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

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

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

Contact us

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

MT9D115

MT9D115 1/5-inch System-On-a-Chip (SOC) CMOS Digital Image Sensor

ON Semiconductor®

www.onsemi.com

Table 1. KEY PERFORMANCE PARAMETERS

Parameter	Typical Value
Pixel Size	1.75 μm × 1.75 μm
Optical Format	1/5-inch
Array Format (Active)	1600 (H) × 1200 (V) = 1.92 Mp
Imaging Area	2.8 mm × 2.10 mm, 3.50 mm Diagonal (4:3 Aspect Ratio)
CRA	25°
Color Filter Array	RGB Bayer
Scan Mode	Progressive
Shutter	Electronic Rolling Shutter (ERS)
Input Clock Range	6–54 MHz
Output Pixel Clock Maximum	85 MHz
Output MIPI Data Rate Maximum	512 Mb/s
Max. Frame Rate	15 fps Full Res 30 fps 800 x 600
Responsivity	0.65 V/Lux−sec (550 nm)
Signal-to-Noise Ratio	39 dB (MAX)
Dynamic Range	63.9 dB (Pixel)
Supply Voltage Digital Analog I/O MIPI	 1.8 V (Nominal) 2.8 V (Nominal) 1.8 V or 2.8 V (Nominal) 1.7−1.95 V
Power Consumption	196 mW (Note 1)
Operating Temperature Range	–30°C to 70°C (at Junction)
Package	Bare Die, CSP

1. Power consumption for typical voltages at 800 × 600 video mode.

Features

- 2 Mp Resolution (1600 (H) × 1200 (V))
- 1/5-inch Optical Format
- Same or Better Image Quality Compared to MT9D112
- Individual Module ID Support Through One-time Programmable (OTP) Memory
- Surface Fit Lens Correction (LC) to Compensate for Lens/Small Pixel Vignetting and Corner Color Variations

ORDERING INFORMATION

See detailed ordering and shipping information on page 2 of this data sheet.

- Automatic Functions: Exposure, White Balance, Black Level Offset Correction, Flicker Detection and Avoidance, Color Saturation Control, Defect Identification and Correction, Aperture Correction, and GPIO
- Programmable Controls: Exposure, White Balance, Horizontal and Vertical Blanking, Color, Sharpness, Gamma, Lens Shading Correction, Horizontal and Vertical Image Flip, Zoom, Windowing, Sampling Rates, and GPIO
- 15 Frames per Second (fps) at 1600(H) × 1200 (V) with Moderate Pixel Clock Frequency (≤ 64 MHz) to Minimize Baseband Reception Interference and 30 fps at 800 (H) × 600 (V)
- 2 × 2 Pixel Binning to Improve Low-light Image Quality
- Support for External LED or Xenon Flash
- On-chip Phase-locked Loop (PLL) to Minimize the Number of System Clocks
- Low Power Modes to Prolong Battery Life of Portable Devices
- Fail-safe I/Os with Programmable Output Slew Rate
- Industry Standard Two-wire Serial Interface for Controls
- 10-bit Parallel or MIPI Serial Interfaces for Image Data

Applications

- Cellular Phones
- PC Cameras
- PDAs

ORDERING INFORMATION

Table 2. AVAILABLE PART NUMBERS

Part Number	Product Description	Orderable Product Attribute Description
MT9D115D00STCK25AC1−200	2 MP 1/5″ SOC	Die Sales, 200 μm Thickness
MT9D115EB3STC−CR	2 MP 1/5″ CIS SOC	Chip Tray without Protective Film
MT9D115W00STCK25AC1−750	2 MP 1/5″ SOC	Wafer Sales, 750 μm Thickness

FUNCTIONAL DESCRIPTION

The ON Semiconductor MT9D115 is a 1/5-inch 2 Mp CMOS digital image sensor with an integrated advanced camera system. This camera system features a microcontroller (MCU), a sophisticated image flow processor (IFP), MIPI and parallel output ports (only one output port can be used at a time). The microcontroller manages all functions of the camera system and sets key operation parameters for the sensor core to optimize the quality of raw image data entering the IFP. The IFP will be responsible for processing and enhancing the image.

The entire system-on-a-chip (SOC) has superior low-light performance that is particularly suitable for PC camera applications. The MT9D115 features ON Semiconductor's breakthrough low-noise CMOS imaging technology that achieves near-CCD image quality (based on signal-to-noise ratio and low-light sensitivity) while maintaining the inherent size, cost, and integration advantages of CMOS.

The ON Semiconductor MT9D115 can be operated in its default mode or programmed for frame size, exposure, gain, and other parameters. The default mode output is a 800×600 image size at 30 frames per second (fps), assuming a 24 MHz input clock. It outputs 8-bit data, using the parallel output port.

ARCHITECTURE OVERVIEW

The MT9D115 combines a 2 Mp sensor core with an IFP to form a stand-alone solution for both image acquisition and processing. Both the sensor core and the IFP have internal registers that can be controlled by the user. In normal operation, an integrated microcontroller autonomously controls most aspects of operation. The processed image data is transmitted to the host system either through the parallel or MIPI interface.

Figure 1 shows the major functional blocks of the MT9D115.

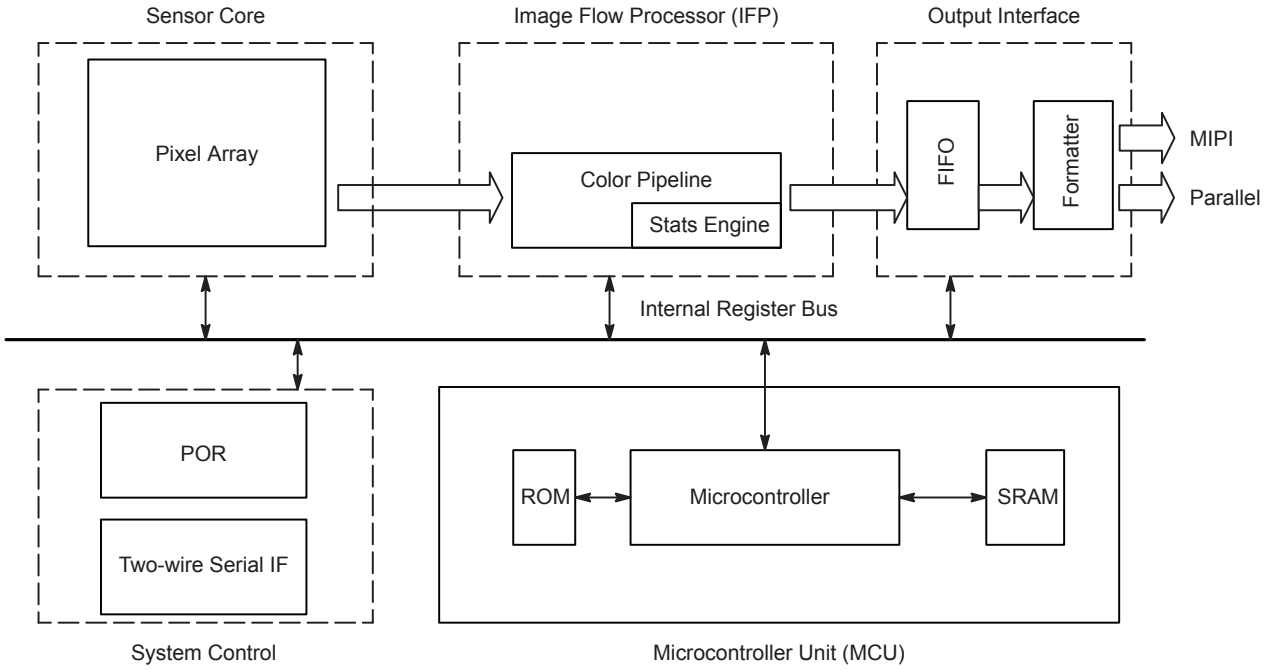
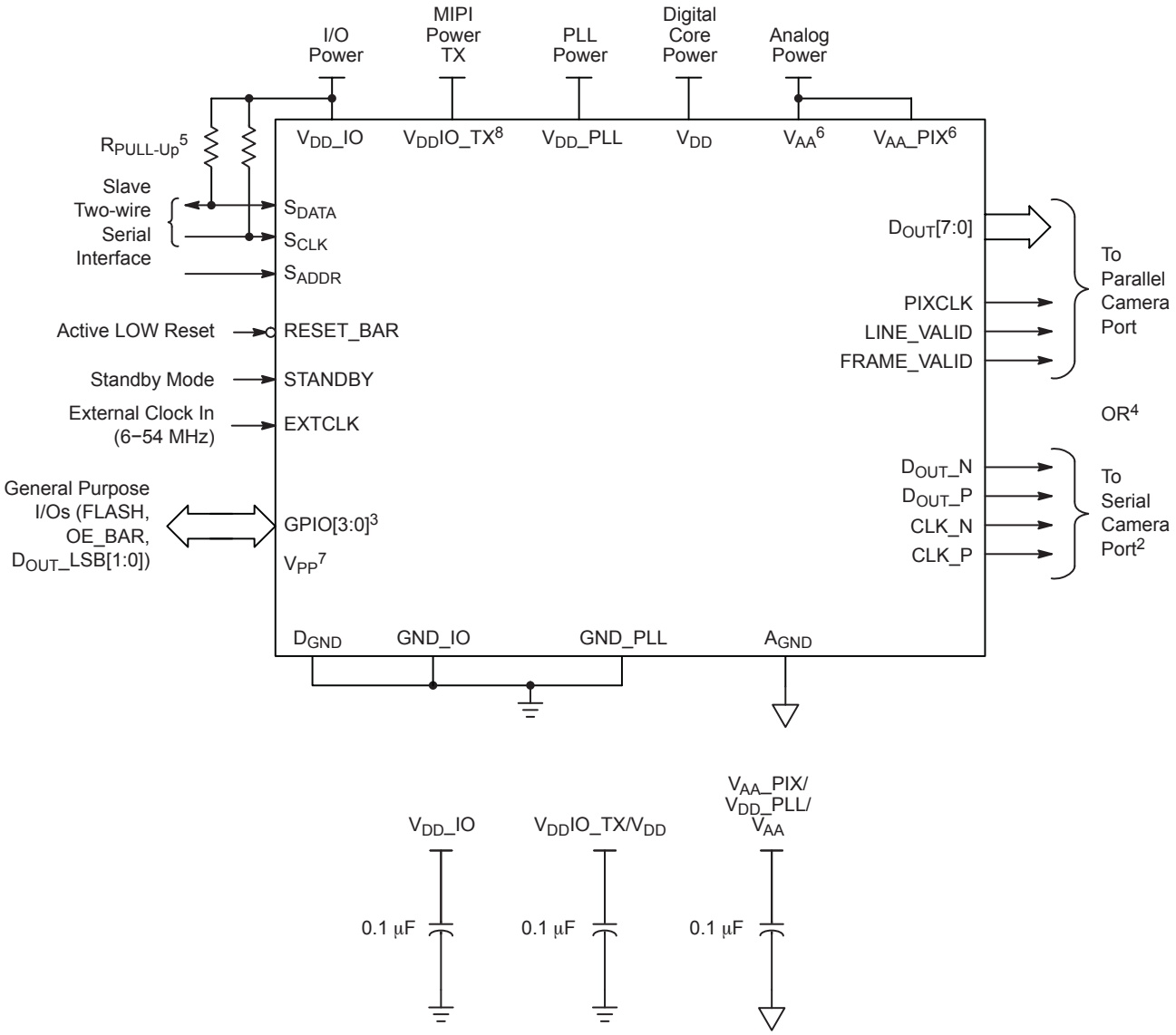


