TMS320DM357 DMSoC
Video Processing Front End (VPFE)

User's Guide

Literature Number: SPRUG39
November 2008

Contents

List of Figures

List of Tables

Read This First

About This Manual

This document describes the video processing front end (VPFE) in the TMS320DM357 Digital Media System-on-Chip (DMSoC).

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.

Related Documentation From Texas Instruments

The following documents describe the TMS320DM357 Digital Media System-on-Chip (DMSoC). Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

SPRUG06 — ***TMS320DM357 DMSoC Video Processing Back End (VPBE) User's Guide.*** Describes the video processing back end (VPBE) in the TMS320DM357 Digital Media System-on-Chip (DMSoC) video processing subsystem. Included in the VPBE is the video encoder, on-screen display, and digital LCD controller.

SPRUG25 — ***TMS320DM357 DMSoC ARM Subsystem Reference Guide.*** Describes the ARM subsystem in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The ARM subsystem is designed to give the ARM926EJ-S (ARM9) master control of the device. In general, the ARM is responsible for configuration and control of the device; including the video processing subsystem, and a majority of the peripherals and external memories.

SPRUG26 —***TMS320DM357 DMSoC Universal Asynchronous Receiver/Transmitter (UART) User's Guide.*** This document describes the universal asynchronous receiver/transmitter (UART) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The UART peripheral performs serial-to-parallel conversion on data received from a peripheral device, and parallel-to-serial conversion on data received from the CPU.

SPRUG27 — ***TMS320DM357 DMSoC Inter-Integrated Circuit (I2C) Peripheral User's Guide.*** Describes the inter-integrated circuit (I2C) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The I2C peripheral provides an interface between the DMSoC and other devices compliant with the I2C-bus specification and connected by way of an I2C-bus. External components attached to this 2-wire serial bus can transmit and receive up to 8-bit wide data to and from the DMSoC through the I2C peripheral. This document assumes the reader is familiar with the I2C-bus specification.

SPRUG28 — *TMS320DM357 DMSoC 64-Bit Timer User's Guide.* Describes the operation of the software-programmable 64-bit timer in the TMS320DM357 Digital Media System-on-Chip (DMSoC). Timer 0 and Timer 1 are used as general-purpose (GP) timers and can be programmed in 64-bit mode, dual 32-bit unchained mode, or dual 32-bit chained mode; Timer 2 is used only as a watchdog timer. The GP timer modes can be used to generate periodic interrupts or enhanced direct memory access (EDMA) synchronization events. The watchdog timer mode is used to provide a recovery mechanism for the device in the event of a fault condition, such as a non-exiting code loop.

SPRUG29 — *TMS320DM357 DMSoC Serial Peripheral Interface (SPI) User's Guide.* Describes the serial peripheral interface (SPI) in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The SPI is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (1 to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communication between the DMSoC and external peripherals. Typical applications include an interface to external I/O or peripheral expansion via devices such as shift registers, display drivers, SPI EPROMs and analog-to-digital converters.

SPRUG30 — *TMS320DM357 DMSoC Host Port Interface (HPI) Reference Guide.* This document describes the host port interface in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The HPI provides a parallel port interface through which an external host processor can directly access the TMS320DM357 DMSoC processor's resources (configuration and program/data memories).

SPRUG31 — *TMS320DM357 DMSoC General-Purpose Input/Output (GPIO) User's Guide.* Describes the general-purpose input/output (GPIO) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The GPIO peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an input, you can detect the state of the input by reading the state of an internal register. When configured as an output, you can write to an internal register to control the state driven on the output pin.

SPRUG32 — *TMS320DM357 DMSoC Multimedia Card (MMC)/Secure Digital (SD) Card Controller User's Guide.* Describes the multimedia card (MMC)/secure digital (SD) card controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The MMC/SD card is used in a number of applications to provide removable data storage. The MMC/SD controller provides an interface to external MMC and SD cards. The communication between the MMC/SD controller and MMC/SD card(s) is performed by the MMC/SD protocol.

SPRUG33 — *TMS320DM357 DMSoC Asynchronous External Memory Interface (EMIF) User's Guide.* Describes the asynchronous external memory interface (EMIF) in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EMIF supports a glueless interface to a variety of external devices.

SPRUG34 — *TMS320DM357 DMSoC Enhanced Direct Memory Access (EDMA) Controller User's Guide.* Describes the operation of the enhanced direct memory access (EDMA3) controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EDMA3 controller’s primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the DMSoC.

SPRUG35 — *TMS320DM357 DMSoC Audio Serial Port (ASP) User's Guide.* Describes the operation of the audio serial port (ASP) audio interface in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The primary audio modes that are supported by the ASP are the AC97 and IIS modes. In addition to the primary audio modes, the ASP supports general serial port receive and transmit operation, but is not intended to be used as a high-speed interface.

SPRUG36 — *TMS320DM357 DMSoC Ethernet Media Access Controller (EMAC)/Management Data Input/Output (MDIO) Module User's Guide.* Discusses the ethernet media access controller (EMAC) and physical layer (PHY) device management data input/output (MDIO) module in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EMAC controls the flow of packet data from the DMSoC to the PHY. The MDIO module controls PHY configuration and status monitoring.

SPRUG37 — *TMS320DM357 DMSoC Pulse-Width Modulator (PWM) Peripheral User's Guide.* Describes the pulse-width modulator (PWM) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC).

SPRUG38 — *TMS320DM357 DMSoC DDR2 Memory Controller User's Guide.* Describes the DDR2 memory controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The DDR2 memory controller is used to interface with JESD79D-2A standard compliant DDR2 SDRAM devices.

SPRUG39 — *TMS320DM357 DMSoC Video Processing Front End (VPFE) User's Guide.* Describes the video processing front end (VPFE) in the TMS320DM357 Digital Media System-on-Chip (DMSoC) video processing subsystem. Included in the VPFE is the preview engine, CCD controller, resizer, histogram, and hardware 3A (H3A) statistic generator.

SPRUGH2 —*TMS320DM357 DMSoC Peripherals Overview Reference Guide.* This document provides an overview of the peripherals in the TMS320DM357 Digital Media System-on-Chip (DMSoC).

SPRUGH3 —*TMS320DM357 DMSoC Universal Serial Bus Controller User's Guide.* This document describes the universal serial bus (USB) controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The USB controller supports data throughput rates up to 480 Mbps. It provides a mechanism for data transfer between USB devices and also supports host negotiation.

User's Guide
SPRUG39–November 2008

Video Processing Front End (VPFE)

1 Introduction

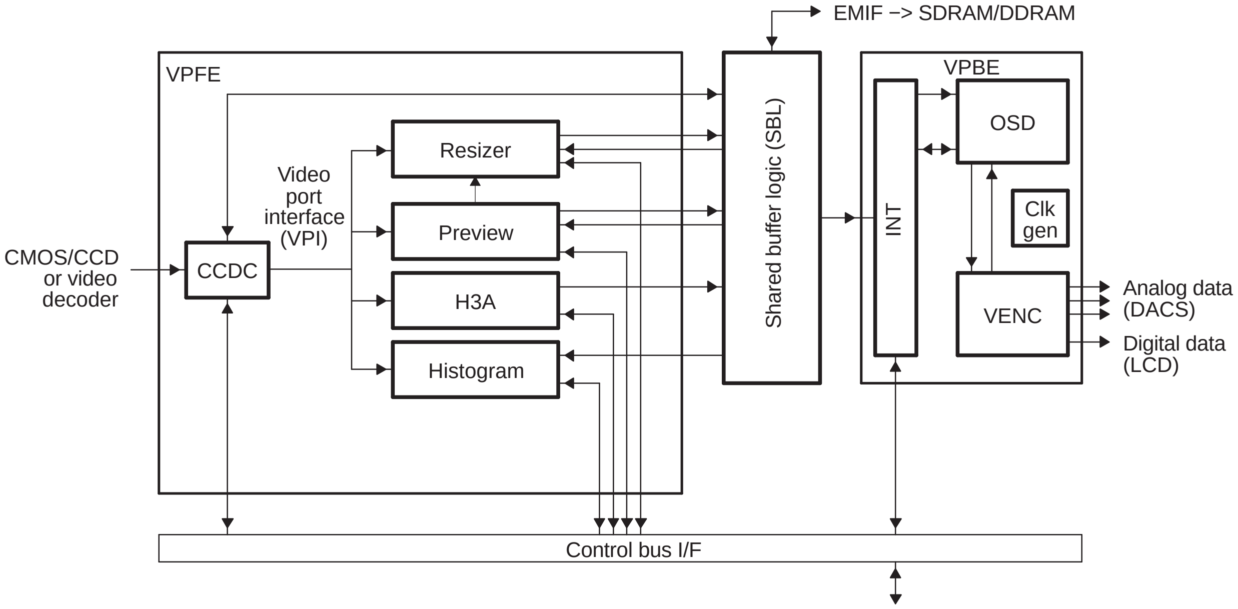
1.1 Purpose of the Video Processing Front End

The video processing subsystem (VPSS), Figure 1, provides an input interface (VPFE) for external imaging peripherals such as image sensors, video decoders, etc. and an output interface (video processing back end, (VPBE)) for display devices, such as analog SDTV displays, digital LCD panels, HDTV video encoders, etc.

There is a set of common buffer memory and DMA controls to ensure efficient use of the DDR2 burst bandwidth in addition to these peripherals. The shared buffer logic/memory is a unique block that is tailored to allow seamless integration of the VPSS into an image/video processing system. The shared buffer logic/memory acts as the primary source or sink to all of the VPFE and VPBE modules that are either requesting or transferring data to/from DDR2. In order to use the external DDR2 bandwidth efficiently, the shared buffer logic/memory interfaces with the DMA system via a high bandwidth bus (64-bit wide). The shared buffer logic/memory also interfaces with all of the VPFE and VPBE modules via a 128-bit wide bus. The shared buffer logic/memory (divided into the read and write buffers and arbitration logic) is capable of performing the following functions:

1. It is imperative that the VPSS use DDR2 bandwidth efficiently due to both its large bandwidth requirements and the real-time requirements of the VPSS modules.
2. A set of user-accessible registers is provided to monitor overflows or failures in data transfers because it is possible to configure the VPSS modules in a way that exceeds DDR2 bandwidth.

Figure 1. Video Processing Subsystem (VPSS) Block Diagram



1.2 Features

The VPFE is comprised of the CCD controller (CCDC), preview engine image pipe (IPIPE), hardware 3A statistic generator (H3A), resizer, and histogram blocks. Together, these modules provide a powerful and flexible front-end interface. These modules can be broken down into two distinct types:

- The first type consists of modules that are in the direct data flow path and affect the input image data stream:
  - The CCD controller provides an interface to image sensors and digital video sources.
  - The preview engine IPIPE is a parameterized hard-wired image processing block whose image processing functions can be customized for each sensor type to realize good image quality and video frame rates for displays and video recording modes.
  - The resizer module provides a means to size the input image data to the desired display or video encoding resolution.
- The second type consists of modules that provide statistics on the incoming images to aid camera systems designers:
  - The H3A module is designed to support the control loops for auto focus (AF), auto white balance (AWB), and auto exposure (AE) by collecting metrics on the raw image data from the CCD controller.
  - The histogram module bins input color pixels, depending on the amplitude, and provides statistics required to implement various H3A (AE/AF/AWB) algorithms and tune the final image/video output. The histogram module can operate on raw image data from CCD controller or DDR2.

1.2.1 CCD Controller (CCDC)

The CCD controller is responsible for accepting raw (unprocessed) image/video data from a sensor (CMOS or CCD). In addition, the CCD controller can accept YUV video data in numerous formats, typically from video decoder devices. In the case of raw inputs, the CCD controller output requires additional image processing to transform the raw input image to the final processed image. This additional image processing can be done either on-the-fly in the preview engine IPIPE, or in software. In parallel, raw data input to the CCD controller can also be used to compute various statistics (H3A, Histogram) to eventually control the image/video tuning parameters. The CCD controller is programmed via control and parameter registers. The following features are supported by the CCD controller module:

- Conventional Bayer pattern sensor formats
- Generates HD/VD timing signals and field ID to an external timing generator or synchronizes to the external timing generator
- Support for progressive and interlaced sensors (hardware support for up to 2 fields)
- Support for up to 90 MHZ sensor clocks
- Support for REC656/CCIR-656 standard (YCbCr 422 format, either 8-bit or 10-bit)
- Support for YCbCr 422 format, either 8- or 16-bit with discrete H and VSYNC signals
- Support for up to 16-bit input.
- Generates optical black clamping signals
- Support for shutter signal control
- Support for digital clamping and black level compensation
- Support for 10-bit to 8-bit A-law compression
- Support for a low-pass filter prior to writing to SDRAM. If this filter is enabled, 2 pixels each in the left and right edges of each line are cropped from the output
- Support for generating output to range from 16-bits to 8-bits wide (8-bits wide allows for 50% saving in storage area)
- Support for down-sampling via programmable culling patterns
- Ability to control output to the DDR2 via an external write enable signal
- Support for up to 32K pixels (image size) in both the horizontal and vertical directions

1.2.2 Preview Engine – Image Pipe (IPIPE)

The preview engine image pipe (IPIPE) is responsible for transforming raw (unprocessed) image/video data from a sensor (CMOS or CCD) into YCbCr 422 data that is amenable for compression or display. Typically, the output of the preview engine is used for both video compression and displaying it on an external display device, such as a NTSC/PAL analog encoder or a digital LCD. The preview engine is programmed via control and parameter registers. The preview engine supports the following features:

- Conventional Bayer pattern
- Accepting the input image/video data from either the CCD/CMOS controller or the SDRAM/DDRAM
- An output width of up to 1280 pixels wide
- Automatic/mandatory cropping of pixels/lines when edge processing is performed. If all of the corresponding modules are enabled, a total of 14 pixels per line (7 left-most and 7 right-most) and 8 lines (4 top-most and 4 bottom-most) will not be output. For more information, see Section 2.
- Simple horizontal averaging (by factors of 2, 4, or 8) to handle input widths that are greater than 1280 (plus the cropped number) pixels wide
- Ability to capture a dark frame (instead of applying the conventional image processing to the raw data) and store it in the SDRAM/DDRAM
- Ability to subtract a dark frame (fetched from the SDRAM/DDRAM memory) for every input raw data frame pixel-by-pixel to improve video quality
- Ability to perform lens shading compensation instead of the dark frame subtract. Each input pixel is multiplied with a corresponding 8-bit gain value and the result is right shifted by a programmable parameter (0-7 bits)
- Support for A-law decompression to transform non-linear 8-bit data to 10-bit linear data. This feature, which allows data in the SDRAM/DDRAM to be 8-bits only, saves 50% of the area if the input to the preview engine is from the SDRAM/DDRAM
- A horizontal median filter for reducing temperature induced noise in pixels
- A programmable noise filter that operates on a 3 × 3 grid of the same color (effectively, this is a five line storage requirement)
- Digital gain and white balance (color separate gain for white balance)
- Programmable CFA interpolation that operates on a 5 × 5 grid
- Programmable RGB-to-RGB blending matrix (9 coefficients for the 3 × 3 matrix)
- Fully programmable gamma correction (1024 entries for each color held in an on-chip RAM)
- Programmable color conversion (RGB to YUV) coefficients (9 coefficients for the 3 × 3 matrix)
- Luminance enhancement (non-linear) and chrominance suppression and offset

1.2.3 Resizer

The resizer module can accept input image/video data from either the preview engine or DDR2. The output of the resizer module will be sent to the SDRAM/DDRAM. The resizer module is programmed via its registers that are accessible by a host processor in the system. The resizer module supports the following features:

- Maximal output width of 1280 horizontal pixels
- Input from either the preview engine (on-the-fly processing) or from external SDRAM/DDRAM
- Support for up to 4× upsampling (digital zoom)
  - Bi-cubic interpolation (4-tap horizontal, 4-tap vertical) can be implemented with the programmable filter coefficients
  - 8 phases of the filter coefficients are supported
  - Optionally select bi-linear interpolation for the chrominance components
  - If the input source is the preview engine, this can be performed on-the-fly
- Support for up to 1/4× down-sampling (reducing image size to store more pictures in the memory card)
  - 4-tap horizontal and 4-tap vertical filter coefficients (with 8-phases) for 1× to 1/2× down-sampling
  - For 1/2× to 1/4× down-sampling, use 7-tap mode with 4-phases
  - If the input source is the preview engine, this can be performed on-the-fly

- There are further constraints for real-time preview-output resizing due to the limited on-chip memory and processing resources. Horizontal resizer stage output rate is limited to resizer_clock/2.
  - SDRAM-input path has no such restrictions.
  - For example, at a pixel clock of 75 MHZ, no upsampling of full input width can exist. Taking 3/4 of the width and upsampling by 4/3 to full width is possible. At a pixel clock of 37.5 MHZ, upsampling by 2× of the full input width is affordable. By taking 3/4 of the full width, upsampling by as much as 8/3 can occur.
- Support for resizing either YUV 422 packed data (16-bits) or color separate data (assumed to be 8-bit data) that is contiguous. The input source for the color separate data should be the DDR2.
- Separate/independent resizing factor for the horizontal and vertical directions
- Available upsampling and down-sampling ratios are: 256/N, with N ranging from 64 to 1024
- Programmable luminance sharpening after the horizontal resizing and before the vertical resizing step

1.2.4 Hardware 3A (H3A)

The H3A module is designed to support the control loops for auto focus (AF), auto white balance (AWB), and auto exposure (AE) by collecting metrics about the imaging/video data. The metrics are used to adjust the various parameters for processing the imaging/video data. There are two main blocks in the H3A module:

- Auto focus (AF) engine
- Auto exposure (AE) and auto white balance (AWB) engine

The AF engine extracts and filters the red, green, and blue data from the input image/video data and provides either the accumulation or peaks of the data in a specified region. The specified region is a two-dimensional block of data and is referred to as a paxel for the case of AF.

The AE/AWB engine accumulates the values and checks for saturated values in a sub-sampling of the video data. In the case of the AE/AWB, the two-dimensional block of data is referred to as a “window”. Thus, other than referring to them by different names, a paxel and a window are essentially the same thing. However, the number, dimensions, and starting position of the AF paxels and the AE/AWB windows are separately programmable.

1.2.4.1 Auto Focus (AF) Engine

The following features are supported by the AF engine:

- Support for a Peak Mode in a paxel (a paxel is defined as a two-dimensional block of pixels)
- Accumulate the maximum Focus Value of each line in a paxel
- Support for an accumulation/sum mode (instead of peak mode)
- Accumulate focus value in a paxel
- Support for up to 36 paxels in the horizontal direction and up to 128 paxels in the vertical direction. The number of horizontal paxels is limited by the memory size, while the vertical number of paxels is not. Therefore, the number of paxels in horizontal direction is smaller than the number of paxels in vertical direction.
- Programmable width and height for the paxel. All paxels in the frame are the same size.
- Programmable red, green, and blue position within a 2 × 2 matrix
- Separate horizontal start for paxel and filtering
- Programmable vertical line increments within a paxel
- Parallel IIR filters configured in a dual-biquad configuration with individual coefficients (2 filters with 11 coefficients each). The filters compute the sharpness/peaks in the frame to focus on.

1.2.4.2 Auto Exposure (AE) and Auto White Balance (AWB) Engine

The following features are supported by the AE/AWB engine:

- Accumulate clipped pixels along with all non-saturated pixels
- Support for up to 36 horizontal windows
- Support for up to 128 vertical windows
- Programmable width and height for the windows. All windows in the frame are the same size.
- Separate vertical start coordinate and height for a black row of paxels that is different than the remaining color paxels
- Programmable horizontal sampling points in a window
- Programmable vertical sampling points in a window

1.2.5 Histogram

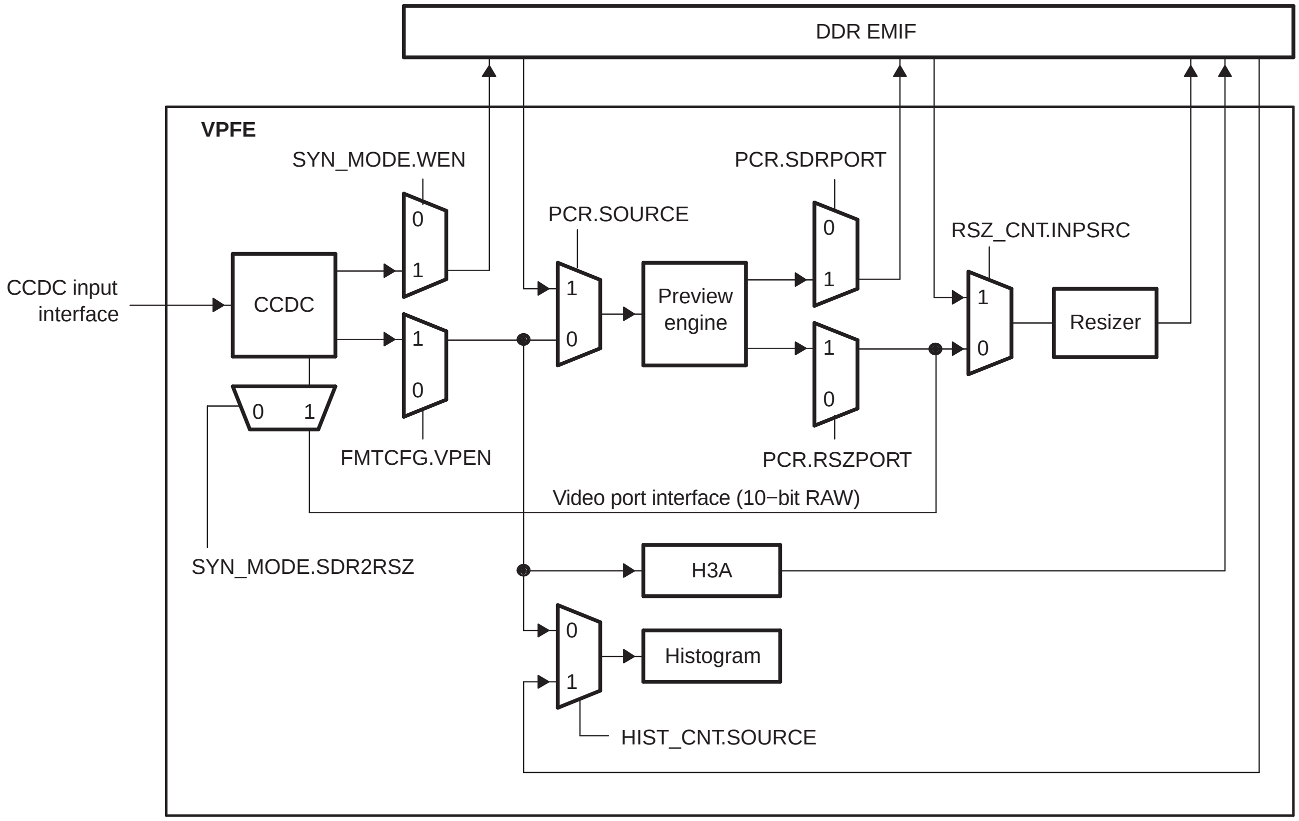
The histogram module accepts raw image/video data (either 3 or 4 colors) and bins the pixels on a value (and color separate) basis. The value of the pixel itself is not stored; but, each bin contains the number of pixels that are within the appropriate set range. The source of the raw data for the histogram is typically a CCD/CMOS sensor (via the CCD controller module) or optionally from SDRAM/DDRAM. The following features are supported by the histogram module:

- Support for up to four regions/areas
  - Each region has its own horizontal/vertical start and end position.
  - When regions overlap, pixels from the overlapped area are accumulated into the highest priority region only (the priority is region0 > region1 > region2 > region3)
- Support for conventional Bayer pattern sensors. Each region is capable of accumulating 4 colors separately.
- Support for 32, 64, 128, or 256 bins per color per region
  - If the number of regions is 1, then 32, 64, 128, or 256 bins per color is allowed.
  - If the number of regions is 2, then 32, 64, or 128 bins per color is allowed.
  - If the number of regions is 3, then 32 or 64 bins per color is allowed.
  - If the number of regions is 4, then 32 or 64 bins per color is allowed.
- Support for automatic clear of the histogram RAM once the ARM reads that location (programmable register)
- Support for saturation of the pixel count if the count exceeds the maximum value that the memory location can hold (each memory location is 20-bits wide)
- Support for a downshift ranging from 0 to 7 bits (this implies that the maximum range of each bin will be 128)
- The last bin (highest range of values) will accumulate any value that is higher than the lower bound. For example, if 32 bins are set up so that each bin accumulates a range of 8 or a downshift of 3 (0 to 7, 8 to 15, etc.), the last bin shall accumulate all values higher than 248 and not just the range of values from 248 to 255.

1.3 Functional Block Diagram

Figure 2 shows a high-level block diagram of the VPFE functional blocks, along with the different data flow paths. These data flow paths show how the various modules of the VPFE interact and the data source(s) for the statistics generation modules (H3A and histogram).

Figure 2. Video Processing Front End (VPFE) Block Diagram



1.4 Use Case Statement

The VPFE supports image data acquisition from sensor and digital video sources in various modes/formats. YUV sources have minimal image processing applied and can either be passed directly to external memory/DDR2 or passed to the resizer for scaling prior to writing to DDR2. Raw imager data modes (non-YUV sources) are supported by the statistics collection modules (H3A and histogram) as well as full preview engine image signal processing functions, plus resizing after preview.

The same processing options are supported when processing data sourced from DDR2. The only exception is that the H3A module cannot operate on data from DDR2.

Zooming at ratios greater than the 4× ratio in a single pass are not supported by the resizer. However, this can be done by passing the resized data from DDR2 through the resizer again as long as the real-time deadlines can be met. This will be discussed in more detail in Section 5.6.5.1.

2 Camera Subsystem Environment

The VPFE interface signals are shown in Table 1.

Note: These signals can take on different meanings for the DMSoC, depending on the specific interface chosen. Pin multiplexing is controlled from the System module. The following sections describe each of the supported scenarios.

Table 1. Interface Signals for Video Processing Front End

Pin Name	I/O	Description	Default	Tristate	Pullup/Pulldown
PCLK	In	Pixel Clock	In	-	-
VD	Bi	V sync	In	tri	-
HD	Bi	H sync	In	tri	-
CI7/CCD15	Bi	C IN signal/CCD in signal	In	tri	PD
CI6/CCD14	Bi	C IN signal/CCD in signal	In	tri	PD
CI5/CCD13	Bi	C IN signal/CCD in signal	In	tri	PD
CI4/CCD12	Bi	C IN signal/CCD in signal	In	tri	PD
CI3/CCD11	In	C IN signal/CCD in signal	In	-	PD
CI2/CCD10	In	C IN signal/CCD in signal	In	-	PD
CI1/CCD9	In	C IN signal/CCD in signal	In	-	PD
CI0/CCD8	In	C IN signal/CCD in signal	In	-	PD
YI7/CCD7	In	Y IN signal/CCD in signal	In	-	PD
YI6/CCD6	In	Y IN signal/CCD in signal	In	-	PD
YI5/CCD5	In	Y IN signal/CCD in signal	In	-	PD
YI4/CCD4	In	Y IN signal/CCD in signal	In	-	PD
YI3/CCD3	In	Y IN signal/CCD in signal	In	-	PD
YI2/CCD2	In	Y IN signal/CCD in signal	In	-	PD
YI1/CCD1	In	Y IN signal/CCD in signal	In	-	PD
YI0/CCD0	In	Y IN signal/CCD in signal	In	-	PD
C_WE	Bi	CCD Write Enable signal	In	tri	-
C_FIELD/R0	Bi	CCD Field signal/R0 (VPBE)	In	tri	-

2.1 Parallel Generic Configuration (Raw)

The generic raw interface configuration is typically used for interfacing to image sensors. The VPFE supports up to 16 bits of resolution for each sample, but sensors typically only output 8, 10, 12, or 14 bits of useful resolution, depending on the imager and the associated AFE.

2.1.1 Parallel Generic Configuration (Raw) Signal Interface

Table 2 shows the interface connections for the Raw Mode interface. The device can support up to 16 bits of resolution for each sample but sensors typically only output 8, 10, 12, or 14 bits of useful resolution depending on the imager and associated AFE. When the number of data lines is less than 16, it is recommended to connect the Raw data to the lower data lines of CCD[15-0]. Then the SYN_MODE.DATSIZ register can be used to indicate the bit size of the input so that the hardware ignores the upper bits that are not connected.

Table 2. Interface Signals for Raw Mode

Pin Name	Description
PCLK	Pixel Clock
VD	V sync
HD	H sync
CCD15	CCD in signal
CCD14	CCD in signal
CCD13	CCD in signal
CCD12	CCD in signal
CCD11	CCD in signal
CCD10	CCD in signal
CCD9	CCD in signal
CCD8	CCD in signal
CCD7	CCD in signal
CCD6	CCD in signal
CCD5	CCD in signal
CCD4	CCD in signal
CCD3	CCD in signal
CCD2	CCD in signal
CCD1	CCD in signal
CCD0	CCD in signal
C_WE	CCD Write Enable signal
C_FIELD/R0	CCD Field signal/R0 (VPBE)

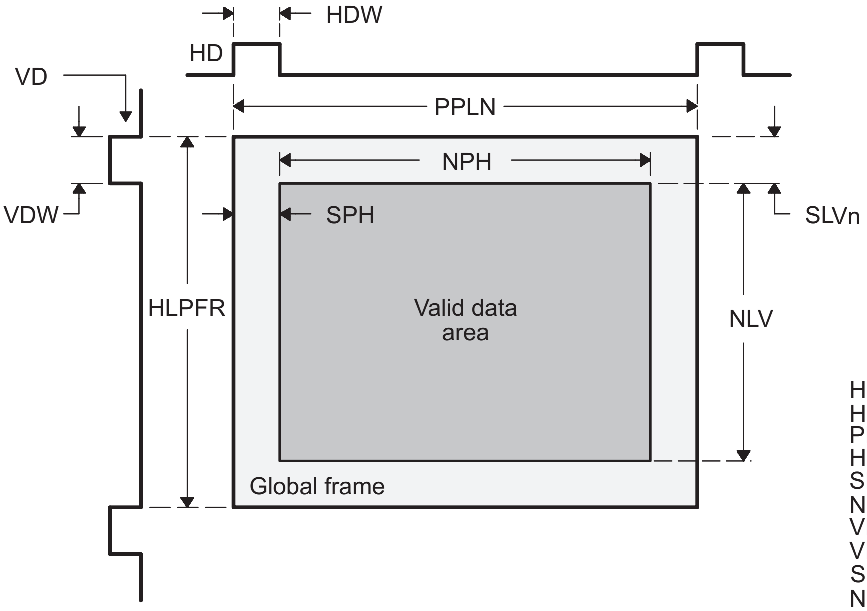
2.1.2 Parallel Generic Configuration (Raw) Signal Interface Description

The VPFE can either generate the H/V sync signals needed to interface with sensors or source them from the sensor or timing generator. The PCLK or the pixel clock must always be provided as an input.

2.1.3 Parallel Generic Configuration (Raw) Protocol and Data Formats

The timing generator in the CCD controller either enables the use of external sync signals (HD/VD) or internally-generated timing signals. Figure 3 shows various CCD controller register settings related to the timing. The shaded area is the physical imager size and the gray area is the valid data area. The image data in this area is processed and stored to external SDRAM/DDRAM or sent out to the various VPFE modules. The vertical start position for even and odd fields can be configured independently.

Figure 3. Raw Mode Timing Diagram



HDW − Horizontal data width
HD − Horizontal data
PPLN − Pixels per line
HLPFR – Half lines per frame
SPH − Start pixel horizontal
NPH − Number of pixels horizontal
VDW − Vertical data width
VD − Vertical data
SLVn – Start line vertical field 0 or field 1
NLV − Number of lines vertical

The bits of data from each pixel are stored in the lower bits of a 16-bit SDRAM word, and the unused bit positions are filled with zeros. The DDR data format is shown in Figure 4. There is an optional 10-bit to 8-bit A-Law compression so that 10-bit data can be reduced to 8-bit dynamic range and packed to save DDR memory usage.

Figure 4. DDR2 Output Format

	Upper Word		**Lower Word**	
	MSB (31)	LSB (16)	MSB (15)	LSB (0)
16 bit		Pixel1		Pixel0
15 bit	0	Pixel1	0	Pixel0
14 bit	0	Pixel1	0	Pixel0
13 bit	0	Pixel1	0	Pixel0
12 bit	0	Pixel1	0	Pixel0
11 bit	0	Pixel1	0	Pixel0
10 bit	0	Pixel1	0	Pixel0
9 bit	0	Pixel1	0	Pixel0
8 bit	0	Pixel1	0	Pixel0
8-bit pack	Pixel3	Pixel2	Pixel1	Pixel0

2.2 ITU-R BT.656 Configuration Functional Interface

ITU-R BT.656 (sometimes referred to as either CCIR-656 or REC656) is a specification that provides a method to transfer YCbCr-4:2:2 formatted digital video data over an 8/10-bit wide interface.

2.2.1 ITU-R BT.656 Configuration Signal Interface

Table 3 shows the interface connections for the ITU-R BT.656 interface.

Data and timing codes are transferred over the same 8/10-bit interface. When in BT.656 mode, only the data lines and clock signal are connected between the external device and the CCD controller module of the VPFE. An NTSC/PAL decoder is an example of an external device that may be connected to the CCIR-656 interface.

Data lines CCD[7:0] are used for 8-bit YCbCr data and data lines CCD[9:0] are used for 10-bit YCbCr data. The video timing signals, HD, VD, and FIELD are generated internally by the CCD controller module of the VPFE.

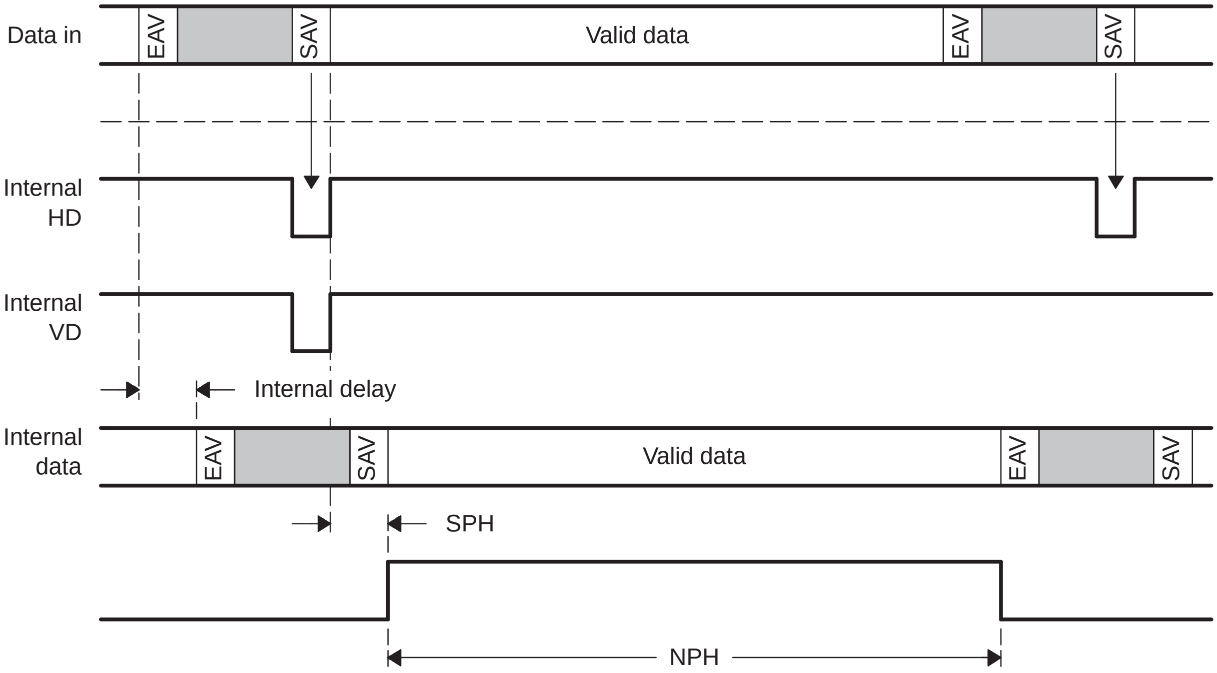
Table 3. Interface Signals for ITU-R BT.656 Mode

Pin Name	Description
PCLK	Pixel Clock
CCD9	CCD Data/BT.656 Data (optional, for 10-bit interface)
CCD8	CCD Data/BT.656 Data (optional, for 10-bit interface)
CCD7	CCD Data/BT.656 Data
CCD6	CCD Data/BT.656 Data
CCD5	CCD Data/BT.656 Data
CCD4	CCD Data/BT.656 Data
CCD3	CCD Data/BT.656 Data
CCD2	CCD Data/BT.656 Data
CCD1	CCD Data/BT.656 Data
CCD0	CCD Data/BT.656 Data

2.2.2 ITU-R BT.656 Configuration Signal Interface Description

Two timing reference codes synchronize HD, VD, and FIELD to the video data. At the start and end of each video data block, the device sends a unique timing reference code. The start code is called the start of active video signal (SAV), and the end code is called the end of active video signal (EAV). The SAV and EAV codes proceed and follow valid data, as shown in Figure 5. HD, VD, and FIELD are generated internally by the CCD controller, based on the SAV and EAV codes. Other CCD controller register settings allow you to control when to read/save valid data to DDR.

Figure 5. BT.656 Signal Interface



2.2.3 ITU-R BT.656 Configuration Protocol and Data Formats

Both timing reference signals, SAV and EAV, consist of a four word sequence in the following format: FF 00 00 XY, where FF 00 00 are a set preamble and the fourth word defines the field identification, the state of vertical field blanking, the state of horizontal line blanking, and protection (error correction) codes. The bit format of the fourth word is shown in Table 4 and the definitions for bits, F, V, and H, are given in Table 5. F, V, and H are used in place of the usual horizontal sync, vertical sync, and blank timing control signals. Bits P3, P2, P1, and P0 are protection (error correction) bits for F, V, and H. The relationship between F, V, and H and the protection (error correction) bits is given in Table 6. To enable error correction, set the ECCFVH bit in the REC656IF register to 1. The CCD controller will automatically detect and apply error correction when the ECCFVH bit is enabled.

When operating in CCIR-656 mode, data is stored in SDRAM according to the format shown in Figure 6 when the PACK8 bit in SYN_MODE is set to 1.

Note that the CCD controller outputs the XY code in the SAV and EAV into the SDRAM. In order to eliminate this, you should set the SPH field in HORZ_INFO to SPH + 1. In addition, the NPH field in HORZ_INFO should be set to accurately represent the number of active pixels.

Table 4. Video Timing Reference Codes for SAV and EAV

Data Bit Number	First Word (FF)	Second Word (00)	Third Word (00)	Fourth Word (XY)
9 (MSB)	1	0	0	1
8	1	0	0	F
7	1	0	0	V
6	1	0	0	H
5	1	0	0	P3
4	1	0	0	P2
3	1	0	0	P1
2	1	0	0	P0
1	1	0	0	0
0	1	0	0	0

Table 5. F, V, H Signal Descriptions

Signal	Value	Command
F	0	Field 1
	1	Field 2
V	0	0
	1	Vertical blank
H	0	SAV
	1	EAV

Table 6. F, H, V Protection (Error Correction) Bits

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

Figure 6. BT.656 Mode Data Format in SDRAM

	31			0
8 bit × 4	Pixel3 (Y1/Cr0)	Pixel2 (Cr0/Y1)	Pixel1 (Y0/Cb0)	Pixel0 (Cb0/Y0)

2.3 Generic YUV Interface

The CCD controller can accept generic YCbCr-4:2:2 formatted digital video data over an 8/16-bit wide interface..

2.3.1 Generic YUV Configuration Signal Interface

Table 7 shows the interface connections for the generic YUV interface.

Unlike the BT.656 mode, discrete HD and VD signals are required. An example of an external device that may be connected to the YUV interface is the NTSC/PAL decoder.

In 8-bit mode, data lines YI[7:0] or CI[7:0] can be used for input. When using an 8-bit interface, the YI[7:0] inputs are typically used; however, either set of data inputs can be used. Alternately, two separate imagers can be physically connected (but only one can be active at any given time). The YCINSWP bit in CCDCFG determines which set of 8-bit inputs are active.

In 16-bit mode, data lines YI[7:0] and CI[7:0] are used for input with the Cr/Cb data multiplexed on the CI[7:0] signals. The YCINSWP bit in CCDCFG is used to swap the Y and Cr/Cb data lines.

Table 7. Interface Signals for YUV Interface

Pin Name	Description
PCLK	Pixel Clock
VD	V sync
HD	H sync
CI7	C IN/Y IN
CI6	C IN/Y IN
CI5	C IN/Y IN
CI4	C IN/Y IN
CI3	C IN/Y IN
CI2	C IN/Y IN
CI1	C IN/Y IN
CI0	C IN/Y IN
YI7	Y IN/C IN
YI6	Y IN/C IN
YI5	Y IN/C IN
YI4	Y IN/C IN
YI3	Y IN/C IN
YI2	Y IN/C IN
YI1	Y IN/C IN
YI0	Y IN/C IN
C_FIELD/R0	CCD Field signal/R0 (VPBE)

2.3.2 Generic YUV Configuration Protocol and Data Formats

In 8-bit mode, the position on the Y data in relation to Cr/Cb data is configurable by the Y8POS bit setting in CCDCFG. The byte ordering of data is swapped by the BSWD bit setting in CCDCFG. For 8-bit inputs, the PACK8 bit in SYN_MODE must be set to pack the data in SDRAM properly.

Table 8. DDR Storage Format for YCbCr Processing

	SDRAM Data Format			
	Upper word		Lower word	
SDRAM Address	MSB (31)	LSB (16)	MSB (15)	LSB (0)
N	Y1	Cr0	Y0	Cb0
N + 1	Y3	Cr2	Y2	Cb2
N + 2	Y5	Cr4	Y4	Cb4

2.4 *VPFE/Camera Subsystem I/O Multiplexing*

The various VPFE imager interface modes have unique pin multiplexing options as shown in Table 9. Some of these settings are controlled in the system module. The remaining settings are controlled via the mode in which the controller is placed.

Table 9. Signals for VPFE Imager Interface Modes

Pin Name	PRGB	YCC16	YCC8	REC656
PCLK	PCLK	PCLK	PCLK	PCLK
VD	VD	VD	VD	-
HD	HD	HD	HD	-
CI7	D[15]	CI[7] / YI[7]	Y7,Cb7,Cr7	-
CI6	D[14]	CI[6] / YI[6]	Y6,Cb6,Cr6	-
CI5	D[13]	CI[5] / YI[5]	Y5,Cb5,Cr5	-
CI4	D[12]	CI[4] / YI[4]	Y4,Cb4,Cr4	-
CI3	D[11]	CI[3] / YI[3]	Y3,Cb3,Cr3	-
CI2	D[10]	CI[2] / YI[2]	Y2,Cb2,Cr2	-
CI1	D[9]	CI[1] / YI[1]	Y1,Cb1,Cr1	Y9,Cb9,Cr9
CI0	D[8]	CI[0] / YI[0]	Y0,Cb0,Cr0	Y8,Cb8,Cr8
YI7	D[7]	YI[7] / CI[7]	Y7,Cb7,Cr7	Y7,Cb7,Cr7
YI6	D[6]	YI[6] / CI[6]	Y6,Cb6,Cr6	Y6,Cb6,Cr6
YI5	D[5]	YI[5] / CI[5]	Y5,Cb5,Cr5	Y5,Cb5,Cr5
YI4	D[4]	YI[4] / CI[4]	Y4,Cb4,Cr4	Y4,Cb4,Cr4
YI3	D[3]	YI[3] / CI[3]	Y3,Cb3,Cr3	Y3,Cb3,Cr3
YI2	D[2]	YI[2] / CI[2]	Y2,Cb2,Cr2	Y2,Cb2,Cr2
YI1	D[1]	YI[1] / CI[1]	Y1,Cb1,Cr1	Y1,Cb1,Cr1
YI0	D[0]	YI[0] / CI[0]	Y0,Cb0,Cr0	Y0,Cb0,Cr0
GPIO1	C_WE	C_WE	C_WE	-
GPIO4	C_FIELD	C_FIELD	C_FIELD	-

2.4.1 Y/C Data BUS Swap

There is an option to swap the upper and lower portion of the 16-bit YUV data bus (using the YCINSWP bit in CCDCFG). This swaps the luma and chroma samples in 16-bit YUV mode. Swapping portions of the 16-bit YUV data bus determines which half of the bus is used as the input source in 8-bit mode and can be used in 8-bit YUV mode to support two separate YUV input ports. It cannot be used in REC656 mode.

2.4.2 CCD and LCD Control Signal Multiplexing

The CCD and LCD controllers in the VPSS require additional control signals for certain modes of operation. Each of these signals has a separate enable bit in the PINMUX0 register which selects between the control signal function and GPIO. These are summarized in Table 10.

Table 10. CCD/LCD Supplemental Control Multiplexing

PINMUX0 Register Bit				Pin Functions			
CFLDEN	**LFLDEN**	**CWEN**	**LOEEN**	**C_FIELD/R0/ GPIO[4]**	**LCD_FIELD/B0/ GPIO[3]**	**C_WE/ GPIO[1]**	**LCD_OE/ GPIO[0]**
x	x	x	0	x	x	x	GPIO[0]
x	x	x	1	x	x	x	LCD_OE
x	x	0	x	x	x	GPIO[1]	x
x	x	1	x	x	x	C_WE	x
x	0	x	x	x	B0/GPIO[3]	x	x
x	1	x	x	x	LCD_FIELD	x	x
0	x	x	x	R0/GPIO[4]	x	x	
1	x	x	x	C_FIELD	x	x	x

3 Integration

This section describes how the VPFE subsystem is integrated into the DMSoC.

3.1 Clocking, Reset and Power Management Scheme

3.1.1 Clocks

The VPFE module is a DMA master and resides in the CLKDIV3 clock domain. This clock is the VPSSmstr module in the Power and Sleep Controller (PSC) and it is shared with the video processing back end (VPBE). Thus, this clock can be gated off to conserve power, but this also precludes use of the VPBE. The VPFE modules utilize auto clock gating on a clock-by-clock basis to conserve dynamic power during periods of inactivity. Note that the clock should only be disabled when the VPFE is not operational. The clocks should be enabled prior to any other operations on the VPFE (including reading/writing other registers).

The VPFE module also includes a Slave Port for the control registers that resides in the CLKDIV6 clock domain.

An external pixel clock is required to use the CCD/CMOS controller to bring data into the VPFE from the external sensor interface. To ensure proper timing, the maximum pixel clock period must be 1 ns more than 2× the period of the VPFE clock, or CLKDIV3. The maximum pixel clock rate cannot be more than 90 MHZ. (See the DM357 datasheet for supported maximum pixel rate information.)

Note that there is an option to drive the VPBE module with the VPFE pixel clock.

3.1.2 Resets

The VPFE module resets are tied to the device reset signals.

In addition, the VPFE modules can be reset by transitioning to the SyncReset state of the PSC. Note that the VPFE has two module domains, the VPSSmstr processing domain and the VPSSslv register interface.

3.1.3 Power Domain and Power Management

The VPFE module resides in the “Always On” power domain, along with the ARM core and most other peripherals. When powered, the VPFE modules utilize auto clock gating on a clock-by-clock basis to conserve dynamic power during periods of inactivity.

As a result, there is no way to disable the VPFE module using the Power and Sleep Controller (PSC) and the PSC states of Disable and SwRstDisable have no meaning.

3.2 Hardware Requests

3.2.1 Interrupt Requests

The VPFE generates interrupts to the ARM as shown in Table 11. With the exception of the CCD controller interrupts, these all indicate an end-of-frame event (that is, processing has completed for a frame). These can be unique per VPFE module because each can have different processing limits. For more details on interrupts, see Section 5.

Table 11. ARM Interrupts

INT Number	Acronym	Source
0	VDINT0	VPSS - CCD Controller
1	VDINT1	VPSS - CCD Controller
2	VDINT2	VPSS - CCD Controller
3	HISTINT	VPSS - Histogram
4	H3AINT	VPSS – AE/AWB/AF
5	PRVUINT	VPSS - Previewer
6	RSZINT	VPSS - Resizer

3.2.2 EDMA Requests

There are four VPFE-related EDMA events. These all indicate an end-of-frame event (that is, processing has completed for a frame).

