CX93510
JPEG Encoder with a 656 Camera Interface and Optional Microphone Input

Data Sheet

DSH-202155B
April 2010

CONEXANT

Ordering Information

Model Number	Description	Package
CX93510-11z*	128 KByte buffer with Differential JPEG disabled	48-pin eMLF/QFN, 6x6 mm Body. 0.5 mm Lead Pitch
CX93510-12z*	256 KByte buffer with Differential JPEG enabled	
*Lead-free (Pb Free) and RoHS compliant		

Revision History

Revision	Date	Description
p1	March 16, 2009	Preliminary p1 release
p2	July 31, 2009	Preliminary p2 release
A	October 8, 2009	Revision A release
B	April 30, 2010	Revision B release

Preliminary Information

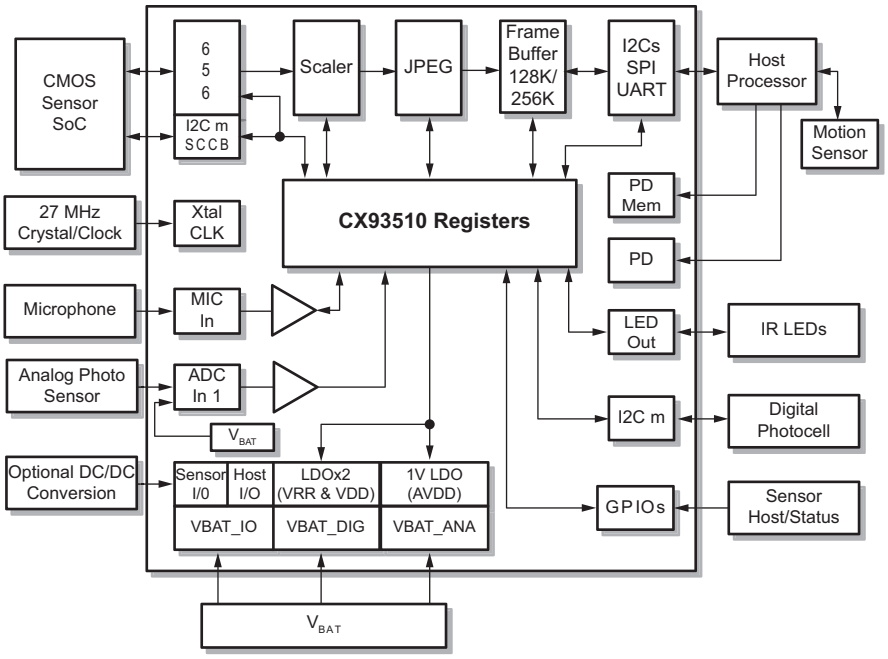
This document contains information on a new product. The parametric information, although not fully characterized, is the result of testing initial devices.

CX93510

JPEG Encoder with a 656 Camera Interface and Optional Microphone Input

The CX93510 is a monolithic mixed signal Application-Specific Standard Product (ASSP) designed for low cost and low power motion detection surveillance camera applications. Used with an external CMOS image sensor, the CX93510 can compress and store several images in an internal 256KB/128KB frame buffer. An optional 2:1 scaler can provide Quarter Video Graphics Array (QVGA) images from the VGA input. With a microphone input and 4 KB of optionally allocated buffering, a 2- or 4-bit ADPCM audio session can be recorded simultaneously during image captures. The stored images and audio data are then passed on to an external microprocessor for uploading to a desired medium. The CX93510 is controlled through a simple register set via the microprocessor interface, and the variety of available interfaces (Serial Peripheral Interface [SPI], UART, and I^2C) allow for wide flexibility in microprocessor selection. With both digital and analog photocell sensor inputs available, the CX93510 can facilitate the measurement of ambient light and use an on-chip LED driver to control an external infrared LED during low-light conditions. With low cost and low power, the CX93510 is ideally suited for security applications requiring visual verification.

Functional Block Diagram



Distinguishing Features

- Supported Applications and operating Modes
  - PIR Sensor with Video
    - ✓ Visual verification of intruder via image sensor interface
    - ✓ VGA - B&W or Color @ up to 30 fps
    - ✓ VGA to QVGA Filtered ½ res Scaling
    - ✓ JPEG & MJPEG-DPCM Image Compression (ISO/IEC 10918-1/2)
    - ✓ Differential JPEG Mode
    - ✓ Continuous streaming and variable image modes
    - ✓ 256 KB or 128 KB frame buffer for compressed images (no external memory)
    - ✓ Interface to external µP though SPI, UART, or I^2C
    - ✓ Variable IR illumination control port
    - ✓ A/D for Photocell sensor, Battery voltage monitor, or microphone inputs
    - ✓ Sleep mode - SoC off except frame buffer in retention mode
  - Video Intercom/Door Phone
  - Baby monitor applications
  - Remote home monitoring
- Interfaces
  - Sensor i/f
    - ✓ 8-b 4:2:2 YCbCr with BT-656 embedded timing codes or frame/line sync support up to 27 MHz, progressive mode
    - ✓ Resolutions: VGA (640x480) & QVGA (320x240)
    - ✓ 27 MHz clk output
    - ✓ 2/3 wire control i/f: I^2C Master port or SCCB
  - 4-wire I2C/SPI/UART slave port to external µP
  - 8 GPIO (5 dedicated pins, 3 shared pins)
  - IR illumination with variable DAC and PWM control
  - Microphone input, mic boost 0-36 dB in 6 dB steps, 2- and 4- bit ADPCM
  - DC measurement battery monitor
  - Photocell sensor input - analog or I^2C (shared with GPIO)
  - Support for battery operation: 3.6 V to 1.8 V
- 48-pin eMLF/QFN
  - -10 °C to +85 °C ambient, +100 °C junction
  - 12 mA in operational mode
  - 10 nA in sleep mode

Contents

Figures

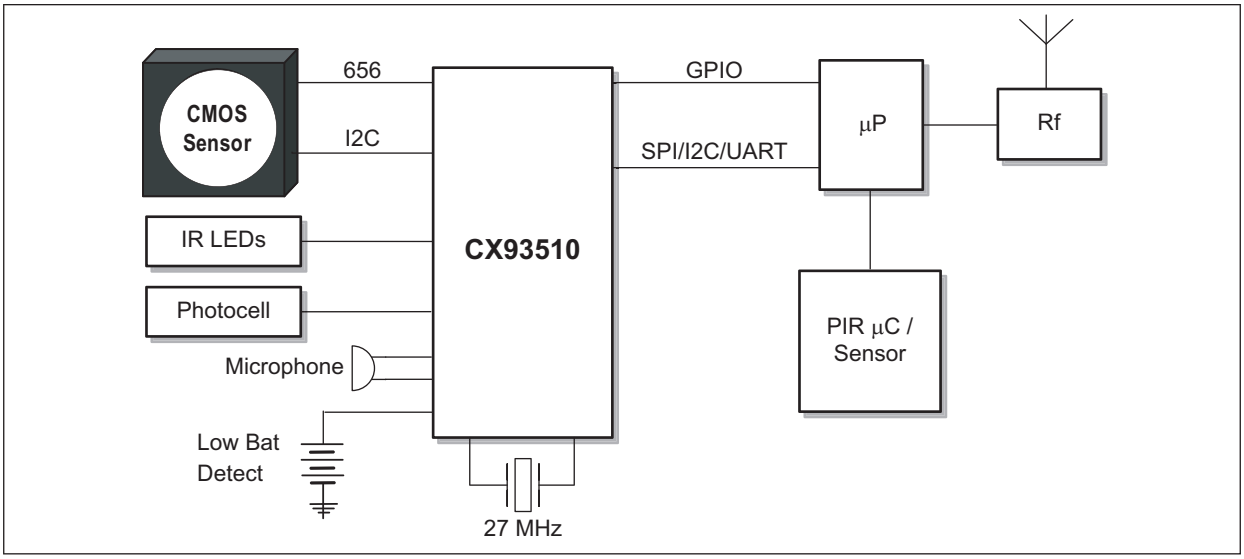
Tables

1 Overview

1.1 Passive Infrared Security System

Figure 1 illustrates a Typical Passive Infrared (PIR) security system with video and audio verification. Upon a motion event, the host processor wakes and configures the CX93510 in preparation for image/audio captures. Depending on the selected CMOS sensor, the system should be able to wake, measure ambient light conditions, configure the CMOS sensor, illuminate the IR LEDs for low light conditions, and begin image/audio captures within 100 ms of motion being detected.

Figure 1. PIR Sensor with Visual Verification System Block Diagram



As the image and audio data are acquired, both are compressed and stored in the internal 256 KB (or 128 KB option) frame buffer. Depending on the selected device, image data is encoded using JPEG or MJPEG_DPCM (Differential JPEG). Audio data is compressed into 4-bit ADPCM format, or a non-standard 2-bit ADPCM for maximum compression. In a typical scenario the host will read the contents of the frame buffer and send the information over an RF link to a base station where the image and audio data can be sent to desired location such as a cell phone, email address, or security service.

For systems that are battery powered, the CX93510 can operate between 3.3 V and 1.8 V. To conserve power, it can also remain in a very deep sleep during idle periods, essentially maintaining an off state until awakened by the host processor. Should the host not want to download the data immediately, an optional memory retention setting can be enabled to save the contents of the frame buffer while the rest of the device is shut down. A battery measurement input allows the host to monitor the battery level and alert the user if battery replacement is needed. Most CMOS sensors have one or more power supply feeds, of which some cannot drop too low in voltage. In these situations, the sensor and the sensor I/O on the CX93510 can be powered using a DC-DC buck-boost converter, which can be controlled using a GPIO from the CX93510. Refer to the following sections for more details on the available power options and configurations.

1.2 Power Supplies

The nominal core power supply is 0.8 V (± 5%)

The nominal AFE power supply is 1.0 V (± 5%)

The nominal I/O power supply is 3.3 V with a package pin specification of 3.47 V maximum, and 1.8 V minimum.

During normal active functional mode, power-savings modes, or clock-gating are used for non-active components, interfaces, or functions. The frame buffer is on its own supply system in order to support memory retention during sleep mode.

There are two I/O power supplies: VDDO1 and VDDO2. All the image sensor I/F pins are connected to VDDO2, and the digital I/O pins (host I/F, GPIO, test, etc.) are connected to VDDO1. Each can be at a different voltage.

1.3 Voltage Regulators

The CX93510 includes three internal linear voltage regulators that generate:

- 1.0 V supply for all analog circuitry
- 0.8 V supply for the digital core
- 0.8 V supply for the RAM

Each supply voltage can be adjusted independently by register bits. Also the analog regulator can be enabled/disabled by a register bit, while the digital and RAM regulators are enabled/disabled via the power on VBAT_DIG.

All supplies must be externally decoupled with a 1 μF capacitor.

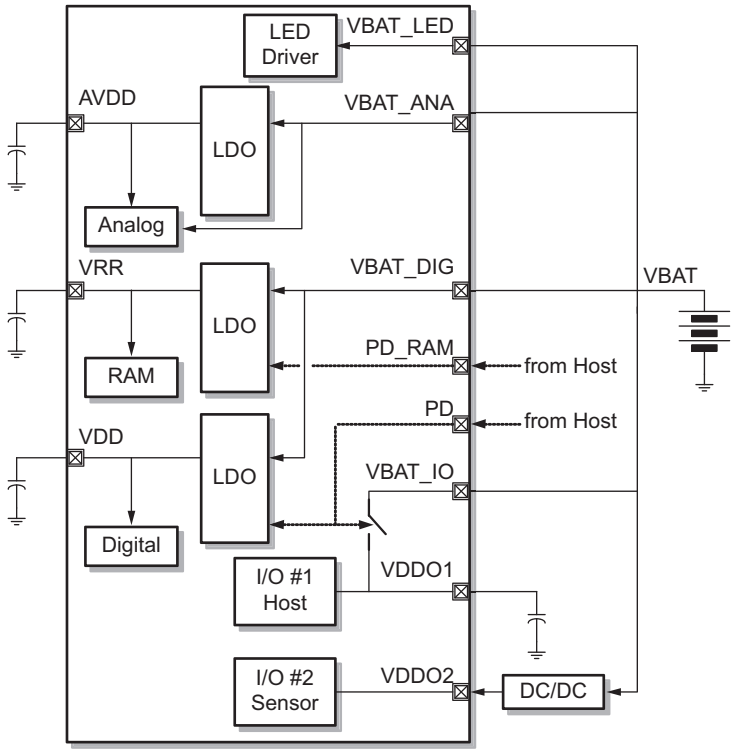
The user can choose to not use one or more internal regulators; in that case, the regulator input should be left unconnected or grounded. The output power pin can then be connected to an external power source. Note that VBAT_DIG, if connected, powers both the RAM and Digital regulators. See power management diagrams below.

Each regulator provides a POR (power-on reset) signal that goes to logic high when its output has reached a given percentage (typ 95 percent) of its final target.

1.4 Power Management

With the availability of internal voltage regulators and I/O connection options, the CX93510 allows for flexible power management options. A few examples are shown in Figures 2 through 4.

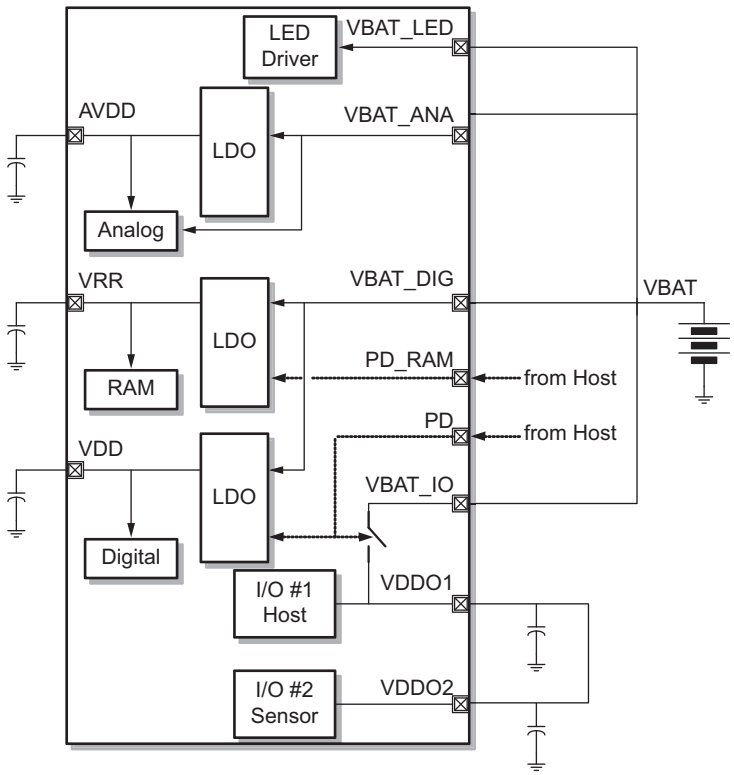
Figure 2. Basic Configuration



GENERAL NOTES:

1. Internal LDOs used for all core voltages.
2. PD pin controls VDD and VDDO1 rails.
3. Sensor interface is powered by external DC/DC converter; VDDO2 rail uses the same power rail.
4. If using a DC-DC converter to power VDDO2, gate its enable pin with the PD enable pin such that the converter will shut down when PD is driven low. In other words, VDDO2 should be shut down whenever the device is put to sleep.

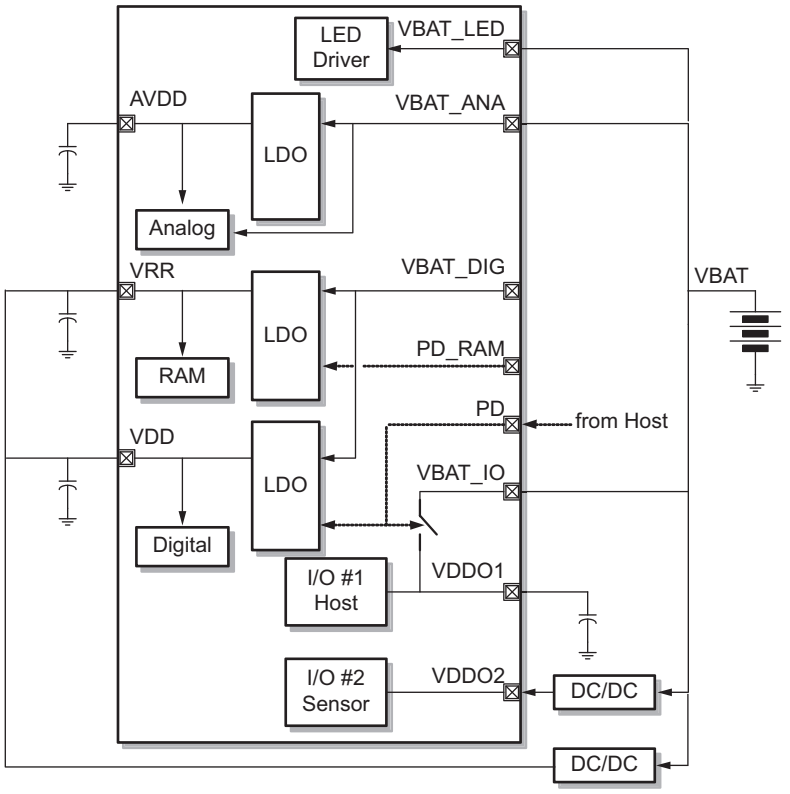
Figure 3. CMOS Sensor Runs on Battery Voltage



GENERAL NOTES:

1. VDDO2 is shorted externally to VDDO1.

Figure 4. External DC/DC Converter for Core Supplies



GENERAL NOTES:

1. VBAT_DIG pin is left unconnected; internal regulators for VDD and VRR are disabled.
2. PD_RAM pin status is not relevant.
3. PD pin controls only VDDO1 rail.
4. If using a DC-DC converter to power VDDO2, gate its enable pin with the PD enable pin such that the converter will shut down when PD is driven low.

1.5 Power Modes

For most applications the CX93510 will be in Sleep power mode > 99% of the time. When an event occurs, the external μP will awaken the CX93510 causing it to enter Idle mode as registers are loaded and the CMOS image sensor is setup. As a series of images are captured, the CX93510 will alternate between Active and Idle power modes. If the host does not wish to immediately retrieve the images but there is a desire to retain the captured images, Retention power mode can be enabled.

The following power modes are referred to within this document:

1.5.1 Sleep Mode

The CX93510 is powered off. There is zero current except for that used by the PD and PD_RAM pins to monitor the wakeup condition. All the I/O and core logic and SRAMS are powered off.

1.5.2 Idle Mode

Audio and Video processing are clock gated off. The only activity is the host interface read/write to registers or frame buffer access, and sensor i/f r/w register access.

1.5.3 Active Mode

Full activity consisting of video capture, JPEG encoding, and/or audio encoding.

1.5.4 Retention Mode

The CX93510 is powered off except for the configured portion of the frame buffer maintaining its memory state.

1.6 Crystal Input

The CX93510 clock input is from an attached 27 MHz crystal or clock source with a tolerance of 100 ppm. The clock source should have a 50 percent duty cycle, ±10 percent.

1.7 Reset

The CX93510 is reset automatically upon power-up on the VBAT_DIG pin or power applied at the VDD pin if using an external power source. The internal LDO will in turn provide a reset signal to the device.

2 Pin Description

The pins and their functions are listed in Table 1. Figure 5 provides a pin diagram, and Figure 6 provides an example schematic of the device interface connections.

Table 1. Pin Descriptions *(1 of 2)*

Pin Name	Dir	Pin Number	Description
Analog Front End			
MIC_IN	I	35	Microphone input.
MIC_BIAS	O	33	Microphone bias.
VREF	O	34	Decoupling cap for common-mode voltage reference. Use a 0.1 µF capacitor.
XTI	I	29	27.0 MHz crystal oscillator input, or single-ended clk.
XTO	O	30	Crystal buffer output.
PCELL_IN	I	36	Analog Photocell input. (0-1V)
PD	I	23	Power-down (=0). Enable (=1) signal for PIR ckts.
PD_RAM	I	24	Power-down RAM. Enable for image retention buffer.
LED_OUT	O	28	IR LED driver output.
LED_FB	I	27	IR LED driver feedback signal.
AVDD	I/O	32	1.0 V power supply for AFE. Connect 1 µF capacitor to ground.
VBAT_DIG	I	25	Battery voltage to VDD and VRR regulators.
VBAT_IO	I	20	Battery voltage to I/O power switch.
VBAT_ANA	I	31	Battery voltage to analog and AVDD regulator.
VBAT_LED	I	26	Battery voltage to LED driver.
Digital Power Supply			
VDD	I/O	21, 43	Digital Standard Cell Core Power. 0.8 V nominal. Connect both pins together and add 1 µF capacitor to ground.
VRR	I/O	6, 22	Digital Frame Buffer Core Power. 0.8 V nominal. Connect both pins together and add 1 µF capacitor to ground.
VDDO1	O	12, 19	Pad Ring Power. 3.3 V nominal. Connect both pins together and add 1 µF capacitor to ground.
VDDO2	I	48	Pad Ring Power. 3.3 V nominal.