Figure 1. MT9D155 Block Diagram

TYPICAL CONNECTION



Notes:

1. This typical configuration shows only one scenario out of multiple possible variations for this sensor.
2. If a MIPI Interface is not required, the following pads must be left floating: D_{OUT}_P, D_{OUT}_N, CLK_P, and CLK_N.
3. The general purpose input/output (GPIO) pads can serve multiple features that can be reconfigured. The function and direction will vary by applications.
4. Only one of the output modes (serial or parallel) can be used at any time.
5. ON Semiconductor recommends a resistor value of 1.5 kΩ to V_{DD}_IO for the two-wire serial interface $R_{PULL-UP}$; however, greater values may be used for slower transmission speed.
6. V_{AA} and V_{AA}_PIX may be tied together. Although separate decoupling capacitors are recommended for V_{AA} and V_{AA}_PIX, decoupling capacitors can be shared if one would like to reduce module size.
7. V_{PP} is the one-time programmable memory (OTPM) programming voltage and should be left floating during normal operation.
8. 1.8 V supply shared by MIPI interface and V_{DD} to reduce number of decoupling caps, hence, module size. V_{DD}IO_TX must be connected to a 1.8 V power supply source even though MIPI interface is not used.
9. ON Semiconductor recommends that 0.1 µF and 1 µF decoupling capacitors for each power supply are mounted as close as possible to the pad and that a 10 µF capacitor be placed nearby off-module. Actual values and results may vary depending on layout and design considerations. Please follow ON Semiconductor's recommended capacitor Recommendations.
10. V_{DD}_PLL and V_{AA} can share the same power source in which case GND_PLL must be connected to GND.
11. Internal pull-up in RESET_BAR pin and can be left floating when not connected.

Figure 2. Typical Configuration (Connection)

DECOUPLING CAPACITOR RECOMMENDATIONS

It is important to provide clean, well-regulated power to each power supply. The customer is ultimately responsible for ensuring that clean power is provided for their own designs because hardware design is influenced by many factors, including layout, operating conditions, and component selection.

The recommendations for capacitor placement and values listed below are based on the ON Semiconductor internal demo camera design and verified in hardware.

ON Semiconductor recommends the following, in order of preference:

1. Mount 0.1 μF and 1 μF decoupling capacitors for each power supply as close as possible to the pad and place a 10 μF capacitor nearby off-module.
2. If module limitations allow for only six decoupling capacitors for a three-regulator design (V_{DD}1V2 tied to external regulator), use a 0.1 μF and 1 μF capacitor for each of the three regulated supplies. ON Semiconductor also recommends placing a 10 μF capacitor for each supply off-module, but close to each supply.
3. If module limitations allow for only three decoupling capacitors, use a 1 μF capacitor (preferred) or a 0.1 μF capacitor for each of the three regulated supplies. ON Semiconductor also recommends placing a 10 μF capacitor for each supply off-module but close to each supply.
4. Give priority to the V_{AA} supply for additional decoupling capacitors.

ON Semiconductor does not recommend inductive filtering components.

Follow best practices when performing physical layout. Refer to the AND9503/D.

SIGNAL DESCRIPTIONS

Table 3. SIGNAL DESCRIPTION AND DIRECTION

Name	Type	Description
STANDBY	Input	Hardware standby
EXTCLK	Input	External clock input
S_{ADDR}	Input	Two-wire interface device select address
S_{CLK}	Input	Two-wire interface serial clock
RESET_BAR (Note 4)	Input	Hardware reset
CLK_N (Note 5)	Output	MIPI differential clock N
CLK_P (Note 5)	Output	MIPI differential clock P
D_{OUT}_N (Note 5)	Output	MIPI differential data N
D_{OUT}_P (Note 5)	Output	MIPI differential data P
D_{OUT}[7:0] (Note 1)	Output	Parallel image data
FRAME_VALID (Note 1)	Output	Parallel pixel bus frame valid
LINE_VALID (Note 1)	Output	Parallel pixel bus line valid
PIXCLK (Note 1)	Output	Parallel pixel bus pixel clock
S_{DATA}	Bidirectional	Two-wire interface serial data
GPIO_0/D_{OUT}_LSB[0] (Note 2)	Bidirectional/Output	General-purpose I/O or LSB for Raw 10 data output during SOC Bypass
GPIO_1/D_{OUT}_LSB[1] (Note 2)	Bidirectional/Output	General-purpose I/O or LSB for Raw 10 data output during SOC Bypass
GPIO_3/OE_BAR (Note 2)	Bidirectional/Output	General-purpose I/O or output enable
GPIO_2/FLASH (Note 2)	Bidirectional/Output	General-purpose I/O or flash control
V_{AA}	Supply	Analog core power source 2.8 V nominal
V_{AA}_PIX	Supply	Analog core power source 2.8 V nominal
V_{DD}	Supply	Digital core power source 1.8 V nominal
V_{DD}_IO	Supply	Digital IO power source 1.8 V or 2.8 V nominal
V_{DD}_PLL	Supply	Digital PLL power source 2.8 V nominal
V_{DD}IO_TX	Supply	Digital MIPI IO power source 1.8 V nominal.

1. In serial only mode, D_{OUT}[7:0], PIXCLK, and GPIO[3:0] can be left floating by setting R0x0026[1] =1. If GPIO signals are required, D_{OUT}[7:0] and PIXCLK must be tied to D_{GND} and OE_BAR must be tied to V_{DD}_IO. GPIO_3 should be configured as an input for OE_BAR function and set R0x001A[8] = 1.
2. GPIO can be left floating if not used and must be programmed as outputs.
3. Must be connected to V_{DD}_IO, internal 100 kΩ typical at 2.8 V VDDIO used.
4. Can be left floating if not used.
5. Must be connected to V_{DD}, even in designs where the MIPI interface is not used.

ARCHITECTURE

The MT9D115 from ON Semiconductor is the third-generation, two-megapixel camera SOC. It is a microprocessor-based camera system that combines a sensor core with an image flow processor (IFP) to form a standalone solution that includes image acquisition and processing. Both the sensor core and IFP have internal registers that can be accessed by an external host. In normal operation, the integrated system microprocessor autonomously controls most aspects of operation. The image data is transmitted to the external host system either through a parallel bus or a serial MIPI interface (see Figure 3).

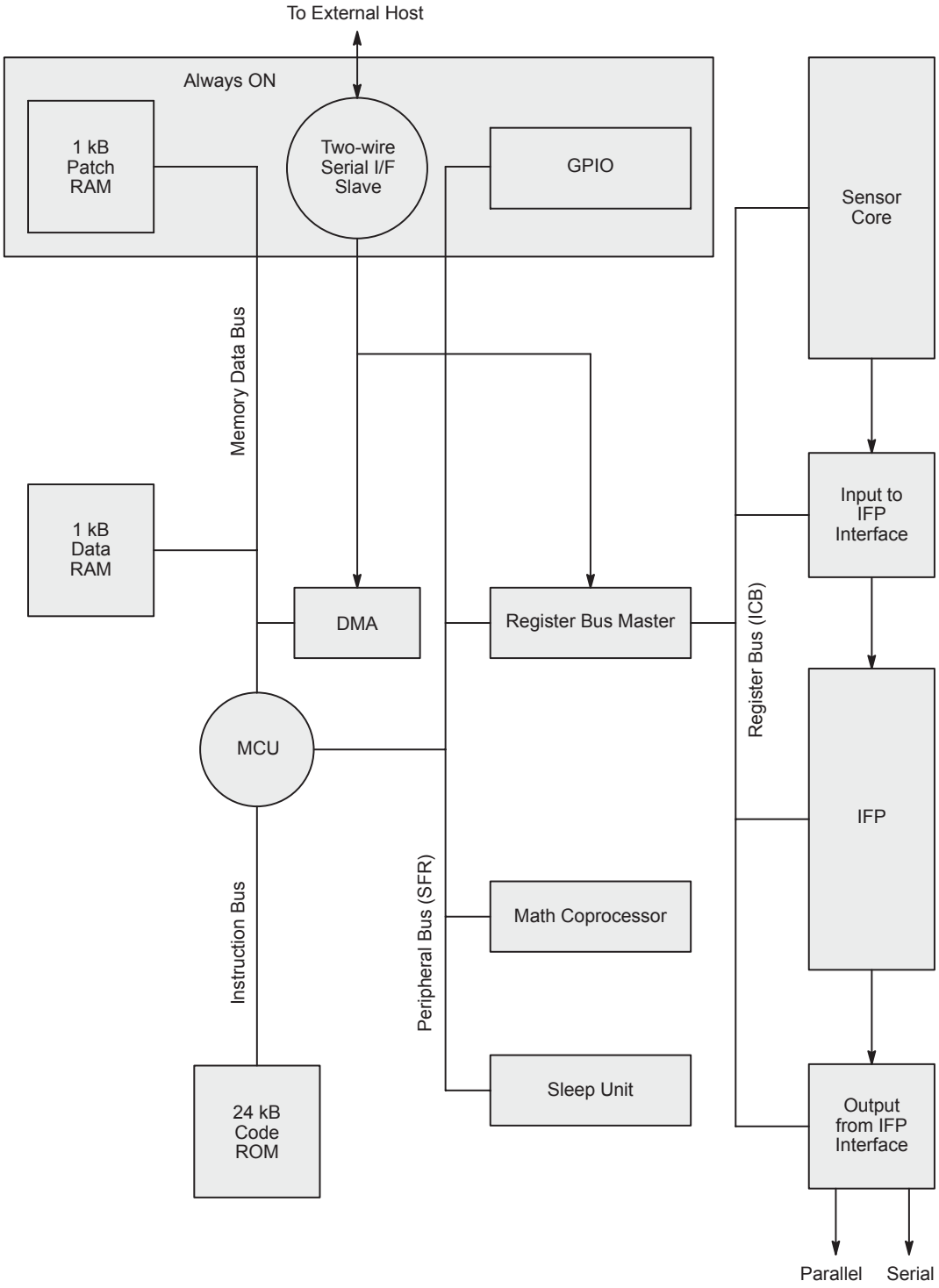


Figure 3. SOC Block Diagram

The external host and the integrated microprocessor (MCU) can both access all the internal resources (RAM and registers). The external host always has higher priority. The following sections briefly describe the functionality of each key component of the system.

Firmware

The firmware implements all automatic camera functions, such as auto exposure (AE), auto white balance (AWB), and flicker detection/avoidance (FD), as well as control functions such as sequencer, mode/context, and histogram (see Table 4). The firmware consists of drivers, generally one driver for each major automatic or control function (see Figure 4).