Table 12. EDMA Events

Event Number	Binary	Event Name
4	000 0100	VPSS : HISTEVT
5	000 0101	VPSS : H3AEVT
6	000 0110	VPSS : PRVUEVT
7	000 0111	VPSS : RSZEVT

The purpose of routing the events to the EDMA is to allow for update of the module registers by using the DMA scheme versus direct CPU write. Normally, the ARM performs this function, but in some cases the ARM can be tied up with other activities and the interrupt latency is critical when dealing with the VPFE modules. Consider the following example; a digital zoom of 10× is required. The resizer needs to be operated in two passes, the first pass using input from the preview engine and the second pass using input from the SDRAM/DDR (using the output of the first pass). The resizer registers need to be altered as soon as the first pass is complete. Tying the resizer event to the EDMA allows instantaneous DMA of the new register settings for the second resize pass. For more details on this example, see the resizer programming model in Section 5.6.5.1.

3.3 VPFE Top-Level Register Mapping Summary

The register mapping for the VPFE is shown in Table 13.

Table 13. VPFE Module Register Map

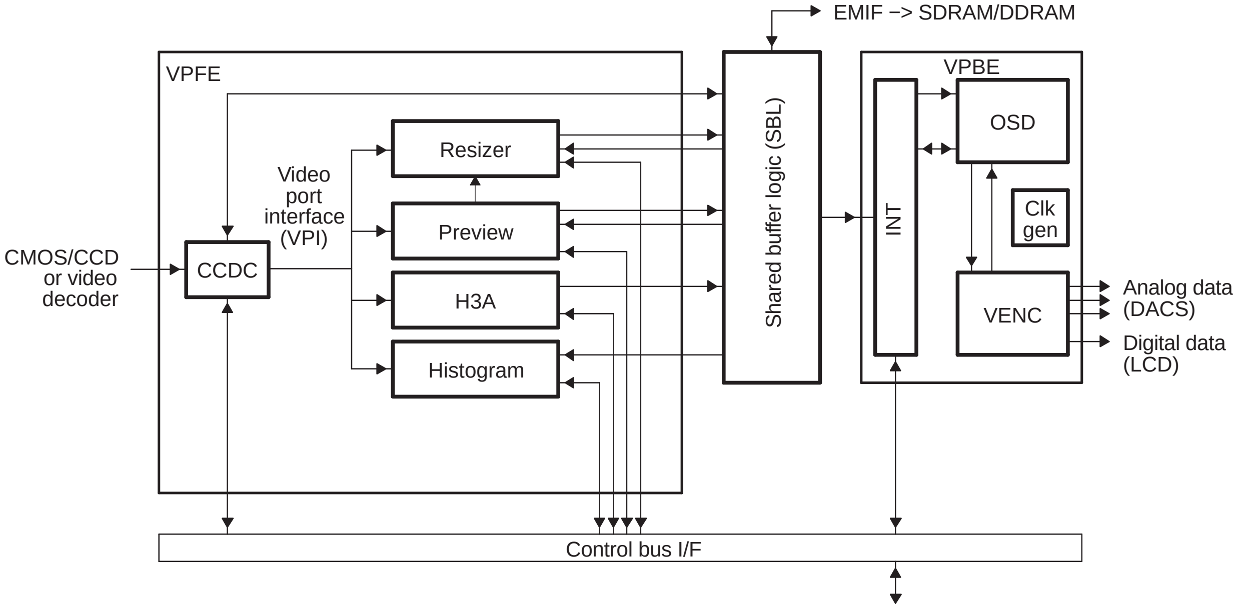
VPSS Registers	Address Range		Size
Reserved	01C7 0000h	01C7 03FFh	1K
CCD Controller	01C7 0400h	01C7 07FFh	1K
Preview Engine	01C7 0800h	01C7 0BFFh	1K
Resizer	01C7 0C00h	01C7 0FFFh	1K
Histogram	01C7 1000h	01C7 13FFh	1K
Hardware 3A	01C7 1400h	01C7 17FFh	1K

4 Functional Description

4.1 Block Diagram

The VPSS block diagram is shown in Figure 7. Additional detailed block diagrams are shown in the Interface and Image Processing subsections.

Figure 7. Video Processing Subsystem Block Diagram



4.2 *Interfacing with Image Sensors*

The CCD controller supports sensor/image input interfaces and they are described in the following sections. See the programming model descriptions (Section 5) for further information on configuring the CCD controller for operation in the various modes.

4.2.1 Generic Parallel Interface – Raw CCD Data

The generic parallel or raw CCD interface supports up to a 16-bit data path to CMOS or CCD sensors (CCD + AFE) sensors. The signal interface is described in Table 14.

Table 14. CCD Interface Signals

Name	I/O	Function
CCD[15:0]	I	Image Data – mode set by SYN_MODE.INPMOD (not REC656IF.R656ON) • Bit width can be configured between 8 bits and 16 bits (SYN_MODE.DATSIZ). • The polarity of the input image data can be reversed (SYN_MODE.DATAPOL).
VD	I/O	VSYNC - Vertical sync signal • This signal can be configured as an input or an output (SYN_MODE.VDHDOUT). • When configured as an input, the CCD or CMOS sensor must supply the VD signal. • When configured as an output, the VPFE will supply the VD signal and the VD width (HD_VD_WID.VDW) and lines per frame (PIX_LINES.HLPFR) must be configured • The polarity of VD can be reversed (SYN_MODE.VDPOL).
HD	I/O	HSYNC - Horizontal sync signal • This signal can be configured as an input or an output (SYN_MODE.VDHDOUT). • When configured as an input, the CCD or CMOS sensor must supply the HD signal. • When configured as an output the VPFE will supply the HD signal and the HD width (HD_VD_WID.HDW) and pixels per line (PIX_LINES.PPLN) must be configured. • The polarity of HD can be reversed (SYN_MODE.HDPOL).
C_FIELD	I/O	Field identification signal (optional – SYN_MODE.FLDMODE) • This signal can be configured as an input or an output (SYN_MODE.FLDOUT). • When configured as an input, the CCD or CMOS sensor must supply the field identification signal. • When the field identification signal is set to be an input to the VPFE, this signal can be configured to be latched by the VD signal (CCDCFG.FIDMD). • When configured as an output, the VPFE will supply the field identification signal. • The polarity of the field identification signal can be reversed (SYN_MODE.FLDPOL).
C_WE	I	CCD Write Enable signal (optional – SYN_MODE.EXWEN) • This signal will determine when data is captured/processed/saved to memory or sent for further processing. • If enabled (SYN_MODE.EXWEN), image data will only be captured/processed/saved to memory or sent for further processing depending on the state of CCDCFG.WENLOG. • Data can be saved when either C_WE is active AND the pixels are within the internal frame (HORZ_INFO.SPH, HORZ_INFO.NPH, VERT_START.SLV, VERT_LINES.NLV) or data can be saved only when C_WE is active OR the pixels are within the internal frame (CCDCFG.WENLOG).
PCLK	I	Pixel clock • This signal is the pixel clock used to load image data into the CCD controller. • The CCD controller can be configured to capture on either the rising or falling edge of the PCLK signal (PCLKINV in System module). • The maximum pixel clock rate is 90 MHZ (generically, 1 ns more than 2× the period of the VPFE clock, or CLKDIV3. See the device data sheet for supported maximum pixel rate information.)

As described in Table 14, the VPFE can be separately configured to either source or sink the VD/HD, and Field ID signals. If any of these signals are sourced, then the VPFE CCD controller must be configured via register settings for the proper timing generation. The definition of the captured frame must be set regardless of the control signal settings but these settings are described in the data processing section.

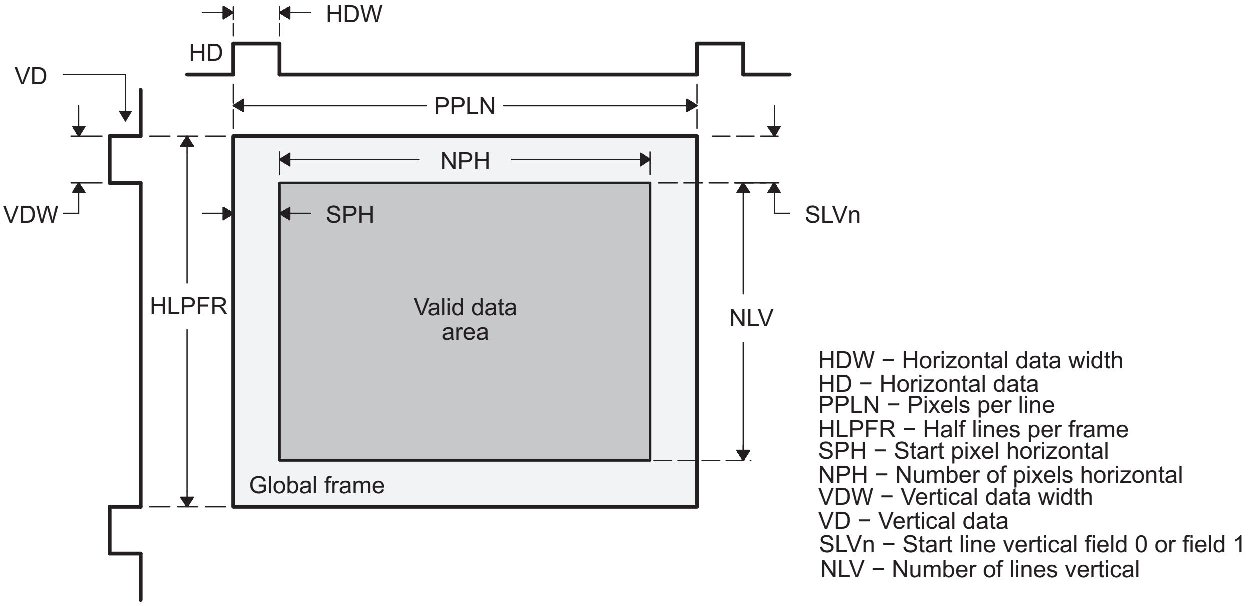
4.2.1.1 Mode Information – Always Required

- SYN_MODE.INPMOD – input mode
- SYN_MODE.DATSIZ – size (bit width) of input data – always stored in LSBs
- SYN_MODE.DATAPOL – polarity of input data
- SYN_MODE.VDPOL – VD polarity
- SYN_MODE.HDPOL – HD polarity
- SYN_MODE.VDHDOUT – VD/HD signal direction
- SYN_MODE.FLDMODE – Field mode
- SYN_MODE.FLDOUT – C_Field signal direction

4.2.1.2 Timing Information – Optional, Depending on Control Signals and Sensor Mode

- If SYN_MODE.FLDMODE is enabled:
  - SYN_MODE.FLDOUT – C_FIELD signal direction
  - SYN_MODE.FLDPOL – C_FIELD polarity
  - CCDCFG.FIDMD – C_FIELD latch information
- If SYN_MODE.VDHDOUT is output:
  - HD_VD_WID.VDW – VD width
  - PIX_LINES.HLPFR – Half lines per frame
  - HD_VD_WID.HDW – HD width
  - PIX_LINES.PPLN – Pixels per line
- If SYN_MODE.EXWEN is enabled (external C_WE signal):
  - CCDCFG.WENLOG – determines when data is valid along with frame settings

Figure 8. CCD Controller Frame and Control Signal Definitions


HDW
HD
VD
PPLN
NPH
VDW
SPH
SLVn
HLPFR
Valid data area
NLV
Global frame
HDW − Horizontal data width
HD − Horizontal data
PPLN − Pixels per line
HLPFR − Half lines per frame
SPH − Start pixel horizontal
NPH − Number of pixels horizontal
VDW − Vertical data width
VD − Vertical data
SLVn − Start line vertical field 0 or field 1
NLV − Number of lines vertical

4.2.2 ITU-R BT.656 Interface

The REC656 interface supports either 8-bit or 10-bit processing if input video YUV data. See the programming guide section for configuring this mode.

Since the sync information is carried along with the data lines, there are no sync signal interfaces or CCD controller configuration settings to make other than the start/end pixels and the line length and vertical frame size.

The signal interface is described in Table 15.

Table 15. ITU-R BT.656 Interface Signals

Name	I/O	Function
CCD[9:0]	I	Image Data – mode set by REC656IF.R656ON • Bit width can be configured to either 8 bits or 10 bits (CCDCFG.BW656). • The polarity of the input image data can be reversed (SYN_MODE.DATAPOL).
PCLK	I	Pixel clock • This signal is the pixel clock used to load image data into the CCD controller. • The CCD controller can be configured to capture on either the rising or falling edge of the PCLK signal (PCLKINV in System module). • The maximum pixel clock rate is 90 MHZ (generically, 1 ns more than 2× the period of the VPFE clock, or CLKDIV3. See the device data sheet for supported maximum pixel rate information.)

4.2.3 Digital YUV Interface

The digital YUV interface supports either 8-bit or 16-bit devices. The signal interface is described in Table 16.

Table 16. CCD Interface Signals

Name	I/O	Function
CCD[15:0] = YI[7:0]/CI[7:0]	I	Image Data – mode set by SYN_MODE.INPMOD (not REC656IF.R656ON) • Bit width can be configured to either 8 bits or 16 bits (SYN_MODE.INPMOD). • The polarity of the input image data can be reversed (SYN_MODE.DATAPOL). • When 16-bit interface is used, the Y and C inputs can be swapped (CCDCFG.YCINSWP). • When 8-bit interface is used, either half of the bus can be connected (CCDCFG.YCINSWP). • When 8-bit interface is used, the position of the Y data can be set to either the even or odd pixel (CCDCFG.Y8POS).
VD	I/O	VSYNC - Vertical sync signal • This signal can be configured as an input or an output (SYN_MODE.VDHDOUT). • When configured as an input, the signal source sensor must supply the VD signal. • When configured as an output, the VPFE will supply the VD signal and the VD width (HD_VD_WID.VDW) and lines per frame (PIX_LINES.HLPFR) must be configured. • The polarity of VD can be reversed (SYN_MODE.VDPOL).
HD	I/O	HSYNC - Horizontal sync signal • This signal can be configured as an input or an output (SYN_MODE.VDHDOUT). • When configured as an input, the signal source must supply the HD signal. • When configured as an output, the VPFE will supply the HD signal and the HD width (HD_VD_WID.HDW) and pixels per line (PIX_LINES.PPLN) must be configured. • The polarity of HD can be reversed (SYN_MODE.HDPOL).
C_FIELD	I/O	Field identification signal (optional – SYN_MODE.FLDMODE) • This signal can be configured as an input or an output (SYN_MODE.FLDOUT). • When configured as an input, the signal source must supply the field identification signal. • When the field identification signal is set to be an input to the VPFE, this signal can be configured to be latched by the VD signal (FIDMD). • When configured as an output, the VPFE will supply the field identification signal. • The polarity of the field identification signal can be reversed (SYN_MODE.FLDPOL).
PCLK	I	Pixel clock • This signal is the pixel clock used to load image data into the CCD controller. • The CCD controller can be configured to capture on either the rising or falling edge of the PCLK signal (PCLKINV in SYSTEM module). • The maximum pixel clock rate is 90 MHZ (generically, 1 ns more than 2× the period of the VPFE clock, or CLKDIV3. See the device data sheet for supported maximum pixel rate information).

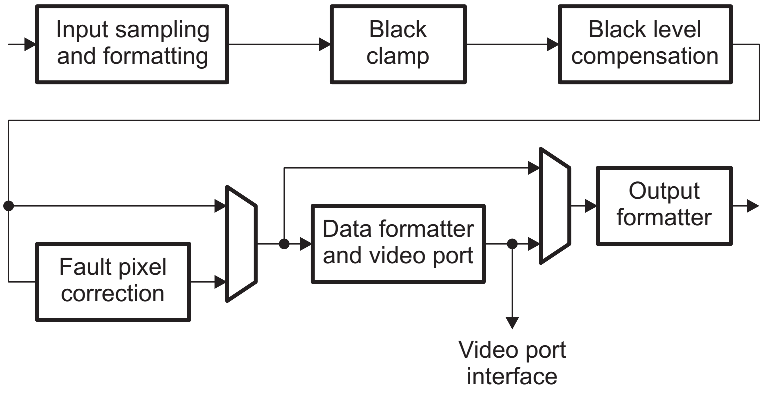
4.3 VPFE Data/Image Processing

This section describes the image/data processing of each module in the VPFE in more detail.

4.3.1 CCD Controller Processing – Raw Data Mode

The CCD controller interfaces with external image sources, not just CCD sensors. It supports both raw Bayer data from CCD/CMOS sensors and processed YUV data from either a CMOS sensor with integrated image processing or a Video Decoder interface. A high-level block diagram of the CCD controller is shown in Figure 9. Depending on the input data type, some blocks may not be applied and others must be explicitly enabled or disabled.

Figure 9. CCD Controller Processing Block Diagram – Raw Data Mode



The following sections describes the CCD controller processing for the CCD/CMOS Raw Data input mode (SYN_MODE.INPMODE = 0 && REC656IF.REC656ON = 0) in more detail. In this mode, raw sensor data, typically one color per pixel in a color filter array (or CFA) at 8 to 6 bits in dynamic range is input (usually between 10-14 bits). The color filter array applied is typically a Bayer pattern as show in Figure 10 for RGB color space.

Figure 10. CCD Controller Color Patterns

R	Gr
Gb	B

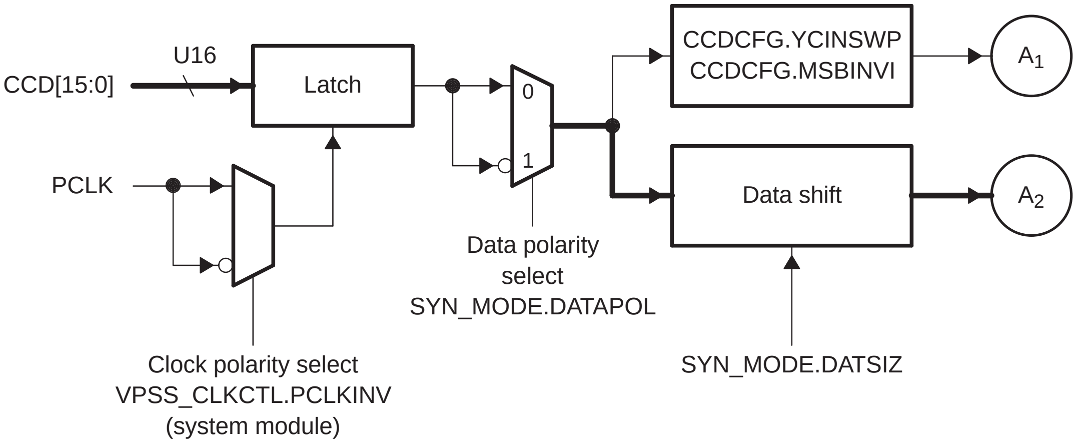
Bayer format with R/Gr and Gb/B in alternate lines
−Horizontal distance between same colors is 2.

4.3.1.1 CCD Controller Input Sampling and Formatting – Raw Data Mode

The CCD controller input sampling is shown in Figure 11. The bold data path (A_2 output) is the raw data path through the CCD controller; the upper data path is only applicable to YUV input modes.

- Data is latched by the pixel clock
- Pixel clock polarity can be either rising or falling edge. This is set in the System module via the register field: VPSS_CLKCTL.PCLKINV
- Data can be interpreted as either normal or inverted (SYN_MODE.DATAPOL)
- Data is right-shifted to align the data in the least significant bits of the data bus and provide the maximum dynamic range for the remainder of the processing (SYN_MODE.DATSIZ). This also sets the maximum data size allowed in subsequent clipping/limiting operations and is the output data alignment if data is written to DDR2.

Figure 11. CCD Controller Input Sampling Block Diagram – Raw Data Mode

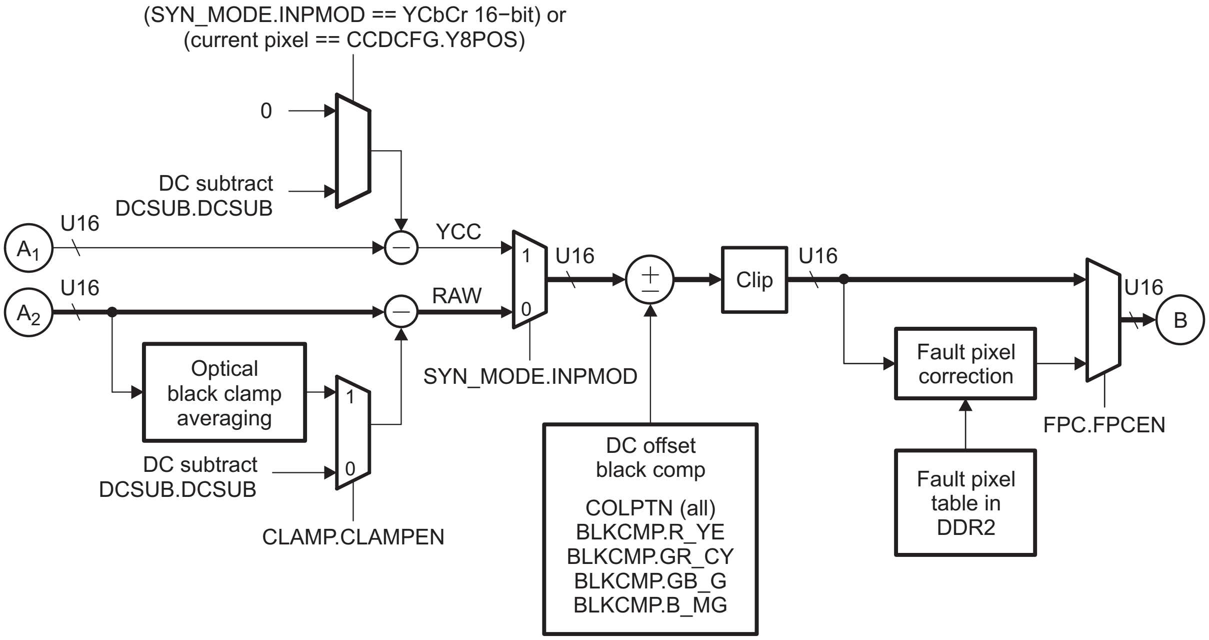


4.3.1.2 CCD Controller Initial Processing – Raw Data Mode

The initial CCD controller processing is shown in Figure 12 and includes the following functions:

- Optical Black Clamping
- Black Level Compensation
- Fault Pixel Correction

Figure 12. CCD Controller Initial Processing Block Diagram – Raw Data Mode



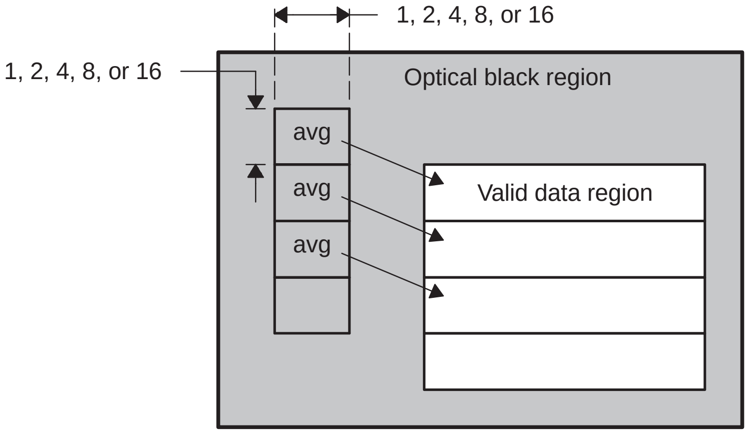
4.3.1.2.1 Optical Black Clamp

Sensor manufacturers typically provide some optically masked pixels at the beginning/end of each line to allow you to determine the noise floor on any given frame of data. The Optical Black Clamping function provides a means to average the optically black pixels and subtract that value from each input pixel as a first step in reducing the noise on the input pixels.

The averaging circuit takes an average of masked (black) pixel values from the image sensor, averaging pixels at the start (CLAMP.OBST) of each line (CLAMP.OBSLEN) and for the number of indicated lines (CLAMP.OBSLN) plus an optional gain adjustment (CLAMP.OBGAIN) and this value is subtracted from the image data at the succeeding line. You can control the position of the black pixels, the number of pixels (1, 2, 4, 8, or 16) in each line averaged, and the number of lines (1, 2, 4, 8, or 16) averaged.

Alternately, you can disable black clamp averaging (CLAMP.CLAMPEN) and select a constant black value for subtraction (DCSUB.DCSUB), instead of using the calculated average value.

Figure 13. CCD Controller Optical Black Averaging and Application



4.3.1.2.2 Black Level Compensation

After the Digital Clamp is applied to the data, black level compensation is applied. In this operation, a fixed value can be subtracted from the data depending on the color; that is, R, Gr, Gb, and B. The offset (BLKCMP register fields: R, GR, GB, and B) applied to each data sample is selected according to the pixel position and the color (0/1/2/3) specified for each pixel position (COLPTN).

4.3.1.2.3 Fault Pixel Correction

The CCDC module implements an optional (FPC.FPCEN) fault pixel correction operation using a lookup table. The lookup table is stored in external SDRAM/DDR memory and contains information about the horizontal and vertical position of the pixels to be corrected as well as the type of operation to be performed on the pixel. The FPC_ADDR register specifies the starting address for the Fault Pixel Correction Table in SDRAM/DDR memory.

Note: The DDR2 address should be 64-byte aligned (6 LSBs are ignored).

There are three selections available for performing the fault pixel correction:

- Replace current pixel x with the average of x - 1 and x + 1 of the same color
- Replace current pixel x with x - 1 of the same color
- Replace current pixel x with x + 1 of the same color

The programmer is responsible for specifying the number of fault pixels in the frame to be corrected (FPC.FPCNUM) along with the starting address of the fault pixel correction table. Each entry is 32-bits wide (Table 17) with 13 bits used for the vertical position, 14 bits being used for the horizontal position, and 5 bits used for the method to correct the fault pixel (Table 18). The horizontal distance between two pixels of the same color can be between 1 and 6.

Table 17. Fault Pixel Table Format

Bits 31:19	Bits 18:5	Bits 4:0
Vertical position of bad pixel	Horizontal Position of bad pixel	Correction Method

Table 18. Fault Pixel Correction Method

Bits 4:0 Value	Correction Method
0	Replace pixel at x with the average of x - 1 and x + 1
1h	Replace pixel at x with x - 1
2h	Replace pixel at x with x + 1
3h	Replace pixel at x with the average of x - 2 and x + 2
4h	Replace pixel at x with x - 2
5h	Replace pixel at x with x + 2
6h	Replace pixel at x with the average of x - 3 and x + 3
7h	Replace pixel at x with x - 3
8h	Replace pixel at x with x + 3
9h	Replace pixel at x with the average of x - 4 and x + 4
Ah	Replace pixel at x with x - 4
Bh	Replace pixel at x with x + 4
Ch	Replace pixel at x with the average of x - 5 and x + 5
Dh	Replace pixel at x with x - 5
Eh	Replace pixel at x with x + 5
Fh	Replace pixel at x with the average of x - 6 and x + 6
10h	Replace pixel at x with x - 6
11h	Replace pixel at x with x + 6
12h-1Fh	Reserved

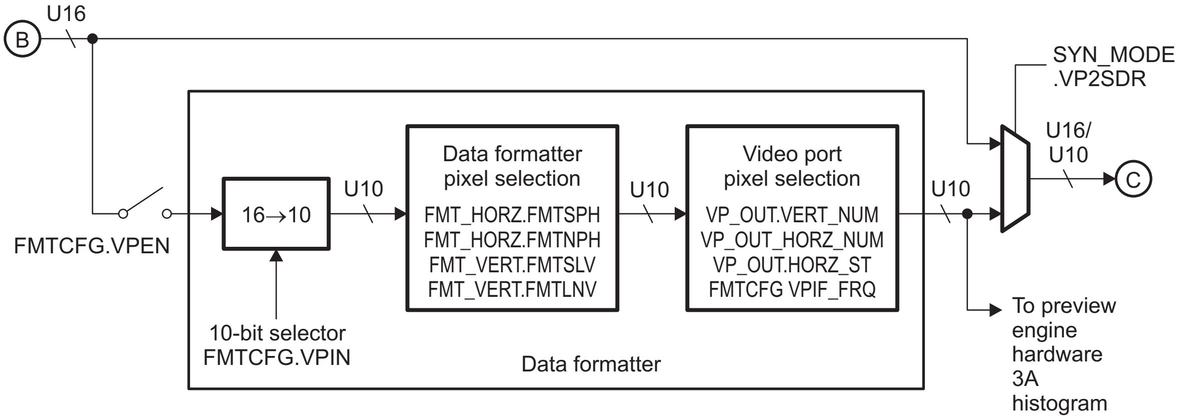
Note: The CCD/CMOS controller must request the fault pixel entries from external SDRAM/DDRAM. The CCDC is capable of buffering up to a total of 128 fault entries. The 128 entries are arranged as two 64 entry blocks in a ping-pong scheme. On every new frame, the CCDC issues a request to the system DMA to transfer 64 entries. In order to allow time to fetch the fault pixels from SDRAM/DDRAM, the number of fault pixels to be corrected in a certain time will be limited to the system DMA bandwidth and latency. At a minimum, the time to transfer 64 entries from the external memory should be less than the time to fault-pixel correct the 64 entries residing in the block. If this requirement is not met at any instant of time, then the FPC.FPERR bit will be set to 1 and halt processing for that frame. If there is an error, no more fault pixels will be corrected in that frame.

4.3.1.3 Video Port Interface and Data Formatter – Raw Data Mode

The next portion of the CCD/CMOS controller processing is shown in Figure 14. This processing includes the Data Formatter Processing and the Video Port Interface processing/formatting. The Video Port Interface is the internal VPFE interface to the following Modules:

- Preview Engine
- Hardware 3A (H3A)
- Histogram

Figure 14. CCD Controller Video Port Interface and Data Formatter Block Diagram – Raw Data Mode



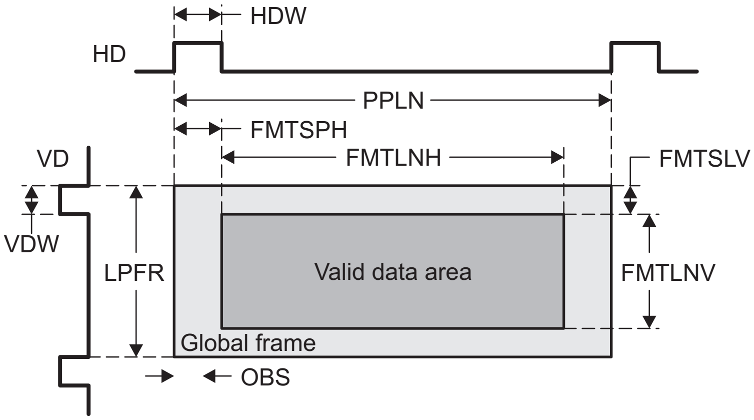
The Video Port/Data Formatter output can also be saved to DDR2 instead of the RAW CCD data. The setting SYN_MODE.VP2SDR sends the Video Port data to the CCDC's Output Formatter. In addition, the normal DDR2 write enable (SYN_MODE.WEN) also must be enabled to store the output to DDR2.

The Data Formatter and VPI are only 10-bits wide so the input data must be adjusted as it enters these modules. For flexibility, the bits to be retained can be selected by FMTCFG.VPIN.

4.3.1.3.1 Conversion Area Select Parameters

Output HD/VD signals that are controlled with the settings: PPLN, HDW, LPFR and VDW can still be generated when the Data Formatter is enabled. The settings must reflect those for the CCD sensor readout frame, not the resultant reformatted frame. The registers FMT_HORZ and FMT_VERT serve to control the interpretation of the input data frame, much as the registers HORZ_INFO, VERT_START, and VERT_LINES do in Normal mode, as seen in Figure 15.

Figure 15. CCD Controller Data Formatter Conversion Area Selection



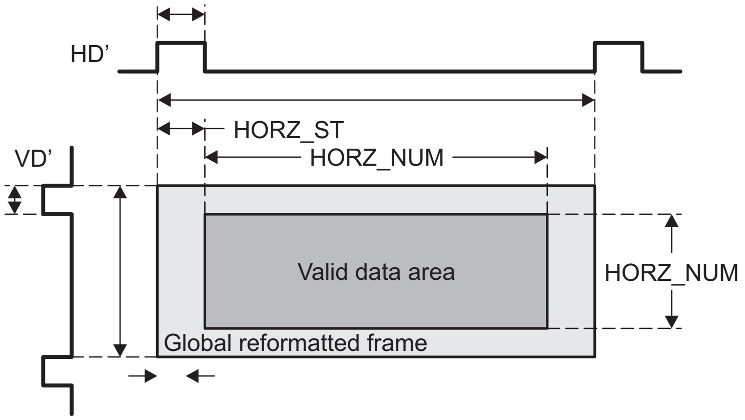
HDW - Horizontal sync width
HD - Horizontal sync
PPLN - Pixels per line
LPFR - Lines per frame
VDW - Vertical sync width
VD - Vertical sync

FMTSPH - Start pixel horizontal
FMTLNH - Size horizontal of valid area
FMTSLV - Start line vertical
FMTLNV - Size vertical of valid area

4.3.1.3.2 Video Port Framing

After the Data Formatter, the data is reframed by the setting in the VP_OUT register, where the data region to be sent on for further processing is defined. This is in addition to the framing applied at the beginning of the Data Formatter so take care to make these settings relative to that frame. See Figure 16.

Figure 16. CCD Controller Video Port Framing



HD - Horizontal sync
VD - Vertical sync

HORZ_ST - Start pixel horizontal
HORZ_NUM - Size horizontal of valid area
VERT_NUM - Size vertical of valid area

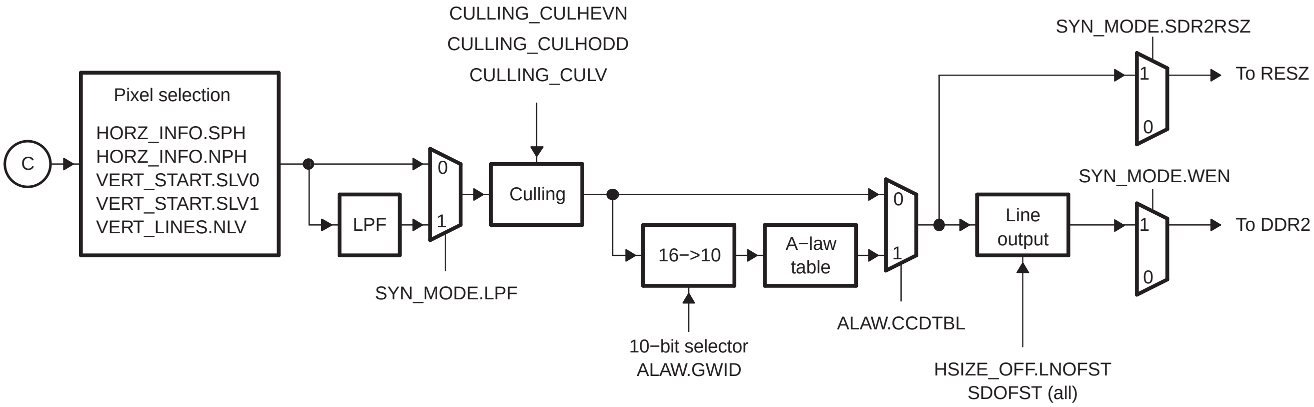
4.3.1.4 CCD Controller Output Formatter – Raw Data Mode

The final stage of the CCD controller processing is the output formatter, as shown in Figure 17. Again, a framing selection is applied to limit the processing area by the settings in the HORZ_INFO, VERT_START, and VERT_LINES registers.

Note that this is in addition to the framing applied at the beginning and at the end of the Data Formatter operation, if the video port path to the DDR2 is selected (SYN_MODE.VP2SDR). Be careful to make these settings relative to that frame.

Note also that the option to send the CCD controller output to the resizer module (SYN_MODE.SDR2RSZ) should not be used when in Raw Data mode because the resizer only operates on YUV422 format data. When resizing is desired in Raw Data mode, use the preview engine YUV422 output.

Figure 17. CCD Controller Output Formatter Block Diagram – Raw Data Mode



4.3.1.4.1 Low-Pass Filter

An optional low-pass filter can be applied (SYN_MODE.LPF) after the reframing. The low-pass filter consists of a simple 3-tap (1/4 , 1/2 , 1/4) filter. Two pixels on the left and two pixels on the right of each line are cropped, if the filter is enabled. Use of the low-pass filter is intended for bandwidth reduction, if culling is enabled.

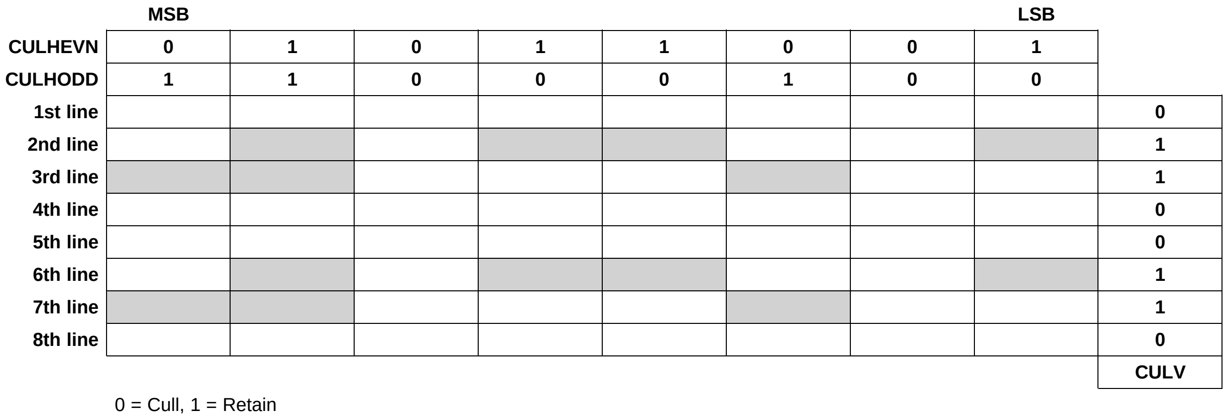
4.3.1.4.2 Culling

An optional culling operation can be enabled, which culls (deletes) selected pixel data from a line (CULLING.CULHEVN, CULLING.CULHODD, 8-bit repeating mask, one per field) and selected lines from a frame (CULLING.CULV).

Figure 18 is an example showing how the register values apply the decimation pattern to the data. The pixels in white are saved to DDR2 and the shaded pixels are discarded. In the case, CULLING = 59C4 0066h:

- CULHEVN = 59h
- CULHODD = C4h
- CULV = 66h

Figure 18. Example for Decimation Pattern



	MSB							LSB	
CULHEVN	0	1	0	1	1	0	0	1	
CULHODD	1	1	0	0	0	1	0	0	
1st line									0
2nd line		▨		▨	▨			▨	1
3rd line	▨	▨				▨			1
4th line									0
5th line									0
6th line		▨		▨	▨			▨	1
7th line	▨	▨				▨			1
8th line									0
									CULV

0 = Cull, 1 = Retain

4.3.1.4.3 A-Law Transformation

An optional 10-to-8-bit A-Law transformation using a fixed A-Law table can be applied (ALAW.CCDTBL) as the final processing stage. Using this causes the data width to be reduced to 8-bits and allows packing to 8-bits/pixel when saving to DDR2. Since the data resolution can be greater than 10-bits at this stage, the 10-bits for input to the A-Law operation must be selected (ALAW.GWID).

Note that other VPFE modules have an inverse A-Law table option so they can reverse this nonlinear operation if this saved data is to be read back in for further processing.

Figure 19. A-Law Table

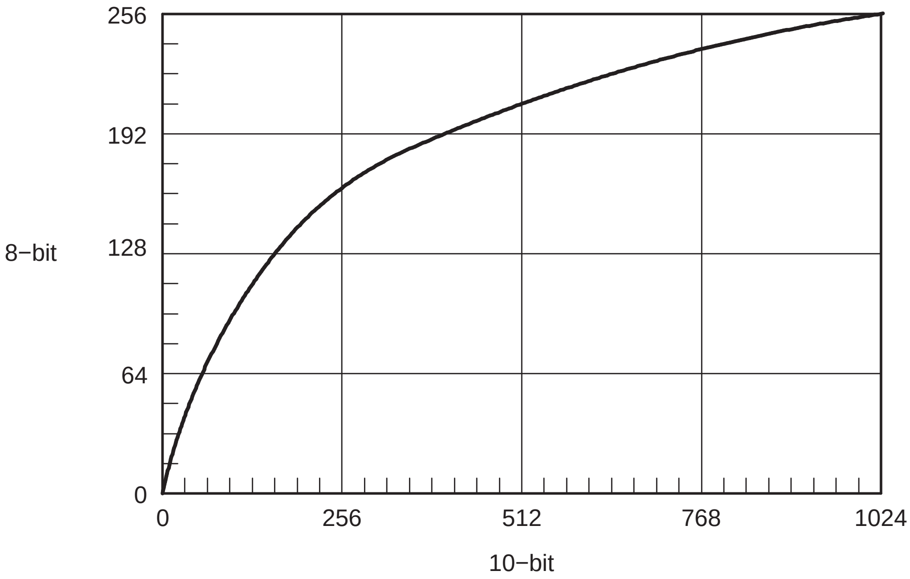


Table 19. A-Law Table – Part 1

Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
0	0	64	64	128	112	192	140	256	161	320	176	384	189	448	200
1	1	65	65	129	113	193	141	257	161	321	176	385	189	449	200
2	2	66	66	130	113	194	141	258	161	322	177	386	189	450	200
3	3	67	67	131	114	195	142	259	161	323	177	387	189	451	200
4	4	68	68	132	114	196	142	260	162	324	177	388	190	452	200
5	5	69	69	133	115	197	142	261	162	325	177	389	190	453	200
6	6	70	70	134	115	198	143	262	162	326	177	390	190	454	201
7	7	71	71	135	116	199	143	263	162	327	178	391	190	455	201
8	8	72	72	136	116	200	143	264	163	328	178	392	190	456	201
9	9	73	73	137	117	201	144	265	163	329	178	393	190	457	201
10	10	74	74	138	117	202	144	266	163	330	178	394	191	458	201
11	11	75	75	139	118	203	144	267	163	331	178	395	191	459	201
12	12	76	76	140	118	204	145	268	164	332	179	396	191	460	201
13	13	77	77	141	119	205	145	269	164	333	179	397	191	461	202
14	14	78	78	142	119	206	145	270	164	334	179	398	191	462	202
15	15	79	78	143	120	207	146	271	164	335	179	399	191	463	202
16	16	80	79	144	120	208	146	272	165	336	179	400	192	464	202
17	17	81	80	145	121	209	146	273	165	337	180	401	192	465	202
18	18	82	81	146	121	210	147	274	165	338	180	402	192	466	202
19	19	83	82	147	122	211	147	275	166	339	180	403	192	467	202
20	20	84	83	148	122	212	147	276	166	340	180	404	192	468	203
21	21	85	84	149	123	213	148	277	166	341	181	405	193	469	203
22	22	86	84	150	123	214	148	278	166	342	181	406	193	470	203
23	23	87	85	151	124	215	148	279	167	343	181	407	193	471	203
24	24	88	86	152	124	216	149	280	167	344	181	408	193	472	203
25	25	89	87	153	125	217	149	281	167	345	181	409	193	473	203
26	26	90	88	154	125	218	149	282	167	346	182	410	193	474	204
27	27	91	88	155	125	219	150	283	168	347	182	411	194	475	204
28	28	92	89	156	126	220	150	284	168	348	182	412	194	476	204
29	29	93	90	157	126	221	150	285	168	349	182	413	194	477	204
30	30	94	91	158	127	222	151	286	168	350	182	414	194	478	204
31	31	95	91	159	127	223	151	287	168	351	183	415	194	479	204
32	32	96	92	160	128	224	151	288	169	352	183	416	194	480	204
33	33	97	93	161	128	225	152	289	169	353	183	417	195	481	205
34	34	98	93	162	129	226	152	290	169	354	183	418	195	482	205
35	35	99	94	163	129	227	152	291	169	355	183	419	195	483	205
36	36	100	95	164	129	228	152	292	170	356	184	420	195	484	205
37	37	101	96	165	130	229	153	293	170	357	184	421	195	485	205
38	38	102	96	166	130	230	153	294	170	358	184	422	195	486	205
39	39	103	97	167	131	231	153	295	170	359	184	423	196	487	205
40	40	104	98	168	131	232	154	296	171	360	184	424	196	488	206
41	41	105	98	169	132	233	154	297	171	361	185	425	196	489	206
42	42	106	99	170	132	234	154	298	171	362	185	426	196	490	206
43	43	107	100	171	132	235	155	299	171	363	185	427	196	491	206
44	44	108	100	172	133	236	155	300	172	364	185	428	196	492	206
45	45	109	101	173	133	237	155	301	172	365	185	429	197	493	206
46	46	110	102	174	134	238	155	302	172	366	185	430	197	494	206
47	47	111	102	175	134	239	156	303	172	367	186	431	197	495	207
48	48	112	103	176	134	240	156	304	173	368	186	432	197	496	207
49	49	113	103	177	135	241	156	305	173	369	186	433	197	497	207
50	50	114	104	178	135	242	157	306	173	370	186	434	197	498	207
51	51	115	105	179	136	243	157	307	173	371	186	435	198	499	207
52	52	116	105	180	136	244	157	308	173	372	187	436	198	500	207
53	53	117	106	181	136	245	157	309	174	373	187	437	198	501	207
54	54	118	106	182	137	246	158	310	174	374	187	438	198	502	208
55	55	119	107	183	137	247	158	311	174	375	187	439	198	503	208
56	56	120	108	184	137	248	158	312	174	376	187	440	198	504	208
57	57	121	108	185	138	249	159	313	175	377	188	441	198	505	208
58	58	122	109	186	138	250	159	314	175	378	188	442	199	506	208
59	59	123	109	187	139	251	159	315	175	379	188	443	199	507	208
60	60	124	110	188	139	252	159	316	175	380	188	444	199	508	208
61	61	125	110	189	139	253	160	317	175	381	188	445	199	509	208
62	62	126	111	190	140	254	160	318	176	382	188	446	199	510	209
63	63	127	112	191	140	255	160	319	176	383	189	447	199	511	209

Table 20. A-Law Table – Part 2

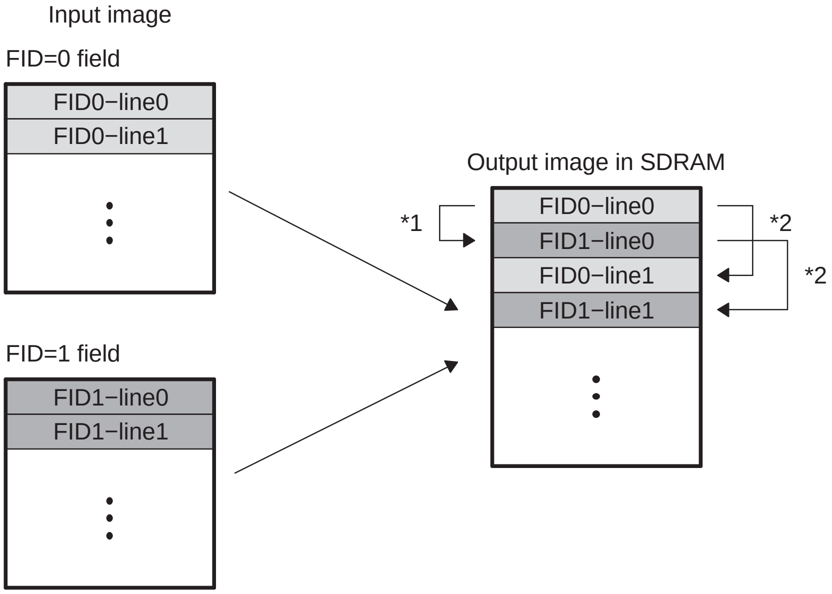
Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
512	209	576	217	640	224	704	231	768	237	832	243	896	248	960	253
513	209	577	217	641	225	705	231	769	237	833	243	897	248	961	253
514	209	578	217	642	225	706	231	770	237	834	243	898	248	962	253
515	209	579	217	643	225	707	231	771	237	835	243	899	248	963	253
516	209	580	218	644	225	708	232	772	238	836	243	900	248	964	253
517	210	581	218	645	225	709	232	773	238	837	243	901	248	965	253
518	210	582	218	646	225	710	232	774	238	838	243	902	248	966	253
519	210	583	218	647	225	711	232	775	238	839	243	903	249	967	253
520	210	584	218	648	225	712	232	776	238	840	243	904	249	968	253
521	210	585	218	649	225	713	232	777	238	841	244	905	249	969	253
522	210	586	218	650	226	714	232	778	238	842	244	906	249	970	254
523	210	587	218	651	226	715	232	779	238	843	244	907	249	971	254
524	211	588	219	652	226	716	232	780	238	844	244	908	249	972	254
525	211	589	219	653	226	717	232	781	238	845	244	909	249	973	254
526	211	590	219	654	226	718	233	782	238	846	244	910	249	974	254
527	211	591	219	655	226	719	233	783	239	847	244	911	249	975	254
528	211	592	219	656	226	720	233	784	239	848	244	912	249	976	254
529	211	593	219	657	226	721	233	785	239	849	244	913	249	977	254
530	211	594	219	658	226	722	233	786	239	850	244	914	249	978	254
531	211	595	219	659	227	723	233	787	239	851	244	915	249	979	254
532	212	596	220	660	227	724	233	788	239	852	244	916	250	980	254
533	212	597	220	661	227	725	233	789	239	853	245	917	250	981	254
534	212	598	220	662	227	726	233	790	239	854	245	918	250	982	254
535	212	599	220	663	227	727	233	791	239	855	245	919	250	983	254
536	212	600	220	664	227	728	233	792	239	856	245	920	250	984	255
537	212	601	220	665	227	729	234	793	239	857	245	921	250	985	255
538	212	602	220	666	227	730	234	794	240	858	245	922	250	986	255
539	212	603	220	667	227	731	234	795	240	859	245	923	250	987	255
540	213	604	220	668	227	732	234	796	240	860	245	924	250	988	255
541	213	605	221	669	228	733	234	797	240	861	245	925	250	989	255
542	213	606	221	670	228	734	234	798	240	862	245	926	250	990	255
543	213	607	221	671	228	735	234	799	240	863	245	927	250	991	255
544	213	608	221	672	228	736	234	800	240	864	245	928	250	992	255
545	213	609	221	673	228	737	234	801	240	865	246	929	250	993	255
546	213	610	221	674	228	738	234	802	240	866	246	930	251	994	255
547	214	611	221	675	228	739	235	803	240	867	246	931	251	995	255
548	214	612	221	676	228	740	235	804	240	868	246	932	251	996	255
549	214	613	221	677	228	741	235	805	240	869	246	933	251	997	255
550	214	614	222	678	229	742	235	806	241	870	246	934	251	998	255
551	214	615	222	679	229	743	235	807	241	871	246	935	251	999	255
552	214	616	222	680	229	744	235	808	241	872	246	936	251	1000	255
553	214	617	222	681	229	745	235	809	241	873	246	937	251	1001	255
554	214	618	222	682	229	746	235	810	241	874	246	938	251	1002	255
555	215	619	222	683	229	747	235	811	241	875	246	939	251	1003	255
556	215	620	222	684	229	748	235	812	241	876	246	940	251	1004	255
557	215	621	222	685	229	749	235	813	241	877	246	941	251	1005	255
558	215	622	222	686	229	750	236	814	241	878	247	942	251	1006	255
559	215	623	223	687	229	751	236	815	241	879	247	943	252	1007	255
560	215	624	223	688	230	752	236	816	241	880	247	944	252	1008	255
561	215	625	223	689	230	753	236	817	242	881	247	945	252	1009	255
562	215	626	223	690	230	754	236	818	242	882	247	946	252	1010	255
563	216	627	223	691	230	755	236	819	242	883	247	947	252	1011	255
564	216	628	223	692	230	756	236	820	242	884	247	948	252	1012	255
565	216	629	223	693	230	757	236	821	242	885	247	949	252	1013	255
566	216	630	223	694	230	758	236	822	242	886	247	950	252	1014	255
567	216	631	223	695	230	759	236	823	242	887	247	951	252	1015	255
568	216	632	224	696	230	760	236	824	242	888	247	952	252	1016	255
569	216	633	224	697	230	761	237	825	242	889	247	953	252	1017	255
570	216	634	224	698	231	762	237	826	242	890	247	954	252	1018	255
571	217	635	224	699	231	763	237	827	242	891	248	955	252	1019	255
572	217	636	224	700	231	764	237	828	242	892	248	956	252	1020	255
573	217	637	224	701	231	765	237	829	243	893	248	957	253	1021	255
574	217	638	224	702	231	766	237	830	243	894	248	958	253	1022	255
575	217	639	224	703	231	767	237	831	243	895	248	959	253	1023	255

4.3.1.4.4 Line Output Control

The CCD controller's final stage is the Line Output Control, which controls how the input sensor lines are written to DDR2. The value SDR_ADDR.ADR defines the starting address where the frame should be written in DDR2. The value HSIZE_OFF.LNOFST defines the distance between the beginning of output lines, in bytes. Both the starting address and line offset values must be aligned to a 32-byte boundaries; that is, either 16 or 32 pixels, depending on the SYN_MODE.PACK8 setting. The SDOFST register can be used to define additional offsets depending on the Field ID and even/odd line numbers. This provides a means to de-interlace an interlaced, 2-field input and also to invert an input image vertically. See Figure 20 and Figure 21 for examples.