Table 1. Pin Descriptions *(2 of 2)*

Pin Name	Dir	Pin Number	Description
General Purpose I/O			
GPIO[4:0]	I/O	7, 8, 9, 10, 11	5-b General purpose I/O port. GPIO[4:3] may be used for optional photocell I2C i/f.
Sensor Interface			
SDATA[7:0]	I	39, 40, 41, 42, 44, 45, 46, 47	Pixel data bus
SVREF	I/O	37	Frame (Vertical Active) indicates active region of frame, or GPIO[5]
SHREF	I/O	38	Line (Horizontal Active) indicates active region of line, or GPIO[6]
SPCLK	I	1	Pixel clock – up to 27 MHz; used to sample above signals
SCLK	O	2	27 MHz Clock output to Sensor.
S_SCL	I/O	4	I2C Serial Clock or SCCB SIO_C
S_SDA	I/O	3	I2C Serial Data or SCCB SIO_D
S_GP	I/O	5	SCCB_E or GPIO[7].
Host mP Slave SPI/I2C/UART Port			
HSCLK	I/O	13	Serial Clock (or SCL or TxD)
HSIMO	I/O	14	Serial Data input (or RTS)
HSOMI	I/O	15	Slave Output (or SDA or CTS)
HSS	I	16	Slave select (I2C adr sel or RxD)
Miscellaneous Chip Control			
TEST	I	17	Test Mode (tie low for normal operation).
Reserved			Reserved
Reserved			Reserved
Reserved			Reserved
Reserved			Reserved

GENERAL NOTES:

1. The exposed ground pad serves as the only ground reference and must be properly grounded. All internal grounds are down-bonded to this pad.
2. Add 1 μF decoupling capacitors near VDDO1, VDD, VRR, and AVDD (pins 19, 21, 22, and 32 respectively).
3. The CX93510 must see at least three rising edges of SPCLK before it is able to detect the first SVREF edge and capture the first frame.

Figure 5. Pin Diagram

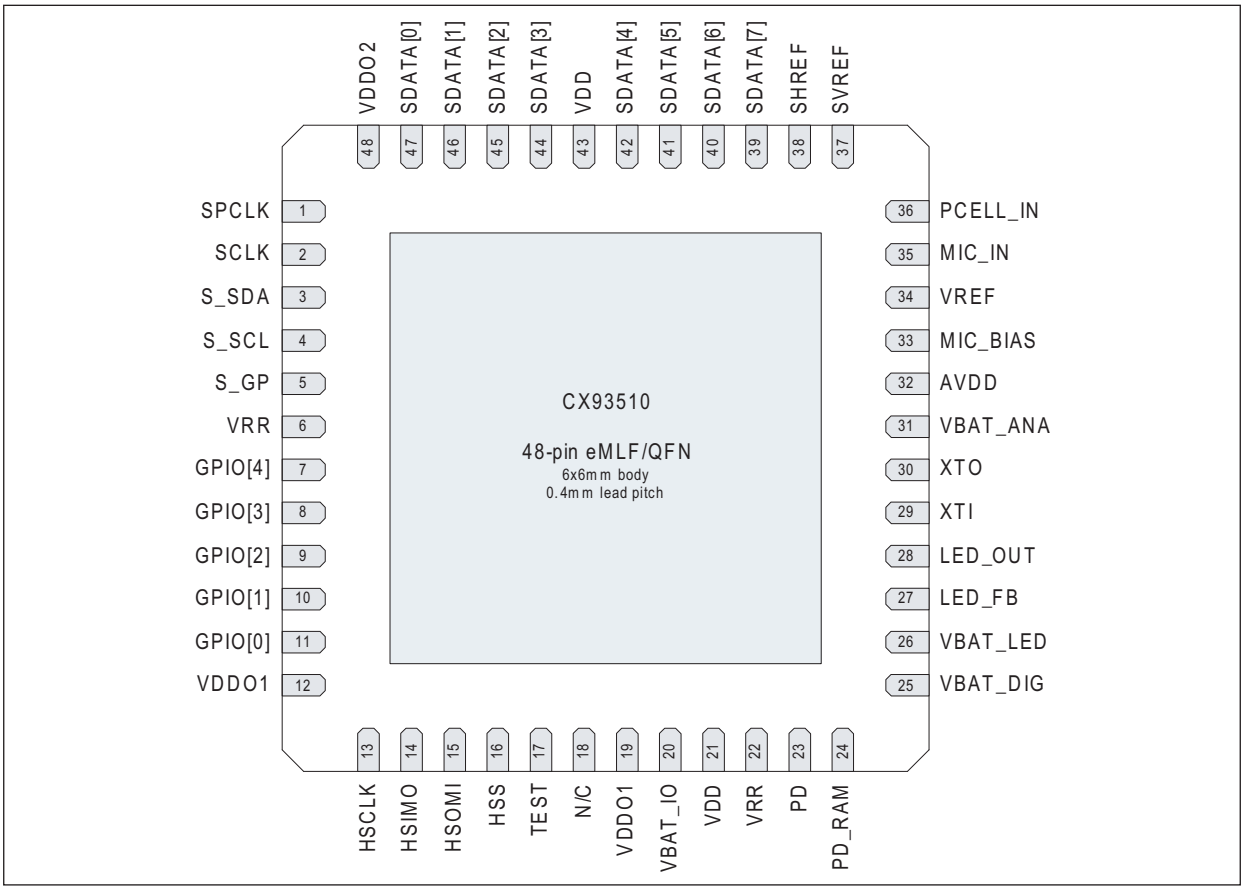
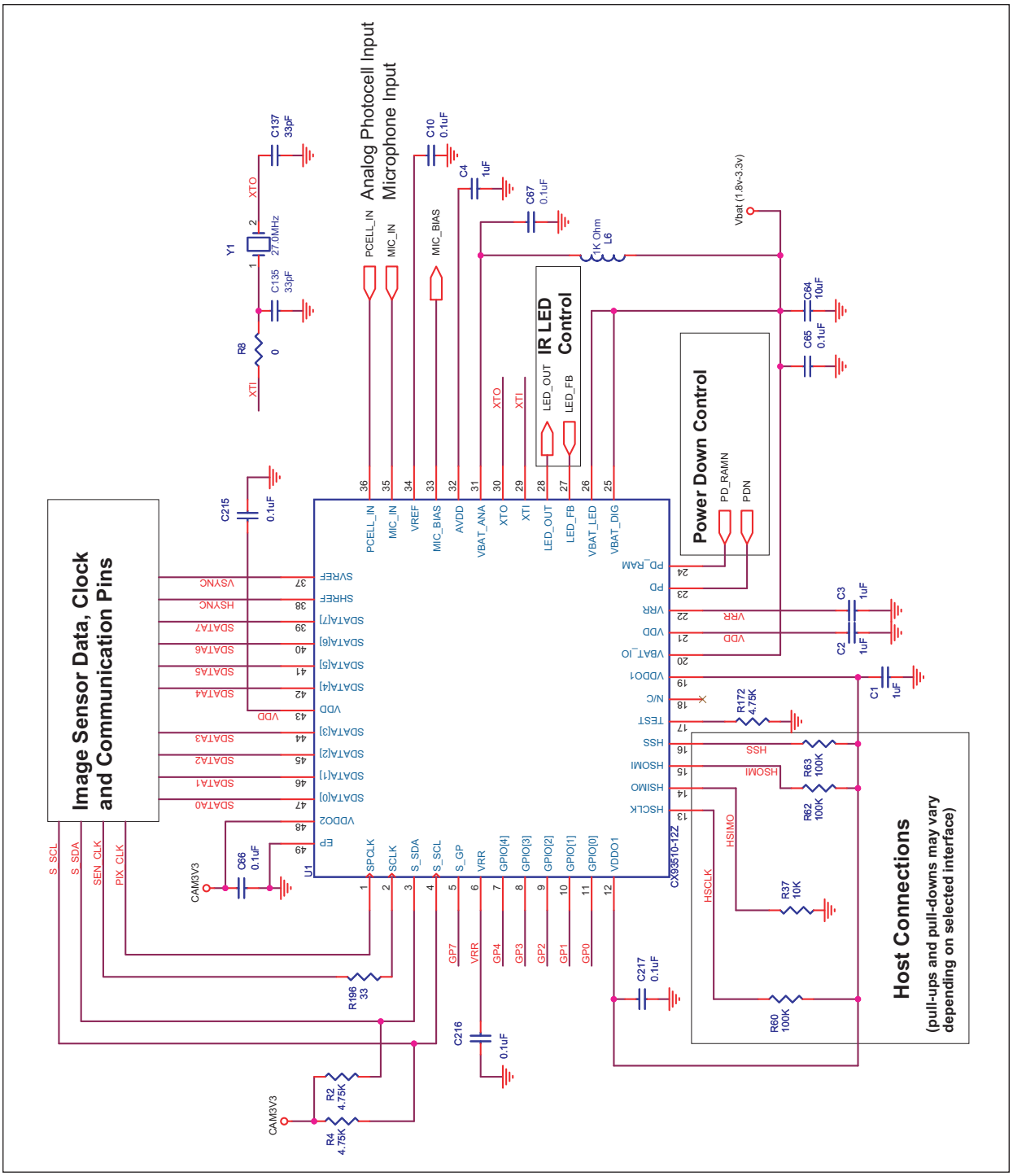


Figure 6. Example Device Interface Connections



3 Electrical Specifications

3.1 Absolute Maximum Ratings

Absolute maximum ratings are listed in Table 2, DC characteristics and operating conditions are listed in Table 3, analog characteristics are listed in Table 4, and timing specifications are listed in Table 5.

Table 2. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Unit
3.3 V supplies	-0.3	4.2	V
1.0 V supply(1)	-0.3	1.2	V
0.8 V supplies(1)	-0.3	1.2	V
Voltage on any signal pin	-0.3	Vdd+ 0.3	V
Storage Temperature	-65	150	°C
FOOTNOTES: (1) With regulators turned off and power supplied externally.			

3.2 DC Characteristics and Operating Conditions

Table 3. DC Characteristics and Operating Conditions *(1 of 2)*

Parameter	Direction	Symbol	Minimum	Typical	Maximum	Unit	Condition
Parameter	Direction	Symbol	Min	Typ	Max	Unit	Condition
Regulator input Voltage	I	VBAT-DIG	1.8	3.3	3.47	V	
Analog Voltage	I	VBAT-ANA	1.8	3.3	3.47	V	
Non-regulated Analog Voltage	I(1)	AVDD	0.95	1.0	1.05	V	LDO_ANA shut down
IO Voltage	I	VBAT-IO	1.8	3.3	3.47	V	
LED Driver Voltage	I	VBAT-LED	1.8	3.3	3.47	V	
Non-regulated Core Voltage	I(1)	VDD	0.76	0.8	0.84	V	LDO_DIG shut down

Table 3. DC Characteristics and Operating Conditions (2 of 2)

Parameter	Direction	Symbol	Minimum	Typical	Maximum	Unit	Condition
Non-regulated RAM Voltage	I(1)	VRR	0.76	0.8	0.84	V	RET_LDO_DIG shut down
Sensor IO Voltage	I	VDDO2	1.8	3.3	3.47	V	
Host IO Voltage	0	VDDO1	1.8	3.3	3.47	V	
Input voltage HIGH		V_{IH}	2.0			V	
Input voltage LOW		V_{IL}			0.8	V	
Output voltage HIGH		V_{OH}	2.4			V	
Output voltage LOW		V_{OL}			0.4	V	
Operating Power (PD=1, PD_RAM=1)				12		mA	Vbat = 3.3 V
Retention Mode Power (PD=0, PD_RAM=1)				2-3		mA	Vbat = 3.3 V
Sleep Power (PD=0, PD_RAM=0)				10		nA	Vbat = 3.3 V
Operating Temperature		T_A	-10		+85	^oC	
Junction Temperature		T_J			+100	^oC	

FOOTNOTES:
(1) These are typically regulated outputs of the device but internal regulators can be disabled and voltage source can optionally be fed externally.

3.3 Analog Characteristics

Table 4. Analog Characteristics

Parameter	Symbol	Maximum	Unit
Parameter	Symbol	Max	Unit
Microphone Input	MIC_IN	1.0	Vp-p
Analog Photocell Input	PCELL_IN	1.0	V
LED Output	LED_OUT	VBAT_LED	V
LED Feedback	LED_FB	1.8(1)	V

FOOTNOTES:
(1) With high gain mode enabled. Typical range is 0-1.0 V.

3.4 Timing Specifications

Table 5. Timing Specifications *(1 of 2)*

Pin Name	I/O	Interface Name {dir}	Input Setup (ns)	Input Hold (ns)	Clock-to-Output (ns)	Clock	Clock Period ([ns)
GPIO							
GPIO[4]	IO	gpio4 [IO] / I_SCL2 [I] / O_SCL2 [O]	50 / 15 / NA	3 / 3 / NA	54 / NA / 15	XTI/2 / XTI / XTI	74.07 / 37.07 / 37.07
GPIO[3]	IO	gpio3 [IO] / I_SDA2 [I] / O_SDA2 [O]	50 / 15	3 / 3 / NA	54 / NA / 15	XTI/2 / XTI / XTI	74.07 / 37.07 / 37.07
GPIO[2]	IO	gpio2 [IO]	50	3	54	XTI/2	74.07
GPIO[1]	IO	gpio1 [IO]	50	3	54	XTI/2	74.07
GPIO[0]	IO	gpio0 [IO]	50	3	54	XTI/2	74.07
Sensor Interface							
SDATA[7]	IO	sdata7 [I]	23	3		spclk	37.037
SDATA[6]	IO	sdata6 [I]	23	3		spclk	37.037
SDATA[5]	IO	sdata5 [I]	23	3		spclk	37.037
SDATA[4]	IO	sdata4 [I]	23	3		spclk	37.037
SDATA[3]	IO	sdata3 [I]	23	3		spclk	37.037
SDATA[2]	IO	sdata2 [I]	23	3		spclk	37.037
SDATA[1]	IO	sdata1 [I]	23	3		spclk	37.037
SDATA[0]	IO	sdata0 [I]	23	3		spclk	37.037
SVREF	IO	svref [I] / gpio5 [IO]	23 / 50	3 / 3	54	spclk / XTI/2	37.037 / 74.07
SHREF	IO	shref [I] / gpio6 [IO]	23 / 50	3 / 3	54	spclk / XTI/2	37.037 /74.07
SPCLK	I	spclk [I]					

Table 5. Timing Specifications (2 of 2)

Pin Name	I/O	Interface Name {dir}	Input Setup (ns)	Input Hold (ns)	Clock-to-Output (ns)	Clock	Clock Period ([ns)
Sensor Interface (continued)							
SCLK	O	sclk [O]					
S_SCL	IO	s_scl [IO]	23		24	XTI	37.037
S_SDA	IO	s_sda [IO]	23		24	XTI	37.037
S_GP	IO	sccb_e [O] / gpio7 [IO]	NA / 47	NA / 3	24 / 54	XTI / hstclk [I] XTI/2 [O]	37.037 / 74.07
Host Interface							
HSCLK	IO	hsclk [I] / scl [I] / txd [O]	NA / 60 / NA	NA / 3 / NA	NA / NA / 54	NA / XTI/2 / XTI/2	NA / 74.07 / 74.07
HSIMO	IO	hsimo [I] / rts [O]	26 / NA	3 / NA	NA / 52	hsclk_inv / XTI/2	37.037 / 74/07
HSOMI	IO	spi_out [O] / sda_out [O] / sda_in [I] / cts [I]	NA / NA / 50 / 50	NA / NA / 3 / 3	10 / 52 / NA / NA	hsclk / XTI/2 / XTI/2 / XTI/2	37.037 / 74.07 / 74.07 / 74.07
HSS	I	hss(auto-detect) [I] / hss (SPI) [I] / rxd [I]	50 / 25 / 50	3 / 3 / 3		XTI/2 / hsclk / XTI/2	74.07 / 37.037 / 74.07

4 Sensor Interface

4.1 Supported Resolutions and Frame/Data Rates

4.1.1 Requirements

Input resolutions	Max 640x480 (VGA), Min 320x240 (QVGA) or any subset in between with multiple 16x16, so 512x256, 352x240 (CIF) etc., are acceptable. If the scaler is enabled, subsets in between the minimum and maximum frame sizes must be multiples of 32x32, therefore limiting the minimum input resolution to 320x256.
Sensor I/F: 8-bit data	Uses 656-compatible embedded SAV/EAV timing codes or separate horizontal and vertical signaling.
Input pixel clock rate	Up to 27 MHz; internal clock rate: 27 MHz

NOTE: The input pixel clock rate must be less than the internal clock rate.

The interface will support cropping of the incoming image so that sensors of other resolutions can be supported. However, internal buffer sizes will constrain the resulting image size to 640x480 or smaller.

4.2 Supported Data Formats

The following input data formats are supported:

4.2.1 YCbCr 4:2:2

Configuration registers will allow swapping of Cr and Cb and/or luminance and chrominance values. Scanning must be in progressive (non-interlaced) mode.

4.2.2 Monochrome (Y Only)

To support monochrome mode, the image sensor interface can be configured to receive only Y values (8bpp). Registers will also allow the design to receive and ignore Cr and Cb values. In either of these monochrome modes, the sensor interface will output only Y (luminance) data.

4.3 Continuous vs. Limited Frame Mode

Configuration registers allow the sensor to capture frames continuously, or capture a limited group of frames (up to 63) and then stop. In Limited Frame mode, the FRAME_NUM register field indicates how many frames to capture before stopping. In continuous mode, the sensor continues to capture frames indefinitely.

In either mode, the FRAME_SKIP register field is used to indicate how many frames to skip between captured frames. For example, if this value is set to 0x02, then every third frame is captured. This allows the user to capture data at 10 fps if the sensor is running at 30 fps.

4.4 Sensor Interface Timing

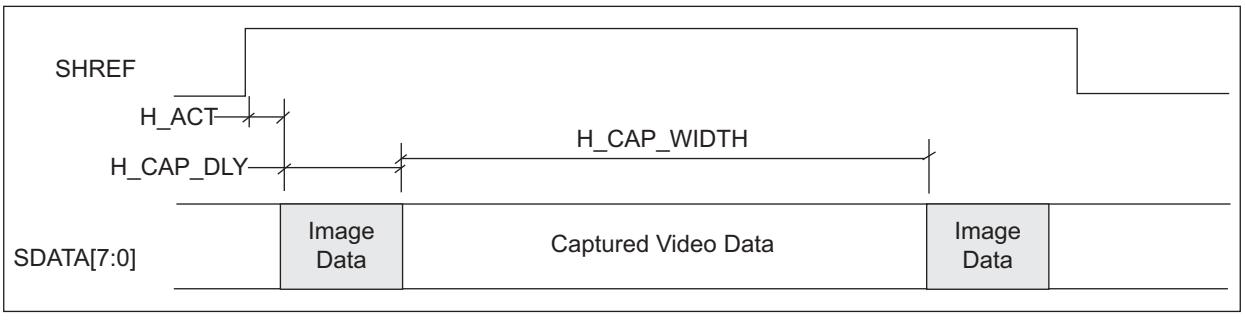
The sensor interface operates in one of two modes. The first mode uses the SHREF and SVREF inputs to determine the timing of image data. The second mode uses packets embedded in the data stream to determine the timing.

NOTE: The CX93510 must see at least three rising edges of SPCLK before it is able to detect the first SVREF edge and capture the first frame.

4.4.1 Discrete Timing Mode

This mode of operation uses the SHREF, SVREF, and SPCLK inputs to determine timing of image data. The SPCLK (pixel clock) input can be up to 27 MHz. The following diagram illustrates how the settings in the H_ACT, H_CAP_DLY and H_CAP_WIDTH registers relate to the capture of data on the SDATA bus:

Figure 7. Discrete Timing Mode



The H_POL register bit will internally invert the SHREF input, allowing the timing to be measured relative to the falling edge of SHREF instead of the rising edge. Data and SHREF are clocked in on the rising edge of SPCLK. Note that the value in the H_ACT register is measured in SCLKs. Values in the H_CAP_DLY and H_CAP_WIDTH registers are in multiples of 8 pixels.

The vertical timing is similar, except that V_ACT, V_CAP_DLY and V_CAP_HEIGHT values are compared to a count of the active-going edges of SHREF instead of rising SCLKs. The value in the V_ACT register is used to count lines between the active edge of SVREF and the start of image data. V_CAP_DLY counts multiples of 8 lines from the start of image data to the start of capture, and V_CAP_HEIGHT * 8 indicates how many lines are captured in each frame.

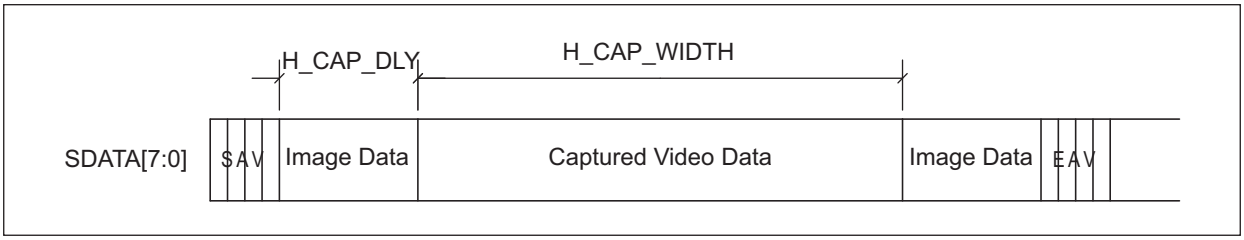
4.5 Embedded Timing Mode

These embedded packets are similar to those used in the ITU-BT656 standard. The H_ACT, H_CAP_DLY, H_CAP_WIDTH, V_ACT, V_CAP_DLY, and V_CAP_WIDTH registers operate as they do in discrete timing mode, with the following differences:

- The H_ACT count begins upon receipt of an SAV code with the H bit cleared after previously receiving an EAV code with the H bit set.
- The V_ACT count begins upon receipt of an SAV code with the V bit cleared after previously receiving an SAV code with the V bit set.
- The vertical counters count SAV packets.
- Receipt of an EAV packet will terminate a line even if the H_CAP_WIDTH count has not been reached. If the H_CAP_WIDTH count has not been reached, image capture will stop, and the EMBD_CDE_ERR register bit will be set.
- Receipt of an SAV or EAV packet with the V bit set will terminate a frame even if the V_CAP_HEIGHT count has not been reached. If the V_CAP_HEIGHT count has not been reached, image capture will stop, and the EMBD_CDE_ERR register bit will be set.

Figure 8 depicts a typical scenario, in which video data is being captured from a window within the active region. Note that in most likely scenarios, H_CAP_DLY will be set to 0, (indicating that data capture will begin immediately after the SAV code), and H_CAP_WIDTH will be set such that the end of capture occurs just before the EAV code.

