

1/3.2-Inch 13 Mp CMOS Digital Image Sensor

AR1337 Datasheet, Rev. 1

For the latest datasheet, please visit www.hidmgroup.com

Features

- 13 Mp CMOS sensor with advanced 1.1 μm pixel BSI technology
- Phase Detection Auto Focus supported
- On Chip PDAF Optimized Defect Pixel Correction
- On Chip PDAF Calculations
- HSYNC output signal on GPIO for synchronizing PWM/OIS Driver Implementation
- Data interfaces: two-, three-, and four-lane serial mobile industry processor interface (MIPI)
- Bit-depth compression available for MIPI Interface: 10-8 and 10-6 to enable lower bandwidth receivers for full frame rate applications
- 3D synchronization controls to enable stereo video capture
- 6.8 kbits one-time programmable memory (OTPM) for storing shading correction coefficients and module information
- Programmable controls: gain, horizontal and vertical blanking, auto black level offset correction, frame size/rate, exposure, left-right and top-bottom image reversal, window size, and panning
- Two on-die phase-locked loop (PLL) oscillators for super low noise performance
- On-chip temperature sensor
- Bayer pattern horizontal down-size scaler
- Simple two-wire fast-mode+ serial interface
- Low dark current
- On-chip lens shading correction
- Support for external mechanical shutter
- Support for external LED or Xenon Flash
- Extended Flash duration up to start of frame readout

Applications

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

Table 1: Key Performance Parameters

Parameter		Value
Optical format		1/3.2 -inch 13 Mp (4:3)
Active pixels		4208H x 3120V
Pixel size		1.1 μm Back Side Illuminated (BSI)
Chief ray angle (CRA)		32°
Die size		6.3 mm x 5.7 mm
Input clock frequency		6 - 48 MHz
Interface		4-lane MIPI (2- and 3-lane supported); Max data rate: 1.2Gbps/lane
Subsampling modes		Column full resolution and row skip2 Column full resolution and row bin2 Column full resolution and row skip4 Column full resolution and row skip2bin2step4
ADC resolution		10 bits, on-die
Analog gain		1x – 7.75x
Digital gain		Up to 7.98x
Scaler		Adjustable scaling up to 8x
Temperature sensor		10-bit, controlled by two-wire serial I/F
Compression		DPCM: 10-8-10, 10-6-10
3D support		Frame rate and exposure synchronization
Supply voltage	VAA, VAA_PIX	2.6 V - 2.9 V (2.7 V nominal ¹)
	VDD_IO, VDDIO_ANA	1.8 V (1.7 V < V _{ripple} < 1.9 V)
	VDD, VDD_ANA, VDD_PLL, VDD_PHY	1.2 V (1.14 V < V _{ripple} < 1.3 V)
Power consumption		TBD mW at 60°C (TYP) at 13 Mp 30 fps
Responsivity		TBD e ⁻ /lux-sec
SNR _{MAX}		TBD dB
Dynamic Range		TBD dB
Operating Temperature Range (at junction) - Tj		-30°C to +70°C

Note: 1. Do not use supply voltage higher than max operation voltage to avoid damage to the sensor.



AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor Ordering Information

Table 2: Mode of Operation and Power

Mode	Resolution	Mode	HFOV	FPS	Power Consumption [mW]
4:3 Snapshot Mode (PDAF supported)					
13 M full resolution	4208 x 3120	13M full mode	100%	30	TBD
16:9 Video Mode 30FPS (PDAF supported)					
4K UHD	3840 x 2160	cropping	91%	30	TBD
4K Cinema	4096 x 2160	cropping	97%	30	TBD
1080p	1920 x 1080	Crop+Subsampling+Scaling	91%	30	TBD
16:9 Video Mode 60FPS (PDAF supported)					
1080p	1920 x 1080	Crop+Subsampling+Scaling	91%	60	TBD
4:3 Snapshot Mode (PDAF not supported)					
VGA	640 x 480	Crop+Subsampling+Scaling	61%	120	TBD
QVGA	320 x 240	Crop+Subsampling+Scaling	30%	240	TBD
16:9 Video Mode 30FPS (PDAF not supported)					
720p	1280 x 720	Crop+Subsampling+Scaling	91%	30	TBD
16:9 Video Mode 60FPS (PDAF not supported)					
720p	1280 x 720	Crop+Subsampling+Scaling	91%	60	TBD

Ordering Information

Table 3: Available Part Numbers

Part Number	Product Description	Orderable Product Attribute Description
AR1337CSSC32SMD20	13 MP 1/3" CIS	Recon Die
AR1337CSSC32SMD20-E	13 MP 1/3"	CIS Recon Die Samples

See the HIDM Device Nomenclature document (TND310/D) for a full description of the naming convention used for image sensors. For reference documentation, including information on evaluation kits, please visit our web site at www.hidmgroup.com.



Table of Contents

Features.....	1
Applications.....	1
Ordering Information.....	2
General Description.....	6
Functional Overview.....	6
Pixel Array.....	9
Operating Modes.....	9
Typical Connections.....	10
Signal Descriptions.....	11
Output Data Format.....	12
Two-Wire Serial Register Interface.....	12
Registers.....	17
Reading the Sensor Revision Number.....	20
Programming Restrictions.....	21
Control of the Signal Interface.....	24
MIPI Virtual Channel and Data Type Interleave.....	25
System States.....	27
Soft Reset Sequence.....	29
General Purpose Input and Output.....	30
Streaming/Standby Control.....	30
Clocking.....	31
Features.....	33
Additional Features.....	44
Gain.....	46
Sensor Core Digital Data Path.....	49
Phase Detection Autofocus (PDAF).....	54
HSYNC for OIS.....	56
Timing Specifications.....	58
Image Sensor Characteristics.....	62
Electrical Characteristics.....	64
Revision History.....	71

List of Figures

Figure 1:	Block Diagram	7
Figure 2:	Pixel Color Pattern Detail (Top Right Corner)	9
Figure 3:	Typical Connections	10
Figure 4:	Single READ from Random Location	14
Figure 5:	Single READ from Current Location	14
Figure 6:	Sequential READ, Start from Random Location	15
Figure 7:	Sequential READ, Start from Current Location	15
Figure 8:	Single WRITE to Random Location	15
Figure 9:	Sequential WRITE, Start at Random Location	16
Figure 10:	Effect of Limiter on the Data Path	22
Figure 11:	Timing of Data Path	23
Figure 12:	Virtual Channel Interleave	26
Figure 13:	Data Type Interleave	26
Figure 14:	AR1337 System States	28
Figure 15:	AR1337 Profile 1/2 Clocking Structure	31
Figure 16:	Effect of horizontal mirror on Readout Order	35
Figure 17:	Effect of vertical flip on Readout Order	35
Figure 18:	Pixel Readout (No Subsampling)	36
Figure 19:	Pixel Readout (x_odd_inc = 1, y_odd_inc = 3)	37
Figure 20:	Pixel Readout (x_odd_inc = 1, y_odd_inc = 5)	37
Figure 21:	Pixel Readout (x_odd_inc = 1, y_odd_inc = 7)	38
Figure 22:	Xenon Flash Enabled	43
Figure 23:	LED Flash Enabled	43
Figure 24:	100 Percent Color Bars Test Pattern	51
Figure 25:	Fade-to-Gray Color Bars Test Pattern	52
Figure 26:	Walking 1s 10-bit Pattern	52
Figure 27:	Walking 1s 8-bit Pattern	53
Figure 28:	PDAF Pixel Pattern	55
Figure 29:	PDAF Pixel at Each Row	56
Figure 30:	Power-Up Sequence	58
Figure 31:	Power-Down Sequence	59
Figure 32:	Hard Standby and Hard Reset	60
Figure 33:	Soft Standby and Soft Reset	61
Figure 34:	Quantum Efficiency vs. Wavelength TBD	62
Figure 35:	Chief Ray Angle vs. Image Height	63
Figure 36:	Two-Wire Serial Bus Timing Parameters	64

List of Tables

Table 1:	Key Performance Parameters	1
Table 2:	Mode of Operation and Power	2
Table 3:	Available Part Numbers	2
Table 4:	Signal Descriptions	11
Table 5:	Address Space Regions	17
Table 6:	Data Formats	18
Table 7:	Programming Rules	21
Table 8:	CSI-2 Packet Header Control Register Fields	25
Table 9:	XSHUTDOWN and PLL in System States	28
Table 10:	Signal State During Reset	29
Table 11:	General Purpose Input and Output Pad Functions	30
Table 12:	Streaming/STANDBY	30
Table 13:	Definitions for Programming Rules	36
Table 14:	Row Address Sequencing During Subsampling	39
Table 15:	Minimum Row Time and Blanking Numbers	42
Table 16:	Minimum Frame Time and Blanking Numbers	42
Table 17:	Compression Registers	44
Table 18:	Recommended Gain Settings	46
Table 19:	Reference of ISO-speed-based Gain Settings	48
Table 20:	Test Patterns	49
Table 21:	Pixel Information	54
Table 22:	PDAF Line Number	56
Table 23:	Power-Up Sequence	59
Table 24:	Recommended Power-Down Timing	59
Table 25:	Array Dimensions	63
Table 26:	CRA Value	63
Table 27:	Two-Wire Serial Register Interface Electrical Characteristics	65
Table 28:	Two-Wire Serial Interface Timing Specifications	66
Table 29:	Electrical Characteristics (EXTCLK)	67
Table 30:	Electrical Characteristics (Serial MIPI Pixel Data Interface)	68
Table 31:	DC Electrical Characteristics (Control Interface)	68
Table 32:	DC Electrical Definitions and Characteristics	69



AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor General Description

General Description

The HIDM AR1337 is a 1/3.2-inch BSI (back side illuminated) PDAF supported CMOS active-pixel digital image sensor with a pixel array of 4208H x 3120V (4224H x 3136V including border pixels). The unique feature of AR1337 is high performance Phase Detect Auto Focus (PDAF) pixel technology which enables a fast auto focus camera system. It uses on-chip PDAF pixel defect correction to output a fully corrected image and on-chip calculations which provide AF correlation data (or raw PDAF data) via virtual channel or second MIPI data type.

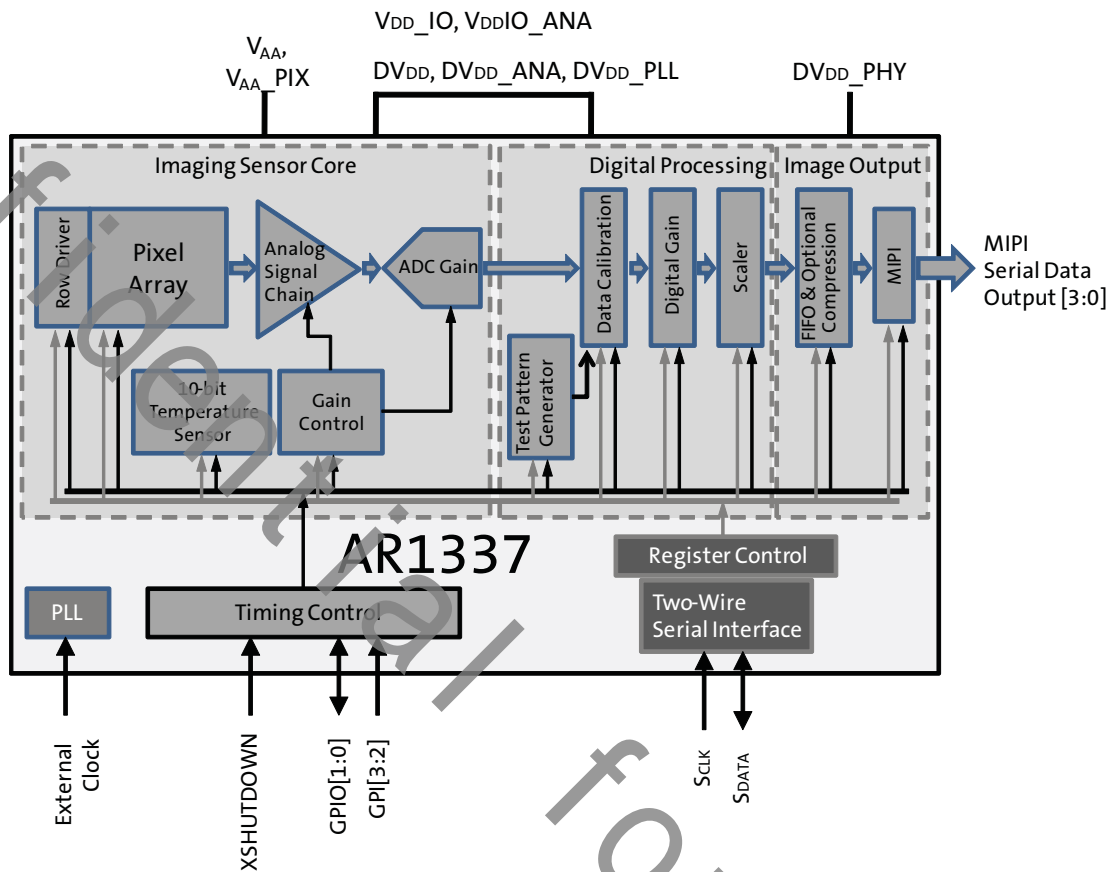
It incorporates sophisticated on-chip camera functions such as mirroring, column and row skip modes, and snapshot mode. It is programmable through a simple two-wire serial interface and has very low power consumption.

The AR1337 digital image sensor features HIDM'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 AR1337 sensor can generate full resolution image at up to 30 frames per second (fps). An on-chip analog-to-digital converter (ADC) generates a 10-bit value for each pixel.

Functional Overview

The AR1337 is a progressive-scan sensor that generates a stream of pixel data at a constant frame rate. It uses an on-chip, phase-locked loop (PLL) to generate all internal clocks from a single master input clock running between 6 and 48 MHz. The maximum data rate is 1.2 Gbps per lane. A block diagram of the sensor is shown in Figure 1.

Figure 1: Block Diagram


The core of the sensor is a 13Mp active-pixel array. The timing and control circuitry sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and reading that row, the pixels in the row integrate incident light. The exposure is controlled by varying the time interval between reset and readout. Once a row has been read, the data from the columns is sequenced through an analog signal chain (providing offset correction and gain), and then through an ADC. The output from the ADC is a 10-bit value for each pixel in the array. The ADC output passes through a digital processing signal chain (which provides further data path corrections and applies digital gain).

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

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

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



AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor Functional Overview

The control registers, timing and control, and digital processing functions shown in Figure 1 on page 7 are partitioned into three logical parts:

- A sensor core that provides array control and data path corrections. The output of the sensor core is a 10-bit serial pixel data stream qualified by a 4-lane MIPI output clock.
- Data processing functions, including a digital shading correction block to compensate for color/brightness shading introduced by the lens or chief ray angle (CRA) curve mismatch, digital gain, and dynamic defect correction
- Output processing functions, including a horizontal scaler, a limiter, a data compressor, an output FIFO, and a serializer.

The output FIFO is present to prevent data bursts by keeping the data rate continuous.

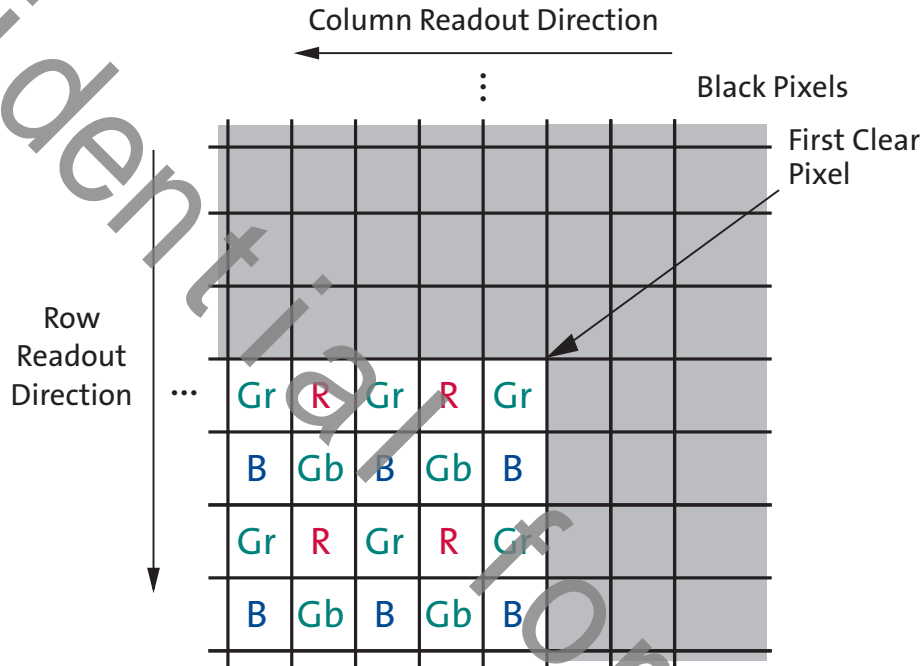
Programmable slew rates are also available to reduce the effect of electromagnetic interference from the output interface.

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

Pixel Array

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

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



Operating Modes

The AR1337 supports MIPI serial output, which can be configured in 2-, 3-, and 4-lanes. There is no parallel data output port. Typical configurations are shown in Figure 3 on page 10. These operating modes are described in “Control of the Signal Interface” on page 24.