- SDOFST.FIINV – invert interpretation of the Field ID signal
- SDOFST.LOFTS0 – offset, in lines, between even lines on even fields (field 0)
- SDOFST.LOFTS1 – offset, in lines, between odd lines on even fields (field 0)
- SDOFST.LOFTS2 – offset, in lines, between even lines on odd fields (field 1)
- SDOFST.LOFTS3 – offset, in lines, between odd lines on odd fields (field 1)

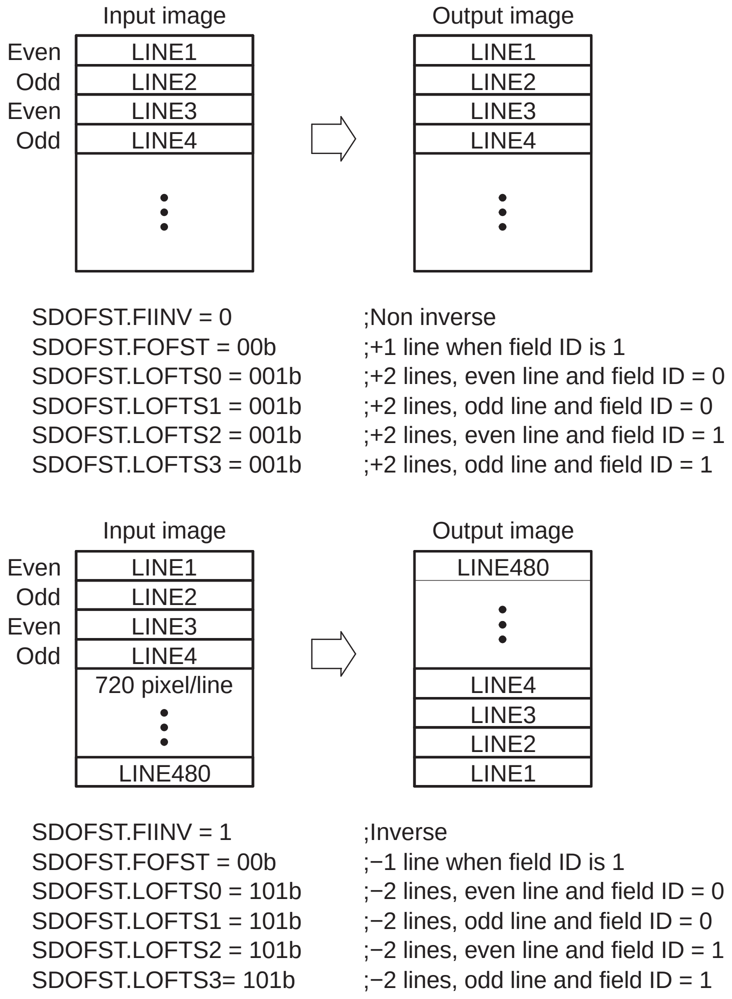
Figure 20. Frame Image Format Conversion (de-interlaced, 2-field input)



*1 − SDOFST.FOFST = 00b; +1 line for FID=1 line

*2 − SDOFST.LOFTS0=001b; +2 lines
SDOFST.LOFTS1=001b; +2 lines
SDOFST.LOFTS2=001b; +2 lines
SDOFST.LOFTS3=001b; +2 lines

Figure 21. Example Formats of Input and Output Image



```
SDOFST.FIINV = 0            ;Non inverse
SDOFST.FOFST = 00b          ;+1 line when field ID is 1
SDOFST.LOFTS0 = 001b        ;+2 lines, even line and field ID = 0
SDOFST.LOFTS1 = 001b        ;+2 lines, odd line and field ID = 0
SDOFST.LOFTS2 = 001b        ;+2 lines, even line and field ID = 1
SDOFST.LOFTS3 = 001b        ;+2 lines, odd line and field ID = 1
```

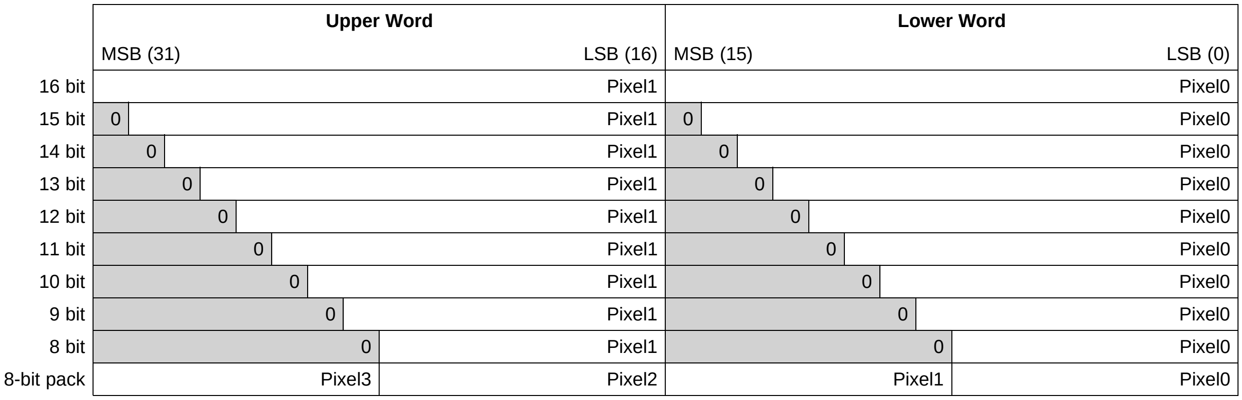
```
SDOFST.FIINV = 1            ;Inverse
SDOFST.FOFST = 00b          ;−1 line when field ID is 1
SDOFST.LOFTS0 = 101b        ;−2 lines, even line and field ID = 0
SDOFST.LOFTS1 = 101b        ;−2 lines, odd line and field ID = 0
SDOFST.LOFTS2 = 101b        ;−2 lines, even line and field ID = 1
SDOFST.LOFTS3= 101b         ;−2 lines, odd line and field ID = 1
```

4.3.1.4.5 Output Format

The data bits comprising each pixel are stored in the lower bits of a 16-bit DDR2 word and the unused bits are zero-filled. The DDR2 data format is shown in Figure 22, with the format used determined by the SYN_MODE.DATSIZ setting for all but the packed format. If 8-bit data is input, or if the A-Law compression is applied, the data can be packed via the SYN_MODE.PACK8 setting so that a pixel will only occupy 8-bits.

Note that data is only output to DDR2 when enabled via the SYN_MODE.WEN setting.

Figure 22. DDR2 Output Format

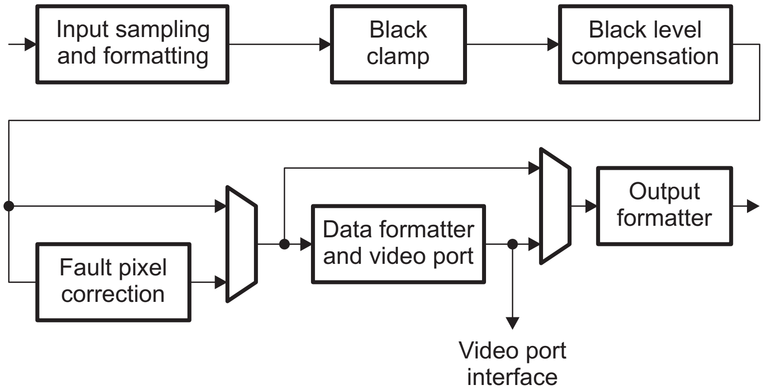


	Upper Word		Lower Word	
	MSB (31)	LSB (16)	MSB (15)	LSB (0)
16 bit		Pixel1		Pixel0
15 bit	0	Pixel1	0	Pixel0
14 bit	0	Pixel1	0	Pixel0
13 bit	0	Pixel1	0	Pixel0
12 bit	0	Pixel1	0	Pixel0
11 bit	0	Pixel1	0	Pixel0
10 bit	0	Pixel1	0	Pixel0
9 bit	0	Pixel1	0	Pixel0
8 bit	0	Pixel1	0	Pixel0
8-bit pack	Pixel3	Pixel2	Pixel1	Pixel0

4.3.2 CCD Controller Processing – YUV Modes

The previous section described the data processing for raw data modes from CCD/CMOS sensors. When interfacing to YUV data sources, most of the CCD processing stages should be explicitly bypassed. The high-level CCD controller block diagram is shown in Figure 23, with the expected YUV data flow.

Figure 23. CCD Controller Processing Block Diagram – YUV Modes



The following sections describes the CCD controller processing for the YUV input modes (SYN_MODE.INPMODE = 1 or 2 || REC656IF.REC656ON = 1) in more detail. In this mode, YUV422 input data typically at 8-bits per luma/chroma sample is input.

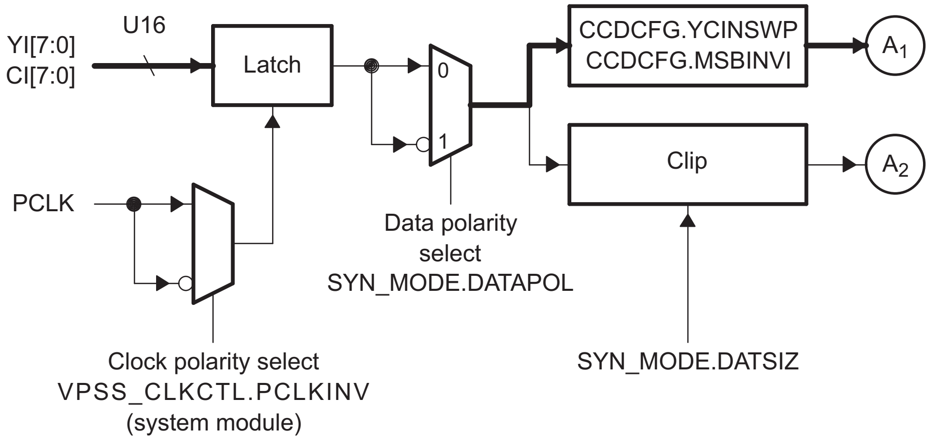
Processing in REC656 mode is identical to that of the other YUV modes. There is an additional processing block, not shown, that extracts the sync information from the data signal and generates the HD/VD/Field signals for downstream processing.

4.3.2.1 CCD Controller Input Sampling and Formatting – YUV Modes

The CCD controller input sampling is shown in Figure 24. The bold data path (A_1 output) is the YUV data path through the CCD controller; the lower data path is only applicable to the Raw Data input mode.

- Data is latched by the pixel clock
- Pixel clock polarity can be either rising or falling edge. This is set in the System module via the register field: VPSS_CLKCTL.PCLKINV.
- Data can be interpreted as either normal or inverted (SYN_MODE.DATAPOL).
- There is an option to swap the upper and lower portion of the 16-bit YUV data bus (CCDCFG.YCINSWP). This will swap the luma and chroma samples in 16-bit YUV mode. This will determine which half of the bus is used as the input source in 8-bit mode and can be used in 8-bit YUV mode to support two separate YUV input ports.
- The MSB of the chroma signal can also be inverted (CCDCFG.MSBINVI).

Figure 24. CCD Controller Input Sampling Block Diagram – YUV Modes

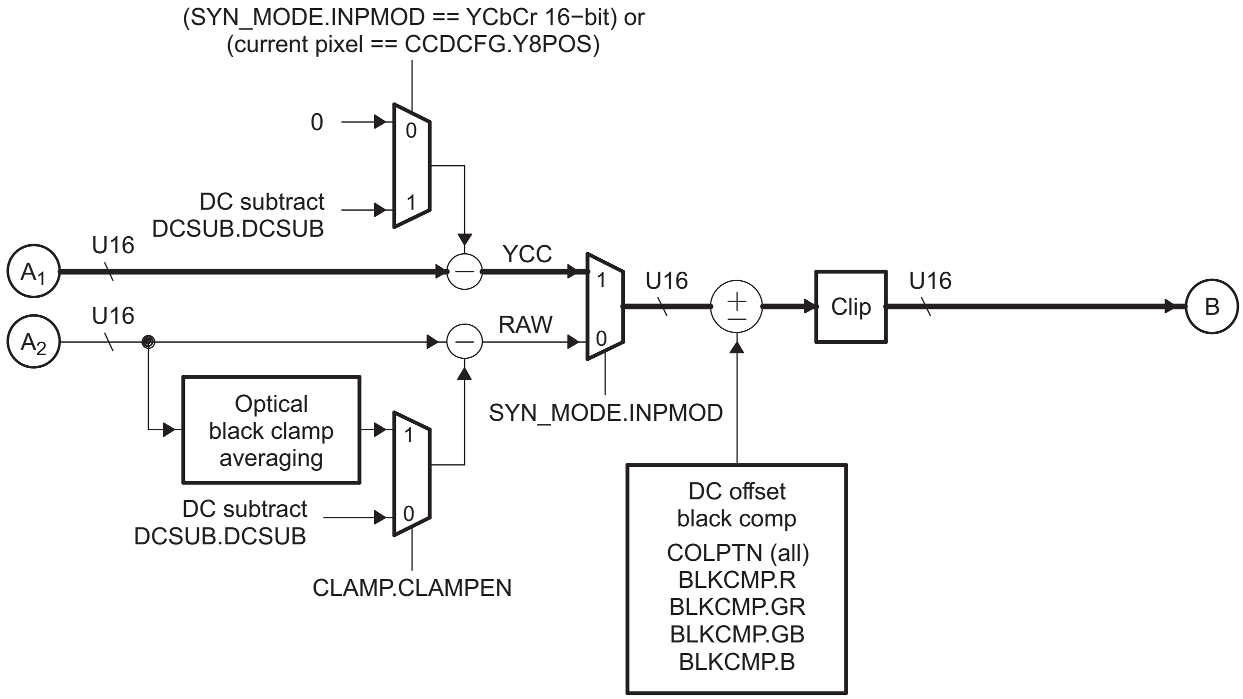

U16
YI[7:0]
CI[7:0]
Latch
0
1
CCDCFG.YCINSWP
CCDCFG.MSBINVI
A1
Clip
A2
PCLK
Data polarity select SYN_MODE.DATAPOL
Clock polarity select VPSS_CLKCTL.PCLKINV (system module)
SYN_MODE.DATSIZ

4.3.2.2 CCD Controller Initial Processing – YUV Modes

The initial CCD controller processing for the YUV data path is shown in Figure 25 and includes the following functions:

- Optical Black Clamping
- Black Level Compensation

Figure 25. CCD Controller Initial Processing Block Diagram – YUV Modes



4.3.2.2.1 Optical Black Clamp

Note: This function does not clip negative results to 0 for YUV 8-bit input mode (SYN_MODE.INPMOD = 2h) or REC656 input mode (REC656IF.REC656ON = 1).

For YUV data, this operation subtracts a fixed value (DCSUB.DCSUB) from the luma sample. To disable this operation, clear the subtraction value to 0.

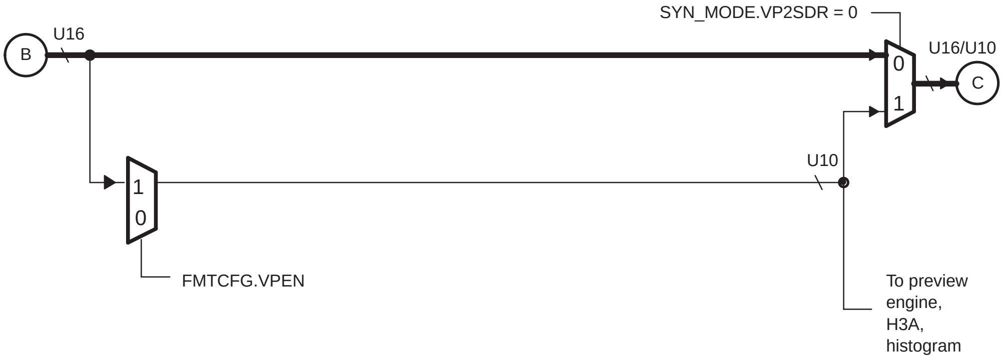
4.3.2.2.2 Black Level Compensation

Black level compensation is applicable only to raw data mode and should be disabled in YUV modes by clearing the black compensation register values (BLKCMP register fields: R, GR, GB, and B) to 0.

4.3.2.3 Video Port Interface and Data Formatter – YUV Modes

This portion of the CCD controller processing (Figure 26) should be bypassed in YUV modes (FMTCFG.VPEN = 0 && SYN_MODE.VP2SDR = 0).

Figure 26. CCD Controller Video Port Interface and Data Formatter Block Diagram – YUV Modes

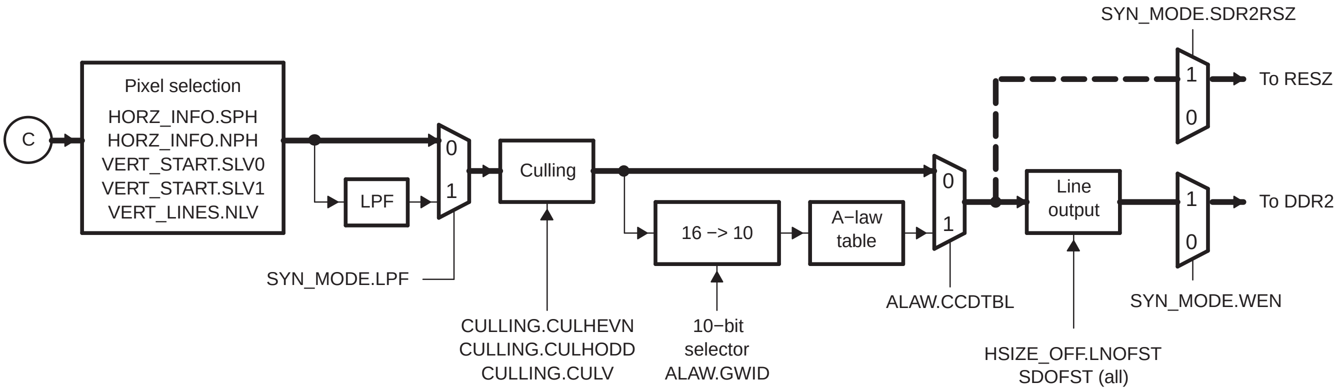


4.3.2.4 CCD Controller Output Formatter – YUV Modes

The final stage of the CCD controller processing is the output formatter, as shown in Figure 27. A framing selection is applied to limit the processing area by the settings in the HORZ_INFO, VERT_START, and VERT_LINES registers.

There is also an option to send the CCD controller YUV output to the resizer module (SYN_MODE.SDR2RSZ).

Figure 27. CCD Controller Output Formatter Block Diagram – YUV Modes



4.3.2.4.1 Low-Pass Filter

The low-pass filter should be disabled (SYN_MODE.LPF = 0).

4.3.2.4.2 Culling

It is theoretically possible to use culling in YUV mode, but care must be taken to preserve the 422 output format.

4.3.2.4.3 A-Law Transformation

The A-Law transformation should not be used in YUV modes (ALAW.CCDTBL = 0).

4.3.2.4.4 Line Output Control

The CCD controller’s final stage of Line Output Control can be used for YUV modes. See the description in Section 4.3.1.4.4 covering raw data processing.

4.3.2.4.5 Output Format

In 16-bit input mode, data is stored in SDRAM in packed YUV422 mode, with two pixels per 32-bits, as shown in Table 21. If 8-bit data is input, the data can be packed via the SYN_MODE.PACK8 setting so that each Y or C input will only occupy 8 bits. If SYN_MODE.PACK8 is not enabled in 8-bit or 10-bit input, then each Y or C will occupy the LSBs of each 16-bit word in SDRAM.

Table 21. DDR Output Format for YUV422 Mode

	SDRAM Data Format			
	Upper word		Lower word	
SDRAM Address	MSB (31)	LSB (16)	MSB (15)	LSB (0)
N	Y1	Cr0	Y0	Cb0
N + 1	Y3	Cr2	Y2	Cb2
N + 2	Y5	Cr4	Y4	Cb4

4.3.3 Preview Engine/Image Signal Processor

The preview engine accepts raw image/video data in a Bayer pattern only pattern and transforms it into YUV data in the 422 format. The preview engine is not used for YUV modes.

A number of image processing steps are necessary to achieve raw color to YUV422 color space conversion. The output of the preview engine is typically used for both video compression and display (via analog or digital interface). The preview engine is capable of processing up to 75 Mpixels/second. The processing flow of the preview engine is shown in Figure 28. In Figure 28, optional blocks (processing that can be enabled/disabled) are distinguished by dashed, clear boxes; mandatory blocks (always enabled and must be programmed) are shown as solid, shaded boxes.

4.3.3.1 Input Interface

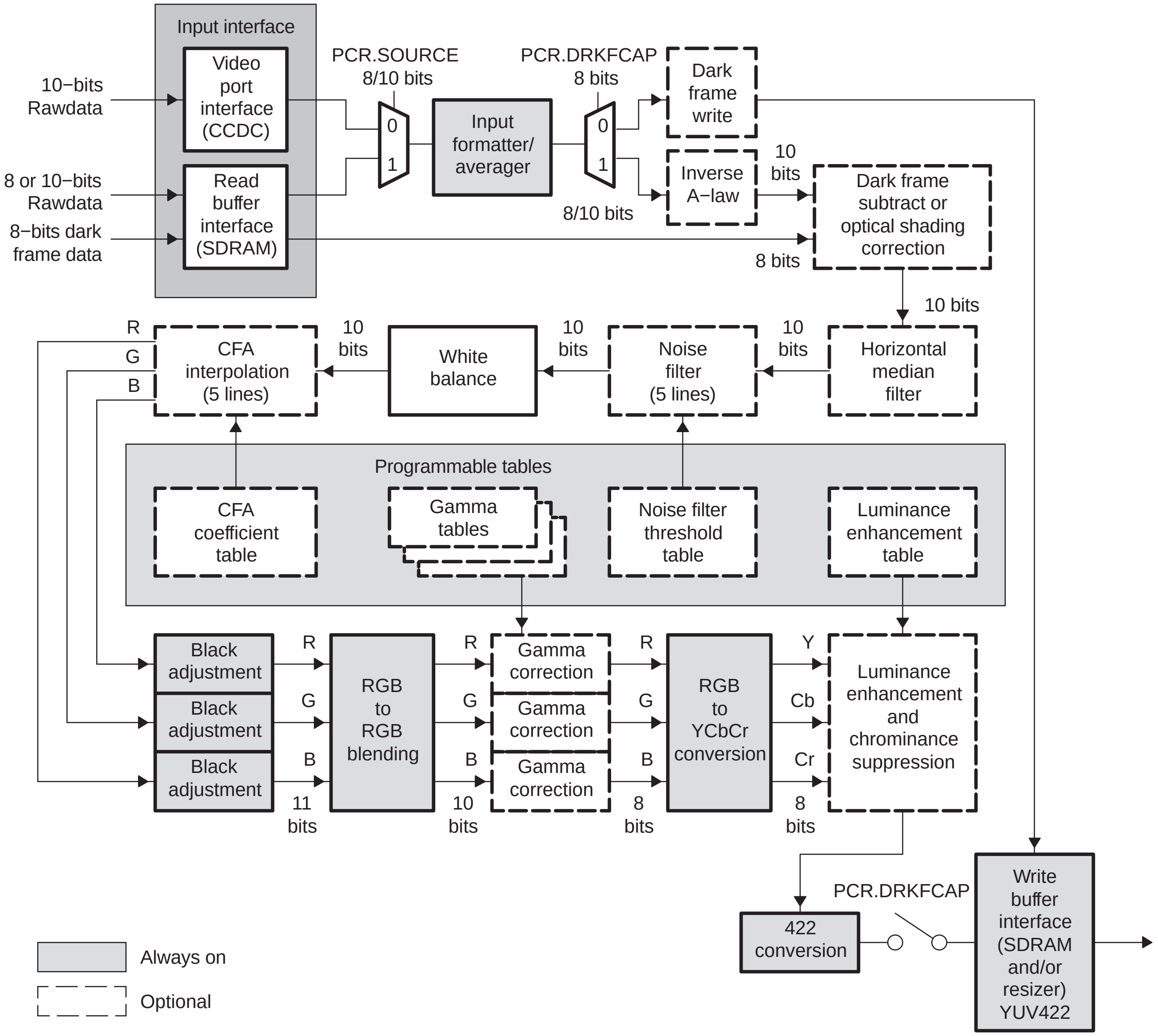
The preview engine receives raw image/video data from either the video port interface via the CCD/CMOS controller module (which is interfaced to an external CCD/CMOS sensor) or from the read buffer interface via the SDRAM/DDRAM (PCR.SOURCE). The input data is 10-bits wide if the source is the video port interface. When the input source is the read buffer interface, the data can either be 8-bit or 10-bits (PCR.WIDTH). The 8-bit data can either be linear or non-linear, as explained in Section 4.3.3.4. In addition, the preview engine can optionally fetch a dark frame from the SDRAM/DDRAM with each pixel being 8-bits wide (see Section 4.3.3.3 and Section 4.3.3.5 for more details).

The frame input size is configured using the HORZ_INFO and VERT_INFO registers. If the input source is the CCD/CMOS controller, then care must be taken to ensure that the input height set in the preview engine is less than or equal to the output height of the video port output of the CCD/CMOS controller. Additionally the input width must be at least 4 pixels smaller than the CCDC output width (SPH ≥ 2; EPH ≥ 2 pixels before the last pixel sent from the CCDC).

The input SDRAM/DDRAM address (RSDR_ADDR) and line offset (RADR_OFFSET) registers must be aligned on 32-byte boundaries when the input source is set to SDRAM/DDRAM. Furthermore, the dark frame input address (DSDR_ADDR) and line offset (DRKF_OFFSET) must also be aligned on 32-byte boundaries when the Dark Frame Subtract function is enabled.

When the input source is the SDRAM/DDRAM, the preview engine always operates in the one-shot mode. After enabling the preview engine and processing a frame, the enable bit is turned off and it is up to firmware to re-enable it to process the next frame from SDRAM/DDRAM. Optionally, when the input source is the CCD controller, the preview engine can be configured to operate in either one-shot mode or continuous mode (PCR.ONESHOT).

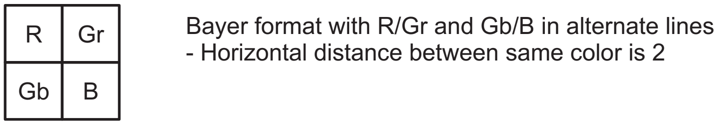
Figure 28. Preview Engine Processing Flow Block Diagram



4.3.3.2 Input Formatter/Averager

The preview engine output is limited to 1280 pixels per horizontal line due to line memory width restrictions in the noise filter and CFA interpolation blocks. In order to support sensors that output greater than 1280 pixels per line, an averager is incorporated to downsample by factors of 1 (no averaging), 2, 4, or 8 in the horizontal direction (AVE.COUNT). The horizontal distance between two consecutive pixels of the same color to be averaged is selectable between 1, 2, 3, or 4 for both even (AVE.EVENDIST) and odd (AVE.ODDDIST) lines. This must be configured to 2 to match the input Bayer pattern. The valid output of the input formatter/averager is either 8-bits or 10-bits wide.

Figure 29. Horizontal Distance for Bayer Pattern input



R	Gr
Gb	B

Bayer format with R/Gr and Gb/B in alternate lines
- Horizontal distance between same color is 2

4.3.3.3 Dark Frame Write

The preview engine is capable of capturing and saving a dark frame to the SDRAM/DDRAM instead of performing the conventional processing steps (PCR.DRKFCAP). This dark frame can later be subtracted from the raw image data, as described in Section 4.3.3.5, to eliminate the repeatable baseline noise level in the frame. Each input pixel is written out as an 8-bit value; if the input pixel value is greater than 255, it is saturated to 255. If a dark pixel is greater than 255, it is more likely to be a fault pixel and can be corrected by the Fault Pixel Correction module in the CCDC. If properly corrected, the value should be less than 255 when it reaches the preview engine. The WSDR_ADDR and WADD_OFFSET registers should be used to indicate the output address and line offset respectively of the dark frame output in memory as discussed in Section 4.3.3.13.

4.3.3.4 Inverse A-law

In order to save SDR/DDR capacity and bandwidth, the CCD/CMOS controller includes an option to apply 10-bit to 8-bit A-Law compression and to pack the sensor data to 1-byte per pixel. In order to process this data properly, the inverse A-law block is provided to decompress the 8-bit non-linear data back to 10-bit linear data if enabled (PCR.INVALAW). Even if the inverse A-law block is not enabled, but the input is still 8-bits (PCR.WIDTH), the data is shifted left by 2 to make it 10-bit data. If the input is 10-bits wide, no operation is performed on the data.

4.3.3.5 Darkframe Subtract or Shading Compensation

The preview engine is capable of optionally fetching a dark frame containing 8-bit values from SDRAM/DDRAM and subtracting it, pixel-by-pixel, from the incoming input frame (PCR.DRKFEN). This function is used to remove pattern noise in the sensor. The output of the dark frame subtract operation is 10-bits wide. There must be adequate SDRAM/DDRAM bandwidth if this feature is enabled. At its peak (operating at 75 MP/s), the dark frame subtract read bandwidth is 75 MB/s. If the data fetched from memory arrives late, then the PCR.DRK_FAIL status bit is set to indicate a fail.

Instead of performing the dark frame subtract, the preview engine can perform lens shading compensation (if PCR.SHADE_COMP is set along with PCR.DRKFEN). In this case, the 8-bit value that is fetched from SDRAM/DDRAM is multiplied with the incoming pixel and the result is right shifted by the number of bits specified by the PCR.SHADE_SFT parameter (0-7 bits).

4.3.3.6 Noise Filter

Following the darkframe subtract, a programmable noise filter that operates on a 3 × 3 grid of same color pixels can be used to reduce the noise in the image/video data. This filter always operates (identifies neighborhood same-color pixels that are close in value) on nine pixels of the same color. The noise filter can optionally be enabled or disabled (PCR.NFEN).

Image Size Reduction: If the noise filter is enabled, the preview engine will reduce the length of the output line of this stage by 4 pixels (2 starting pixels - left edge and 2 ending pixels - right edge) and 4 lines in each frame (2 starting lines - top edge and 2 ending lines - bottom edge). For example, if the input size is 656 × 490 pixels, the output will be 652 × 486 pixels. There will be no truncation of the input data if this block is disabled.

4.3.3.7 White Balance

The white balance module has two gain adjusters, a digital gain adjuster and a white balance adjuster. In the digital gain adjuster (WB_DGAIN), the raw data is multiplied by a fixed-value gain regardless of the color pixel to be processed. In the white balance gain adjuster (WBGAIN), the raw data is multiplied by a selected gain corresponding to the color of the processed pixel. The white balance gain can be selected from four possible 8-bit values depending on the position of the current pixel. Firmware can assign any combination of 4 pixels in the horizontal and vertical direction, 16 total locations, using WBSEL. For example, the white balance gain selected for pixel #0 and line #0 can be different than pixel #2 and line #0.

4.3.3.8 CFA Interpolation

The CFA interpolation block is responsible for populating the 2 missing color pixels at a given location resulting in a 3-color RGB pixel. It does this by interpolating data from neighboring pixels of the same color. The CFA function can optionally be enabled (PCR.CFAEN) and configured to different interpolation modes (PCR.CFAFMT).

Optional CFA Interpolation processing capabilities include the following:

- CFA interpolation can be configured to produce only 1 missing color (instead of the conventional 2 missing colors) if the sensor produces 2 colors per location. Such sensors are supposed to produce an image that is down-sampled by 2× in the horizontal direction. By programming the appropriate CFA coefficients, it is possible to either phase-correct the two input colors per location or not correct the phase and treat the two input colors as if they belong to the same physical location in the sensor.
- CFA interpolation can be configured to operate on sensors that are supposed to produce an image that is down-sampled by 2× in both the horizontal and vertical directions. In this case, there is no missing color and the physical pixels are supposed to be fully populated. The two diagonal green pixels in the Bayer pattern (assumed to be the forward slash always; firmware should take care of setting the first pixel to be red to allow for this) are averaged to obtain the green pixel.
- If CFA interpolation is disabled, then the incoming pixel is copied out on to the three output colors as is.

Image Size Reduction: If the CFA interpolation is enabled, the preview engine will reduce the output of this stage. Two pixels/lines in the left, right, top, and bottom edges are truncated in the Bayer/conventional CFA modes.

If the CFA format is 2× down-sampling in both horizontal and vertical directions, only 2 lines at the top and bottom of the image are truncated. The two left and right most pixels are processed.

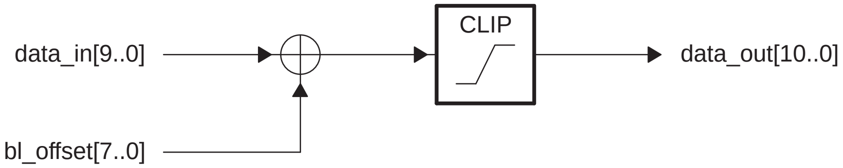
4.3.3.9 Black Adjustment

The output of the CFA interpolation is three pixels (red, blue, and green values) and this is fed as input to the black adjustment module. The black adjustment module performs the following calculation for an adjustment of each color level.

data_out = data_in + bl_offset

The *bl_offset* values for each color are programmable in the BLKADJOFF register. Figure 30 shows the block diagram of this black adjuster module.

Figure 30. Black Adjustment Functional Model



4.3.3.10 RGB Blending

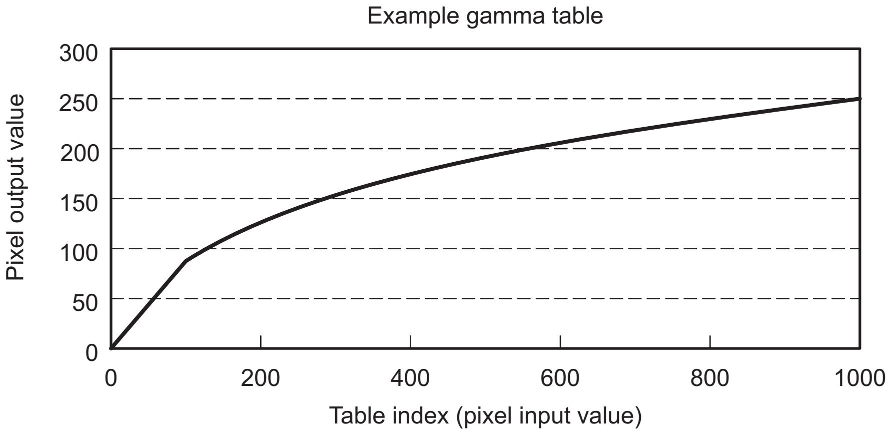
The RGB2RGB blending module has a general 3 × 3 square matrix and redefines the RGB data from the CFA interpolation module, which can be used as a function of a color correction. This is programmable (RGB_MAT1, RGB_MAT2, RGB_MAT3, RGB_MAT4, RGB_MAT5, RGB_OFF1, and RGB_OFF2) so that the color spectrum of the sensor can be adjusted to the human color spectrum. In this module, the following calculation is made:

$$\begin{bmatrix} R_{out} \\ G_{out} \\ B_{out} \end{bmatrix} = \begin{bmatrix} MTX\_RR & MTX\_GR & MTX\_BR \\ MTX\_RG & MTX\_GG & MTX\_BG \\ MTX\_RB & MTX\_GB & MTX\_BB \end{bmatrix} \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix} + \begin{bmatrix} R\_offset \\ G\_offset \\ B\_offset \end{bmatrix}$$

4.3.3.11 Gamma Correction

Gamma correction can be performed on each of the R, G, and B pixels separately by indexing programmable gamma lookup tables. Each table has 1024 8-bit entries. The input data value is used to index into the table and the table content is the output. The gamma table can be optionally bypassed (PCR.GAMMA_BYPASS). In this case, the output of the gamma correction is the 8 MSB's of the 10-bit input. The gamma table can only be written while the preview engine is disabled. Figure 31 shows a sample gamma table.

Figure 31. Gamma Table Example



4.3.3.12 *RGB to YCbCr Conversion, Luminance Enhancement and Chrominance Suppression, Contrast and Brightness, and 4:2:2 Down Sampling and Output Clipping*

4.3.3.12.1 RGB to YCbCr Conversion

The RGB to YCbCr conversion module has a 3 × 3 square matrix and converts the RGB color space of the image data into the YCbCr color space. In this module, the following calculation is made using the contents of the CSC0, CSC1, CSC2, and CSC_OFFSET registers:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} CSCRY & CSCGY & CSCBY \\ CRCRCB & CSCGCB & CSCBCB \\ CSCRCR & CSCGCR & CSCBCR \end{bmatrix} \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix} + \begin{bmatrix} YOFST \\ OFSTCB \\ OFSTCR \end{bmatrix}$$

4.3.3.12.2 Non-Linear Luminance Enhancement

The non-linear luminance enhancement functions as an edge enhancer (crossed in the horizontal direction). It can optionally be enabled or disabled using the PCR.YNENHEN parameter. If it is enabled, a lookup table with 128 20-bit entries must be programmed. Each entry contains a 10-bit signed offset value in the most significant bits, and a 10-bit signed slope in the least significant bits. This format of each entry is shown in Table 22.

Table 22. Non-linear Luminance Enhancement Table Entry Format

MSB	LSB
19-10	9-0
Offset	Slope

4.3.3.12.3 Chrominance Suppression

Occasionally, in very bright portions of an image, only one or two of the color channels may be saturated but the remaining channel(s) may not be. Chrominance suppression can be used to correct this issue. The chrominance suppression operation can optionally be enabled or disabled using the PCR.SUPEN parameter.

Line Width Reduction: If the non-linear luminance enhancer or the chrominance suppression is enabled, the preview engine will reduce the output of this stage by 2 pixels (1 starting pixel - left edge and 1 ending pixel - right edge) in each line. For example, if the input size is 656 × 490 pixels, the output will be 654 × 490 pixels. There will be no truncation of the input data line if both the non-linear luminance enhancer and chrominance suppression are disabled.

4.3.3.12.4 Contrast and Brightness

The luminance component can optionally be adjusted for contrast (scaling/multiplication) and brightness (offset/addition). The contrast is set in the CNT_BRT.CNT field (U8Q4 precision), and the brightness is set in the CNT_BRT.BRT field (U8Q0 precision).

4.3.3.12.5 4:2:2 Down Sampling and Output Clipping

The 4:2:2 conversion module converts the image data to YCbCr-4:2:2 format by averaging every other Cb and Cr component in the horizontal direction. Before outputting the data, the preview engine performs clipping on the YCC components separately. The minimum and maximum threshold values for the Y and C values are specified using the SETUP_YC register. If no clipping is desired, then the register should be set to its reset values of FFh for the maximum Y and C values, and 0 for the minimum Y and C values.

4.3.3.13 Write Buffer Interface

The output of the preview engine may be passed directly to the Resizer (PCR.RSZPORT) and/or written to SDRAM (PCR.SDRPORT). If the output is written to SDRAM, the write address (WSDR_ADDR) and line offset (WADD_OFFSET) should be on 32-byte boundaries. The output format of the YCC data is programmable by setting the PCR.YCPOS parameter. Table 23 indicates the available options.

Table 23. YCC422 Programmable Output Options

MSB		LSB	
31-24	**23-16**	**15-8**	**7-0**
Y1	Cr0	Y0	Cb0
Y1	Cb0	Y0	Cr0
Cb0	Y1	Cr0	Y0
Cr0	Y1	Cb0	Y0

The final width and height of the output will vary depending on which processing functions are enabled. Table 24 indicates how many edge pixels/lines are truncated by enabling certain modules within the preview engine. These values should be subtracted from the input height and width after the averager in order to determine the output size of the preview engine.

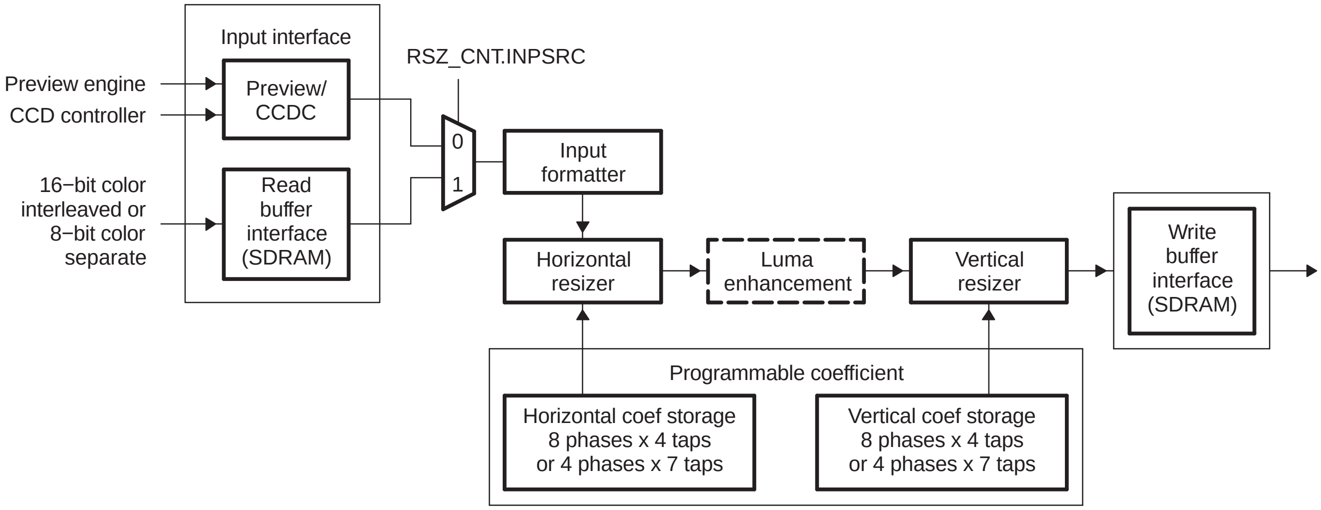
Table 24. Image Cropping by Preview Functions

Function	Pixels/Line	Lines
Noise Filter	4	4
CFA (Bayer)	4	4
CFA (2x down-sampling)	0	2
Color Suppression OR Luminance Enhancement	2	0
Maximum Total	10	8

4.3.4 Resizer

The resizer module performs either upsampling (digital zoom) or down-sampling on image/video data within the range 0.25× to 4×. The input source can be set to either the preview engine/CCD controller or SDRAM/DDRAM and the output is sent to the SDRAM/DDRAM. The resizer module performs horizontal resizing then vertical resizing independently. In between there is an optional edge enhancement feature. This processing flow is shown in Figure 32.

Figure 32. Resizer Processing Flow Block Diagram


Input interface
Preview engine
CCD controller
Preview/ CCDC
16−bit color interleaved or 8−bit color separate
Read buffer interface (SDRAM)
RSZ_CNT.INPSRC
0
1
Input formatter
Horizontal resizer
Luma enhancement
Vertical resizer
Write buffer interface (SDRAM)
Programmable coefficient
Horizontal coef storage 8 phases x 4 taps or 4 phases x 7 taps
Vertical coef storage 8 phases x 4 taps or 4 phases x 7 taps

4.3.4.1 Input and Output Interfaces

The input source can be set to either the preview engine/CCD controller or SDRAM/DDRAM (RSZ_CNT.INPSRC). The input width (IN_SIZE.HORZ) must be at least 32 pixels.

4.3.4.1.1 Preview Engine/CCD Controller Input Mode

In the preview engine/CCD controller input mode, hardware signaling defines input frames. The horizontal starting byte (IN_START.HORZ_ST) and vertical starting line (IN_START.VERT_ST) defines a starting pixel with respect to the upper-left corner of an input image (signaled via horizontal and vertical syncs). The input width and height in the IN_SIZE register specify the exact input range (relative to the starting pixel) needed to generate an output frame of specified width/height. Care must be taken to ensure that the input sizes specified by the IN_START and IN_SIZE registers are less than or equal to the output from the preview engine or CCD controller; otherwise, incorrect hardware operation may occur. SDR_INADD and SDR_INOFF must be programmed to be 0. Also, the output ports of the CCD controller (CCDC.SYN_MODE.SDR2RSZ) and preview engine (PREV.PCR.RSZPORT) to the resizer must be configured so that only one of them is enabled. If both of them are enabled, then the CCD controller will gain control of this interface. If the input is from the CCD controller, then the output of the CCD controller must be in YUV422 format (the resizer does not support resizing raw data from the CCD controller).