Figure 8. Typical Video Data Capture



The SAV and EAV packets are four bytes long, with the first three bytes being FF, 00, 00. The fourth byte includes the V bit in bit 5, and the H bit in bit 4. All other bits in the fourth byte (including protection bits) are ignored.

Table 6. SAV and EAV Packets

Condition	656-Compatible Code
Start of frame	8'bXX00XXXX
End of Frame	8'bXX1XXXXX
Start of Line	8'bXX00XXXX
End of Line	8'bXXX1XXXX

4.6 Sensor Clock

The CX93510 provides a 27 MHz clock output (SCLK) to be used as the clock input to the attached image sensor.

Some sensors will use an internal PLL to produce a slower clock for the pixel data interface, SPCLK must be < 27 MHz.

Some sensors include modes in which SPCLK is gated during blanking regions. Some sensors also gate SPCLK if the sensor is scaling the image (which results in a delay between pixels). Because of this behavior, the only logic that can use SPCLK is the logic which interfaces directly with the sensor. The data capture must be synchronized to an internal clock before any data pipeline. Some sensors can also stop SPCLK between lines and/or frames. For this reason, all logic that uses SPCLK must be in a state to accept the start of a new line on the SPCLK edge following the end of the current line. Similarly, the logic must be able to react to the start of a new frame on the SPCLK edge following the end of the current frame. When embedded codes are used for timing, it is assumed that the clock will operate while the codes are sent on the bus.

4.7 LED Control

The sensor interface module can be configured to trigger the LED to turn on and off relative to the end of frame. When this feature is enabled, the LED will be turned off when the last pixel of the image is captured, and the LED will be turned on again after a configurable delay. If en_lfc is used, the LED will remain off after the last frame is captured.

If the frame_skip feature is enabled, the LED will not be toggled at the end of a skipped frame. The LED_ON_DELAY should be programmed to a large enough value to keep the LED off until the start of the next non-skipped frame.

4.8 Sensor Control Interface

An attached image sensor is controlled using an I2C or SCCB interface. Depending on the specifics of the connected sensor, this interface allows for programming of image size, flash, shutter, auto-focus, brightness, contrast, white balance, color correction, and other settings.

The interface includes a FIFO for posting write transactions to the sensor. The FIFO is capable of storing an 8- or 16-bit address and an or 16-bit data value in each entry, and can contain up to 4 entries. The host can add a transaction by writing to the I2C_ADDR and I2C_DATA registers, and the number of entries currently in the FIFO can be determined by reading the I2C_CFG_FIFO_CNT register field. The count in this register will decrement upon completion of the write transaction on the I2C/SCCB bus, and will increment with writes to the I2C_DATA registers. See the register description for more details.

The interface is configured to I2C by default. If SCCB is to be used, the sensor will have to be reset after configuring the interface to SCCB.

The interface allows the host to enqueue up to 4 write transactions by repeatedly writing to the address and write-data registers. Read transactions cannot be pipelined in this manner. The address for write entries can be auto-incremented.

4.8.1 I²C

A master I²C or Serial Camera Control Bus (SCCB) interface is necessary to configure and operate the sensor. This allows a system master to use this interface to program a sensor with an I²C or SCCB interface even if the system master does not interface to I²C or SCCB.

The I²C interface supports sub-address sizes of 0, 1, or 2 bytes. (A sub address size of 0 bytes will produce no sub-address phase during the I²C transaction.) The interface also supports data sizes of 1 or 2 bytes per transaction.

The I²C master interface can be routed to the S_SCL and S_SDA pins, which are on the sensor's power domain, or to the GPIO[4:3] pins, which are on the host's power domain. Configuration registers allow the host to switch between these two sets of pins between I²C transactions, as long as the S_SCL and S_SDA pins are being used for I²C and not SCCB. (in which case only the GPIO[4:3] pins can be used for I²C transactions.)

A data bit rate of 100/400 kHz is supported.

4.8.2 SCCB

The SCCB control bus consists of four signals, as listed in Table 7:

Table 7. SCCB Signals

Signal Name	I/O (Relative to CX93510)	Description
SCCB_E	Output	Chip Select Output; Drive to 0 to initiate transaction or when in suspend mode
SIO_C	Output	Clock signal; similar to I²C clock
SIO_D	Input/Output	Data signal; similar to I²C data
PWDN	Output	Power Down – implemented using GPIO. A configuration bit is used to transition the above signals between operational mode and SCCB suspend mode.

Although the SCCB SIO_C and SIO_C signals are similar to I2C signals, there are some slight differences in the protocol. Details can be found in the SCCB Specification. SCCB_E is not used on all sensors. This functionality can be disabled, which allows this pin to be used as a GPIO[7].

4.9 Miscellaneous Interface Controls

In addition to the data interface and the control interface, some image sensor chips require additional inputs (CX93510 outputs) that can be implemented using GPIO pins. These signals include: Reset, Power Down, and Standby.

In order to initiate a suspend condition on the SCCB bus, the GPIO that is used for the PWDN signal is manipulated separately from the SCCB_SUSPEND configuration bit. Software must drive the PWDN signal active, and then set the PWDN_MODE configuration bit to cause the SCCB interface logic to drive the other SCCB signals low. To end the suspend condition, software must clear the PWDN_MODE bit and then deactivate the PWDN signal.

4.10 Usage Scenarios

This section highlights the requirement for the external uP to stay involved in managing the sensor during capture and in between captured frames to optimize power usage.

4.10.1 Limited/Single Frame Mode

One likely usage scenario is for the host to capture multiple frames at regular intervals. To do this, host firmware will need to follow these steps:

1. Apply Power/Reset to the CX93510 and the CMOS image sensor.
2. Configure the I2C/SCCB interface to allow communication with the image sensor.
3. Configure the image sensor via the I2C/SCCB interface.
4. Configure the CX93510 to receive and process the frame.
5. Initiate capture of a single frame using limited frame mode.
6. Initiate capture of the frame in the image sensor. Depending on the sensor and the hardware configuration, this may require manipulation of a CX93510 GPIO to trigger the picture, or it could require writing to a configuration register in the sensor.
7. Poll the sensor interface configuration register or the frame buffer status register to determine when capture has completed.
8. If necessary, turn off the image sensor.
9. Process the resulting data.
10. After the appropriate amount of time has passed (e.g. 100 ms for 10fps capture), repeat steps 5-9.

Some image sensors can be configured to take a single frame, so step 8 may be unnecessary.

4.10.2 Continuous Frame Mode

Some sensors can also be configured to capture images periodically (i.e. “movie mode”). In this case, a streamlined programming sequence can be followed:

1. Apply Power/Reset to the CX93510 and the CMOS image sensor.
2. Configure the I2C/SCCB interface to allow communication with the image sensor.
3. Configure the image sensor via the I2C/SCCB interface.
4. Configure the CX93510 to receive and process the frames.
5. Initiate capture using continuous frame mode.
6. Initiate capture of frames in the image sensor.
7. Poll the sensor frame buffer status register to determine when capture and compression of frames has completed, and process frames as they are received.
8. Once the required number of frames has been received, the host can shut off the image sensor and turn off the continuous frame mode bit.

5 Video/Image Processing

5.1 Format and Scaling

All video data is input, maintained, and output as 8-b (per component) YCbCr 4:2:2, 8-b Y only for black and white case. Half H and V scaling rate is supported as well as pass through (bypass). The output can range from VGA to a minimum size of 160x128. When using the scaler, the input must be a 32x32 multiple. Typically, this scaler is for re-sizing VGA down to QVGA.

5.2 Bandwidth and Storage Requirements

The Sensor I/F is used as the interface mechanism to transport images (video) into the CX93510 device. The input rate is a function of the sensor resolution (H,V), sensor delivery of data (fps and i/f rate), and the format (b/p).

The video stream output consists of MJPEG where each frame is compressed as an independent still image using the JPEG standard or MJPEG-DPCM (differential mode). The video bandwidth & storage requirements depend on the input, processing, and output.

5.2.1 Storage of Compressed Images

In the PIR Sensor application, images are input & compressed at the rate of up to 10 fps. The images are buffered in an integrated 256 KB/128 KB image frame buffer that can retain the captured compressed images while the rest of the SoC is put to sleep.

The number of images that can be accumulated in the frame buffer is dependant on the JPEG compression obtained. Note that the compression ratio is not a selectable option on the device itself. The compression obtained is highly dependant on the image being captured. Color and image detail can greatly influence the amount of compression achieved. In the case of differential JPEG mode, the amount of motion will also cause the file sizes of the difference frames to vary widely. Table 8 illustrates the number of possible images stored based on arbitrary compression ratios and are shown as an example only.

Table 8. Size and Number of Images for 256 KB Frame Buffer

	VGA				QVGA			
	B&W		Color		B&W		Color	
	KB	Number	KB	Number	KB	Number	KB	Number
Uncomp	300		600		75		150	
10:1	30	8	60	4	7.5	32	15	17
15:1	20	12	40	6	5	51	10	25
20:1	15	17	30	8	3.75	68	7.5	32

5.3 Frame Buffer

5.3.1 Frame Buffer Overview

The CX93510 has a 256 KB Frame Buffer (FB) on chip for the purpose of storing and retrieving compressed JPEG and audio data. The memory is organized as four (4) separate single port SRAMS of 8K words by 64 bits each (non-interleaved). A 128 KB option is also available.

The FB operates in either continuous mode or non-continuous mode. In continuous mode the buffer acts like a circular buffer, automatically resetting the JPEG encoder write address to zero once the end of the buffer has been reached. When operating in non-continuous mode the FB halts all further JPEG encoder accesses when the JPEG encoder write address reaches the end of the buffer.

In order to help save power, when performing a read/write access to the frame buffer, only one of the four SRAMS is active at any one time.

The frame buffer can be optionally configured to contain a 4KB contiguous memory area for audio data storage (latency padding). When not enabled, this 4KB block will revert to video memory use. The audio frame buffer configuration option cannot be changed dynamically.

5.3.2 Frame Buffer Memory Map and Format

The basic format for JPEG frames stored in frame buffer memory is shown in Table 9. Each block of data is preceded by a qword of data status. The data status field contains the following information:

Table 9. Basic JPEG Frame Format

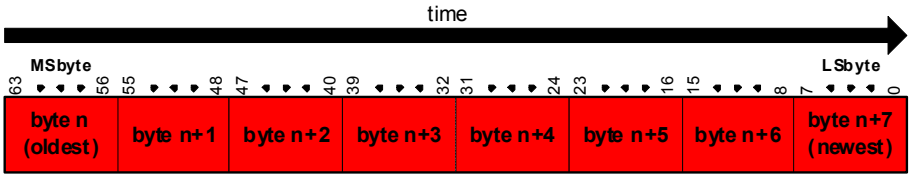
Bits	Name	Description
[63]	ref_frm	Reference Frame Flag – when set to a “1” indicates that the frame that follows is a reference frame.
[62]	cfg_dat	Configuration Data Flag – when set to a “1” indicates that the data that follows is decoder configuration data.
[61]	dif_frm	Difference Frame Flag – when set to a “1” indicates that the frame that follows is a difference frame.
[60]	dir_frm	Direct Frame Flag – when set to a “1” indicates that the frame that follows is a direct frame. Directly encoded frames are generated when differential jpeg mode is disabled.
[59:48]	reserved	Reserved
[47:32]	timestamp	This is a frame timestamp used to sync up video with audio. It is based upon the 8 kHz audio clock. When audio is not enabled the timestamp value will be zero. The timestamp is produced by sampling a free running 8 kHz counter at the end of the image being sent into frame buffer memory by the encoder.
[31:16]	reserved	Reserved
[15:0]	frm_size	Data Size – gives the total data size in bytes. The data size only includes the size of the data. It does not include the status qword or padded zeros components.

The status qword provides data type and data size information. The data type information is used to convey to the JPEG decode block and the host what type of data follows. In the case of the JPEG decode block, this data is used to distinguish between configuration data (used for setting up quantization tables, huffman tables, etc.) and reference frames (used for difference frame encoding). The data size information tells the JPEG decode block and the host the extent of a particular block of data. In the case of the JPEG decode block, the size information is used to determine how much data to fetch from frame buffer memory for decode/configuration. For the host, this information is used to tell it where the next frame begins or how much data it needs to transfer or where the end of all frame data resides.

All JPEG frame data begins on a qword boundary (8 bytes), immediately following the frame status qword. All JPEG frame data, if it doesn't end on a qword boundary, will be padded with zeros such that it ends on a qword boundary. In order for the host to determine where the stored JPEG frame data ends, the qword immediately following the last JPEG frame is written with zeros.

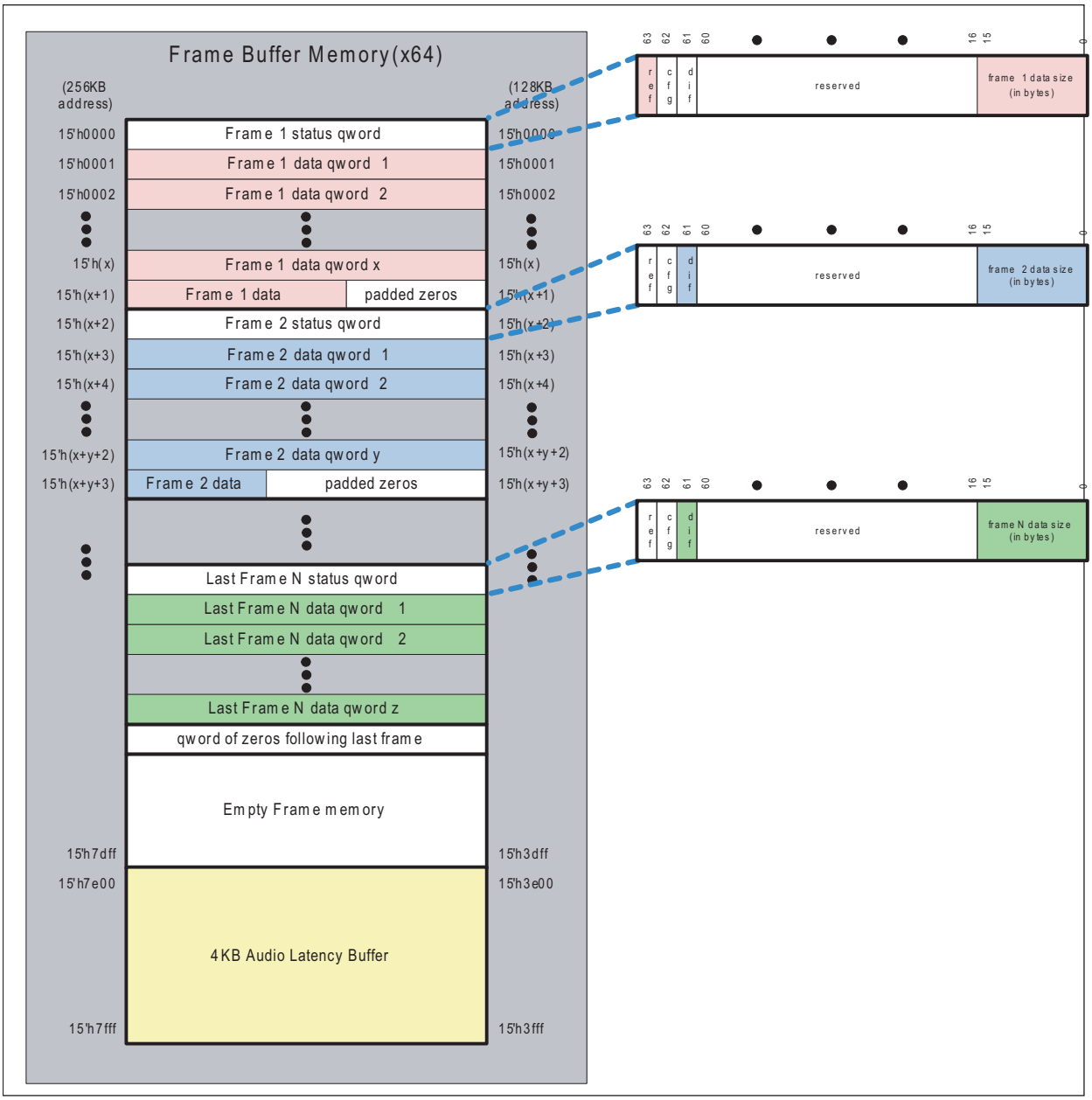
Figure 9 shows the frame buffer loaded with N JPEG frames. The first frame is a reference frame followed by a difference frame. The last frame (N) is a difference frame. The audio latency buffer feature has been enabled.

The ordering of bytes in a frame buffer qword goes from oldest at the most significant byte location (MSbyte) to the newest at the least significant byte location (LSbyte).



Frame Buffer Qword Byte Ordering

Figure 9. Example Frame Buffer Memory Map with Audio Buffer Enabled



5.3.3 Configuration Data

The configuration data that is read from the frame buffer (688 bytes) contains the Huffman and Quantization tables. These 688 bytes follow the 8 byte Status QWORD when the status bit cgf_dat is set and is the first block of data placed in the FB at the start of any capture sequence. This information is required to decode the JPEG frames (direct, reference, or difference frames). This information does not change from one frame to the next, nor does it change between power-ups, B/W or color, VGA or QVGA.

The 688 byte Configuration data structure is as follows:

- SOI (Start of Image, 0xFFD8)
- DQT (Define Quantization Table, 0xFFDB)
- DHT (Define Huffman Table, 0xFFC4) JPEG standard (ISO 10918-1) section K.3.3 are used.
- EOI (End of Image, 0xFFD9)

The JPEG compliant image frames which follow are each preceded with a Status QWORD and structured as follows:

- SOI (Start of Image, 0xFFD8) - 2bytes
- SOF0 (Start of Frame, 0xFFC0) - 19 bytes
- SOS (Start of Scan, 0xFFDA) - 13 bytes
- Encoded Image Data
- EOI (End of Image, 0xFFD9) - 2 bytes

5.3.4 Frame Buffer Full

When the frame buffer is operating in non-continuous buffer mode the frame buffer will halt all further video write operations. The frame buffer full condition is triggered when the JPEG encoder block writes data to the last frame buffer qword address available for JPEG data. It is then up to the host to read out all useful data. After all useful data has been retrieved by the host, the host may restart frame buffer writing a one to the RST_COMP bit in the DIFF_JPG_CTRL register. This will reset the frame buffer write pointer to zero and clear the compression encoder/decoder blocks, so as to prepare a new stream of video data.

The frame buffer has no mechanism to indicate when it is full with video data while operating in continuous mode. As such, the host processor must manage the frame buffer carefully in order to avoid data corruption issues caused by the compression block overwriting existing data that has not been read out by the host yet. The host read must be fast enough to prevent this type of situation.

The 4KB audio frame buffer, if enabled, will produce a full indicator. The host microprocessor will read audio buffer waterline indicators that tell it when it is time to transfer portions of audio data out of frame buffer memory. If the host does not read out the data in a timely manner, the frame buffer will fill up but will not overwrite the current audio data. The AUD_BUF_STAT register will indicate a full condition. This condition can be reset by toggling the “Audio Enable” bit in the HWR_FB_EN register.

6 Image Compression

6.1 JPEG Controller

After the image data from the CMOS sensor passes through the scaler, it proceeds to the JPEG controller for compression. The incoming samples contain Luma (Y) and Chrominance (Cb, Cr) data. The controller supports 4:2:2 format only and up to 300 Kpixel frame size. The input samples are 8 bits wide, originating from the sensor block or scaling/processing block that is fed by the sensor output. The JPEG controller needs to store the color space samples (Luma only if in monochrome mode) after reordering until each block of samples is assembled and ready to be read by the encoder.

After the data has been encoded, it passes to the frame buffer, where it can be read by the host. When the frame buffer is full in non-continuous mode, the JPEG controller stops. The frame buffer indicates to host that frame buffer is full, via register bit FRM_BUF_FULL. The host must then assert register bit RST_COMP.

6.2 MJPEG-DPCM

MJPEG-DPCM is a low bit rate motion JPEG using differential encoding.

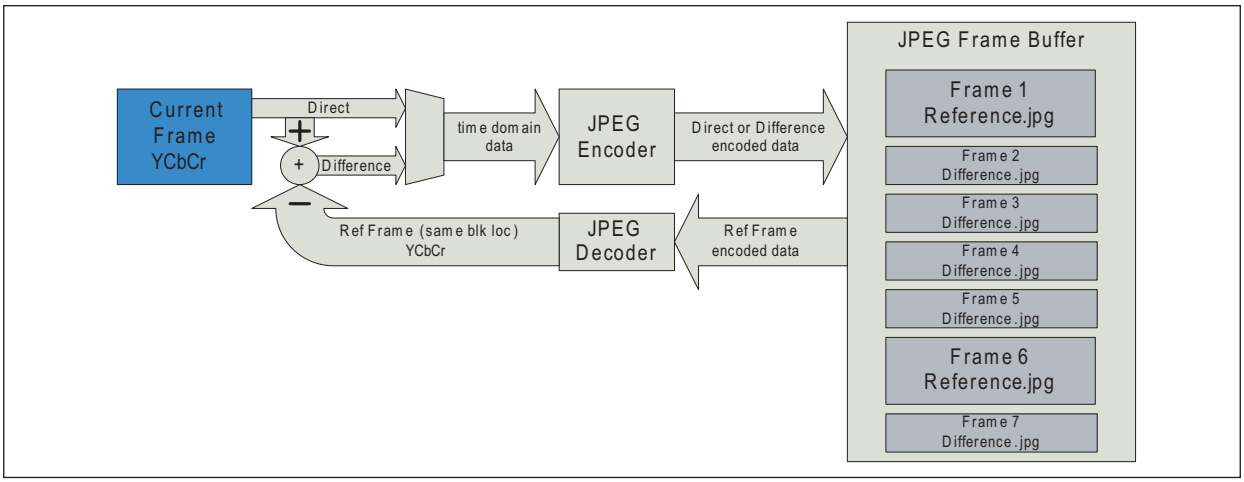
When enabled, Differential JPEG mode allows for additional compression savings due to differencing frames. It is analogous to MPEG (I/P baseline mode) motion compensation without motion estimation. The difference between frames (the current and a reference) is transformed and quantized.

