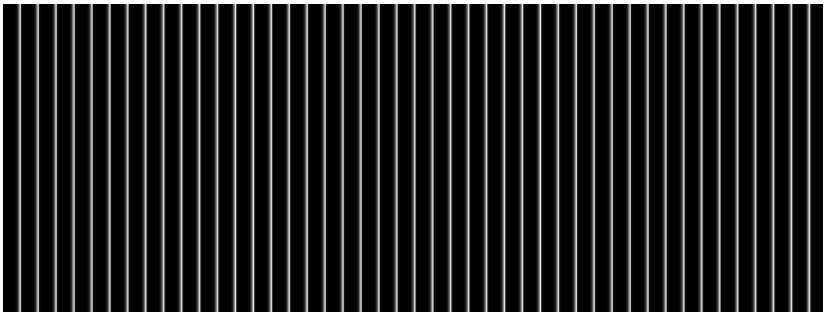
Figure 32: Vertical Corlor Bars



Walking 1s

When the walking 1s mode is selected, a walking 1s pattern will be sent through the digital pipeline. The first value in each row is 1. See Figure 33:

Figure 33: Walking 1s



Two-Wire Serial Register Interface

The two-wire serial interface bus enables read/write access to control and status registers within the MT9M021/MT9M031.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_IO 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 MT9M021/MT9M031 uses SCLK as an input only and therefore never drives it LOW. For detailed timing and electrical specifications for the HiSPi interface, please refer to the MT9M021/MT9M031 Datasheet.

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 MT9M021/MT9M031 are 0x20 (write address) and 0x21 (read address) in accordance with the specification. Alternate slave addresses of 0x30 (write address) and 0x31 (read address) can be selected by enabling and asserting the SADDR input.

An alternate slave address can also be programmed through R0x31FC.

Message Byte

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

Acknowledge Bit

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

No-Acknowledge Bit

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

Typical Sequence

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

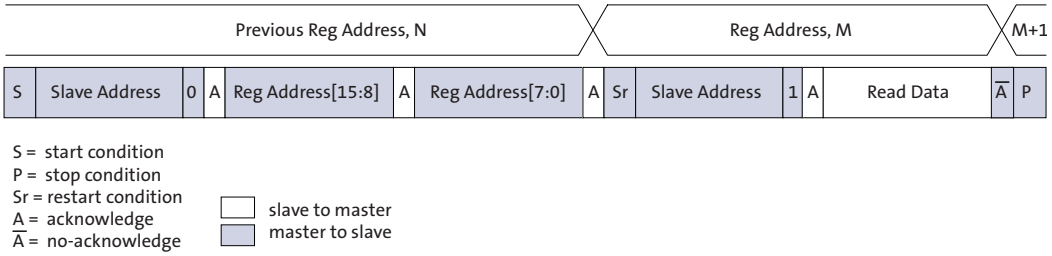
If the request was a WRITE, the master then transfers the 16-bit register address to which the WRITE should take place. This transfer takes place as two 8-bit sequences and the slave sends an acknowledge bit after each sequence to indicate that the byte has been received. The master then transfers the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

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

Single READ from Random Location

This sequence (Figure 34 on page 52) 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 34 shows how the internal register address maintained by the MT9M021/MT9M031 is loaded and incremented as the sequence proceeds.

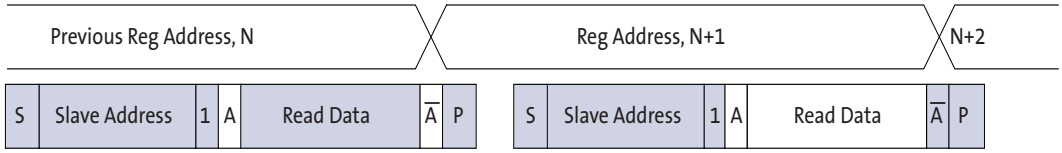
Figure 34: Single READ from Random Location



Single READ from Current Location

This sequence (Figure 35) performs a read using the current value of the MT9M021/MT9M031 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.