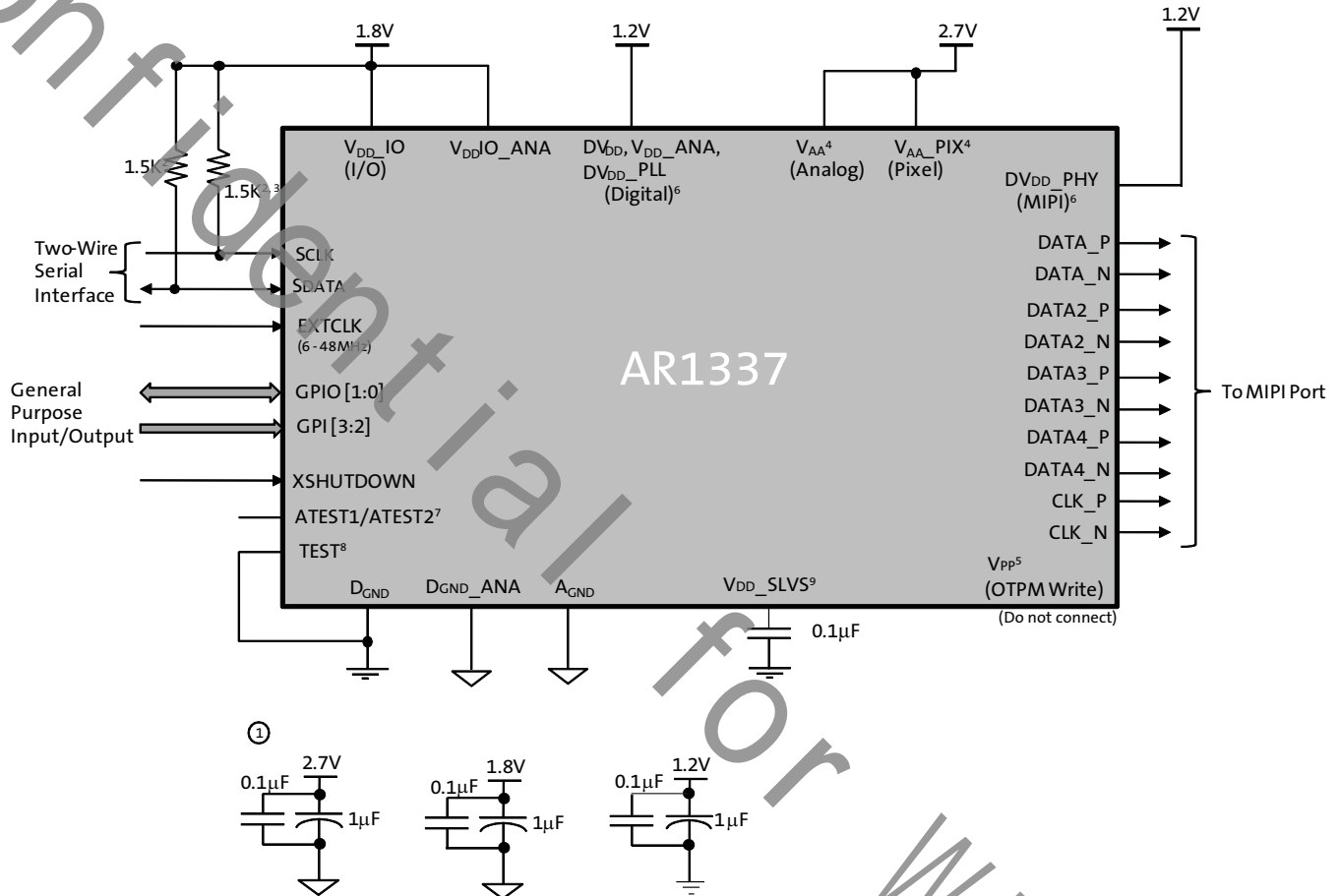
For low-noise operation, the AR1337 requires separate power supplies for analog and digital. Incoming digital and analog ground conductors can be tied together next to the die. Both power supply rails should be decoupled from ground using capacitors as close as possible to the die.

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

Typical Connections

Figure 3 shows the typical AR1337 connections.

Figure 3: Typical Connections



For connectivity above:

- Notes:
1. All power supplies should be adequately decoupled; recommended cap values are:
 - 2.7V: 1.0µF and 0.1µF
 - 1.2V: 1.0uF and 0.1µF
 - 1.8V: 1.0uF and 0.1µF
 2. Resistor value 1.5kΩ is recommended, but may be greater for slower two-wire speed.
 3. This pull-up resistor is not required if the controller drives a valid logic level on SCLK at all times.
 4. VAA and VAA_PIX must be tied together.
 5. Internal charge pump is used for OTPM programming.
 6. Digital and MIPI supply can be tied together.
 7. ATEST1/ATEST2 must be left floating.
 8. TEST pin must be tied to DGND.
 9. VDD_SLVS must be connected to DGND through a bypass cap (0.1uF).

Signal Descriptions

Table 4 provides signal descriptions for AR1337 die. For pad location and aperture information, refer to the AR1337 die data sheet

Table 4: Signal Descriptions

Name	Type	Description
EXTCLK	Input	Master clock input, 6-48 MHz
XSHUTDOWN	Input	Asynchronous active LOW reset. This pin will turn off all power domains and is the lowest power state of the sensor. When asserted, data output stops and when de-asserted all internal registers are restored to their factory default settings.
SCLK	Input	Serial clock for access to control and status registers
GPI[3:2]	Input	General purpose inputs. After reset, these pads are powered-down by default; this means that it is not necessary to bond to these pads. These pads can be configured to provide hardware control of: GPI[2]: SADDR, Trigger signal for slave mode and standby. GPI[3]: 3D daisy chain communication input and all options in GPI[2]. HIDM recommends that unused GPI pins be tied to DGND, but can also be left floating.
GPIO[1:0]	I/O	General purpose inputs and outputs. After reset, these pads are not powered-down since its default use is as output. These pads can be configured to provide hardware control of: GPIO[0]: Flash output (default), all input options in GPI[2]. GPIO[1]: Shutter output (default), 3-D daisy chain communication output and all options in GPI[2]. HIDM recommends that unused GPIO pins be tied to DGND, but they can also be left floating
TEST	Input	TEST must be tied to DGND for normal operation
SDATA	I/O	Serial data from reads and writes to control and status registers
DATA_P	Output	Differential MIPI (sub-LVDS) serial data 1st lane (positive)
DATA_N	Output	Differential MIPI (sub-LVDS) serial data 1st lane (negative)
DATA2_P	Output	Differential MIPI (sub-LVDS) serial data 2nd lane (positive)
DATA2_N	Output	Differential MIPI (sub-LVDS) serial data 2nd lane (negative)
DATA3_P	Output	Differential MIPI (sub-LVDS) serial data 3rd lane (positive)
DATA3_N	Output	Differential MIPI (sub-LVDS) serial data 3rd lane (negative)
DATA4_P	Output	Differential MIPI (sub-LVDS) serial data 4th lane (positive)
DATA4_N	Output	Differential MIPI (sub-LVDS) serial data 4th lane (negative)
CLK_P	Output	Differential MIPI (sub-LVDS) serial clock/strobe (positive).
CLK_N	Output	Differential MIPI (sub-LVDS) serial clock/strobe (negative)
VPP	Supply	Do not connect
DVDD_PLL	Supply	PLL power supply
DVDD_PHY	Supply	Digital PHY power supply. Digital power supply for the serial interface (1.2V)
VAA	Supply	Analog power supply (2.7V)
VAA_PIX	Supply	Analog power supply for the pixel array (2.7V)
AGND	Supply	Analog ground
DVDD, DVDD_ANA	Supply	1.2V digital power supply inputs
VDD_IO, VDDIO_ANA	Supply	I/O power supply (1.8V)
DGND	Supply	Common ground for digital and I/O

Table 4: Signal Descriptions

Name	Type	Description
VDD_SLVS	Supply	The VDD_SLVS is an internally driven 0.4V reference voltage. It must connect to DGND through a bypass cap (0.1uF).

Output Data Format

Serial Pixel Data Interface

The AR1337 serial pixel data interface implements data/clock and data/strobe signaling in accordance with the MIPI specifications. The RAW10, RAW8, and RAW6 image data formats are supported.

Two-Wire Serial Register Interface

The two-wire serial interface bus enables read/write access to control and status registers within the AR1337. The interface protocol uses a master/slave model in which a master controls one or more slave devices. The sensor acts as a slave device. The master generates a clock (SCLK) that is an input to the sensor and is used to synchronize transfers. Data is transferred between the master and the slave on a bidirectional signal (SDATA). SDATA is pulled up to VDD off-chip by a 1.5kΩ resistor. Either the slave or master device can drive SDATA LOW—the interface protocol determines which device is allowed to drive SDATA at any given time.

The protocols described in the two-wire serial interface specification allow the slave device to drive SCLK LOW; the AR1337 uses SCLK as an input only and therefore never drives it LOW.

Protocol

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

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

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

Start Condition

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

Stop Condition

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



AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor Two-Wire Serial Register Interface

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 AR1337 for the MIPI-configured sensor are 0x6C (write address) and 0x6D (read address) in accordance with the MIPI specification. Alternate slave addresses of 0x6E (write address) and 0x6F (read address) can be selected by enabling and asserting the SADDR signal through the GPI pad.

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.

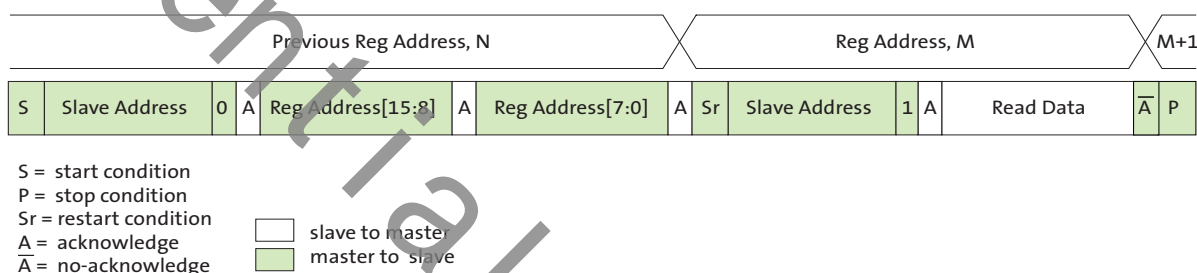
If the request was a READ, the master sends the 8-bit write slave address/data direction byte and 16-bit register address, the same way as with a WRITE request. The master then generates a (re)start condition and the 8-bit read slave address/data direction byte, and clocks out the register data, eight 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 4 on page 14) 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 4 shows how the internal register address maintained by the AR1337 is loaded and incremented as the sequence proceeds.

Figure 4: Single READ from Random Location



Single READ from Current Location

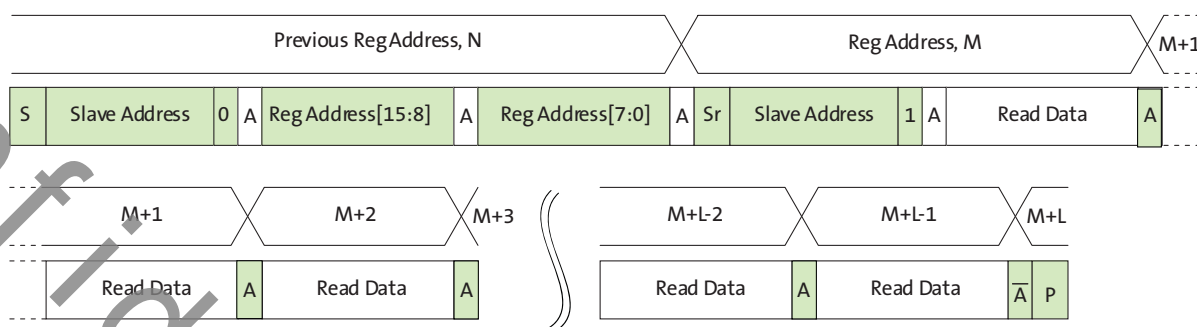
This sequence (Figure 5) performs a read using the current value of the AR1337 internal register address. The master terminates the READ by generating a no-acknowledge bit followed by a stop condition. The figure shows two independent READ sequences.

Figure 5: Single READ from Current Location



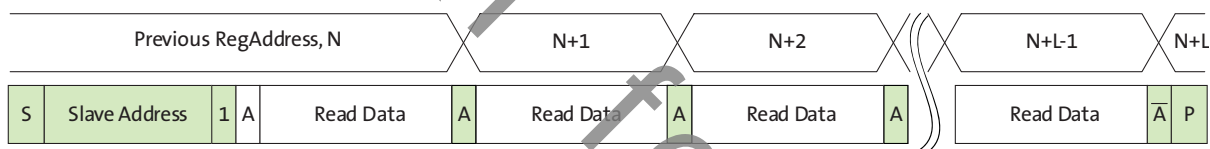
Sequential READ, Start from Random Location

This sequence (Figure 6) starts in the same way as the single READ from random location (Figure 4). 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 6: Sequential READ, Start from Random Location


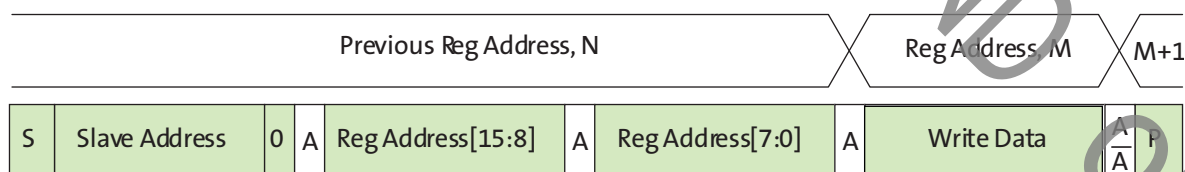
Sequential READ, Start from Current Location

This sequence (Figure 7) starts in the same way as the single READ from current location (Figure 5 on page 14). 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 7: Sequential READ, Start from Current Location


Single WRITE to Random Location

This sequence (Figure 8) 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.

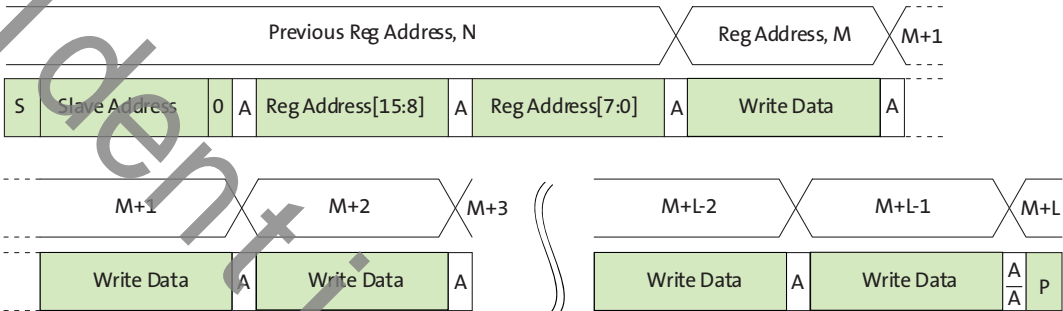
Figure 8: Single WRITE to Random Location




Sequential WRITE, Start at Random Location

This sequence (Figure 9) starts in the same way as the single WRITE to random location (Figure 8). 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 9: Sequential WRITE, Start at Random Location



Registers

The AR1337 provides a 16-bit register address space accessed through a serial interface (“Two-Wire Serial Register Interface” on page 12). Each register location is 8 or 16 bits in size.

The address space is divided into the five major regions shown in Table 5. The remainder of this section describes these registers in detail.

Table 5: Address Space Regions

Address Range	Description
0x0000–0x0FFF	Configuration registers (read-only and read-write dynamic registers)
0x1000–0x1FFF	Parameter limit registers (read-only static registers)
0x2000–0x2FFF	Image statistics registers (none currently defined)
0x3000–0x3FFF	Manufacturer-specific registers (read-only and read-write dynamic registers)
0x4000–0xFFFF	Reserved (undefined)

Register Notation

The underlying mechanism for reading and writing registers provides byte write capability. However, it is convenient to consider some registers as multiple adjacent bytes. The AR1337 uses 8-bit, 16-bit, and 32-bit registers, all implemented as 1 or more bytes at naturally aligned, contiguous locations in the address space.

In this document, registers are described either by address or by name. When registers are described by address, the size of the registers is explicit. For example, R0x3024 is an 8-bit register at address 0x3024, and R0x3000–1 is a 16-bit register at address 0x3000–0x3001. When registers are described by name, the size of the register is implicit. It is necessary to refer to the register table to determine that model_id is a 16-bit register.

Register Aliases

A consequence of the internal architecture of the AR1337 is that some registers are decoded at multiple addresses. Some registers in “configuration space” are also decoded in “manufacturer-specific space.” To provide unique names for all registers, the name of the register within manufacturer-specific register space has a trailing underscore. For example, R0x0000–1 is model_id, and R0x3000–1 is model_id_. The effect of reading or writing a register through any of its aliases is identical.

Bit Fields

Some registers provide control of several different pieces of related functionality, and this makes it necessary to refer to bit fields within registers. As an example of the notation used for this, the least significant 4 bits of the model_id register are referred to as model_id[3:0] or R0x0000–1[3:0].

Bit Field Aliases

In addition to the register aliases described above, some register fields are aliased in multiple places. For example, R0x0100 (mode_select) has only one operational bit, R0x0100[0]. This bit is aliased to R0x301A–B[2]. The effect of reading or writing a bit field through any of its aliases is identical.

Byte Ordering

Registers that occupy more than one byte of address space are shown with the lowest address in the highest-order byte lane to match the byte-ordering on the data bus. For example, the model_id register is R0x0000-1. In the register table the default value is shown as 0x0153. This means that a read from address 0x0000 would return 0x01, and a read from address 0x0001 would return 0x53. When reading this register as two 8-bit transfers on the serial interface, the 0x01 will appear on the serial interface first, followed by the 0x53.

Address Alignment

All register addresses are aligned naturally. Registers that occupy 2 bytes of address space are aligned to even 16-bit addresses, and registers that occupy 4 bytes of address space are aligned to 16-bit addresses that are an integer multiple of 4.

Bit Representation

For clarity, 32-bit hex numbers are shown with an underscore between the upper and lower 16 bits. For example: 0x3000_01AB.

Data Format

Most registers represent an unsigned binary value or set of bit fields. For all other register formats, the format is stated explicitly at the start of the register description. The notation for these formats is shown in Table 6.

Table 6: Data Formats

Name	Description
FIX16	Signed fixed-point, 16-bit number: two's complement number, 8 fractional bits. Examples: 0x0100 = 1.0, 0x8000 = -128, 0xFFFF = -0.0039065
UFIX16	Unsigned fixed-point, 16-bit number: 8.8 format. Examples: 0x0100 = 1.0, 0x280 = 2.5
FLP32	Signed floating-point, 32-bit number: IEEE 754 format. Example: 0x4280_0000 = 64.0

Register Behavior

Registers vary from “read-only,” “read/write,” and “read, write-1-to-clear.”

Double-Buffered Registers

Some sensor settings cannot be changed during frame readout. For example, changing R0x0344-5 (x_addr_start) partway through frame readout would result in inconsistent row lengths within a frame. To avoid this, the AR1337 double-buffers many registers by implementing a “pending” and a “live” version. Reads and writes access the pending register. The live register controls the sensor operation.

The value in the pending register is transferred to a live register at a fixed point in the frame timing, called frame start. Frame start is defined as the point at which the first dark row is read out internally to the sensor. In the register tables the “Frame Sync'd” column shows which registers or register fields are double-buffered in this way.