4.3.4.1.2 SDRAM Input Mode

In the SDRAM-input mode, the SDRAM address in SDR_INADD points to the 32-byte aligned SDRAM address where the starting pixel resides. The horizontal starting pixel (IN_START.HORZ_ST) defines a starting pixel within that 32-byte alignment; IN_START.HORZ_ST is constrained from 0 to 15 pixels for the YUV422 format, and from 0 to 31 pixels for the RGB format. The vertical starting pixel (IN_START.VERT_ST) must be zero in SDRAM-input mode. The SDR_INOFF register specifies address offset between rows of input data. The input width and height in the IN_SIZE register specify the exact input range (relative to the starting pixel) needed to generate an output frame of specified width/height.

4.3.4.1.3 Input Format

When the input source is from SDRAM/DDRAM, the resizer can be configured for color separate (8 bits/pixel) resizing by setting the RSZ_CNT.INPTYP parameter. In color separate mode, only one color component can be resized at a time. For example, in order to resize all three components (Y, Cb, and Cr) of a color separate image in memory, three separate resize operations must be setup and performed. When the input source is not from SDRAM/DDRAM, then the RSZ_CNT.INPTYP parameter must always be set for YUV422 color interleaved (16 bits/pixel).

When the input is set to 16-bit color interleaved data, the RSZ_CNT.YCPOS parameter specifies the order that the luma and chroma data is received for each input pixel (YC or CY).

4.3.4.1.4 Output Interfaces

In both input modes, the OUT_SIZE register specifies output width/height, the SDR_OUTADD register specifies output starting pixel (upper-left corner) SDRAM address, and the SDR_OUTOFF register specifies SDRAM address offset between the beginning of output rows. Resizer output always goes to SDRAM.

Note that SDR_INADD, SDR_OUTADD, SDR_INOFF, and SDR_OUTOFF all need to be 32-byte aligned; the lower 5-bit of the byte address is assumed zero.

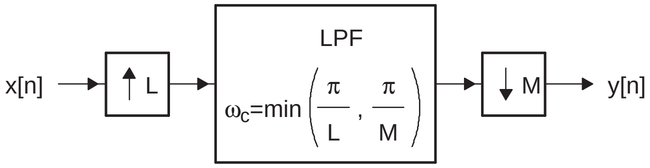
Output Width Constraints: The output width (OUT_SIZE.HORZ) must be at least 16 pixels, and be even (so that the same number of Cb and Cr components are output). Due to the vertical memory size constraint, the output width (OUT_SIZE.HORZ) cannot be greater than 1280 pixels if the vertical resizing ratio is between 0.5× to 4× (RSZ_CNT.VRSZ ≤ 512) and 640 pixels wide if the vertical resizing ratio is between 0.25× to 0.5× (RSZ_CNT.VRSZ > 512).

4.3.4.2 Horizontal and Vertical Resizing

The resizer module has the ability to upsample or downsample image data with independent resizing factors in the horizontal and vertical directions (HRSZ and VRSZ). The HRSZ (RSZ_CNT.HRSZ) and VRSZ (RSZ_CNT.VRSZ) parameters can range from 64 to 1024 to give a resampling range from 0.25× to 4× (256/RSZ).

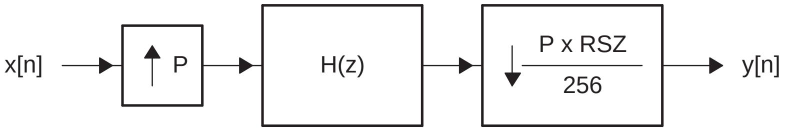
The resizer module uses the same resampling algorithm for both the horizontal and vertical directions. The resizing/resampling algorithm makes use of a programmable polyphase sample rate converter (resampler). Figure 33 depicts a general sample rate converter where the resampling rate is equal to L/M.

Figure 33. Typical Sample Rate Converter



In a typical polyphase implementation, L phases are used. The resizer module, however, fixes the number of phases to 8 phases for the resizing range of 0.5× ~ 4× (RSZ = 64 ~ 512) or 4 phases for a resizing range of 0.25× ~ 0.5× (RSZ = 513 ~ 1024). In this way, the upsampling value (L) is fixed to either 8 or 4, and the down-sampling value (M) is based on RSZ. In order to achieve a resizing ratio of 256/RSZ, the down-sampling value (M) is equal to (P × RSZ)/256, where P is the number of phases. The resizer's functionality can be better depicted in Figure 34.

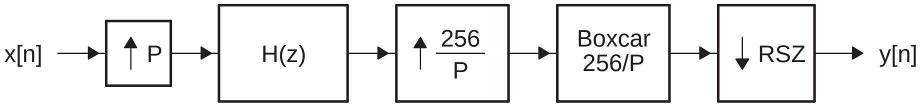
Figure 34. Resizer's Functionality



P = 8, when RSZ = 64 ~ 512; P = 4, when RSZ = 513 ~ 1024

In order to resolve the non-integer down-sampling ratio, this can be better modeled in Figure 35. Here the interpolated output from the filter is upsampled and replicated 256/P times before it is down-sampled by the RSZ factor.

Figure 35. Model of Resizer's Approximation Scheme



P = 8, when RSZ = 64 ~ 512; P = 4, when RSZ = 513 ~ 1024

This implementation means that a resizing ratio with 256/RSZ times the input size can be obtained. However, each output pixel is rounded to the nearest interpolated output of 1/P input pixel precision. For more details on the resampling algorithm, see Section 4.3.4.4.

The polyphase filter coefficients are programmable so than any user-specified filter can be implemented. It is recommended that coefficient sets are chosen to implement a sample rate converter where a low pass filter is used with the following cutoff frequency:

$$\omega_c = \min\left[\frac{\pi}{P}, \frac{\pi}{\left(\frac{P \times RSZ}{256}\right)}\right]$$

If this polyphase resampling methodology is used, then all upsampling factors can share the same set of coefficients. However, a different coefficient set is required when changing between 8-phase and 4-phase modes, and with different down-sampling factors.

There are 32 programmable coefficients available for the horizontal direction (HFILT10–HFILT3130 registers) and another 32 programmable coefficients for the vertical direction (VFILT10–VFILT3130 registers). The 32 programmable coefficients are arranged as either 4-taps and 8-phases for the resizing range of 0.5× ~ 4× (RSZ = 64 ~ 512) or 7-taps and 4-phases for a resizing range of 0.25× ~ 0.5× (RSZ = 513 ~ 1024). Table 25 shows the arrangement of the 32 filter coefficients. Each tap is arranged in an S10Q8 format (signed value of 10-bits with 8 of them being the fraction).

Table 25. Arrangement of the Filter Coefficients

	0.5× to 4×		0.25× to ~0.5×	
Filter Coefficient	**Phase**	**Tap**	**Phase**	**Tap**
0	0	0	0	0
1		1		1
2		2		2
3		3		3
4	1	0		4
5		1		5
6		2		6
7		3		Not used
8	2	0	1	0
9		1		1
10		2		2
11		3		3
12	3	0		4
13		1		5
14		2		6
15		3		Not used
16	4	0	2	0
17		1		1
18		2		2
19		3		3
20	5	0		4
21		1		5
22		2		6
23		3		Not used
24	6	0	3	0
25		1		1
26		2		2
27		3		3
28	7	0		4
29		1		5
30		2		6
31		3		Not used

The indexing scheme of coefficients is oriented for dot-product (or inner product), rather than for impulse response. In other words, the first data point contributing toward a particular output is multiplied with the coefficient associated with tap 0, and last data point is multiplied with the coefficient associated with tap 3 or tap 6 (depending on 4-tap or 7-tap mode). The normal raster-scan order is used where the upper-left corner gets the (0, 0) coordinate. Pixel 0 is the left-most column of pixels for horizontal resizing, and the top-most row of pixels for vertical resizing. Figure 36 shows an example of the alignment of the input pixels to tap coefficients using a simple 1:1 resize case (4-tap mode). In this example, the output of only one phase is needed.

Figure 36. Alignment of Input Pixels to Tap Coefficients

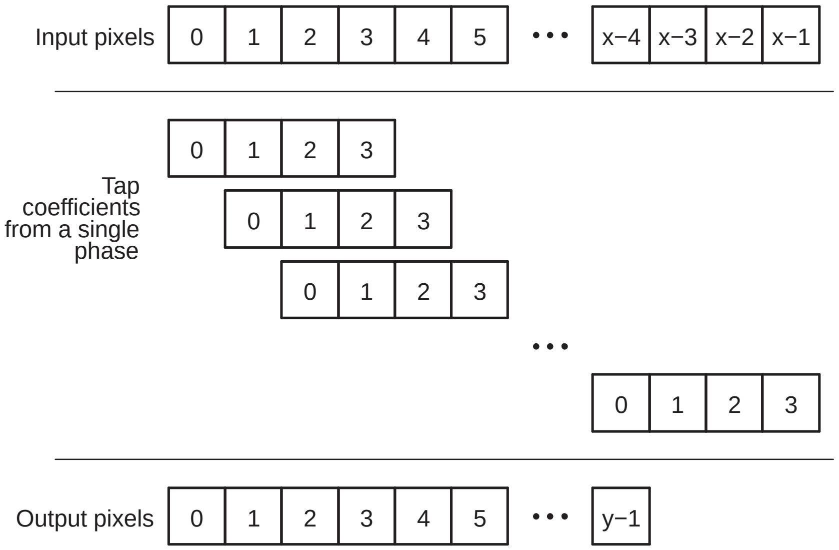


Figure 36 also presents how the first output is computed when all of the taps have been aligned with actual input pixels. In order to compute the last several pixels in each line/column, the filter needs more input pixels than the following equation calculates:

$$\textit{input size} = \textit{output size} \times \textit{RSZ/256}$$

For the example above where the input size is X and the output size is Y, three extra input pixels are required to generate the correct number of output pixels:

$$X = Y \times 256/256 + 3 \text{ (to account for the extra pixels required for filtering)}$$

The actual input size calculation also depends on the starting phase and rounding issues in the algorithm. Table 26 lists the actual input size calculations that are derived from the algorithm description in Section 4.3.4.4. The input width and height parameters must be programmed strictly according to these equations, otherwise, incorrect hardware operation may occur.

Table 26. Input Size Calculations[1]

	8-Phase, 4-Tap Mode	4-Phase, 7-Tap Mode
IN_SIZE.HORZ	(32 × sph + (ow – 1) × hrsz + 16) >> 8 + 7	(64 × sph + (ow – 1) × hrsz + 32) >> 8 + 7
IN_SIZE.VERT	(32 × spv + (oh – 1) × vrsz + 16) >> 8 + 4	(64 × spv + (oh – 1) × vrsz + 32) >> 8 + 7

[1] sph = Horizontal starting phase (RSZ_CNT.HSTPH)
spv = Vertical starting phase (RSZ_CNT.VSTPH)
ow = Output width (OUT_SIZE.HORZ)
oh = Output height (OUT_SIZE.VERT)
hrsz = Horizontal resizing value (RSZ_CNT.HRSZ)
vrsz = Vertical resizing value (RSZ_CNT.VRSZ)
Extra = 0, when YENH.ALGO = 0 (edge enhancement disabled); 4, when YENH.ALGO ≠ 0 (edge enhancement enabled)

The horizontal and vertical starting phases can be programmed in the RSZ_CNT.HSTPH and RSZ_CNT.VSTPH fields, respectively. The chrominance data can be resized using either bilinear interpolation, or the same algorithm as the luminance data (RSZ_CNT.CBILIN). For more information on how these fields are used in the algorithm, see Section 4.3.4.4.

4.3.4.3 Luma Edge Enhancement

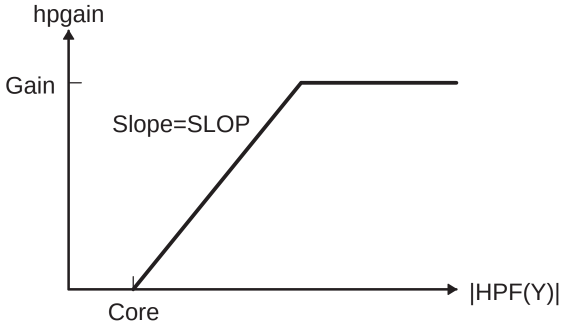
Edge enhancement can be optionally applied to the horizontally resized luminance component before the output of the horizontal stage is sent to the line memories and the vertical stage. The YENH.ALGO bit is cleared to disable edge enhancement, or select either a 3-tap or a 5-tap horizontal high-pass filter for luminance enhancement.

If edge enhancement is selected, the two left-most and two right-most pixels in each line are not output to the line memories and the vertical stage. The OUT_SIZE.HORZ setting is the final output width, up to 1280 pixels when vertical 4-tap mode is used and up to 640 pixels when vertical 7- tap mode is used. When edge enhancement is enabled, the horizontal resizer output width used to calculate the required input width must be OUT_SIZE.HORZ + 4.

The high-pass gain is computed by mapping the absolute value of high passed luma with the curve in Figure 37.

YENH.CORE is in U8Q0, or unsigned 8-bit integer format. YENH.SLOP is in U4Q4, or unsigned 4-bit fraction format. YENH.GAIN is in U4Q4, or unsigned 4-bit fraction format. Hpgain is computed with sign/integer bits plus 4-bit of fraction, but can be saturated from 0 to 15 (representing 0..15/16) before clipping by GAIN.

Figure 37. High-Pass Gain as a Function of Absolute High Passed Luma



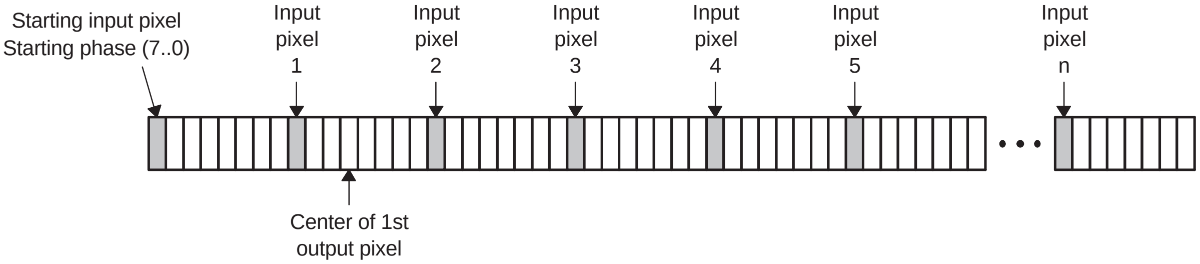
4.3.4.4 Resampling Algorithm

The resizer module uses the same resampling algorithm for both the horizontal and vertical directions. For the rest this section, the horizontal direction is used in describing the resampling algorithm. The algorithm is described first for the 4-tap/8-phase mode, and then the 7-tap/4-phase mode.

4.3.4.4.1 4-Tap/8-Phase Mode

In the 4-tap/8-phase mode, the coefficients for each of the 8 phases may be set to interpolate 8 intermediate pixels in between each input pixel. For each output pixel calculation, a fine input pointer with 1/256 input pixel precision is incremented by the RSZ value. A coarse input pointer with 1/8 input pixel precision (corresponds to one of the 8 phases) is calculated by rounding the fine input pointer to the nearest 1/8 pixel. The output pixel is calculated by the dot product of the coefficients of the phase filter (selected by the coarse input pointer) and the appropriate four input pixels. Figure 38 shows a pseudo-code description of the resizer algorithm in the 4-tap/8-phase mode.

Figure 38. Pseudo-Code Description of the Resizer Algorithm: 4-Tap/8-Phase Mode

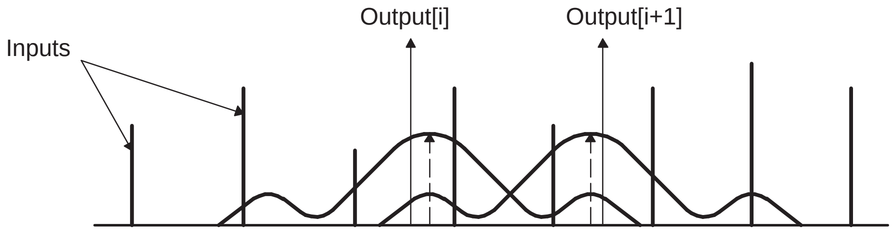


- The starting input pixel location (in whole pixels) and the starting phase (in 1/8 pixel) are programmed via the resizer registers.
- A fine input pointer is maintained in 1/256 pixel precision.
- A coarse input pointer and a pixel input pointer are computed for each output based on fine input pointer.
- The coarse input pointer is in 1/8 pixel precision. The pixel input pointer is in whole pixel precision.
- Initially fine input pointer = 256 × starting input pixel + 32 × starting phase – 256. The fine input pointer defines the starting 1/8 pixel location covered by the filter waveform.
- For each output pixel:

```
Coarse input pointer = (fine input pointer + 16) >> 5   /* round to nearest phase */
Pixel input pointer = (coarse input pointer >> 3) + 1   /* round up to a whole pixel, when already
                                                           on an integer pixel, go to next one to
                                                           simplify coefficient organization */
Coefficient phase = (coarse input pointer & 7)          /* 3 LSBs = phase */
Output = dot product of the 4 coefficients and the 4 inputs starting with pixel input pointer
Clip output to 8-bit unsigned for luma, 8-bit signed for chroma
Fine input pointer = fine input pointer + RSZ
    /* distance between outputs = 1/resize_factor = RSZ/256 = RSZ in 1/256 precision */
```

- Same algorithm in both the horizontal and vertical directions, but with separate initial pixel/phase values and separate RSZ values

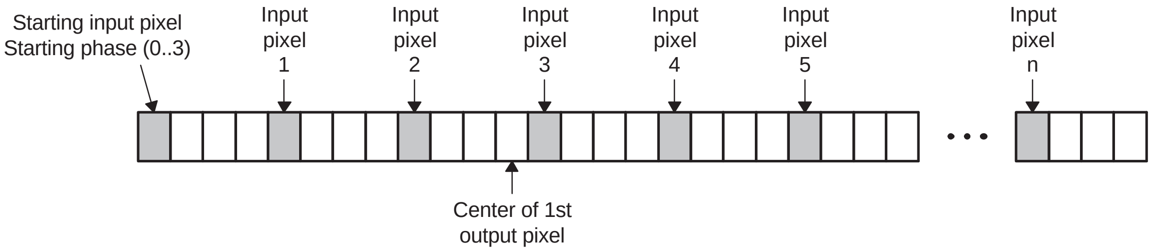
Figure 39. Resampling Algorithm for 4 Taps and 8 Phases



4.3.4.4.2 7-Tap/4-Phase Mode

In the 7-tap/4-phase mode, the coefficients for each of the 4 phases may be set to interpolate 4 intermediate pixels in between each input pixel. For each output pixel calculation, a fine input pointer with 1/256 input pixel precision is incremented by the RSZ value. A coarse input pointer with 1/4 input pixel precision (corresponds to one of the 4 phases) is calculated by rounding the fine input pointer to the nearest 1/4 pixel. The output pixel is calculated by the dot product of the coefficients of the phase filter (selected by the coarse input pointer) and the appropriate 7 input pixels. Figure 40 shows a pseudo-code description of the resizer algorithm in the 7-tap/4-phase mode.

Figure 40. Pseudo-Code Description of the Resizer Algorithm: 7-Tap/4-Phase Mode

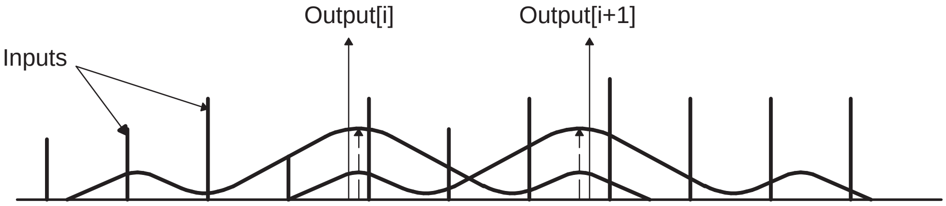


- The starting input pixel location (in whole pixels) and the starting phase (in 1/4 pixel) are programmed via the resizer registers.
- A fine input pointer is maintained in 1/256 pixel precision.
- A coarse input pointer and a pixel input pointer are computed for each output based on fine input pointer.
- The coarse input pointer is in 1/4 pixel precision. The pixel input pointer is in whole pixel precision.
- Initially fine input pointer = 256 × starting input pixel + 64 × starting phase – 256. The fine input pointer defines the starting 1/4 pixel location covered by the filter waveform.
- For each output pixel:

```
Coarse input pointer = (fine input pointer + 32) >> 6    /* round to nearest phase */
Pixel input pointer = (coarse input pointer >> 2) + 1    /* round up to a whole pixel, when
already
                                                            on an integer pixel, go to next one to
                                                            simplify coefficient organization */
Coefficient phase = (coarse input pointer & 3)           /* 2 LSBs = phase */
Output = dot product of the 7 coefficients and the 7 inputs starting with pixel input pointer
Clip output to 8-bit unsigned for luma, 8-bit signed for chroma
/* It is acceptable to require the 8th coefficients to be filled with zero by firmware so that
   8 coefficients and 8 inputs, the last input being don't care value, are multiply-added */
Fine input pointer = fine input pointer + RSZ
/* distance between outputs = 1/resize_factor = RSZ/256 = RSZ in 1/256 precision */
```

- Same algorithm in both the horizontal and vertical directions, but with separate initial pixel/phase values and separate RSZ values

Figure 41. Resampling Algorithm for 7 Taps and 4 Phases



Note that the pixel input pointer, pip, in the algorithm description "points to" pixels, not bytes or shorts in the memory. The fine input pointer, fip, points to a 1/256 resolution subpixel position. The coarse input point, cip, points to an 1/8 or 1/4 resolution subpixel location, depending on the number of phases.

4.3.4.4.3 Horizontal Resizing With Interleaved Chroma

Chroma inputs, Cb and Cr, are 8-bit unsigned values that represents 128-biased 8-bit signed values (the signed chroma is called U and V instead of Cb and Cr). During the resizing computation, the chroma values have the 128 bias subtracted to convert to the 8-bit signed format. After vertical resizing, the 128 bias is added back to convert back to 8-bit unsigned format.

Chroma components, which are 2:1 horizontally down-sampled with respect to luma, have two methods of horizontal resizing processing: filtering with luma and bilinear interpolation. This option can be selected in the RSZ_CNT.CBILIN field independent of the HRSZ/VRSZ parameters. However, filtering with luma is intended only for down-sampling, and bilinear interpolation is intended only for upsampling.

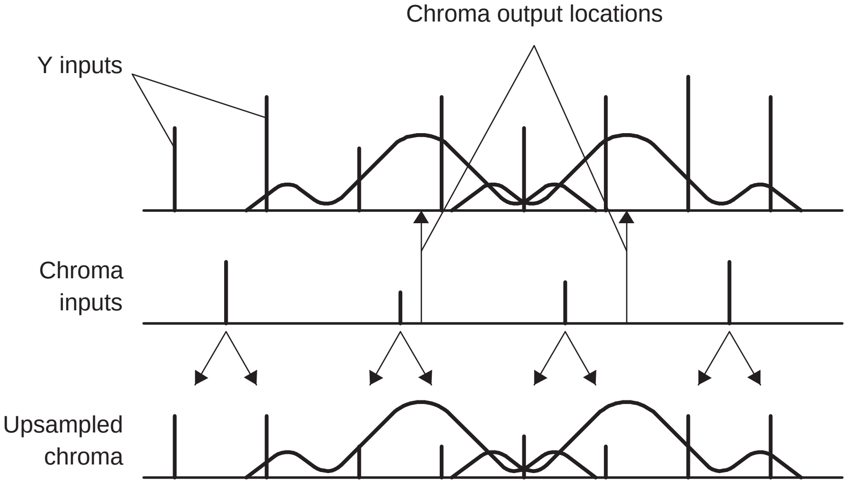
Note that the algorithm descriptions in Figure 34 and Figure 35 are color-neutral, but are for full resolution samples; thus, are applicable for horizontal resizing of Y and vertical resizing of Y/Cb/Cr. For horizontal resizing of Y/Cb/Cr in a combined filtering flow, the algorithm is modified as shown in the following algorithm descriptions:

Filter Chroma with Luma (4-tap/8-phase mode):

```
For (I=0; i<output_width; I++) {
   Coarse input pointer = (fine input pointer + 16) >> 5    /* round to nearest phase */
   Pixel input pointer = (coarse input pointer >> 3) + 1    /* round up to a whole pixel */
   Coefficient phase = pixel input pointer & 7              /* 3 LSBs = phase */
   if (I & 1 == 0) { /* even output pixel, generate YCbCr */
       Yout = dot product of the 4 coefficients and the 4 Y inputs starting with pixel input
pointer
       Cbout = dot product of the 4 coefficients and the 4 upsampled Cb inputs starting with
pixel input pointer
       Crout = dot product of the 4 coefficients and the 4 upsampled Cr inputs starting with
pixel input pointer
       Clip outputs to 8-bit unsigned for luma, 8-bit signed for chroma

   }
   Else { /* odd output pixel, generate Y only */
       Yout = dot product of the 4 coefficients and the 4 Y inputs starting with pixel input
pointer
       Clip output to 8-bit unsigned
   }
   Fine input pointer = fine input pointer + RSZ
 }
}
```

```
Filter Chroma with Luma (7-tap/4-phase mode):
For (I=0; i<output_width; I++) {
    Coarse input pointer = (fine input pointer + 32) >> 6    /* round to nearest phase */
    Pixel input pointer = (coarse input pointer >> 2) + 1    /* round up to a whole pixel */
    Coefficient phase = pixel input pointer & 3              /* 2 LSBs = phase */
    if (I & 1 == 0) { /* even output pixel, generate YCbCr */
       Yout = dot product of the 7 coefficients and the 7 Y inputs starting with pixel input
pointer
       Cbout = dot product of the 7 coefficients and the 7 upsampled Cb inputs starting with
pixel input pointer
       Crout = dot product of the 7 coefficients and the 7 upsampled Cr inputs starting with
pixel input pointer
       Clip outputs to 8-bit unsigned for luma, 8-bit signed for chroma
    }
    Else { /* odd output pixel, generate Y only */
       Yout = dot product of the 7 coefficients and the 7 Y inputs starting with pixel input
pointer
       Clip output to 8-bit unsigned
    }
    Fine input pointer = fine input pointer + RSZ
  }
}
```

Figure 42. Chroma Processing Option 1: Filter With Luma

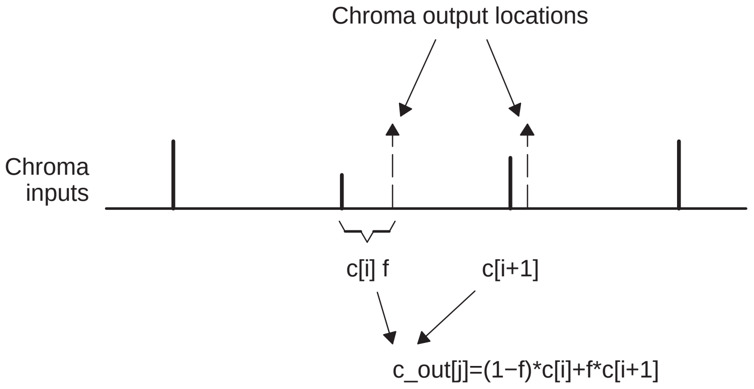


The chroma input values are internally replicated to realize 1:2 upsampling to line up with luma input values. Only required chroma outputs are computed; they correspond to the even luma outputs.

For the bilinear interpolation flow of chroma horizontal resizing, the algorithm is adapted as follows. For the bilinear interpolation option, it is not necessary to replicate chroma samples.

```
Bilinear Interpolation (either 4-tap or 7-tap):

For (I=0; i<output_width; I++) {
  if (I & 1 == 0) { /* even output pixel, generate YCbCr */
     Coarse input pointer = ...
     Pixel input pointer = ...
     Yout = dot product of ...
     C_fine_input_pointer = fine_input_pointer + 128*ntaps    /* points to center of filter
kernel */
     Cidx = C_fine_input_pointer >> 9            /* truncate to even pixel grid to find left
value */
     Cbin[0] = Cb[Cidx]
     Cbin[1] = Cb[Cidx + 1]
     Crin[0] = Cr[Cidx]
     Crin[1] = Cr[Cidx + 1]
     frac = C_fine_input_pointer & 511                /* 9-bit fraction */
     Cbout = ((512 – frac) * Cbin[0] + frac * Cbin[1] + 256) >> 9
     Crout = ((512 – frac) * Crin[0] + frac * Crin[1] + 256) >> 9
     Clip outputs to 8-bit unsigned for luma, 8-bit signed for chroma
     Fine input pointer = fine input pointer + RSZ
  }
  else { /* odd output pixel, generate Y */
     ...
  }
}
```

Figure 43. Chroma Processing Option 2: Bilinear Interpolation



In Figure 43, the *f* variable represents a real-number quantity, so we use *f* and 1 – *f* for the weights in the interpolation. On the algorithm above, fixed-point arithmetic is used. The variable *frac* is an unsigned integer representing the fraction *f*. Thus 1 – *f* becomes 512 – *frac*. After the sum of products, 256, representing 0.5 real-number, is added to the sum, then the sum is right-shifted by 9 bits to get back to the integer chroma representation.

In either algorithm options, the chroma outputs computed are interleaved with luma values to generate the YCbYCr output format (or the alternate format specified in RSZ_CNT.YCPOS).

Note that in the vertical resizing stage, the two chroma planes are processed interleaved as one separate image. Since there is no resolution issue vertically, and no horizontal dependency in vertical resizing, the vertical scheme is consistent with conventional processing flow, and will not be analyzed here.

4.3.4.4.4 Example of Algorithm Functionality

An example of 1:2.56 (hrsz = 100) horizontal resizing is shown in Table 27 to illustrate the address calculation and chroma processing in 4:2:2 format (4-tap 8-phase mode). The starting pixel and phase are assumed to be zero.

Table 27. Processing Example for 1:2.56 Horizontal Resize

Output	Y0	Cb0	Cr0	Y1	Y2	Cb2	Cr2	Y3	Y4	Cb4	Cr4	Y5
fip (+= hrsz)	-256			-156	-56			44	144			244
cip (= (fip + 16) >> 5)	-8			-5	-2			1	5			8
pip (= (cip >> 3) + 1)	0			0	0			1	1			2
coef ph (= cip & 7)	0			3	6			1	5			0
Inputs needed (chroma filtered like luma)	Y0	Cb0	Cr0	Y0	Y0	Cb0	Cr0	Y1	Y1	Cb0	Cr0	Y2
	Y1	Cb0	Cr0	Y1	Y1	Cb0	Cr0	Y2	Y2	Cb2	Cr2	Y3
	Y2	Cb2	Cr2	Y2	Y2	Cb2	Cr2	Y3	Y3	Cb2	Cr2	Y4
	Y3	Cb2	Cr2	Y3	Y3	Cb2	Cr2	Y4	Y4	Cb4	Cr4	Y5
Cfip (= fip + 512)		256				488				720		
Cidx (= Cfip >> 9)		0				0				1		
Inputs needed for chroma bilinear interpolation		Cb0	Cr0			Cb0	Cr0			Cb2	Cr2	
		Cb2	Cr2			Cb2	Cr2			Cb4	Cr4	

Note the distinction between using {Cb0, Cb0, Cb2, Cb2} and {Cb0, Cb2, Cb2, Cb4} as input to the filter. The 4 filter taps are applied in order, so with the different chroma component repetition, the result will be different (even when the coefficient phase is the same).

Also note the chroma bilinear interpolation flow’s Cidx points to the chroma sample in linear array order, so Cidx = 1 means we are grabbing Cb2 and Cb4.

4.3.5 Camera Control Modules

4.3.5.1 Hardware 3A (H3A)

The H3A module is designed to support the control loops for Auto Focus (AF), Auto White Balance (AWB) and Auto Exposure (AE) by collecting metrics about the imaging/video data. The H3A module only works on raw color Bayer pattern data. The H3A module does not work with YUV modes.

Metrics collected by the H3A module are used to adjust the various parameters for processing the imaging/video data. There are 2 main blocks in the H3A module:

- Auto Focus (AF) Engine. The AF engine extracts and filters the red, green, and blue data from the input image/video data and provides either the accumulation or peaks of the data in a specified region.
- Auto Exposure and Auto White Balance (AE/AWB) Engine. The AE/AWB engine accumulates the values and checks for saturated values in a subsampling of the video data.

The number, dimensions, and starting position of the AF regions and the AE/AWB regions windows are separately programmable.

4.3.5.1.1 Optional Preprocessing

The input of the H3A module is 10-bit raw data from the CCD controller. A 10-bit to 8-bit A-law compression step may be separately enabled and disabled for both the AF engine and the AE/AWB engine. In the case of the A-law table being enabled, the output is still 10-bits with the upper two bits filled with a 0. A-law compression offers added protection against overflowing the accumulators.

For the AF processing flow, a horizontal median filter can be enabled and disabled before the A-law compression. This filter is useful for reducing temperature induced noise effects.

4.3.5.1.2 Auto Focus Engine

The Auto Focus (AF) engine works by extracting each red, green, and blue pixel from the video stream and subtracting a fixed offset from the pixel value (128 when A-law is enabled or 512 when A-law is disabled). The offset value is then passed through two IIR filters, and the absolute values of the filter outputs are the focus values (FV).

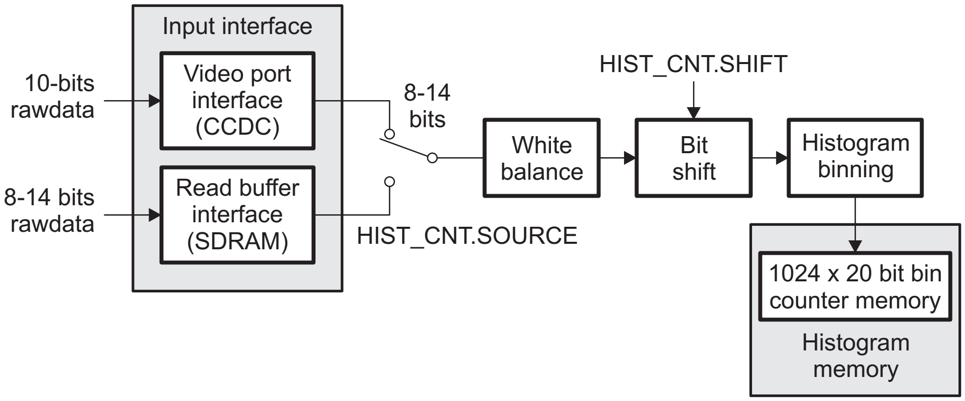
4.3.5.1.3 Auto Exposure and Auto White Balance Engine

The Auto Exposure and Auto White Balance (AE/AWB) engine starts by dividing the frames into windows and further sub-sampling each window into 2 × 2 blocks. Then, for each of the sub-sampled 2 × 2 blocks, each pixel is accumulated. Also, each pixel is compared to a limit set in a register. If any of the pixels in the 2 × 2 block are greater than or equal to the limit, then the block is not counted in the unsaturated block counter. All pixels greater than the limit are replaced by the limit and the value of the pixel is accumulated.

4.3.5.2 Histogram (HIST)

The histogram accepts RAW image/video data from either the video port interface of the CCD/CMOS controller or SDRAM/DDRAM, performs a color-separate gain on each pixel (white/channel balance), and bins them according to the amplitude, color, and region which are all specified via its register settings. It can support 4 color Bayer, and up to 4 regions simultaneously. Figure 44 shows the processing flow of the histogram module.

Figure 44. Histogram Processing Flow



4.3.5.2.1 Input Interface

The histogram receives RAW image/video data from either the video port interface via the CCD/CMOS controller module (which is interfaced to an external CCD/CMOS sensor) or from the read buffer interface via the SDRAM/DDRAM (HIST_CNT.SOURCE). The input data is 10-bits wide if the source is the video port interface.

When the input source is from SDRAM/DDRAM, the data bit width can range from 8 to 14 bits. If the input data is 8 bits packed (memory contains two 8-bit pixels for every 16 bits), then the HIST_CNT.DATSIZ bit should be set. Likewise, if memory contains one pixel for every 16 bits, then the HIST_CNT.DATSIZ bit should be cleared. The input SDRAM/DDRAM address (RADD) and line offset (RADD_OFF) registers are used to specify the location of the input frame in memory. Both of these registers must be aligned on 32-byte boundaries. The frame input width and height is configured using the H_V_INFO.HSIZE and H_V_INFO.VSIZE register fields, respectively.

The histogram module supports 4-color Bayer color pattern. See Figure 45. The HIST_CNT.CFA field is used to select the color pattern of the input data (must be Bayer).

Figure 45. Color Pattern Indices

Bayer

0	1
2	3

4.3.5.2.2 White Balance

A white balance gain can be separately applied to each of the color channels by programming the fields in the WB_GAIN register. Table 28 indicates which pixel index in the color pattern corresponds to each field in the WB_GAIN register.

Table 28. White Balance Field-to-Pattern Assignments

WB_GAIN Field	Bayer
WB00	1
WB01	2
WB02	3
WB03	4

Each gain constant is 8-bits wide with 5-bits of decimal precision (U8Q5).

4.3.5.2.3 Histogram Binning

The histogram bins the input data by amplitude, color, and region. Each bin is a counter, counting the number of pixels of a color in the range associated with the bin. The number of bins can be programmed to 32, 64, 128, or 256 bins in the HIST_CNT.BINS field. However, due to the limited histogram memory size (1024 words), the number of bins (times 4 colors) limits the number of regions that can be active. See Table 29.

Table 29. Regions and Bins for Histogram

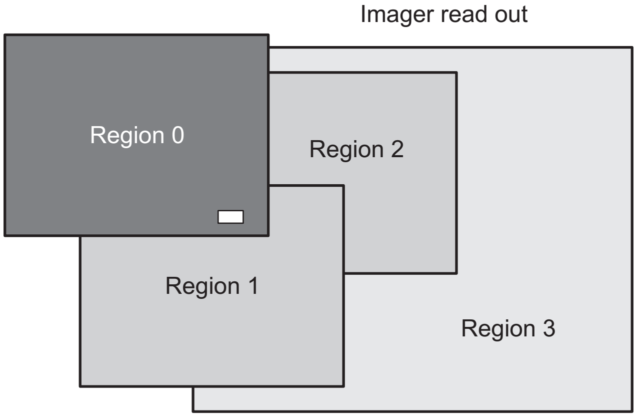
Number of Bins	Number of Regions Allowed
256	1
128	2
64	4
32	4

As indicated in Table 29, up to four overlapping regions can be designated within the frame. Each region is defined by the horizontal starting (Rn_HORZ.HSTART) and ending (Rn_HORZ.HEND) pixel, and vertical starting (Rn_VERT.VSTART) and ending (Rn_VERT.VEND) line (where n is the region number 0..3). If the starting and ending pixel/line is the same, then the region size is treated as zero, and there is no binning for such a region.

4.3.5.2.3.1 Region Priority

Up to four regions can be active at any time, but a pixel is only binned into one region. The priority is Region 0 > Region 1 > Region 2 > Region 3. The white pixel in Figure 46 will be binned only for Region 0, although it is present in all the four regions.

Figure 46. Region Priority



4.3.5.2.3.2 Bin Clipping and Right Shifting

The number of least-significant input bits that are used for binning is equal to $\log_2$(Number of Bins). If the input bit width is larger than this, then the higher bits will be clipped to the highest bin location. This allows data from above the bin range to be included in the upper-most bin.

Example 1.

1 Region enabled, 256 bins per color ($\log_2(256) = 8$)

Input bit width = 10 bits (so the upper 2 bits will be clipped to the highest bin)

Shift = 0, Input Pixel value = 1000

Pixel value (1000) > Max bin index (255), (so bin 255 is incremented)

Before the data is evaluated and binned, it is right shifted by the number of bits specified in the HIST_CNT.SHIFT register field. The value of the HIST_CNT.SHIFT register field determines the range of each bin. Right shifting the input data basically increases the range of input values that will be sent to a single bin. The number of input pixel values stored in a single bin is equal to 2^{SHIFT}. For example, a right shift of 0 will have only one pixel value per bin (except possibly the highest bin), while a right shift of 4 will have 2^4=16 pixel values per bin. This can be used to avoid clipping so that the entire range of bins can be used for different input width sizes. The following calculation can be used to avoid bin clipping:

HIST_CNT.SHIFT = (Input Bit Width) - $\log_2$(Number of Bins)

Table 30 implements this equation for different input bit widths and number of bins (the shaded row indicates the input bit width when the input is from the video port interface of the CCD/CMOS controller).

Table 30. Recommended SHIFT Value to Avoid Bin Clipping

	Number of Bins / log2(bins)			
Input Bit Width	**256/ 8**	**128/ 7**	**64/ 6**	**32/ 5**
8	0	1	2	3
9	1	2	3	4
10	2	3	4	5
11	3	4	5	6
12	4	5	6	7
13	5	6	7	clip
14	6	7	clip	clip

4.3.5.2.3.3 Bin Saturation

The histogram bin counter memory is 20-bits wide. If incrementing a histogram bin would cause the value to become greater than what this memory word could hold, the value is saturated to the maximum value, which is $2^{20} - 1$.

4.3.5.2.4 Histogram Memory Output Format

The histogram output memory contains 1024 20-bit words. The data can be written or read using the HIST_ADDR and HIST_DATA registers. The memory offset address is first written to the HIST_ADDR register, and then the HIST_DATA register points to the specified memory location so that a read or write can be made to the counter. Each access to the HIST_DATA register auto-increments the address, so contiguous memory reads can be done without changing the HIST_ADDR address pointer. This memory can explicitly be cleared by writing zeros to each address. Optionally, the HIST_CNT.CLR register field can be set so that each address will have its data automatically cleared after a read.

The histogram output memory is first partitioned into regions, depending on the number of bins used. Table 31 indicates the starting address of each region in the histogram output memory.

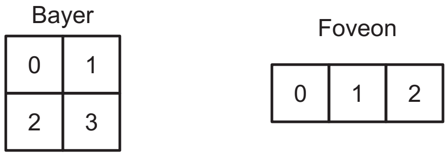
Table 31. Region Offset Addresses

	Number of Bins			
Region Number	**256**	**128**	**64**	**32**
Region0	0	0	0	0
Region1		512	256	128
Region2			512	256
Region3			768	384

Each region is further divided into color areas. Using the color pattern indices from Figure 47, the following equation is used to determine the color offset into each region.

Color Offset = (Color Index) × *(Number of Bins)*

Figure 47. Color Pattern Indices



This equation is implemented in Table 32.

Table 32. Color Offset Addresses within Each Region

	Number of Bins			
Color	**256**	**128**	**64**	**32**
0	0	0	0	0
1	256	128	64	32
2	512	256	128	64
3	768	384	192	96

Within each color area are the individual pixel amplitude value bins. The address offset into each color area represents the bin value.

4.4 VPFE Arbitration and Data Transfer

The Shared Buffer Logic (SBL) Central Resource manages the flow of data between VPSS modules and the DDR EMIF. In order to efficiently utilize the external DDR2 bandwidth, the shared buffer logic/memory via a high-bandwidth bus (64-bit wide) interfaces with the DMA system. The shared buffer logic/memory also interfaces with all the VPFE and VPBE modules via a 128-bit wide bus.

Due to the real-time demands of image processing at high speeds and image resolutions, the VPSS demands the highest bandwidth requirements in the system. Table 33 lists the maximum data throughput capabilities of each module in the VPSS.

Table 33. Maximum Data Throughput Capabilities

Module	Read Bandwidth (MB/s)	Write Bandwidth (MB/s)
CCD controller	~10	SYSCLK/3
Preview Engine	225	SYSCLK/3
Resizer	SYSCLK/3	4 × SYSCLK/3
Histogram	SYSCLK/3	-
Hardware 3A (H3A)	-	~2
On-Screen Display (OSD)	175	-

Sustained performance at these rates cannot be obtained, since the sum of these rates is greater than the DDR2 peak available bandwidth. However, instantaneous peak traffic can be supported by internal buffering. The VPSS DMA master is required to be the highest priority (default) in the system to assure functionality. This priority can be lowered (but not advised) by modifying the PCR.DMA_PRI register field of the VPSS module.

4.4.1 VPSS DMA Transfer Behavior

The information given in this section discusses hardware operation that is transparent to you, but an understanding of this may enable optimization of system performance.

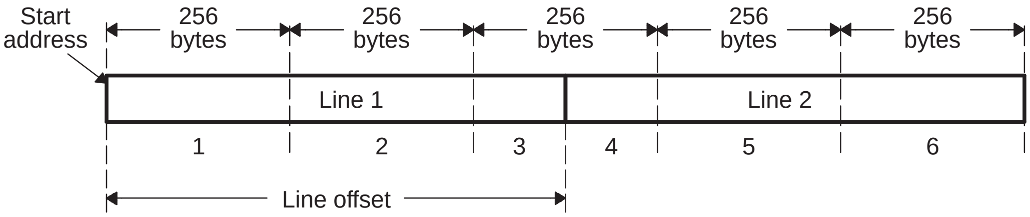
Each DDR2 request is for up to 256 bytes and may take up to 40 DMA cycles to complete the transfer:

(8 bytes/cycle × 32 cycles) + (8 cycle setup overhead) =
256 bytes in 40 cycles

Each data transfer cannot span a 256-byte boundary in DDR memory. If the starting address of a read/write request is on a 256-byte boundary, then the first request transfers the entire 256-byte block beginning at the starting address. However, if the starting address is not on a 256-byte boundary, then the first request only transfers the data from the starting address to the first 256-byte boundary it comes to, resulting in a transfer that is less than 256 bytes. Then subsequent requests each transfer 256 bytes (since they will be aligned to 256-byte boundaries) until it gets to the end of the line. At the end of each line, only the amount needed to complete the line is transferred.

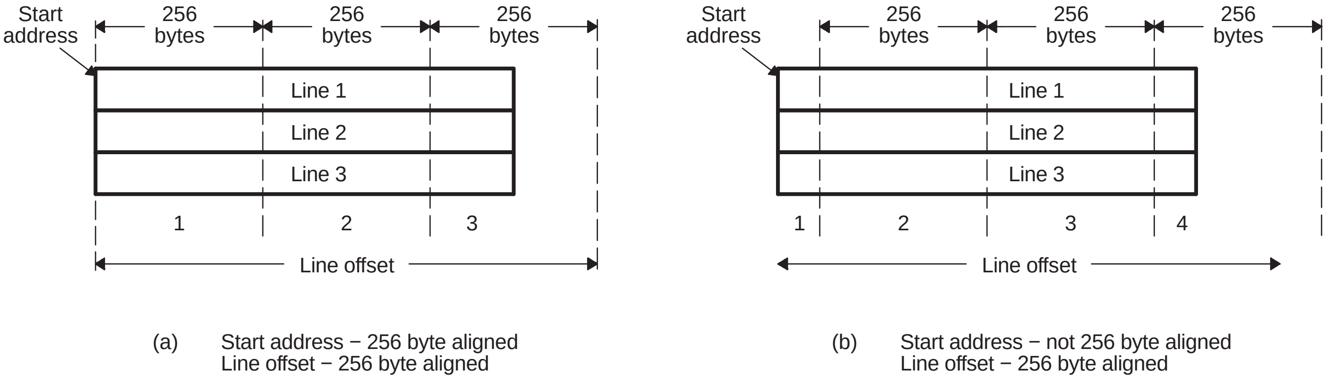
Each line of a frame is transferred separately. In other words, data from the end of one line and data from the beginning of the next line cannot be transferred in the same request, even if they are in the same 256-byte block of memory. Instead, one request is made for the last part of the first line, and a new request is made for the first part of the next line. In Figure 48, the starting address is on a 256-byte boundary and each line width and line offset are set to 640 bytes, each line has 3 requests for a total of 6 requests for both lines, not 5 requests.

Figure 48. Number of DMA Transfer Requests per Line



Depending on the width of each line, not programming the starting address and/or line offset registers to be on 256-byte boundaries (32- or 64-byte boundaries are required) may result in more DMA requests. Figure 49 shows two cases of programming the starting address and line offset register.

Figure 49. Alignment of Starting Address Pointer



In case (a), where both the start address and line offset are 256-byte aligned, there are consistently 3 requests for each line. In case (b), where only the line offset was 256-byte aligned, there are consistently 4 requests for each line. In the cases where the line offset is not 256-byte aligned, occasionally a line starts on a 256-byte boundary, but the rest of the time it is not aligned. This is more difficult to model. Table 34 indicates this truth table, where Y stands for being 256-byte aligned, and N stands for not 256-byte aligned. The - in the performance column means somewhere in between the best case and the worst case.