The reference frame is encoded directly, while the current difference frames are encoded as a relative difference in the time domain to the reference frame. This set of images can be referred to as a Group of Pictures (GOP). Typically, the first frame contains background imagery that is redundant to successive frame captures and each is optimally compressed using this method. Any frame may be allowed to be designated as the reference frame.

The REF_FREQ register controls how many frames are in each GOP. This is used to decode a reference frame versus differential frames.

As shown in Figure 10, the reference frame is directly encoded and stored in the frame buffer. While the reference frame data is output to the frame buffer control, the ref_frame signal indicates that all output data from JPEG controller is for reference frame. For differential frames, the JPEG controller requests reference frame data to be read and decoded by the frame buffer state machine. The difference between incoming pixel data and decoded reference pixel data is calculated. The same reference frame is used to calculate the difference for all other frames in the same GOP. The JPEG decoder reverses the encode process decoding the headers, and Huffman codes and performing the inverse QT. It is only used to reconstruct the first reference frame. Subsequent encoded difference data does not pass through the decoder.

Figure 10. MJPEG Differential Encoding Algorithm



6.3 Reconstruction of Frame Data

The CX93510 can output three different types of frames, as listed below. These frames can be generally decoded by any standard JPEG decoder with proper configuration data added:

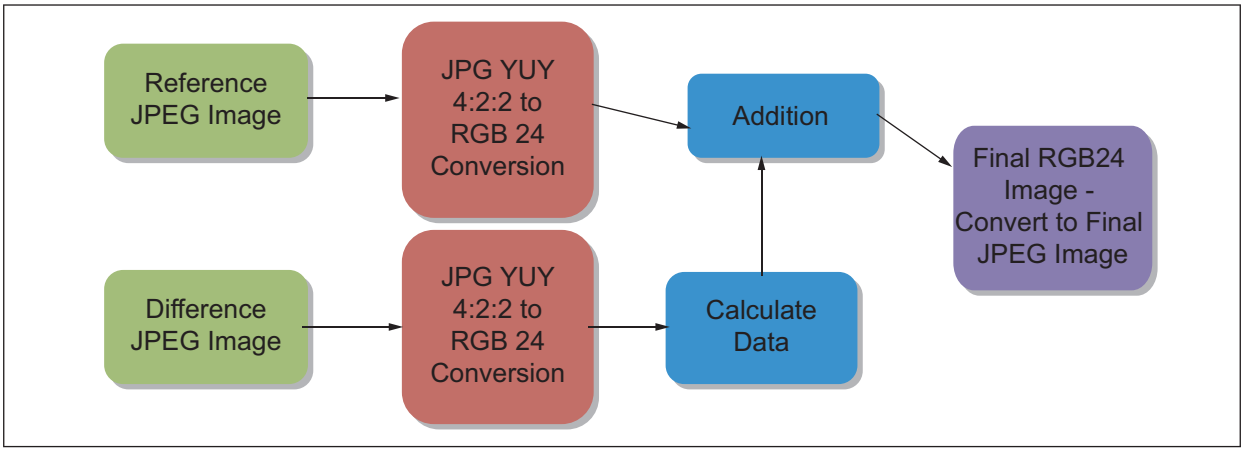
- Direct Frame
- Reference Frame
- Difference Frame

The Direct frame is an individual JPEG frame that does not have any references or differences attached to it. This is obtained when bit 0 of the DIFF_JPEG_CTRL register is set to 0. Every frame received in the frame buffer represents a full JPEG image frame.

When bit 0 of the DIFF_JPEG_CTRL register is set to 1 (default), differential mode is enabled. A Reference frame is initially produced, followed by a series of difference frames. The difference images contain only the difference between the current frame capture and the reference frame.

Figure 11 illustrates the differential image reconstruction process:

Figure 11. Differential Image Reconstruction Process Block Diagram



6.4 Mathematical Equations

Below are the mathematical equations that are used to calculate final image values in RGB 8:8:8 or RGB24 color space. The example below represents the calculations for the 'RED' color component. Similar calculations will be performed on 'Green' and 'Blue' components.

RFinalDiff = RRef + RDelta;

RDelta = clip(round(RDiff) - 128) * 2;

Now add RRef and RDelta values with proper rounding and clipping between 0 & 255 to obtain the final R value.

RFinalDiff = clip(round (RRef + RDelta));

6.5 Differences Between Reference and Difference Frames

Table 10 shows the differences between the Configuration and Image frames as received from the CX93510.

Table 10. Differences Between Reference and Difference Frames

Standard JPEG Frame	Config Data	Image Frame
SOI (Start of Image, 0xFFD8)	Yes	Yes
APPn (Application Marker, 0xFFEn) – **optional**	No	No
DRI (Define Restart Interval, 0xFFDD) – **optional**	No	No
DQT (Define Quantization Table, 0xFFDB)	Yes	No
DHT (Define Huffman Table, 0xFFC4)	Yes	No
SOF0 (Start of Frame, 0xFFC0)	N/A	Yes
SOS (Start of Scan, 0xFFDA)	N/A	Yes
Encoded Image Data	N/A	Yes
RSTn (Restart count, 0xFFDn) – **optional**	No	No
EOI (End of Image, 0xFFD9)	Yes	Yes

7 μP and Miscellaneous Interfaces

7.1 μP Interface

This interface is slaved to the external microprocessor. It can operate in 1 of 3 modes using the four interface pins: HSCLK, HSIMO, HSOMI, and HSS. The mode is automatically detected via the protocol of the first transaction at this slave input interface.

At power-up, a protocol differentiation detector identifies the protocol being used by listening to incoming stream on the four interface pins. A sequence unique to each of the three interfaces is used to identify the protocol being used. Once identified, select bits in the SLAVE_SEL_CTRL control register are automatically set to identify the interface type and the pins are enabled accordingly. The combination of the bit selects are as follows:

00 = I^2C

01 = SPI

10 = UART

11= No Slave Selected

These bits are read only and cannot be altered by the host.

The pin muxing is shown in Table 11:

Table 11. Pin Muxing

Pin Name	SPI (Direction)	UART (Direction)	I^2C (Direction)
HSCLK	HSCLK (I)	TXD (O)	SCL (I)
HSIMO	HSIMO (I)	RTS (O)	Alt_slave_ID[(1)]
HSOMI	HSOMI (O)	CTS (I)	SDA (I/O)
HSS	HSS (I)	RXD (I)	1'b1

FOOTNOTES:

(1) For I^2C operation, HSIMO is used for selecting between default and alternate slave IDs. The default value of this pin is 0, which results in the default slave ID of 0x44. When this pin is driven high, the slave ID used is 0x46.

The following are unique sequences for each of the three interfaces:

- A new transaction in SPI always starts with HSS being asserted to 0 from the SPI master at the negative edge of HSCLK (negative edge or positive edge is programmable).
- A new transaction starts in I^2C with SDA being pulled low during the high period of SCL.
- A new transaction starts in UART with RXD being asserted to 0 for 1 baud rate duration marking the START bit. The UART is pre-determined to work at 1 Mbps, and hence the duration of the START bit is 1 µs.

The following algorithm is used to determine the interface used:

1. Wait for a set time (~1ms) from reset allowing the signals to settle.
2. After timer has expired, if SDA goes low when SCL is high, then it is an I^2C protocol. Move to state, I^2C. Set the control register bits to indicate that the I^2C interface is being used.
3. After timer has expired, if HSS/RXD pin is observed to go low, it could either be UART or SPI. Move to state, UART_SPI.
4. Create a 3 µs window (default value) in the host clock domain. SPI can typically work from 1 MHz to 27 MHz, and UART baud rate is fixed at 1 Mbps. Count the edges of HSCLK during 3 µs window.
5. At the end of the window, if clock count is > 2, then it is SPI. Set the appropriate control bits.
6. If at the end of the window, clock count == 0, then it is UART. Set the appropriate control bits.

For SPI and UART, during the detection process, the first transaction will be lost since the interface has not yet been properly configured. It is required that the host re-transmit the first command after power-up/wake-up from Sleep mode. However, this is not necessary if the I^2C interface is used.

Initial conditions required for auto-detect to work:

To prevent a false interface detection, the host must drive the inputs to known states prior to bringing the device out of reset. Following requirements for each of the interfaces:

SPI:

HSCLK: Host should drive this to 1 or 0 based on CPOL,CPHA = 11 or 00 respectively.

HSS: Host should drive this to 1.

HSIMO: No requirement.

HSOMI: Need a weak external pull-up if CPOL,CPHA = 11 to prevent confusing with an I^2C start condition scenario. No requirement if CPOL,CPHA = 00.

I^2C:

HSCLK: Host should drive this to 1.

HSIMO: Need a weak pull-up or pull-down to select the appropriate slave ID.

HSOMI: Host should drive this to 1.

HSS: Need an external weak pull-up.

UART:

HSCLK: No requirement

HSIMO: No requirement if host does not have flow control enabled at power-up/auto-detect. Or else, an external pull-down is required*.

HSOMI: No requirement

HSS: Host should drive this to 1.

*: HSIMO is multiplexed as RTS for the UART interface. If the host is flow control enabled, and listens on the RTS line to even send the first transaction, then this will lead to a catch-22 situation. Either the host's flow control should be disabled during auto-detect, or a weak pull-down should be attached on HSIMO.

7.1.1 SPI Timing

The SPI commands are sent by the master on the HSIMO pin at the rising edge of HSCLK. The commands are 3 bytes wide.

Write: 10000000---xxxxxxxx---00000000---yyyyyyyy (The 8-bits following the 1st byte is the address). The following 8-bits will be 0s, and then following that (yyyyyyyy) will be the data to be written.

Read: 00000000---xxxxxxxx---00000000 (The 8-bits following the 1st byte is the address).

Once the read is issued, the data will be placed on the HSOMI line at the subsequent falling edge of HSCLK.

In terms of clock polarity and phase, CPOL = 1, CPHA = 1 or CPOL= 0, CPHA =0 are supported.

Figure 12. SPI Read Timing

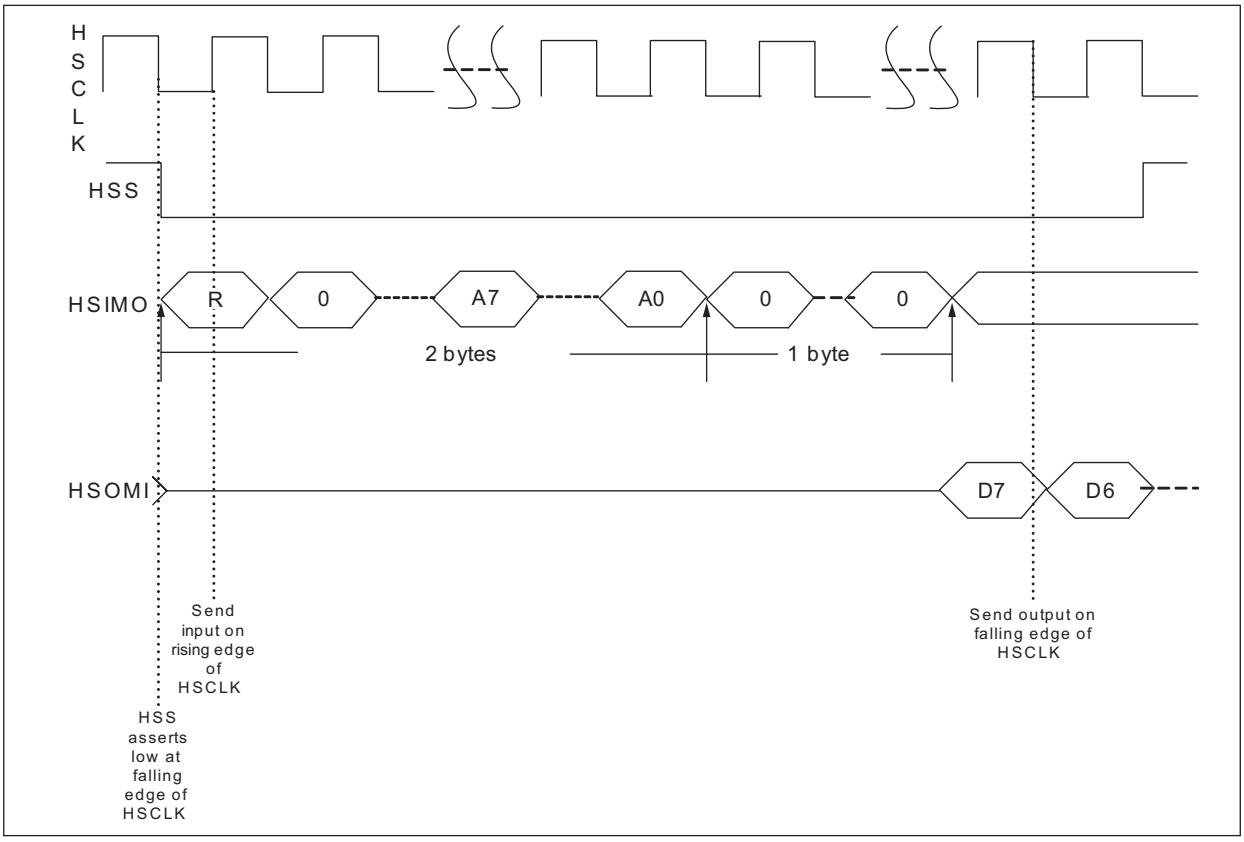


Figure 13. SPI Write Timing

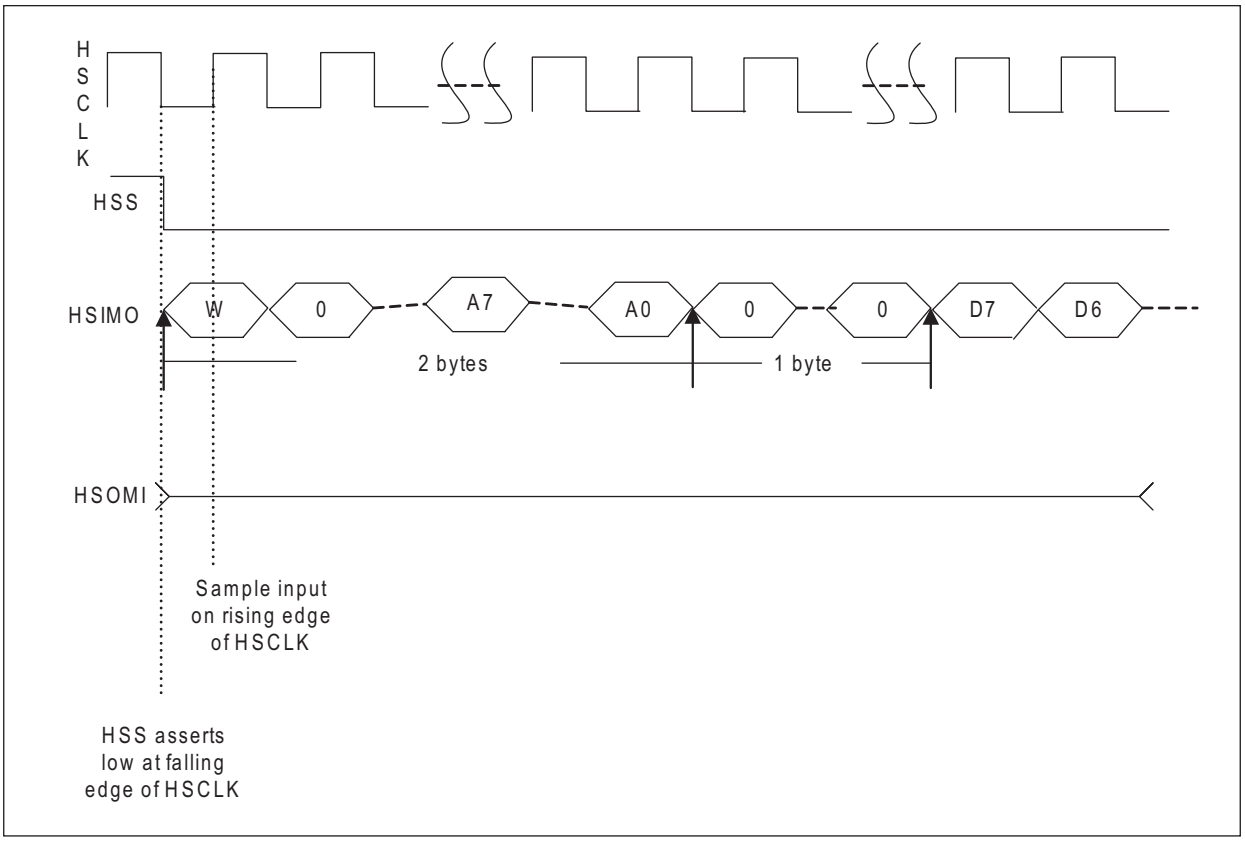


Figure 14. SPI Burst Read

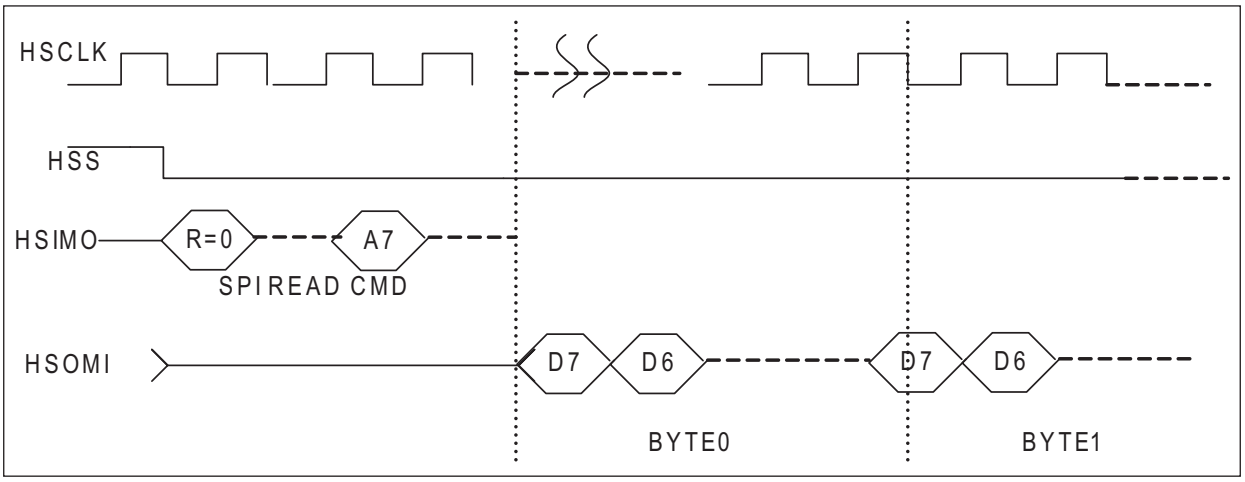
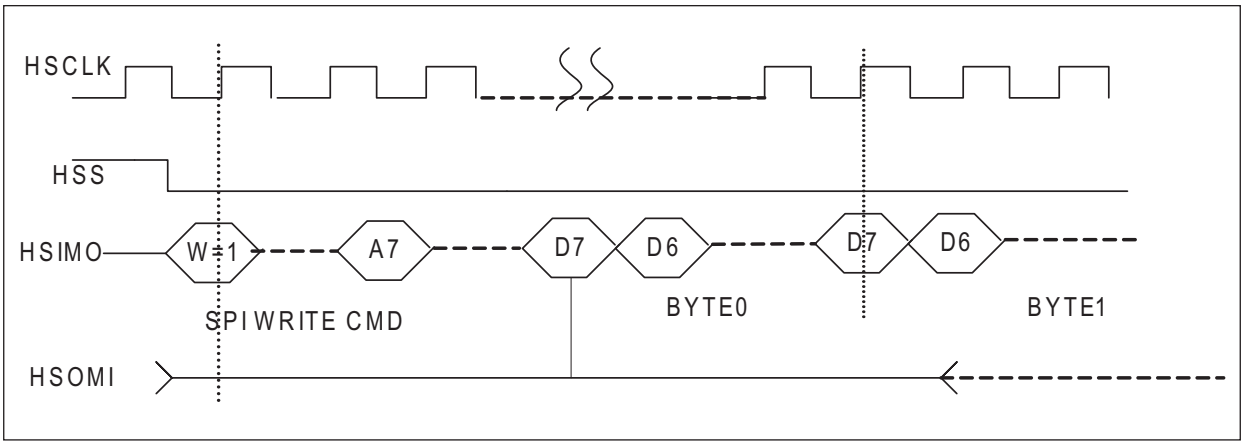


Figure 15. SPI Burst Write



Note that although not explicitly shown in the burst diagrams, the byte of 0s following the address is still required.

As long as HSS is asserted, the transactions are valid. If HSS stays asserted beyond 1 byte of data, the next data byte will be written to the next address location in an auto increment mode. The same is true in the case of reads. As long as HSS is asserted, registers will be read in auto increment mode.

The exception is when read is addressed to 0xCC; this will then be treated as a request to fetch FB data. Continuous reads will fetch successive FB data. For writes to FB, address 0x56 will be written to byte after byte. The host interface module will be responsible to append 8-bytes worth of data and create a request to write the quad word into the FB.

NOTE: The SPI slave state machine runs on the 13.5MHz host interface clock. The SPI master needs to keep hss de-asserted for a length of more than 1 host clock period (>74 ns) in between transactions for the slave to be able to detect an end of transaction.

7.1.2 I²C Timing

The I²C interface can operate at 400 kHz or 1 MHz. The HSIMO pin is used to select between the default and the alternate slave IDs of 0x44 and 0x46, respectively. This interface operates with 7-bit slave address, 8-bit addressing mode, no clock stretching, and no extended mode operations. Write and read transactions are treated as burst mode transactions until a STOP is issued by the master. Register reads/ writes in burst mode are in auto-increment mode. A read to register 0xCC will be treated as a host request to access the FB. Continuous reads will fetch successive FB data.