Using grouped_parameter_hold

Register grouped_parameter_hold (R0x0104) can be used to inhibit transfers from the pending to the live registers. When the AR1337 is in streaming mode, this register should be written to “1” before making changes to any group of registers where a set of changes is required to take effect simultaneously. When this register is written to “0,” all transfers from pending to live registers take place on the next frame start.

An example of the consequences of failing to set this bit follows:

An external auto exposure algorithm might want to change both gain and integration time between two frames. If the next frame starts between these operations, it will have the new gain, but not the new integration time, which would return a frame with the wrong brightness that might lead to a feedback loop with the AE algorithm resulting in flickering.

Bad Frames

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

Many changes to the sensor register settings can cause a bad frame. For example, when line_length_pck (R0x0342–3) is changed, the new register value does not affect sensor behavior until the next frame start. However, the frame that would be read out at that frame start will have been integrated using the old row width, so reading it out using the new row width would result in a frame with an incorrect integration time.

By default, bad frames are masked. If the masked bad frame option is enabled, both LV and FV are inhibited for these frames so that the vertical blanking time between frames is extended by the frame time.

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

N—No. Changing the register value will not produce a bad frame.

Y—Yes. Changing the register value might produce a bad frame.

YM—Yes; but the bad frame will be masked out when mask_corrupted_frames (R0x0105) is set to “1.”

Changes to Integration Time

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 $(n + 1)$. The actual time that rows start integrating using the new integration time is dependent upon the new value of the integration time.
4. When frame $(n + 2)$ 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.



AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor Reading the Sensor Revision Number

Changes to Gain Settings

Usually, when the gain settings are changed, the gain is updated on the next frame start. When the integration time and the gain are changed at the same time, the gain update is held off by one frame so that the first frame output with the new integration time also has the new gain applied. In this case, a new gain should not be set during the extra frame delay. There is an option to turn off the extra frame delay by setting R0x301A–B[14].

Reading the Sensor Revision Number

Follow the steps below to obtain the revision number of the image sensor:

1. Set the register bit field R0x301A[5] = 1.
2. Read the register bit fields R0x31FE[3:0].
3. Convert the binary number to decimal to obtain customer revision.
For example, binary value “0001” = sensor revision 1.

Programming Restrictions

Table 7 shows a list of programming rules that must be adhered to for correct operation of the AR1337. It is recommended that these rules are encoded into the device driver stack—either implicitly or explicitly.

Table 7: Programming Rules

Parameter	Minimum Value	Maximum Value
coarse_integration_time	8 rows	frame_length_lines - coarse_integration_time_max_margin
digital_gain_*	digital_gain_min	digital_gain_max
digital_gain_* is an integer multiple of digital_gain_step_size		
line_length_pck	$\min_line_length_pck$ $((x_addr_end - x_addr_start + x_odd_inc) / xskip) + \min_line_blanking_pck$ $\diamond line_length_pck \geq (x_output_size + constant) \times (vt_pix_clk \text{ period}) / (op_pix_clk \text{ period})$ Note: Constant is 0x20 and 0x68 for MIPI.	max_line_length_pck
frame_length_lines	$\min_frame_length_lines$ $((y_addr_end - y_addr_start + y_odd_inc) / yskip) + \min_frame_blanking_lines$	max_frame_length_lines
x_addr_start (must be an even number)	x_addr_min	x_addr_max
x_addr_end (must be an odd number)	x_addr_start	x_addr_max
$(x_addr_end - x_addr_start + x_odd_inc)$	Must be multiple of 8 for skip1, 16 for skip2, 32 for skip4	must be positive
y_addr_start (must be an even number)	y_addr_min	y_addr_max
y_addr_end (must be an odd number)	y_addr_start	y_addr_max
$(y_addr_end - y_addr_start + y_odd_inc)$	Must be multiple of 8 for skip1, 16 for skip2, 32 for skip4	must be positive
x_odd_inc (must be an odd number)	min_odd_inc	max_odd_inc
y_odd_inc (must be an odd number)	min_odd_inc	max_odd_inc
scale_m	scaler_m_min	scaler_m_max
x_output_size (must be even number – this is enforced in hardware)	320	4224
y_output_size (must be even number – this is enforced in hardware)	2	frame_length_lines

Output Size Restrictions

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

- When `R0x0112 [7:0] = 10` (RAW10/RAW8 data), `x_output_size` must be a multiple of 16 (`x_output_size[3:0] = 0`).

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

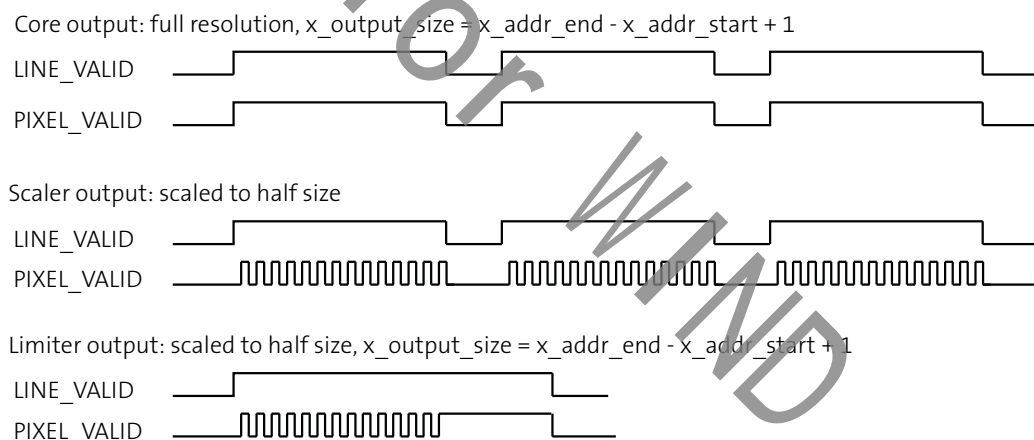
There is an additional restriction that `x_output_size` must be small enough such that the output row time (set by `x_output_size`, the framing and CRC overhead of 12 bytes and the output clock rate) must be less than the row time of the video array (set by `line_length_pck` and the video timing clock rate).

Effect of Scaler on Legal Range of Output Sizes

When the scaler is enabled, it is necessary to adjust the values of `x_output_size` to match the image size generated by the scaler. The AR1337 will operate incorrectly if the `x_output_size` is significantly larger than the output image.

To understand the reason for this, consider the situation where the sensor is operating at full resolution and the scaler is enabled with a scaling factor of 32 (half the number of pixels in each direction). This situation is shown in Figure 10 on page 22.

Figure 10: Effect of Limiter on the Data Path



In Figure 10, three different stages in the data path (see “Timing Specifications” on page 58) are shown. The first stage is the output of the sensor core. The core is running at full resolution and `x_output_size` is set to match the active array size. The LV signal is asserted once per row and remains asserted for N pixel times. The `PIXEL_VALID` signal toggles with the same timing as LV, indicating that all pixels in the row are valid.

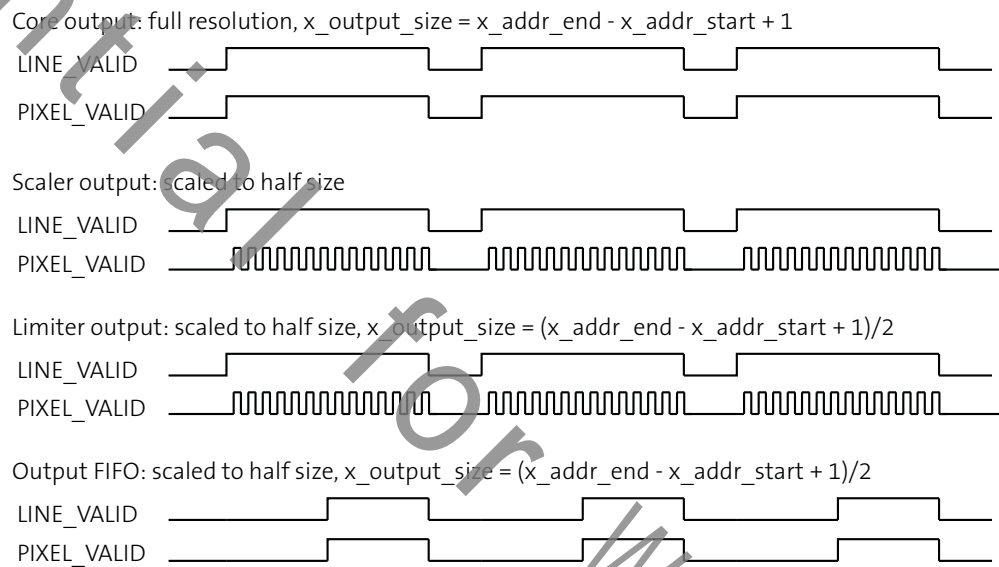
The second stage is the output of the scaler, when the scaler is set to reduce the image size by one-half in x dimension. The effect of the scaler is to combine groups of pixels. Therefore, the row time remains the same, but only half the pixels out of the scaler are valid. This is signaled by transitions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

The third stage is the output of the limiter when the `x_output_size` is still set to match the active array size. Because the scaler has reduced the amount of valid pixel data without reducing the row time, the limiter attempts to pad the row with $(N/2)$ additional pixels. If this has the effect of extending LV across the whole of the horizontal blanking time, the AR1337 will cease to generate output frames.

A correct configuration is shown in Figure 11 on page 23, in addition to showing the `x_output_size` reduced to match the output size of the scaler. In this configuration, the output of the limiter does not extend LV.

Figure 11 on page 23 also shows the effect of the output FIFO, which forms the final stage in the data path. The output FIFO merges the intermittent pixel data back into a contiguous stream. Although not shown in this example, the output FIFO is also capable of operating with an output clock that is at a different frequency from its input clock.

Figure 11: Timing of Data Path



Output Data Timing

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

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

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



Caution If this constraint is not met, the FIFO will either underrun or overrun. FIFO underrun or overrun is a fatal error condition that is signaled through the datapath_status register (R0x306A).

Control of the Signal Interface

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

The AR1337 sensor provides a MIPI serial interface for pixel data.

MIPI Serial Pixel Data Interface

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

- DATA_P
- DATA_N
- DATA2_P
- DATA2_N
- DATA3_P
- DATA3_N
- DATA4_P
- DATA4_N
- CLK_P
- CLK_N

The signal pairs use both single-ended and differential signaling, in accordance with the MIPI specification. The serial pixel data interface is enabled by default at power up and after reset.

The DATA_P, DATA_N, DATA2_P, DATA2_N, CLK_P, and CLK_N pads are set to the Ultra Low Power State (ULPS) if the SMIA serial disable bit is asserted (R0x301A-B[12]=1) or when the sensor is in the hardware standby or soft standby system states.

The data_format (R0x0112-3) register can be programmed to the following data format setting:

- 0x0A0A – Sensor supports RAW10 uncompressed data format. This mode is supported by discarding all but the upper 10 bits of a pixel value.
- 0x0808 – Sensor supports RAW8 uncompressed data format. This mode is supported by discarding all but the upper 8 bits of a pixel value.
- 0x0A08 – Sensor supports RAW8 data format in which an adaptive compression algorithm is used to perform 10-bit to 8-bit compression on the upper 10 bits of each pixel value

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

- 0x0202 – Sensor supports 2-lane MIPI operation
- 0x0203 – Sensor supports 3-lane MIPI operation
- 0x0204 – Sensor supports 4-lane MIPI operation

MIPI Virtual Channel and Data Type Interleave

Introduction

MIPI Virtual Channel and Data Type Interleave support are implemented in AR1337. RAW10 Bayer and PDAF data in a frame can be transmitted separately either thru different Virtual Channel or Data Type. When PDAF data is transmitted in 16-bit format, 10-bit PDAF data is expanded and packed to RAW10 CSI-2 format before transmission to MIPI serial interface that produces 60% increase in pixel data.

In Virtual Channel Interleave mode, `chan_num[1:0]` input to Framer is set to "01" programmed in `virtual_chan[1:0]` register (0x31d6[7:6]) when PDAF data is being transmitted. Likewise, RAW10 Bayer data is transmitted using `virtual_chan[1:0]`="00".

CSI-2 Packet Headers and Payloads

In order to foster interoperability with the number of CSI-2 receivers, the AR1337 sets the contents of transmitted CSI-2 Packet Header Data Type (DT) and Virtual Channel (VC) fields based on the data being transmitted. Table 1 shows data transmitted in CSI-2 Packet Header.

Table 8: CSI-2 Packet Header Control Register Fields

Data	VC	DT	Definition
Bayer	0	0x2B (RAW10)	Sets the contents of the CSI-2 packet header DT fields in long packets with pixel data payloads
PDAF	0/1	0x30 (user_defined)	Sets the contents of the CSI-2 packet header DT fields in long packets with PDAF data payloads

The CSI-2 specification requires one pair of Frame Start (FS) / Frame End (FE) short packets for each Virtual Channel. Output lines produced by combination of Bayer and PDAF data may be transmitted on the MIPI link using either CSI-2 "Data Type" or "Virtual Channel" interleaving. Data Type Interleaving uses different Packet Header Data Type from `data_type_xmit` (within the same Virtual Channel) to discriminate between Bayer and PDAF data, whereas Virtual Channel interleaving transmits a different Virtual Channel.

Data Type vs. Virtual Channel Interleaving

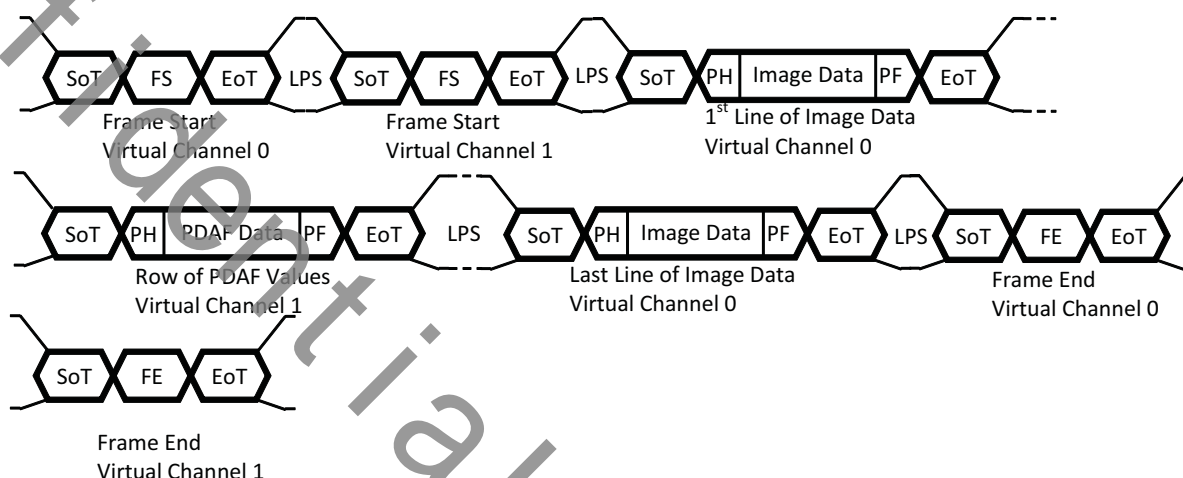
The user may choose between CSI-2 compliant "Data Type" or "Virtual Channel" interleaving using the `virtual_chan` or `data_type_xmit` control registers. If `data_type_xmit` is set differently than `data_type`, Data Type interleaving is enabled, and all output lines are transmitted using a single Virtual Channel. As shown in figure 2, each frame transmitting image/ PDAF data, using Data Type interleaving starts and ends with single FS (frame start) and FE (frame end) short packets, respectively, with VC fields set to "00".

Figures 1 and 2 below are the MIPI CSI packets showing the PDAF pixel data or the PDAF correlations and the normal image data. A key for the different modes and packet types is:

SoT - Start of Transmission	EoT - End of Transmission
PH - Packet Header	PF - Packet Footer
FS - Frame Start	FE - Frame End
LPS - Low Power State	

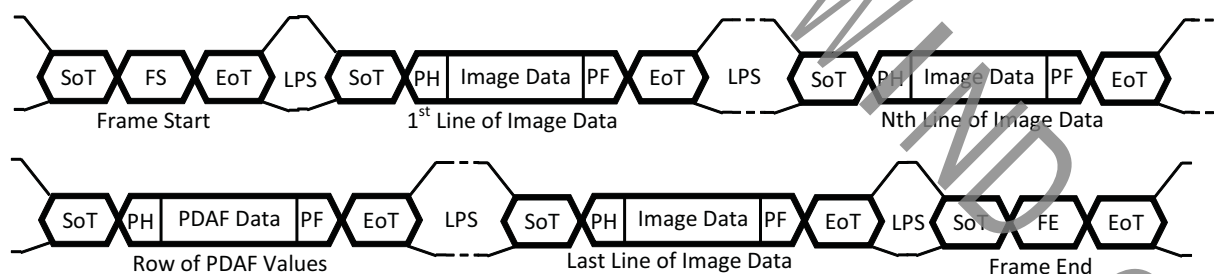
When using multiple Virtual Channels, each Virtual Channel needs a separate frame start and frame end short packets. Both frame start packets will be sent before any image data is sent for the frame and the frame end packets will both be sent after the data for the complete frame has been sent. This will result in a MIPI CSI output stream as follows when different Virtual Channels are used:

Figure 12: Virtual Channel Interleave



PDAF values can be sent to the AP using a different MIPI CSI Data Type instead of using two Virtual Channels. When different Data Types are used, there will only be one frame start and frame end short packets for each frame. When a different Data Type is used for the normal image data stream and the PDAF values or their correlations, the MIPI CSI output stream would be as follows:

Figure 13: Data Type Interleave



The sensor can either send the PDAF raw pixel values or region based correlations that have been computed from the PDAF values. The AP can compute an error from the correlation values so that the AP can determine how reliable the PDAF values are for each zone. The length of each PDAF row may be different than the length of the normal image data, but each row of PDAF values sent over MIPI CSI will be the same length.



AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor System States

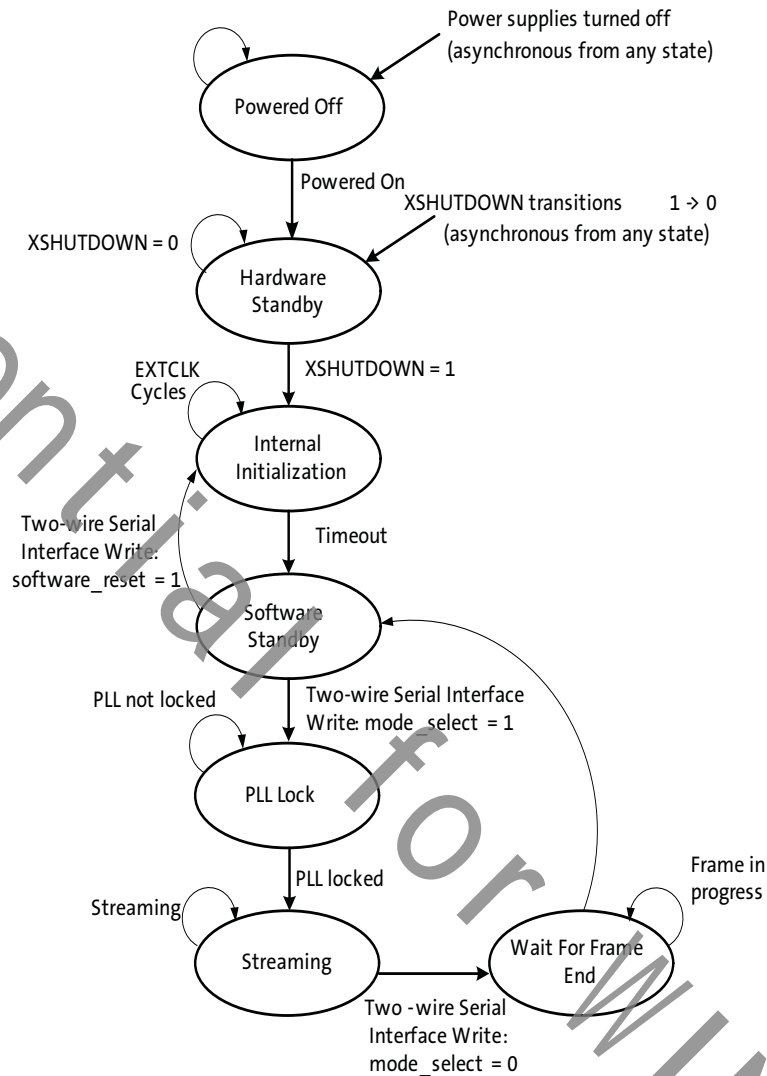
When PDAF pixels are sent to the AP, the values for those pixels are sent twice per frame. The corrected PDAF values (corrected by treating the PDAF pixels as bad pixels) will appear in the normal image data stream and the raw PDAF pixel values will be sent using a different Virtual Channel or Data Type.

The PDAF raw pixel values can be sent as packed RAW10 values, like the image data or as 16-bit unpacked values with the ability to select the endianness of the data. For unpacked 16-bit values bit zero of the pixel data, the least significant bit, will be in bit 0 of the 16-bit word and the top 6 bits of the 16-bit values will be 0. If PDAF correlations are being sent to the AP instead of PDAF raw values, the data can be sent as 16-bit values with selectable endianness. The MIPI CSI Data Type field that will be used in the packet header for the PDAF values or PDAF correlations is selected by programming the desired values on the sensor.

System States

The system states of the AR1337 are represented as a state diagram in Figure 14 and described in subsequent sections. The effect of XSHUTDOWN on the system state and the configuration of the PLL in the different states are shown in Table 9 on page 28.

The sensor's operation is broken down into three separate states: hardware standby, software standby, and streaming. The transition between these states might take a certain amount of clock cycles as outlined in Table 9 on page 28.

Figure 14: AR1337 System States

Table 9: XSHUTDOWN and PLL in System States

State	XSHUTDOWN	PLL
Powered off	x	VCO powered down
Hardware standby	0	
Internal initialization	1	
Software standby		VCO powering up and locking, PLL output bypassed
PLL Lock		
Streaming		VCO running, PLL output active
Wait for frame end		

Soft Reset Sequence

The AR1337 can be reset under software control by writing “1” to software_reset (R0x0103). A software reset asynchronously resets the sensor, truncating any frame that is in progress. The sensor starts the internal initialization sequence, while the PLL and analog blocks are turned off. At this point, the behavior is exactly the same as for the power-on reset sequence.

Signal State During Reset

Table 10 shows the state of the signal interface during hardware standby (XSHUTDOWN asserted) and the default state during software standby (after exit from hardware standby and before any registers within the sensor have been changed from their default power-up values).

Table 10: Signal State During Reset

Pad Name	Pad Type	Hardware Standby	Software Standby
EXTCLK	Input	Enabled. Must be driven to a valid logic level.	
XSHUTDOWN	Input	Enabled. Must be driven to a valid logic level.	
SCLK	Input	Enabled. Must be pulled up or driven to a valid logic level.	
SDATA	I/O	Enabled as an input. Must be pulled up or driven to a valid logic level.	
GPIO[1:0]	I/O	High-Z.	Logic 0.
DATA_P	Output	MIPI: Ultra Low-Power State (ULPS), represented as an LP-00 state on the wire (both wires at 0V).	
DATA_N	Output		
DATA2_P	Output		
DATA2_N	Output		
DATA3_P	Output		
DATA3_N	Output		
DATA4_P	Output		
DATA4_N	Output		
CLK_P	Output		
CLK_N	Output		
GPI[3:2]	Input	Powered down. Can be left disconnected/floating.	
TEST	Input	Must be driven to 0 for normal operation.	

General Purpose Input and Output

The AR1337 provides four general purpose input and output pads, as listed in Table 11. GPIO[1:0] are defined as bidirectional (input and output), GPI[3:2] input only.

After power on reset, GPIO[0] is assigned as an output of Flash signal and GPIO[1] is assigned as an output of Shutter signal. Other two GPI are powered down if not used.

Table 11: General Purpose Input and Output Pad Functions

PIN Names	Functions
GPIO[0]	General Input and one Output a. (Default Output) Flash b. (Input) All options in GPI2
GPIO[1]	General Input and two Output functions a. (Default Output) Shutter b. (Output) 3-D daisy chain communication output c. (Input) all options in GPI2
GPI[2]	General Input a. SADDR, second I ² C device address b. Trigger signal for Slave Mode c. Standby
GPI[3]	General Input a. 3-D daisy chain communication input b. All options in GPI2

Streaming/Standby Control

The AR1337 can be switched between its soft standby and streaming states under pin or register control, as shown in Table 12. Selection of a pin to use for the STANDBY function is described in “General Purpose Input and Output” on page 30. The state diagram for transitions between soft standby and streaming states is shown in Figure 14 on page 28.

Table 12: Streaming/STANDBY

STANDBY	Streaming R0x301A-B[2]	Description
Disabled	0	Soft standby
Disabled	1	Streaming
X	0	Soft standby
0	1	Streaming
1	X	Soft standby

Clocking

Default setup provides a physical 220 MHz internal clock for an external input clock of 25 MHz. Maximum allowed for vt_pix_clk is 220 MHz and the design VCO range is between 320 – 1200 MHz.

The sensor contains two phase-locked loops (PLL). Each PLL contains a prescaler to divide the input clock applied on EXTCLK, a VCO to multiply the prescaler output, and a set of dividers to generate the output clocks.

Note that the data rate of the pixel domain cannot be less than the data rate of the output domain or there will be underflow errors.

Figure 15 shows the different clocks and the names of the registers that contain or are used to control their values.

Figure 15: AR1337 Profile 1/2 Clocking Structure

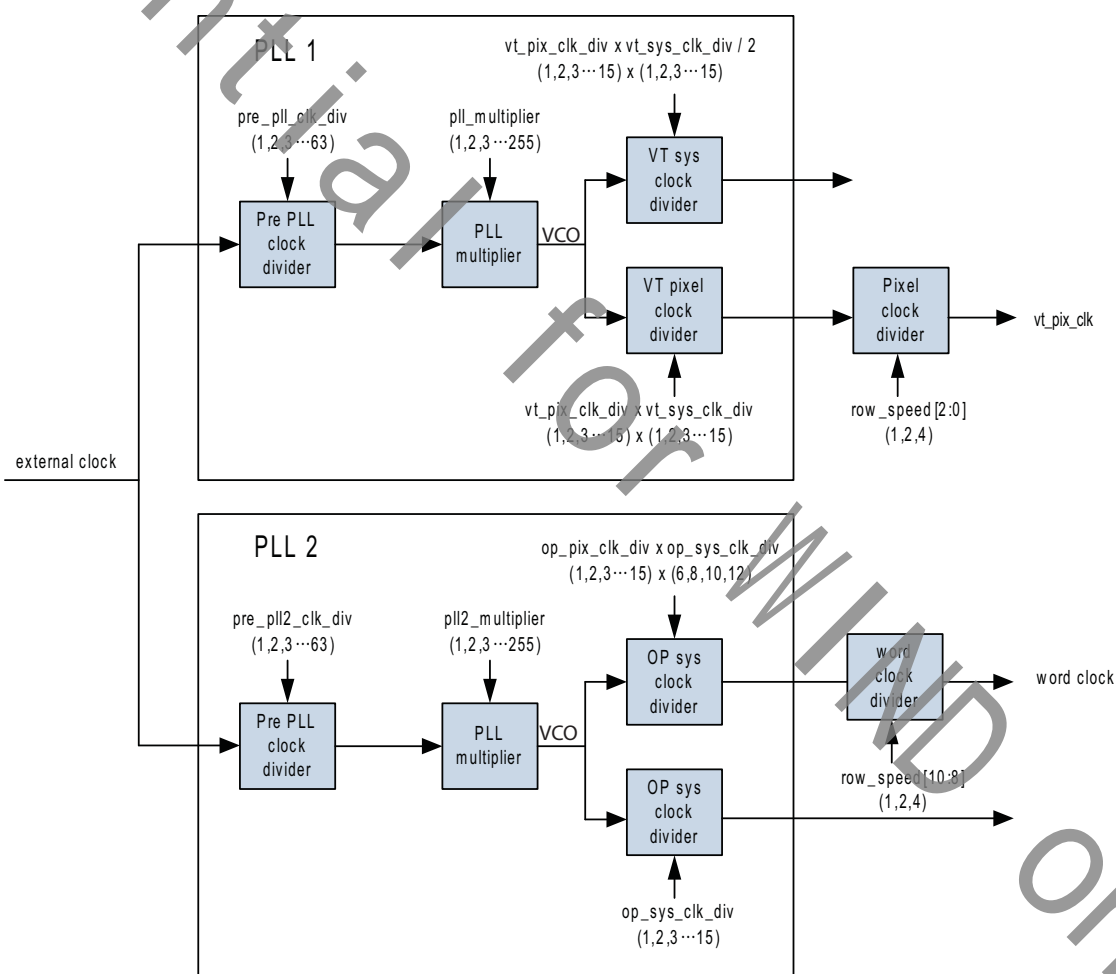


Figure 15 shows the different clocks and the names of the registers that contain or are used to control their values. The figure shows the default setting for each divider/multiplier control register and the range of legal values for each divider/multiplier control register.

As per the diagram, the clock frequencies can be calculated as follows:

- pixel_clock (This is the main clock of the sensor). Two data values are processed per clock cycle

$$clk_pixel_freq_mhz = \frac{ext_clk_freq_mhz \times pll_multiplier1}{pre_pll_clk_div1 \times vt_sys_clk_div \times vt_pix_clk_div \times row_speed[2:0]} \quad (EQ\ 1)$$

- Word_clock (This is the output domain clock). Two data values are processed per clock cycle:

$$clk_word_freq_mhz = \frac{ext_clk_freq_mhz \times pll_multiplier2}{pre_pll_clk_div2 \times op_sys_clk_div \times op_pix_clk_div \times row_speed[10:8]} \quad (EQ\ 2)$$

- Bit_clock (This is the serial domain clock), depending on the number of lanes the data rate will vary

$$clk_bit_freq_mhz = \frac{ext_clk_freq_mhz \times pll_multiplier2}{pre_pll_clk_div2 \times op_sys_clk_div} \quad (EQ\ 3)$$

Note: In Profile 0, RAW10 data format is required. As a result, op_pix_clk_div must be set to 10. Also, due to the inherent design of the AR1337 sensor, vt_pix_clk_div must be set to 5 for profile 0 mode.)

PLL Clocking

The PLL divisors should be programmed while the AR1337 is in the software standby state. After programming the divisors, it is necessary to wait for the VCO lock time before enabling the PLL. The PLLs are enabled by entering the streaming state.

The effect of programming the PLL divisors while the AR1337 is in the streaming state is undefined.

Influence of ccp_data_format

R0x0112-3 (ccp_data_format) controls whether the pixel data interface will generate 10 or 8 bits or 6 bits per pixel.

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

Features

Shading Correction (SC)

Lenses tend to produce images whose brightness is significantly attenuated near the edges. There are also other factors causing fixed pattern signal gradients in images captured by image sensors. The cumulative result of all these factors is known as image shading. The AR1337 has an embedded shading correction module that can be programmed to counter the shading effects on each individual Red, GreenB, GreenR and Blue color signal.

The Correction Function

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

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

$$P_{corrected}(row, col) = P_{sensor}(row, col) * f(row, col) \quad (EQ\ 4)$$

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

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

One-Time Programmable Memory (OTPM)

The AR1337 features 6.8 kbits of one-time programmable memory (OTPM) for storing shading correction coefficients, individual module, and sensor specific information. The user may program which set to be used. Additional bits are used by the error detection and correction scheme. OTPM can be accessed through two-wire serial interface. The AR1337 uses the auto mode for fast OTPM programming and read operations.

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

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

Image Acquisition Mode

The AR1337 supports an image acquisition mode, the electronic rolling shutter (ERS) mode. This is the normal mode of operation. When the AR1337 is streaming, it generates frames at a fixed rate, and each frame is integrated (exposed) using the ERS. When the ERS is in use, timing and control logic within the sensor sequences through the rows of the array, resetting and then reading each row in turn. In the time interval between resetting a row and subsequently reading that row, the pixels in the row integrate incident light. The integration (exposure) time is controlled by varying the time between row reset and row readout. For each row in a frame, the time between row reset and row readout is fixed, leading to a uniform integration time across the frame. When the integration time is changed (by using the two-wire serial interface to change register settings), the timing and control logic controls the transition from old to new integration time in such a way that the stream of output frames from the AR1337 switches cleanly from the old integration time to the new while only generating frames with uniform integration. See “Changes to Integration Time” on page 19.

Window Control

The sequencing of the pixel array is controlled by the `x_addr_start`, `y_addr_start`, `x_addr_end`, and `y_addr_end` registers. The output image size is controlled by the `x_output_size` and `y_output_size` registers.

Pixel Border

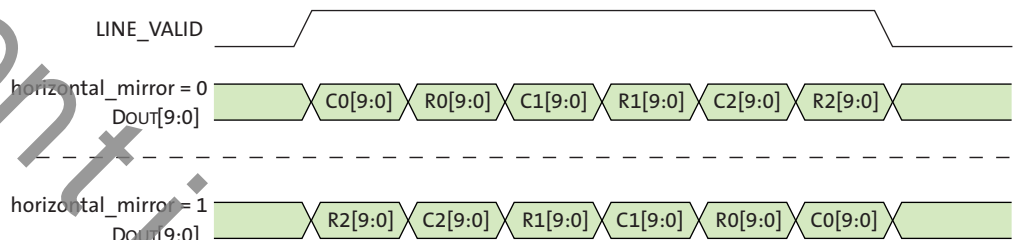
The default settings of the sensor provide a 4208H x 3120V image. A border of up to 8 pixels (4 in subsampling) on each edge can be enabled by reprogramming the `x_addr_start`, `y_addr_start`, `x_addr_end`, `y_addr_end`, `x_output_size`, and `y_output_size` registers accordingly.

Readout Modes

Horizontal Mirror

When the horizontal_mirror bit is set in the image_orientation register, the order of pixel readout within a row is reversed, so that readout starts from x_addr_end and ends at x_addr_start. Figure 16 on page 35 shows a sequence of 6 pixels being read out with horizontal_mirror = 0 and horizontal_mirror = 1. Changing horizontal_mirror causes the Bayer order of the output image to change; the new Bayer order is reflected in the value of the pixel_order register.

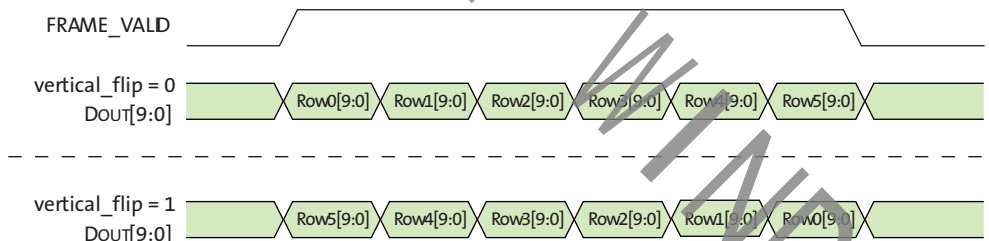
Figure 16: Effect of horizontal_mirror on Readout Order



Vertical Flip

When the vertical_flip bit is set in the image_orientation register, the order in which pixel rows are read out is reversed, so that row readout starts from y_addr_end and ends at y_addr_start. Figure 17 shows a sequence of 6 rows being read out with vertical_flip = 0 and vertical_flip = 1. Changing vertical_flip causes the Bayer order of the output image to change; the new Bayer order is reflected in the value of the pixel_order register.

Figure 17: Effect of vertical_flip on Readout Order





Subsampling

The AR1337 supports subsampling with or without the PDAF function. Subsampling reduces the amount of data processed by the analog signal chain in the AR1337 thereby allowing the frame rate to be increased. Subsampling is enabled by setting y_odd_inc. Values of 1, 3, 5, and 7 can be supported.