Table 4. LIST OF DRIVERS

Name	Type	Description
ID = 1	Sequencer	Controls of camera main function
ID = 2	AE	Auto exposure
ID = 3	AWB	Auto white balance
ID = 4	Flicker Detection	Flicker detection and avoidance
ID = 7	Mode/Context	Context variables
ID = 11	Histogram	Reduce image flare and analyze image histogram

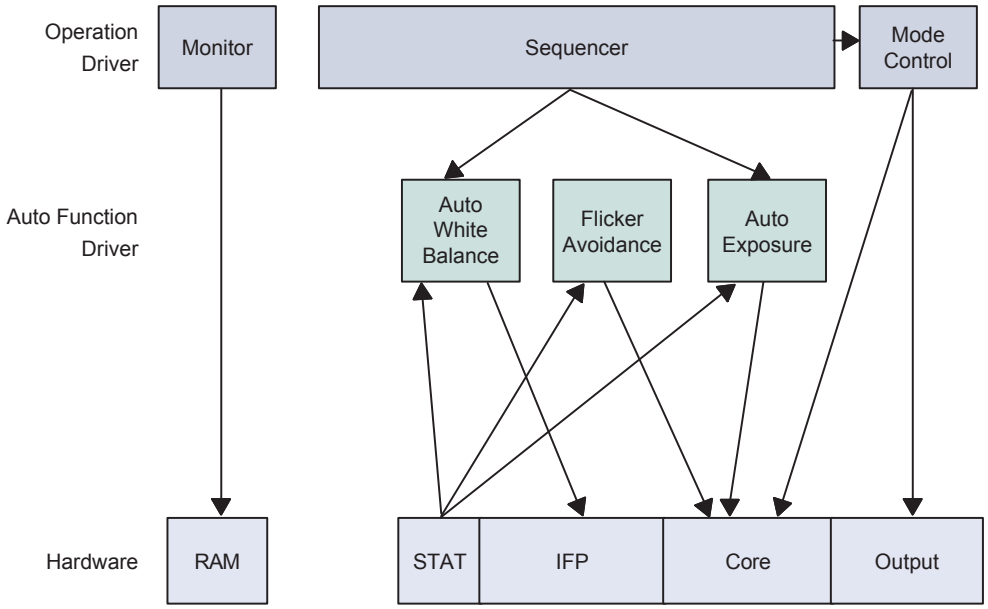


Figure 4. Firmware Architecture

External Host Interface (Two-wire Slave-only Interface)

The MT9D115 will appear as a two-wire serial interface slave to the external host. Its base address is selectable by the external S_{ADDR} pin input (when S_{ADDR} = 0 then base address = 0x78; when S_{ADDR} = 1 then base address = 0x7A). There are 32K addressable 16-bit registers (that is, the starting address of each register always falls into even addresses) within the MT9D115 but not all of them are being used (see Figure 5).

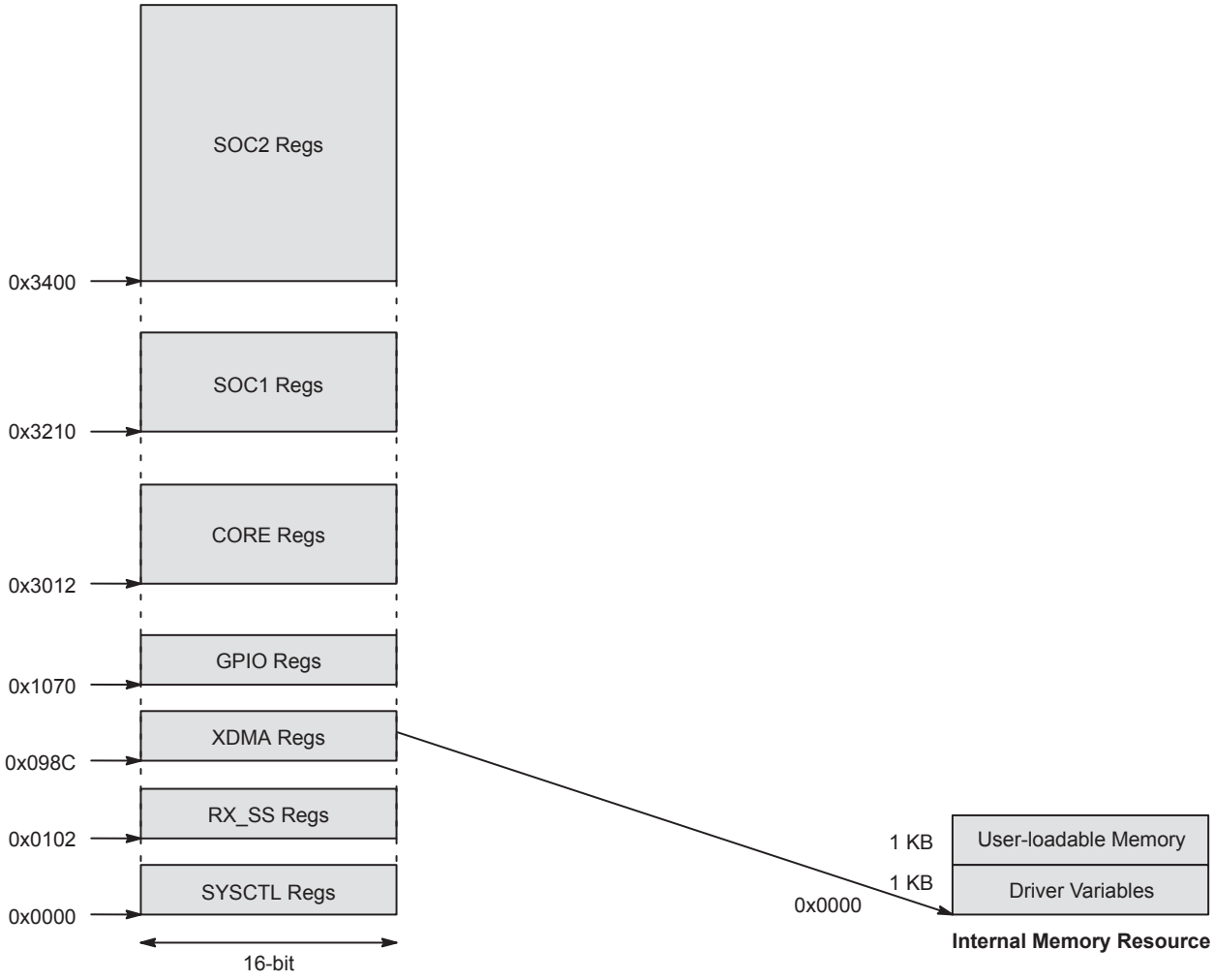


Figure 5. External Host Register Map

The MT9D115 register reference provides detailed register explanations. Although most registers are self-explanatory, the next paragraphs contain enhanced information about XDMA registers are worth explaining here.

The XDMA registers allow the external host to indirectly access the internal memory resources of the MT9D115, which include the firmware driver variables. To access the variables, use logical accesses provided by the XDMA registers.

The external host interface is implemented through a two-wire interface that enables direct read/write access to hardware registers and indirect access to firmware variables within the MT9D115. The interface is designed to be compatible with the MIPI alliance standard for Camera Serial Interface 2 (CSI−2) 1.0, which uses the electrical characteristics and transfer protocols of the two-wire serial interface specifications.

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 to the external host which acts as a master device. The master generates a clock (S_{CLK}) that is an input to the sensor and used to synchronize the transactions at the interface.

Data is transferred between the master and the slave on a bidirectional serial data bus (S_{DATA}). Both S_{CLK} and S_{DATA} are pulled up to V_{DD_IO} off-chip by a 1.5 kΩ resistor. Either the slave or master device can drive S_{DATA} to LOW − the interface determines which device is allowed to drive S_{DATA} at any given time.

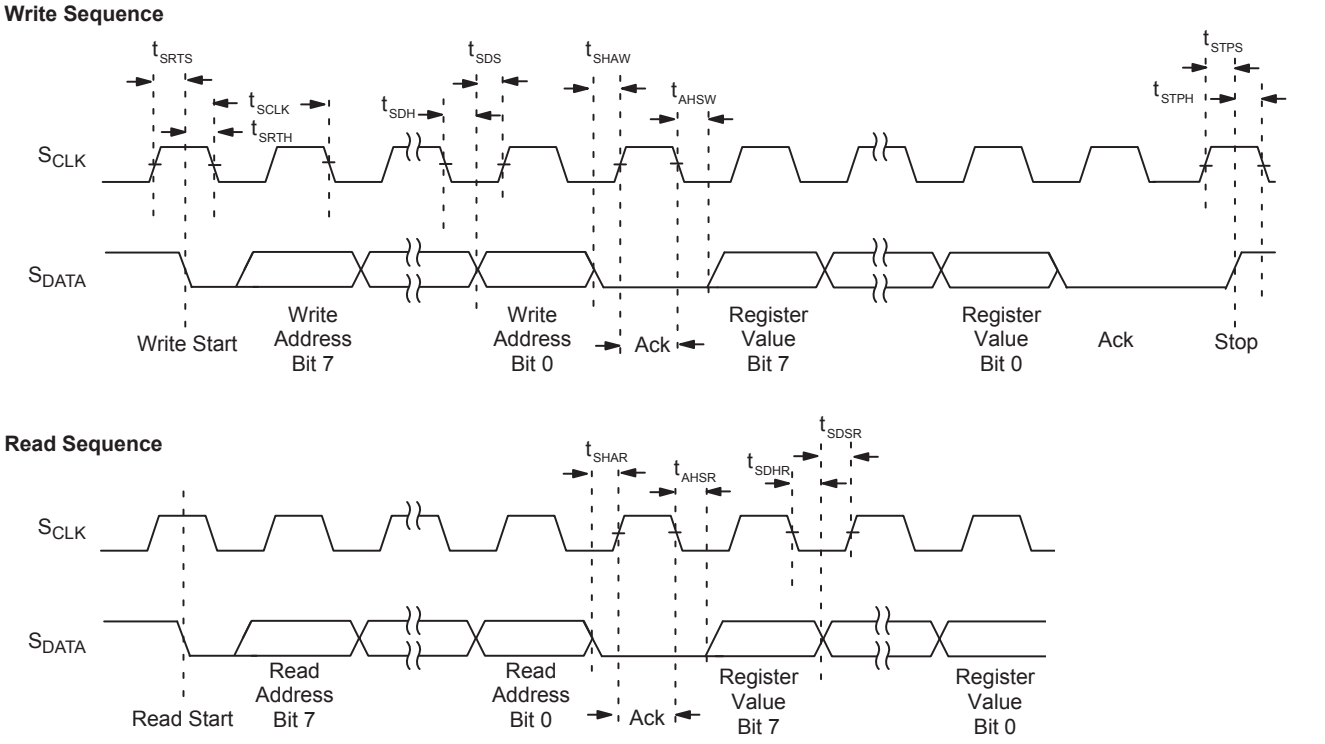


Figure 6. Two-wire Serial Control Bus Timing

Table 5. TWO-WIRE SERIAL INTERFACE TIMING DATA

(f_{EXTCLK} = 14 MHz; V_{DD} = 1.8 V; V_{DD}_IO = 1.8 V; V_{AA} = 2.8 V; V_{AA}_PIX = 2.8 V; V_{DD}_PLL = 2.8 V; V_{DD}_PHY = NA; T_J = 70°C; C_{LOAD} = 68.5 pF)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{SCLK}	Serial Interface Input Clock Frequency		100	–	400	kHz
t_{SCLK}	Serial Interface Input Clock Period		2.5	–	10	μs
	SCLK Duty Cycle		33	50	50	%
t_r	SCLK/SDATA Rise Time		–	–	300	ns
t_{SRTS}	Start Setup Time	Master Write to Slave	600	–	–	ns
t_{SRTH}	Start Hold Time	Master Write to Slave	300	–	–	ns
t_{SDH}	SDATA Hold	Master Write to Slave	5	–	900	ns
t_{SDS}	SDATA Setup	Master Write to Slave	100	–	–	ns
t_{SHAW}	SDATA Hold to Ack	Master Write to Slave	150	–	–	ns
t_{AHSW}	Ack Hold to SDATA	Master Write to Slave	150	–	–	ns
t_{STPS}	Stop Setup Time	Master Write to Slave	300	–	–	ns
t_{STPH}	Stop Hold Time	Master Write to Slave	600	–	–	ns
t_{SHAR}	SDATA Hold to Ack	Master Read from Slave	300	–	–	ns
t_{AHSR}	Ack Hold to SDATA	Master Read from Slave	300	–	–	ns
t_{SDHR}	SDATA Hold	Master Read from Slave	300	–	650	ns
t_{SDSR}	SDATA Setup	Master Read from Slave	300	–	–	ns

NOTE: t_R and t_F are dependent on system-level parameters such as the value of pull-up resistor used, how the two-wire serial bus is routed, whether there are other devices on the serial bus, and the strength of the supply used to pull-up the serial bus.