I²C timing diagrams are shown in Figures 16 through 17.

Figure 16. Write Transaction in I²C

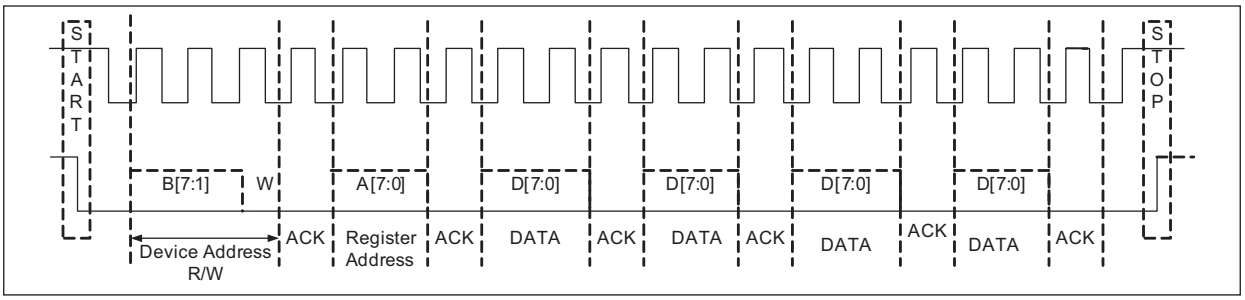
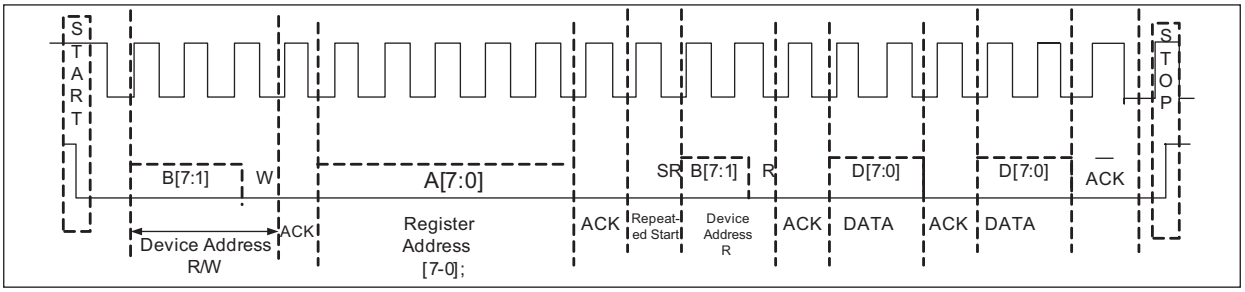


Figure 17. Read Transaction in I²C



7.1.3 UART Timing and Protocol

The UART interface is a slave device which will work at a fixed baud rate of 1Mbps (can handle baud rates in the range 0.99 MHz to 1.04 MHz only). This is a 4-wire interface with the pins, RXD (receive), TXD (transmit), CTS (Clear to send) and RTS (Request to Send). Unlike SPI and I^2C, all command, data and address bytes are LSB to MSB (bit 0 through bit 7) order on RXD. The read data on TXD also will be in the same order, LSB first through MSB in accordance to the RS-232 standard. No parity bit will be checked on RXD or will be sent on TXD.

Unlike I^2C, UART does not come with a standard that defines a set of command streams for read/write, etc. The protocol used in the CX93510 is as follows:

Byte0 received on RXD: bit[7:6] = R/W'/stop,

bit[5:0] = No. of transactions.= {0 means 1 transaction {1 means 2 transactions,…

Bit[7:6] = 00, for BRST_READ (controlled burst)

= 01, for WRITE

= 10, for READ (uncontrolled burst)

= 11, for stopping an ongoing uncontrolled burst read transaction (READ_STOP)

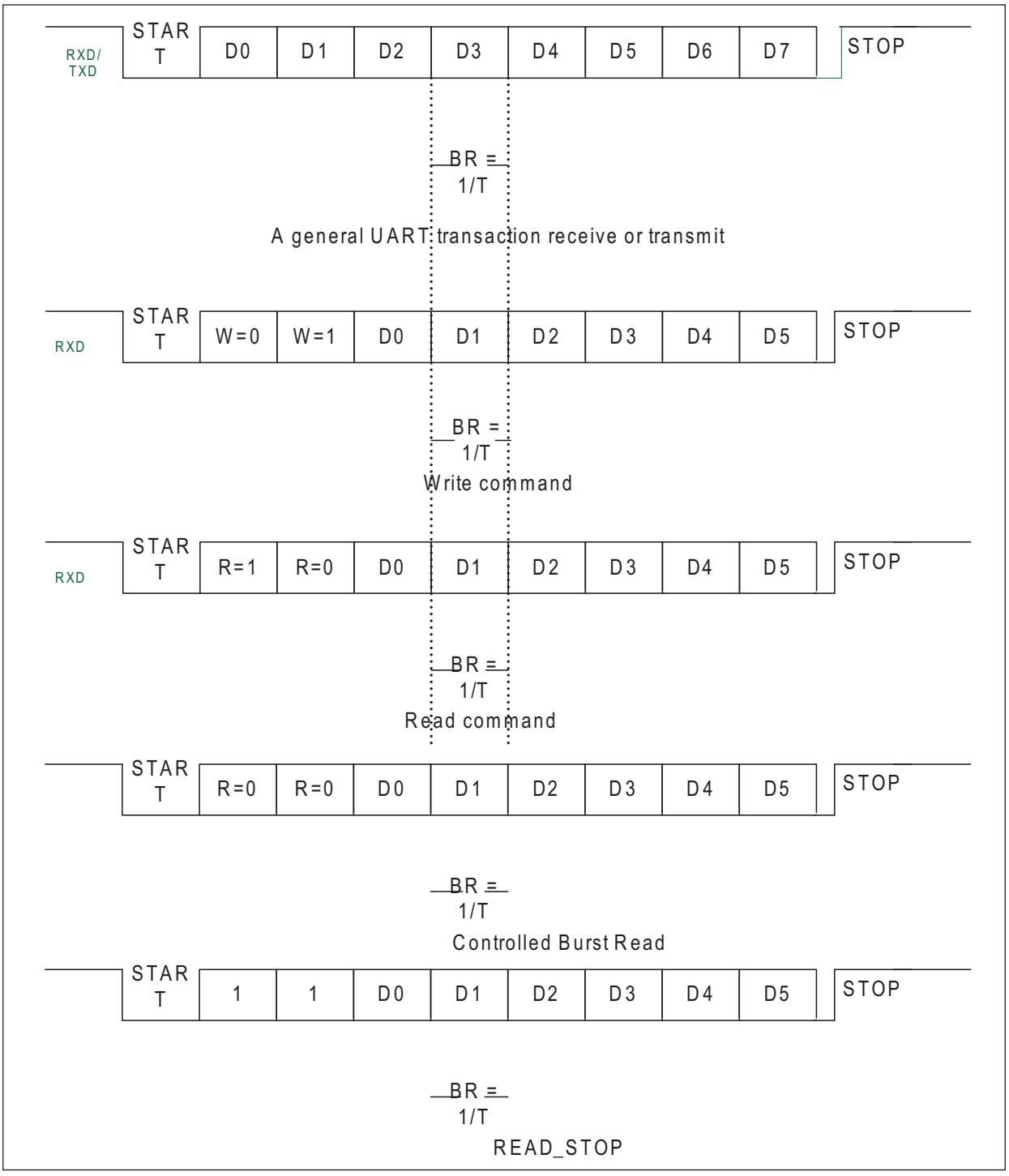
Byte1 received on RXD is the start address of a register

In case of write, Byte2 received on RXD is the data to be written to the register, and so on.

In case of a read, TXD will transmit the read data after Byte1 is received.

If more than one transaction is mentioned, continuous registers will be accessed to read/write in auto-increment mode (see Figure 18).

Figure 18. A Typical UART Transaction (T = 1MHz)



A typical UART transaction. Here T = 1 MHz

NOTE: The gap shown above between D0 and STOP is only to emphasize that a high is used to mark a STOP

BRST_READ (cmd = 00) is different from READ (cmd = 10) in that the controlled burst will have a count of the number of bytes that the host is requesting embedded in byte0 in the 6 remaining bits following the command.

As shown in the diagram above, D5 through D0 holds the count of the number bytes requested by the host. If count = 0, then slave will transmit 1 byte. If count = 31, then slave will transmit 32 bytes.

For READ, the command is also treated as a burst transaction but the slave does not keep track of the number of bytes being transmitted. It is the responsibility of the host to issue a READ_STOP command to tell the slave to stop sending more bytes.

Note that the no. of transactions will be interpreted [7:2] although it is received in [2:7] order.

For example, the WRITE command will look like:

Cmd byte = {START,0,1,0,0,0,0,0,1,STOP}. This would be interpreted as no. of transactions = 32.

Start addr = {START,0,0,0,1,0,0,1,1,STOP}. Addr = 0xC8

Data Byte = {START,0,1,0,0,1,0,0,0,STOP}. Data = 0x12

And so on…

Every transaction is 10 bits long, starting with a START bit, and ending with a STOP bit. The middle 8 bits is the actual byte. The duration of each bit will be 1 μs long.

RTS and CTS are both active low signals. They both are flow control signals which are asserted by the slave and master respectively.

Data will be transmitted on the TXD line, as long as CTS is asserted low. In other words, CTS asserted low enables data transmission. However, if CTS is de-asserted in the middle of a byte, then that byte will be transmitted and the subsequent bytes will be stopped. When CTS is asserted again, the subsequent bytes to be transmitted will follow.

RTS will be asserted high by the UART slave in an event when it cannot accept anymore commands from the master, or cannot accept anymore data. As long as the write path is clear, RTS will be held low allowing incoming data on the RXD line.

In the event that the host/master stops the slave TX data through CTS flow control, no further incoming command on the RXD line will be honored until CTS is asserted low again and all the bytes are transmitted from the slave.

By default, it is assumed that flow control is enabled on the host side, and will have RTS asserted in order to receive the first command. Unless, bit[2] of SLAVE_SELECT register (0x50) is enabled (set to 1), it will be assumed that no flow control is enabled and the CTS input will not be looked at in order to transmit data.

Burst transactions are back to back transactions with a START bit immediately following the STOP bit from the previous byte.

The write burst transactions are as described above defined by the number of transactions embedded in bits [5:0] in Byte0.

The burst read transactions are also similar, except that following a read command and the start address, the following bytes will be on the TXD line byte after byte back to back.

If it were regular register access in burst mode, registers will be accessed in auto-increment mode. However, a read to register 0xCC will be interpreted as a data fetch from the FB.

NOTES:

1. If the UART master issues a burst read (BRST_READ) command (controlled), the slave will honor any incoming receive data as a command only when all the data bytes are read out.
2. READ_STOP will be honored only for READ (uncontrolled burst reads).
3. All bytes on RXD following a WRITE command will be interpreted as data bytes for as long as the number of bytes mentioned in the command. Any new byte after this will be treated as a new command.
4. The UART slave does not support any error recovery mechanism currently.

Note on UART auto-detect: The first transaction for the auto-detect should be only the command byte when the mode will be detected. The following address and data bytes should not be transmitted to the slave. The second byte following the auto-detect dummy byte that the slave will receive will be decoded as the actual command.

7.2 GPIO

Five General Purpose I/O are available. Example uses are: Auto focus, auto zoom, camera activity LED, Flash/shutter, Capture Trigger. GPIO[4:3] are shared pins with an I^2C interface to the Photocell Sensor. An additional 3 GPIO (GPIO[7:5]) are shared with the Sensor I/F pins {S_GP, SHREF, SVREF}.

The input pins are read via the GPIN register. The output pins are controlled by writing to the output data register GPOUT and GPOE (output enable controls).

7.3 ADPCM

The CX93510 captures 16-bit, 8 kHz mono mic data and compresses it into 4-bit IMA ADPCM (Interactive Multimedia Association - Adaptive Differential Pulse Code Modulation). The data/file format is compatible with standard multimedia players for listening on a PC. Any header data is supplied by the external μP or at the uplink receiving end. A non standard mode of 2 bit @ 8 kHz is also available.

7.3.1 Block Size

The continuous mic waveform is segmented into blocks whereby the variation of the waveform samples are predicted within each block. Each block consists of a preamble(4 bytes) and a series of coded ADPCM 4-b nibbles or 2-b codes. The first 2-bytes of the preamble is the first PCM sample of each block in little endian. Each nibble represents both a sign/magnitude number and a table index.

The block size includes the preamble and the encoded bits.

Example:

If the number of samples is programmed to 505, then the first sample forms 2-bytes of the preamble, and the remaining 504 samples are encoded to 252 bytes in the 4-b mode or 126 bytes in the 2-b mode.

Block size in 4-b = 4-bytes preamble + 252 encoded bytes = 256 bytes

Block size in 2-b = 4-bytes preamble + 126 encoded bytes = 130 bytes

It is required that the samples/block be programmed as an odd number so that samples/block – 1 is divisible by 2 or 4 based on 4-b ADPCM or 2-b ADPCM respectively.

There are 10 bits available to program the samples per block in registers ADPCM_N_0 and ADPCM_N_1.

7.3.2 Monaural Preamble

As an example, MSFT IMA ADPCM (MSFT media file like.wav that is encoded with IMA ADPCM) would have the following preamble:

Byte	
0-1	first sample in each block, in little-endian format, or X(0)
2	step index calculated using the first sample, or si(1)
3	unknown, usually 0 (reserved)

The remaining bytes in the chunk are the IMA nibbles. Each byte is decoded bottom nibble first, then top nibble (little-endian).

7.4 Register Interface

The Host Interface has direct memory-mapped r/w access to the CX93510 register set.

The registers can be accessed in single access mode or in burst mode. When operating in burst mode, the addresses will be auto incremented after every byte of write or read.

7.5 Host Frame Buffer Interface

7.5.1 Host Serial Interface Bandwidth

The I^2C is the slowest of the 3 serial interfaces, and works up to a rate of 1Mb/s.

SCL runs at 1 MHz @ 1 bit per clock. This translates to 1 µs x (8 bits x 30KB) = 240 ms for one frame of compressed VGA frame at a 20:1 compression ratio.

The UART runs at 1 MHz @ 1 bit per clock. This translates to 240 ms for the same data as above.

The SPI, when running at 27 MHz @ 1 bit per clock, translates to 8.9 ms for the same data as above.

Table 12 details the output streaming rate for each of the serial interfaces at various compression ratios. Note that the compression ratio is not a selectable option on the device itself. The compression obtained is highly dependant on the image being captured. Color and image detail can greatly influence the amount of compression achieved. In the case of differential JPEG mode, the amount of motion will also cause the file sizes of the difference frames to vary widely.

Table 12. Host Serial Interface Streaming Rate for Various Compression Modes

Ratio	VGA								QVGA							
	B&W				Color				B&W				Color			
	Size (KB)	Time (ms)			Size (KB)	Time (ms)			Size (KB)	Time (ms)			Size (KB)	Time (ms)		
		I^2C	UART	SPI		I^2C	UART	SPI		I^2C	UART	SPI		I^2C	UART	SPI
10:1	30	240	240	8.9	60	480	480	17.8	7.5	60	60	2.25	15	120	120	4.5
15:1	20	160	160	5.9	40	320	320	11.8	5	40	40	1.5	10	80	80	3
20:1	15	120	120	4.45	30	240	240	8.9	3.75	30	30	1.12	7.5	60	60	2.25

GENERAL NOTES:
1. The above values are for the worst case compression rates for color, and black and white for various resolutions.

7.5.2 Frame Buffer Access

The FB data interface is 64 bits wide (quad-word/QWORD) and accessed internally through an 18-bit address. The host only accesses the upper 15 bits to obtain bytes on a QWORD boundary. When data is requested from the FB, the FB will put forth 64 bits worth of data to the internal host interface block, which will unpack the data into a 16x8 FIFO. At any given point, the host requests either audio or video data, by writing to the command registers, the frame buffer address (not used for audio), and setting the appropriate command bits in FB_ADDR_2[4:3].

The FB commands for FB_ADDR_2[4:3] are as follows:

= 01 (for audio fetch)

= 10 (for video fetch)

= 11 (for host write)

= 00 (No transaction)

If an error condition occurs, it will be registered in the ERR_STAT register. The external host will read this register and service the error accordingly.

The audio portion in the FB starts at address 15'b 0x7E00 for 256KB option and at 15'b 0x3E00 for the 128KB option. All other locations are used for compressed video and video related data. After the CX93510 has been configured as desired and the EN_LFC or EN_CFC bit has been set to begin frame captures, the host requests to access frame buffer data (audio or video) in the following manner:

1. Write the starting address to registers FB_ADDR 0 & 1 (note that only address bits 14:0 are written in FB_ADDR_ 0 and 1 [not used for audio])
2. Write the desired transaction value to bits [4:3] of FB_ADDR_2
3. Wait for the prefetch_done bit in FB_ADDR_2 =1. This will occur after the data has been fetched and written to the FIFO. This bit need only be monitored after the first read request when using the SPI host interface due to its higher speed. The I^2C and UART interfaces are slower and do not require the use of this bit.
4. Begin to read FB data from register 0xCC. Continue to read from this register until all bytes desired are read. The host does not have to increment the address in FB_ADDR 0 & 1. The FB address will automatically increment internally as reading continues. Reading can be in burst or non-burst format.

Note, for video requests in non-continuous mode (CONT_MODE=0), the FB will reach the end of the FB addressing and initiate the FRM_BUF_FULL bit in register 0x20. After reading the desired data, the host must set the RST_COMP bit in the same register to continue reading data starting back at frame buffer address 0x0000. In continuous mode (CONT_MODE=1), the host continues to read data without interruption. The FB address will automatically wrap around internally. This will occur whether in Limited Capture Mode (EN_LFC=1) or Continuous Capture Mode (EN_CFC=1).

The read requests to access the FB data from the external host are byte addressable. However, the host interface block has access to the FB on a quad word boundary alignment. In other words, the host could request for data not starting from a quad word aligned location, and the host interface block's responsibility is to manage this internally.

host_addr[14:0] (32K 64-bit wide locations) will address on a quad word boundary. The FB will put forth 8 bytes of data per request. The host interface block is responsible to unpack the 64-bit data from the requested byte number.

For example, the FB address locations are 18'h0, 18'h8, 18'h10, etc. That would translate to host_addr[14:0] = 15'h0, 15'h1, 15'h2, etc.

But if desired, the host can, as an example, request bytes starting from 18'h3 (a non QWORD boundary). The request to the FB from the host interface will be only the most significant 15 bits, which in this case will be 15'h0. The least significant 3 bits of the address will be used within the host interface block to calculate the byte number from where reads from the FIFO will begin.

7.5.3 ADPCM Output to Frame Buffer

A portion of the frame buffer will be allocated for storing ADPCM data, if HWR_FB_EN [1] is enabled. 4KB of the FB (1second worth of data) is allocated for audio data at the end address. This will be controlled in a FIFO like fashion where-in full, empty, half-full conditions will be flagged and registered in AUD_BUFF_STAT status register for the host to read and appropriately request or not request audio data. The read and write pointers of the audio space will be maintained internally.

The host requests audio data in the same way as is requests video data, except the host does not load a starting address. The initial request automatically starts from the beginning of the audio buffer and when switching from video reads back to audio reads, the subsequent requests for audio data will be from where it stopped in the previous transaction. For debugging purposes, the read pointer will be registered in AUD_BUFF_RD_PTR, by which the host can read the value and know from where it left off in the previous transaction.

1. Host reads AUD_QWORDS, which holds the count of qwords in the Audio FB. If sufficient bytes are present, the host will write register FB_ADDR_2[4:3] = 01, which represents an audio request.
2. The read will begin from the pointer stored in AUD_BUFF_RD_PTR.
3. In case of an SPI interface running at 27 MHz, the host should wait for the prefetch_done bit to set before reading data.
4. If the host empties the audio buffer, the transaction request bits in FB_ADDR_2 will be cleared. The host must then re-program FB_ADDR_2 to fetch more audio. If the host only reads out partially, then the host need not re-program FB_ADDR_2.

NOTE: When the audio buffer indicates half full, read an amount of audio data equal to AUD_QWORD - 2 Qwords in order not to completely empty the audio buffer. This will prevent the audio pointers from getting reset and will allow the host to return to an audio transaction request and continue reading audio data from where it left off.

Also note that while the audio data buffer maintains a read pointer, the video data buffer does not and will forget its place when switching between video and audio reads. If the host transitions from an audio to video data request, it must keep track of the next video data address so that it can load it into FB_ADDR_0 and FB_ADDR_1 to resume video data reads when it transitions back from audio reads.

If the host does not read out the audio data and allows the audio portion of the FB to get full, no new entries will be written to the FB. Only after the host reads data will there be new entries written to the FB. The AUD_FULL bit is set in AUD_BUFF_STAT register when the Audio buffer is filled.

Baby monitor mode: Audio streaming can have the following controls: START, STOP, CONTINUE and RESTART.

START is when the MIC_ADC_EN=1 and AUDIO_ENABLE =1

STOP is when MIC_ADC_EN =0, AUDIO_ENABLE = 1

RESTART is when MIC_ADC_EN =1, AUDIO_ENABLE = 1, AUD_RESTART = 1. This will reset the audio FB pointers.

CONTINUE is when MIC_ADC_EN = 1, AUDIO_ENABLE = 1. Same as START except that the audio read/write pointers are in the middle of the audio FB.

During STOP and RESTART, the ADPCM encode is also switched off until CONTINUE or START happens.

MIC_ADC_EN is bit[0] of register of ADC_CTRL_DIG_1.