Table 13: Definitions for Programming Rules

Name	Definition
yskip	yskip = 1 if y_odd_inc = 1 yskip = 2 if y_odd_inc = 3 yskip = 3 if y_odd_inc = 5 yskip = 4 if y_odd_inc = 7

Figure 18: Pixel Readout (No Subsampling)

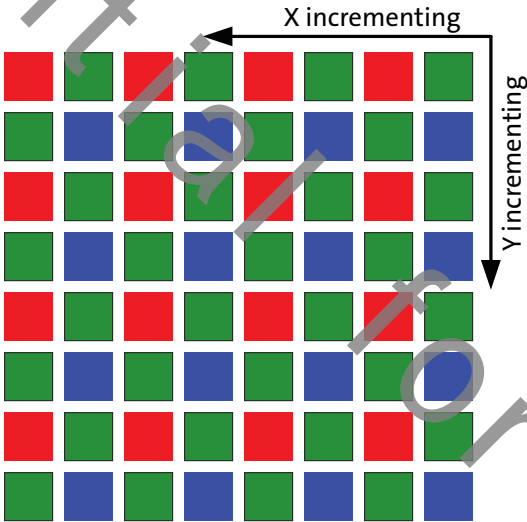


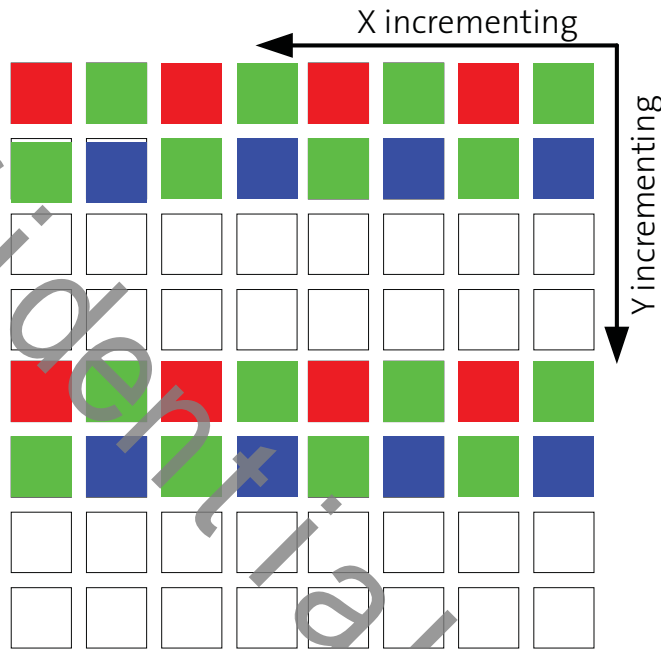
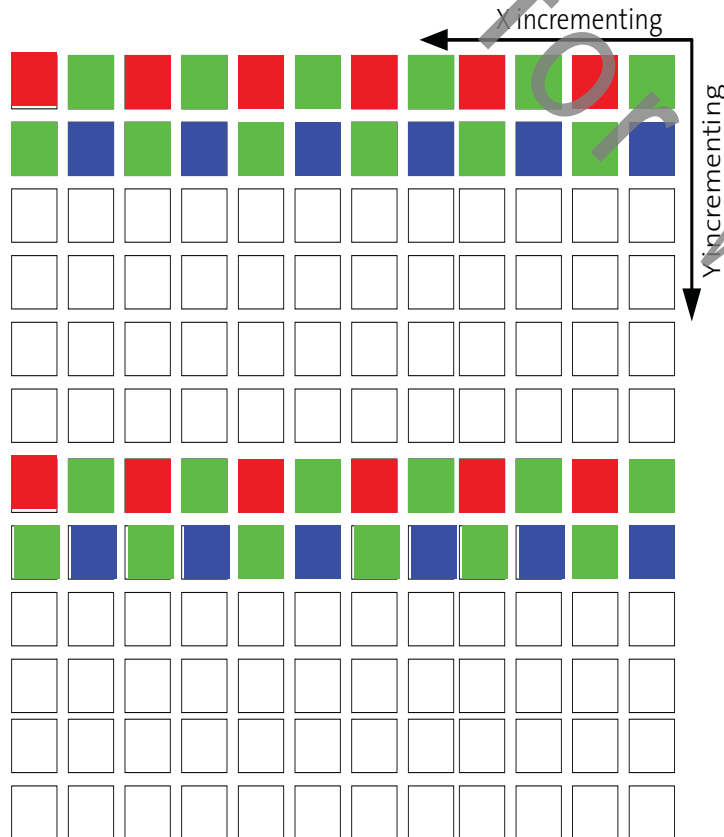
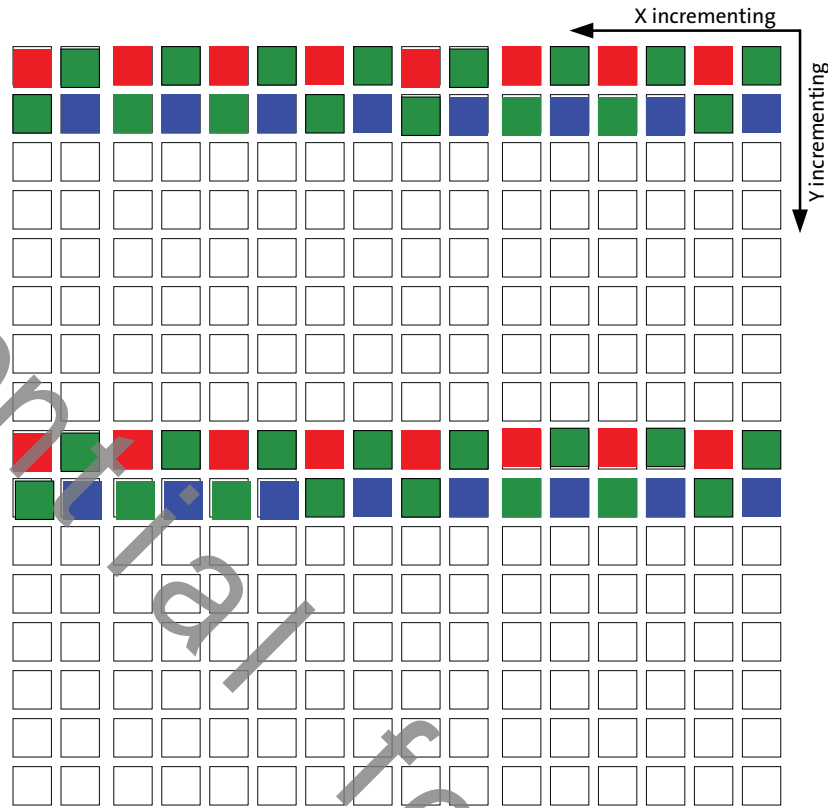
Figure 19: Pixel Readout ($x_odd_inc = 1, y_odd_inc = 3$)

Figure 20: Pixel Readout ($x_odd_inc = 1, y_odd_inc = 5$)


Figure 21: Pixel Readout ($x_odd_inc = 1, y_odd_inc = 7$)

Programming Restrictions When Subsampling

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

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

- Set `x_odd_inc = 1`
- `x_skip_factor = (x_odd_inc + 1) / 2`
- `y_skip_factor = (y_odd_inc + 1) / 2`
- `x_addr_start` should be a multiple of `x_skip_factor × 4`
- `(x_addr_end - x_addr_start + x_odd_inc)` should be a multiple of `x_skip_factor × 4`
- `(y_addr_end - y_addr_start + y_odd_inc)` should be a multiple of `y_skip_factor × 4`

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

- $\text{columns/rows} = (\text{addr_end} - \text{addr_start} + \text{odd_inc}) / \text{skip_factor}$

Table 14 shows the row or column address sequencing for normal and subsampled readout. In the 2X skip case, there are two possible subsampling sequences (because the subsampling sequence only reads half of the pixels) depending upon the alignment of the start address. Similarly, there will be four possible subsampling sequences in the 4X skip case (though only the first two are shown in Table 14).

Table 14: Row Address Sequencing During Subsampling

odd_inc=1(Normal)	odd_inc=3(2X Skip)	odd_inc=5(3X Skip)	odd_inc=7(4X Skip)
Start=0	Start=0	Start=0	Start=0
0	0	0	0
1	1	1	1
2			
3			
4	4		
5	5		
6		6	
7		7	
8	8		8
9	9		9
10			
11			
12	12	12	
13	13	13	
14			
15			
16	16		16
17	17		17
18		18	

Table 14: Row Address Sequencing During Subsampling (continued)

odd_inc=1(Normal)	odd_inc=3(2X Skip)	odd_inc=5(3X Skip)	odd_inc=7(4X Skip)
Start=0	Start=0	Start=0	Start=0
19		19	
20	20		
21	21		
22			
23			
24	24	24	24
25	25	25	25
26			
27			
28	28		
29	29		
30		30	
31		31	
32	32		32
33	33		33
34			
35			
36	36	36	
37	37	37	
38			
39			
40	40		40
41	41		41
42		42	
43		43	
44	44		
45	45		
46			
47			

Scaler

Scaling is a “zoom out” operation to reduce the size of the output image while covering the same extent as the original image. Each scaled output pixel is calculated by taking a weighted average of a group input pixels which is composed of neighboring pixels. The input and output of the scaler is in Bayer format.

When compared to skipping, scaling is advantageous because it uses all pixel values to calculate the output image which helps avoid aliasing.

The AR1337 sensor is capable of horizontal scaling only.

The scaling factor, programmable in 1/16 steps, is used for horizontal scaling.

The scale factor is determined by:

- m, which is adjustable with register R0x0404

Legal values for m are 16 through 128, giving the user the ability to scale from 1:1 (m=16) to 1:8 (m=128).

Frame Rate Control

The formulas for calculating the frame rate of the AR1337 are shown below.

The line length is programmed directly in pixel clock periods through register `line_length_pck`. For a specific window size, the minimum line length can be found from in Equation 5:

$$\text{minimum line_length_pck} = \left(\frac{x_addr_end - x_addr_start + 1}{\text{subsampling factor}} + \text{min_line_blanking_pck} \right) \quad (\text{EQ } 5)$$

Note that `line_length_pck` also needs to meet the minimum line length requirement set in register `min_line_length_pck`. The row time can either be limited by the time it takes to sample and reset the pixel array for each row, or by the time it takes to sample and read out a row. Values for `min_line_blanking_pck` are provided in “Minimum Row Time” on page 42.

The frame length is programmed directly in number of lines in the register `frame_line_length`. For a specific window size, the minimum frame length can be found in Equation 6:

$$\text{minimum frame_length_lines} = \left(\frac{y_addr_end - y_addr_start + 1}{\text{subsampling factor}} + \text{min_frame_blanking_lines} \right) \quad (\text{EQ } 6)$$

The frame rate can be calculated from these variables and the pixel clock speed as shown in Equation 7:

$$\text{frame rate} = \frac{2 \times \text{vt_pixel_clock_mhz} \times 1 \times 10^6}{\text{line_length_pck} \times \text{frame_length_lines}} \quad (\text{EQ } 7)$$

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

Minimum Row Time

The minimum row time and blanking values are shown in Table 15.

Table 15: Minimum Row Time and Blanking Numbers

	No Column Binning	Column Binning
min_line_blanking_pck	212	132
min_line_length_pck	4640	4640

In addition, enough time must be given to the output FIFO so it can output all data at the set frequency within one row time. There are therefore three checks that must all be met when programming line_length_pck:

- line_length_pck ≥ min_line_length_pck in Table 15.
- line_length_pck > (x_addr_end - x_addr_start + x_odd_inc) / subsampling_factor + min_line_blanking_pck
- line_length_pck > (x_output_size / #_of_mipi_lane + 0x68) × (pix_clock_freq_mhz / word_clock_freq_mhz) × 2

Minimum Frame Time

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

Table 16: Minimum Frame Time and Blanking Numbers

	No Row Binning	Row Binning
min_frame_blanking_lines	0xE	0xC

Integration Time

The integration (exposure) time of the AR1337 is controlled by the coarse_integration_time registers.

The limits for the coarse integration time are defined by:

$$coarse_integration_time_min \leq coarse_integration_time \quad (EQ\ 8)$$

The actual integration time is given by:

$$integration_time = \frac{coarse_integration_time \times line_length_pck}{2 \times vt_pixel_clock_freq_mhz \times 10^6} \quad (EQ\ 9)$$

It is required that:

$$coarse_integration_time \leq (frame_length_lines - coarse_integration_time_max_margin) \quad (EQ\ 10)$$

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

Flash Timing Control

The AR1337 supports both xenon and LED flash timing through GPIO[0] (default output). The timing of the FLASH signal with the default settings is shown in Figure 22 (Xenon) and Figure 23 (LED). The flash and flash_count registers allow the timing of the flash to be changed. The flash can be programmed to fire only once, delayed by a few frames when asserted, and (for xenon flash) the flash duration can be programmed.

Enabling the LED flash will cause one bad frame, where several of the rows only have the flash on for part of their integration time. This can be avoided either by first enabling mask bad frames ($R0x301A[9] = 1$) before the enabling the flash or by forcing a restart ($R0x301A[1] = 1$) immediately after enabling the flash; the first bad frame will then be masked out, as shown in Figure 23. Read-only bit flash[14] is set during frames that are correctly integrated; the state of this bit is shown in Figures 22 and Figure 23.

Figure 22: Xenon Flash Enabled

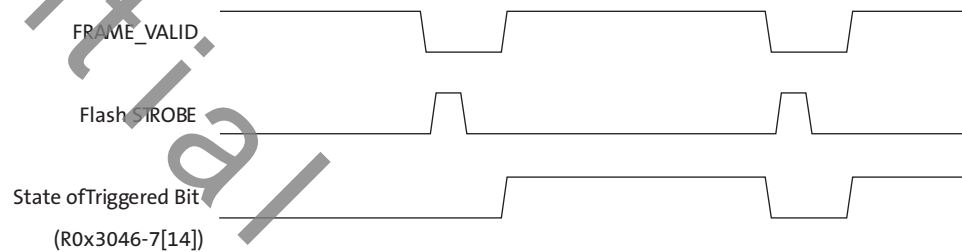
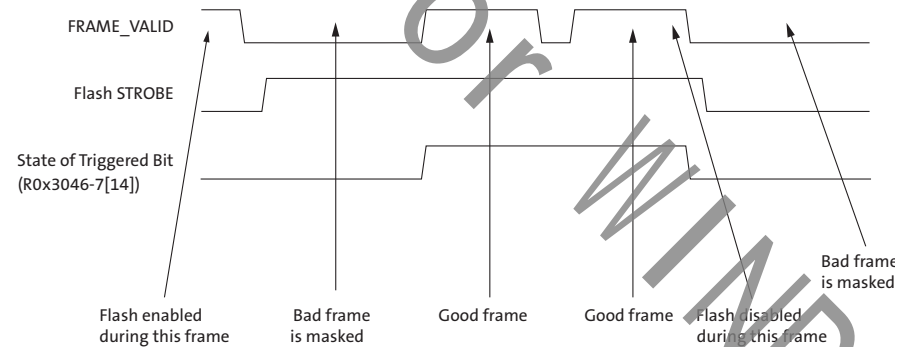


Figure 23: LED Flash Enabled



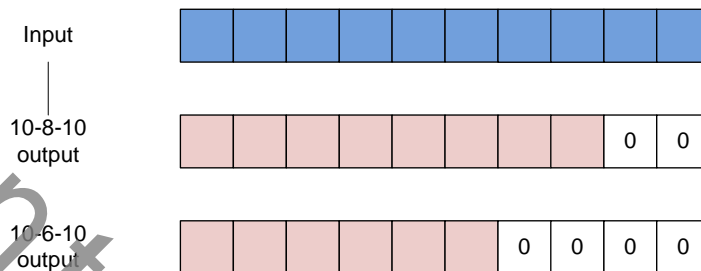
Note: An option to invert the flash output signal through $R0x3046[7]$ is also available.

Additional Features

Bit-depth Compression

AR1337 supports SMIA DPCM with 10-8-10 and 10-6-10 compression.

The output compressed data is MSB-aligned. When compression is enabled, '0's are padded to the LSB side.



The registers listed in Table 17 are related to compression.

Table 17: Compression Registers

Address	Name	Register	Description
0x0112	data_format	RW	[3:0]: The bit-width of the compressed pixel data [11:8]: The bit-width of the uncompressed data
0x31AE	serial format	RW	[9:8]: serial interface type 2: MIPI 3: Reserved [2:0]: serial data lanes

3D Support

This function uses GPI (as an input) and GPIO1 (as an output). The input is sampled on rising CLK (EXTCLK) and the output changes on rising CLK (EXTCLK). In order to use this function, GPIO1 must be enabled as an output.

The CHAIN_CONTROL register should only be written when the sensor is not in streaming mode; the effect of writing to this register when the sensor is in streaming mode is UNDEFINED.

When chain_enable=1 and master=0, transitions on the SADDR input are propagated synchronously to the GPIO1 output.

When chain_enable=1 and master=1, GPIO1 generates 'events' under certain circumstances. An event is a 'start' bit followed by a 4-bit code.

When chain_enable=1 and sync_enable=1 and master=0, and the sensor core enters streaming mode (bit 2 in reset_register is changed to 1), the actual entry to streaming mode is delayed until the sensor receives an event.



Integration Time for Interlaced HDR Readout

Tint1 (integration time 1) and Tint2 (integration time 2)

The limits for the coarse integration time are defined by:

$$coarse_integration_time_min \leq coarse_integration_time \leq (frame_length_lines - coarse_integration_time_max_margin) \quad (EQ 11)$$

$$coarse_integration_time2_min \leq coarse_integration_time2 \leq (frame_length_lines - coarse_integration_time2_max_margin) \quad (EQ 12)$$

The actual integration time is given by:

$$integration_time = \frac{coarse_integration_time \times line_length_pck}{2 \times vt_pixel_clock_freq_mhz \times 10^6} \quad (EQ 13)$$

$$integration_time2 = \frac{coarse_integration_time2 \times line_length_pck}{2 \times vt_pixel_clock_freq_mhz \times 10^6} \quad (EQ 14)$$

If this limit is broken, the frame time will automatically be extended to $(coarse_integration_time + coarse_integration_time_max_margin)$ to accommodate the larger integration time.

The ratio between even and odd rows is typically adjusted to 1x, 2x, 4x, and 8x.

Gain

AR1337 supports both analog and digital gain.