Always-On Power Domain

The always-on power domain (AOPD) provides an area of functionality that will always be active while power is applied to the MT9D115. The external host interface is located in the AOPD. The domain also includes miscellaneous clock and reset controls as well as configuration registers for the sensor core, processor core, clock configuration, and clock reset control. The user-loadable patch memory is also included in this domain. This memory will remain powered when the main core power is shut down using the standby command.

Sensor Core

The sensor core of the MT9D115 is a progressive-scan sensor that generates a stream of pixel data at a constant frame rate, qualified by LINE_VALID (LV) and FRAME_VALID (FV). The maximum pixel rate is 30 Mp/s, corresponding to a pixel clock rate of 63.25 MHz. See Figure 7 for a block diagram of the sensor core. It includes a 2.0 Mp active-pixel array. The timing and control circuitry sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and reading that row, the pixels in the row integrate incident light. The exposure is controlled by varying the time interval between reset and readout. After a row is read, data from the columns are sequenced through an analog signal chain that provides offset correction and gain, and then through an ADC. The output from the ADC is a 10-bit value for each pixel in the array.

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

The sensor core contains a set of control and status registers that can be used to control many aspects of the sensor behavior including the frame size, exposure, and gain settings. These registers are controlled by the firmware and can be accessed through a two-wire serial interface. Register values written to the sensor core can be overwritten by firmware.

The output from the core is a Bayer pattern, where 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.

A flash strobe output signal is provided to allow an external xenon or LED light source to synchronize with the sensor exposure time.

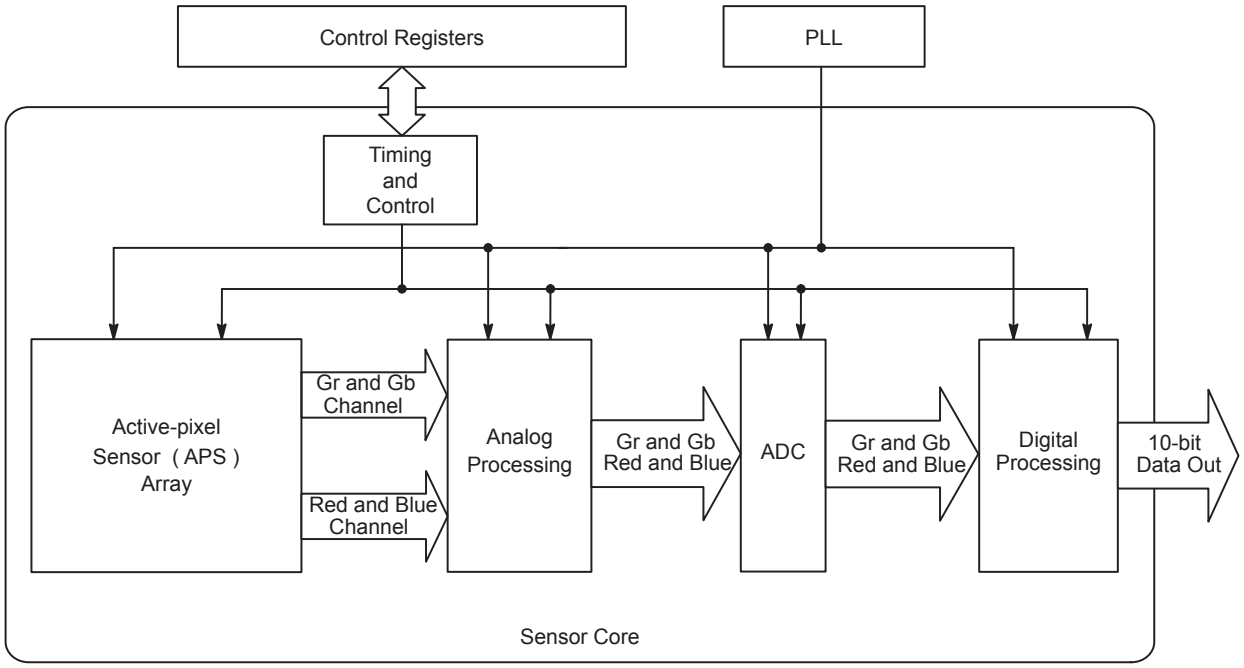


Figure 7. Sensor Core Block Diagram

Pixel Array

The sensor core uses a Bayer color pattern (see Figure 8). The even-numbered rows contain green and red pixels. The odd-numbered rows contain blue and green pixels. Even-numbered columns contain green and blue pixels. Odd-numbered columns contain red and green pixels.

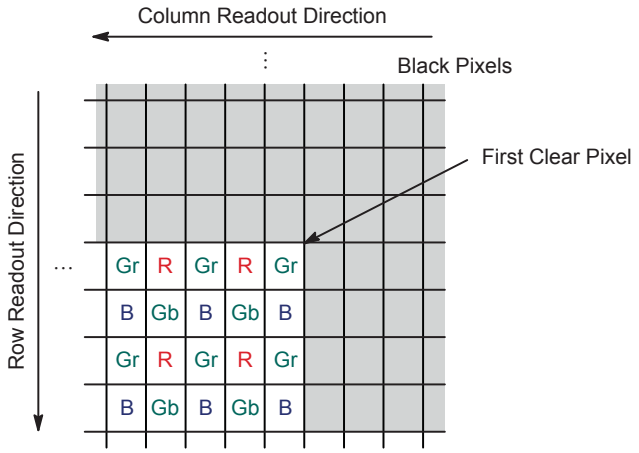


Figure 8. Pixel Color Pattern Detail (Top Right Corner)

Default Readout Order

By convention, the sensor core pixel array is shown with pixel (0,0) in the top right corner. This reflects the actual layout of the array on the die. When the sensor is operating in a system, the active surface of the sensor faces the scene (see Figure 9).

When the image is read out of the sensor, it is read one row at a time, with the rows and columns sequenced. By convention, data from the sensor is shown with the first pixel read out in the case of the sensor core in the top left corner.

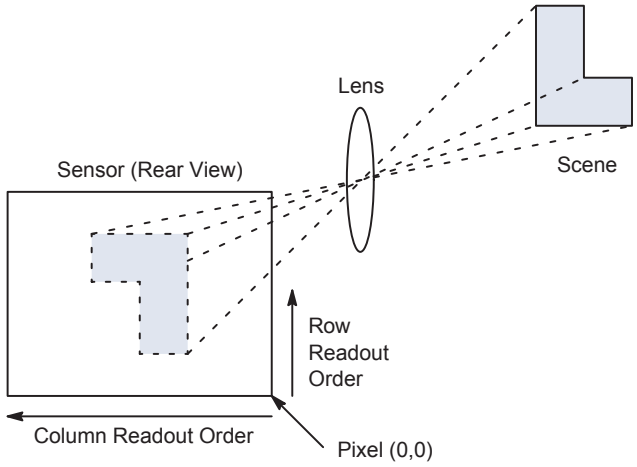


Figure 9. Imaging a Scene

Analog Processing

Analog Readout Channel

The sensor core features an analog readout channel, (see Figure 7). The readout channel consists of a gain stage, a sample-and-hold stage with black level calibration capability, and a 10-bit ADC.

Gain Options

The MT9D115 provides per-color gain control as well as the option of global gain control. The 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.

Integer digital gains in the range 0–7 can be programmed. A digital gain of 0 sets all pixel values to 0 (the pixel data will simply represent the value applied by the pedestal block). Gain settings are updated in every frame by the MCU auto functions such as AWB, AE, and FD. To make manual adjustments to gain settings, the MCU automatic exposure and automatic white balance adjustment features must be disabled.

Integration Time

The integration time (exposure) of the MT9D115 is controlled by variables. While coarse integration time controls the integration duration in terms of row times, fine integration time allows for sub-row times accuracy in terms of pixel clocks. Integration time is updated in every frame by the MCU auto feature. Disable the MCU auto features to make manual adjustments to integration time.

Because of the basic operation of the Electronic Roller Shutter (ERS), it is not advisable to set an integration time that is greater than the frame time.

It is not necessary to reprogram the frame time on the MT9D115 to make longer integration times available because the frame time adjusts automatically. However, long integration times increase the likelihood of image degradation because of increased accumulation of dark current.

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

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

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

When the integration time and the gain are changed at the same time, the gain update is held off by one frame so that the first frame output with the new integration time also has the new gain applied.

External Generated Master Clock

If application does not use PLL, then the clock bypass bit in R0x0014 must be set before exiting soft standby state as follows:

1. Write 0x25F9 to R0x0014 to set clock bypass bit.
2. Delay min. of 100 ms.
3. Write 0x4028 to R0x0018 to exit from soft standby state.
4. After successful exit from soft standby state, disable the clock bypass bit by writing 0x21F9 to R0x0014.

PLL-Generated Master Clock

The PLL can generate a master clock signal whose frequency is up to 85 MHz (input clock from 6 MHz through 54 MHz).

PLL Setup

Because the input clock frequency is unknown, the sensor starts up with the PLL disabled. The PLL takes time to power up. During this time, the behavior of its output clock signal is not guaranteed. The PLL output frequency is determined by two constants, M and N, and the input clock frequency.

$$VCO = \frac{F_{in} \times 2 \times M}{N + 1} \quad \text{(eq. 1)}$$

$$\text{PLL Output Frequency} = \frac{VCO}{P1 + 1} \quad \text{(eq. 2)}$$

Digital Processing

Readout Options

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

For preview modes, the sensor core supports both skipping and pixel averaging in x and y directions.

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

Window Size

The image output size is set with registers x_addr_start, x_addr_end, y_addr_start, and y_addr_end. The edge pixels in the 1600 × 1200 array are present to avoid edge defects and should not be included in the visible window. Binning will change the image output size.

Readout Modes

Horizontal Mirror

When the sensor is configured to mirror the image horizontally, the order of pixel readout within a row is reversed, so that readout starts from x_addr_end and ends at x_addr_start. Figure 10 shows a sequence of 6 pixels being read out with normal readout and reverse readout. The SOC corrects for this change in sensor core output.

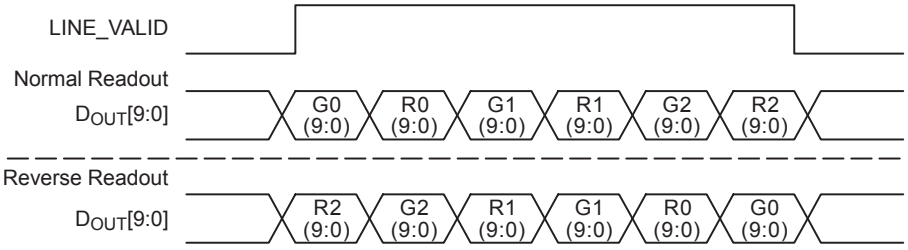


Figure 10. Six Pixels in Normal and Column Mirror Readout Modes

Vertical Flip

When the sensor is configured to flip the image vertically, the order in which pixel rows are read out is reversed, so that row readout starts from y_addr_end and ends at y_addr_start. Figure 11 shows a sequence of six rows being read out with normal readout and reverse readout. The SOC corrects for this change in sensor core output.