AUD_RESTART is bit[2] of register HWR_FB_EN

AUDIO_ENABLE is bit[1] of register HWR_FB_EN

7.5.3.1 Audio Stop

Zero Padding

When in ADPCM 2-b/4-b mode, a STOP of audio will entail padding of as many 0's as there are remaining samples left to encode within the current ADPCM block. This is a default operation when in ADPCM mode, and cannot be turned off.

Example:

If the ADPCM block size is set to 505 and a STOP is issued when 489 audio samples have been encoded, then the remaining 16 samples that are left to be encoded will be padded with 8 bytes of 0's or in other words, 1 QWORD of 0's will be written to the audio buffer.

NOTE: A block size of 505 equals one 16-bit header and 504 16-bit samples of encoded audio with each 16 bit sample equaling four 4-bit audio samples in 4bit ADPCM mode.

Requirement for STOP

The zero padding is not gated by the remaining empty space available in the audio buffer. It is up to the host software to know when to appropriately STOP the audio. When the STOP command is issued, the host must ensures that at least one entire block size of buffer space is available for potential 0 padding of that amount. If a STOP is issued as soon as a new ADPCM block encode process has begun, the entire block size will be padded with 0's. In the example above, where the ADPCM block size is

505 (translates to a size of 256 bytes or 32 QWORDS), a STOP could pad up to 32 QWORDS of 0's. Hence a STOP should not be issued when less than 32 QWORDS of audio space is left in the audio buffer. (Remaining space = 512 - AUD_QWORDS). Please refer to section 6.3.1 for block size calculation.

Violating the above requirement could potentially overflow the audio buffer by overwriting the unread audio samples rendering the 4KB worth of audio corrupt.

Audio Flow

It is required that the host can keep up with the rate at which audio samples fill up the audio buffer.

- For RAW PCM mode, 4 samples at 8KHz each will cause one QWORD increment of the audio qword count, or 500us to fill one entry of the audio buffer.
- For 4-b ADPCM, 16 samples at 8KHz each will cause one QWORD increment of the audio qword count, or 2ms to fill one entry of the audio buffer.
- For 2-b ADPCM, 32 samples at 8KHz each will cause one QWORD increment of the audio qword count, or 4ms to fill one entry of the audio buffer.

Following are a couple of sequences that can be adopted to maintain a continuous audio flow and on how to handle a full condition or nearly full condition

1.

```
while(audio_transfer) {

QW = Read(AUD_QWORDS);

Threshold = 512 - ADPCM_N_QW; //ADPCM_N_QW is block size of ADPCM in QWORDS

if (QW =< Threshold) {

                    Fetch_Audio(QW); //Fetch QW amount of QWORDS of audio

} //end if

else { //"Nearly Full"

                    Fetch_Audio(QW);

AUD_STOP;

                   AUD_RESTART;

}   //end else

} //end while
```

2.

```
while(audio_transfer) {

QW = Read(AUD_QWORDS) ;

AUD_BUFF_FULL = Read(AUD_BUFF_STAT) & 0x2;
```

```
if (!AUD_BUFF_FULL || QW =< 512) {

                    Fetch_Audio(QW); //Fetch QW amount of QWORDS of audio

} //end if

else {

                    AUD_STOP;

                    AUD_RESTART; //The last 4KB of audio is discarded.

}   //end else

} //end while
```

Summary of Audio Flow Requirements

- Audio reads/fetch must be QWORD aligned
- Every new audio fetch after emptying the previous amount of QWORDS requires FB_ADDR_2 to be re-programmed to fetch audio.
- Audio STOP should be issued when more than 1 ADPCM block size of audio buffer space is available to avoid corruption.

NOTE: Audio fetch: When the host switches from audio to video data read, then it recommended that the switch occurs when audio data that was read was QWORD aligned. If not, the remaining bytes within the current QWORD being read will be lost.

7.5.4 Host Write to Frame Buffer

The host can write to FB directly through any of the active serial interface, if HWR_FB_EN [0] is enabled.

The host can write to the frame buffer, by writing to FB_ADDR _0 and FB_ADDR _1. The writes are always quad-word aligned, and so only 15 bits of address are required. In other words, host writes are not byte addressable. Setting FB_ADDR_2[4:3] to 11 would indicate that host is requesting a write operation.

1. Set FB_ADDR_0 and FB_ADDR_1 with the start address from where data should be written.
2. Then write to HWDATA_FB.

A write to this register in a burst mode, will not cause auto-increment of the address. The bytes written to this address will be concatenated in a 64-bit register in the host interface block till all 8 bytes are filled.

As long as the writes are coming through, the frame buffer address locations will be incremented for every new quad word to be written. However, if the audio buffer is enabled and the host reaches the end of the video location (0x7DFF in 256KB mode or 0x3DFF in 128KB mode), the next address location to be written will be wrapped to the start address (0x0000).

If not, the end address will be the frame buffer end (0x7FFF in 256KB mode or 0x3FFF in 128KB mode) where the next address location to be written will wrap to 0x0000.

NOTE: The host interface does not keep track of frame buffer full situation. As long as the host keeps writing, the address will keep incrementing until the end of video portion of the FB and then wrap around to the beginning. It is up to the host to keep track of the number of QWORDS being written and to stop when appropriate.

7.5.5 Debug and ADPCM bypass mode

There is capability to pre-load the entire FB with host writes, and then reading the audio or video FB separately.

The host can pre-load the entire FB, then enable AUDIO_ENABLE in register HWR_FB_EN. If the entire audio portion of the FB needs to be read, then set the AUD_DEBUG bit in HWR_FB_EN, and create an audio fetch request as explained above. The entire audio portion of the FB can be streamed out.

Here, the audio read and write pointers will be manipulated such that the read will begin from the middle of the audio buffer, and the AUD_QWORDS will read 512 qwords and will be updated as the bytes are read out.

To be able to go back to non-debug mode, the AUD_RESTART signal should be set to reset the audio read/write pointers.

To read video, the AUD_DEBUG bit need not be set, and the data from the video portion of the FB can be read like normal video reads.

Raw audio mode:

By setting RAW_AUD bit in HWR_FB_EN register, the ADPCM encode will be disabled and the direct mic data will be written to the audio portion of the FB.

8 Analog Interfaces

8.1 Microphone Input

A microphone input allows the CX93510 to record audio simultaneously during video capture. Audio is sampled at 8 kHz and the 16 bit data is compressed and saved as IMA-ADPCM 4 bit data in the Frame Buffer. A non standard 2 bit mode is also available.

A mic gain of 0-36 dB in 6 dB steps is available through register control. The mic input has an impedance of 10 kΩ and allows a maximum level of 1 Vp-p at 0 dB gain.

A low-noise, buffered 1.65 V reference is available to bias the microphone via an external resistor.

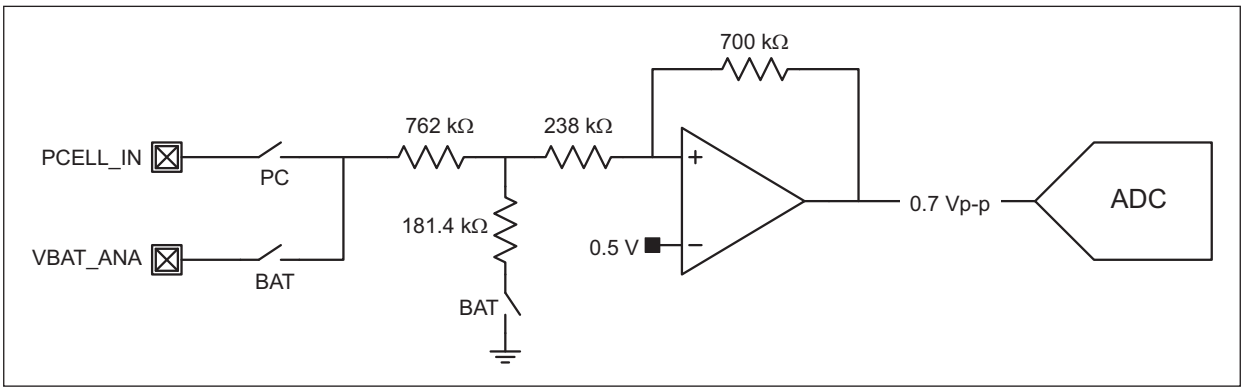
8.2 Battery Monitor

The Vbat voltage is measured using an internal ADC and available for the host to monitor low battery conditions. The analog photo sensor is muxed with same ADC used for battery monitoring. A control bit in register ADC1 selects between the external analog photo sensor input or the internal Vbat input.

8.2.1 Equation for Battery Monitoring

Figure 19 illustrates the internal path of the ADC used to measure battery voltage and photo-cell levels.

Figure 19. ADC Internal Path



The full scale is hardwired to 1 Vp-p. A swing of 0 to 1 V at the input of the ADC will give an output of -1 to +1.

Battery measurement equation:

ADCOUT = 1.8 - 0.7(Vbat)

ADCOUT is on a 16-bit 2's complement scale in the range of -2^{15} to $+2^{15}$.

There is an internal -4.8 dB (0.575) attenuation in the path, and given a full 16-bit scale (32768), the equation for battery measurements is:

Vbat = [((ADCOUT / 32768) / 0.575) +1.8] / 0.7

Vbat = (ADCOUT / 18841.6) / 0.7

Vbat = (ADCOUT / 13189) + 2.57

Example Calculations:

At 3.3 V, the ADCOUT value read is 0x297C = 10620 dec

Vbat = (10620/13189) + 2.57 = 3.37 V

At 2.3 V, the ADCOUT value read is 0xF4DD = -2850 dec

Vbat = (-2850/13189) + 2.57 = 2.35 V

Note the above equation assumes the default gain setting of 0dB in register 0x0B.

8.3 Photocell Sensor Input

In order to measure ambient light conditions, the CX93510 supports the use of analog or digital photo sensors (digital via I^2C). The digital sensor uses GPIO3 and GPIO4 and is enabled using bit 3 of register I2C_CTL_2. The analog sensor uses dedicated inputs and is muxed with the same ADC used for battery monitoring. A control bit in register ADC1 selects between the external analog photo sensor input or the internal Vbat input. The analog PCELL input accepts a voltage range of 0-1 V.

8.4 IR Illumination

An LED driver is provided for low-light compensation support. The driver has a PWM dimming logic block and allows pulse width control using the 6 bit led_pwm_ctrl bits. The cycle period is 200 Hz, and the duty cycle of this 200 Hz is varied, depending on the control input.

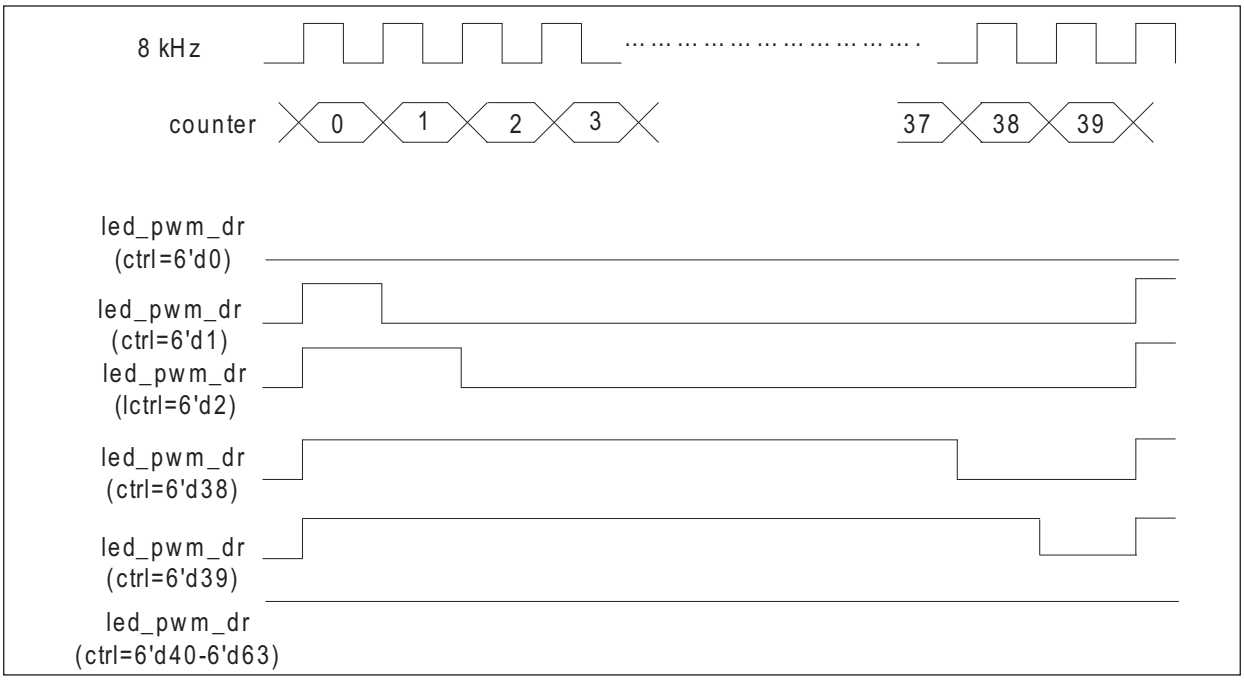
An 8 kHz clock is used to generate the 200 Hz period. The led_pwm_ctrl input is used to vary the 200 Hz period and to generate driver's output. The 8 kHz is divided in to 40 cycles, and the on-off period is varied between 0 to 40 cycles.Table 13 indicates the driver output based on the 6 bit setting.

Table 13. Driver Output Based on 6-Bit Setting

led_pwm_ctrl[5:0]	Driver output
6'd0	Off
6'd1	1-ON;39-OFF
6'd2	2-ON; 38-OFF
6'd3	3-ON; 37-OFF
....	....
6'd37	37-ON;3-OFF
6'd38	38-ON;2-OFF
6'd39	39-ON;1 –OFF
6'd40	On
6'd41-63	On

Table 20 illustrates an example of a driver output wave form.

Figure 20. Driver Output Waveform

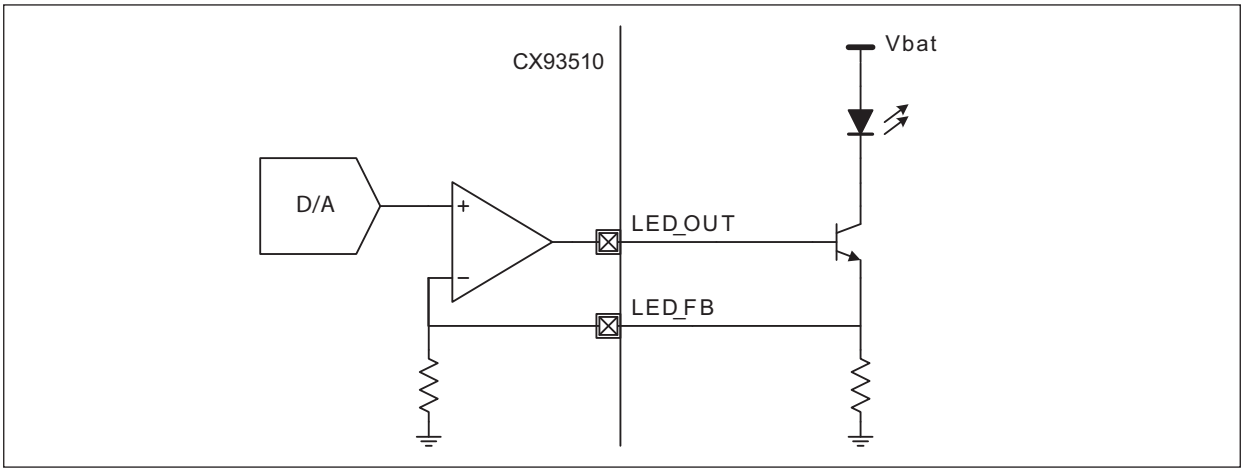


The driver can be configured either as a current source using an external FET (or NPN) -resistor combination, or as a voltage source that can be used to control an external DC-DC converter.

The voltage range on pin LED_FB is normally 0V to 1V.

In order to extend the output range beyond 0V-1V, a resistive divider on the feedback path can be enabled. When enabled, the output range is 0V to 1.8 V.

Figure 21. Current Source Configuration

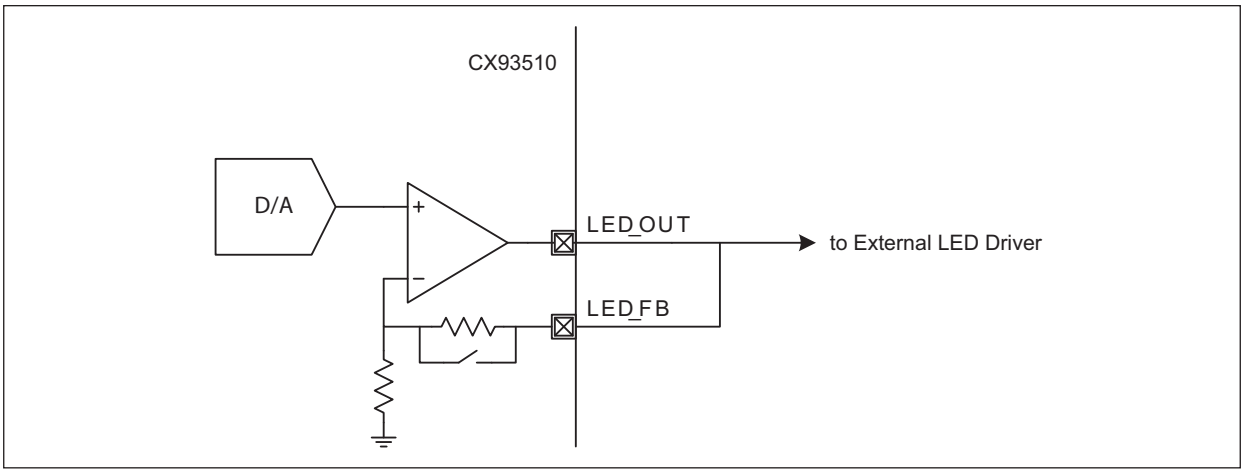


The D/A has a voltage range of 0 V to1 V and 5-bit resolution which gives 32.2 mV steps.

In case a NPN transistor is chosen instead of an nFET, the current drive capability for the CX93510 LED control output is 10 mA max. The NPN and resistor components should be chosen accordingly.

The max capacitive load on LED_OUT pin is 1000 pF.

Figure 22. Voltage Source Configuration



By shorting pins LED_OUT and LED_FB externally, the LED driver can be configured as a voltage buffer. The D/A functionality can then be used to provide an analog control voltage to an external LED driver.

9 Registers

9.1 Sensor Interface

9.1.1 Sensor Interface Configuration

Register: SI_CFG_1 **Address: 8'hA0**

Bits	Type	Default	Name	Description
[7:6]	RO	2'h0	reserved	Reserved = 0
[5]	RW	1'b0	POL_SVREF	SVREF Polarity: 0 = rising edge 1 = falling edge
[4]	RW	1'b0	POL_SHREF	SHREF Polarity: 0 = rising edge 1 = falling edge
[3:2]	RW	2'b00	B_ORD	YCbCr 4:2:2 bytes ordering: 00 = Cb, Y0, Cr, Y1 01 = Y0, Cb, Y1, Cr 10 = Cr, Y1, Cb, Y0 11 = Y1, Cr, Y0, Cb
[1]	RW	1'b0	Y_ONLY	Luminance Only: 0 = Color (4:2:2) 1 = B&W (8bpp, Y only)
[0]	RW	1'b0	CHRM_OFF	When set to 1, removes incoming Cr and Cb values and propagates only Y values. No effect if bit [1] is set.

Register: SI_CFG_2 **Address: 8'hA1**

Bits	Type	Default	Name	Description
[7]	RW	1'b0	EN_LFC	Enable capture of limited frames. The number of frames to be captured is defined in the FRAME_NUM field. The value in this register is reset once capture is complete. This bit must be cleared if EN_CFC is set.
[6]	RW	1'b0	EN_CFC	Enable continuous frame capture. This bit must be cleared if EN_LFC is set.
[5]	RW	1'b0	reserved	Reserved.
[4]	RW	1'b0	EMBD_CDE_ERR	This bit is set if an embedded code error is detected. This means that the end-of-line or end-of-frame was received before capture of the line/frame was completed. In this event, EN_LFC and EN_CFC will also be cleared and capture will stop. Writing a 0 to this bit will clear it; writing a 1 to this bit has no effect.
[3:0]	RW	4'h0	FRAME_SKIP	This field indicates how many frames to skip between captured frames.

Register: SI_CFG_3 **Address: 8'hA2**

Bits	Type	Default	Name	Description
[7]	RO	1'b0	reserved	Reserved
[6]	RW	1'b0	EN_ EMBD_CDE	Enable embedded Code mode: 0 = Use sync signals SHREF and SVREF 1 = Use embedded 656 (SHREF/SVREF can be used as GPIO[6:5])
[5:0]	RW	6'h0	FRAME_NUM	In Limited Frame mode, this field indicates how many frames to capture before stopping. This field is ignored in Continuous mode.

9.1.2 Horizontal Active

Register: H_ACT **Address: 8'hA3**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h00	H_ACT	This value indicates the number of **input clocks** from the rising edge of SHREF to the start of active input data.

9.1.3 Horizontal Capture Delay

Register: H_ CAP_DLY **Address: 8'hA4**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h00	H_CAP_DLY	This value multiplied by 8 Indicates the number of **pixels** to delay capture from start of active input data (or from the end of the SAV code if in Embedded Code mode).

9.1.4 Horizontal Capture Width

Register: H_ CAP_WIDTH **Address: 8'hA5**

Bits	Type	Default	Name	Description
[7]	RO	1'b0	reserved	Reserved = 0
[6:0]	RW	7'h50	H_CAP_WIDTH	This value multiplied by 8 indicates the number of pixels to capture for each line. The maximum supported register value is 0x50 (80 decimal), which corresponds to 640 pixels.

9.1.5 Vertical Active

Register: V_ACT **Address: 8'hA6**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h00	V_ACT	This value indicates the number of lines from the active edge of SVREF to the start of active input data.