Table 34. Alignment Performance

Start Address	Line Offset	Performance
Y	Y	Optimal
Y	N	-
N	Y	Worst
N	N	-

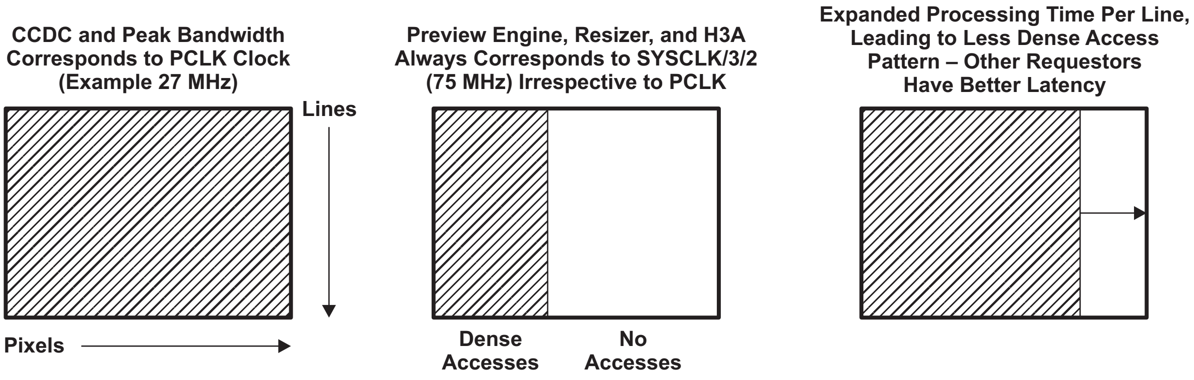
4.4.2 VPSS DMA Bandwidth Adjustments

Since the VPSS is the highest priority in the system, it processes and transfers data at the highest rate possible. In the case where this processing returns results long before real-time deadlines, this may have a negative effect on the performance of other peripherals in the system by unnecessarily wasting DMA resources. The VPSS offers two methods of adjustments that can be made to slow down data processing in this situation. One can be made when the sensor input to the CCD controller is the input source, and the other can be made when the SDRAM/DDRAM is the source of the input image.

4.4.2.1 Input from CCD Controller Video Port Interface

The video port interface delivers data at a rate independent of the PCLK. By default, this rate is set to (SYSCLK/3)/2. When the PCLK is at a lower frequency, then it is unnecessary for the video port interface to operate at such a high frequency. The FMTCFG.VPIF_FRQ field of the CCD controller can be programmed to reduce the rate at which the video port delivers new data to the other modules (preview engine, H3A, and histogram). In effect, FMTCFG indirectly controls the output bandwidth of the preview engine and H3A. Depending on the input sensor clock, you can set this field appropriately and balance the bandwidth requirements to SDRAM. Figure 50 demonstrates how this register can expand processing time per line for lower PCLK frequencies.

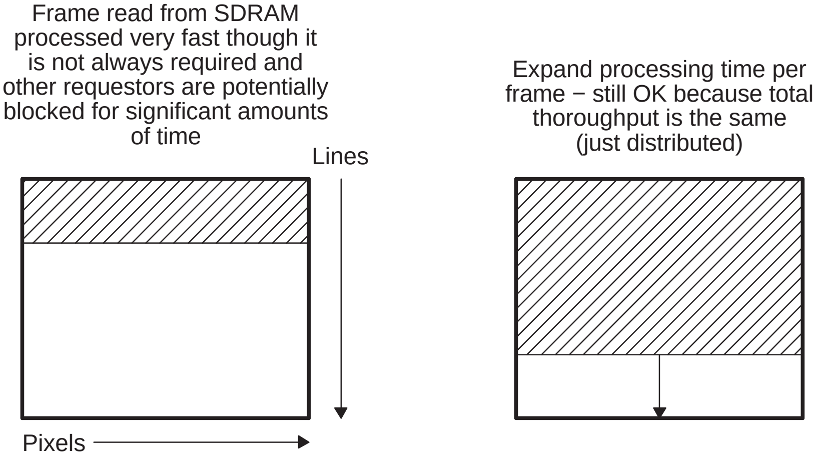
Figure 50. Video Port Interface Bandwidth Balancing


CCDC and Peak Bandwidth
Corresponds to PCLK Clock
(Example 27 MHz)
Lines
Pixels
Preview Engine, Resizer, and H3A
Always Corresponds to SYSCLK/3/2
(75 MHz) Irrespective to PCLK
Dense
Accesses
No
Accesses
Expanded Processing Time Per Line,
Leading to Less Dense Access
Pattern – Other Requestors
Have Better Latency

4.4.2.2 Input from SDRAM/DDRAM

When the input image is from SDRAM/DDRAM, the data is fetched from memory and processed at a steady-state rate of SYSCLK/3 MB/s. Depending on the image size and real-time deadline for each frame, this may be much faster than is necessary. Such activity can also starve out other processes in the system. The SDR_REQ_EXP register in the VPSS module can be programmed to control the rate at which a VPSS module (preview engine, resizer, and histogram) reads the input frame from memory. This indirectly controls the output bandwidth of the preview engine and resizer. Depending on the size of the images and the real-time deadlines, you can set this field appropriately and balance the bandwidth requirements to SDRAM. Figure 51 demonstrates how this register can expand processing time for lower real-time requirements.

Figure 51. SDRAM/DDRAM Read Bandwidth Balancing


Frame read from SDRAM processed very fast though it is not always required and other requestors are potentially blocked for significant amounts of time
Lines
Pixels
Expand processing time per frame − still OK because total thoroughput is the same (just distributed)

Internally, when a VPSS module is receiving input from DDR, the VPSS makes a read request to the DDR EMIF whenever there is available memory in its internal buffers. The number of minimum cycles (SYSCLK/3) in between read requests used to program the SDR_REQ_EXP register is determined based on frame size and real-time requirement using the following equation:

Number of cycles/request = (DMA cycles/frame)/(DMA read requests/frame)

In the equation, (DMA cycles/frame) is based on the real-time requirement. For example, if the real-time requirement is a frame rate of 1/30th second and the SYSCLK/3 is 153 MHZ, then this is calculated as:

DMA cycles/frame = SYSCLK/3× frame rate
= 153 MHZ × 1/30 = 5.1Mcycles

In the equation, (DMA read requests/frame) is based on the frame size and the alignment in memory (see Section 4.4.1). For a VGA (640 × 480) frame size and optimal alignment conditions:

DMA read requests/frame = transfers per line × number of lines
= ((640 pixels/line × 2 bytes/pixel)/256 bytes/transfer) × 480 lines
= 2400 requests/frame

In this example, the final equation is:

Number of cycles/request = 5.1Mcyles/2400 requests = 2125 cycles/request

The maximum values that can be written to the SDR_REQ_EXP register for the different read requesters is 1023. For the histogram and the preview engine, this should be sufficient for the typical sizes of RAW data frames. However, since the resizer can read a variety of video frame sizes, the field for the resizer is internally multiplied by 32. So for this example, the SDR_REQ_EXP.RESZ_EXP field is programmed to FLOOR(2125/32) = 66.

This example equation provides an estimate or a starting point for programming SDR_REQ_EXP. Depending on system loads and available bandwidth, this number may need to be reduced to compensate for a heavy loaded system.

4.5 Error Reporting

The Shared Buffer Logic or SBL described in the previous section includes user accessible error reporting that flags when a VPSS read or write deadline is missed. Memory overflow or underflow errors can occur in the read and write buffers of the VPFE. When an overflow from any of the write buffers occurs, a corresponding fail bit is set in the VPSS.PCR register as shown in Table 35. If the fault pixel correction is used, a read fail of fault pixel data from DDR2 is identified by the CCDC.FPC.FPERR bit. Finally, if darkframe subtract or shading compensation is used, a read fail of the data from DDR2 is identified by the PREV.PCR.DARK_FAIL bit.

Table 35. VPSS Error Indicators

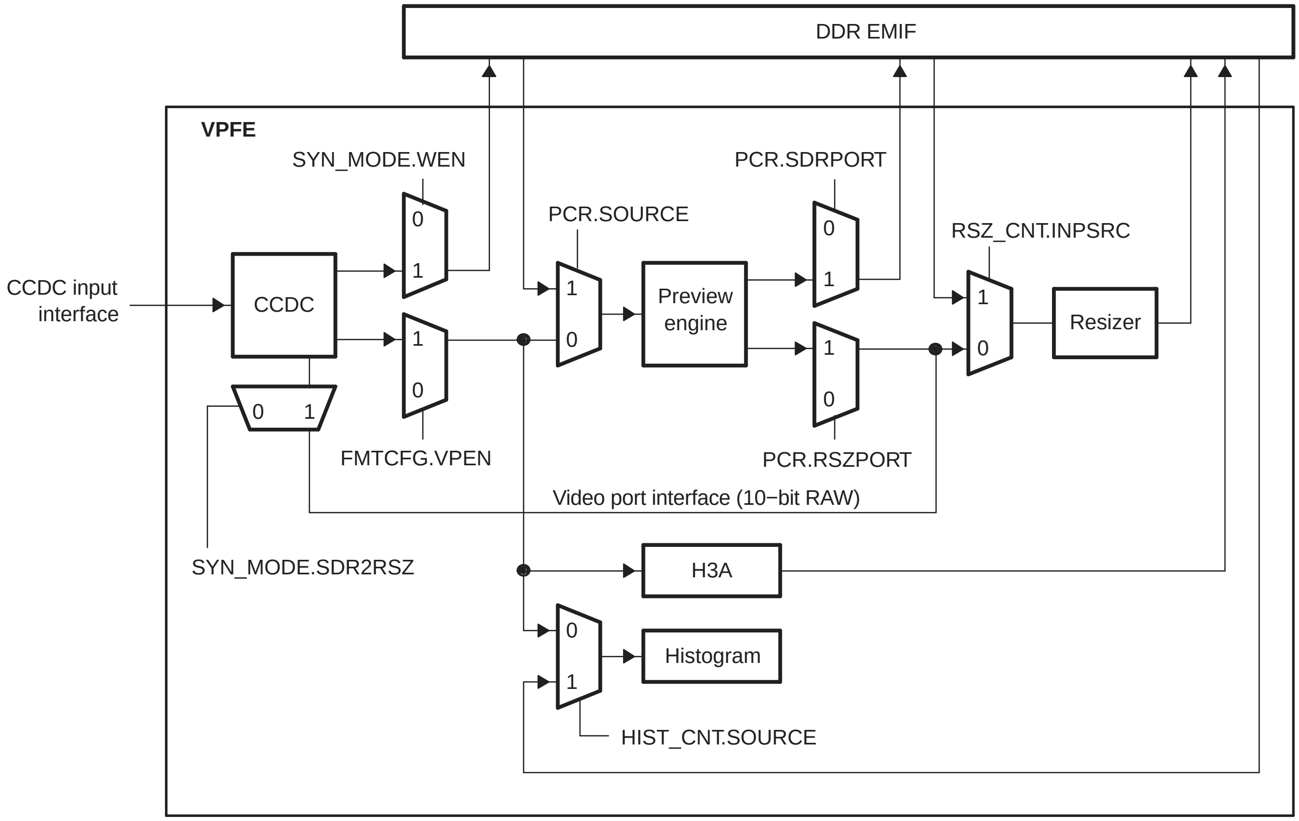
PCR Bit	Error Condition
CCDC_WBL_O	Write buffer memory overflow (CCDC)
PRV_WBL_O	Write buffer memory overflow (preview engine)
RSZ1_WBL_O	Write buffer memory overflow (Resizer line 1)
RSZ2_WBL_O	Write buffer memory overflow (Resizer line 2)
RSZ3_WBL_O	Write buffer memory overflow (Resizer line 3)
RSZ4_WBL_O	Write buffer memory overflow (Resizer line 4)
AF_WBL_O	Write buffer memory overflow (AF)
AEW_WBL_O	Write buffer memory overflow (AE/AWB)

5 Programming Model

5.1 Setup for Typical Configuration

A typical configuration of the VPFE for a digital camera application would include interfacing to a CCD controller, timing generator (TG), and analog front end (AFE) imager subsystem. In addition to programming these external devices, the VPFE would be programmed in Preview mode to capture data at a 30 Hz frame rate. The VPFE submodule would be configured to capture/read this data and format for display via the preview engine and, optionally, the resizer. Simultaneously, the statistics modules would collect data to be used by user-defined focusing algorithms, etc. Figure 52 shows the VPFE data flow. The interfaces between the modules are configured using the register field values indicated. The input to the CCD controller is either raw image data from an AFE or YCC data. The input to the preview engine, H3A, and histogram is only raw image data. The input to the resizer is only YCC data.

Figure 52. VPFE Data Flow Block Diagram



5.2 Resetting the Camera Subsystem

The entire VPSS subsystem (VPFE and VPBE) can be reset via the power and sleep controller (PSC).

5.3 Configuring the Clocks and the Control Signals

The input pixel data clock must be provided by the external imager device.

If the VPFE is to be configured in master mode, in which the HD/VD sync signals are output, then the proper frame definition must be configured in the CCD controller. Otherwise, in slave mode, the VPFE will sync to the externally provided signals.

5.4 Programming the CCD Controller

This section discusses issues related to the software control of the CCD controller. It lists the registers that are required to be programmed in different modes, how to enable and disable the CCD controller, how to check the status of the CCD controller, discusses the different register access types, and enumerates several programming constraints.

5.4.1 Hardware Setup/Initialization

This section discusses the configuration of the CCD controller that is required before image processing begins.

5.4.1.1 Reset Behavior

Upon hardware reset of the VPSS, all of the registers in the CCD controller are set to their default values.

5.4.1.2 Register Setup

Prior to enabling the CCD controller, the hardware must be properly configured via register writes. Table 36 identifies the register parameters that must be programmed before enabling the CCD controller.

Table 37 identifies additional configuration requirements depending on if the corresponding condition is met. The table can be read as:

```
if(Condition is TRUE) then
            Configuration Required parameters must be programmed
```

Table 36. CCD Controller Required Configuration Parameters

	Configuration Required	
Function	**Register**	**Field**
External pin signal configuration	SYN_MODE	VDHDOUT
	SYN_MODE	VDHDEN
	SYN_MODE	VDPOL
	SYN_MODE	HDPOL
	SYN_MODE	FLDMODE
	SYN_MODE	FLDOUT
	SYN_MODE	FLDPOL
	SYN_MODE	EXWEN
	SYN_MODE	DATAPOL
	CCDCFG	VDLC = 1
Input mode	REC656	R656ON
	SYN_MODE	INPMOD
Color pattern	COLPTN	
Black compensation	BLKCMP	
Data path configuration	FMTCFG	VPEN
	SYN_MODE	VP2SDR
	SYN_MODE	WEN
	SYN_MODE	SDR2RSZ

Table 37. CCD Controller Conditional Configuration Parameters

Function	Condition	Configuration Required
VD/HD set as outputs	SYN_MODE.VDHDOUT	VD_HD_WID PIX_LINES
Interlaced fields	SYN_MODE.FLDMODE	CCDFCG.FIDMD
External WEN	SYN_MODE.EXWEN	CCDCFG.WENLOG
REC656 input	REC656.R656ON	REC656.ECCFVH CCDCFG.BW656
YCC input	SYN_MODE.INPMOD != 0 && !REC656.R656ON	CCDCFG.YCINSWP CCDCFG.MSBINVI DCSUB
8-bit YCC input	SYN_MODE.INPMOD == 2 && !REC656.R656ON	CCDCFG.Y8POS
Raw input	SYN_MODE.INPMOD == 0 && !REC656.R656ON	SYN_MODE.DATSIZ CLAMP.CLAMPEN
Optical black clamp enabled	CLAMP.CLAMPEN && SYN_MODE.INPMOD == 0	CLAMP.OBGAIN CLAMP.OBST CLAMP.OBSLN CLAMP.OBSLEN
Optical black clamp disabled	!CLAMP.CLAMPEN && SYN_MODE.INPMOD == 0	DCSUB
Write to SDRAM or resizer	SYN_MODE.WEN ||SYN_MODE.SDR2RSZ	HORZ_INFO VERT_START VERT_LINES SYN_MODE.LPF CULLING ALAW.CCDTBL SYN_MODE.PACK8 CCDCFG.BSWD
Write to SDRAM	SYN_MODE.WEN	SDR_ADDR HSIZE_OFF SDOFST
A-law	ALAW.CCDTBL	ALAW.GWID
Interrupt usage	VDINT[1:0] interrupts are enabled	VDINT
Fault pixel correction	FPC.FPCEN	FPC.FPNUM FPC_ADDR Fault Pixel Table should be in SDRAM/DDRAM
Video port (Data Formatter) enabled	FMTCFG.VPEN	FMTCFG.VPIN FMT_HORZ FMT_VERT VP_OUT FMTCFG.VPIF_FRQ

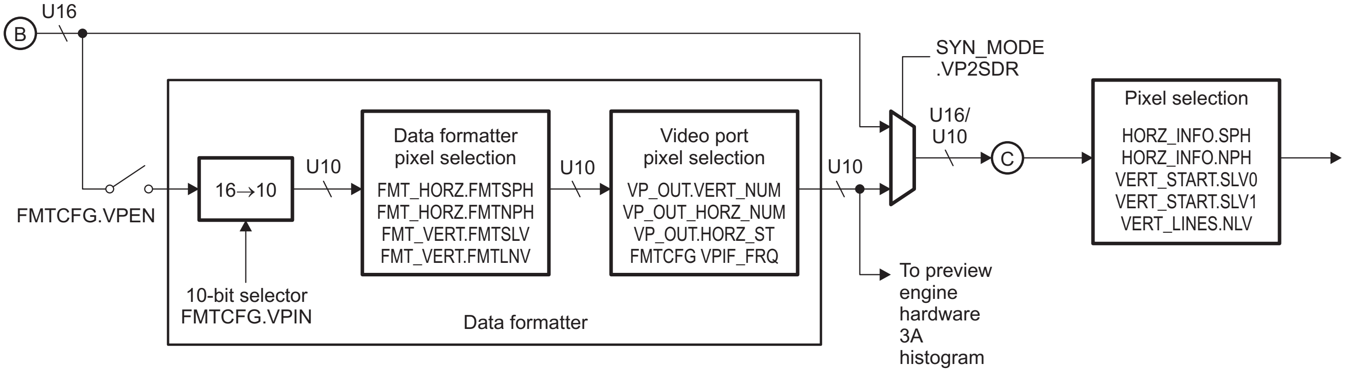
5.4.1.3 Pixel Selection (Framing) Register Dependancies

There are three locations in the data flow where the Valid Frame Data can be defined:

- Data Formatter Input Pixel Selection
- Video Port Output Pixel Selection
- Output Formatter Pixel Selection

Care must be taken to insure that the frame definitions are with respect to the output of the upstream frame definitions. When the video port is enabled, VP_OUT.VERT_NUM must be less than FMT_VERT.FMTLNV.

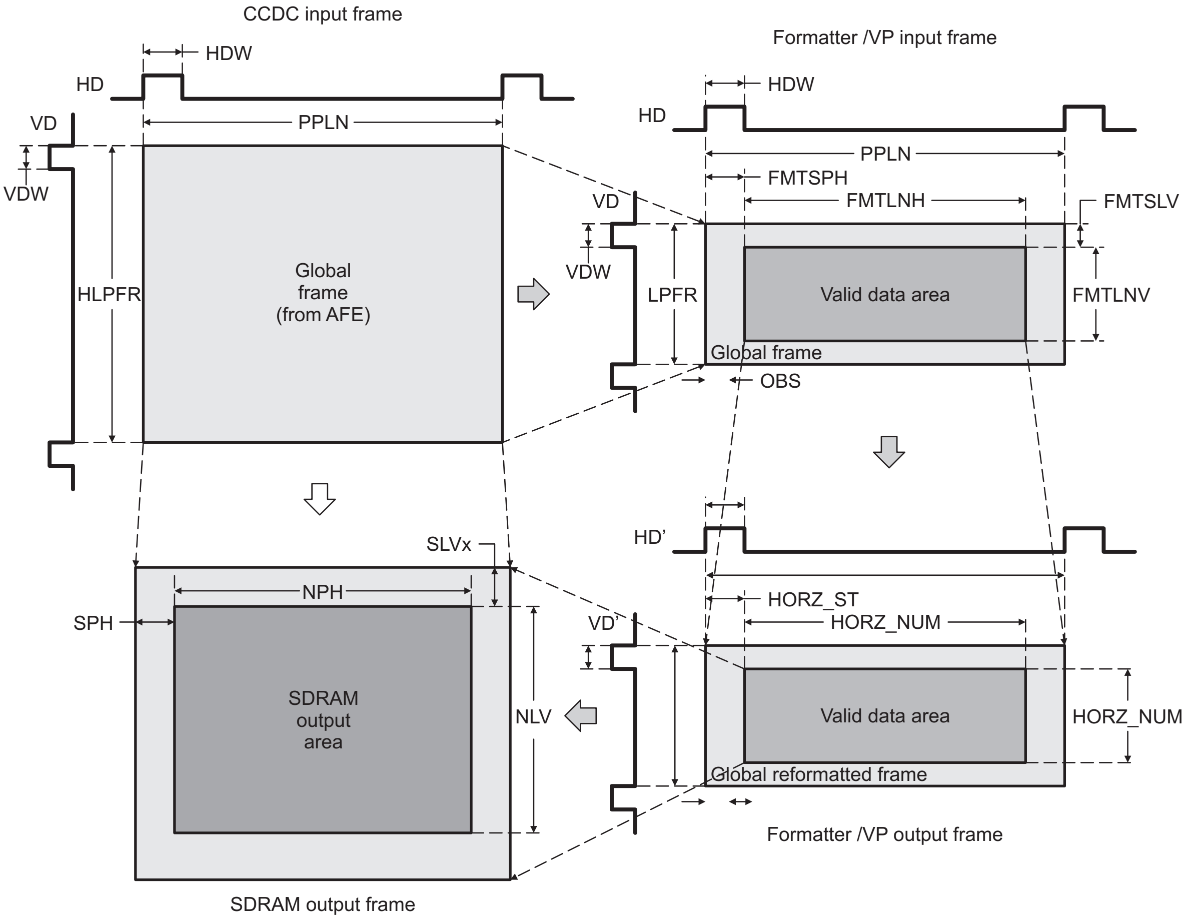
Figure 53. Pixel Selection Locations in Data Flow Diagram



There are two possible data paths through the CCDC which affect the programming of the pixel selection registers, depending on the value of the VP2SDR field.

- VP2SDR = 0: The input data bypasses the Data Formatter/Video Port. In this case, only the SDRAM Output Frame parameters apply. This data path is represented by the white arrow in Figure 54.
- VP2SDR = 1: The input data passes through Data Formatter/Video Port. In this case, both data formatter frame definitions apply to the Video Port Output, and all three frame definitions apply to the SDRAM output. This data path is represented by the green shaded arrow in Figure 54.

Figure 54. Dependencies Among Framing Settings in Data Flow



5.4.2 Enable/Disable Hardware

Note: When the CCD controller is disabled (PCR.ENABLE = 0), the CCD controller continues to generate interrupts. If your design is not expecting interrupts, you must disable those interrupts at the interrupt controller (INTC). See the *TMS320DM357 DMSoC ARM Subsystem Reference Guide* (SPRUG25) for information on the INTC.

Setting the PCR.ENABLE bit to 1 enables the CCD controller. This should be done after all of the required registers, mentioned in the previous section, are programmed.

The CCD controller always operates in continuous mode. After enabling the CCD controller, it continues to process sequential frames until the PCR.ENABLE bit is cleared by software. When this happens, the frame being processed continues until completion before the CCD controller is disabled.

When the CCD controller is in master mode (HD/VD signals set to outputs), then fetching and processing of the frame begins immediately upon setting the PCR.ENABLE bit.

When the CCD controller is in slave mode (HD/VD signals set to inputs), then processing of the frame is dependent upon the input timing of the external sensor/decoder. In order to assure that data from the external device is not missed, the CCD controller should be enabled prior to data transmission from the external device. In this way, the CCD controller waits for data from the external device.

5.4.3 Events and Status Checking

The CCD controller generates three different interrupts: VDINT0, VDINT1, and VDINT2. Note that the SYN_MODE.VDHDEN bit should be enabled to receive any of the CCD controller interrupts.

5.4.3.1 VDINT0 and VDINT1 Interrupts

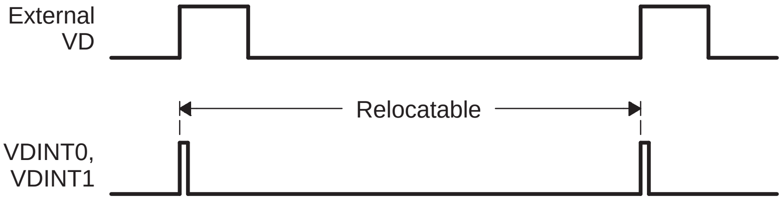
Note: When the CCD controller is disabled (PCR.ENABLE = 0), the CCD controller continues to generate interrupts. If your design is not expecting interrupts, you must disable those interrupts at the interrupt controller (INTC). See the *TMS320DM357 DMSoC ARM Subsystem Reference Guide* (SPRUG25) for information on the INTC.

As shown in Figure 55 and Figure 56, the VDINT0 and VDINT1 interrupts occur relative to the VD pulse. The trigger timing is selected by using the SYN_MODE.VDPOL setting. VDINT0 and VDINT1 occurs after receiving the number of horizontal lines (HD pulse signals) set in the VDINT.VDINT0 and VDINT.VDINT1 register fields, respectively.

Note: In the case of BT.656 input mode, there is a VD at the beginning of each field; therefore, there will be two interrupts for each frame (one for each field).

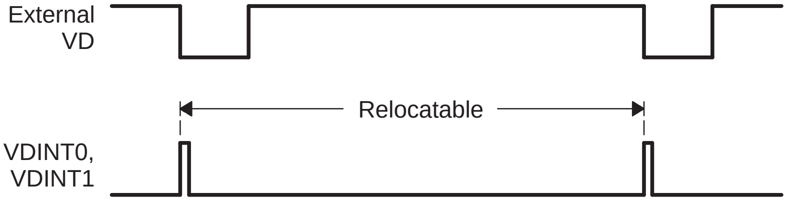
In the case of the SYN_MODE.VDPOL = 0 (Figure 55), the VDINT0 and VDINT1 HD counters begin counting HD pulses from the rising edge of the external VD.

Figure 55. VDINT0/VDINT1 Interrupt Behavior when VDPOL = 0



In the case of the SYN_MODE.VDPOL = 1 (Figure 56), the VDINT0 and VDINT1 HD counters begin counting HD pulses from the falling edge of the external VD.

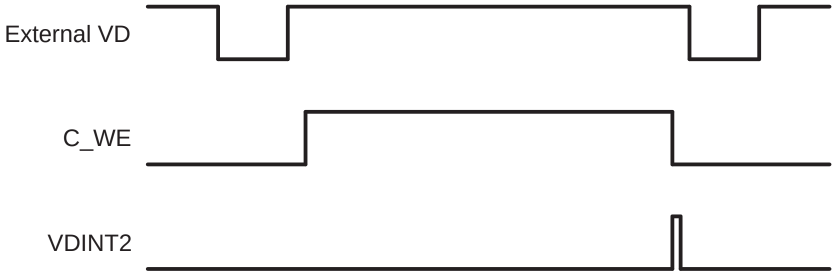
Figure 56. VDINT0/VDINT1 Interrupt Behavior when VDPOL = 1



5.4.3.2 VDINT2 Interrupt

In addition to the VDINT0 and VDINT1 interrupts, the CCD controller also has a VDINT2 interrupt (Figure 57). The VDINT2 interrupt always occurs at the falling edge of the C_WE signal (via external pin). There are no registers in the CCD controller module to configure this interrupt.

Figure 57. VDINT2 Interrupt Behavior



5.4.3.3 Status Checking

The PCR.BUSY status bit is set when the start of frame occurs (if the PCR.ENABLE bit is 1 at that time). It is automatically reset to 0 at the end of a frame. The PCR.BUSY status bit may be polled to determine the end of frame status.

The FPC.FPERR status bit is set when the fault pixel data fetched from memory arrives late. This bit can be reset by writing a 1 to the bit.

5.4.4 Register Accessibility During Frame Processing

There are three types of register access in the CCD controller:

- **Shadowed registers.** In the CCD controller, there are three different register fields that are shadowed in different ways. Shadowed registers are those that can be read and written at any time, but the written values only take effect (are latched) at certain times based on some event. Note that reads will still return the most recent write even though the settings are not used until the specific event occurs. The following register/fields are shadowed:
  - **PCR.ENABLE:** Written values take effect only at the start of a frame event (rising edge of VD, if SYN_MODE.VDPOL is positive; falling edge of VD, if SYN_MODE.VDPOL is negative).
  - **SDR_ADDR:** When CCDCFG.VDLC is cleared to 0, written values take effect only at the start of a frame event (rising edge of VD, if SYN_MODE.VDPOL is positive; falling edge of VD, if SYN_MODE.VDPOL is negative). When CCDCFG.VDLC is set to 1, written values take effect only at the start of the frame being output to SDRAM (when the input has reached the HORZ_INFO.SPH pixel of the VERT_START.SLV*n* line of each field).
  - **CCDCFG.YCINSWP:** Written values take effect only during the active period of VD (when VD is high, if SYN_MODE.VDPOL is positive; when VD is low, if SYN_MODE.VDPOL is negative).
- **Busy-writeable registers.** These registers/fields can be read or written even if the module is busy. Changes to the underlying settings takes place instantaneously. All register fields not listed as shadowed or optionally shadowed/busy-writeable are busy-writeable registers.

- **Optionally shadowed/busy-writeable registers.**These register/fields can be all set as shadow registers or optionally set as busy-writeable registers. When CCDCFG.VDLC is cleared to 0, these registers are shadowed; when CCDCFG.VDLC is set to 1, these registers are busy-writeable.

Note: CCDCFG.VDLC must be set to 1 by software if the CCD controller is to be used; therefore, these registers will be busy-writeable. If CCDCFG.VDLC remains cleared to 0 (default), indeterminate results may occur for any register access in the CCD controller, not just the following registers.

SYN_MODE.SDR2RSZ
SYN_MODE.VP2SDR
SYN_MODE.VDHDEN
SYN_MODE.WEN
SYN_MODE.LPF
HD_VD_WID
PIX_LINES
HORZ_INFO
VERT_START
VERT_LINES
CULLING
HSIZE_OFF
SDOFST
CLAMP.CLAMPEN
FMTCFG.FMTEN

5.4.5 Inter-frame Operations

In between frames, it may be necessary to enable/disable functions or modify the memory pointers. Since the PCR register and memory pointer registers are shadowed, these modifications take place any time before the end of the frame, and the data gets latched in the next frame. The host controller can perform these changes upon receiving an interrupt.

5.4.6 Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the CCD controller. It can be used as a quick checklist. More detailed register setting constraints are found in the individual register descriptions.

- PCLK can not be higher than 90 MHZ (generically, 1 ns more than 2× the period of the VPFE clock, or CLKDIV3. See the device data sheet for supported maximum pixel rate information).
- If SDRAM output port is enabled, the memory output line offset and address should be on 32-byte boundaries.
- External WEN can not be used when VP2SDR path is enabled.
- The horizontal number for the video port (VP_OUT.HORZ_NUM) must be $\leq 1376 \times 4$.
- If the video port is enabled, the vertical number for the video port (VP_OUT.VERT_NUM) must be < FMT_VERT.FMTLNV.
- In YCC input mode:
  - COLPTN should be cleared to 0.
  - BLKCMP should be cleared to 0.
  - The video port should be disabled.
  - The formatter should be disabled.
  - The VP2SDR should be disabled.
  - The low-pass filter should be disabled.
  - The ALAW should be disabled.
- In raw input mode, the resizer output path should not be enabled.
- If Fault Pixel Correction is enabled, the FPC address should be on a 64-byte boundary.

5.5 Programming the Preview Engine

This section discusses issues related to the software control of the preview engine. It lists which registers are required to be programmed in different modes, how to enable and disable the preview engine, how to check the status of the preview engine, discusses the different register access types, and enumerates several programming constraints.

5.5.1 Hardware Setup/Initialization

This section discusses the configuration of the preview engine required before image processing can begin.

5.5.1.1 Reset Behavior

Upon hardware reset of the VPSS, all of the registers in the preview engine are reset to their reset values. However, since the preview engine programmable tables (gamma, noise filter, luminance enhancer, and CFA coefficients) are stored in internal RAM, their contents do not have reset values. If the reset is a chip-level power-on reset (reset after power is applied), then the contents of these tables are unknown. If the reset is a VPSS module reset (when power remains active) then the contents of these tables remains the same as before the reset.

5.5.1.2 Register Setup

Prior to enabling the preview engine, the hardware must be properly configured via register writes. Table 38 identifies the register parameters that must be programmed before enabling the preview engine.

Table 38. Preview Engine Required Configuration Parameters

Function	Configuration Required
Function Enable/Disable	PCR.INVALAW
	PCR.DRKFCAP
	PCR.DRKFEN
	PCR.SHADE_COMP
	PCR.HMEDEN
	PCR.NFEN
	PCR.CFAEN
	PCR.GAMMA_BYPASS
	PCR.YNENHEN
	PCR.SUPEN
I/O Ports	PCR.SOURCE
	PCR.SDRPORT
	PCR.RSZPORT
Input Size	HORZ_INFO
	VERT_INFO
Averager	AVE
White Balance	PCR.CFAFMT
	WB_DGAIN
	WBGAIN
	WBSEL
Black Adjustment	BLKADJOFF
RGB to RGB Blending	RGB_MAT[5:1]
	RGB_OFF[2:1]
RGB to YCbCr Conversion	CSC[3:0]
Contrast and Brightness	CNT_BRT

Table 38. Preview Engine Required Configuration Parameters (continued)

Function	Configuration Required
YCC Output Format	SETUP_YC PCR.YCPOS

The PCR register contains several control bits which enable or disable optional functions and module IO ports. If an optional function or port is enabled, then there may be more registers or configuration information required for the preview engine to operate properly. Table 39 identifies additional configuration requirements depending on if the corresponding condition is met. Table 39 can be read as:

```
if(Condition is TRUE)
       then Configuration Required parameters must be programmed
```

Table 39. Preview Engine Conditional Configuration Parameters

Function	Condition	Configuration Required
Read from CCDC	!PCR.SOURCE	PCR.ONESHOT
Read from SDRAM	PCR.SOURCE	PCR.WIDTH RSDR_ADDR RADR_OFFSET
Dark Frame Subtract	PCR.DRKFEN	DSDR_ADDR DRKF_OFFSET Dark frame should be in SDRAM/DDRAM
Shading Correction	PCR.SHADE_COMP and PCR.DRKFEN	PCR.SHADE_SFT DSDR_ADDR DRKF_OFFSET Dark frame should be in SDRAM/DDRAM
Noise Filter	PCR.NFEN	NF Setup Noise Filter Table
CFA Interpolation	PCR.CFAEN	CFA Setup CFA Coefficient Table
Gamma Correction	!PCR.GAMMA_BYPASS	Setup Gamma Correction Tables
Luminance Enhancement	PCR.YNENHEN	Setup Luminance Enhancement Table
Chrominance Suppression	PCR.SUPEN	CSUP
Write to SDRAM	PCR.SDRPORT	WSDR_ADDR WADD_OFFSET

5.5.1.3 *Table Setup*

The three gamma memories, noise filter threshold memory, luminance enhancer memory, and the CFA coefficient memory must be filled in prior to the operation of the preview engine if their respective functions are enabled. There are two registers that allow for the memory contents to be read and written. The address register (SET_TBL_ADDRESS) is used to select the specific table entry. The data register (SET_TBL_DATA) contains the data to be written to the specified location. While the data register is 20-bits wide, only the 8 LSB data is used for the gamma, noise filter, and CFA filter tap memories. Table 40 identifies the addressing range for each memory provided in the preview engine.

Table 40. Preview Engine Memory Address Ranges

Start	End	Bit Width	Memory
000h	3FFh	8	Red Gamma Table
400h	7FFh	8	Green Gamma Table
800h	BFFh	8	Blue Gamma Table
C00h	CFFh	8	NF Threshold Table
1000h	107Fh	20	Non-linear Enhancement Table
1400h	16F7h	8	CFA Filter Coefficient Table

The preview engine supports linear increments on reads and writes automatically. The following examples show how the programmer can read/write the memory. If data is read/written, the address pointer is automatically incremented. For random/non-contiguous reads/writes, the SET_TBL_ADDRESS register needs to be modified. The address is not auto-incremented when the preview engine is busy and the user tries to read/write the tables.

Example 2. Read/write all the entries of the CFA table (using linear increment)

```
WRITE (SET_TBL_ADDRESS, 0x1400);
READ (SET_TBL_DATA, 0xvalue1);
WRITE (SET_TBL_DATA, 0xvalue2);
READ (SET_TBL_DATA, 0xvalue3);
READ (SET_TBL_DATA, 0xvalue163F);
```

Example 3. Read/write selective entries of the tables (have to program the address separately for each read/write)

```
WRITE (SET_TBL_ADDRESS, 11);
READ (SET_TBL_DATA, value11);
WRITE (SET_TBL_ADDRESS, 564);
WRITE (SET_TBL_DATA, value564);
```

5.5.2 Enable/Disable Hardware

Setting the PCR.ENABLE bit enables the preview engine. This should be done after all of the required registers and tables mentioned in the previous section are programmed.

When the input source is the SDRAM/DDRAM, the preview engine always operates in the one shot mode. In other words, after enabling the preview engine, the PCR.ENABLE bit is automatically turned off (set to 0) and only a single frame is processed from memory. In this mode, fetching and processing of the frame begins immediately upon setting the PCR.ENABLE bit.

When the input source is the CCD/CMOS controller, the preview engine can be configured to operate in either one-shot mode, or continuous mode (PCR.ONESHOT). Processing of the frame is dependent upon the timing of the CCDC. In order to guarantee that data from the CCDC is not missed, the preview engine should be enabled prior to the CCDC. In this way, the preview engine will wait for data from the CCDC.

Note: In one-shot mode, upon setting the PCR.ENABLE bit, the processing of the frame begins and the PCR.ENABLE, PCR.ONESHOT, and PCR.SOURCE bits are reset to their reset values.

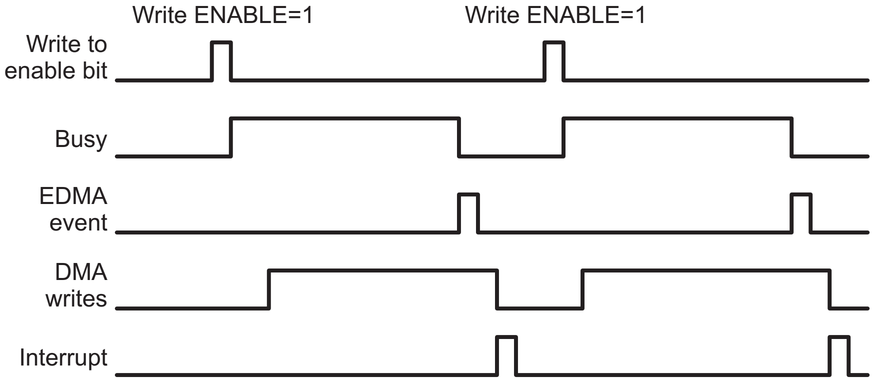
When the preview engine is in continuous mode, it can be disabled by clearing the PCR.ENABLE bit during the processing of the last frame. The disable will be latched in at the end of the frame it was written in.

5.5.3 Events and Status Checking

The preview engine will generate both an interrupt and an EDMA event for each frame as shown in Figure 58.

The PCR.BUSY status bit is set when the start of frame occurs (if the PCR.ENABLE bit is 1 at that time). The PCR.BUSY bit is automatically cleared to 0 and the EDMA event is triggered when it is safe to modify the busy-lock registers (see next sectionSection 5.5.4) to setup the next frame. When this happens, the frame may not yet fully have reached its destination. Instead, the interrupt signal indicates when all the data has reached its destination. The PCR.BUSY status bit may be polled to determine when the processor can update the registers, or the EDMA trigger can be used to trigger an EDMA transfer to update the registers. Configuration registers and filter coefficients should be programmed in-between busy periods, before writing ENABLE=1. Figure 58 below indicates this interaction in SDRAM input-mode.

Figure 58. Firmware Interaction for SDRAM-Input Preview



The PCR.DRK_FAIL status bit is set when the darkframe data fetched from memory arrives late. This bit can be reset by writing a 1 to the bit.

5.5.4 Register Accessibility During Frame Processing

There are three types of register access in the preview engine.

- Shadow registers
  - These registers/fields can be read and written (if the field is writeable) at any time. However, the written values take effect only at the start of a frame. Note that reads will still return the most recent write even though the settings are not used until the next start of frame.
  - The PCR, RSDR_ADDR, RADR_OFFSET, DSDR_ADDR, DRKF_OFFSET, WSDR_ADDR, and WADD_OFFSET registers are the only shadowed registers in the preview engine.
- Busy-writeable registers
  - These registers/fields can be read or written even if the module is busy. Changes to the underlying settings takes place instantaneously.
  - The WB_DGAIN and WBGAIN registers are busy-writeable.
- Busy-lock registers
  - All registers EXCEPT the shadow and busy-writeable registers belong to this category. Busy-lock registers cannot be written when the module is busy. Writes will be allowed to occur, but no change will occur in the registers (blocked writes from hardware perspective, but allowed write from the software perspective). Once the PCR.BUSY bit is reset to 0, the busy-lock registers can be written.
  - The SET_TBL_DATA register cannot be read when the preview engine is busy since this register is mapped to memories internally. Such reads will return indeterminate data. Byte enables are not implemented for reading the preview engine memories.

The ideal procedure for changing the preview engine registers is:

```
IF (PCR.BUSY == 0) OR IF (EOF interrupt occurs)
               DISABLE PREVIEW ENGINE
               CHANGE REGISTERS
               ENABLE PREVIEW ENGINE
```

5.5.5 Inter-frame Operations

In between frames, it may be necessary to enable/disable functions or modify the memory pointers. Since the PCR register and memory pointer registers are shadowed, these modifications can take place any time before the end of the frame, and the data will get latched in for the next frame. The host controller can perform these changes upon receiving an interrupt, or an EDMA transfer can be programmed to make these changes upon receiving an event.

5.5.6 Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the preview engine. It can be used as a quick checklist. More detailed register setting constraints can be found in the individual register descriptions.

- The first input pixel should be the RED pixel in order for the register names to be aligned with the appropriate colors.
- If SDRAM output port is enabled:
  - The memory output line offset and address should be on 32 byte boundaries.
- Output Width should be less than or equal to 1280.
- Output Width should be even.
- Input Width should be a multiple of the AVE parameters.
- Input height should be smaller than CCD/CMOS controller output height.
- Input width should be at least 4 pixels smaller than CCD/CMOS controller output width.
  - SPH at least 2, EPH at least 2 pixels before last pixel from CCDC.

5.6 Programming the Resizer

This section discusses issues related to the software control of the resizer. It lists the registers that are required to be programmed in different modes, how to enable and disable the resizer, how to check the status of the resizer, discusses the different register access types, and enumerates several programming constraints.

5.6.1 Hardware Setup/Initialization

This section discusses the configuration of the resizer that is required before image processing begins.

5.6.1.1 Reset Behavior

Upon hardware reset of the VPSS, all of the registers in the resizer are set to their default values.

5.6.1.2 Register Setup

Prior to enabling the resizer, the hardware must be properly configured via register writes. Table 41 identifies the register parameters that must be programmed before enabling the Resizer.

Table 41. Resizer Required Configuration Parameters

Function	Configuration Required
Resizer control parameters	RSZ_CNT
I/O sizes	OUT_SIZE IN_START IN_SIZE
Memory addresses	SDR_INADD SDR_INOFF SDR_OUTADD SDR_OUTOFF
Filter coefficients	HFILT[31:0] VFILT[31:0]
Edge enhancement	YENH.ALG

The edge enhancement function is optional. If it is disabled, then the rest of the YENH register does not need to be programmed. However, if it is enabled, then the edge enhancement parameters in Table 42 need to be programmed so that the edge enhancement function operates properly.

Table 42. Resizer Conditional Configuration Parameters

Function	Condition	Configuration Required
Edge enhancement	YENH.ALG != 0	YENH.GAIN YENH.SLOP YENH.CORE

5.6.2 Enable/Disable Hardware

Setting the PCR.ENABLE bit enables the resizer. This should be done after all of the required registers mentioned in the previous section are programmed.

The resizer always operates in the one-shot mode. After enabling the resizer, the PCR.ENABLE bit is automatically turned off (cleared to 0) and only a single frame is processed from memory.

When the input source is the SDRAM/DDRAM, fetching and processing of the frame begins immediately upon setting the PCR.ENABLE bit.

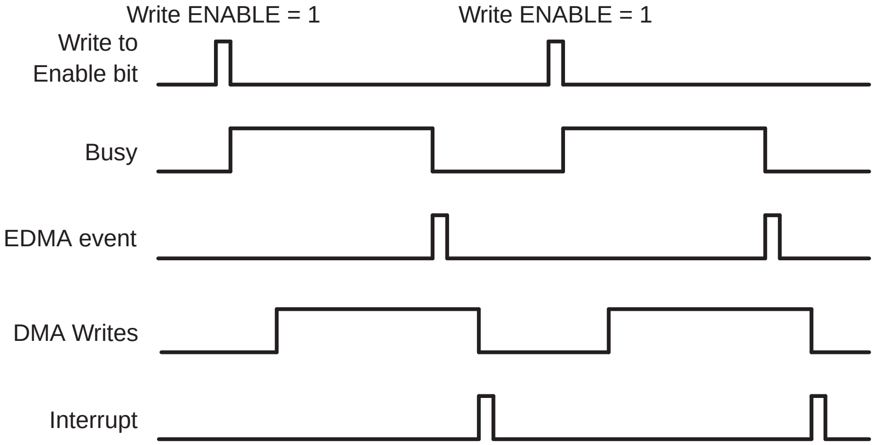
When the input source is the CCD controller or preview engine, processing of the frame is dependent upon the timing of the CCD controller. In order to assure that data from the CCD controller or preview engine is not missed, the resizer should be enabled prior to these upstream modules. In this way, the resizer will wait for data from the CCD controller or the preview engine.

5.6.3 Events and Status Checking

The resizer generates an interrupt and an EDMA event at the end of each frame.

The PCR.BUSY status bit is set when the start of frame occurs (if the PCR.ENABLE bit is 1 at that time). The PCR.BUSY bit is automatically cleared to 0 and the EDMA event is triggered when it is safe to modify the busy-lock registers (see Section 5.6.4) to setup the next frame. Note that when this happens, the frame may not yet be fully in SDRAM. Instead, the interrupt signal indicates when all the data is actually in SDRAM. The PCR.BUSY status bit may be polled to determine when the processor can update the registers, or the EDMA trigger can be used to trigger an EDMA transfer to update the registers. Figure 59 shows the firmware/hardware interaction. Configuration registers and filter coefficients should be programmed in-between busy periods, before writing ENABLE = 1.

Figure 59. Firmware Interaction for SDRAM-Input Resizing



5.6.4 Register Accessibility During Frame Processing

There are two types of register access in the resizer:

- **Shadow registers.** These registers/fields are read and written (if the field is writeable) at any time. However, the written values take effect only at the start of a frame. Note that reads still return the most recent write even though the settings are not used until the next start of frame. The following register/fields are shadowed in the resizer:

PCR
SDR_INADD
SDR_INOFF
SDR_OUTADD
SDR_OUTOFF

- **Busy-lock registers.** All registers except the PCR, SDR_INADD, SDR_INOFF, SDR_OUTADD, and SDR_OUTOFF registers are busy-lock registers. Busy-lock registers cannot be written when the module is busy. Writes are allowed to occur, but no change occurs in the registers (blocked writes from hardware perspective, but allowed write from the software perspective). Once the PCR.BUSY bit is reset to 0, the busy-lock registers can be written.

The ideal procedure for changing the resizer registers is:

```
IF (PCR.BUSY == 0) OR IF (EOF interrupt occurs)
          DISABLE RESIZER
          CHANGE REGISTERS
          ENABLE RESIZER
```

5.6.5 Inter-Frame Operations

In between frames, it may be necessary to modify the memory pointers before processing the next frame. Since the PCR.ENABLE bit and memory pointer registers are shadowed, these modifications take place any time before the end of the frame, and the data gets latched in the next frame. The host controller can perform these changes upon receiving an interrupt, or an EDMA transfer can be programmed to make these changes upon receiving an event.

Note that firmware is responsible to compute/upload the filter coefficients. If the polyphase resampling methodology is used, a different set is required when changing between 4-tap and 7- tap modes, and with different down-sampling factors; all up-sampling factors can share the same set of coefficients. If back-to-back resizes are required, which change busy-lock registers (such as the coefficients, resizing ratios, input and output sizes), then any changes to these registers must wait until after the first resize is finished. The following section describes some scenarios where this is required.