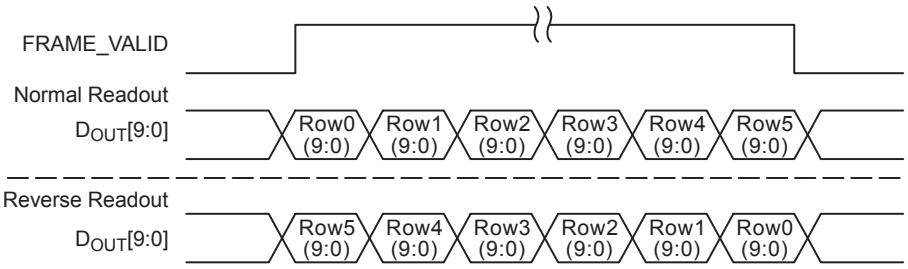


Figure 11. Six Rows in Normal and Row Mirror Readout Modes

Column and Row Skip

The sensor core supports subsampling. Subsampling reduces the amount of data processed by the analog signal chain in the sensor and thereby allows the frame rate to be increased. This reduces the amount of row and column data processed and is equivalent to the skip2 readout mode provided by earlier ON Semiconductor image sensors. Set the proper image output and crop sizes before enabling subsampling.

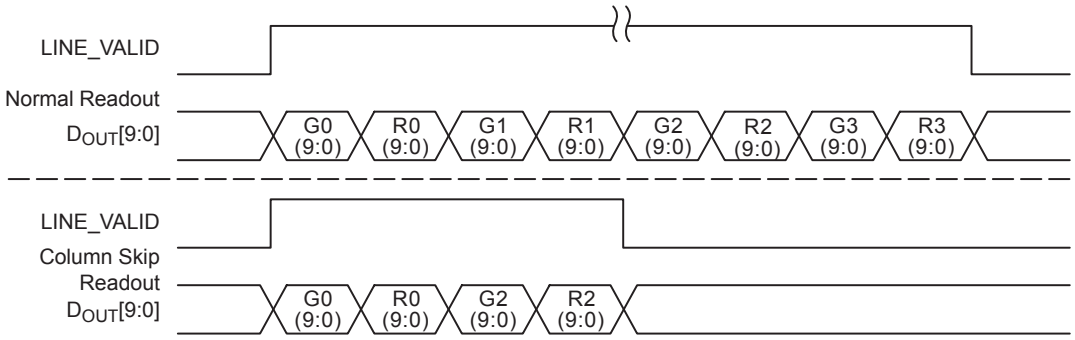


Figure 12. Eight Pixels in Normal and Column Skip 2X Readout Modes

Pixel Readouts

Figure 13 through Figure 16 show a sequence of data being read out with no skipping, with *x_odd_inc = 3* and *y_odd_inc = 1*, with *x_odd_inc = 1* and *y_odd_inc = 3*, and with *x_odd_inc = 3* and *y_odd_inc = 3*.

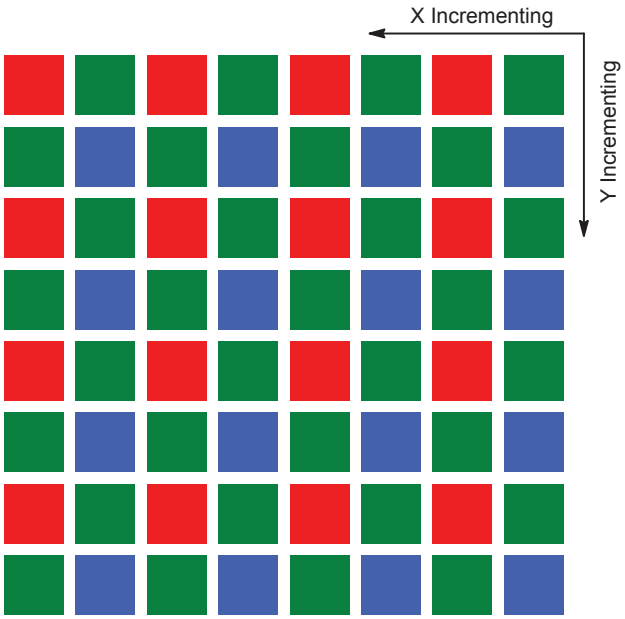


Figure 13. Pixel Readout (No Skipping)

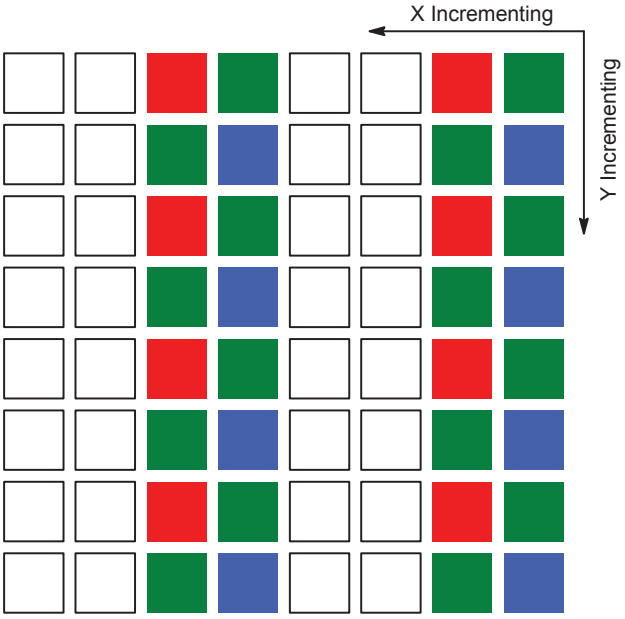


Figure 14. Pixel Readout (x_odd_inc = 3, y_odd_inc = 1)

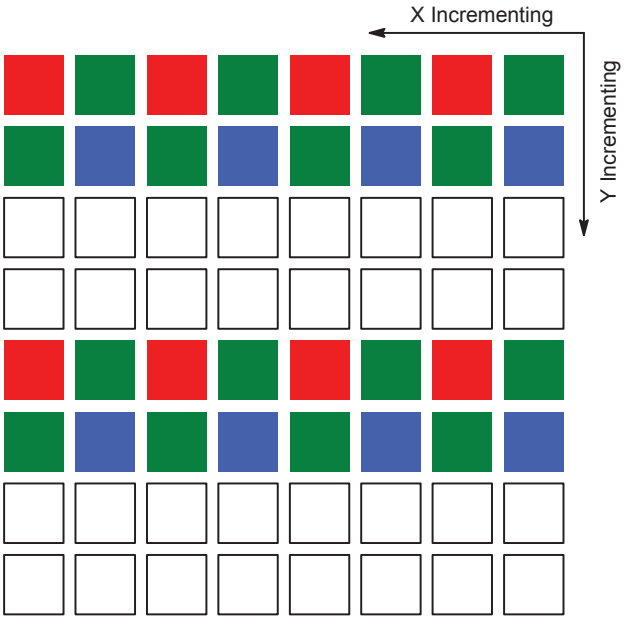


Figure 15. Pixel Readout (x_odd_inc = 1, y_odd_inc = 3)

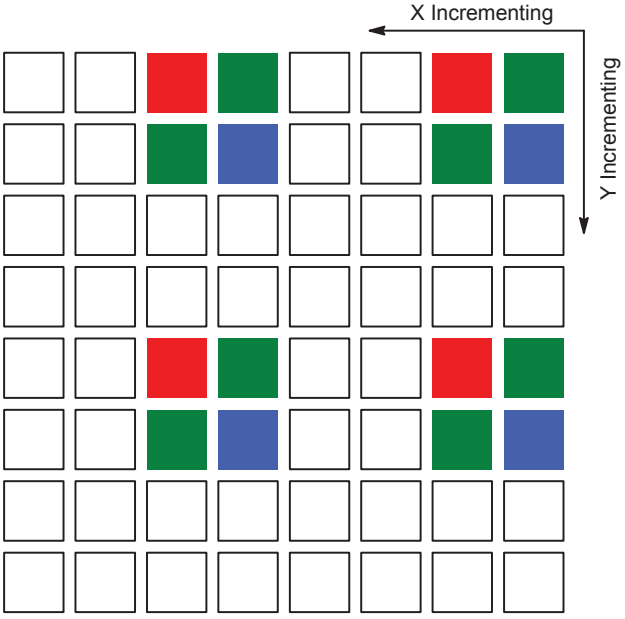


Figure 16. Pixel Readout (x_odd_inc = 3, y_odd_inc = 3)

Programming Restrictions when Skipping

When skipping is enabled as a viewfinder mode, and the sensor is switched back and forth between full resolution and skipping, keep *line_length_pck* constant. This allows the same integration times to be used in each mode.

When subsampling is enabled, it might be necessary to adjust the x_addr_end and y_addr_end settings. The values for these registers must correspond with rows/ columns that form part of the subsampling sequence. Use the following rules to make the adjustment.

$$\text{remainder} = (\text{addr\_end} - \text{addr\_start} + 1) \text{ AND } 4 \quad \text{(eq. 3)}$$

$$\text{if (remainder} == 0) \text{ addr\_end} = \text{addr\_end} - 2 \quad \text{(eq. 4)}$$

Table 6 shows the row address sequencing for normal and subsampled (with *y_odd_inc = 3*) readout. The same sequencing applies to column addresses for subsampled readout. Because the subsampling sequence only reads half of the rows and columns, there are two possible subsampling sequences, depending upon the alignment of the start address.

Table 6. ROW ADDRESS SEQUENCING (SAMPLING)

Normal	Subsampled Sequence 1	Subsampled Sequence 2
0	0	No Data
1	1	No Data
2	No Data	2
3	No Data	3
4	4	No Data
5	5	No Data
6	No Data	6
7	No Data	7

Binning

The MT9D115 sensor core supports 2×1 and 2×2 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. However, because binning gathers image data from all pixels in the active window, rather than from a subset of pixels, it achieves superior image quality and avoids the aliasing artifacts that can be a characteristic side effect of subsampling.

Enable binning by selecting the appropriate subsampling settings (*x_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 register. As for subsampling, x_addr_end and y_addr_end might require adjustment when binning is enabled.

The effect of the different subsampling settings is shown in Figure 17 and in Figure 18.