9.1.6 Vertical Capture Delay

Register: V_CAP_DLY **Address: 8'hA7**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h00	V_CAP_DLY	This value multiplied by 8 indicates the number of lines to delay capture of data from the start of active input data (or from the start of frame code if in Embedded Code mode).

9.1.7 Vertical Capture Height

Register: V_CAP_HEIGHT **Address: 8'hA8**

Bits	Type	Default	Name	Description
[7:6]	RO	2'h0	reserved	Reserved = 0
[5:0]	RW	6'h3C	V_CAP_HEIGHT	This value multiplied by 8 indicates the number of lines to capture for each frame. The maximum supported register value is 0x3C (60 decimal), which corresponds to 480 lines.

9.1.8 I2C/SCCB Device Address

Register: I2C_DADDR **Address: 8'hA9**

Bits	Type	Default	Name	Description
[7:1]	RW	N/A	I2C_DADDR	Device Address for I2C/SCCB transaction to image sensor
[0]	RW	1'b0	PWDN_MODE	When set to 1, places the bus into low-power/idle mode. Must be reset to 0 to allow transactions. Set this bit before activating the PWDN signal on the SCCB bus, and clear it after clearing PWDN.

9.1.9 I2C/SCCB Address

Register: I2C_LO_ADDR **Address: 8'hAA**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h0	I2C_LO_ADDR	Address for I2C/SCCB transaction to image sensor, bits [7:0]

Register: I2C_HI_ADDR **Address: 8'hAB**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h0	I2C_HI_ADDR	Address for I2C/SCCB transaction to image sensor, bits [15:8] (Not used for SCCB).

9.1.10 I2C/SCCB Read/Write Data - Low Byte

Register: I2C_LO_DATA **Address: 8'hAC**

Bits	Type	Default	Name	Description
[7:0]	WO	N/A	I2C_LO_WDATA	Write data[7:0] for I2C/SCCB transaction to image sensor. If the I2C_DATA_16 field (in the I2C_CTL_3 register) is set to 1, the value in this register corresponds to the first byte of a 2-byte data transfer, and writes are enqueued by writing to the I2C_HI_DATA register. When the I2C_DATA_16 field is cleared (0), the value in this register corresponds to the data for the write transfer, and writing this register causes a write transaction to be queued for execution on the I2C or SCCB bus.
[7:0]	RO	N/A	I2C_LO_RDATA	Read data for I2C/SCCB transaction from image sensor. (If I2C_DATA_16 is set, this corresponds to the first byte of data returned during a read transaction.)

9.1.11 I2C/SCCB Read/Write Data - High Byte

Register: I2C_HI_DATA **Address: 8'hAD**

Bits	Type	Default	Name	Description
[7:0]	WO	N/A	I2C_HI_WDATA	Write data[15:8] for I2C/SCCB transaction to image sensor. If the I2C_DATA_16 field (in the I2C_CTL_3 register) is set to 1, the value in this register corresponds to the second byte of a 2-byte write data transfer, and writing this register causes a write transaction to be queued for execution on the I2C bus. If the I2C_DATA_16 field is cleared (0), this register is not used, and writes will have no effect.
[7:0]	RO	N/A	I2C_HI_RDATA	Read data for I2C/SCCB transaction from image sensor.

9.1.12 I2C Control

Register:I2C_CTL_1 **Address: 8'hAE**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h11	I2C_PERIOD	This number determines how many internal clock periods are used for each phase of SCL. There are typically four such phases per SCL clock period. For example, if the internal clock is running at 27 MHz (~37 ns clock period), and we want an SCL period of 400 KHz (2500ns period = 625 ns phase), we would want to set this value to 625 / 37, which is slightly less than 17, or 0x11.

Register:I2C_CTL_2 Address: 8'hAF

Bits	Type	Default	Name	Description
[7:6]	RW	2'h0	I2C_SADDR_LEN	Number of bytes in i2c sub-address. A value of "0" indicates no sub-address.
[5]	R/W	1'b0	I2C_SYNC	1 = allow slave to insert wait states
[4]	R/W	1'b0	I2C_READ_SA	1 = Reads generate write to set sub address first
[3]	RW	1'b0	I2C_PCS	Controls the Photocell Sensor I2C port shared with GPIO[4:3]. 0 = GPIO[4:3] used in GPIO mode. 1 = GPIO[4:3] used as Photocell Sensor I2C interface {SCL= GPIO[4]),SDA = GPIO[3]} from same I2C Master as in Sensor I/F blk.
[2]	RW	1'b0	I2C_SADDR_INC	Enable automatic incrementing of I2C address on each transaction
[1]	RW	1'b0	I2C_2_EN	I2C Secondary Interface Enable When this bit is 0, transactions will be routed to the image sensor interface's I2C (or SCCB) interface while the photo sensor I2C interface is kept idle. When this bit is 1, transactions will be routed to the photo sensor I2C interface while the image sensor's I2C (or SCCB) interface is kept idle. Do not change the state of this bit if the SI_CFG Write Fifo contains entries, or indeterminate behavior may result.
[0]	RW	1'b0	I2C_SCCB_EN	Determines if the image sensor interface is I2C (0) or SCCB (1) Note that if SCCB is to be used, the sensor must be reset after this bit is set.

Register:I2C_CTL_3 Address: 8'hB0

Bits	Type	Default	Name	Description
[7]	RW	1'b0	I2C_DATA_16	When set to 1, the i2c data interface is 16 bits wide, and the I2C_LO_DATA and I2C_HI_DATA registers are used to access read and write data. When cleared (0), the i2c data interface is 8 bits wide, and I@C_HI_DATA is not used. This bit must be cleared when using the SCCB protocol.
[6]	RW	1'b0	SCCB_E_EN	If SCCB_EN is 1, this bit determines if the SCCB_E signal is used. This signal is optional for the SCCB protocol. When this bit is 0, the pin can be used as GPIO[7].
[5]	R/W	1'b0	I2C_READ_WRN	transaction is read (1) or write (0); do not change the state of this bit while writes are queued up. A write to this register with this bit set to 1 will trigger a read transaction. When performing a write transaction, clear this bit before writing data to the I2C_LO_DATA and I2C_HI_DATA registers.
[4:2]	RO	3'h0	SI_CFG_FIFO_CNT	Indicates how many pending entries are in the SI CFG Write Fifo. The FIFO is capable of holding 4 entries. If it is necessary to perform more than 4 write transactions on the SCCB, system software must monitor the state of this count to avoid an overflow.
[1]	RO	1'b1	I2C_RACK	0 when transaction in progress or acknowledge has not been received from slave. 1 otherwise.
[0]	RO	1'b0	XFER_IN_PROG	I2C or SCCB Transfer in progress

9.1.13 Filter Config

Register: FILT_CFG **Address: 8'hB1**

Bits	Type	Default	Name	Description
[7:4]	RO	4'h0	reserved	Reserved = 0
[3:2]	RW	2'h0	reserved	Reserved
[1]	RW	1'b0	EN_BLACK	Black Fill 1 = use "00" for pixel values beyond edges of image for filter. 0 = replicate edge pixel for these values
[0]	RW	1'b0	EN_FILTER	1 = Enable 2:1 downscaling filter. 0 = no downscaling.

9.2 JPEG Controller

9.2.1 MJPEG-DPCM Control

Register: DIFF_JPEG_CTRL **Address: 8'h20**

Bits	Type	Default	Name	Description
[7]	RW	1'b0	RST_COMP	Soft reset of JPEG compression. Need to be used by Host when Frame Buffer is full in non-continuous mode or when there is a JPEG Encoder or Decoder error condition. This bit must be manually cleared after the INIT_DONE bit in address 8'h2B = 1.
[6]	RO	1'b0	FRM_BUF_FULL	Used to indicate when frame buffer is full when CONT_MODE is disabled ('0') 1 = Frame Buffer is full in Non-Continuous Mode 0 = Frame Buffer is not full in Non-Continuous Mode
[5]	RW	1'b0	CONT_MODE	Enables continuous processing of encoded frame data allowing automatic wrap of frame buffer memory. 0 = Continuous mode disabled. Data will fill the frame buffer, then stop when full. 1 = Continuous mode enabled. Data will continue to fill the frame buffer in a circular fashion. If data is not read, it will get overwritten as needed to accommodate new in-coming data. This setting is independent of the "Limited" or "Continuous" capture setting in register 0xA1.
[4:1]	RW	4'h0	REF_FREQ	Number of frames per Group of Pictures. Not relevant when DIFF_EN is disabled. Values 0 and 1 are not relevant during Differential JPEG mode.
[0]	RW	1'b1	DIFF_EN	Enables Differential JPEG mode 0 = Disabled 1 = Enabled

9.2.2 JPEG Encoder Status

Register: JPEG_ENC_STAT Address: 8'h21

Bits	Type	Default	Name	Description
[7]	RO	1'b0	ENC_HF_ERROR	Set when undefined Huffman table symbol is referenced
[6]	RO	1'b0	CTL_ERROR	Set when invalid SOF parameter is detected
[5]	RO	1'b0	HT_ERROR	Set when invalid DHT segment is detected
[4]	RO	1'b0	QT_ERROR	Set when invalid DQT segment is detected
[3]	RO	1'b0	ENC_ERROR	Set when any of EncFlags[7:4] are set
[2]	RO	1'b0	PIX_IN_PROG	Set when first sample of first 8x block is input to Encoder and de-asserted when last pixel of last block of the scan is input
[1]	RO	1'b0	ENC_IN_PROG	Set when encoding of scan is in progress and de-asserted when decoding of scan is complete
[0]	RW	1'b0	JPG_IN_PROG	Set when Encoder core starts processing configuration data and de-asserted at end of frame.
GENERAL NOTES: The RST_COMP bit should be set by the host if any "error" status bit is set by the device. This bit must be manually cleared after the INIT_DONE bit in address 8'h2B = 1. The RELOAD_TABLES bit in address 8'h26 should be set afterwards. Other status bits are for informative purposes only.				

Register: JPEG_ENC_STAT2 Address: 8'h22

Bits	Type	Default	Name	Description
[7:3]	RO	5'h0	reserved	Reserved
[2]	RO	1'b0	FULL_LM	Indicates Luma RBC buffer is full
[1]	RO	1'b0	FULL_CB	Indicates Cb RBC buffer is full
[0]	RO	1'b0	FULL_CR	Indicates Cr RBC buffer is full

Register: JPEG_ENC_PVAL_TYPE Address: 8'h23

Bits	Type	Default	Name	Description
[7]	RO	1'b0	reserved	Reserved
[6]	RO	1'b0	reserved	Reserved
[5]	RO	1'b0	reserved	Reserved
[4]	RO	1'b0	ENC_PVALID	Indicates valid coding parameters.
[3:0]	RW	4'h0	ENC_PTYPE	Specifies which parameter types are to be placed in the PVALUE registers

Register: JPEG_ENC_PVALUE1 Address: 8'h24

Bits	Type	Default	Name	Description
[7:0]	RO	8'h00	PVALUE_MSB	Most significant byte of encoder parameter value bus. Parameters placed on bus are selected by the PTYPE register value.

Register: JPEG_ENC_PVALUE2 Address: 8'h25

Bits	Type	Default	Name	Description
[7:0]	RO	8'h00	PVALUE_LSB	Least significant byte of encoder parameter value bus. Parameters placed on bus are selected by the PTYPE register value.

Register: JPEG_ENC_CTL1 Address: 8'h26

Bits	Type	Default	Name	Description
[7:6]	RO	2'b00	reserved	Reserved
[5:1]	RW	5'h10	reserved	Reserved - do not change default value.
[0]	RW	1'b0	RELOAD_TABLES	Reload DCT and Huffman Tables Set this bit after setting RST_COMP if an error should unlikely occur.

Register: JPEG_ENC_DCT_LM Address: 8'h27

Bits	Type	Default	Name	Description
[7:0]	RW	8'h00	DCT_TAB_LM	DCT Table number for Luma (0x00 or 0x02). 00 = Standard Table 02 = High Compression Table

Register: JPEG_ENC_DCT_CH Address: 8'h28

Bits	Type	Default	Name	Description
[7:0]	RW	8'h01	DCT_TAB_CH	DCT Table number for Chroma (0x01 or 0x03). 01 = Standard Table 03 = High Compression Table

9.2.3 JPEG Decoder Status

Register: JPEG_DEC_STAT1 Address: 8'h29

Bits	Type	Default	Name	Description
[7]	RO	1'b0	DEC_HF_ERR	Set when the JPEG decoder has encountered an undefined Huffman table symbol.
[6]	RO	1'b0	CTL_ERR	Set when the JPEG decoder has encountered an invalid SOF parameter.
[5]	RO	1'b0	HT_ERR	Set when the JPEG decoder has encountered an invalid DHT segment.
[4]	RO	1'b0	QT_ERR	Set when the JPEG decoder has encountered an invalid DQT segment.
[3]	RO	1'b0	DEC_ERR	Set when the JPEG decoder has encountered the errors defined in bits [7:4] above, when any SOF marker is detected other than SOF0 or when an incomplete Huffman or quantization definition is detected.
[2]	RO	1'b0	IDCT_IN_PROG	Set when the first sample of the first 8x8 block is output into the decoder core and de-asserted when the last pixel of the last block of the scan is output.
[1]	RO	1'b0	DEC_IN_PROG	For each scan this signal is asserted after SigSOS signal has been output from the core and is de-asserted when decoding is complete. It indicates that the core is in the decoding state.
[0]	RO	1'b0	JPG_IN_PROG	Set when core starts to process input data and de-asserted when decoding has been completed i.e when the last pixel of the last block of the image is output.
GENERAL NOTES: The RST_COMP bit should be set by the host if any "error" status bit is set by the device. This bit must be manually cleared after the INIT_DONE bit in address 8'h2B = 1. The RELOAD_TABLES bit in address 8'h26 should be set afterwards. Other status bits are for informative purposes only.				

Register: JPEG_DEC_STAT2 Address: 8'h2A

Bits	Type	Default	Name	Description
[7]	RO	1'b0	Reserved	
[6]	RO	1'b0	PVALID	Indicates valid coding parameters.
[5]	RO	1'b0	SIGSOS	Indicates that an SOS segment has been input via jpgin input and the decoder is about to start decoding a scan.
[4]	RO	1'b0	INIT_PROG	Indicates that the decoder core is currently initializing its memories.
[3:0]	RW	4'h0	DEC_PTYPE	Specifies which parameter types are to be placed in the PVALUE registers

Register: JPEG_DEC_STAT3 Address: 8'h2B

Bits	Type	Default	Name	Description
[7:6]	RO	2'h0	reserved	Reserved
[5]	R/W	1'b0	BYCNT_OVF_CLR	Setting the bit will clear the BYCNT_OVF status bit.
[4]	RO	1'b0	BYCNT_OVF	Jpeg frame byte count overflow condition. Indicates when the compression block frame buffer port experiences a frame status qword byte count overflow. The overflow is triggered when a particular frame exceeds: 256K and no audio buffer: 256KB 256K and audio buffer: 252KB 128K and no audio buffer: 128KB 128K and audio buffer: 124KB This status can be cleared by setting the BYCNT_OVF_CLR bit or issuing a soft reset to the compression block (RST_COMP)
[3]	RO	1'b0	DEC_PROC_ERR	Jpeg decoder has erroneously started processing data that is neither configuration data nor a reference frame
[2]	RO	1'b0	DEC_PROC_REF	Jpeg decoder currently processing reference frame
[1]	RO	1'b0	DEC_PROC_CFG	Jpeg decoder currently processing configuration data
[0]	RO	1'b0	INIT_DONE	Compression block initialization complete

9.3 Host Interface

9.3.1 Slave Select Control

Register: SLAVE_SEL_CTRL **Address: 8'h50**

Bits	Type	Default	Name	Description
[7:3]	RO	5'h00	reserved	Reserved
[2]	RW	1'b0	U_FLW_CTRL	UART flow control enable. Host writes a 1 to this bit to indicate flow control is enabled. If 0, flow control is ignored.
[1:0]	RO	2'b11	SLAVE_SELECT	2'b00 = I2C 2'b01 = SPI 2'b10 = UART 2'b11 = No interface selected

9.3.2 Frame Buffer 18-b Address

Register: FB_ADDR _0 **Address: 8'h51**

Bits	Type	Default	Name	Description
[7]	RO	1'b0	reserved	Reserved
[6:0]	RW	7'h00	FB_ADDR_0	Upper 7 bits of the video data FB address [14:8] on a qword boundary

Register: FB_ADDR _1 **Address: 8'h52**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h00	FB_ADDR_1	Lower 8 bits of the video data FB address [7:0] on a qword boundary

Register: FB_ADDR _2 **Address: 8'h53**

Bits	Type	Default	Name	Description
[7]	RO	1'b0	reserved	
[6]	RO	1'b0	VID_REQ_ERR	Request happens from audio space like a video request. New request with the correct address will clear bit. Error occurs only when AUDIO_ENABLE=1. Mirrored bit from ERR_STAT bit[2].
[5]	RW	1'b0	PREFETCH_DONE	prefetch_done. After writing to the TRANSACTION bits below, the PREFETCH_DONE bit will set, indicating that the frame buffer may be read. It does not indicate that frame data is necessarily ready. It merely gives the green light to read from the frame buffer.
[4:3]	RW	2'h0	TRANSACTION	00 = No transaction 01 = Audio request 10 = Video request 11 = Host Write Writing to this register prefetches 2 QWORDS from the FB. Any change in transaction request requires this register to be written. During audio transfer, if the audio buffer is emptied, these bits will be cleared.
[2:0]	RW	3'h0	QWORD_BYTE_PTR	Least significant bits of FB address. Points to the byte within the qword. Used only for video reads.

9.3.3 Error Status

Register: ERR_STAT **Address: 8'h54**

Bits	Type	Default	Name	Description
[7]	RO	1'b0	FB_ERR	Summary bit for all of the frame buffer related error status. If bit is set, check the error status bits of the frame buffer to zero-in on the exact error.
[6]	RO	1'b0	SENS_ERR	Summary bit for all of the sensor interface related error status. If bit is set, check the error status bits of the sensor interface to zero-in on the exact error.
[5]	RO	1'b0	DEC_SUMM_ERR	Summary bit for all of the JPEG decoder related error status. If bit is set, check the error status register bits of the JPEG decoder to zero-in on the exact error.
[4]	RO	1'b0	ENC_SUMM_ERR	Summary bit for all of the JPEG encoder related error status. If bit is set, check the error status register bits of the JPEG encoder to zero-in on the exact error.
[3]	RO	1'b0	WR_REQ_ERR	Host writes to FB with start address in audio space. Bit will be cleared when request is re-submitted with the correct address. Error occurs only when AUDIO_ENABLE=1.
[2]	RO	1'b0	VID_REQ_ERR	Request happens from audio space like a video request. New request with the correct address will clear bit. Error occurs only when AUDIO_ENABLE=1.
[1]	RO/W1C	1'b0	FIFO_UNDERRUN	Out FIFO is read from when empty. Write 1 to clear
[0]	RO/W1C	1'b0	GNT_ERR	Host is not granted access to a request. FB can deny grant if request is made in the even cycle. Write 1 to clear.

9.3.4 Host Interface Enable to Write to FB

Register: HWR_FB_EN **Address: 8'h55**

Bits	Type	Default	Name	Description
[5]	RW	1'b0	VID_BYP_EN	When enabled, bypasses the raster-to-block conversion and JPEG encoding (direct or MJPEG). Sensor input pixels are fed directly to Frame Buffer in raster order.
[4]	WO	1'b0	AUD_DEBUG	Used in FB debug mode where FB is pre-loaded and data is read out of audio FB like a normal audio read transaction. Bit will be cleared internally when audio pointers are manipulated.
[3]	RW	1'b0	RAW_AUD	Bypass ADPCM encode block and write to audio FB, raw audio data
[2]	WO	1'b0	AUD_RESTART	This bit will reset the audio read/write FB pointers. Will be cleared internally when pointers are reset
[1]	RW	1'b0	AUDIO_ENABLE	Enabling this bit would allocate 4KB of space in FB, and store ADPCM out in FB
[0]	RW	1'b0	HOST_WRITE_ENABLE	Enable bit set to be able for the host to write to FB

9.3.5 Write/Read Data From/To host

Register: HWDATA_FB **Address: 8'h56**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h0	HWDATA_FB	Host writes through this register to write to FB. Every write should be multiples of 8 bytes (qwords).

Register: HRDATA_FB **Address: 8'hCC**

Bits	Type	Default	Name	Description
[7:0]	RO	8'h0	HRDATA_FB	Host reads FB data from this location.

9.3.6 Audio Buffer Status

Register: AUD_BUFF_STAT **Address: 8'h57**

Bits	Type	Default	Name	Description
[7:3]	RO	5'h0	reserved	Reserved
[2]	RO	1'b1	AUD_EMPTY	Host should not issue audio read request until this bit is de-asserted. Represents that Audio FB is empty.
[1]	RO	1'b0	AUD_FULL	Indicates audio buffer is full when =1.
[0]	RO	1'b0	AUD_HALF_FULL	Water line mark. Host reads this bit, and requests audio data.

9.3.7 Audio Buffer Read Pointer

Register: AUD_ RD_PTR_0 **Address: 8'h58**

Bits	Type	Default	Name	Description
[7]	RO	1'b0	reserved	Reserved
[6:0]	RO	7'h7E / 7'h3E based on 256KB/128KB BO	AUD_ RD_PTR_0	Audio buffer read pointer holding the pointer address of the next audio sample location. MSB. Must read this register before reading the LSB register (used for debug purposes only).

Register: AUD_ RD_PTR_1 **Address: 8'h59**

Bits	Type	Default	Name	Description
[7:0]	RO	8'h00	AUD_ RD_PTR_1	Audio buffer read pointer LSB. Must read this register after reading the MSB register (used for debug purposes only).