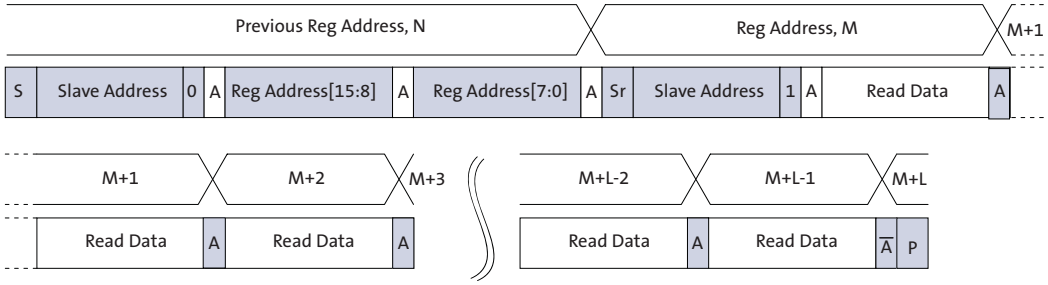
Figure 35: Single READ from Current Location



Sequential READ, Start from Random Location

This sequence (Figure 36) starts in the same way as the single READ from random location (Figure 34). 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.

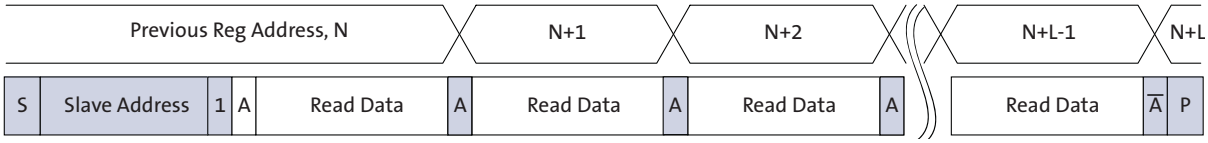
Figure 36: Sequential READ, Start from Random Location



Sequential READ, Start from Current Location

This sequence (Figure 37) starts in the same way as the single READ from current location (Figure 35 on page 52). 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.

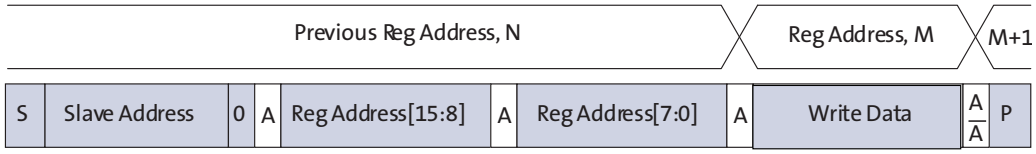
Figure 37: Sequential READ, Start from Current Location



Single WRITE to Random Location

This sequence (Figure 38) 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.

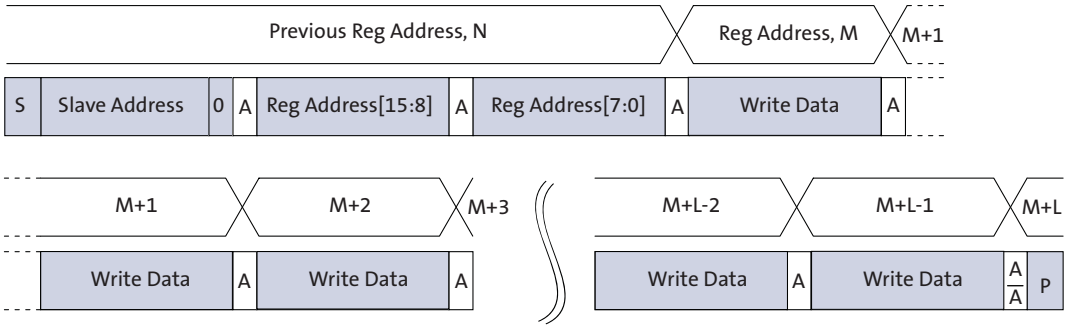
Figure 38: Single WRITE to Random Location



Sequential WRITE, Start at Random Location

This sequence (Figure 39) starts in the same way as the single WRITE to random location (Figure 38). 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 39: Sequential WRITE, Start at Random Location



Revision History

Rev. B 9/5/12

- Updated Equation 25 on page 24
- Updated Equation 36 on page 25
- Updated “PLL-Generated Master Clock” on page 26
- Updated Table 11: “PLL Parameters for the Serial Interface,” on page 27
- Updated Table 12: “Example PLL Configurations for the Serial Interface,” on page 28
- Updated “Auto Exposure” on page 35

Rev. A 6/5/12

- Initial release

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