Analog Gain

Analog gain is provided by adjusting the column-parallel ADC reference voltage. The fine gain is implemented through adjusting the ADC reference voltage with fine steps. Global gain register (R0x305E) sets the analog gain. Analog coarse gain is set by setting bits R0x305E[6:4] while analog fine gain is set by setting bits R0x305E[3:0]. The maximum analog gain is 7.75x. Table 18 shows the recommended gain settings:

Table 18: Recommended Gain Settings

Gain Codes R0x305E[15:0]	Analog Gain	Digital Gain	Total Gain
0x2010	1	1	1
0x2014	1.25	1	1.25
0x2018	1.5	1	1.5
0x201C	1.75	1	1.75
0x2020	2	1	2
0x2022	2.25	1	2.25
0x2024	2.5	1	2.5
0x2026	2.75	1	2.75
0x2028	3	1	3
0x202A	3.25	1	3.25
0x202C	3.5	1	3.5
0x202E	3.75	1	3.75
0x2030	4	1	4
0x2031	4.25	1	4.25
0x2032	4.5	1	4.5
0x2033	4.75	1	4.75
0x2034	5	1	5
0x2035	5.25	1	5.25
0x2036	5.5	1	5.5
0x2037	5.75	1	5.75
0x2038	6	1	6
0x2039	6.25	1	6.25
0x203A	6.5	1	6.5
0x203B	6.75	1	6.75
0x203C	7	1	7
0x203D	7.25	1	7.25
0x203E	7.5	1	7.5
0x203F	7.75	1	7.75
0x213F	7.75	1.03125	8
0x223F	7.75	1.0625	8.25
0x233F	7.75	1.09375	8.5
0x243F	7.75	1.125	8.75
0x253F	7.75	1.15625	9
0x263F	7.75	1.1875	9.25

Table 18: Recommended Gain Settings (continued)

Gain Codes R0x305E[15:0]	Analog Gain	Digital Gain	Total Gain
0x273F	7.75	1.21875	9.5
0x28BF	7.75	1.265625	9.75
0x29BF	7.75	1.296875	10
0x2ABF	7.75	1.328125	10.25
0x2BBF	7.75	1.359375	10.5
0x2CBF	7.75	1.390625	10.75
0x2DBF	7.75	1.421875	11
0x2EBF	7.75	1.453125	11.25
0x2FBF	7.75	1.484375	11.5
0x30BF	7.75	1.515625	11.75
0x31BF	7.75	1.546875	12
0x32BF	7.75	1.578125	12.25
0x33BF	7.75	1.609375	12.5
0x34BF	7.75	1.640625	12.75
0x35BF	7.75	1.671875	13
0x36BF	7.75	1.703125	13.25
0x37BF	7.75	1.734375	13.5
0x393F	7.75	1.78125	13.75
0x3A3F	7.75	1.8125	14
0x3B3F	7.75	1.84375	14.25
0x3C3F	7.75	1.875	14.5
0x3D3F	7.75	1.90625	14.75
0x3E3F	7.75	1.9375	15
0x3F3F	7.75	1.96875	15.25
0x403F	7.75	2	15.5
0x413F	7.75	2.03125	15.75
0x423F	7.75	2.0625	16
0x433F	7.75	2.09375	16.25
0x443F	7.75	2.125	16.5
0x453F	7.75	2.15625	16.75
0x463F	7.75	2.1875	17
0x473F	7.75	2.21875	17.25
0x48BF	7.75	2.265625	17.5
0x49BF	7.75	2.296875	17.75
0x4ABF	7.75	2.328125	18
0x4BBF	7.75	2.359375	18.25
0x4CBF	7.75	2.390625	18.5
0x4DBF	7.75	2.421875	18.75
0x4EBF	7.75	2.453125	19
0x4FBF	7.75	2.484375	19.25
0x50BF	7.75	2.515625	19.5
0x51BF	7.75	2.546875	19.75
0x52BF	7.75	2.578125	20
0x53BF	7.75	2.609375	20.25
0x54BF	7.75	2.640625	20.5

Table 18: Recommended Gain Settings (continued)

Gain Codes R0x305E[15:0]	Analog Gain	Digital Gain	Total Gain
0x55BF	7.75	2.671875	20.75
0x56BF	7.75	2.703125	21
0x57BF	7.75	2.734375	21.25
0x593F	7.75	2.78125	21.5
0x5A3F	7.75	2.8125	21.75
0x5B3F	7.75	2.84375	22
0x5C3F	7.75	2.875	22.25
0x5D3F	7.75	2.90625	22.5
0x5E5F	7.75	2.9375	22.75
0x5F3F	7.75	2.96875	23
0x603F	7.75	3	23.25
0x613F	7.75	3.03125	23.5
0x623F	7.75	3.0625	23.75
0x633F	7.75	3.09375	24

Table 19: Reference of ISO-speed-based Gain Settings

ISO-speed	Total Gain Value	Register 0x305E Value
100	1.3125x	0x2015
200	2.625x	0x2025
400	5.25x	0x2035
800	10.53x	0x2BBF
1,600	21.05x	0x573F
3,200	42.1x	0xAE3F

Note: * ISO-speed calculations based on 5100K light source and “AP2” IR-cut filter.

Digital Gain

Digital gain provides both per-color and fine (sub 1x) gain. The analog and digital gains are multiplicative to give the total gain. Digital gain is set by setting bits R0x305E[15:7] to set global gain or by individually setting digital color gain R0x3056-C[15:7] where these 11 bits are designed in 3p6 format, that is, 3 MSB provide gain up to 7x in step of 1x while 6 LSB provide sub-1x gain with a step size of 1/64. This sub-1x gain provides the fine gain control for the sensor.

Total Gain

Maximum total gain is 7.75x (analog) and 7.98x (digital). The total gain equation can be formulated as:

$$Total\ gain = (2^{(Analog_coarse_gain-1)}) \times (16 + analog_fine_gain) / 16 \times (digital_gain / 64) \quad (EQ\ 15)$$

Sensor Core Digital Data Path

Test Patterns

The AR1337 supports a number of test patterns to facilitate system debug. Test patterns are enabled using `test_pattern_mode` (R0x0600–1). The test patterns are listed in Table 20.

Table 20: Test Patterns

test_pattern_mode	Description
0	Normal operation: no test pattern
1	Solid color
2	100% color bars
3	Fade-to-gray color bars
256	Walking 1s (10-bit)
257	Walking 1s (8-bit)

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

For all of the test patterns, the AR1337 registers must be set appropriately to control the frame rate and output timing. This includes:

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

Effect of Data Path Processing on Test Patterns

Test patterns are introduced early in the pixel data path. As a result, they can be affected by pixel processing that occurs within the data path. This includes:

- Black pedestal adjustment
- Lens and color shading correction

These effects can be eliminated by the following register settings:

- R0x3044–5[10] = 0
- R0x30CA–B[0] = 1
- R0x30D4–5[15] = 0
- R0x31E0–1[0] = 0
- R0x301A–B[3] = 0 (enable writes to data pedestal)
- R0x301E–F = 0x0000 (set data pedestal to “0”)
- R0x3780[15] = 0 (turn off lens/color shading correction)

Solid Color Test Pattern

In this mode, all pixel data is replaced by fixed Bayer pattern test data. The intensity of each pixel is set by its associated test data register.

100% Color Bars Test Pattern

In this test pattern, shown in Figure 24 on page 51, all pixel data is replaced by a Bayer version of an 8-color color-bar chart (white, yellow, cyan, green, magenta, red, blue, black). Each bar is 1/8 of the width of the pixel array ($4208/8 = 526$ pixels). The pattern repeats after $8 \times 526 = 4208$ pixels.

Each color component of each bar is set to either 0 (fully off) or 0x3FF (fully on for 10-bit data).

The pattern occupies the full height of the output image.

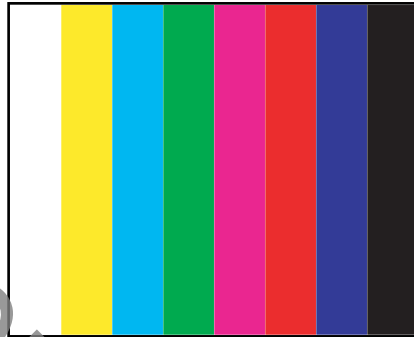
The image size is set by `x_addr_start`, `x_addr_end`, `y_addr_start`, `y_addr_end` and may be affected by the setting of `x_output_size`, `y_output_size`. The color-bar pattern is disconnected from the addressing of the pixel array, and will therefore always start on the first visible pixel, regardless of the value of `x_addr_start`. The number of colors that are visible in the output is dependent upon `x_addr_end - x_addr_start` and the setting of `x_output_size`: the width of each color bar is fixed at 526 pixels.

The effect of setting `horizontal_mirror` in conjunction with this test pattern is that the order in which the colors are generated is reversed: the black bar appears at the left side of the output image. Any pattern repeat occurs at the right side of the output image regardless of the setting of `horizontal_mirror`. The state of `vertical_flip` has no effect on this test pattern.

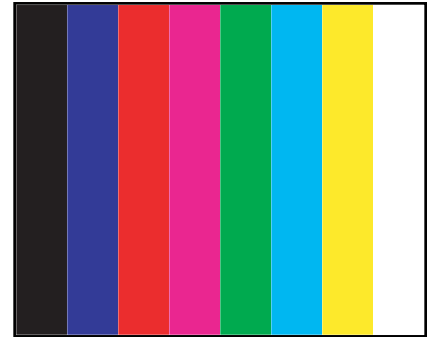
The effect of subsampling and scaling of this test pattern is undefined.

Figure 24: 100 Percent Color Bars Test Pattern

Horizontal mirror = 0



Horizontal mirror = 1


Fade-to-gray Color Bars Test Pattern

In this test pattern, shown in Figure 25 on page 52, all pixel data is replaced by a Bayer version of an 8-color, color-bar chart (white, yellow, cyan, green, magenta, red, blue, black). Each bar is 1/8 of the width of the pixel array ($4208/8 = 526$ pixels). The test pattern repeats after 4208 pixels.

Each color bar fades vertically from zero or full intensity at the top of the image to 50 percent intensity (mid-gray). The pattern of periodic repeating rows is corresponding to half the full intensity. For example, repeating every 512 rows in 10-bit mode. Each color bar is divided into a left and a right half, in which the left half fades smoothly and the right half fades in quantized steps.

The speed at which each color fades is dependent on the sensor's data width and the height of the pixel array. We want half of the data range (from 100 or 0 to 50 percent) difference between the top and bottom of the pattern. Because of the Bayer pattern, each state must be held for two rows.

The rate-of-fade of the Bayer pattern is set so that there is at least one full pattern within a full-sized image for the sensor. Factors that affect this are the resolution of the ADC (10-bit) and the image height. For example, the AR1337 fades the pixels by 2 LSB for each two rows. With 10-bit data, the pattern is 3120 pixels high and repeats after that, if the window is higher.

The image size is set by `x_addr_start`, `x_addr_end`, `y_addr_start`, `y_addr_end` and may be affected by the setting of `x_output_size`, `y_output_size`. The color-bar pattern starts at the first column in the image, regardless of the value of `x_addr_start`. The number of colors that are visible in the output is dependent upon `x_addr_end - x_addr_start` and the setting of `x_output_size`: the width of each color bar is fixed at 526 pixels.

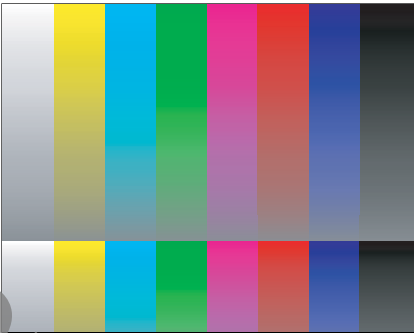
The effect of setting `horizontal_mirror` or `vertical_flip` in conjunction with this test pattern is that the order in which the colors are generated is reversed: the black bar appears at the left side of the output image. Any pattern repeat occurs at the right side of the output image regardless of the setting of `horizontal_mirror`.

The effect of subsampling and scaling of this test pattern is undefined.

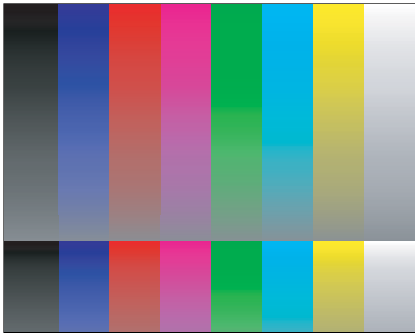


Figure 25: Fade-to-Gray Color Bars Test Pattern

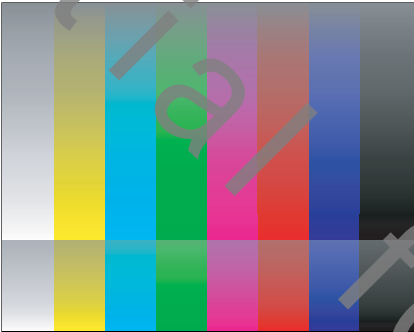
Horizontal mirror = 0, Vertical flip = 0



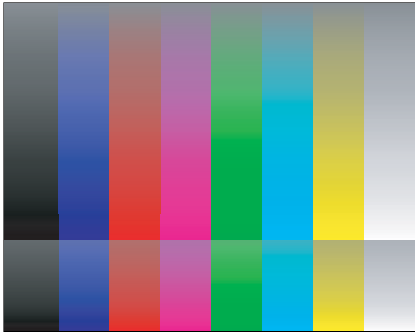
Horizontal mirror = 1, Vertical flip = 0



Horizontal mirror = 0, Vertical flip = 1



Horizontal mirror = 1, Vertical flip = 1



Walking 1s

When selected, a walking 1s pattern will be sent through the digital pipeline. The first value in each row is 0. Each value will be valid for 2 pixels.

Figure 26: Walking 1s 10-bit Pattern

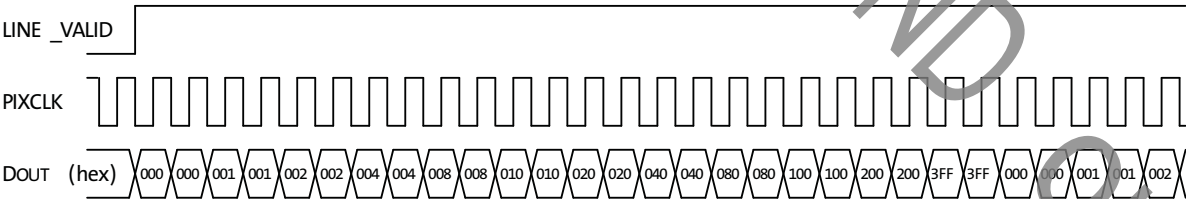
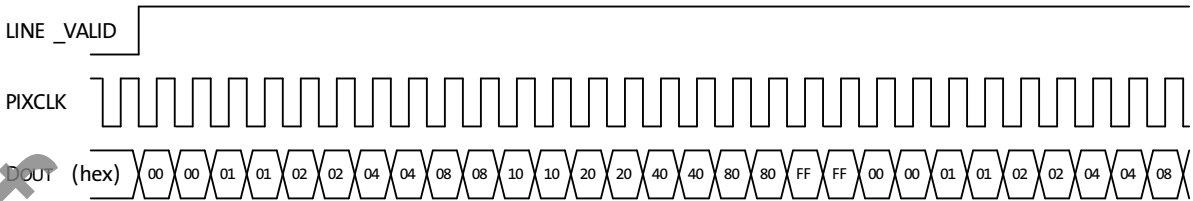




Figure 27: Walking 1s 8-bit Pattern



The walking 1s pattern was implemented to facilitate assembly testing of modules.

The walking 1 test pattern is not active during the blanking periods, hence the output would reset to a value of 0x0. When the active period starts again, the pattern would restart from the beginning. The behavior of this test pattern is the same between full resolution and subsampling modes. RAW10 and RAW8 walking 1 modes are enabled by different test pattern codes.

Pedestal

This block adds the value from R0x0008-9 or (data_pedestal_) to the incoming pixel value.

The data_pedestal register is read-only by default but can be made read/write by clearing the lock_reg bit in R0x301A-B.

The only way to disable the effect of the pedestal is to set it to 0.

Phase Detection Autofocus (PDAF)

HIDM's AR1337 solution provides a phase detection pixel solution for 1/3.2-inch PDAF sensors built on an industry proven 13 MP image quality based sensor.

PDAF Concept

AR1337 Phase Detection AF (PDAF) pixels support faster focus by determining the direction and amount of lens movement to achieve focus. Phase detect pixel accept light from a limited range of angles to determine if an object is in focus (PDAF pixels have to be on an edge to be useful for focus). Two adjacent phase detect pixels will measure the signal from two directions. When the signals are equal the object is in focus. Many phase pixels are needed because these pixels have to be near an edge to generate a difference and in low light the SNR of the phase difference will be low. PDAF correlations are analyzed on the AP microprocessor. When a focus correction is needed the lens is moved, resulting in the lens being close to focus. With faster AF information from HIDM PDAF, the number of steps to find the optimal focus can be significantly reduced. 0.5% of the pixels on AR1337 active array will be PDAF pixels.

Table 21: Pixel Information

	X,Y (in μm)	X, Y (in μm)	X,Y (in resolution)
	from Die Center	from Center of Pad 1	from First Clear Pixel
First clear pixel	2322.8 μm , 2027.0 μm	-655.2 μm , -370.6 μm	(0,0)
First readout pixel (as 13 Mp)	2314 μm , 2018.2 μm	-664 μm , -379.4 μm	(-8,-8)
First PDAF row	2102.8 μm , 1780.6 μm	-875.2 μm , -617 μm	(-200,-224)

The addresses of the first PDAF row (in resolution) are in reference to the first clear pixel. Hence from sensor addressing space point of view, the first PDAF line is at row-224(= 216+8); the step size is 192-line between two PDAF lines.