5.6.5.1 Multiple Passes for Larger Resizing Operations

The resizer supports multiple passes of processing for larger resizing operations. The several meanings of larger are:

- *Wider output than 1280 pixels.* This only works in SDRAM input mode. The input can be partitioned into multiple resizer blocks and each block is separately resized, and stitched together. Having input/output SDRAM line offsets, input starting pixel and starting phase are essential to make this work. The basic idea is to begin subsequent slices at exactly where previous images left off. The starting phase and pixel registers are programmed to this exact location. This location is calculated using the algorithm details in Section 4.3.4.4.
- *Larger than 4× upsampling.* Resizing can be applied in multiple passes. For example, 10× upsampling is realized by first a 4× upsampling, then a 2.5× upsampling. The first pass is performed on-the-fly with preview. The second pass is only performed with input from SDRAM, and for 10× digital zoom, there is time outside the active picture region to perform the second pass. See Section 5.11.1.1.1 for more details.
- *Larger than 4:1 down-sampling.* Although it is rare to generate a very small image from a big image, it is supported by the hardware. For example, 10× down-sampling is realized first with 4× down-sampling on-the-fly with preview, then 2.5× down-sampling in SDRAM-input path. There may not be much time outside the active data region for the second pass, but since it is already reduced to 1/16 of original size, there is no need for a lot of time. Typically, CCD sensor or video input has 10% to 20% of vertical blanking that can be used.

For all of these scenarios, the second pass is configured and initiated from an interrupt service routine triggered by the resizer end-of-frame interrupt.

5.6.5.2 Processing Time Calculation

The following equation can be used to determine the processing time of the resizer when the input is from SDRAM:

$$Time = \frac{bytes\_per\_pixel \times W \times input\_height}{DSPCLK/3}$$

Where *bytes_per_pixel* = 1, when RSZ_CNT.INPTYP = 1 (color separate)
2, when RSZ_CNT.INPTYP = 0 (YUV422)

If the input is YUV422, and horizontal down-sampling is performed:

```
if((RSZ_CNT.INPTYP == 0) && (RSZ_CNT.HRSZ > 256))
W = average(input width, output width*);
else
W = max(input width, output width*);
```

* Including extra 4 pixels, if edge enhancement is enabled.

This time is the baseline steady-state calculation of the hardware and does not include the time it takes for the hardware to fetch the first input and fill the pipeline. It also does not include the time spent from the last output from the resizer to get back to the SDRAM memory when the resizer interrupt occurs. However, these beginning and ending times are relatively negligible. Depending on real-time constraints, this processing time may be much faster than is required. See Section 4.4.2.2 for details on delaying data fetches from SDRAM, in order to free more bandwidth for use by other system peripherals.

5.6.6 Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the resizer. It can be used as a quick checklist. More detailed register setting constraints are found in the individual register descriptions.

- Vertical and horizontal resize ratio values (RSZ_CNT.VRSZ and RSZ_CNT.HRSZ) should be within the range: 64 to 1024.
- Output Width:
  - Should be within the maximum limit:
    - width < 1280 (if vertical resize value is in the range: RSZ_CNT.VRSZ = 64 to 512)
    - width < 640 (if vertical resize value is in the range: RSZ_CNT.VRSZ = 513 to 1024)
  - Should be even
  - Should be a multiple of 16 bytes (for vertical upsizing)
- When input is from the preview engine/CCD controller:
  - The input height and width should be ≤ the output of the preview engine/CCD controller.
  - The input address and offset should be 0.
  - The input can not be color separate data.
- If source is SDRAM:
  - The vertical start pixel should be 0.
  - The horizontal start pixel should be within the range: [0 to 15] for color interleaved, [0 to 31] for color separate data.
  - The memory output line offset and address should be on 32-byte boundaries.
  - Input height and width must adhere to the equations listed in Table 26.
- Input height and width must adhere to the equations listed in Table 26.

5.7 Programming the H3A

This section discusses issues related to the software control of the H3A module. It lists which registers are required to be programmed in different modes, how to enable and disable the H3A, how to check the status of the H3A, discusses the different register access types, and enumerates several programming constraints.

5.7.1 Hardware Setup/Initialization

This section discusses the configuration of the H3A required before image processing can begin.

5.7.1.1 Reset Behavior

Upon hardware reset of the VPSS, all of the registers in the H3A are reset to their reset values.

5.7.1.2 Register Setup

For register configuration purposes, the AF Engine and the AEW Engine of the H3A can independently be configured. There are separate enable bits for each engine, so this section will be divided into the AF Engine and the AEW Engine.

5.7.1.2.1 AF Engine

Prior to enabling the AF Engine, the hardware must be properly configured via register writes. Table 43 identifies the register parameters that must be programmed before enabling the AF Engine of the H3A.

Table 43. AF Engine Required Configuration Parameters

Function	Configuration Required
AF Optional Preprocessing	PCR.AF_MED_EN PCR.AF_ALAW_EN
AF Mode Configuration	PCR.RGBPOS PCR.FVMODE
Paxel Start and Size Information	AFPAX1 AFPAX2 AFPAXSTART AFIIRSH
Memory Address	AFBUFST
Filter Coefficients	AFCOEF0[10:0] AFCOEF1[10:0]

The horizontal median filter function is optional. If it is disabled, then the PCR.MED_TH does not need to be programmed. However, if it is enabled, then the PCR.MED_TH parameter in Table 44 needs to be programmed so that the horizontal median filter function operates properly.

Table 44. AF Engine Conditional Configuration Parameters

Function	Condition	Configuration Required
Horz. Median Filter	PCR.AF_MED_EN	PCR.MED_TH

The following references offer guidelines on how to program the filter coefficients and make use of the H3A output.

- M. Gamadia, V. Peddigari, N. Kehtarnavaz, S-Y. Lee, G. Cook, “Real-time Implementation of Auto Focus on the TI DSC Processor”, *Proceedings of SPIE Real-Time Imaging Conference*, Jan 2004

- N. Kehtarnavaz, H-J. Oh, "Development and real-time implementation of a rule-based auto-focus algorithm", Journal of Real-Time Imaging, 9, 197-203, 2003

5.7.1.2.2 AEW Engine

Prior to enabling the AEW Engine, the hardware must be properly configured via register writes. Table 45 identifies the register parameters that must be programmed before enabling the AEW Engine of the H3A.

Table 45. AEW Engine Required Configuration Parameters

Function	Configuration Required
AEW Optional Preprocessing	PCR.AEW_ALAW_EN
Saturation Limit	PCR.AVE2LMT
Window Start and Size Information	AEWWIN1 AEWINSTART AEWINBLK AEWSUBWIN
Memory Address	AEWBUFST

5.7.2 Enable/Disable Hardware

Setting the PCR.AF_EN bit enables the AF Engine, and the PCR.AEW_EN bit enables the AEW Engine. This should be done after all of the required registers mentioned in the previous section are programmed.

The H3A always operates in continuous mode. Since the input to the H3A module is the video port interface of the CCD/CMOS controller, processing of the frame is dependent upon the timing of the CCD/CMOS controller. In order to guarantee that data from the CCD/CMOS controller is not missed, the H3A should be enabled prior to the CCD/CMOS controller. In this way, the H3A will wait for data from the CCD/CMOS controller.

The AF Engine or the AEW Engine can be disabled by clearing the PCR.AF_EN or PCR.AEW_EN bit, respectively, during the processing of the last frame. The disable will be latched in at the end of the frame it was written in.

5.7.3 Events and Status Checking

Both the AF Engine and the AEW Engine generates an interrupt and an EDMA event for each frame. The PCR.BUSYAF and/or PCR.BUSYAEAWB status bits are set when the start of frame occurs (if the corresponding ENABLE bit is 1 at that time). The PCR.BUSYAF and/or PCR.BUSYAEAWB bits are automatically cleared to 0 and the EDMA event is triggered when it is safe to modify the busy-lock registers (see Section 5.7.4) to setup the next frame. When this happens, the frame may not yet fully have reached its destination. Instead, the interrupt signal indicates when all the data has reached its destination.

The AF and AEW interrupts (and events) are internally tied together so that only one H3A interrupt signal (and H3A EDMA event) is seen by the interrupt controller (and EDMA). If the AF Engine and AEW Engine are not processing the same frame concurrently, then this shouldn't be an issue. However, if both the AF and AEW Engines are running concurrently, then one of two outcomes may occur:

- The H3A interrupt may seem to only trigger once for each frame.

  This can happen when the processing for both the AF Engine and the AEW Engine is finished at/near the same time. The interrupt service routine does not have enough time to clear the interrupt flag for the first interrupt before the second interrupt occurs.
- The H3A interrupt may trigger twice for each frame.

  This can happen when either the AF Engine or the AEW Engine finishes processing the frame much earlier than the other one. In this case, the interrupt service routine does have enough time to clear the interrupt flag for the first interrupt by the time the second interrupt occurs.

The outcome is dependent on the difference in location of the last paxel/window in the frame (determines when processing is finished), the frequency of the relative clocks in the system, the occurrence and triggering of other interrupts in the system, and the latencies of the context switching and interrupt service routine execution.

The PCR.BUSYAF and/or PCR.BUSYAEAWB status bits are set when the start of frame occurs (if the PCR.AF_EN and/or PCR.AEW_EN bits are 1 at that time). They are automatically reset to 0 at the end of processing a frame. The PCR.BUSYAF and/or PCR.BUSYAEAWB status bits may be polled to determine the end of frame status.

5.7.4 Register Accessibility During Frame Processing

There are two types of register access in the H3A module.

- Shadow registers
  - These registers/fields can be read and written (if the field is writeable) at any time. However, the written values take effect only at the start of a frame. Reads will still return the most recent write even though the settings are not used until the next start of frame.
  - The PCR, AFBUFST, and AEWBUFST registers are the only shadowed registers in the H3A module.
- Busy-lock registers
  - All registers EXCEPT the PCR, AFBUFST, and AEWBUFST registers belong to this category.
  - Busy-lock registers cannot be written when the module is busy. Writes will be allowed to occur, but no change will occur in the registers (blocked writes from hardware perspective, but allowed write from the software perspective). Once the busy bit in the PCR register is reset to 0, the busy-lock registers can be written.

The ideal procedure for changing the H3A registers is:

```
IF (busy == 0) OR IF (EOF interrupt occurs)
          DISABLE AF or AE/AWB
          CHANGE REGISTERS
          ENABLE AF or AE/AWB
```

5.7.5 Inter-frame Operations

In between frames, it may be necessary to modify the memory pointers before processing the next frame. Since the PCR and memory pointer registers are shadowed, these modifications can take place any time before the end of the frame, and the data will get latched in for the next frame. The host controller can perform these changes upon receiving an interrupt, or an EDMA transfer can be programmed to make these changes upon receiving an event.

5.7.6 Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the H3A. It can be used as a quick checklist. More detailed register setting constraints can be found in the individual register descriptions.

- The output addresses must be on 64 byte boundaries.

AF Engine:

- The paxel horizontal start value must be greater than or equal to the IIR horizontal start position.
- The width and height of the paxels must be an even number.
- The minimum width of the auto focus paxel must be 6 pixels.
- Paxels cannot overlap the last pixel in a line.
- Paxels must be adjacent to one another.

AEW Engine:

- The width and height of the windows must be an even number.
- Sub-sampling windows can only start on even numbers.
- The minimum width of the AE/AWB windows must be 6 pixels.

5.8 Programming the Histogram

This section discusses issues related to the software control of the histogram module. It lists which registers are required to be programmed in different modes, how to enable and disable the histogram, how to check the status of the histogram, discusses the different register access types, and enumerates several programming constraints.

5.8.1 Hardware Setup/Initialization

This section discusses the configuration of the histogram required before image processing can begin.

5.8.1.1 Reset Behavior

Upon hardware reset of the VPSS, all of the registers in the histogram are reset to their reset values. However, since the histogram output memory is stored in internal RAM, its contents do not have reset values. If the reset is a chip-level power-on reset (reset after power is applied), then the contents of this memory is unknown. If the reset is a VPSS module reset (when power remains active), then the contents of this memory remains the same as before the reset.

5.8.1.2 Reset of Histogram Output Memory

Before enabling the histogram, the output memory should be cleared. This can be done by two ways.

- Writing zeros to the RAM via software.
- If the HIST_CNT.CLR bit is set, reading the memory will cause it to be reset after the read.

Reads and writes to the output memory are blocked when the PCR.BUSY bit is 1.

5.8.1.3 Register Setup

Prior to enabling the histogram module, the hardware must be properly configured via register writes. Table 46 identifies the register parameters that must be programmed before enabling the histogram.

Table 46. Histogram Required Configuration Parameters

Function	Configuration Required
Histogram Control Bits	HIST_CNT.SOURCE HIST_CNT.CFA HIST_CNT.BINS HIST_CNT.SHIFT HIST_CNT.CLR
White Balance Gain	WB_GAIN
Region 0 Size and position	R0_HORZ R0_VERT

Table 47 identifies additional configuration requirements depending on if the corresponding condition is met. The table can be read as:

```
if(Condition is TRUE)
       then Configuration Required parameters must be programmed
```

Table 47. Histogram Conditional Configuration Parameters

Function	Condition	Configuration Required
Input from SDRAM	HIST_CNT.SOURCE	HIST_CNT.DATSIZ RADD RADD_OFF H_V_INFO
Less than 256 bins	HIST_CNT.BINS < 3	R1_HORZ R1_VERT
Less than 128 bins	HIST_CNT.BINS < 2	R2_HORZ R2_VERT R3_HORZ R3_VERT

5.8.2 Enable/Disable Hardware

Setting the PCR.ENABLE bit enables the histogram module. This should be done after all of the required registers mentioned in the previous section are programmed and the output memory has been cleared.

When the input source is the SDRAM/DDRAM, the histogram module always operates in the one shot mode. In other words, after enabling the histogram, the PCR.ENABLE bit is automatically turned off (set to 0) and only a single frame is processed from memory. In this mode, fetching and processing of the frame begins immediately upon setting the PCR.ENABLE bit.

When the input source is the CCD/CMOS controller, the histogram always operates in continuous mode. Processing of the frame is dependent upon the timing of the CCD/CMOS controller. In order to guarantee that data from the CCD/CMOS controller is not missed, the histogram should be enabled prior to the CCD/CMOS controller. In this way, the histogram will wait for data from the CCD/CMOS controller.

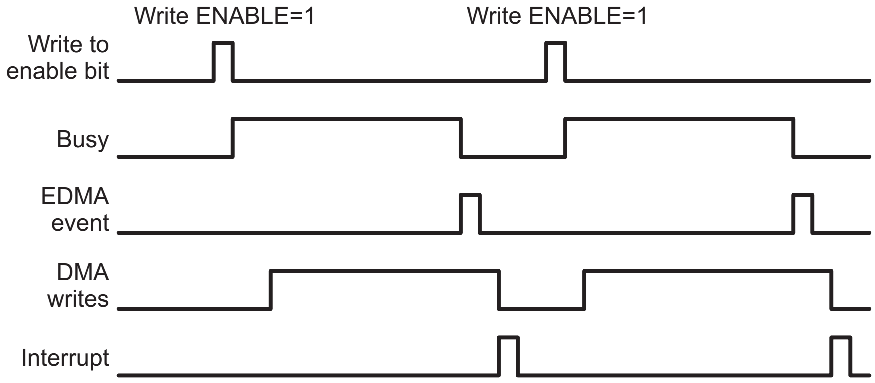
When the histogram is in continuous mode, it can be disabled by clearing the PCR.ENABLE bit during the processing of the last frame. The disable will be latched in at the end of the frame it was written in.

5.8.3 Events and Status Checking

The histogram will generate both an interrupt and an EDMA event for each frame as shown in Figure 60.

The PCR.BUSY status bit is set when the start of frame occurs (if the PCR.ENABLE bit is 1 at that time). The PCR.BUSY bit is automatically cleared to 0 and the EDMA event is triggered when it is safe to modify the busy-lock registers (see Section 5.8.4) to setup the next frame. When this happens, the frame may not yet fully have reached its destination. Instead, the interrupt signal indicates when all the data has reached its destination. The PCR.BUSY status bit may be polled to determine when the processor can update the registers, or the EDMA trigger can be used to trigger an EDMA transfer to update the registers. Configuration registers and filter coefficients should be programmed in-between busy periods, before writing ENABLE=1. Figure 60 indicates this interaction in SDRAM input-mode.

Figure 60. Firmware Interaction for SDRAM-Input Histogram



5.8.4 Register Accessibility During Frame Processing

There are two types of register access in the histogram module.

- Shadow registers
  - These registers/fields can be read and written (if the field is writeable) at any time. However, the written values take effect only at the start of a frame. Reads will still return the most recent write even though the settings are not used until the next start of frame.
  - The PCR, RADD, and RADD_OFF registers are the only shadowed registers in the histogram module.
- Busy-lock registers
  - All registers except the PCR, RADD, and RADD_OFF registers belong to this category.
  - Busy-lock registers cannot be written when the module is busy. Writes will be allowed to occur, but no change will occur in the registers (blocked writes from hardware perspective, but allowed write from the software perspective). Once the busy bit in the PCR register is reset to 0, the busy-lock registers can be written.
  - The HIST_DATA register cannot be read when the histogram is busy since this register is mapped to memories internally. Such reads will return indeterminate data. Byte enables are not implemented for reading the histogram memory.

The ideal procedure for changing the histogram registers is:

```
IF (PCR.BUSY == 0) OR IF (EOF interrupt occurs)
          DISABLE HISTOGRAM
          CHANGE REGISTERS
          ENABLE HISTOGRAM
```

5.8.5 Inter-frame Operations

In between frames read from memory, it may be necessary to modify the input memory pointers before processing the next frame. Since the PCR and memory pointer registers are shadowed, these modifications can take place any time before the end of the frame, and the data will get latched in for the next frame. The host controller can perform these changes upon receiving an interrupt, or an EDMA transfer can be programmed to make these changes upon receiving an event.

If continuous frames are processed without clearing the histogram output memory, the bin counters will contain the counts of however many images are processed since they were last cleared. In order to read the bin counters for each frame, the bin counters must be read after each frame is completed but before the next frame begins (since the counters can not be read while the PCR.BUSY bit is 1).

If the input source is SDRAM (one-shot mode), then the PCR.ENABLE bit needs to be set once for the frame, and after the frame is completed, the bin counters can be read/cleared before enabling the next frame.

When the input source is the video port interface of the CCD/CMOS controller (continuous mode), then the PCR.ENABLE bit needs to be set to enable processing of the frame, and cleared once the frame processing has begun (the disable will be latched in at the end of the frame). This procedure allows only one frame to be processed. After the frame is completed, the bin counters can be read/cleared before enabling the histogram for the next frame.

5.8.6 Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the histogram. It can be used as a quick checklist. More detailed register setting constraints can be found in the individual register descriptions.

- The input address and line offset must be on 32 byte boundaries.
- A region dimension of 1 (horizontal or vertical or both) is not allowed.

5.9 Programming the Shared Buffer Logic (VPSS Registers)

The shared buffer logic (SBL) controls the data interactions among the modules within the VPSS and DMA interface to memory. There are a few registers that configure the priority of the VPSS DMA transfers and the latencies associated with VPFE module DMA reads from SDRAM/DDRAM memory. This section discusses issues related to the software control of the VPSS SBL and lists the registers required to be programmed in different modes, how to check the status of the SBL overflow bits, and enumerates several programming constraints.

5.9.1 Hardware Setup/Initialization

This section discusses the configuration of the SBL (VPSS MMR) required before image processing can begin.

5.9.1.1 Reset Behavior

Upon hardware reset of the VPSS, all of the registers in the SBL (VPSS MMR) are reset to their reset values.

5.9.1.2 Register Setup

Prior to enabling any of the VPSS modules, the hardware must be properly configured via register writes to the VPSS registers. The PCR.CPRIO parameter must be set. This parameter configures the DDR EMIF priority in the system. The default value, 0 (highest priority), is recommended.

If the preview engine, Resizer, or the histogram will be reading from SDRAM/DDRAM, then the SDR_REQ_EXP register must be programmed. The values programmed in each of the three fields (PRV_EXP, RESZ_EXP, HIST_EXP) determines how many DMA cycles to space in between DMA memory read requests for each module.

5.9.2 Enable/Disable Hardware

The SBL functionality is always enabled, unless the PSC idles the clocks to the VPSS.

5.9.3 Events and Status Checking

The status bits in the PCR can be used to monitor if a memory overflow (which causes data corruption) has occurred. See Table 48.

Table 48. VPSS Error Indicators

PCR Bit	Error Condition
CCDC_WBL_O	Write buffer memory overflow (CCDC)
PRV_WBL_O	Write buffer memory overflow (preview engine)
RSZ1_WBL_O	Write buffer memory overflow (Resizer line 1)
RSZ2_WBL_O	Write buffer memory overflow (Resizer line 2)
RSZ3_WBL_O	Write buffer memory overflow (Resizer line 3)
RSZ4_WBL_O	Write buffer memory overflow (Resizer line 4)
AF_WBL_O	Write buffer memory overflow (AF)
AEW_WBL_O	Write buffer memory overflow (AE/AWB)

5.9.4 Register Accessibility During Frame Processing

The SBL VPSS registers are all busy-writable registers. These registers/fields can be read or written even if the module is busy. Changes to the underlying settings takes place instantaneously.

5.9.5 Inter-frame Operations

None noted.

5.9.6 Summary of Constraints

None noted.

5.10 Error Identification

Memory overflow or underflow errors can occur in the read and write buffers of the VPFE. When an overflow from any of the write buffers occurs, a corresponding fail bit is set in the VPSS.PCR register. If the fault pixel correction is used, a read fail of fault pixel data from DDR2 is identified by the CCDC.FPC.FPERR bit. Finally, if darkframe subtract or shading compensation is used, a read fail of the data from DDR2 is identified by the PREV.PCR.DARK_FAIL bit. Depending on the read buffers and write buffers used in a particular mode, these bits can be monitored by software to detect a fail and make the necessary adjustments. Since these types of fails involve the transfer of data to and from the DDR2, the corrective action may be to increase the clocks or reduce the memory bandwidth usage of the system.

5.11 Supported Use Cases

The VPFE is designed to support a variety of video and imaging applications. For the purposes of describing the configuration of the VPFE for typical use cases, the application space can be divided into the following two input types: CCD/CMOS sensor data and YUV video data. This section discusses typical VPFE configurations for both of these input types separately, and then discusses how both applications use the resizer to resize or change the aspect ratio of processed video or image data.

Figure 52 shows all the possible data paths through the VPFE. Each mode described in this section has a unique data path through the various modules.

5.11.1 CCD/CMOS Sensor Input Specific Applications

Digital still cameras and digital video cameras are the primary applications that use CCD or CMOS sensor input sources. CCD or CMOS sensors output analog data at a rate determined by a timing generator (TG). The analog front end (AFE) converts this data to a digital signal and transmits this digital raw sensor data to the input interface of the CCD controller. Depending on the sensor, this data is typically in a Bayer pattern where every pixel represents only one of the three primary colors (RGB). The VPFE contains programmable functions that capture and digitally process this raw data into YUV-formatted video or image data that can be compressed or displayed directly on an external display.

In this application, there are three basic modes of operation that require different VPFE data paths and configurations:

- Preview/movie capture mode
- Still image capture mode
- Still image processing mode

5.11.1.1 Preview/Movie Capture Mode

In a digital still camera or video camera, preview and movie capture modes are where the VPFE receives raw video data from the sensor, converts it to YUV format, and displays it on the display in real-time. There is only one distinguishing detail between preview mode and movie capture mode. In preview mode, the video data is only temporarily stored in a circular buffer in the DDR memory until it can be displayed and/or transmitted; in movie capture mode, the video data is additionally compressed and stored in non-volatile memory (for example, Flash, digital video tape, DVD, hard disk, etc). Both modes have the same data path through the VPFE as shown in Figure 61. The register configuration for both modes is shown in Table 49.

As shown in Figure 61, all of the modules in the VPFE may be enabled for this mode. The camera application spends the majority of its time in this mode so the DDR2 bandwidth consumption is minimized by having a single image data path to memory (in addition to the H3A statistic data path), which is resized to the correct display size. The ARM or an EDMA transfer must read the statistic data from the histogram memory-mapped registers after a frame has completed.

Typically in this mode, the CCD controller receives the digital raw data from the sensor/AFE in a down-sampled resolution format. The 10-bit raw data is output on the video port interface feeding the preview engine, H3A, and histogram modules. The preview engine performs most of the image signal processing (CFA interpolation, white balance, noise filtering, etc) and coverts the raw data to YUV 4:2:2 video format. If the output of the preview engine is not the correct size or aspect ratio for display, and/or if digital zoom is required, then it can be sent directly to the resizer. The image output from the preview engine and/or resizer is finally sent to the DDR2 in a circular buffer where it is consumed for display and/or compression. End-of-frame interrupts from the resizer and/or other VPFE modules can trigger ARM interrupt service routines to change the address of the write buffer in the preview engine and/or resizer module for each frame. The image statistics from the H3A and histogram can by used by the ARM for implementing algorithms to modify the image processing parameters of the preview engine and/or focus lens of the imager for subsequent frames.

Figure 61. Preview/Movie Capture Mode Data Path

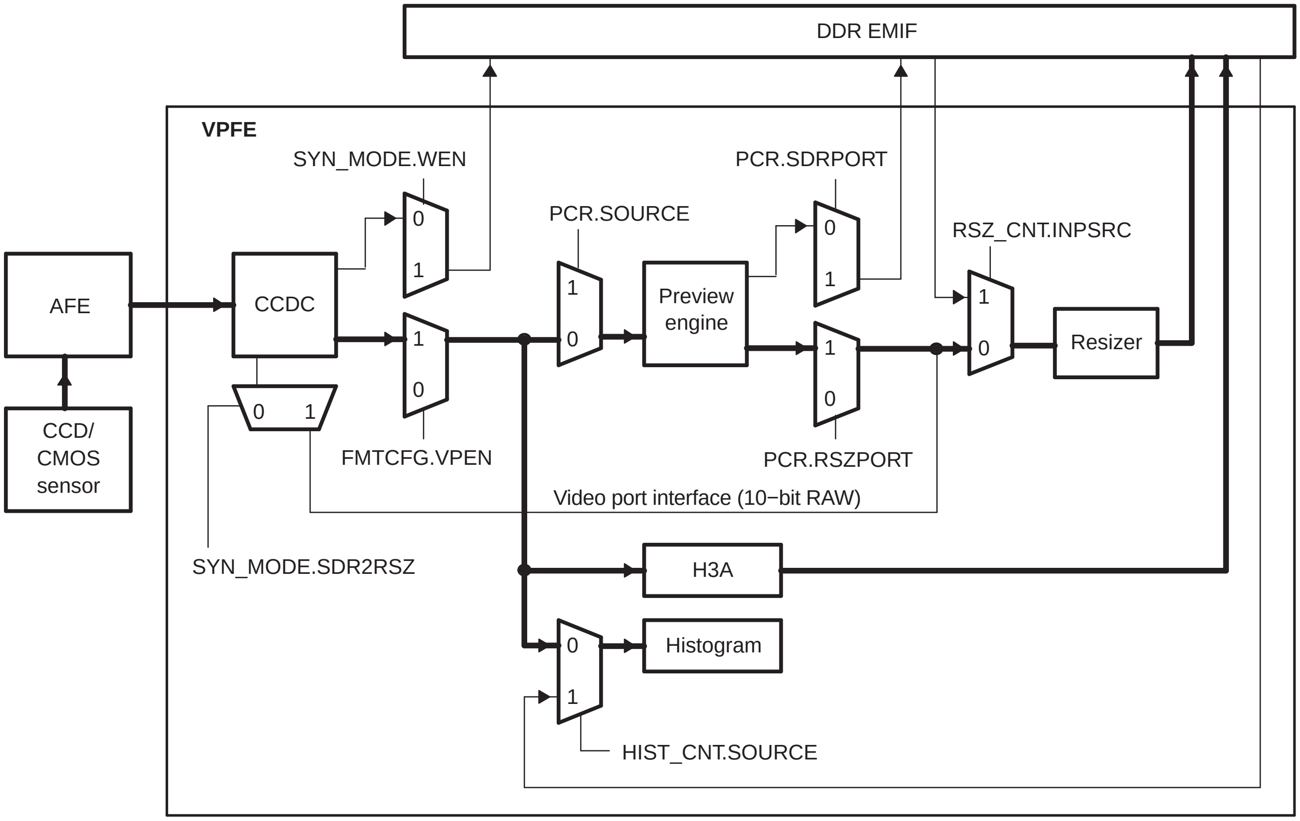


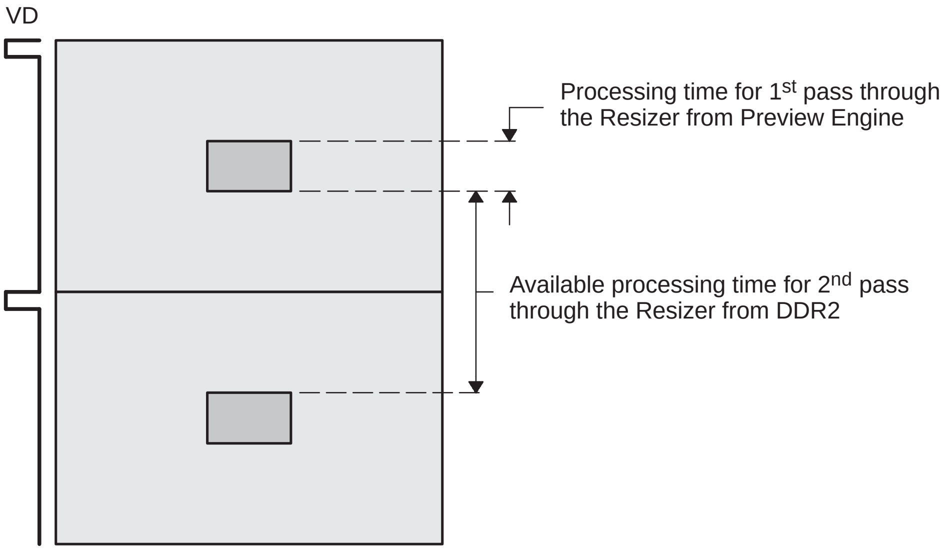
Table 49. Preview/Movie Capture Mode Data Path Register Configuration

Module	Register	Field	Setting
CCD Controller	FMTCFG	VPEN	1
	SYN_MODE	WEN	0
	SYN_MODE	SDR2RSZ	0
Preview Engine	PCR	SOURCE	0
	PCR	RSZPORT	1
	PCR	SDRPORT	0
Resizer	RSZ_CNT	INPSRC	0
Histogram	HIST_CNT	SOURCE	0

5.11.1.1.1 Two-Pass Resize in Preview/Movie Capture Mode

A single pass through the resizer can resize an image down to 1/4 or up to 4 times the input width and/or height. In order to achieve resize operations beyond this range, multiple passes through the resizer are required. The VPFE supports a two-pass resize operation in preview/movie capture mode to achieve a resize zoom range beyond 4× and up to 16× digital zoom. The first pass is performed “on-the-fly” directly from the output of the preview engine (RESZ.RESZ_CNT.INPSRC = 0). Since this is a zoom in, only a small portion of the image is resized up in the first pass. The second pass is performed from the DDR (RESZ.RESZ_CNT.INPSRC = 1) during the time that the inactive region of the image is being sent to the CCD controller. Since the resizer operation from the DDR2 is at 1/2 the DMA clock speed (typically faster than the PCLK) and is performed on an image smaller than the original (with blanking), there is enough time for the second pass to complete before switching the input source of the resizer to the preview engine again for the next frame, as shown in Figure 62.

Figure 62. Preview Mode Two-Pass Resize Processing Time


VD
Processing time for 1st pass through the Resizer from Preview Engine
Available processing time for 2nd pass through the Resizer from DDR2

The pseudo-code in Example 4 shows the control sequence that needs to be performed by an interrupt service routine or EDMA transfer of register writes to the resizer after the resizer is initialized for the first resize. Since both passes perform up-sampling, the filter coefficients do not need to be changed once they are initially set. This example performs a 10× resize where the first pass is 4× and the second pass is 2.5×.

Memory/Bandwidth Tradeoff Note: When determining which resize ratio to use in the first pass versus the second pass, consider the following as it relates to the system constraints: a smaller ratio in the first pass will minimize the intermediate buffer memory used, but will result in a higher instantaneous bandwidth requirement during the second pass. Depending on the real-time requirements, the instantaneous bandwidth consumption can be reduced by employing the techniques described in Section 4.4.2.

Example 4. Preview Mode 10× Resize Pseudo-Code Example

```
void reszISR()
{
     if(pass0)
     {
         RSZ_CNT.INPSRC = 0;                  // Input source is Preview Engine
         SDR_INADD = 0;                       // Required to be 0 for PREV input
         SDR_INOFF = 0;                       // Required to be 0 for PREV input
         SDR_OUTADD = outBuff_pass0;          // Output buffer address
         SDR_OUTOFF = offset_pass0;           // Output buffer offset
         RSZ_CNT.VRSZ = vert4X;               // Vertical Resize ratio
         RSZ_CNT.HRSZ = horz4X;               // Horizontal Resize ratio
         IN_SIZE = insize_pass0;              // Horizontal & Vertical input size
         OUT_SIZE = outsize_pass0;            // Horizontal & Vertical output size
         IN_START = instart_pass0;            // Horizontal & Vertical input start
     }
     else
     {
         buffNum %= NUM_BUFFS;                // Implementation of a circular buffer
         RSZ_CNT.INPSRC = 1;                  // Input source is DDR
         SDR_INADD = outBuff_pass0;           // Output of 1st pass
         SDR_INOFF = offset_pass0;            // Output of 1st pass
         SDR_OUTADD = outBuff[buffNum];       // Output buffer address
         SDR_OUTOFF = offset_pass1;           // Output buffer offset
         RSZ_CNT.VRSZ = vert2_5X;             // Vertical Resize ratio
         RSZ_CNT.HRSZ = horz2_5X;             // Horizontal Resize ratio
         IN_SIZE = insize_pass1;              // Horizontal & Vertical input size
         OUT_SIZE = outsize_pass1;            // Horizontal & Vertical output size
         IN_START = instart_pass1;            // Horizontal & Vertical input start
         buffNum++;                              // Increments the output buffer number
     }

     PCR.ENABLE = 1;

     pass0 = !pass0;
}
```

5.11.1.2 Still Image Capture Mode

In a digital still camera or video camera, still image capture mode is where the VPFE is receiving full resolution raw image data from the sensor and storing it to the DDR to be analyzed, processed, and/or later stored to non-volatile memory (for example, Flash, digital video tape, DVD, hard disk, etc). The data path through the VPFE for still image capture is shown in Figure 63. The register configuration for this mode is shown in Table 50.

As shown in Figure 63, only the CCD controller module in the VPFE needs to be enabled. Typically in this mode, the CCD controller receives the full resolution of the digital raw data from the sensor/AFE. Some sensors (typically CMOS) read the data out in progressive format, whereas others (typically CCD) read the data out in multiple fields. If the input format is field based, then the software should set the line offset and starting address of the CCD controller output accordingly so that the frame is de-interleaved as it is stored into the DDR memory.

Most of the time, a single frame is captured in this mode. However, the VPFE also supports a burst (continuous) capture mode for as many frames as can fit in the DDR2. Software implements this by managing the output buffer address in the CCD controller for each frame.

Figure 63. Still (Raw) Image Capture Mode Data Path

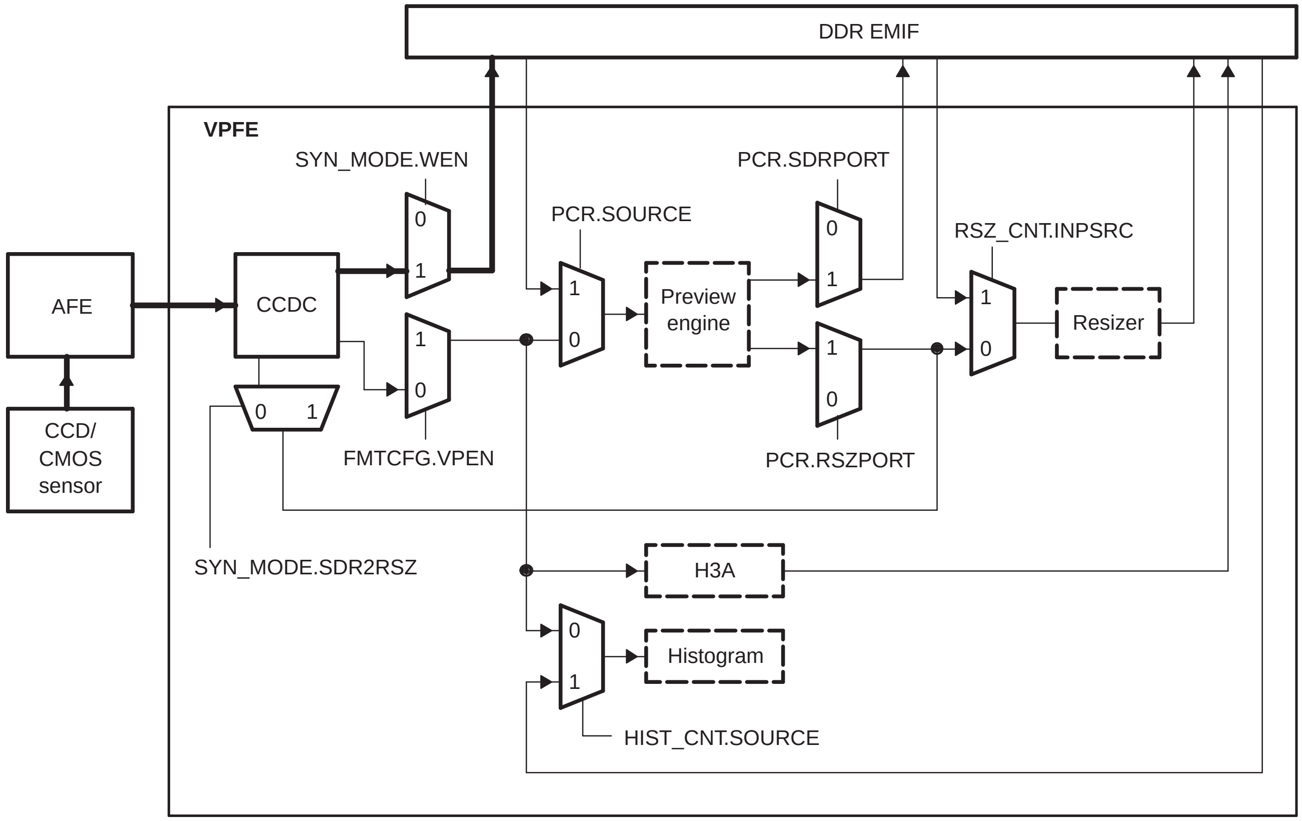


Table 50. Still Image Capture Mode Data Path Register Configuration

Module	Register	Field	Setting
CCD Controller	FMTCFG	VPEN	0
	SYN_MODE	WEN	1
	SYN_MODE	SDR2RSZ	0
Preview Engine	PCR	SOURCE	x
	PCR	RSZPORT	x
	PCR	SDRPORT	x
Resizer	RSZ_CNT	INPSRC	x
Histogram	HIST_CNT	SOURCE	x

5.11.1.3 Still Image Processing Mode

In a digital still camera or video camera, still image processing mode is when the raw image data that was captured to the DDR2 during still capture mode is processed into YUV image data that can be compressed and stored in non-volatile memory. This mode can optionally be performed by the preview engine of the VPFE. If performed by the VPFE, the data path through the VPFE is shown in Figure 64. The register configuration for this mode is shown in Table 51.

Figure 64. Still Image Processing Mode Data Path

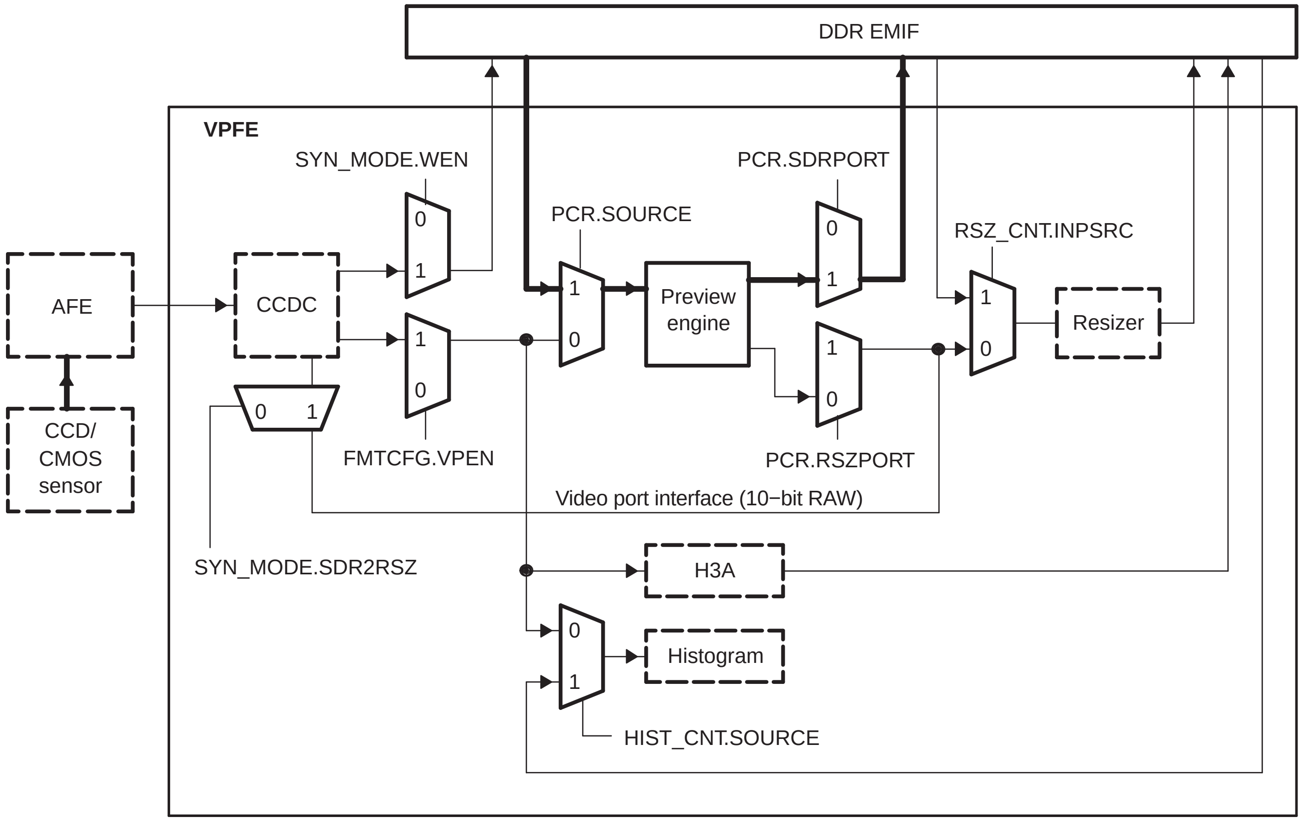


Table 51. Still Image Processing Mode Data Path Register Configuration

Module	Register	Field	Setting
CCD Controller	FMTCFG	VPEN	x
	SYN_MODE	WEN	x
	SYN_MODE	SDR2RSZ	x
Preview Engine	PCR	SOURCE	1
	PCR	RSZPORT	0
	PCR	SDRPORT	1
Resizer	RSZ_CNT	INPSRC	x
Histogram	HIST_CNT	SOURCE	x

Since the internal line memory is optimized for video resolutions, the preview engine can process a maximum output width of 1280 pixels in a single pass. Therefore, it usually requires multiple passes through the preview engine to process a full resolution still-captured image. This is done by partitioning the input image into multiple overlapping vertical slices and aligning the output in such a way that the processed image is seamlessly stitched together. The number of slices required is:

$$\textit{Number of slices} = [\textit{FLOOR} \times ((\textit{output\_width} - 1)/1280)] + 1$$

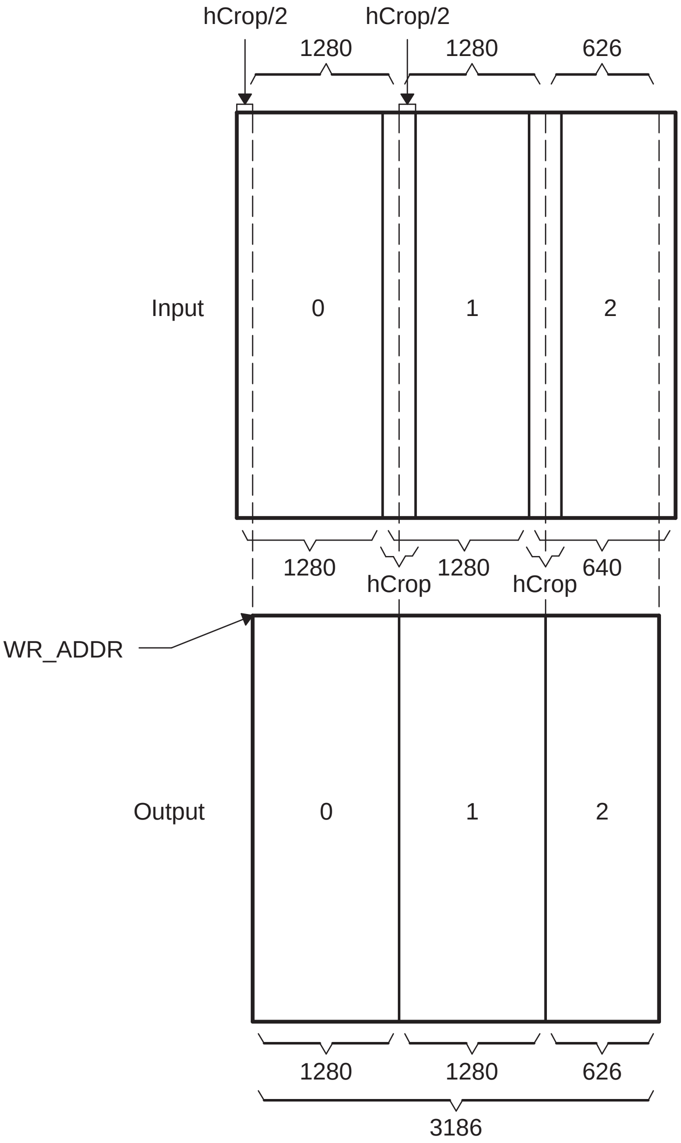
The vertical slices processed by the preview engine are required to overlap, due to the truncating of the image data by internal filtering processes as discussed in Section 4.3.3.13. The amount of overlap is determined by which filtering functions are enabled during processing. Table 52 indicates how many edge pixels/lines are truncated by enabling certain functions within the preview engine.

Table 52. Image Cropping by Preview Functions

Function	Pixels/line	Lines
Horizontal Median Filter	4	0
Noise Filter	4	4
CFA	4	4
Color Suppression OR Luminance Enhancement	2	0
Maximum Total	14	8

Figure 65 and the pseudo-code in Example 5 illustrate the implementation of multi-pass still image processing through the preview engine. For the purposes of this discussion, *hCrop* refers to the pixels per line that are truncated from the image due to internal filtering functions. The amount of overlap of the slices is equal to *hCrop* pixels.

Figure 65. Multi-Pass Processing Through Preview Engine



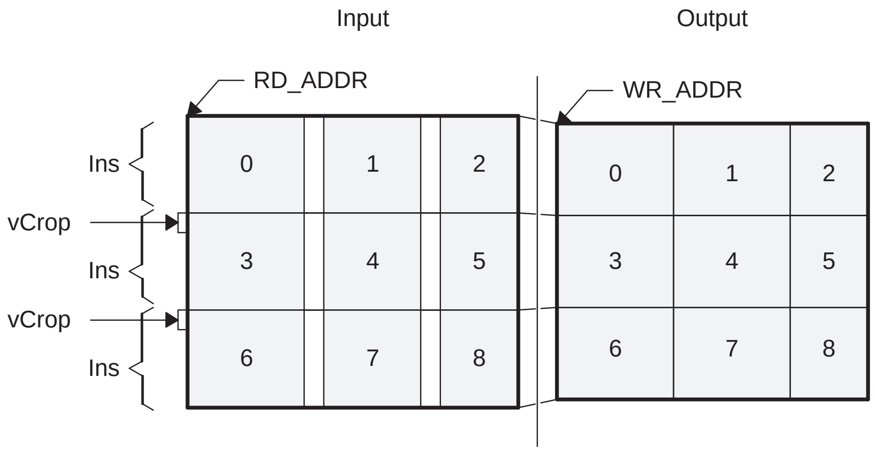
Example 5. Still Image Processing Pseudo-Code Example

```
#define HCROP             14           // cropped pixels (all functions on)
#define OUTWIDTH          3186         // required output width
#define VSTRIPEWIDTH      1280         // must be a multiple of 16 pixels

int vStripeCnt, lastPassNum, lastPassSize;

void stillProcess()        // assumes all other PREVIEW parameters initialized
{
     int inWidth;

     vStripeCnt = 0;                    // initialize vStripe counter

     inWidth = OUTWIDTH + HCROP;        // 3200 (set CCDC output width to this)

     lastPassNum = (OUTWIDTH-1) / VSTRIPEWIDTH;    // 2
     lastPassSize = OUTWIDTH % VSTRIPEWIDTH;       // 626

     if(lastPassSize == 0)        // if output width is less than 1280
lastPassSize = OUTWIDTH;

     HORZ_INFO.SPH = 0;                           // start pixel horizontal
     HORZ_INFO.EPH = (VSTRIPEWIDTH-1) + HCROP;    // end pixel horizontal
     RADR_ADDR = input_buff;                      // Read starting address
     WSDR_ADDR = output_buff;                     // Write starting address

     PCR.ENABLE = 1;            // Enable Preview Engine
     ...
}

void prevISR()
{
    vStripeCnt++;
    RADR_ADDR = input_buff + 2*(vStripeCnt * VSTRIPEWIDTH);
    WSDR_ADDR = output_buff + 2*(vStripeCnt * VSTRIPEWIDTH);

    if(vStripeCnt == lastPassNum)
         HORZ_INFO.EPH = (lastPassSize-1) + HCROP;

    if(vStripeCnt <= lastPassNum)
         PCR.ENABLE = 1;        // Enable Preview Engine
}
```

5.11.1.3.1 Horizontal Slicing

In addition to vertical slicing, it may be required to perform horizontal slicing of the processing so that compression of the output image can be pipelined in parallel. This concept can be implemented in much the same way as the vertical slicing. The number of lines overlapping is equal to the number of lines that are truncated from the image (*vCrop*) as specified in Table 52. Figure 66 and the pseudo-code in Example 6 illustrate the implementation of vertical and horizontal slicing through the preview engine.