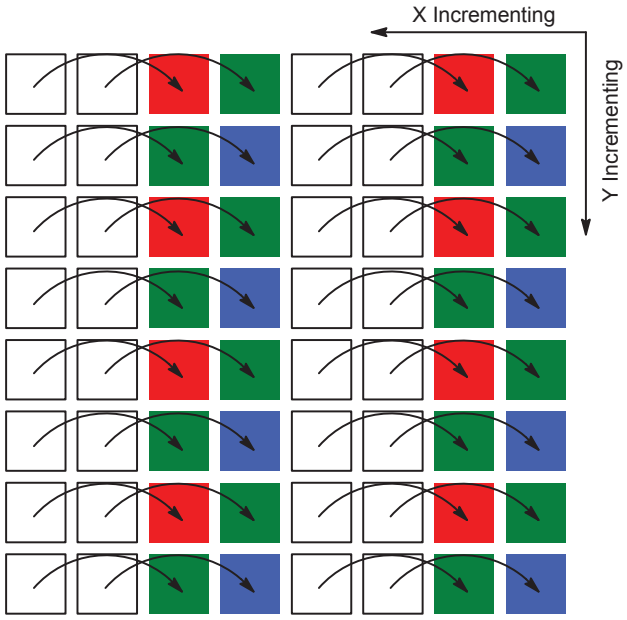


Figure 17. Pixel Readout (x_odd_inc = 3, y_odd_inc = 1, x_bin = 1)

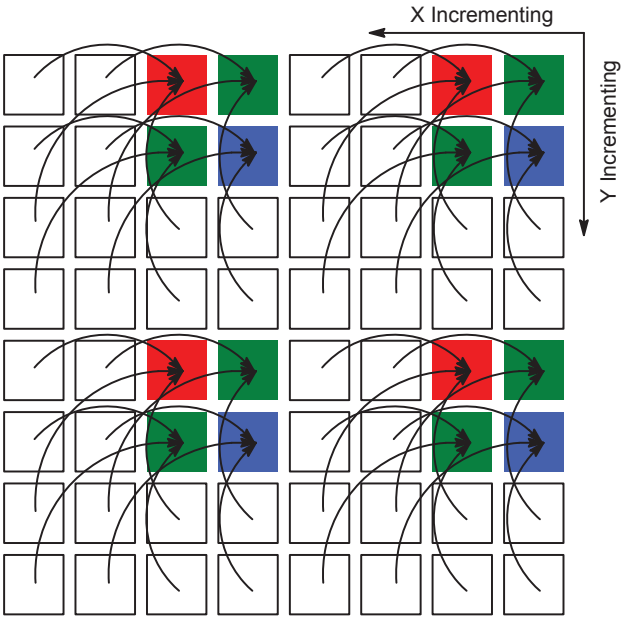


Figure 18. Pixel Readout (x_odd_inc = 3, y_odd_inc = 3, xy_bin = 1)

Binning Limitations

Binning requires a 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.

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. None of these adjustments are required for x-binning. The sensor must be taken out of streaming mode before switching between binned and non-binned operation. The row addresses for various binning modes are shown in Table 7.

Table 7. ROW ADDRESS SEQUENCING (BINNING)

Normal	Binning Sequence 1	Binning Sequence 2
0	0, 2	No Data
1	1, 3	No Data
2	No Data	2, 4
3	No Data	3, 5
4	4, 6	No Data
5	5, 7	No Data
6	No Data	6, 8
7	No Data	7, 9

Raw Data Format

The sensor core image data is read out in a progressive scan. Valid image data is surrounded by horizontal blanking and vertical blanking, (see Figure 19). The amount of horizontal blanking and vertical blanking is programmable. LV is HIGH during the shaded region of the figure.

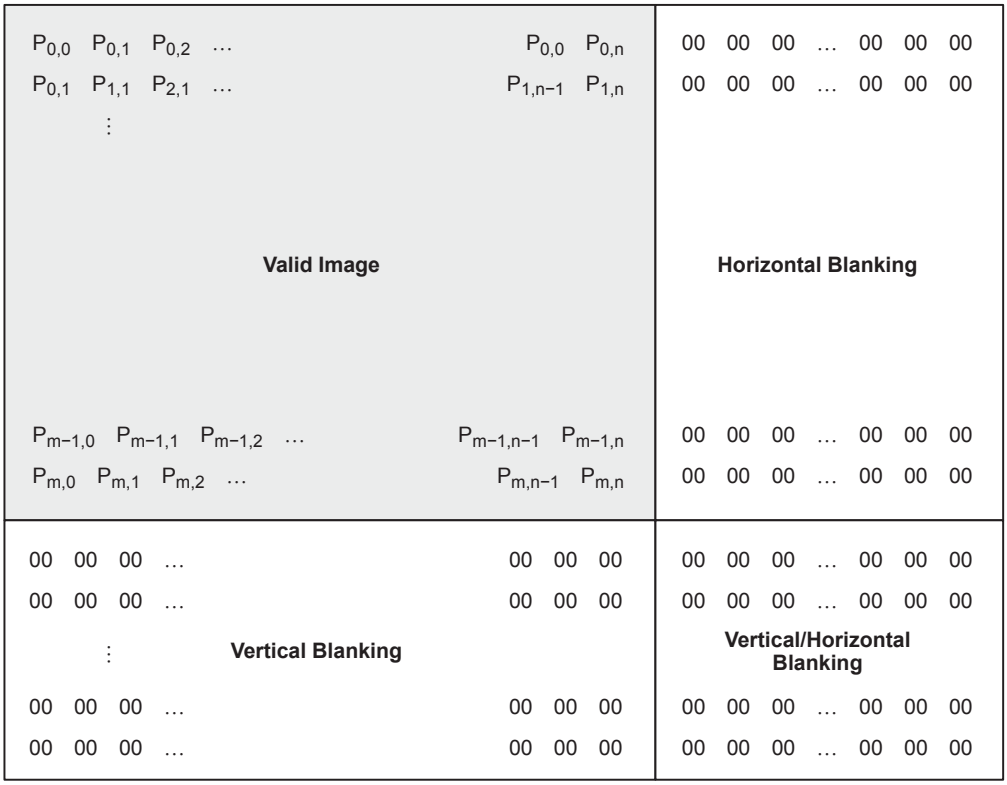


Figure 19. Valid Image Data

Raw Data Timing

The sensor core output data is synchronized with the PIXCLK output. When LV is HIGH, one pixel's data is output on the 10-bit D_{OUT} output bus every PIXCLK period. By default, the PIXCLK signal runs at the same frequency as the master clock, and its falling edges occur one-half of a master clock period after transitions on LV, FV, and D_{OUT}[9:0] (see Figure 20). This allows PIXCLK to be used as a clock to sample the data. PIXCLK is continuously enabled, even during the blanking period.

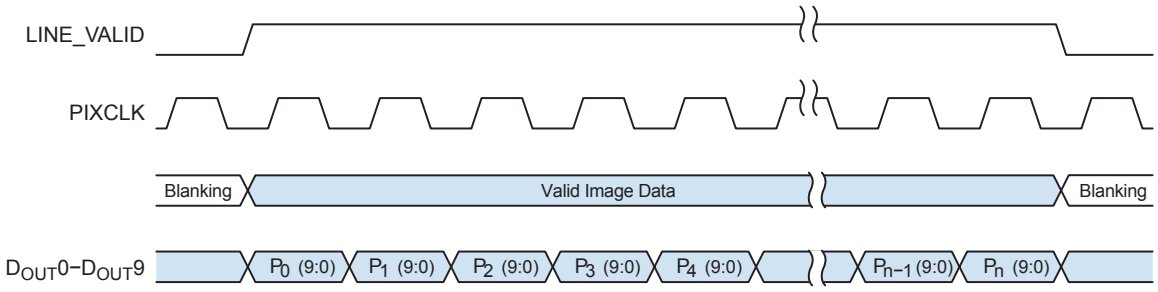


Figure 20. Pixel Data Timing Example

Input Interface to Image Flow Processor

The input interface to the IFP is the front end of the IFP, in which it will choose between the sensor core output or a test pattern generator output. During normal operation, a stream of raw Bayer image data from the sensor is continuously fed into the IFP. For testing purposes, the test generator output is selected. The generator provides a selection of test patterns sufficient for basic testing of the IFP. Program variable (ID = 7, Offset = 0x66) followed by REFRESH command to the sequencer in order to access different test patterns (see Table 8).

Depending on which test pattern has been selected, the user might need to program additional registers in order to see the intended effects.

Table 8. AVAILABLE TEST PATTERNS

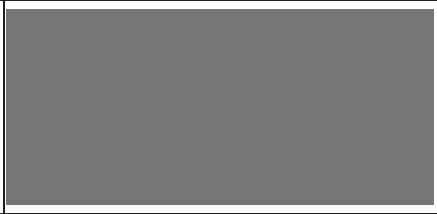
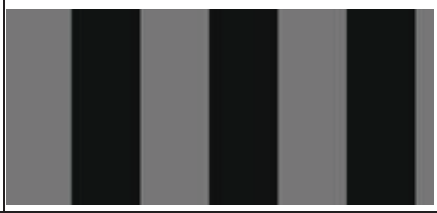
Test Pattern	Registers/Variables	Example
Flat Field	mode_common settings_test_mode (ID = 7, Offset = 0x66) = 1 test_pxl_red (R0x0102) = 0x1ff test_pxl_g1 (R0x0104) = 0x1ff test_pxl_g2 (R0x0106) = 0x1ff test_pxl_blue (R0x0108) = 0x1ff	
Vertical Ramp	mode_common_ mode_settings test_mode (ID = 7, Offset = 0x66) = 2	
Color Bar	mode_common_ mode_settings test_mode (ID = 7, Offset = 0x66) = 3	
Vertical Stripes	mode_common settings_test_mode (ID = 7, Offset = 0x66) = 4 test_pxl_red (R0x0102) = 0x1ff test_pxl_g1 (R0x0104) = 0x17d test_pxl_g2 (R0x0106) = 0x000 test_pxl_blue (R0x0108) = 0x000	
Pseudo-Random	mode_common_mode settings_test_mode (ID = 7, Offset = 0x66) = 5	
Horizontal Stripes	mode_common settings_test_mode (ID = 7, Offset = 0x66) = 6 test_pxl_red (R0x0102) = 0x1ff test_pxl_g1 (R0x0104) = 0x17d test_pxl_g2 (R0x0106) = 0x000 test_pxl_blue (R0x0108) = 0x000	

Image Flow Processor

Most IFP functions can be controlled directly by the MCU. The external host will control it indirectly through the driver variables. IFP processing can be broken into three different phases: pixel reconstruction, color rendering/ statistics collection, and digital scaling/output format. Figure 21 shows a simplified IFP block diagram and the operating color space at each processing phase.