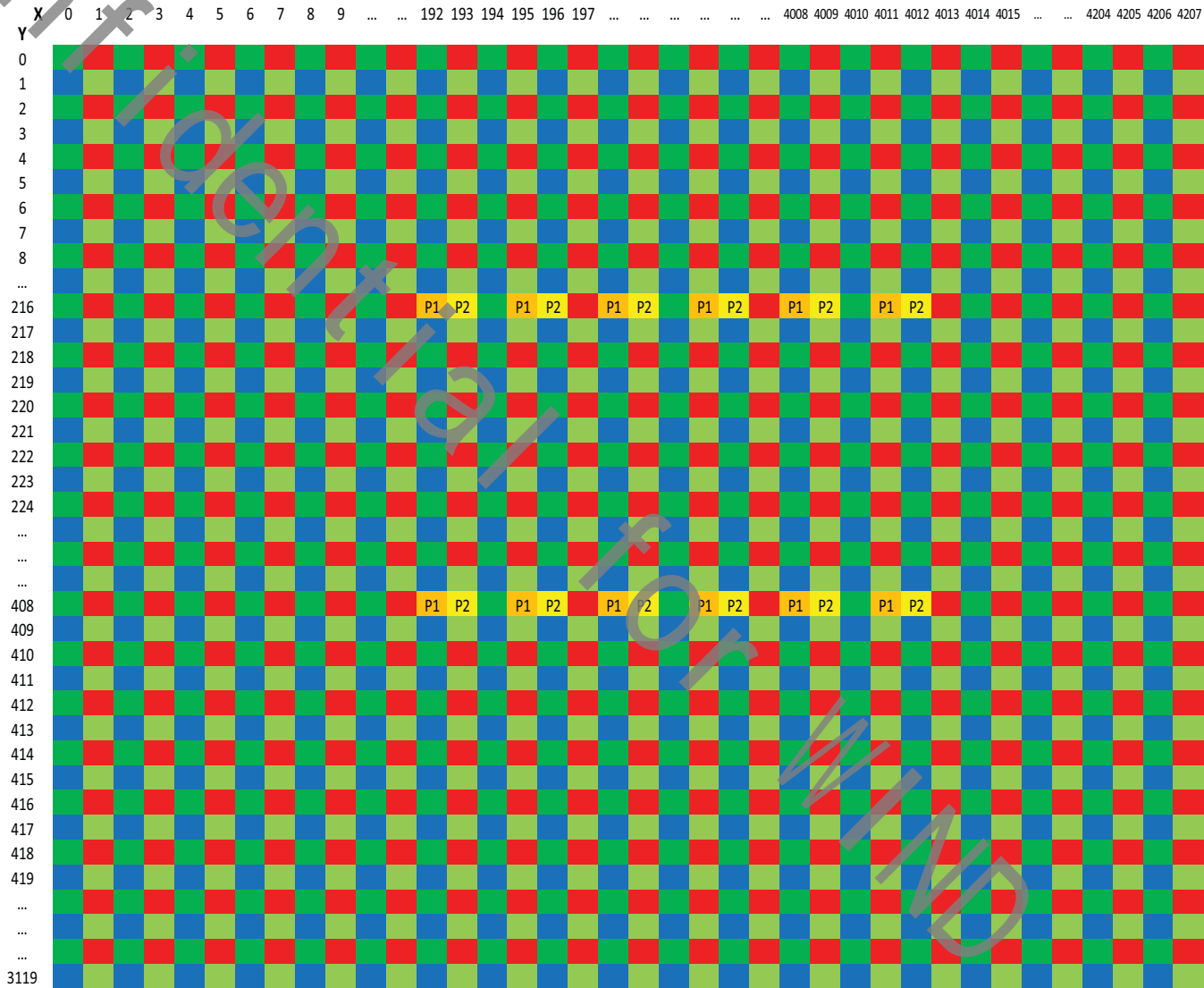


AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor
Phase Detection Autofocus (PDAF)

PDAF Layout

Figure 28: PDAF Pixel Pattern

- Active Array size: 4208 x 3120
- X/Y Starting Address is (8, 8)



Note: PDAF pixels use green CFA and are shown in Yellow only to highlight them in Figure 28.

PDAF Line Distribution

The Array contains a total of 15 lines of PDAF pixels

- Active array size: 4208 x 3120
- X/Y starting address is (8,8)
- First PDAF line is located at row -216
- There are 192 lines between each PDAF Line
- PDAF pixels represent 0.5% of all pixels

Table 22 below references active pixel area (8 pixel border excluded).

Table 22: PDAF Line Number

Start 216, step 192	
PDAF line #	row #
0	216
1	408
2	600
3	792
4	984
5	1176
6	1368
7	1560
8	1752
9	1944
10	2136
11	2328
12	2520
13	2712
14	2904

PDAF Pixels start at Column 192 and end at column 4012. Each row consists of PDAF pixel pairs with standard color (red or green) pixels in between" (see Figure 29 below).

Figure 29: PDAF Pixel at Each Row

0	1	2	...	192	193	194	195	196	197	198	199	200	...	4010	4011	4012	4013	4014	4015	...	4205	4206	4207
G	R	G	...	P1	P2	G	P1	P2	R	P1	P2	G	...	G	P1	P2	R	G	R	...	R	G	R

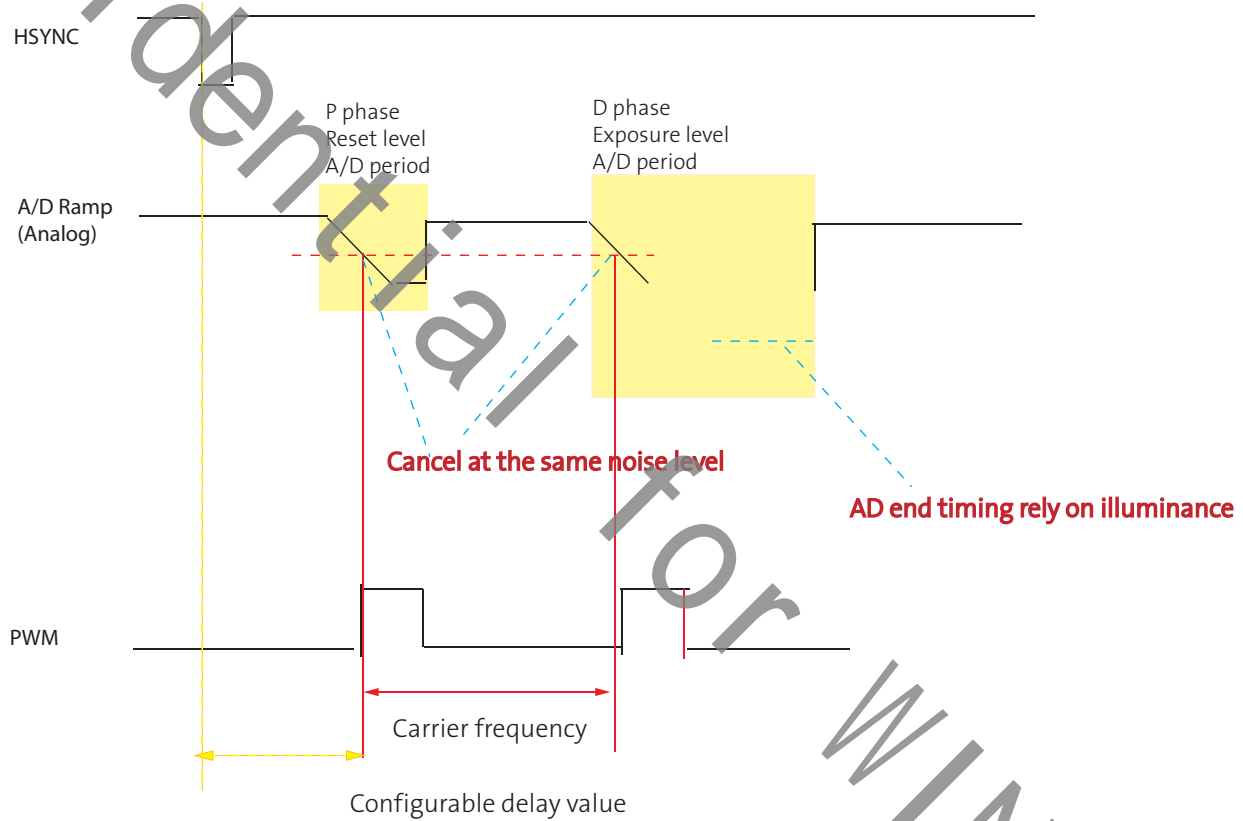
Note: All of these numbers exclude 8 pixel border

HSYNC for OIS

PWM low noise method

To reduce PWM noise, it needs to synchronize PWM carrier frequency with CDS timing of image sensor. If both noise levels between RST and SIG are same, PWM noise could be reduced. So PWM frequency needs to be equal to CDS frequency. HSYNC of image sensor is used to align PWM start timing at every line process.

Figure 30: Synchronization Between AR1337 and PWM/OIS Device



Timing Specifications

By design, the 1.8V DVDD_IO is an always-on-power domain. The 1.2V DVDD_PHY is used in MIPI and is also always-on without power switching. The rest of the digital power supplies (DVDD, DVDD_PLL) and the analog power supplies (VAA, VAAPIX, VDDIO_ANA, and DVDD_ANA) are switched power supplies. These power supplies are disconnected from the chip core by power switches in hard standby mode

Power-Up Sequence

The recommended power-up sequence for the AR1337 is shown in Figure 31.

1. Set XSHUTDOWN LOW
2. Power up VDD_IO (1.8V-only), VDDIO_ANA (1.8V-only)
3. After 1-500ms, power up VDD, VDD_ANA, VDD_PLL, VDD_PHY (1.2V) and VAA, VAA_PIX (2.7V) (any order).
4. Apply EXTCLK (can be applied anytime).
5. After 1-500ms, set XSHUTDOWN HIGH
6. HOST configuration through I²C (PLL, output, etc).
7. Set mode_select = 1 bit.
8. PLL internally enables and locks.
9. AR1337 enters streaming mode.

Figure 31: Power-Up Sequence

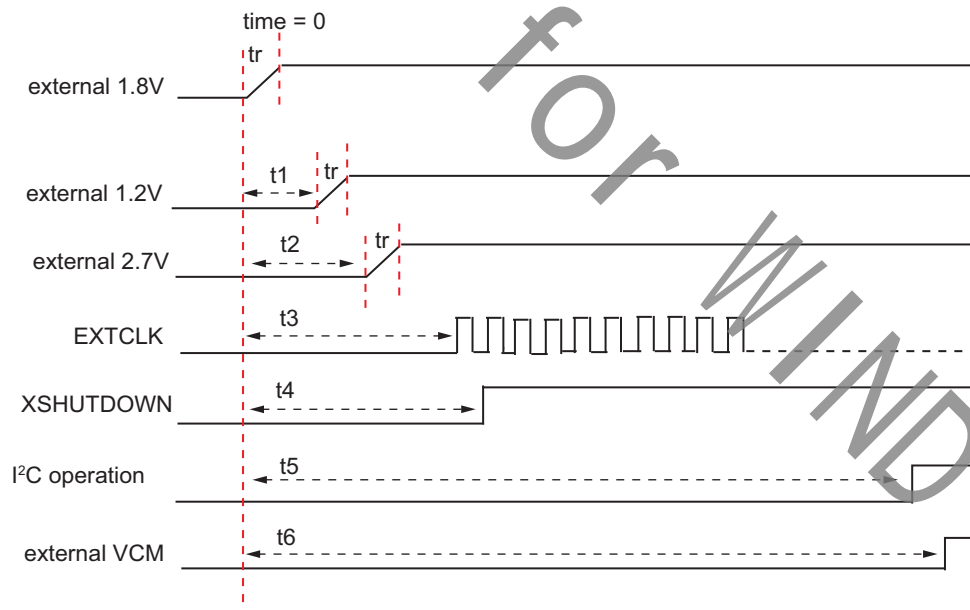


Table 23: Power-Up Sequence

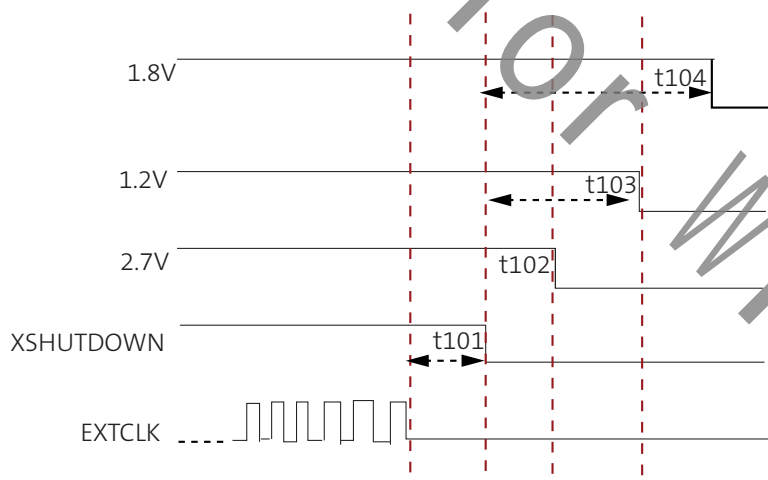
Timing	Minimum	Maximum
t1	1ms	—
t2	t1 + 1ms	—
t3	—	< t4
t4	t2 + 1ms	—
t5	t4 + 1ms	—
tr	100μs	—

A minimum ramping time of 100uS for “tr” is needed to minimize surge of supply current when powering up the 1.2V, 1.8V, and 2.7V external power supplies.

Power-Down Sequence

The recommended power-down sequence for the AR1337 is shown in Figure 32. The available power supplies—VDD_IO, VDDIO_ANA, VDD, VDD_ANA, VDD_PLL, VDD_PHY, VAA, VAA_PIX—can be turned off at the same time or have the separation specified below.

1. Set software standby mode (mode_select = 0) register.
2. Wait till the end of the current frame.
3. Set XSHUTDOWN LOW.
4. Power off VDD, VDD_PLL, VDD_PHY, (1.2V) and VAA, VAA_PIX (2.7V) (any order).
5. Power off VDD_IO (1.8V).

Figure 32: Power-Down Sequence**Table 24: Recommended Power-Down Timing**

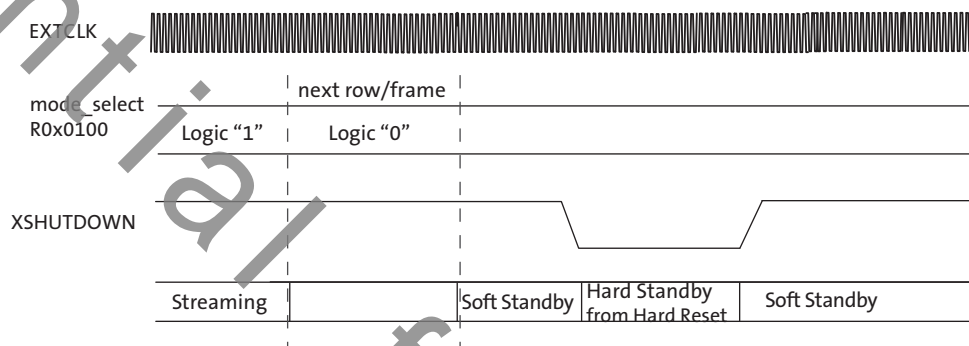
Timing	Minimum	Maximum
t101	10μs	10ms
t102	10ms	—
t103	10ms	—
t104	max(t102, t103) + 10ms	—

Hard Standby and Hard Reset

The hard standby state is reached by the assertion of the XSHUTDOWN pads. Register values are not retained by this action, and will be returned to their default values once hard reset is completed. The minimum power consumption is achieved by the hard standby state. The details of the sequence are described below and shown in Figure 33.

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

Figure 33: Hard Standby and Hard Reset



Soft Standby and Soft Reset

The AR1337 can reduce power consumption by switching to the soft standby state when the output is not needed. Register values are retained in the soft standby state. Once this state is reached, soft reset can be enabled optionally to return all register values back to the default. The details of the sequence are described below and shown in Figure 34 on page 61.

Soft Standby

1. Disable streaming if output is active by setting mode_select = 0 (R0x0100).
2. The soft standby state is reached after the current row or frame, depending on configuration, has ended.
3. The soft standby with external clock disabled retains register values, uses minimum power, and allows faster power-up.



Soft Reset

- 1. Follow the soft standby sequence list above.
- 2. Set software_reset = 1 (R0x0103) to start the internal initialization sequence.
- 3. After 15000 EXTCLKs, the internal initialization sequence is completed and the current state returns to soft standby automatically. All registers, including software_reset, returns to their default values.

Figure 34: Soft Standby and Soft Reset

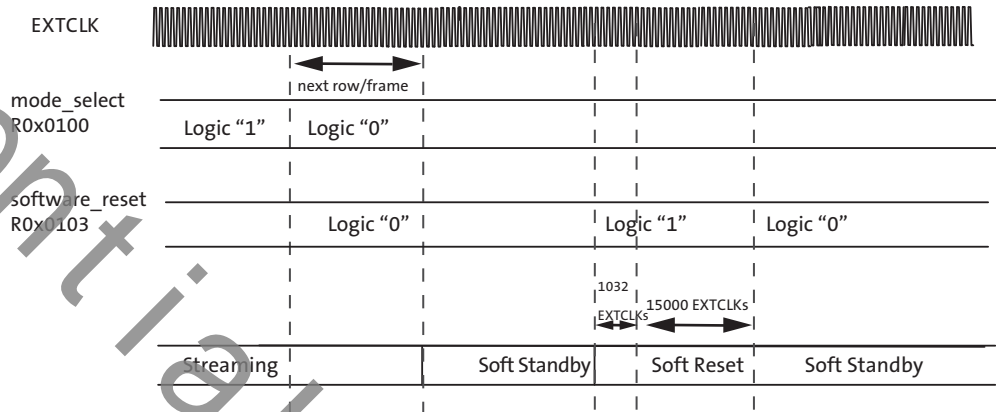


Image Sensor Characteristics

Spectral Characteristics

Quantum efficiency curves are measured using a monochromator with step size of 10 nm. Green pixel responsivity is calculated for an illuminant with spectral power density of a black body at 5100K, and optical transmission of a K088-BIR690UV IR cut filter with a Largan 9570A lens. Spectrum of the incident light is then converted to lux using the photopic response curve of the human eye. Responsivity is calculated as the ratio between charge generation rate and image plane illuminance.