Figure 66. Horizontal Slicing Through Preview Engine



Example 6. Horizontal and Vertical Slicing Pseudo-Code Example

```
#define HCROP             14           // cropped pixels (all functions on)
#define VCROP             8            // cropped lines (all functions on)
#define OUTWIDTH          3186         // required output width
#define OUTHEIGHT         2392         // required output width
#define VSTRIPEWIDTH      1280         // must be a multiple of 16 pixels
#define HSTRIPEHEIGHT     960          // can be anything, not limited

int vStripeCnt, hStripeCnt, lastPassNum, lastPassSize, vLastPassNum, vLastPassSize,
numPasses, passCnt;

void stillProcess()        // assumes all other PREVIEW parameters initialized
{
     int inWidth, inHeight;

     vStripeCnt = 0;                 // initialize vStripe counter
     hStripeCnt = 0;                 // initialize hStripe counter
     passCnt = 0;                    // initialize pass counter

     inWidth = OUTWIDTH + HCROP;          // 3200 (set CCDC output width to this)
     inHeight = OUTHEIGHT + VCROP;        // 2400 (set CCDC output height to this)

     lastPassNum = (OUTWIDTH-1) / VSTRIPEWIDTH;    // 2
     lastPassSize = OUTWIDTH % VSTRIPEWIDTH;       // 626

     if(lastPassSize == 0)        // if output width is less than 1280
        lastPassSize = OUTWIDTH;


     vLastPassNum = (OUTHEIGHT-1) / HSTRIPEHEIGHT;     // 2
     vLastPassSize = OUTHEIGHT % HSTRIPEHEIGHT;        // 472

     if(vLastPassSize == 0)        // if output height is less than 960
        vLastPassSize = OUTHEIGHT;

numPasses = lastPassNum+1 * vLastPassNum+1;
```

Example 6. Horizontal and Vertical Slicing Pseudo-Code Example (continued)

```
     HORZ_INFO.SPH = 0;                          // start pixel horizontal
     HORZ_INFO.EPH = (VSTRIPEWIDTH-1) + HCROP;    // end pixel horizontal
     VERT_INFO.SLV = 0;                          // start line vertical
     VERT_INFO.ELV = (HSTRIPEHEIGHT-1) + VCROP;    // end line vertical
     RSDR_ADDR = input_buff;                       // Read starting address
     WSDR_ADDR = output_buff;                       // Write starting address

     PCR.ENABLE = 1;            // Enable Preview Engine
     ...
}

void prevISR()
{
    vStripeCnt++;
    RADR_ADDR = input_buff + 2*(vStripeCnt * VSTRIPEWIDTH)
                  + (hStripeCnt * HSTRIPEHEIGHT * RADR_OFFSET);

    WSDR_ADDR = output_buff + 2*(vStripeCnt * VSTRIPEWIDTH)
                   + (hStripeCnt * HSTRIPEHEIGHT * WADD_OFFSET);

    if(vStripeCnt == lastPassNum) {
         HORZ_INFO.EPH = (lastPassSize-1) + HCROP;
            vStripeCnt = -1;
            hStripeCnt++;
        }
        else if(hStripeCnt == vLastPassNum)
            VERT_INFO.ELV = (vLastPassSize-1) + VCROP;

    if(++passCnt < numPasses)
         PCR.ENABLE = 1;        // Enable Preview Engine
}
```

5.11.2 YUV Video Input Specific Applications

There are a variety of applications that use YUV video input sources: IP phones, video surveillance systems, digital video recorders, etc. Mostly any application that needs to capture YUV video, compress it, and transmit or store it can be included in this application category.

5.11.2.1 Video Capture Mode

Video capture is where the VPFE is receiving YUV video data from a digital video source and storing it to the DDR for further processing and/or compression. The CCD controller of the VPFE can capture BT.656 formatted video or generic 16-bit or 8-bit YUV digital video data from a digital video source such as an NTSC/PAL video decoder. The possible data paths through the VPFE for video capture are shown in Figure 67. The register configuration for this mode is shown in Table 53.

As shown in Figure 67, the CCD controller is enabled for this mode. Additionally, if the input video is progressive, the output of the CCD controller can be passed directly to the resizer for resizing the video data without having to first go to the DDR2.

Figure 67. Video Capture Mode Data Path

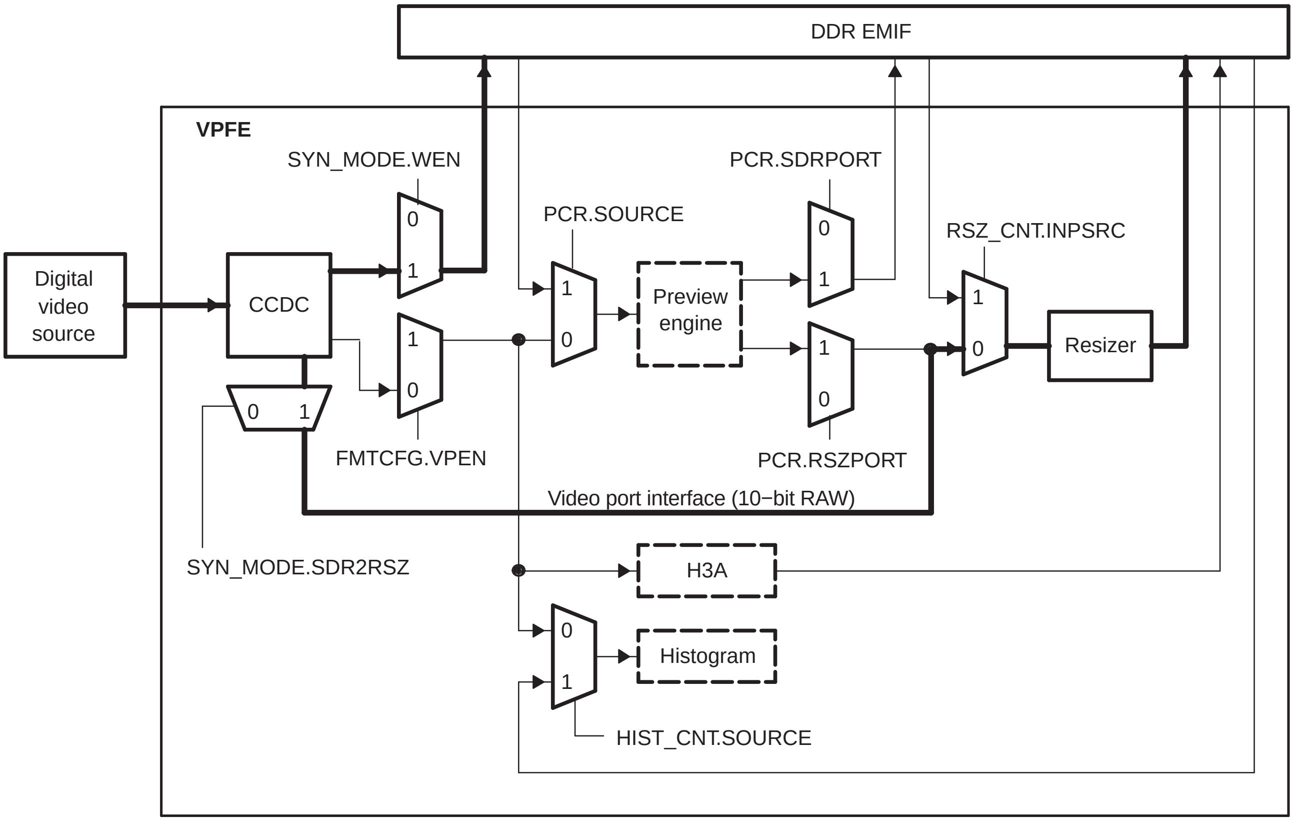


Table 53. Video Capture Mode Data Path Register Configuration

Module	Register	Field	Setting
CCD Controller	FMTCFG	VPEN	0
	SYN_MODE	WEN	1
	SYN_MODE	SDR2RSZ	1
Preview Engine	PCR	SOURCE	x
	PCR	RSZPORT	x
	PCR	SDRPORT	x
Resizer	RSZ_CNT	INPSRC	0
Histogram	HIST_CNT	SOURCE	x

5.11.3 Video/Image Resize Applications

Applications from both CCD/CMOS sensor data and YUV video data input categories use the resizer to resize YUV 4:2:2 formatted video and images from the DDR2 memory. Additionally, the resizer can be configured to process YUV 4:4:4 planar data.

5.11.3.1 *Processed Image Resize*

Processed image resize is where the resizer in the VPFE takes YUV formatted image/video data from the DDR and resizes it back to the DDR. Typically, the ARM, with help from the HMJCP, will uncompress image or video data first, and then have the resizer resize the image to be displayed on a display device or recompressed again. The data path through the VPFE for processed image resizing is shown in Figure 68. The register configuration for this mode is shown in Table 54.

Figure 68. Processed Image Resize Data Path

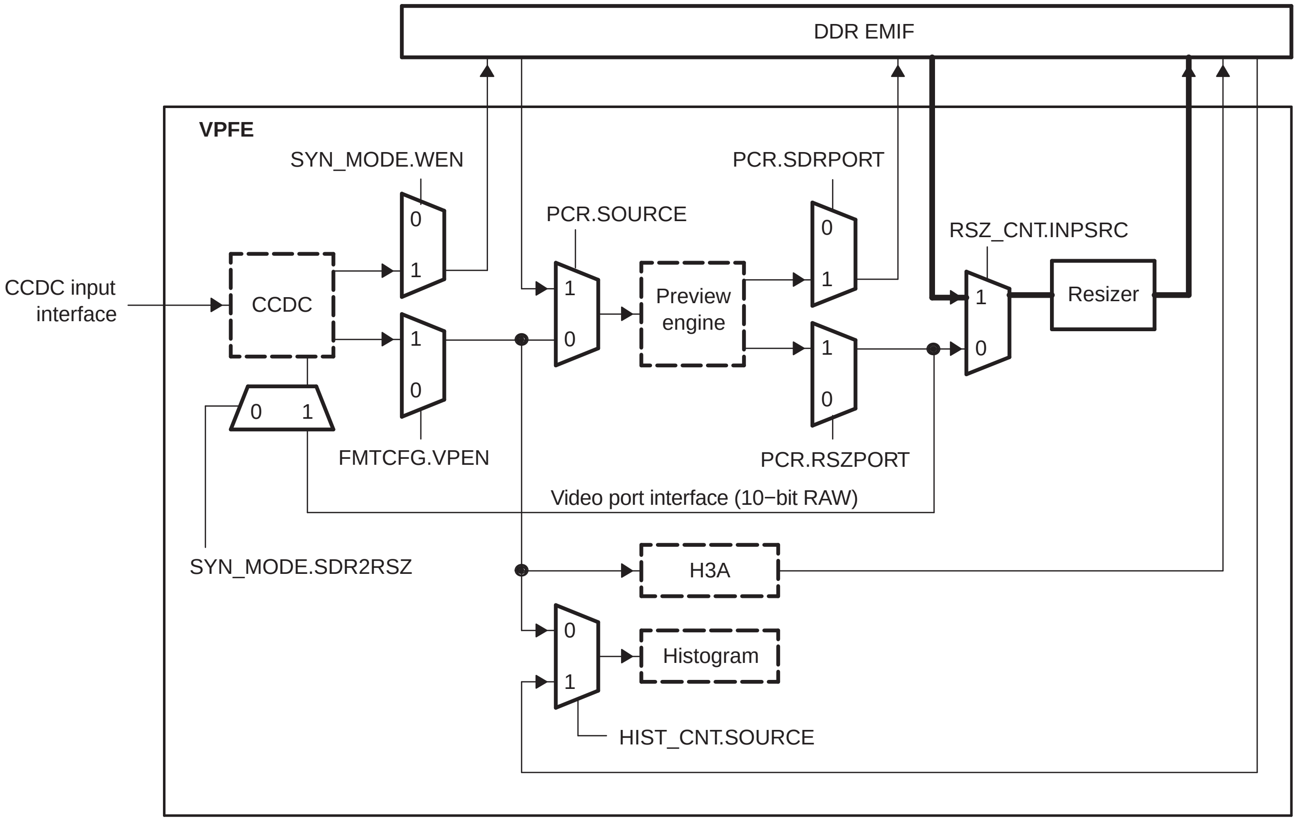


Table 54. Processed Image Resize Data Path Register Configuration

Module	Register	Field	Setting
CCD Controller	FMTCFG	VPEN	x
	SYN_MODE	WEN	x
	SYN_MODE	SDR2RSZ	x
Preview Engine	PCR	SOURCE	x
	PCR	RSZPORT	x
	PCR	SDRPORT	x
Resizer	RSZ_CNT	INPSRC	1
Histogram	HIST_CNT	SOURCE	x

5.11.3.1.1 Multi-Pass Resize

There are two constraints in the resizer that may require multiple passes in order to achieve a desired size:

- Resize ratio range is limited to between 1/4× and 4× resize.
- Maximum output width is 1280 pixels (640 for vertical down-sampling ratios between 1/2 and 1/4).

Resize ratio range: A single pass through the resizer can resize an image down to 1/4 or up to 4× the input width and/or height. In order to achieve resize operations beyond this range, multiple passes through the resizer are required until the final sizing is achieved. For example, if a 10× resize is required, then a 4× resize and a 2.5× resize can be applied in two passes.

Maximum output width: Since the internal line memory is optimized for video resolutions, the resizer can operate on a maximum output width of 1280 (640 for vertical down-sampling ratios between 1/2 and 1/4) pixels in a single pass. Therefore, it requires multiple passes through the resizer to resize to larger images that exceed this horizontal width requirement. This is done by partitioning the input image into multiple vertical slices and aligning the output in such a way that the processed image is seamlessly stitched together.

The basic idea is to begin subsequent slices at exactly where previous images left off. The starting phase and pixel registers can be programmed to this exact location. This location can be calculated using the algorithm details in Section 4.3.4.4.

6 Video Processing Front End (VPFE) Registers

The submodules in the video processing front end (VPFE) subsystem are listed in Table 55.

Table 55. Video Processing Front End Subsystem Module Register Map

Offset	Acronym	Register Description	Section
400h	CCDC	CCD Controller	Section 6.1
800h	PREV	Preview Engine/Image Signal Processor	Section 6.2
C00h	RESZ	Resizer	Section 6.3
1000h	HIST	Histogram	Section 6.4
1400h	H3A	Hardware 3A (Auto-Focus/White Balance/Exposure)	Section 6.5

6.1 CCD Controller (CCDC) Registers

Table 56 lists the memory-mapped registers for the CCD controller. See the device-specific data manual for the memory address of these registers.

Table 56. CCD Controller (CCDC) Registers

Offset	Acronym	Register Description	Section
400h	PID	Peripheral revision and class information	Section 6.1.1
404h	PCR	Peripheral control register	Section 6.1.2
408h	SYN_MODE	SYNC and mode set register	Section 6.1.3
40Ch	HD_VD_WID	HD and VD signal width register	Section 6.1.4
410h	PIX_LINES	Number of pixels in a horizontal line and number of lines in a frame register	Section 6.1.5
414h	HORZ_INFO	Horizontal pixel information register	Section 6.1.6
418h	VERT_START	Vertical line - settings for the starting pixel register	Section 6.1.7
41Ch	VERT_LINES	Number of vertical lines register	Section 6.1.8
420h	CULLING	Culling information in horizontal and vertical directions register	Section 6.1.9
424h	HSIZE_OFF	Horizontal size register	Section 6.1.10
428h	SDOFST	SDRAM/DDRAM line offset register	Section 6.1.11
42Ch	SDR_ADDR	SDRAM address register	Section 6.1.12
430h	CLAMP	Optical black clamping settings register	Section 6.1.13
434h	DCSUB	DC clamp register	Section 6.1.14
438h	COLPTN	CCD color pattern register	Section 6.1.15
43Ch	BLKCMP	Black compensation register	Section 6.1.16
440h	FPC	Fault pixel correction	Section 6.1.17
444h	FPC_ADDR	Fault pixel Correction SDRAM Address	Section 6.1.18
448h	VDINT	VD interrupt timing register	Section 6.1.19
44Ch	ALAW	A-law setting register	Section 6.1.20
450h	REC656IF	REC656 interface register	Section 6.1.21
454h	CCDCFG	CCD configuration register	Section 6.1.22
458h	FMTCFG	Data reformatter/video port configuration register	Section 6.1.23
45Ch	FMT_HORZ	Data reformatter/video input interface horizontal information register	Section 6.1.24
460h	FMT_VERT	Data reformatter/video input interface vertical information register	Section 6.1.25
494h	VP_OUT	Video port output settings register	Section 6.1.26

6.1.1 Peripheral Revision and Class Information Register (PID)

The peripheral revision and class information register (PID) is shown in Figure 69 and described in Table 57 .

Figure 69. Peripheral Revision and Class Information Register (PID)

31 24	23 16
Reserved	TID
R-0	R-1

15 8	7 0
CID	PREV
R-FEh	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 57. Peripheral Revision and Class Information Register (PID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	TID		Peripheral identification
		1	CCD/CMOS controller
15-8	CID		Class identification
		FEh	Video processing front end module
7-0	PREV		Peripheral revision number
		0	Current revision

6.1.2 Peripheral Control Register (PCR)

Note: When the CCD controller is disabled (ENABLE = 0), the CCD controller continues to generate interrupts. If your design is not expecting interrupts, you must disable those interrupts at the interrupt controller (INTC). See the *TMS320DM357 DMSoC ARM Subsystem Reference Guide* (SPRUG25) for information on the INTC.

The peripheral control register (PCR) is shown in Figure 70 and described in Table 58.

Figure 70. Peripheral Control Register (PCR)

31–16
Reserved
R-0

15–2	1	0
Reserved	BUSY	ENABLE
R-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 58. Peripheral Control Register (PCR) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	BUSY		CCD controller busy bit
		0	Not busy
		1	Busy
0	ENABLE		CCD controller enable
		0	Disable
		1	Enable

6.1.3 Sync and Mode Set Register (SYN_MODE)

The sync and mode set register (SYN_MODE) is shown in Figure 71 and described in Table 59.

Figure 71. Sync and Mode Set Register (SYN_MODE)

31	24
Reserved	
R-0	

23	20	19	18	17	16
Reserved		SDR2RSZ	VP2SDR	WEN	VDHDEN
R-0		R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	8
FLDSTAT	LPF	INPMOD		PACK8	DATSIZ	

7	6	5	4	3	2	1	0
FLDMODE	DATAPOL	EXWEN	FLDPOL	HDPOL	VDPOL	FLDOUT	VDHDOUT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 59. Sync and Mode Set Register (SYN_MODE) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved
19	SDR2RSZ		SDRAM port output into resizer input. Controls whether or not SDRAM output data is forwarded to the resizer input port.
		0	Disable
		1	Enable
18	VP2SDR		Video port output into the SDRAM port. Controls whether or not video port data is forwarded to the output formatter, which in turn writes to SDRAM. Note that if VP2SDR is set, then SDRAM line (SLV*n* bit in VERT_START) and pixel start (SPH bit in HORZ_INFO) are with respect to the video port output (and not the original input). This bit is latched by VD.
		0	Disable
		1	Enable
17	WEN		Data write enable. Controls whether or not CCD raw data is written to SDRAM. This bit is latched by VD.
		0	Disable
		1	Enable
16	VDHDEN		VD/HD Enable. If VD/HD are defined as output, activates internal timing generator. If VD/HD are defined as inputs, activates internal timing generator to synchronize with VD/HD.
		0	Disable
		1	Enable
15	FLDSTAT		Field Status. Indicates the status of the current field when in interlaced mode.
		0	Odd field
		1	Even field
14	LPF		3-tap low-pass (anti-aliasing) filter. This bit is latched by VD.
		0	Off
		1	On

Table 59. Sync and Mode Set Register (SYN_MODE) Field Descriptions (continued)

Bit	Field	Value	Description
13-12	INPMOD	0-3h	Data input mode.
		0	CCD raw data
		1h	YCbCr 16-bit
		2h	YCbCr 8-bit
		3h	Reserved
11	PACK8		Pack to 8-bits/pixel (into SDRAM).
		0	Normal (16 bits/pixel)
		1	Pack to 8 bits/pixel
10-8	DATSIZ	0-7h	CCD data width. Valid only when INPMOD bit is cleared to 0.
		0	16 bits
		1h	15 bits
		2h	14 bits
		3h	13 bits
		4h	12 bits
		5h	11 bits
		6h	10 bits
		7h	8 bits
7	FLDMODE		Sensor field mode.
		0	Non-interlaced (progressive)
		1	Interlaced
6	DATAPOL		CCD data polarity.
		0	Normal (no change)
		1	1's complement
5	EXWEN		External WEN selection. When set to 1 and when VDHDEN bit is set to 1, input CCD data is loaded into SDRAM.
		0	Do not use external WEN (write enable).
		1	Use external WEN (write enable).
4	FLDPOL		Field indicator polarity.
		0	Positive
		1	Negative
3	HDPOL		HD sync polarity.
		0	Positive
		1	Negative
2	VDPOL		VD sync polarity.
		0	Positive
		1	Negative
1	FLDOUT		Field ID direction.
		0	Input
		1	Output
0	VDHDOUT		VD/HD sync direction.
		0	Input
		1	Output

6.1.4 HD and VD Signal Width Register (HD_VD_WID)

The HD and VD signal width register (HD_VD_WID) is shown in Figure 72 and described in Table 60.

Figure 72. HD and VD Signal Width Register (HD_VD_WID)

31 28	27 16
Reserved	HDW
R-0	R/W-0

15 12	11 0
Reserved	VDW
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 60. HD and VD Signal Width Register (HD_VD_WID) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	HDW	0-FFFh	Width of HD sync pulse, if output. HDW + 1 pixel clocks. HDW is not used when HD is an input (when VDHDOUT bit in SYN_MODE is cleared to 0). This bit field is latched by VD.
15-12	Reserved	0	Reserved
11-0	VDW	0-FFFh	Width of VD sync pulse, if output. VDW + 1 lines. VDW is not used when VD is an input (when VDHDOUT bit in SYN_MODE is cleared to 0). This bit field is latched by VD.

6.1.5 Number of Pixels in a Horizontal Line and Number of Lines in a Frame Register (PIX_LINES)

The number of pixels in a horizontal line and number of lines in a frame register (PIX_LINES) is shown in Figure 73 and described in Table 61.

Figure 73. Number of Pixels in a Horizontal Line and Number of Lines in a Frame Register (PIX_LINES)

31 16
PPLN
R/W-0

15 0
HLPFR
R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 61. Number of Pixels in a Horizontal Line and Number of Lines in a Frame Register (PIX_LINES) Field Descriptions

Bit	Field	Value	Description
31-16	PPLN	0-FFFFh	Pixels per line. Number of pixel clock periods in one line. HD period = PPLN + 1 pixel clocks. PPLN is not used when HD and VD are inputs (when VDHDOUT bit in SYN_MODE is cleared to 0). This bit field is latched by VD.
15-0	HLPFR	0-FFFFh	Half lines per field or frame. Sets number of half lines per frame or field. VD period = (HLPFR + 1)/2 lines. HLPFR is not used when HD is an input (when VDHDOUT bit in SYN_MODE is cleared to 0). This bit field is latched by VD.

6.1.6 Horizontal Pixel Information (HORZ_INFO)

The horizontal pixel information (HORZ_INFO) is shown in Figure 74 and described in Table 62.

Note: The CCD controller outputs the XY code in the SAV and EAV into the SDRAM. In order to eliminate this, set the SPH field to +1. Also set the NPH field to accurately represent the number of active pixels.

Figure 74. Horizontal Pixel Information (HORZ_INFO)

31	30 - 16
Reserved	SPH
R-0	R/W-0

15	14 - 0
Reserved	NPH
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 62. Horizontal Pixel Information (HORZ_INFO) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-16	SPH	0-7FFFh	Start pixel, horizontal. Sets pixel clock position at which data output to SDRAM begins, measured from the start of HD. This bit field is latched by VD.
15	Reserved	0	Reserved
14-0	NPH	0-7FFFh	Number of pixels, horizontal. Sets number of horizontal pixels that are output to SDRAM = NPH + 1. This bit field is latched by VD.

6.1.7 Vertical Line—Settings for the Starting Pixel (VERT_START)

The vertical line—settings for the starting pixel (VERT_START) is shown in Figure 75 and described in Table 63.

Figure 75. Vertical Line—Settings for the Starting Pixel (VERT_START)

31	30 16
Reserved	SLV0
R-0	R/W-0

15	14 0
Reserved	SLV1
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 63. Vertical Line—Settings for the Starting Pixel (VERT_START) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-16	SLV0	0-7FFFh	Start line, vertical (field 0). Sets line at which data output to SDRAM begins, measured from the start of VD. This bit field is latched by VD.
15	Reserved	0	Reserved
14-0	SLV1	0-7FFFh	Start line, vertical (field 1). Sets line at which data output to SDRAM begins, measured from the start of VD. This bit field is latched by VD.

6.1.8 Number of Vertical Lines (VERT_LINES)

The number of vertical lines (VERT_LINES) is shown in Figure 76 and described in Table 64.

Figure 76. Number of Vertical Lines (VERT_LINES)

31 16
Reserved
R-0

15	14 0
Reserved	NLV
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 64. Number of Vertical Lines (VERT_LINES) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved
14-0	NLV	0-7FFFh	Sets number of vertical lines that are output to SDRAM for each field. The number of lines output to SDRAM for each field = NLV + 1. This bit field is latched by VD.

6.1.9 Culling Information in Horizontal and Vertical Directions (CULLING)

The culling information in horizontal and vertical directions (CULLING) is shown in Figure 77 and described in Table 65.

Figure 77. Culling Information in Horizontal and Vertical Directions (CULLING)

31 24	23 16
CULHEVN	CULHODD
R/W-FFh	R/W-FFh

15 8	7 0
Reserved	CULV
R-0	R/W-FFh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 65. Culling Information in Horizontal and Vertical Directions (CULLING) Field Descriptions

Bit	Field	Value	Description
31-24	CULHEVN	0-FFh	Horizontal culling pattern for even line, 8-bit mask. When 0, cull; when 1, retain. LSB is first pixel and MSB is 8th pixel, then pattern repeats. This bit field is latched by VD.
23-16	CULHODD	0-FFh	Horizontal culling pattern for odd line, 8-bit mask. When 0, cull; when 1retain. LSB is first pixel and MSB is 8th pixel, then pattern repeats This bit field is latched by VD.
15-8	Reserved	0	Reserved
7-0	CULV	0-FFh	Vertical culling pattern, 8-bit mask. When 0, cull; when 1, retain. LSB is first line and MSB is 8th line, then pattern repeats. This bit field is latched by VD.

6.1.10 Horizontal Size (HSIZE_OFF)

The horizontal size (HSIZE_OFF) is shown in Figure 78 and described in Table 66.

Figure 78. Horizontal Size (HSIZE_OFF)

31 16
Reserved
R-0

15 0
LNOFST
R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 66. Horizontal Size (HSIZE_OFF) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	LNOFST	0-FFFFh	Address offset for each line. Sets offset for each output line in SDRAM. Either 16 or 32 pixels, depending on setting of the PACK8 bit in SYN_MODE. This bit field is latched by VD. The 5 LSBs are ignored; the offset is on a 32-byte boundary. For optimal performance in the system, the address offset should be on a 256-byte boundary.

6.1.11 SDRAM/DDRAM Line Offset Register (SDOFST)

The SDRAM/DDRAM line offset register (SDOFST) is shown in Figure 79 and described in Table 67.

Figure 79. SDRAM/DDRAM Line Offset Register (SDOFST)

31 … 16
Reserved
R-0

15	14	13 … 12	11 … 9	8 … 6	5 … 3	2 … 0
Rsvd	FIINV	FOFST	LOFTS0	LOFTS1	LOFTS2	LOFTS3
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 67. SDRAM/DDRAM Line Offset Register (SDOFST) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved
14	FIINV		Field identification signal inverse. This field is latched by VD.
		0	Non-inverse
		1	Inverse
13-12	FOFST	0-3h	Field line offset by number of lines ID = 1. This field is latched by VD.
		0	+1 line
		1h	+2 line
		2h	+3 line
		3h	+4 line
11-9	LOFTS0	0-7h	Line offset values of even line and even field ID = 0. This field is latched by VD.
		0	+1 line
		1h	+2 lines
		2h	+3 lines
		3h	+4 lines
		4h	-1 line
		5h	-2 lines
		6h	-3 lines
		7h	-4 lines
8-6	LOFTS1	0-7h	Line offset values of odd line and even field ID = 0. This field is latched by VD.
		0	+1 line
		1h	+2 lines
		2h	+3 lines
		3h	+4 lines
		4h	-1 line
		5h	-2 lines
		6h	-3 lines
		7h	-4 lines

Table 67. SDRAM/DDRAM Line Offset Register (SDOFST) Field Descriptions (continued)

Bit	Field	Value	Description
5-3	LOFTS2	0-7h	Line offset values of even line and odd field ID = 1. This field is latched by VD.
		0	+1 line
		1h	+2 lines
		2h	+3 lines
		3h	+4 lines
		4h	-1 line
		5h	-2 lines
		6h	-3 lines
		7h	-4 lines
2-0	LOFTS3	0-7h	Line offset values of odd line and odd field ID = 1. This field is latched by VD.
		0	+1 line
		1h	+2 lines
		2h	+3 lines
		3h	+4 lines
		4h	-1 line
		5h	-2 lines
		6h	-3 lines
		7h	-4 lines

6.1.12 SDRAM Address Register (SDR_ADDR)

The SDRAM address register (SDR_ADDR) is shown in Figure 80 and described in Table 68.

Figure 80. SDRAM Address Register (SDR_ADDR)

31	0
SDR_ADDR	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 68. SDRAM Address Register (SDR_ADDR) Field Descriptions

Bit	Field	Value	Description
31-0	SDR_ADDR	0-FFFF FFFFh	32-bit SDRAM starting address for CCD controller output. This bit field is latched by VD. Note that the address should be aligned on a 32-byte boundary; the 5 LSBs are ignored. Reading this register always shows the 5 LSBs cleared to 0. For optimal performance in the system, the address should be on a 256-byte boundary.

6.1.13 Optical Black Clamping Settings Register (CLAMP)

The optical black clamping settings register (CLAMP) is shown in Figure 81 and described in Table 69.

Figure 81. Optical Black Clamping Settings Register (CLAMP)

31	30 28	27 25	24 16
CLAMPEN	OBSLEN	OBSLN	OBST
R/W-0	R/W-0	R/W-0	R/W-0

15 10	9 5	4 0
OBST	Reserved	OBGAIN
R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 69. Optical Black Clamping Settings Register (CLAMP) Field Descriptions

Bit	Field	Value	Description
31	CLAMPEN		Clamp enable. Enables clamping of CCD data based on the calculated average of Optical Black Samples. This bit is latched by VD.
		0	Disable
		1	Enable
30-28	OBSLEN	0-7h	Optical Black Sample Length. Number of Optical Black Sample pixels per line to include in the average calculation.
		0	1 pixel
		1h	2 pixels
		2h	4 pixels
		3h	8 pixels
		4h	16 pixels
		5h-7h	Reserved
27-25	OBSLN	0-7h	Optical Black Sample Lines. 5-7: Reserved. Number of Optical Black Sample lines to include in the average calculation.
		0	1 line
		1h	2 lines
		2h	4 lines
		3h	8 lines
		4h	16 lines
		5h-7h	Reserved
24-10	OBST	0-7FFFh	Start Pixel of Optical Black Samples. Start pixel position of Optical Black Samples, specified from the start of HD in pixel clocks.
9-5	Reserved	0	Reserved
4-0	OBGAIN	0-1Fh	Gain to apply to the optical black average. Multiply the optical black average with the specified gain (range from 0 to ~2 - U5Q4)

6.1.14 DC Clamp Register (DCSUB)

The DC clamp register (DCSUB) is shown in Figure 82 and described in Table 70.

Note: This function does not clip negative results to 0 for YUV 8-bit input mode (SYN_MODE.INPMOD = 2h) or REC656 input mode (REC656IF.REC656ON = 1).

Figure 82. DC Clamp Register (DCSUB)

31–16
Reserved
R-0

15–14	13–0
Reserved	DCSUB
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 70. DC Clamp Register (DCSUB) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved
13-0	DCSUB	0-3FFFh	DC level to subtract from CCD data. The DC value is subtracted from the CCD data when OBS clamping is disabled (when CLAMPEN bit in CLAMP is cleared to 0).

6.1.15 CCD Color Pattern Register (COLPTN)

The CCD color pattern register (COLPTN) is shown in Figure 83 and described in Table 71.

Figure 83. CCD Color Pattern Register (COLPTN)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CP3LPC3	CP3LPC2	CP3LPC1	CP3LPC0	CP2LPC3	CP2LPC2	CP2LPC1	CP2LPC0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CP1LPC3	CP1LPC2	CP1LPC1	CP1LPC0	CP0LPC3	CP0LPC2	CP0LPC1	CP0LPC0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 71. CCD Color Pattern Register (COLPTN) Field Descriptions

Bit	Field	Value	Description
31-30	CP3LPC3	0-3h	Color pattern for 3rd line. pixel counter = 3
		0	R
		1h	Gr
		2h	Gb
		3h	B
29-28	CP3LPC2	0-3h	Color pattern for 3rd line, pixel counter = 2
27-26	CP3LPC1	0-3h	Color pattern for 3rd line, pixel counter = 1
25-24	CP3LPC0	0-3h	Color pattern for 3rd line, pixel counter = 0
23-22	CP2LPC3	0-3h	Color pattern for 2nd line, pixel counter = 3
21-20	CP2LPC2	0-3h	Color pattern for 2nd line, pixel counter = 2
19-18	CP2LPC1	0-3h	Color pattern for 2nd line, pixel counter = 1
17-16	CP2LPC0	0-3h	Color pattern for 2nd line, pixel counter = 0
15-14	CP1LPC3	0-3h	Color pattern for 1st line, pixel counter = 3
13-12	CP1LPC2	0-3h	Color pattern for 1st line, pixel counter = 2
11-10	CP1LPC1	0-3h	Color pattern for 1st line, pixel counter = 1
9-8	CP1LPC0	0-3h	Color pattern for 1st line, pixel counter = 0
7-6	CP0LPC3	0-3h	Color pattern for 0th line, pixel counter = 3
5-4	CP0LPC2	0-3h	Color pattern for 0th line, pixel counter = 2
3-2	CP0LPC1	0-3h	Color pattern for 0th line, pixel counter = 1
1-0	CP0LPC0	0-3h	Color pattern for 0th line, pixel counter = 0

6.1.16 Black Compensation Register (BLKCMP)

The black compensation register (BLKCMP) is shown in Figure 84 and described in Table 72.

Figure 84. Black Compensation Register (BLKCMP)

31–24	23–16
R	GR
R/W-0	R/W-0

15–8	7–0
GB	B
R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 72. Black Compensation Register (BLKCMP) Field Descriptions

Bit	Field	Value	Description
31-24	R	0-FFh	Black level compensation for R pixels (-128 to +127). 2's complement, MSB is sign bit.
23-16	GR	0-FFh	Black level compensation for Gr pixels (-128 to +127). 2's complement, MSB is sign bit.
15-8	GB	0-FFh	Black level compensation for Gb pixels (-128 to +127). 2's complement, MSB is sign bit.
7-0	B	0-FFh	Black level compensation for B pixels (-128 to +127). 2's complement, MSB is sign bit.

6.1.17 Fault Pixel Correction Register (FPC)

The fault pixel correction register (FPC) is shown in Figure 85 and described in Table 73.

Figure 85. Fault Pixel Correction Register (FPC)

31	17	16
Reserved		FPERR
R-0		R/W-0

15	14	0
FPCEN	FPNUM	
R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 73. Fault Pixel Correction Register (FPC) Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved.
16	FPERR		Fault pixel correction error. Write 1 to clear the error. This bit is set when the CCDC is unable to fetch the required fault pixel table entry in time for it to correct it. For example, the current pixel being processed has coordinates of 256/512 (256th line and 512th pixel in that line) and it needs to be corrected. If the entry in the fault pixel table that needs to be used has coordinates 256/256, then the current pixel cannot be corrected since the correct entry is not loaded in time. There is no error recovery mechanism in the CCDC. If this bit is set at anytime in a frame, there will be no more fault pixels corrected in that frame. Firmware is responsible for making sure that there is enough bandwidth in the system to allow for loading of the fault pixel table. Alternately, decreasing the frequency of the fault pixels to be corrected will enhance chances of this bit not being set.
		0	No error.
		1	Error.
15	FPCEN		Fault pixel correction enable. Upon setting this bit and as long as it remains enabled, the fault pixel logic will continue to request data and just start over once the last data has been received. As soon as the register is set, the data will be fetched. Write(FPC_ADDR). Write(FPC) with this bit (FPC.FPCEN) turned off. Write(FPC) with this bit (FPC.FPCEN) turned on while other fields are same as previous write.
		0	Disabled
		1	Enabled
14-0	FPNUM	0-7FFFh	Number of fault pixels to be corrected in the frame. This field is latched by VD.

6.1.18 Fault Pixel Correction SDRAM Address Register (FPC_ADDR)

The fault pixel correction SDRAM address register (FPC_ADDR) is shown in Figure 86 and described in Table 74.

Figure 86. Fault Pixel Correction SDRAM Address Register (FPC_ADDR)

31 ... 0
FPC_ADDR
R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 74. Fault Pixel Correction SDRAM Address Register (FPC_ADDR) Field Descriptions

Bit	Field	Value	Description
31-0	FPC_ADDR	0-FFFF FFFFh	Starting SDRAM/DDRAM address of the fault pixel correction entries. The address should be aligned to a 64-byte boundary, therefore, the 6 LSB's are ignored. Each of the 32-bit fault memory entry contains a 13-bit vertical position, a 14-bit horizontal position, and a 5-bit operation field. The following operations are decoded by the 5-bit operation field (4:0) Note that x denotes the horizontal coordinate.
		0	Replace pixel at x with the average of x - 1 and x + 1.
		1	Replace pixel at x with the x - 1.
		2h	Replace pixel at x with the x + 1.
		3h	Replace pixel at x with the average of x - 2 and x + 2.
		4h	Replace pixel at x with the x - 2.
		5h	Replace pixel at x with the x + 2.
		6h	Replace pixel at x with the average of x - 3 and x + 3.
		7h	Replace pixel at x with the x - 3.
		8h	Replace pixel at x with the x + 3.
		9h	Replace pixel at x with the average of x - 4 and x + 4.
		10h	Replace pixel at x with the x - 4.
		11h	Replace pixel at x with the x + 4.
		12h	Replace pixel at x with the average of x - 5 and x + 5.
		13h	Replace pixel at x with the x - 5.
		14h	Replace pixel at x with the x + 5.
		15h	Replace pixel at x with the average of x - 6 and x + 6.
		16h	Replace pixel at x with the x - 6.
		17h	Replace pixel at x with the x + 6.
		18-31h	Reserved.

6.1.19 VD Interrupt Timing Register (VDINT)

The VD interrupt timing register (VDINT) is shown in Figure 87 and described in Table 75.

Figure 87. VD Interrupt Timing Register (VDINT)

31	30	16
Reserved	VDINT0	
R-0	R/W-0	

15	14	0
Reserved	VDINT1	
R-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 75. VD Interrupt Timing Register (VDINT) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-16	VDINT0	0-7FFFh	VD0 interrupt timing. Specify VDINT0 in units of horizontal lines from the start of VD pulse. Resulting value is VDINT0 + 1. Note that if the rising edge (or falling edge, if programmed) of the HD lines up with the rising edge (or falling edge, if programmed) of VD, the first HD is not counted.
15	Reserved	0	Reserved
14-0	VDINT1	0-7FFFh	VD1 interrupt timing. Specify VDINT1 in units of horizontal lines from the start of VD pulse. Resulting value is VDINT1 + 1. Note that if the rising edge (or falling edge, if programmed) of the HD lines up with the rising edge (or falling edge, if programmed) of VD, the first HD is not counted.

6.1.20 A-Law Setting Register (ALAW)

The A-law setting register (ALAW) is shown in Figure 88 and described in Table 76.

Figure 88. A-Law Setting Register (ALAW)

31	16
Reserved	
R-0	

15	4	3	2	0
Reserved		CCDTBL	GWID	
R-0		R/W-0	R/W-4h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 76. A-Law Setting Register (ALAW) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	CCDTBL		Apply Gamma (A-law) to CCD controller data saved to SDRAM.
		0	Disable
		1	Enable
2-0	GWID	0-7h	Gamma width input (A-law table).
		0	Bits 15-6
		1h	Bits 14-5
		2h	Bits 13-4
		3h	Bits 12-3
		4h	Bits 11-2
		5h	Bits 10-1
		6h	Bits 9-0
		7h	Reserved

6.1.21 REC656 Interface Register (REC656IF)

The REC656 interface register (REC656IF) is shown in Figure 89 and described in Table 77.

Figure 89. REC656 Interface Register (REC656IF)

31 16
Reserved
R-0

15 2	1	0
Reserved	ECCFVH	R656ON
R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 77. REC656 Interface Register (REC656IF) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	ECCFVH		FVH error correction enable.
		0	Disable
		1	Enable
0	R656ON		REC656 interface enable.
		0	Disable
		1	Enable

6.1.22 CCD Configuration Register (CCDCFG)

The CCD configuration register (CCDCFG) is shown in Figure 90 and described in Table 78.

Note: CCDCFG.VDLC must be set to 1 by software if the CCD controller is to be used. If CCDCFG.VDLC remains cleared to 0 (default), indeterminate results may occur for any register access in the CCD controller. See Section 5.4.4 for more details.

Figure 90. CCD Configuration Register (CCDCFG)

31 16
Reserved
R-0

15	14	13	12	11	10 9	8
VDLC	Reserved	MSBINVI	BSWD	Y8POS	Reserved	WENLOG
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

7 6	5	4	3 0
FIDMD	BW656	YCINSWP	Reserved
R/W-0	R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 78. CCD Configuration Register (CCDCFG) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	VDLC		Enable latching function registers on internal VSYNC.
		0	Latched on VSYNC.
		1	Not latched on VSYNC.
14	Reserved	0	Reserved
13	MSBINVI		MSB of Chroma input signal stored to SDRAM inverted.
		0	Normal
		1	MSB inverted
12	BSWD		Byte swap data stored to SDRAM. Number of pixels must be even, if byte packing is enabled (and this field is set to 1).
		0	Normal
		1	Swap bytes
11	Y8POS		Location of Y signal when YCbCr 8-bit data is input.
		0	Even pixel
		1	Odd pixel
10-9	Reserved	0	Reserved
8	WENLOG		Specifies CCD valid area.
		0	Internal valid signal and WEN signal are logical-ANDed.
		1	Internal valid signal and WEN signal are logical-ORed.
7-6	FIDMD	0-3h	Setting of FID detection function.
		0	FID signal is latched at the VSYNC timing.
		1h	FID signal is not latched.
		2h	FID signal is latched at edge of VD.
		3h	FID signal is latched based on phase of VD and HD.
5	BW656		Data width in CCIR656 input mode. If 656 mode is enabled, takes precedence over the INPMOD and DATSIZ bits in the sync and mode set register (SYN_MODE).
		0	8 bits
		1	10 bits
4	YCINSWP		Y input (YIN[7:0]) and C input (CIN[7:0]) swapped. This field is latched on the VSYNC/VD signal. This swaps the luma and chroma samples in 16-bit YUV mode. Swapping portions of the 16-bit YUV data bus determines which half of the bus is used as the input source in 8-bit mode and can be used in 8-bit YUV mode to support two separate YUV input ports. It cannot be used in REC656 mode.
		0	No swap. YIN[7:0] = Y signal/CIN[7:0] = C signal
		1	Swap. YIN[7:0] = C signal/CIN[7:0] = Y signal
3-0	Reserved	0	Reserved

6.1.23 Data Reformatter/Video Port Configuration Register (FMTCFG)

The data reformatter/video port configuration register (FMTCFG) is shown in Figure 91 and described in Table 79.

Figure 91. Data Reformatter/Video Port Configuration Register (FMTCFG)

31 - 19	18 - 16
Reserved	VPIF_FRQ
R-0	R/W-0

15	14 - 12	11 - 0
VPEN	VPIN	Reserved
R/W-0	R/W-4h	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 79. Data Reformatter/Video Port Configuration Register (FMTCFG) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reserved
18-16	VPIF_FRQ	0-7h	Video port data ready frequency. This field allows the firmware to control the rate at which the video port delivers new data to the other modules (preview engine, H3A, and histogram). In effect, this register controls the raw output bandwidth of the preview engine, H3A, and histogram. Depending on the input sensor clock, you can set this field appropriately and balance the bandwidth requirements to SDRAM. If the CCD controller operating clock is SYSCLK/3 (depends on the specific chip, see the chip level clocking scheme), then the following settings apply: • 1/2: Sensor cannot have pixel clock greater than (SYSCLK/3)/2 = 75 MHZ. • 1/3.5: Sensor cannot have pixel clock greater than (SYSCLK/3)/3.5 = 42.8 MHZ. • 1/4.5: Sensor cannot have pixel clock greater than (SYSCLK/3)/4.5 = 33.3 MHZ. • 1/5.5: Sensor cannot have pixel clock greater than (SYSCLK/3)/5.5 = 27.2 MHZ. • 1/6.5: Sensor cannot have pixel clock greater than (SYSCLK/3)/6.5 = 23 MHZ. Depending on the sensor speed, it is advised to apply the highest possible divisor value in this field.
		0	1/2 (one half)
		1h	1/3.5 (one thirdhalf)
		2h	1/4.5 (one fourthhalf)
		3h	1/5.5 (one fifthhalf)
		4h	1/6.5 (one sixthhalf)
		5h-7h	Reserved
15	VPEN		Video port enable
		0	Disable
		1	Enable
14-12	VPIN	0-7h	Video port input select (10-bit input)
		0	Bits 15-6
		1h	Bits 14-5
		2h	Bits 13-4
		3h	Bits 12-3
		4h	Bits 11-2
		5h	Bits 10-1
		6h	Bits 9-0
		7h	Reserved
11-0	Reserved	0	Reserved

6.1.24 Data Reformatter/Video Input Interface Horizontal Information Register (FMT_HORZ)

The data reformatter/video input interface horizontal information register (FMT_HORZ) is shown in Figure 92 and described in Table 80.

Figure 92. Data Reformatter/Video Input Interface Horizontal Information Register (FMT_HORZ)

31 29	28 16
Reserved	FMTSPH
R-0	R/W-0

15 13	12 0
Reserved	FMTLNH
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 80. Data Reformatter/Video Input Interface Horizontal Information Register (FMT_HORZ) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28-16	FMTSPH	0-1FFFh	Start pixel horizontal from the start of HD.
15-13	Reserved	0	Reserved
12-0	FMTLNH	0-1FFFh	Number of pixels in the horizontal direction.

6.1.25 Data Reformatter/Video Input Interface Vertical Information Register (FMT_VERT)

The data reformatter/video input interface vertical information register (FMT_VERT) is shown in Figure 93 and described in Table 81.