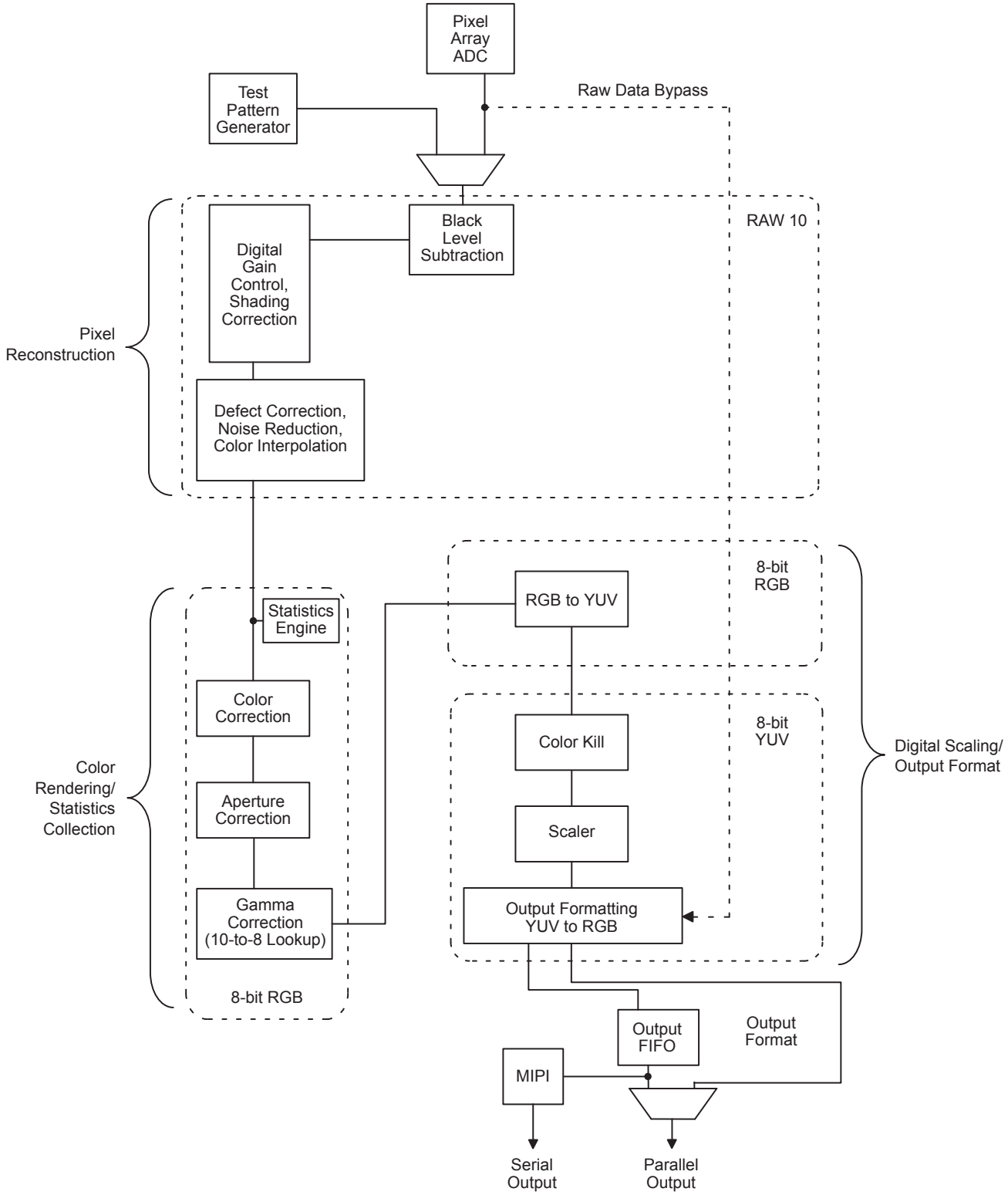


Figure 21. IFP Block Diagram

Pixel Reconstruction (Lens Shading Correction)

First Black Level Subtraction and Digital Gain

Image stream processing starts with black level subtraction and multiplication of all pixel values by a programmable digital gain. Both operations can be independently set to separate values for each color channel (R, Gr, Gb, B). Independent color channel digital gain can be adjusted with registers. Independent color channel black level adjustments can also be made. If the black level subtraction produces a negative result for a particular pixel, the value of this pixel is set to "0".

Shading Correction (SC)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. Other factors also cause fixed pattern signal gradients in images captured by image sensors. The cumulative result of all factors is known as image shading. The MT9D115 has an embedded shading correction module that can be programmed to counter the shading effects on each individual R, Gb, Gr, and B color signal.

The Correction Function

Color-dependent solutions are calibrated using the sensor, lens system, and an image of an evenly illuminated, featureless grey calibration field. The color correction functions can be derived from the resulting image.

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

$$P_{corrected}(row, col) = P_{sensor}(row, col) \times f(row, col) \quad \text{(eq. 5)}$$

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

Defect Correction

The IFP performs continuous defect correction that can mask pixel array defects such as high dark-current (hot) pixels and pixels that are darker or brighter than their neighbors due to photo-response non-uniformity. The module is edge-aware with exposure that is based on configurable thresholds. The thresholds are changed continuously based on the brightness of the current scene.

Noise Reduction

Noise reduction can be enabled or disabled. Thresholds can be set through register settings.

Color Interpolation and Edge Detection

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

Aperture Correction

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

Color Rendering/Statistics Collection (Color Correction)

Color Correction

To achieve good color fidelity of the IFP output, interpolated RGB values of all pixels are subjected to color correction. The IFP multiplies each vector of three pixel colors by a 3×3 color correction matrix. The three components of the resulting color vector are all sums of three 10-bit numbers. Because such sums can have up to 12 significant bits, the bit width of the image data stream is widened to 12 bits per color (36 bits per pixel). The color correction matrix can be programmed by the user or automatically selected by the AWB algorithm implemented in the IFP. Ideally, color correction should produce output colors that are independent of the spectral sensitivity and color crosstalk characteristics of the image sensor. The optimal values of the color correction matrix elements depend on those sensor characteristics and on the spectrum of light incident on the sensor. The color correction variables can be adjusted through register settings.

Image Cropping

By configuring the cropped and output windows to various sizes, different zooming levels (for example 4X, 2X, and 1X) can be achieved. The location of the cropped window is also configurable so that panning is also supported. A separate cropped window is defined for context A and context B. In both contexts, the height and width definitions for the output window must be equal to or smaller than the cropped image.

Gamma Correction

The gamma correction curve (see Figure 22) is implemented as a piecewise linear function with 19 knee points, taking 12-bit arguments and mapping them to 8-bit output. The abscissas of the knee points are fixed at 0, 64, 128, 256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 3584, 3840, and 4096. The 8-bit ordinates are programmable through IFP registers.

The MT9D115 IFP includes a block for gamma correction that can adjust its shape based on brightness to enhance the performance under certain lighting conditions. Two custom gamma correction tables can be uploaded, one corresponding to a brighter lighting condition, the other corresponding to a darker lighting condition. At power-up,

the IFP loads the two tables with default values. The final gamma correction table used depends on the brightness of the scene and can take the form of either uploaded tables or an interpolated version of the two tables. A single (non-adjusting) table for all conditions can also be used.

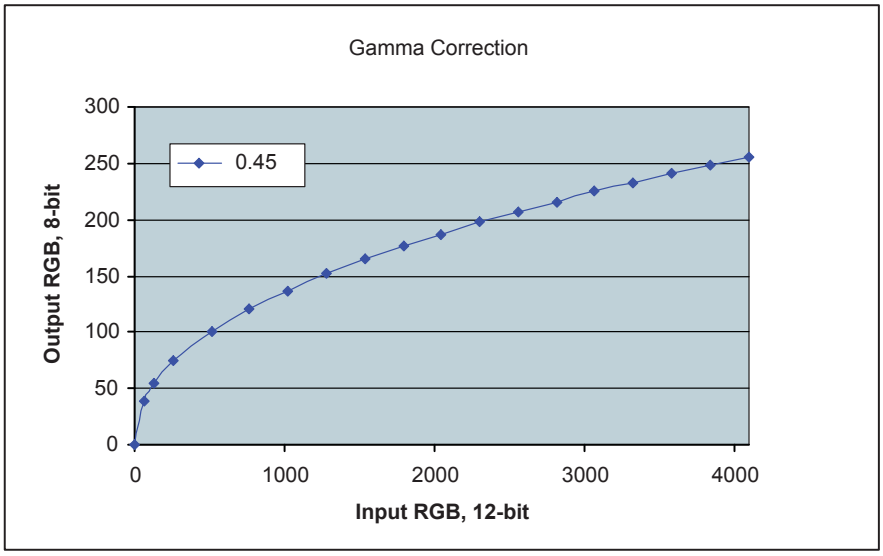


Figure 22. Gamma Correction Curve

Color Kill

A color kill circuit is included to remove high or low light color artifacts. It affects only pixels whose luminance exceeds a certain preprogrammed threshold. The U and V values of those pixels are attenuated proportionally to the difference between their luminance and the threshold.

Digital Scaling/Output Format

Special Effects

Special effects like negative image, sepia, or black and white can be applied to the data stream at this point. These effects can be enabled and selected by registers.

RGB to YUV Conversion

For further processing, the data is converted from RGB color space to YUV color space.

YUV Color Filter

As an optional processing step, noise suppression by one-dimensional low-pass filtering of Y and/or UV signals is possible. A 3- or 5-tap filter can be selected for each signal.

Image Scaling

The IFP includes a scaler module to ensure that the size of images output by the MT9D115 can be tailored to the needs of all users. When enabled, this module performs rescaling of incoming images − shrinks them to arbitrarily selected width and height without reducing the field of view and without discarding any pixel values.

The scaler performs pixel binning − divides each input image into rectangular bins corresponding to individual pixels of the desired output image, averages pixel values in these bins, and assembles the output image from the bin averages. Pixels lying on bin boundaries contribute to more than one bin average; their values are added to bin-wide sums of pixel values with fractional weights. The entire procedure preserves all image information that can be included in the downsized output image and filters out high frequency features that could cause aliasing.

Use the image cropping and scaler module together to implement a digital zoom and pan. If the scaler is programmed to output images smaller than images coming from the sensor core, zoom effect can be produced by cropping the latter from their maximum size down to the size of the output images. The ratio of these two sizes determines the maximum attainable zoom factor. For example, a 1600×1200 image rendered on a 250×150 display can be zoomed up to eight times, because $1280/160 = 1024/128 = 8$. A panning effect can be achieved by fixing the size of the cropping window and moving it around the pixel array.

YUV-to-RGB/YUV Conversion and Output Formatting

The YUV data stream emerging from the scaling module can either exit the color pipeline as-is or be converted before exit to an alternative YUV or RGB data format.

Color Conversion Formulas

- Y'U'V':
  This conversion is ITU−R BT.601 scaled to make YUV range from 0 through 255. This setting is recommended for JPEG encoding and is the most popular.

$$Y' = 0.299 \times R' + 0.587 \times G' + 0.114 \times B' \quad \text{(eq. 6)}$$

$$U' = 0.564 \times (B' - Y') + 128 \quad \text{(eq. 7)}$$

$$V' = 0.713 \times (R' - Y') + 128 \quad \text{(eq. 8)}$$

  There is an option where 128 is not added to U'V'.

- Y'Cb'Cr' Using sRGB Formulas:
  The MT9D115 implements the sRGB standard. This option provides YCbCr coefficients for a correct 4:2:2 transmission.

NOTE: $16 < Y' < 235$; $16 < Cb < 240$; $16 < Cr < 240$; and $0 \geq RGB \geq 255$

$$Y' = (0.2126 \times R' + 0.7152 \times G' + 0.0722 \times B') \times \times (219/256) + 16 \quad \text{(eq. 9)}$$

$$Cb' = 0.5389 \times (B' - Y') \times (224/256) + 128 \quad \text{(eq. 10)}$$

$$Cr' = 0.635 \times (R' - Y') \times (224/256) + 128 \quad \text{(eq. 11)}$$

- Y'U'V' Using sRGB Formulas:
  Similar to the previous set of formulas, but has YUV spanning a range of 0 through 255.

$$Y' = 0.2126 \times R' + 0.7152 \times G' + 0.0722 \times B' \quad \text{(eq. 12)}$$

$$U' = 0.5389 \times (B' - Y') + 128 = \quad \text{(eq. 13)}$$
$$= -0.1146 \times R' - 0.3854 \times G' + 0.5 \times B' + 128$$

$$V' = 0.635 \times (R' - Y') + 128 = \quad \text{(eq. 14)}$$
$$= 0.5 \times R' - 0.4542 \times G' - 0.0458 \times B' + 128$$

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

$$R' = Y + 1.5748 \times (V - 128) \quad \text{(eq. 15)}$$

$$G' = Y - 0.1873 \times (U - 128) - 0.4681 \times (V - 128) \quad \text{(eq. 16)}$$

$$B' = Y + 1.8556 \times (U - 128) \quad \text{(eq. 17)}$$

Output Interface from IFP

The output interface contains parallel image bus, MIPI serial bus interfaces, and walking 1s connectivity test generator.

Parallel and MIPI Output

The user can select to use either the serial MIPI output or the 10-bit parallel output to transmit the data. Only one of the output modes can be used at any time.