9.3.8 Audio Byte Count

Register: AUD_ QWORDS _0 **Address: 8'h5A**

Bits	Type	Default	Name	Description
[7:0]	RO	8'h0	AUD_QWORD_0	Number of QWORDS left in the FB. Must read this register before reading the LSB register. Note: When the audio buffer indicates half full, read an amount of audio data equal to AUD_QWORD - 2 Qwords in order not to completely empty the audio buffer. This will prevent the audio pointers from getting reset.

Register: AUD_ QWORDS _1 **Address: 8'h5B**

Bits	Type	Default	Name	Description
[7:2]	RO	6'h0	reserved	Reserved
[1:0]	RO	2'h0	AUD_QWORD_1	Least Significant 2 bits of Audio QWORD count. Must read this register after reading the MSB register.

9.3.9 ADPCM No. of Samples (10-bit value MIN = 32, MAX = 512)

Register: ADPCM_N_0 **Address: 8'h5C**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h7E	ADPCM_N_0	Represent samples per block (MSB)

Register: ADPCM_N_1 **Address: 8'h5D**

Bits	Type	Default	Name	Description
[1:0]	RW	2'b01	ADPCM_N_1	LSB of the sample size

9.3.10 ADPCM Mode

Register: ADPCM **Address: 8'h5E**

Bits	Type	Default	Name	Description
[7:1]	RO	7'h00	reserved	Reserved
[0]	RW	1'b0	ADPCM_MODE	0 = 4-b MSFT IMA ADPCM 1 = 2-b variant

9.3.11 Photocell Output and Battery Monitor (16 Bits)

Register: PHCELL_OUT _0 **Address: 8'h5F**

Bits	Type	Default	Name	Description
[7:0]	RO	8'h0	PHCELL_OUT_0	MSB of Photocell/Battery reading. This is register must be read prior to the LSB register to ensure that both the values correspond to the same ADC sample.

Register: PHCELL_OUT _1 **Address: 8'h60**

Bits	Type	Default	Name	Description
[7:0]	RO	8'h0	PHCELL_OUT_1	LSB of Photocell/Battery reading. This register must be read after the MSB register to ensure that both the values correspond to the same ADC sample.

9.3.12 GPIO Inputs

Register: GPIN **Address: 8'hF0**

Bits	Type	Default	Name	Description
[7:0]	RO	8'hXX	GPIN	GPIO inputs read directly from the pins.

9.3.13 GPIO Outputs

Register: GPOUT **Address: 8'hF1**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h00	GPOUT	GPIO outputs written directly to the pins. The same value can be read back via GPIN register if the pins are in the output mode.

9.3.14 GPIO Output Enables

Register: GPOE **Address: 8'hF2**

[7:0]	RW	8'h00	GPOE	GPIO output enables controlling the mode of the pins: 0 = input 1 = output

9.3.15 Part Number, Bond Option, and Revision Information

Register: PN_BO_REV **Address: 8'hFF**

Bits	Type	Default	Name	Description
[7:4]	RO	4'h0	PID	Product ID, 0= CX93510
[3:2]	RO	2'b11	reserved	Reserved
[1:0]	RO	2'b00	REV	Revision ID: 00 = RevA0

9.4 AFE

9.4.1 Microphone and Auxiliary ADC

Register: ADC1 **Address: 8'h00**

Bits	Type	Default	Name	Description
[7]	RW	1'b0	en_micadc	Enable for Microphone analog PGA and ADC. Use in addition to digital gain setting in register 0x0A. 0 = OFF 1 = ON
[6:4]	RW	3'b000	mic_gain	Microphone Boost PGA gain control: 000 = 0 dB 001 = 6 dB 010 = 12 dB 011 = 18 dB 100 = 24 dB 101 = 30 dB 110 = 36 dB 111 = 42 dB
[3]	RW	1'b0	pd_micbias	Disable for Microphone Bias voltage 0 = ON 1 = OFF
[2]	RW	1'b0	reserved	Reserved
[1]	RW	1'b0	reserved	Reserved
[0]	RW	1'b0	sel_auxadc	Select input for Battery/Photocell ADC 0 = Photocell voltage measurement 1 = Battery voltage measurement Note: Always discard first read to prevent erroneous values.

9.4.2 LED Driver

Register: LED1 **Address: 8'h02**

Bits	Type	Default	Name	Description
[7]	RW	1'h0	pdb_led	Enable for LED driver 0 = OFF 1 = ON
[6:2]	RW	5'h0	ctrl_led	Output voltage for LED driver (on pin LED_FB, in low gain mode) 00000 = 0 V 00001 = 32 mV … 11111 = 1 V Example: A value of 10000'b (0x10) = 512 mV. Using a 0.5 Ω load resistor in the IR LED circuit will result in approximately 1 A of current. This assumes the higain_led bit below is off.
[1]	RW	1'b0	higain_led	Enable high gain mode 0 = low gain mode 1 = high gain mode: output range is multiplied by 1.8
[0]	RW	1'b0	led_dis_ilim	Disable current limiting on LED driver 0 = current limiting enabled 1 = current limiting disabled

9.4.3 Regulator for Analog Supply

Register: LDO_ANA **Address: 8'h03**

Bits	Type	Default	Name	Description
[7]	RW	1'b0	pd_regana	Disable for analog LDO 0 = ON 1 = OFF
[6:3]	RW	4'h0	ctrl_regana	Output voltage for analog LDO (AVDD) – 25mV steps 0000 = 1.000 V 0001 = 1.025 V …. 0111 = 1.175 V 1000 = 0.800 V 1001 = 0.825 V … 1111 = 0.975 V
[2:0]	RW	3'b000	reserved	Reserved
GENERAL NOTES: All internal regulators default to an appropriate nominal value. Changes to these values should be reserved for special circumstances only and should be performed in small increments as to not trigger a possible reset condition.				

9.4.4 Regulator for Digital and RAM Supplies

Register: LDO_DIG **Address: 8'h04**

Bits	Type	Default	Name	Description
[7:4]	RW	4'h0	ctrl_regvdd	Output voltage for digital LDO (VDD)– 25 mV steps 0000 = 0.800 V 0001 = 0.825 mV …. 0111 = 0.975 V 1000 = 0.600 V 1001 = 0.625 V … 1111 = 0.775 V
[3:0]	RW	4'h0	ctrl_regvrr	Output voltage for memory LDO (VRR) – 25 mV steps 0000 = 0.800 V 0001 = 0.825 mV …. 0111 = 0.975 V 1000 = 0.600 V 1001 = 0.625 V … 1111 = 0.775 V
GENERAL NOTES: All internal regulators default to an appropriate nominal value. Changes to these values should be reserved for special circumstances only and should be performed in small increments as to not trigger a possible reset condition.				

9.4.5 Regulator for RAM Supply - Retention Mode

Register: RET_LDO_DIG **Address: 8'h05**

Bits	Type	Default	Name	Description
[7:4]	RW	4'h0		Reserved
[3:0]	RW	4'h0	ret_ctrl_regvrr	Output voltage for memory LDO (VRR) in retention mode – 25 mV steps 0000 = 0.800 V 0001 = 0.825 mV …. 0111 = 0.975 V 1000 = 0.600 V 1001 = 0.625 V … 1111 = 0.775 V Note: Any change will take place only after the PD pin goes high to low.
GENERAL NOTES: All internal regulators default to an appropriate nominal value. Changes to these values should be reserved for special circumstances only and should be performed in small increments as to not trigger a possible reset condition.				

9.4.6 ADC Decimation Parameters

Register: ADC_CTRL_DIG1 Address: 8'h0A

Bits	Type	Default	Name	Description
[7:4]	RW	4'h0	reserved	Reserved
[3]	RW	1'b0	mic_sample_rate	Output sample rate selection for mic_adc mic_sample_rate=1'b1; fs=4k mic_sample_rate=1'b0; fs=8k(default)
[2:1]	RW	2'b00	mic_adc_gain	This is used to control the digital gain of the mic ADC. S.no / mic _adc_gain / Gain 1 / 00 / 0dB 2 / 01 / 4.8dB 3 / 10 / 3dB 4 / 11 / 7.8dB
[0]	RW	1'b0	mic _adc_en	Enable for digital microphone ADC. Adc is enabled when mic _adc_en=1; Adc is disabled when mic _adc_en=0; Default mic _adc_en =1'b0

Register: ADC_CTRL_DIG2 Address: 8'h0B

Bits	Type	Default	Name	Description
[7:3]	RW	5'h0	reserved	Reserved
[2:1]	RW	2'b00	aux_adc_gain	This is used to control the gain of the battery/photocell monitor ADC. S.no / aux_adc_gain / Gain 1 / 00 / 0dB 2 / 01 / 4.8dB 3 / 10 / 3dB 4 / 11 / 7.8dB
[0]	RW	1'b0	aux_adc_en	Enable for battery monitor ADC. Adc is enabled when aux_adc_en=1; Adc is disabled when aux_adc_en=0; Default aux_adc_en =1'b0

9.4.7 LED PWM Control

Register: LED_PWM_CTRL **Address: 8'h0E**

Bits	Type	Default	Name	Description
[7:2]	RW	6'h3F	LED_PWM_CTRL	Duty cycle control for LED driver – 2.5% steps 6'd0= 0% 6'd1 = 2.5% … 6'd39=97.5% 6'd40 and up =100%
[1:0]	RW	2'h0	Reserved	Reserved

9.4.8 LED ON_DELAY

Register: LED_ON_DELAY **Address: 8'h0F**

Bits	Type	Default	Name	Description
[7:0]	RW	8'h00	LED_ON_DELAY	When this field is set to a value other than 00, the sensor interface will automatically clear the LED_PWM_CTL value that is fed to the AFE. (register 8'h0E, bits 7:2) at the end of each captured image, and then restore the programmed value after a delay. This provides a means of automatically turning off the IR LEDs at the end of each captured frame to save power. Based on the the frame rate used, the user must decide how much delay to program in order for the LEDs to be enabled again prior to the next frame. LED_ON_DELAY defines this turn-on delay in milliseconds. If the sensor interface is programmed for limited frame capture, the LED_PWM_CTL value sent to the AFE will remain 0 after the final frame, until overridden by a write to the LED_PWM_CTRL register. (When the sensor interface clears the value sent to the AFE, the change is not visible on reads of the LED_PWM_CTRL register. When this field is 00, the sensor interface does not affect the LED_PWM_CTRL on or off.

9.5 Extended Temperature Considerations

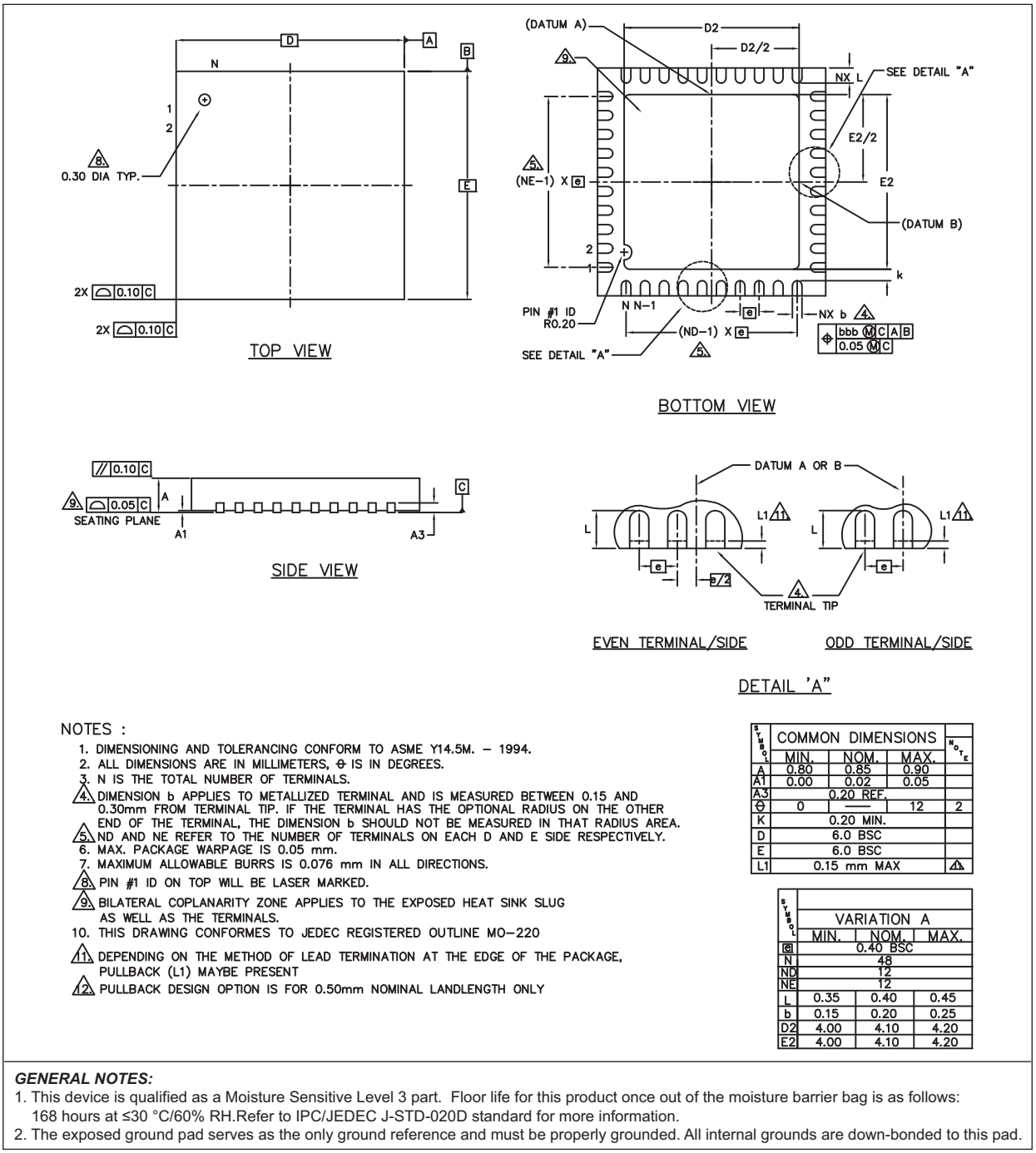
Systems designed to work in an extended temperature range between +70 to +85 ^{o}C should follow proper design rules. High-temperature systems need to follow design guidelines specific to high-temperature operation to help maintain the junction temperature below the maximum of +100 ^{o}C. The design considerations should include, but not be limited to the list below:

1. Do not make the board excessively small, and use a minimum of four layers.
2. Use at least 20 percent Cu content for the top layer.
3. Use a solid Cu plane for the inner layers.
4. The ground pad on the bottom of the device must be tied in to ground through thermal via, as it provides a good path for transferring heat to the PCB.
5. Use a thermal via size of 0.3 mm with 25 μm Cu plating in the via.
6. The thermal via pitch is spaced on a 1.2 x 1.2 mm grid. The epad size is 4.1 x 4.1 mm, therefore the number of thermal vias to be used is ~ 4 x 4 = 16.

10 Package Diagrams

Figure 23 provides the package diagram.

Figure 23. Package Diagram



NOTES :

1. DIMENSIONING AND TOLERANCING CONFORM TO ASME Y14.5M. – 1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS, θ IS IN DEGREES.
3. N IS THE TOTAL NUMBER OF TERMINALS.
4. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15 AND 0.30mm FROM TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
5. ND AND NE REFER TO THE NUMBER OF TERMINALS ON EACH D AND E SIDE RESPECTIVELY.
6. MAX. PACKAGE WARPAGE IS 0.05 mm.
7. MAXIMUM ALLOWABLE BURRS IS 0.076 mm IN ALL DIRECTIONS.
8. PIN #1 ID ON TOP WILL BE LASER MARKED.
9. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.
10. THIS DRAWING CONFORMES TO JEDEC REGISTERED OUTLINE MO-220
11. DEPENDING ON THE METHOD OF LEAD TERMINATION AT THE EDGE OF THE PACKAGE, PULLBACK (L1) MAYBE PRESENT
12. PULLBACK DESIGN OPTION IS FOR 0.50mm NOMINAL LANDLENGTH ONLY

SYMBOL	COMMON DIMENSIONS			NOTE
	MIN.	NOM.	MAX.	
A	0.80	0.85	0.90	
A1	0.00	0.02	0.05	
A3		0.20 REF.		
θ	0	—	12	2
K		0.20 MIN.		
D		6.0 BSC		
E		6.0 BSC		
L1		0.15 mm MAX		11

SYMBOL	VARIATION A		
	MIN.	NOM.	MAX.
e		0.40 BSC	
N		48	
ND		12	
NE		12	
L	0.35	0.40	0.45
b	0.15	0.20	0.25
D2	4.00	4.10	4.20
E2	4.00	4.10	4.20

GENERAL NOTES:

1. This device is qualified as a Moisture Sensitive Level 3 part. Floor life for this product once out of the moisture barrier bag is as follows: 168 hours at ≤30 °C/60% RH.Refer to IPC/JEDEC J-STD-020D standard for more information.
2. The exposed ground pad serves as the only ground reference and must be properly grounded. All internal grounds are down-bonded to this pad.

10.1 Reflow Profile

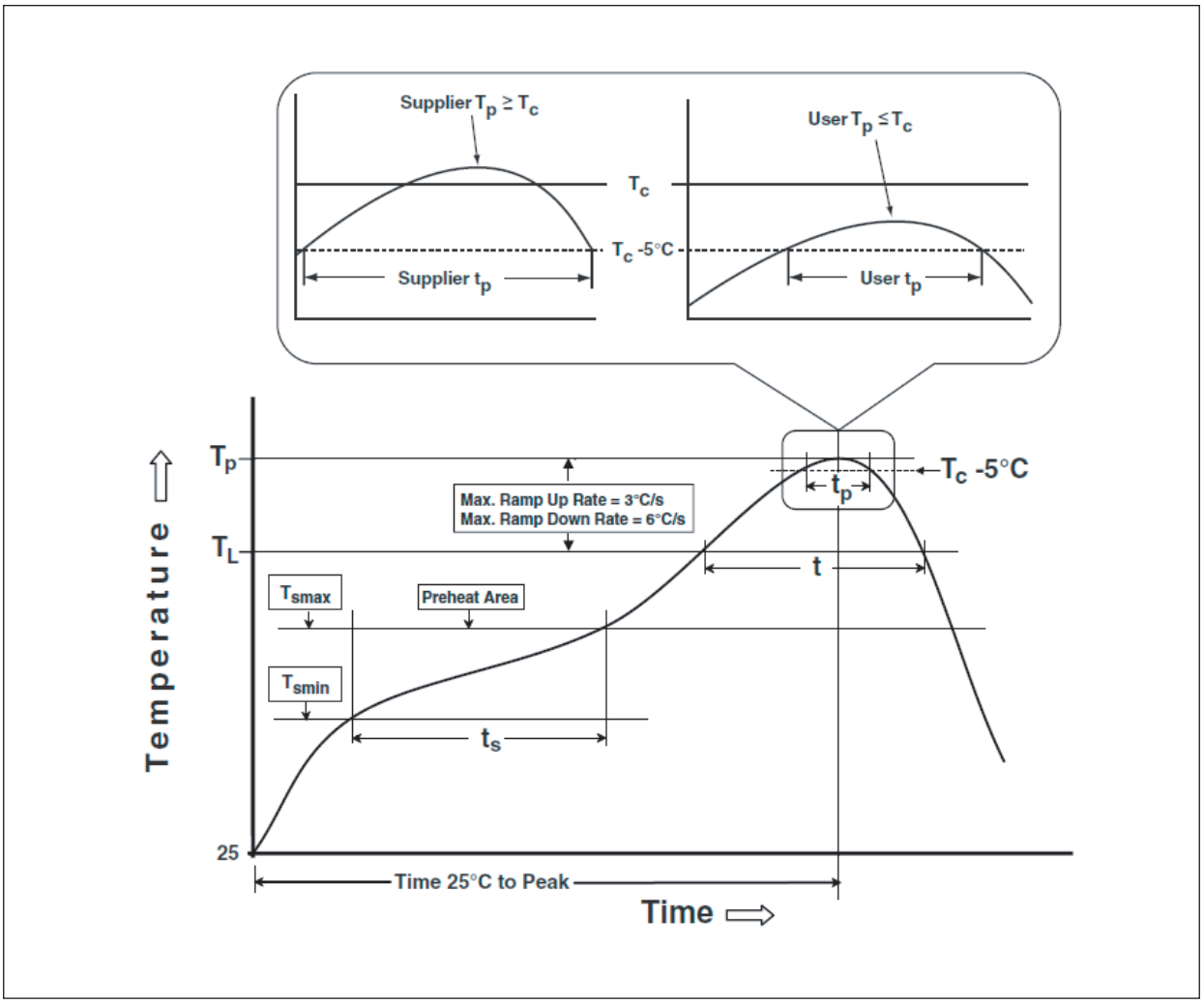
Guidelines for PB-free assembly are shown in Table 14. The classification profile is shown in Figure 24.

Table 14. Classification Reflow Profile

Profile Feature	Pb-Free Assembly
Preheat and Soak Temperature min (T_{smin}) Temperature max (T_{smax}) Time (T_{smin} to T_{smax}) (t_s)	 150 °C 200 °C 60–120 seconds
Average ramp-up rate (T_{smax}) to T_p)	3 °C/second max
Liquidity temperature (T_L) Time at liquidity (t_L)	217 °C 60–150 seconds
Peak package body temperature (T_p)[1]	260 °C
Time (t_P)[2] within 5 °C of the specified classification temperature (T_c)	30[1] seconds
Average ramp-down to peak temperature	6 °C/second max
Time 25 °C to peak temperature	8 minutes max

FOOTNOTES:
(1) Tolerance for peak profile temperature (T_p) is defined as a supplier minimum and a user maximum.
(2) Tolerance for time at peak profile temperature (t_P) is defined as a supplier minimum and a user maximum.

Figure 24. Classification Profile



www.conexant.com
General Information:
U.S. and Canada: (888) 855-4562
International: (949) 483-4600
Headquarters – Newport Beach
4000 MacArthur Blvd.
Newport Beach, CA 92660