Figure 35: Quantum Efficiency vs. Wavelength
TBD

Chief Ray Angle vs. Image Height

Figure 36: Chief Ray Angle vs. Image Height

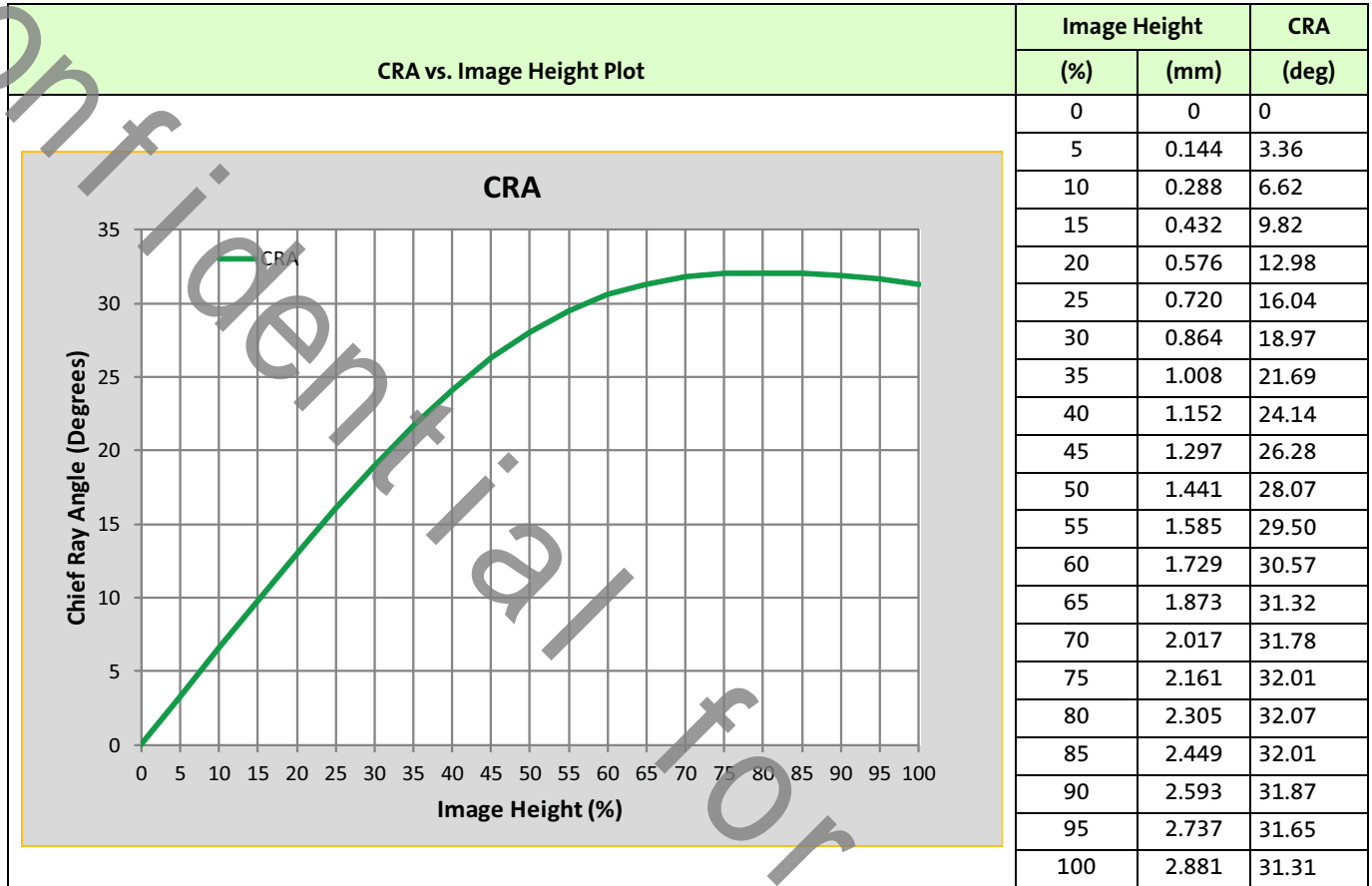


Table 25: Array Dimensions

Pixel Size	Array Size (pixels)		Array Size (mm)	
1.1μm	X _{max}	4208	X _{max}	4.63
	Y _{max}	3120	Y _{max}	3.43
			Diagonal	5.76

Read the Sensor CRA

Follow the steps below to obtain the CRA value of the image sensor:

1. Set the register bit field R0x301A[5] = 1.
2. Read the register bit fields R0x31FE [6:4].
3. Determine the CRA value according to Table 26.

Table 26: CRA Value

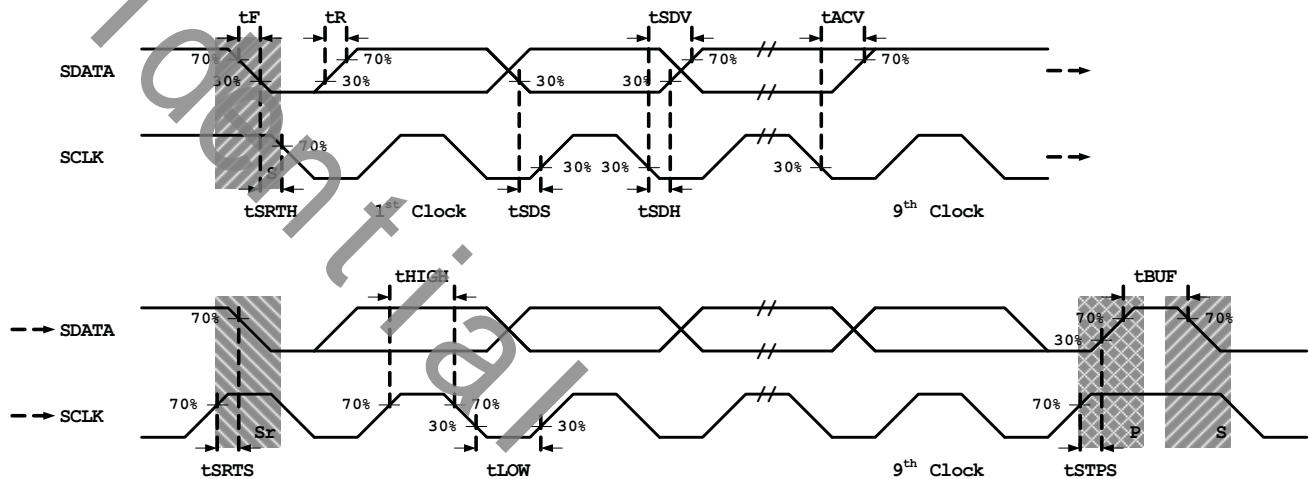
Binary Value of R0x31FE[6:4]	CRA Value
000	32

Electrical Characteristics

Two-Wire Serial Register Interface

The electrical characteristics of the two-wire serial register interface (SCLK, SDATA) are shown in Figure 37 and Table 27 on page 65. Table 28 on page 66 shows the timing specification for the two-wire serial interface.

Figure 37: Two-Wire Serial Bus Timing Parameters



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



AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor Electrical Characteristics

Table.27: Two-Wire Serial Register Interface Electrical Characteristics

$f_{EXTCLK} = 25 \text{ MHz}$; $V_{AA} = 2.7\text{V}$; $V_{AA_PIX} = 2.7\text{V}$; $V_{DD_IO} = 1.8\text{V}$; $V_{DD} \text{ (digital core)} = 1.2\text{V}$; $V_{DD_PHY} = 1.2\text{V}$;
Output load = 68.5pF; $T_J = 60^\circ\text{C}$

Symbol	Parameter	Condition	Fast Mode			Fast Mode Plus			Unit
			Min	Typ	Max	Min	Typ	Max	
SDATA									
V _{IL}	Input LOW voltage	—	-0.3	—	0x3 x V _{DD_IO}	-0.3	—	0x3 x V _{DD_IO}	V
V _{IH}	Input HIGH voltage	—	0.7 x V _{DD_IO}	—	V _{DD_IO} + 0.3	0.7 x V _{DD_IO}	—	V _{DD_IO} + 0.3	V
I _{IL}	Input leakage current	No pull up resistor; V _{IN} = V _{DD_IO} or DGND	10	—	14	10	—	14	μA
V _{OL}	Output LOW voltage	At specified 2mA	0.11	—	0.3	0.11	—	0.3	V
I _{OL}	Output LOW current	At specified V _{OL} 0.4V	3	—	N/A	20	—	N/A	mA
C _{IN}	Input pad capacitance	—	2.4	—	2.62	2.4	—	2.62	pF
C _{LOAD}	Load capacitance	—	-	—	N/A	—	—	N/A	pF
SCLK									
V _{IL}	Input LOW voltage	—	-0.3	—	0x3 x V _{DD_IO}	—	—	—	V
V _{IH}	Input HIGH voltage	—	0.7 x V _{DD_IO}	—	V _{DD_IO} + 0.3	—	—	—	V
I _{IL}	Input leakage current	No pull up resistor; V _{IN} = V _{DD_IO} or DGND	10	—	14	—	—	—	μA
V _{OL}	Output LOW voltage	At specified 2mA	0.11	—	0.3	—	—	—	V
I _{OL}	Output LOW current	At specified V _{OL} 0.1V	3	—	N/A	—	—	—	mA
C _{IN}	Input pad capacitance	—	2.4	—	2.63	—	—	—	pF
C _{LOAD}	Load capacitance	—	-	—	N/A	—	—	—	pF

Table 28: Two-Wire Serial Interface Timing Specifications

Symbol	Definition	Fast Mode		Fast Mode Plus		Unit
		Min	Max	Min	Max	
f _{SCLK}	SCLK Frequency	0	400	0	1000	KHz
t _{HIGH}	SCLK High Period	0.6	—	0.26	—	μs
t _{LOW}	SCLK Low Period	1.3	—	0.5	—	μs
t _{SRTS}	START Setup Time	0.6	—	0.26	—	μs
t _{SRTH}	START Hold Time	0.6	—	0.26	—	μs
t _{SDS}	Data Setup Time	100	—	50	—	ns
t _{SDH}	Data Hold Time	0	—	0	—	μs
t _{SDV}	Data Valid Time	—	0.9	—	0.45	μs
t _{ADV}	Data Valid Acknowledge Time	—	0.9	—	0.45	μs
t _{STPS}	Stop Setup Time	0.6	—	0.26	—	μs
t _{BUF}	Bus Free Time between STOP and START	1.3	—	0.5	—	μs
t _R	SCLK and S _{DATA} Rise Time	20	300	—	120	ns
t _F	SCLK and S _{DATA} Fall Time	20	300	20	120	ns

EXTCLK

The electrical characteristics of the EXTCLK input are shown in Table 29. The EXTCLK input supports an AC-coupled sine-wave input clock or a DC-coupled square-wave input clock.

If EXTCLK is AC-coupled to the AR1337 and the clock is stopped, the EXTCLK input to the AR1337 must be driven to ground or to VDD_IO. Failure to do this will result in excessive current consumption within the EXTCLK input receiver.

Table 29: Electrical Characteristics (EXTCLK)

$f_{EXTCLK} = 25 \text{ MHz}$; $V_{AA} = 2.7\text{V}$; $V_{AA_PIX} = 2.7\text{V}$; $V_{DD_IO} = 1.8\text{V}$; $V_{DD} \text{ (digital core)} = 1.2\text{V}$; $V_{DD_PHY} = 1.2\text{V}$;

Output load = 68.5pF; $T_J = 60^\circ\text{C}$

Symbol	Parameter	Condition	Min	Typ	Max	Unit
f_{EXTCLK}	Input clock frequency	PLL enabled	6	—	48	MHz
t_R	Input clock rise slew rate	$C_{LOAD} < 20\text{pF}$	—	2.883	—	ns
t_F	Input clock fall slew rate	$C_{LOAD} < 20\text{pF}$	—	2.687	—	ns
V_{IN_AC}	Input clock minimum voltage swing (AC coupled)	—	0.5	—	—	V (p-p)
V_{IN_DC}	Input clock maximum voltage swing (DC coupled)	—	—	—	$V_{DD_IO} + 0.5$	V
$f_{CLKMAX} \text{ (AC)}$	Input clock signaling frequency (low amplitude)	$V_{IN} = V_{IN_AC} \text{ (MIN)}$	—	—	25	MHz
$f_{CLKMAX} \text{ ((dc)}$	Input clock signaling frequency (full amplitude)	$V_{IN} = V_{DD_IO}$	—	—	48	MHz
	Clock duty cycle	—	45	50	55	%
t_{JITTER}	Input clock jitter	cycle-to-cycle	—	545	600	ps
t_{LOCK}	PLL VCO lock time	—	—	0.2	1	ms
C_{IN}	Input pad capacitance	—	—	6	—	pF
I_{IH}	Input HIGH leakage current	—	0	—	10	μA
V_{IH}	Input HIGH voltage	—	$0.7 \times V_{DD_IO}$	—	$V_{DD_IO} + 0.5$	V
V_{IL}	Input LOW voltage	—	-0.5	—	$0.3 \times V_{DD_IO}$	V

Serial Pixel Data Interface

The electrical characteristics of the serial pixel data interface are shown in Table 30.

To operate the serial pixel data interface within the electrical limits of the CSI-2 and MIPI specifications, VDD_IO (I/O digital voltage) is restricted to operate in the range 1.7–1.9V.

Table 30: Electrical Characteristics (Serial MIPI Pixel Data Interface)

Symbol	Parameter	Min	Typ	Max	Unit
V _{OD}	High speed transmit differential voltage	140	–	270	mV
V _{CMTX}	High speed transmit static common-mode voltage	150	–	250	mV
ΔV _{CMTX}	V _{CMTX} mismatch when output is Differential-1 or Differential-0	<5	–	–	mV
ΔV _{OD}	V _{OD} mismatch when output is Differential-1 or Differential-0	<10	–	–	mV
Z _{OS}	Single ended output impedance	40	–	62.5	Ω
ΔZ _{OS}	Single ended output impedance mismatch	<10	–	–	%
ΔV _{CMTX(L,F)}	Common level variation between 50-450 MHz	<25	–	–	mV
t _R	Rise time (20-80%)	150	–	0.3*UI	ps
t _F	Fall time (20-80%)	150	–	0.3*UI	ps
V _{OL}	Output LOW level	-50	–	50	mV
V _{OH}	Output HIGH level	1.1	–	1.3	V
Z _{OLP}	Output impedance of low power parameter	110	–	–	Ω
T _{RLP}	15-85% rise time	–	–	25	ns
T _{FLP}	15-85% fall time	–	–	25	ns
ΔV/Δt _{sr}	Slew rate (C _{LOAD} = 0-70pF)	30	–	–	mV/ns

MIPI Specification Reference

The sensor design and this documentation is based on the following MIPI specifications:

- MIPI Alliance Standard for CSI-2 version 1.0
- MIPI Alliance Standard for D-PHY version 1.0

Control Interface

The electrical characteristics of the control interface (XSHUTDOWN, TEST, GPIO0, GPIO1, GPIO2, and GPIO3) are shown in Table 31.

Table 31: DC Electrical Characteristics (Control Interface)

f_{EXTCLK} = 25 MHz; V_{AA} = 2.7V; V_{AA_PIX} = 2.7V; V_{DD_IO} = 1.8V; V_{DD} (DIGITAL CORE) = 1.2V; V_{DD_PHY} = 1.2V;
 Output load = 68.5pF; T_J = 60°C

Symbol	Parameter	Condition	Min	Typ	Max	Unit
V _{IH}	Input HIGH voltage		0.7 x V _{DD_IO}	–	V _{DD_IO} + 0.5	V
V _{IL}	Input LOW voltage		–0.5	–	0.3 x V _{DD_IO}	V
I _{IN}	Input leakage current	No pull-up resistor; V _{IN} = V _{DD_IO} or DGND	–	–	10	μA
C _{IN}	Input pad capacitance		–	6	–	pF



AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor Electrical Characteristics

Operating Voltages

VAA and VAA_PIX must be at the same potential for correct operation of the AR1337.

Table 32: DC Electrical Definitions and Characteristics

$f_{EXTCLK} = 25 \text{ MHz}$; VAA = 2.7V; VAA_PIX = 2.7V; VDD_IO = 1.8V; VDD (digital core) = 1.2V; VDD_PHY = 1.2V;
Output load = 68.5pF; Tj = 60°C

Symbol	Parameter	Condition	Min	Typ	Max	Unit
VDD_IO	I/O digital voltage	—	—	1.8	—	V
VDD	Digital voltage	—	—	1.2	—	V
VDD_PHY	PHY digital voltage	—	—	1.2	—	V
VAA	Analog voltage	—	—	2.7	—	V
VAA_PIX	Pixel supply voltage	—	—	2.7	—	V
Operating Current						
IDD_IO	I/O digital current	13M full mode (4208x3120), 30fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	
IDD_IO	I/O digital current	13M full mode (4208x3120), 24fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	
IDD_IO	I/O digital current	4K UHD full mode (3840 x 2160), 30fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	
IDD_IO	I/O digital current	4K Cinema full mode (4096 x 2160), 30fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	
IDD_IO	I/O digital current	1080p (1920x1080), crop+bin2+scaling, 30fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	
IDD_IO	I/O digital current	1080p LP (1920x1080), crop+bin2+scaling, 30fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	
IDD_IO	I/O digital current	720p (1280x720), crop+skip2+scaling, 30fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	
IDD_IO	I/O digital current	1080p (1920x1080), crop+bin2+scaling, 60fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	
IDD_IO	I/O digital current	1080p LP(1920x1080), crop+bin2+scaling, 60fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	
IDD_IO	I/O digital current	720p(1280x720), crop+skip2+scaling, 60fps	—	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		—	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		—	TBD	TBD	



AR1337: 1/3.2-Inch 13Mp CMOS Digital Image Sensor Electrical Characteristics

Table 32: DC Electrical Definitions and Characteristics (continued)

$f_{EXTCLK} = 25 \text{ MHz}$; $V_{AA} = 2.7\text{V}$; $V_{AA_PIX} = 2.7\text{V}$; $V_{DD_IO} = 1.8\text{V}$; $V_{DD} \text{ (digital core)} = 1.2\text{V}$; $V_{DD_PHY} = 1.2\text{V}$;
Output load = 68.5pF; $T_J = 60^\circ\text{C}$

Symbol	Parameter	Condition	Min	Typ	Max	Unit
IDD_IO	I/O digital current	3M (2000x1500), crop+skip2+scaling	–	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		–	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		–	TBD	TBD	
IDD_IO	I/O digital current	3M LP(2000x1500), crop+skip2+scaling	–	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		–	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		–	TBD	TBD	
IDD_IO	I/O digital current	VGA (640x480), crop+skip4+scaling, 120fps	–	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		–	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		–	TBD	TBD	
IDD_IO	I/O digital current	720p (1280x720), crop+skip2+scaling, 120fps	–	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		–	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		–	TBD	TBD	
IDD_IO	I/O digital current	OVGA (320x240), bin4skip4 240fps	–	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		–	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		–	TBD	TBD	
STANDBY current						
IDD_IO	I/O digital current	XSHUTDOWN signal low (EXTCLK on)	–	TBD	TBD	μA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		–	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		–	TBD	TBD	
IDD_IO	I/O digital current	XSHUTDOWN signal low (EXTCLK off)	–	TBD	TBD	μA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		–	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		–	TBD	TBD	
IDD_IO	I/O digital current	STANDBY current when asserting R0x0100 = 0 (EXTCLK on)	–	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		–	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		–	TBD	TBD	
IDD_IO	I/O digital current	STANDBY current when asserting R0x0100 = 0 (EXTCLK off)	–	TBD	TBD	mA
IDD/IDD_PHY/IDD_ANA	PHY/digital current		–	TBD	TBD	
IAA/IAA_PIX	Analog/Pixel current		–	TBD	TBD	

This table specifies the typical measured currents with AR1337 sensor operating under typical light conditions. The per supply currents are measured with the typical and maximum supply voltages applied to the sensor.



Revision History

Rev. 1	6/23/15
• Initial release	

Confidential for WIND on