The parallel output can be used with an output FIFO whose memory is shared with the MIPI output FIFO to retain a constant pixel output clock independent from the scaling factor.

When scaling the image or skipping lines, the data would be generated in bursts and the pixel clock would turn on and off in intervals, which might lead to EMI problems. The output FIFO will group all active pixel data together so the pixel clock can be run at a constant speed.

The MIPI output transmitter implements a serial differential sub-LVDS transmitter capable of up to 512 Mb/s. It supports multiple formats, error checking, and custom short packets. The MIPI clock is defined as half the data rate frequency.

The virtual channel could be used by host MIPI RX circuits to differentiate between preview and capture image data if the FW changes the channel number when switching between contexts. The host cannot adequately time this switching of contexts and so cannot use channel number in this way without FW support.

The hardware supports a configurable MIPI virtual channel in the MIPI Control register (R0x3400). Firmware provides two variables that allow the MIPI virtual channel to be changed according to context (preview/A, capture/B). The host can specify what channel is used for which context through these variables.

Table 9. DATA FORMATS SUPPORTED BY MIPI INTERFACE

Data Format	Data Type
YUV 422 8-bit	0x1E
565RGB	0x22
555RGB	0x21
444RGB	0x20
RAW8	0x2A
RAW10	0x2B

NOTE: Data will be packed as RAW8 if the data type specified does not match any of the above data types.

Output Format and Timing

YUV/RGB Output

YUV or RGB data can be output either directly from the output formatting block or through a FIFO buffer with a capacity of 1024 bytes. This size is large enough to hold one-fourth of a scan line at full resolution. Buffering of data is a way to equalize the data output rate when image scaling is used. Scaling produces an intermittent data stream consisting of short high-rate bursts separated by idle periods. High pixel clock frequency during bursts may be undesirable due to EMI concerns.

Figure 23 shows the output timing of a YUV/RGB scan line when a scaled data stream is equalized by buffering or when no scaling takes place. The pixel clock frequency remains constant during each LV high period. Scaled data is output at a lower frequency than full size frames, which helps to reduce EMI.

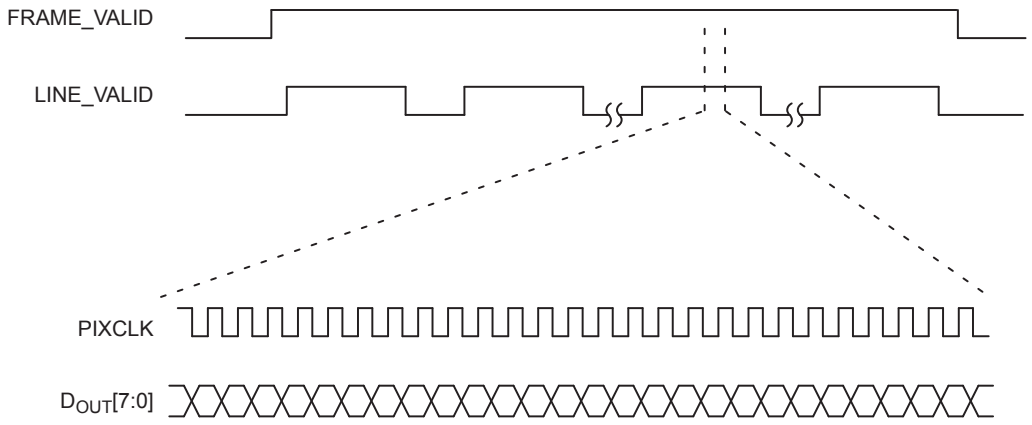


Figure 23. Timing of Full Frame Data or Scaled Data Passing through the FIFO

YUV/RGB Data Ordering

The MT9D115 supports swapping YCbCr mode (see Table 10).

Table 10. YCbCr OUTPUT DATA ORDERING

Mode	Data Sequence			
Default (No Swap)	Cb_i	Y_i	Cr_i	Y_{i+1}
Swapped CrCb	Cr_i	Y_i	Cb_i	Y_{i+1}
Swapped YC	Y_i	Cb_i	Y_{i+1}	Cr_i
Swapped CrCb, YC	Y_i	Cr_i	Y_{i+1}	Cb_i

The RGB output data ordering in default mode is shown in Table 11. The odd and even bytes are swapped when luma/chroma swap is enabled. R and B channels are bit-wise swapped when chroma swap is enabled.

Table 11. RGB ORDERING IN DEFAULT MODE

Mode (Swap Disabled)	Byte	D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
565RGB	Odd	R_7 R_6 R_5 R_4 R_3 G_7 G_6 G_5
	Even	G_4 G_3 G_2 B_7 B_6 B_5 B_4 B_3
555RGB	Odd	0 R_7 R_6 R_5 R_4 R_3 G_7 G_6
	Even	G_4 G_3 G_2 B_7 B_6 B_5 B_4 B_3
444xRGB	Odd	R_7 R_6 R_5 R_4 G_7 G_6 G_5 G_4
	Even	B_7 B_6 B_5 B_4 0 0 0 0
x444RGB	Odd	0 0 0 0 R_7 R_6 R_5 R_4
	Even	G_7 G_6 G_5 G_4 B_7 B_6 B_5 B_4

10-Bit Bypass Output

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

1. Using eight data output signals ($D_{OUT}[7:0]$) and GPIO[1:0]. The GPIO signals are the lowest two bits of data.
2. Using only eight signals ($D_{OUT}[7:0]$) and a special 8 + 2 data format, shown in Table 12.

Table 12. 2-BYTE RGB FORMAT

Byte	Bits Used	Bit Sequence
Odd Bytes	8 Data Bits	D_9 D_8 D_7 D_6 D_5 D_4 D_3 D_2
Even Bytes	2 Data Bits + 6 Unused Bits	0 0 0 0 0 0 D_1 D_0

FIFO

During normal pipeline operation, the output data rate is determined by a number of factors: input image size, degree of scaling, and sensor operation mode. As these parameters change during normal sensor operation, output frequency changes. This output frequency may generate RF noise, interfering with the mobile device. By using an output FIFO to maintain a constant output clock frequency, noise is easily filtered out.

The FIFO accumulates data and after a certain number of bytes are stored, it will output them in a single burst, making sure that the data rate within the burst remains constant. This approach utilizes a free-running clock, thus making possible minimal RF interference.

CONTROL FUNCTIONS

Sequencer: Camera Operating System

The sequencer is a finite state machine that controls the general operations of the camera and switching between operating modes. Camera operation is organized in states such as enter preview, preview, leave preview, enter capture, and capture. The sequencer carries out a number of commands such as run, go preview, go capture, refresh, and refresh mode. The current state of the sequencer is indicated in a variable seqr.state. To execute a command, the user must set a particular command number in the variable *seqr.cmd*. ON Semiconductor recommends that the external host monitor *seqr.state* to know when to change resolution or capture frames.

Each state has its configuration; therefore, the user should set up the state configuration to customize the camera configuration before executing the corresponding program such as GO TO CAPTURE. A typical camera operating scenario is:

1. Configure mode variables after hardware reset.
2. Configure preview mode.
3. Execute sensor REFRESH program.
4. Run in preview until shutter button is pressed.
5. Capture a frame.

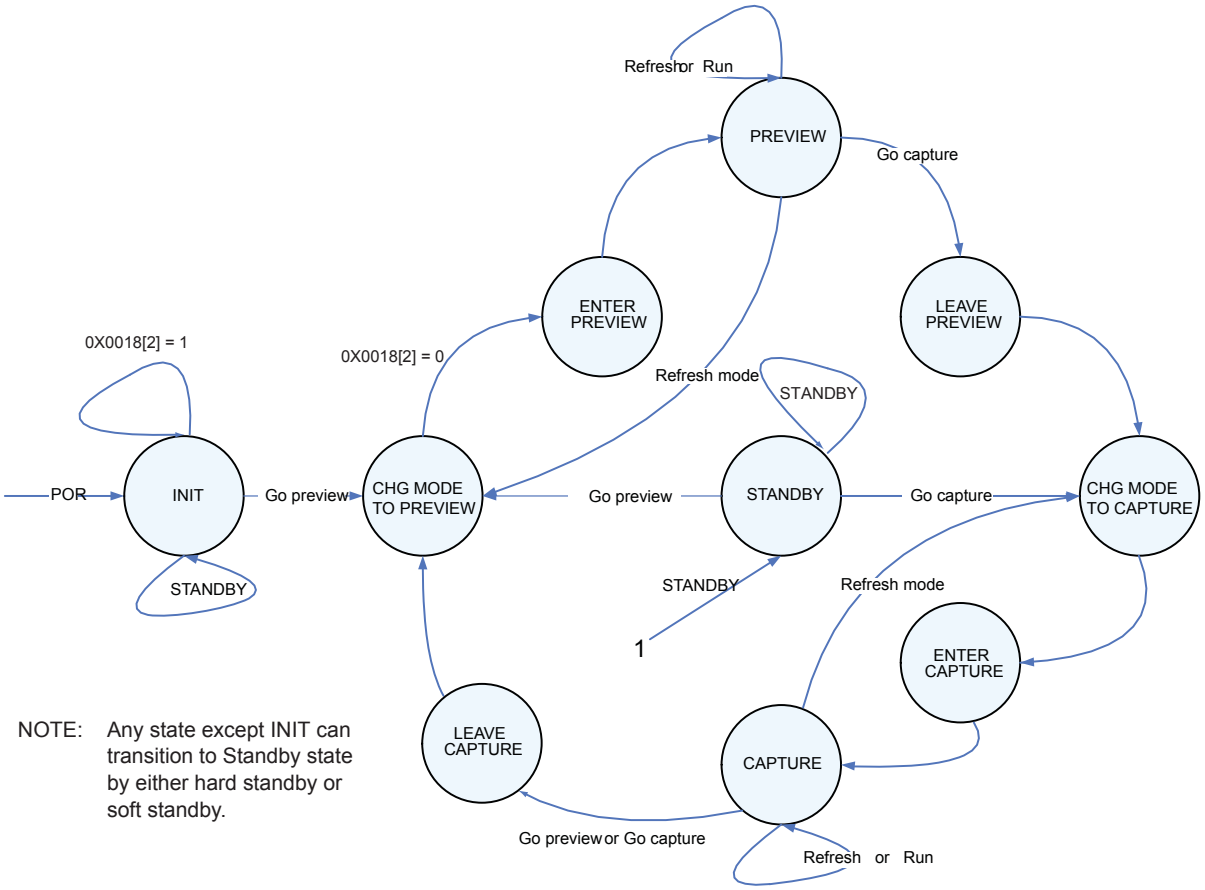


Figure 24. Sequencer Finite State Machine

Mode: Context Information

The mode driver reduces integration efforts by managing most aspects of switching the two contexts. It remembers vital register values for each image acquisition context and loads these values to the appropriate registers in the IFP upon context switching.

For the mode driver variables to take effect, the user changes the variable values in the mode driver (ID = 7). Upon the next mode change or sequencer's REFRESH command, these driver variable values will be loaded to the appropriate physical sensor core and IFP registers. The image processing will take the new values at the beginning of the next frame acquired.

To control the output image size, the user can modify the mode driver variables such as *output_width_A*, *output_height_A*, *output_width_B*, and *output_height_B*. The mode driver will automatically apply any appropriate downscaling filter to achieve this output image size as well as update the watermark of the output FIFO. It is important to set up the sensor core to output an image equal to or larger than the crop window size, which in turn is equal to or larger than the desired output image size.