Figure 93. Data Reformatter/Video Input Interface Vertical Information Register (FMT_VERT)

31 29	28 16
Reserved	FMTSLV
R-0	R/W-0

15 13	12 0
Reserved	FMTLNV
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 81. Data Reformatter/Video Input Interface Vertical Information Register (FMT_VERT) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28-16	FMTSLV	0-1FFFh	Start line from the start of VD.
15-13	Reserved	0	Reserved
12-0	FMTLNV	0-1FFFh	Number of lines in the vertical direction.

6.1.26 Video Port Output Settings Register (VP_OUT)

The video port output settings register (VP_OUT) is shown in Figure 94 and described in Table 82.

Figure 94. Video Port Output Settings Register (VP_OUT)

31	30	17	16
Reserved	VERT_NUM		HORZ_NUM
R-0	R/W-0		R/W-0

15	4	3	0
HORZ_NUM		HORZ_ST	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 82. Video Port Output Settings Register (VP_OUT) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-17	VERT_NUM	0-3FFFh	Number of vertical lines to clock out the video port. The video port output is the input to the preview engine, H3A, and histogram modules (if selected for each module). The number of lines that can be clocked out of the video port should be at least 1 line less than the number of lines input from the sensor. The video port output VSYNC is generated right from the first video port input VSYNC itself.
16-4	HORZ_NUM	0-1FFFh	Number of horizontal pixels to clock out the video port.
3-0	HORZ_ST	0-Fh	Horizontal start pixel in each output line. Maximum offset allowed is 15, the video port output HSYNC is generated from this position for each line. In order to be able to select an offset higher than 15, the input settings to the data reformatter should be configured appropriately. The purpose of this parameter is to allow for sensors that can read out a parallelogram image rather than a rectangular image.

6.2 Preview Engine (PREV) Registers

Table 83 lists the memory-mapped registers for the preview engine (PREV). See the device-specific data manual for the memory address of these registers.

Table 83. Preview Engine (PREV) Registers

Offset	Register	Description	Sections
800h	PID	Peripheral revision and class information register	Section 6.2.1
804h	PCR	Peripheral control register	Section 6.2.2
808h	HORZ_INFO	Horizontal information/setup register	Section 6.2.3
80Ch	VERT_INFO	Vertical information/setup register	Section 6.2.4
810h	RSDR_ADDR	Read address from SDRAM register	Section 6.2.5
814h	RADR_OFFSET	Line offset for the read data register	Section 6.2.5
818h	DSDR_ADDR	Dark frame address from SDRAM register	Section 6.2.7
81Ch	DRKF_OFFSET	Line offset for the dark frame data register	Section 6.2.8
820h	WSDR_ADDR	Write address to SDRAM register	Section 6.2.9
824h	WADD_OFFSET	Line offset for the write data register	Section 6.2.10
828h	AVE	Input formatter/averager register	Section 6.2.11
82Ch	HMED	Horizontal median filter register	Section 6.2.12
830h	NF	Noise filter register	Section 6.2.13
834h	WB_DGAIN	White balance digital gain register	Section 6.2.14
838h	WBGAIN	White balance coefficients register	Section 6.2.15
83Ch	WBSEL	White balance coefficients selection register	Section 6.2.16
840h	CFA	CFA register	Section 6.2.17
844h	BLKADJOFF	Black adjustment offset register	Section 6.2.18
848h	RGB_MAT1	RGB2RGB blending matrix coefficients register	Section 6.2.19
84Ch	RGB_MAT2	RGB2RGB blending matrix coefficients register	Section 6.2.20
850h	RGB_MAT3	RGB2RGB blending matrix coefficients register	Section 6.2.21
854h	RGB_MAT4	RGB2RGB blending matrix coefficients register	Section 6.2.22
858h	RGB_MAT5	RGB2RGB blending matrix coefficients register	Section 6.2.23
85Ch	RGB_OFF1	RGB2RGB blending matrix offsets register	Section 6.2.24
860h	RGB_OFF2	RGB2RGB blending matrix offsets register	Section 6.2.25
864h	CSC0	Color space conversion coefficients register	Section 6.2.26
868h	CSC1	Color space conversion coefficients register	Section 6.2.27
86Ch	CSC2	Color space conversion coefficients register	Section 6.2.28
870h	CSC_OFFSET	Color space conversion offsets register	Section 6.2.29
874h	CNT_BRT	Contrast and brightness settings register	Section 6.2.30
878h	CSUP	Chrominance supression settings register	Section 6.2.31
87Ch	SETUP_YC	Maximum/minimum Y and C settings register	Section 6.2.32
880h	SET_TBL_ADDRESS	Setup table addresses register	Section 6.2.33
884h	SET_TBL_DATA	Setup table data register	Section 6.2.34

6.2.1 Peripheral Revision and Class Information Register (PID)

The peripheral revision and class information register (PID) is shown in Figure 95 and described in Table 84.

Figure 95. Peripheral Revision and Class Information Register (PID)

31 24	23 16
Reserved	TID
R-0	R-2h

15 8	7 0
CID	PREV
R-FEh	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 84. Peripheral Revision and Class Information Register (PID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	TID		Peripheral identification
		2h	Preview engine
15-8	CID		Class identification
		FEh	Video processing front end module
7-0	PREV		Peripheral revision number
		0	Current revision

6.2.2 Peripheral Control Register (PCR)

The peripheral control register (PCR) is shown in Figure 96 and described in Table 85.

Figure 96. Peripheral Control Register (PCR)

31	27	26	25	24
Reserved		GAMMA_BYPASS	Reserved	SHADE_SFT
R-0		R/W-0	R-0	R/W-0

23	22	21	20	19	18	17	16
SHADE_SFT		Reserved	SDRPORT	RSZPORT	YCPOS		SUPEN
R/W-0		R-0	R/W-1	R/W-0	R/W-0		R/W-0

15	14	11	10	9	8
YNENHEN	Reserved		CFAEN	NFEN	HMEDEN
R/W-0	R-0		R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
DRKFCAP	DRKFEN	INVALAW	WIDTH	ONESHOT	SOURCE	BUSY	ENABLE
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 85. Peripheral Control Register (PCR) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Reserved
26	GAMMA_BYPASS		Gamma table bypass.
		0	Gamma tables not bypassed
		1	Gamma tables are bypassed (output is 8 MSB or 10-bit input)
25	Reserved	0	Reserved
24-22	SHADE_SFT	0-7h	Shading compensation right shift after multiplication.
		0	Disable
		1	Enable
21	Reserved	0	Reserved
20	SDRPORT		SDRAM port for the output of preview.
		0	Disable
		1	Enable
19	RSZPORT		Resizer port enable (for input to resizer). Controls if SDRAM output data is forwarded to the resizer input port. This control bit does not depend on the state of the SDRPORT bit. Data is simultaneously written to SDRAM (if SDRPORT bit is set) while sending the same data to the resizer as input. Note that the CCD controller is also capable of directly writing to the resizer input port. The CCD controller setting takes precedence over the preview engine setting.
		0	Disable
		1	Enable
18-17	YCPOS	0-3h	YC position
		0	YCrYCb
		1h	YCbYCr
		2h	CbYCrY
		3h	CrYCbY
16	SUPEN		Color suppression
		0	Disable
		1	Enable

Table 85. Peripheral Control Register (PCR) Field Descriptions (continued)

Bit	Field	Value	Description
15	YNENHEN		Non-linear enhancer
		0	Disable
		1	Enable
14-11	Reserved	0	Reserved
10	CFAEN		CFA enable
		0	Disable
		1	Enable
9	NFEN		Noise filter (no register impact)
		0	Disable
		1	Enable
8	HMEDEN		Horizontal median filter (no register impact)
		0	Disable
		1	Enable
7	DRKFCAP		Dark frame capture
		0	Normal processing
		1	Capture dark frame
6	DRKFEN		Subtract dark frame
		0	Disable
		1	Enable
5	INVALAW		Inverse A-law
		0	Disable
		1	Enable
4	WIDTH		Input data width
		0	10-bit wide
		1	8-bit wide
3	ONESHOT		One-shot mode. If this bit is set to 1, it is reset to 0 after the ENABLE bit is asserted.
		0	Continuous mode (via the video port)
		1	One-shot mode
2	SOURCE		Input source. If this bit is set to 1, it is reset to 0 after the ENABLE bit is asserted.
		0	Video port (via the CCD controller)
		1	SDRAM
1	BUSY		Preview engine busy bit
		0	Preview engine is not busy.
		1	Preview engine is busy.
0	ENABLE		Preview engine enable bit. When the ENABLE bit is asserted, the ONESHOT and SOURCE bits, if set to 1, are reset to 0.
		0	Preview engine is disabled.
		1	Preview engine is enabled.

6.2.3 Horizontal Information/Setup Register (HORZ_INFO)

The horizontal information/setup register (HORZ_INFO) is shown in Figure 97 and described in Table 86.

Note: The CCD controller outputs the XY code in the SAV and EAV into the SDRAM. In order to eliminate this, set the SPH field to +1.

Figure 97. Horizontal Information/Setup Register (HORZ_INFO)

31 30	29 16
Reserved	SPH
R-0	R/W-0

15 14	13 0
Reserved	EPH
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 86. Horizontal Information/Setup Register (HORZ_INFO) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-16	SPH	0-3FFFh	Start pixel horizontal. If the SOURCE bit in the peripheral control register (PCR) is cleared to 0 (CCDC input), SPH must be ≥ 2.
15-14	Reserved	0	Reserved
13-0	EPH	0-3FFFh	End pixel horizontal. The input width of the preview engine must be a multiple of the average count multiplied by the least common multiple of the odd distance and even distance of the settings in the input formatter/averager register (AVE). If the SOURCE bit in the peripheral control register (PCR) is cleared to 0 (CCDC input), EPH must be ≥ 2 pixels before the last pixel is sent from the CCD controller.

6.2.4 Vertical Information/Setup Register (VERT_INFO)

The vertical information/setup register (VERT_INFO) is shown in Figure 98 and described in Table 87.

Figure 98. Vertical Information/Setup Register (VERT_INFO)

31 30	29 16
Reserved	SLV
R-0	R/W-0

15 14	13 0
Reserved	ELV
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 87. Vertical Information/Setup Register (VERT_INFO) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-16	SLV	0-3FFFh	Start line vertical
15-14	Reserved	0	Reserved
13-0	ELV	0-3FFFh	Start line vertical

6.2.5 Read Address From SDRAM Register (RSDR_ADDR)

The read address from SDRAM register (RSDR_ADDR) is shown in Figure 99 and described in Table 88.

Figure 99. Read Address From SDRAM Register (RSDR_ADDR)

31 0
RADR
R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 88. Read Address From SDRAM Register (RSDR_ADDR) Field Descriptions

Bit	Field	Value	Description
31-0	RADR	0-FFFF FFFFh	Read Address. Specifies the initial address of the frame that is to be fetched from SDRAM. The lower 5 bits of this register are always treated as 0 (the starting address should be 32-byte aligned). RADR can be altered even when the preview engine is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

6.2.6 Line Offset for the Read Data Register (RADR_OFFSET)

The line offset for the read data register (RADR_OFFSET) is shown in Figure 100 and described in Table 89.

Figure 100. Line Offset for the Read Data Register (RADR_OFFSET)

31 16
Reserved
R-0

15 0
OFFSET
R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 89. Line Offset for the Read Data Register (RADR_OFFSET) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	OFFSET	0-FFFFh	Line offset. Specifies the offset for each line relative to the previous line. The lower 5 bits of this register are always treated as 0 (the offset should be 32-byte aligned). OFFSET can be altered even when the preview engine is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

6.2.7 Dark Frame Address From SDRAM Register (DSDR_ADDR)

The dark frame address from SDRAM register (DSDR_ADDR) is shown in Figure 101 and described in Table 90.

Figure 101. Dark Frame Address From SDRAM Register (DSDR_ADDR)

31 0
DRKF
R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 90. Dark Frame Address From SDRAM Register (DSDR_ADDR) Field Descriptions

Bit	Field	Value	Description
31-0	DRKF	0-FFFF FFFFh	Read address. Specifies the initial address of the dark frame that is to be fetched from SDRAM. The lower 5 bits of this register are always treated as 0 (the starting address should be 32-byte aligned). DRKF can be altered even when the preview engine is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

6.2.8 Line Offset for the Dark Frame Data Register (DRKF_OFFSET)

The line offset for the dark frame data register (DRKF_OFFSET) is shown in Figure 102 and described in Table 91.

Figure 102. Line Offset for the Dark Frame Data Register (DRKF_OFFSET)

31	16
Reserved	
R-0	

15	0
OFFSET	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 91. Line Offset for the Dark Frame Data Register (DRKF_OFFSET) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	OFFSET	0-FFFFh	Line offset. Specifies the offset for each line relative to the previous line. The lower 5 bits of this register are always treated as 0 (the offset should be 32-byte aligned). OFFSET can be altered even when the preview engine is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

6.2.9 Write Address to the SDRAM Register (WSDR_ADDR)

The write address to the SDRAM register (WSDR_ADDR) is shown in Figure 103 and described in Table 92.

Figure 103. Write Address to the SDRAM Register (WSDR_ADDR)

31	0
ADDR	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 92. Write Address to the SDRAM Register (WSDR_ADDR) Field Descriptions

Bit	Field	Value	Description
31-0	ADDR	0-FFFF FFFFh	Write address. Specifies the output address of the frame that is to be stored into SDRAM/DDRAM. The lower 5 bits of this register are always treated as 0 (the starting address should be 32-byte aligned). ADDR can be altered even when the preview engine is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

6.2.10 Line Offset for the Write Data Register (WADD_OFFSET)

The line offset for the write data register (WADD_OFFSET) is shown in Figure 104 and described in Table 93.

Figure 104. Line Offset for the Write Data Register (WADD_OFFSET)

31	16
Reserved	
R-0	

15	0
OFFSET	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 93. Line Offset for the Write Data Register (WADD_OFFSET) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	OFFSET	0-FFFFh	Line offset. Specifies the offset for each line relative to the previous line. The lower 5 bits of this register are always treated as 0 (the offset should be 32-byte aligned). OFFSET can be altered even when the preview engine is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

6.2.11 Input Formatter/Averager Register (AVE)

The input formatter/averager register (AVE) is shown in Figure 105 and described in Table 94.

Figure 105. Input Formatter/Averager Register (AVE)

31 16
Reserved
R-0

15 6	5 4	3 2	1 0
Reserved	ODDDIST	EVENDIST	COUNT
R-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 94. Input Formatter/Averager Register (AVE) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved	0	Reserved
5-4	ODDDIST	0-3h	Distance between consecutive pixels of same color in the odd line.
		0	1 pixel
		1h	2 pixels
		2h	3 pixels
		3h	4 pixels
3-2	EVENDIST	0-3h	Distance between consecutive pixels of same color in the even line.
		0	1 pixel
		1h	2 pixels
		2h	3 pixels
		3h	4 pixels
1-0	COUNT	0-3h	Number of horizontal pixels to average. COUNT should not be changed when the DRKFEN bit in the peripheral control register (PCR) is set to 1. The input width must be divisible by the number of horizontal pixels to average, indicated by COUNT (that is, if a 4 pixel average is selected, then the input width must be a multiple of 4).
		0	No average
		1h	2 pixels average
		2h	4 pixels average
		3h	8 pixels average

6.2.12 Horizontal Median Filter Register (HMED)

The Horizontal Median Filter Register (HMED) is shown in Figure 106 and described in Table 95.

Figure 106. Horizontal Median Filter Register (HMED)

31 16
Reserved
R-0

15 10	9	8	7 0
Reserved	ODDDIST	EVENDIST	THRESHOLD
R-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 95. Horizontal Median Filter Register (HMED) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9	ODDDIST		Distance between consecutive pixels of same color in the odd line.
		0	1 pixel
		1	2 pixels
8	EVENDIST		Distance between consecutive pixels of same color in the even line.
		0	1 pixel
		1	2 pixels
7-0	THRESHOLD	0-FFh	Horizontal median filter threshold.

6.2.13 Noise Filter Register (NF)

The noise filter register (NF) is shown in Figure 107 and described in Table 96.

Figure 107. Noise Filter Register (NF)

31 16
Reserved
R-0

15 4	3 0
Reserved	STRNGTH
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 96. Noise Filter Register (NF) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3-0	STRNGTH	0-Fh	Strength to use for the noise filter. x = ((x × (16 - STRNGTH)) + (ave × STRNGTH)) >> 4

6.2.14 White Balance Digital Gain Register (WB_DGAIN)

The white balance digital gain register (WB_DGAIN) is shown in Figure 108 and described in Table 97.

Figure 108. White Balance Digital Gain Register (WB_DGAIN)

31	16
Reserved	
R-0	

15	10	9	0
Reserved		DGAIN	
R-0		R/W-100h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 97. White Balance Digital Gain Register (WB_DGAIN) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	DGAIN	0-3FFh	Digital gain for the white balance. Can change anytime following the start of a frame.

6.2.15 White Balance Coefficients Register (WBGAIN)

The white balance coefficients register (WBGAIN) is shown in Figure 109 and described in Table 98.

Figure 109. White Balance Coefficients Register (WBGAIN)

31	24	23	16
COEF3		COEF2	
R/W-20h		R/W-20h	

15	8	7	0
COEF1		COEF0	
R/W-20h		R/W-20h	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 98. White Balance Coefficients Register (WBGAIN) Field Descriptions

Bit	Field	Value	Description
31-24	COEF3	0-FFh	White balance gain - COEF 3. Can change anytime following the start of a frame.
23-16	COEF2	0-FFh	White balance gain - COEF 2. Can change anytime following the start of a frame.
15-8	COEF1	0-FFh	White balance gain - COEF 1. Can change anytime following the start of a frame.
7-0	COEF0	0-FFh	White balance gain - COEF 0. Can change anytime following the start of a frame.

6.2.16 White Balance Coefficients Selection Register (WBSEL)

The white balance coefficients selection register (WBSEL) is shown in Figure 110 and described in Table 99.

Figure 110. White Balance Coefficients Selection Register (WBSEL)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
3_3	3_2	3_1	3_0	2_3	2_2	2_1	2_0
R/W-3h	R/W-2h	R/W-3h	R/W-2h	R/W-1	R/W-0	R/W-1	R/W-0

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
1_3	1_2	1_1	1_0	0_3	0_2	0_1	0_0
R/W-3h	R/W-2h	R/W-3h	R/W-2h	R/W-1	R/W-0	R/W-1	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 99. White Balance Coefficients Selection Register (WBSEL) Field Descriptions

Bit	Field	Value	Description
31-30	3_3	0-3h	Coefficient selection for 3rd line, 3rd pixel.
		0	COEF0
		1h	COEF1
		2h	COEF2
		3h	COEF3
29-28	3_2	0-3h	Coefficient selection for 3rd line, 2nd pixel.
27-26	3_1	0-3h	Coefficient selection for 3rd line, 1st pixel.
25-24	3_0	0-3h	Coefficient selection for 3rd line, 0th pixel.
23-22	2_3	0-3h	Coefficient selection for 2nd line, 3rd pixel.
21-20	2_2	0-3h	Coefficient selection for 2nd line, 2nd pixel.
19-18	2_1	0-3h	Coefficient selection for 2nd line, 1st pixel.
17-16	2_0	0-3h	Coefficient selection for 2nd line, 0th pixel.
15-14	1_3	0-3h	Coefficient selection for 1st line, 3rd pixel.
13-12	1_2	0-3h	Coefficient selection for 1st line, 2nd pixel.
11-10	1_1	0-3h	Coefficient selection for 1st line, 1st pixel.
9-8	1_0	0-3h	Coefficient selection for 1st line, 0th pixel.
7-6	0_3	0-3h	Coefficient selection for 0th line, 3rd pixel.
5-4	0_2	0-3h	Coefficient selection for 0th line, 2nd pixel.
3-2	0_1	0-3h	Coefficient selection for 0th line, 1st pixel.
1-0	0_0	0-3h	Coefficient selection for 0th line, 0th pixel.

6.2.17 CFA Register (CFA)

The CFA register (CFA) is shown in Figure 111 and described in Table 100.

Figure 111. CFA Register (CFA)

31 16
Reserved
R-0

15 8	7 0
GRADTH_VER	GRADTH_HOR
R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 100. CFA Register (CFA) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-8	GRADTH_VER	0-FFh	Gradient threshold vertical.
7-0	GRADTH_HOR	0-FFh	Gradient threshold horizontal.

6.2.18 Black Adjustment Offset Register (BLKADJOFF)

The black adjustment offset register (BLKADJOFF) is shown in Figure 112 and described in Table 101.

Figure 112. Black Adjustment Offset Register (BLKADJOFF)

31 24	23 16
Reserved	R
R-0	R/W-0

15 8	7 0
G	B
R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 101. Black Adjustment Offset Register (BLKADJOFF) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	R	0-FFh	Black level offset adjustment for Red in 2s complement format.
15-8	G	0-FFh	Black level offset adjustment for Green in 2s complement format.
7-0	B	0-FFh	Black level offset adjustment for Blue in 2s complement format.

6.2.19 RGB2RGB Blending Matrix Coefficients Register (RGB_MAT1)

The RGB2RGB blending matrix coefficients register (RGB_MAT1) is shown in Figure 113 and described in Table 102.

Figure 113. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT1)

31 28	27 16
Reserved	MTX_GR
R-0	R/W-100h

15 12	11 0
Reserved	MTX_RR
R-0	R/W-100h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 102. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT1) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	MTX_GR	0-FFFh	Blending value for GR position (S12Q8 format).
15-12	Reserved	0	Reserved
11-0	MTX_RR	0-FFFh	Blending value for RR position (S12Q8 format).

6.2.20 RGB2RGB Blending Matrix Coefficients Register (RGB_MAT2)

The RGB2RGB blending matrix coefficients register (RGB_MAT2) is shown in Figure 114 and described in Table 103.

Figure 114. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT2)

31 28	27 16
Reserved	MTX_RG
R-0	R/W-100h

15 12	11 0
Reserved	MTX_BR
R-0	R/W-100h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 103. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT2) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	MTX_RG	0-FFFh	Blending value for RG position (S12Q8 format).
15-12	Reserved	0	Reserved
11-0	MTX_BR	0-FFFh	Blending value for BR position (S12Q8 format).

6.2.21 RGB2RGB Blending Matrix Coefficients Register (RGB_MAT3)

The RGB2RGB blending matrix coefficients register (RGB_MAT3) is shown in Figure 115 and described in Table 104.

Figure 115. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT3)

31 28	27 16
Reserved	MTX_BG
R-0	R/W-100h

15 12	11 0
Reserved	MTX_GG
R-0	R/W-100h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 104. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT3) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	MTX_BG	0-FFFh	Blending value for BG position (S12Q8 format).
15-12	Reserved	0	Reserved
11-0	MTX_GG	0-FFFh	Blending value for GG position (S12Q8 format).

6.2.22 RGB2RGB Blending Matrix Coefficients Register (RGB_MAT4)

The RGB2RGB blending matrix coefficients register (RGB_MAT4) is shown in Figure 116 and described in Table 105.

Figure 116. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT4)

31 28	27 16
Reserved	MTX_GB
R-0	R/W-100h

15 12	11 0
Reserved	MTX_RB
R-0	R/W-100h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 105. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT4) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	MTX_GB	0-FFFh	Blending value for GB position (S12Q8 format).
15-12	Reserved	0	Reserved
11-0	MTX_RB	0-FFFh	Blending value for RB position (S12Q8 format).

6.2.23 RGB2RGB Blending Matrix Coefficients Register (RGB_MAT5)

The RGB2RGB blending matrix coefficients register (RGB_MAT5) is shown in Figure 117 and described in Table 106.

Figure 117. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT5)

31 16
Reserved
R-0

15 12	11 0
Reserved	MTX_BB
R-0	R/W-100h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 106. RGB2RGB Blending Matrix Coefficients Register (RGB_MAT5) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reserved
11-0	MTX_BB	0-FFFh	Blending value for BB position (S12Q8 format).

6.2.24 RGB2RGB Blending Matrix Offsets Register (RGB_OFF1)

The RGB2RGB blending matrix offsets register (RGB_OFF1) is shown in Figure 118 and described in Table 107.

Figure 118. RGB2RGB Blending Matrix Offsets Register (RGB_OFF1)

31 26	25 16
Reserved	MTX_OFFR
R-0	R/W-0

15 10	9 0
Reserved	MTX_OFFG
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 107. RGB2RGB Blending Matrix Offsets Register (RGB_OFF1) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	MTX_OFFR	0-3FFh	Blending offset value for Red (in 2s complement).
15-10	Reserved	0	Reserved
9-0	MTX_OFFG	0-3FFh	Blending offset value for Green (in 2s complement).

6.2.25 RGB2RGB Blending Matrix Offsets Register (RGB_OFF2)

The RGB2RGB blending matrix offsets register (RGB_OFF2) is shown in Figure 119 and described in Table 108.

Figure 119. RGB2RGB Blending Matrix Offsets Register (RGB_OFF2)

31	16
Reserved	
R-0	

15	10	9	0
Reserved		MTX_OFFB	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 108. RGB2RGB Blending Matrix Offsets Register (RGB_OFF2) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	MTX_OFFB	0-3FFh	Blending offset value for Blue (in 2s complement).

6.2.26 Color Space Conversion Coefficients Register (CSC0)

The color space conversion coefficients register (CSC0) is shown in Figure 120 and described in Table 109.

Figure 120. Color Space Conversion Coefficients Register (CSC0)

31	30	29	20	19	16
Reserved		CSCBY		CSCGY	
R-0		R/W-1Ch		R/W-98h	

15	10	9	0
CSCGY		CSCRY	
R/W-98h		R/W-4Ch	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 109. Color Space Conversion Coefficients Register (CSC0) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-20	CSCBY	0-3FFh	Color space conversion coefficient of B for computing Y. Precision is S10Q8.
19-10	CSCGY	0-3FFh	Color space conversion coefficient of G for computing Y. Precision is S10Q8.
9-0	CSCRY	0-3FFh	Color space conversion coefficient of R for computing Y. Precision is S10Q8.

6.2.27 Color Space Conversion Coefficients Register (CSC1)

The color space conversion coefficients register (CSC1) is shown in Figure 121 and described in Table 110.

Figure 121. Color Space Conversion Coefficients Register (CSC1)

31 30	29 20	19 16
Reserved	CSCBCB	CSCGCB
R-0	R/W-80h	R/W-3ACh

15 10	9 0
CSCGCB	CSCRCB
R/W-3ACh	R/W-3D4h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 110. Color Space Conversion Coefficients Register (CSC1) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-20	CSCBCB	0-3FFh	Color space conversion coefficient of R for computing Cr. Precision is S10Q8.
19-10	CSCGCB	0-3FFh	Color space conversion coefficient of B for computing Cb. Precision is S10Q8.
9-0	CSCRCB	0-3FFh	Color space conversion coefficient of G for computing Cb. Precision is S10Q8.

6.2.28 Color Space Conversion Coefficients Register (CSC2)

The color space conversion coefficients register (CSC2) is shown in Figure 122 and described in Table 111.

Figure 122. Color Space Conversion Coefficients Register (CSC2)

31 30	29 20	19 16
Reserved	CSCBCR	CSCGCR
R-0	R/W-80h	R/W-3ACh

15 10	9 0
CSCGCR	CSCRCR
R/W-3ACh	R/W-3D4h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 111. Color Space Conversion Coefficients Register (CSC2) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-20	CSCBCR	0-3FFh	Color space conversion coefficient of B for computing Cr. Precision is S10Q8.
19-10	CSCGCR	0-3FFh	Color space conversion coefficient of B for computing Cb. Precision is S10Q8.
9-0	CSCRCR	0-3FFh	Color space conversion coefficient of G for computing Cb. Precision is S10Q8.

6.2.29 Color Space Conversion Offsets Register (CSC_OFFSET)

The color space conversion offsets register (CSC_OFFSET) is shown in Figure 123 and described in Table 112.

Figure 123. Color Space Conversion Offsets Register (CSC_OFFSET)

31	24	23	16
Reserved		YOFST	
R-0		R/W-0	

15	8	7	0
OFSTCB		OFSTCR	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 112. Color Space Conversion Offsets Register (CSC_OFFSET) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	YOFST	0-FFh	DC offset value for Y (S8Q0). Yout = Yin + YOFST (-128 to 127).
15-8	OFSTCB	0-FFh	DC offset value for Cb (S8Q0). Cout = Cin + OFSTCB (-128 to 127).
7-0	OFSTCR	0-FFh	DC offset value for Cr (S8Q0). Cout = Cin + OFSTCR (-128 to 127).

6.2.30 Contrast and Brightness Settings Register (CNT_BRT)

The contrast and brightness settings register (CNT_BRT) is shown in Figure 124 and described in Table 113.

Figure 124. Contrast and Brightness Settings Register (CNT_BRT)

31	16
Reserved	
R-0	

15	8	7	0
CNT		BRT	
R/W-10h		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 113. Contrast and Brightness Settings Register (CNT_BRT) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-8	CNT	0-FFh	Contrast Adjustment. Sets the contrast of the Y data. Precision is U8Q4 (0 to 15.9375). Applied after offset adjustments.
7-0	BRT	0-FFh	Brightness Adjustment. Sets the brightness of Y data (0 to 255). Applied after contrast adjustments.

6.2.31 Chrominance Supression Settings Register (CSUP)

The chrominance supression settings register (CSUP) is shown in Figure 125 and described in Table 114.

Figure 125. Chrominance Supression Settings Register (CSUP)

31 – 17	16
Reserved	HPFY
R-0	R/W-0

15 – 8	7 – 0
CSUPTH	CSUPG
R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 114. Chrominance Supression Settings Register (CSUP) Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved
16	HPFY		Use HPF of luminance for chroma suppression.
		0	Disable (use luminance without HPF)
		1	Enable
15-8	CSUPTH	0-FFh	Chroma suppression threshold
7-0	CSUPG	0-FFh	Gain value for chroma suppression function. Precision is U8Q8 (0 to 0.99609375).

6.2.32 Maximum/Minimum Y and C Settings Register (SETUP_YC)

The maximum/minimum Y and C settings register (SETUP_YC) is shown in Figure 126 and described in Table 115.

Figure 126. Maximum/Minimum Y and C Settings Register (SETUP_YC)

31 – 24	23 – 16
MAXY	MINY
R/W-FFh	R/W-0

15 – 8	7 – 0
MAXC	MINC
R/W-FFh	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 115. Maximum/Minimum Y and C Settings Register (SETUP_YC) Field Descriptions

Bit	Field	Value	Description
31-24	MAXY	0-FFh	Maximum Y value. Values greater than set value are clipped to the set value.
23-16	MINY	0-FFh	Minimum Y value. Values less than set value are clipped to the set value.
15-8	MAXC	0-FFh	Maximum Cb and Cr value. Values greater than set value are clipped to the set value.
7-0	MINC	0-FFh	Minimum Cb and Cr value. Values less than set value are clipped to the set value.

6.2.33 Setup Table Addresses Register (SET_TBL_ADDRESS)

The setup table addresses register (SET_TBL_ADDRESS) is shown in Figure 127 and described in Table 116.

Figure 127. Setup Table Addresses Register (SET_TBL_ADDRESS)

31–16
Reserved
R-0

15–13	12–0
Reserved	ADDR
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 116. Setup Table Addresses Register (SET_TBL_ADDRESS) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved
12-0	ADDR	0-1FFFh	13-bit address

6.2.34 Setup Table Data Register (SET_TBL_DATA)

The setup table data register (SET_TBL_DATA) is shown in Figure 128 and described in Table 117.

Figure 128. Setup Table Data Register (SET_TBL_DATA)

31–20	19–16
Reserved	DATA
R-0	R/W-0

15–0
DATA
R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 117. Setup Table Data Register (SET_TBL_DATA) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved
19-0	DATA	0-7FFFFh	Data to be written. All 20-bits are valid for the non-linear enhancer. Only 8 LSBs are chosen for the gamma, noise filter, and CFA coefficient tables.

6.3 Resizer Registers

Table 118 lists the memory-mapped registers for the resizer. See the device-specific data manual for the memory address of these registers.

Table 118. Resizer Registers

Offset	Acronym	Register Description	Section
C00h	PID	Peripheral revision and class information register	Section 6.3.1
C04h	PCR	Peripheral control register	Section 6.3.2
C08h	RSZ_CNT	Resizer control bits register	Section 6.3.3
C0Ch	OUT_SIZE	Output width and height after resizing register	Section 6.3.4
C10h	IN_START	Input starting information register	Section 6.3.5
C14h	IN_SIZE	Input width and height before resizing register	Section 6.3.6
C18h	SDR_INADD	Input SDRAM address register	Section 6.3.7
C1Ch	SDR_INOFF	SDRAM offset for the input line register	Section 6.3.8
C20h	SDR_OUTADD	Output SDRAM address register	Section 6.3.9
C24h	SDR_OUTOFF	SDRAM offset for the output line register	Section 6.3.10
C28h	HFILT10	Horizontal filter coefficients 1 and 0 register	Section 6.3.11
C2Ch	HFILT32	Horizontal filter coefficients 3 and 2 register	Section 6.3.11
C30h	HFILT54	Horizontal filter coefficients 5 and 4 register	Section 6.3.11
C34h	HFILT76	Horizontal filter coefficients 7 and 6 register	Section 6.3.11
C38h	HFILT98	Horizontal filter coefficients 9 and 8 register	Section 6.3.11
C3Ch	HFILT1110	Horizontal filter coefficients 11 and 10 register	Section 6.3.11
C40h	HFILT1312	Horizontal filter coefficients 13 and 12 register	Section 6.3.11
C44h	HFILT1514	Horizontal filter coefficients 15 and 14 register	Section 6.3.11
C48h	HFILT1716	Horizontal filter coefficients 17 and 16 register	Section 6.3.11
C4Ch	HFILT1918	Horizontal filter coefficients 19 and 18 register	Section 6.3.11
C50h	HFILT2120	Horizontal filter coefficients 21 and 20 register	Section 6.3.11
C54h	HFILT2322	Horizontal filter coefficients 23 and 22 register	Section 6.3.11
C58h	HFILT2524	Horizontal filter coefficients 25 and 24 register	Section 6.3.11
C5Ch	HFILT2726	Horizontal filter coefficients 27 and 26 register	Section 6.3.11
C60h	HFILT2928	Horizontal filter coefficients 29 and 28 register	Section 6.3.11
C64h	HFILT3130	Horizontal filter coefficients 31 and 30 register	Section 6.3.11
C68h	VFILT10	Vertical filter coefficients 1 and 0 register	Section 6.3.12
C6Ch	VFILT32	Vertical filter coefficients 3 and 2 register	Section 6.3.12
C70h	VFILT54	Vertical filter coefficients 5 and 4 register	Section 6.3.12
C74h	VFILT76	Vertical filter coefficients 7 and 6 register	Section 6.3.12
C78h	VFILT98	Vertical filter coefficients 9 and 8 register	Section 6.3.12
C7Ch	VFILT1110	Vertical filter coefficients 11 and 10 register	Section 6.3.12
C80h	VFILT1312	Vertical filter coefficients 13 and 12 register	Section 6.3.12
C84h	VFILT1514	Vertical filter coefficients 15 and 14 register	Section 6.3.12
C88h	VFILT1716	Vertical filter coefficients 17 and 16 register	Section 6.3.12
C8Ch	VFILT1918	Vertical filter coefficients 19 and 18 register	Section 6.3.12
C90h	VFILT2120	Vertical filter coefficients 21 and 20 register	Section 6.3.12
C94h	VFILT2322	Vertical filter coefficients 23 and 22 register	Section 6.3.12
C98h	VFILT2524	Vertical filter coefficients 25 and 24 register	Section 6.3.12
C9Ch	VFILT2726	Vertical filter coefficients 27 and 26 register	Section 6.3.12
CA0h	VFILT2928	Vertical filter coefficients 29 and 28 register	Section 6.3.12
CA4h	VFILT3130	Vertical filter coefficients 31 and 30 register	Section 6.3.12
CA8h	YENH	Luminance enhancer register	Section 6.3.13

6.3.1 Peripheral Revision and Class Information Register (PID)

The peripheral revision and class information register (PID) is shown in Figure 129 and described in Table 119.

Figure 129. Peripheral Revision and Class Information Register (PID)

31	24	23	16
Reserved		TID	
R-0		R-10h	

15	8	7	0
CID		PREV	
R-FEh		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 119. Peripheral Revision and Class Information Register (PID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	TID		Peripheral identification
		10h	Resizer
15-8	CID		Class identification
		FEh	Video processing front end module
7-0	PREV		Peripheral revision number
		0	Current revision

6.3.2 Peripheral Control Register (PCR)

The peripheral control register (PCR) is shown in Figure 130 and described in Table 120.

Figure 130. Peripheral Control Register (PCR)

31 16
Reserved
R-0

15 2	1	0
Reserved	BUSY	ENABLE
R-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 120. Peripheral Control Register (PCR) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	BUSY		Busy bit
		0	Resizer module is not busy.
		1	Resizer module is busy.
0	ENABLE		Enable. Resizer always operates in one-shot mode; programmer must enable resizer for each frame that needs to be resized. Enable bit is reset to 0 once the BUSY bit turns to 1. The ENABLE bit must be the last field written to resize a frame. The ENABLE bit can be written when the resizer is busy.
		0	Disable resizer module.
		1	Enable resizer module.

6.3.3 Resizer Control Bits Register (RSZ_CNT)

The resizer control bits register (RSZ_CNT) is shown in Figure 131 and described in Table 121.

Figure 131. Resizer Control Bits Register (RSZ_CNT)

31 30	29	28	27	26	25 23	22 20	19 16
Reserved	CBILIN	INPSRC	INPTYP	YCPOS	VSTPH	HSTPH	VRSZ
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-255

15 10	9 0
VRSZ	HRSZ
R/W-FFh	R/W-FFh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 121. Resizer Control Bits Register (RSZ_CNT) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29	CBILIN		Chrominance horizontal algorithm.
		0	Same as luminance processing
		1	Bilinear interpolation
28	INPSRC		Input source
		0	Preview engine
		1	SDRAM
27	INPTYP		Input type
		0	YUV422 color interleaved
		1	Color separate (8-bit data)
26	YCPOS		Luminance and chrominance position in 16-bit word
		0	YC
		1	CY
25-23	VSTPH	0-7h	Vertical starting phase (0-7)
22-20	HSTPH	0-7h	Horizontal starting phase (0-7)
19-10	VRSZ	0-3FFh	Vertical resizing value (range from 64-1024) plus 1. Vertical resizing ratio is 256/VRSZ.
9-0	HRSZ	0-3FFh	Horizontal resizing value (range from 64-1024) plus 1. Horizontal resizing ratio is 256/HRSZ.

6.3.4 Output Width and Height After Resizing Register (OUT_SIZE)

The output width and height after resizing register (OUT_SIZE) is shown in Figure 132 and described in Table 122.

Figure 132. Output Width and Height After Resizing Register (OUT_SIZE)

31 27	26 16
Reserved	VERT
R-0	R/W-0

15 11	10 0
Reserved	HORZ
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 122. Output Width and Height After Resizing Register (OUT_SIZE) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Reserved
26-16	VERT	0-7FFh	Output height (in the vertical direction)
15-11	Reserved	0	Reserved
10-0	HORZ	0-7FFh	Output width (in the horizontal direction). The maximum output width cannot be greater than 1280 pixels wide (640, if down-sampling greater than 2 is used with 7 filter taps). This value must be even and the number of bytes written to SDRAM must be a multiple of 16-bytes, if the vertical resizing factor is greater than 1× (upsizing).

6.3.5 Input Starting Information Register (IN_START)

The input starting information register (IN_START) is shown in Figure 133 and described in Table 123.

Figure 133. Input Starting Information Register (IN_START)

31 29	28 16
Reserved	VERT_ST
R-0	R/W-0

15 13	12 0
Reserved	HORZ_ST
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 123. Input Starting Information Register (IN_START) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28-16	VERT_ST	0-1FFFh	Vertical starting line. This field makes sense when the resizer obtains its input from the preview engine. When the resizer gets its input from SDRAM, this field must be cleared to 0.
15-13	Reserved	0	Reserved
12-0	HORZ_ST	0-1FFFh	Horizontal starting pixel. This field makes sense when the resizer obtains its input from the preview engine. When the resizer gets its input from SDRAM, this field must be set to ≤15 for YUV 16-bit data and ≤31 for 8-bit color separate data. Horizontal starting pixel value is in number of pixels, if input is from SDRAM. If the input to the resizer is from CCD controller/preview engine, this field needs to be programmed as follows: 1. Program this field using number of bytes (twice number of pixels). 2. Change the lowest bit to reflect start position in pixels (effectively change from a value 0 to a value 1, if required).

6.3.6 Input Width and Height Before Resizing Register (IN_SIZE)

The input width and height before resizing register (IN_SIZE) is shown in Figure 134 and described in Table 124.

Figure 134. Input Width and Height Before Resizing Register (IN_SIZE)

31 29	28 16
Reserved	VERT
R-0	R/W-0

15 13	12 0
Reserved	HORZ
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 124. Input Width and Height Before Resizing Register (IN_SIZE) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28-16	VERT	0-1FFFh	Input height (in the vertical direction).
15-13	Reserved	0	Reserved
12-0	HORZ	0-1FFFh	Input width (in the horizontal direction).

6.3.7 Input SDRAM Address Register (SDR_INADD)

The input SDRAM address register (SDR_INADD) is shown in Figure 135 and described in Table 125.

Figure 135. Input SDRAM Address Register (SDR_INADD)

31 0
SDR_INADD
R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 125. Input SDRAM Address Register (SDR_INADD) Field Descriptions

Bit	Field	Value	Description
31-0	SDR_INADD	0-FFFF FFFFh	Input SDRAM address. The 5 LSBs are forced to be zeros by the hardware to align on a 32-byte boundary; the 5 LSBs are read-only. This field must be programmed to be 0, if the resizer input is from preview engine/CCD controller. This field can be altered even when the resizer is busy. Change will take place only for the next frame. However, note that reading this register will always give the latest value.

6.3.8 SDRAM Offset for the Input Line Register (SDR_INOFF)

The SDRAM offset for the input line register (SDR_INOFF) is shown in Figure 136 and described in Table 126.

Figure 136. SDRAM Offset for the Input Line Register (SDR_INOFF)

31	16
Reserved	
R-0	

15	0
OFFSET	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 126. SDRAM Offset for the Input Line Register (SDR_INOFF) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	OFFSET	0-FFFFh	Byte offset of each line in the SDRAM address. The 5 LSBs are forced to be zeros by the hardware to align on a 32-byte boundary; the 5 LSBs are read-only. This field must be programmed to be 0, if the resizer input is from preview engine/CCD controller. This field can be altered even when the resizer is busy. Change will take place only for the next frame. However, note that reading this register will always give the latest value.

6.3.9 Output SDRAM Address Register (SDR_OUTADD)

The output SDRAM address register (SDR_OUTADD) is shown in Figure 137 and described in Table 127.

Figure 137. Output SDRAM Address Register (SDR_OUTADD)

31	0
SDR_OUTADD	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 127. Output SDRAM Address Register (SDR_OUTADD) Field Descriptions

Bit	Field	Value	Description
31-0	SDR_OUTADD	0-FFFF FFFFh	Output SDRAM address. The 5 LSBs are treated as zeroes; should be on a 32-byte boundary. This field can be altered even when the resizer is busy. Change will take place only for the next frame. However, note that reading this register will always give the latest value.

6.3.10 SDRAM Offset for the Output Line Register (SDR_OUTOFF)

The SDRAM offset for the output line register (SDR_OUTOFF) is shown in Figure 138 and described in Table 128.

Figure 138. SDRAM Offset for the Output Line Register (SDR_OUTOFF)

31 16
Reserved
R-0

15 0
OFFSET
R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 128. SDRAM Offset for the Output Line Register (SDR_OUTOFF) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	OFFSET	0-FFFFh	Offset of each line in the SDRAM address. The 5 LSBs are treated as zeroes; should be on a 32-byte boundary. This field can be altered even when the resizer is busy. Change will take place only for the next frame. However, note that reading this register will always give the latest value.

6.3.11 Horizontal Filter Coefficients Register (HFILT*oe*)

The horizontal filter coefficients register (HFILT*oe*) is shown in Figure 139 and described in Table 129.

Figure 139. Horizontal Filter Coefficients Register (HFILT*oe*)

31 26	25 16
Reserved	COEF*o*
R-0	R/W-0

15 10	9 0
Reserved	COEF*e*
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 129. Horizontal Filter Coefficients Register (HFILT*oe*) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	COEF*o*	0-3FFh	Coefficient - Phase 0, tap *o* (S10Q8 format: range of -2 to 1.255/256, 1 is 100h).
15-10	Reserved	0	Reserved
9-0	COEF*e*	0-3FFh	Coefficient - Phase 0, tap *e* (S10Q8 format: range of -2 to 1.255/256, 1 is 100h).

6.3.12 Vertical Filter Coefficients Register (VFILT*oe*)

The Vertical Filter Coefficients register (VFILT*oe*) is shown in Figure 140 and described in Table 130.

Figure 140. Vertical Filter Coefficients Register (VFILT*oe*)

31 26	25 16
Reserved	COEF*o*
R-0	R/W-0

15 10	9 0
Reserved	COEF*e*
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 130. Vertical Filter Coefficients Register (VFILT*oe*) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	COEF*o*	0-3FFh	Coefficient - Phase 0, tap *o* (S10Q8 format: range of -2 to 1.255/256, 1 is 100h).
15-10	Reserved	0	Reserved
9-0	COEF*e*	0-3FFh	Coefficient - Phase 0, tap *e* (S10Q8 format: range of -2 to 1.255/256, 1 is 100h).

6.3.13 Luminance Enhancer Register (YENH)

The luminance enhancer register (YENH) is shown in Figure 141 and described in Table 131.

Figure 141. Luminance Enhancer Register (YENH)

31 18	17 16
Reserved	ALGO
R-0	R/W-0

15 12	11 8	7 0
GAIN	SLOP	CORE
R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 131. Luminance Enhancer Register (YENH) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved
17-16	ALGO	0-3h	Luminance Algorithm. hpgain = (|HPF(Y)| - CORE) × SLOP, saturate hpgain between 0 and gain. Y' = Y + (HPF(Y) × hpgain + 8) >> 4, saturate Y' between 0 and 255.
		0	Disable
		1h	[-1 2 -1]/2 HPF
		2h	[-1 -2 6 -2 -1]/4 HPF
		3h	Reserved
15-12	GAIN	0-Fh	Max gain (U4Q4)
11-8	SLOP	0-Fh	Slope (U4Q4)
7-0	CORE	0-FFh	Coring offset (U8Q0)

6.4 Histogram Registers

Table 132 lists the memory-mapped registers for the histogram. See the device-specific data manual for the memory address of these registers.

Table 132. Histogram Registers

Offset	Acronym	Register Description	Section
1000h	PID	Peripheral identification register	Section 6.4.1
1004h	PCR	Peripheral control register	Section 6.4.2
1008h	HIST_CNT	Histogram control register	Section 6.4.3
100Ch	WB_GAIN	White/channel balance settings register	Section 6.4.4
1010h	R0_HORZ	Region 0 horizontal information register	Section 6.4.5
1014h	R0_VERT	Region 0 vertical information register	Section 6.4.6
1018h	R1_HORZ	Region 1 horizontal information register	Section 6.4.5
101Ch	R1_VERT	Region 1 vertical information register	Section 6.4.6
1020h	R2_HORZ	Region 2 horizontal information register	Section 6.4.5
1024h	R2_VERT	Region 2 vertical information register	Section 6.4.6
1028h	R3_HORZ	Region 3 horizontal information register	Section 6.4.5
102Ch	R3_VERT	Region 3 vertical information register	Section 6.4.6
1030h	HIST_ADDR	Histogram address register	Section 6.4.7
1034h	HIST_DATA	Histogram data register	Section 6.4.8
1038h	RADD	Read address register	Section 6.4.9
103Ch	RADD_OFF	Read address offset register	Section 6.4.10
1040h	H_V_INFO	Horizontal/vertical information register	Section 6.4.11

6.4.1 Peripheral Identification Register (PID)

The peripheral identification register (PID) is shown in Figure 142 and described in Table 133.

Figure 142. Peripheral Identification Register (PID)

31 - 24	23 - 16
Reserved	TID
R-0	R-8h

15 - 8	7 - 0
CID	PREV
R-FEh	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 133. Peripheral Identification Register (PID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	TID		Identifies type of peripheral.
		8h	Histogram
15-8	CID		Identifies class of peripheral.
		FEh	Video processing front end module
7-0	PREV		Identifies revision of peripheral.
		0	Current revision

6.4.2 Peripheral Control Register (PCR)

The peripheral control register (PCR) is shown in Figure 143 and described in Table 134.

Figure 143. Peripheral Control Register (PCR)

31 - 16
Reserved
R-0

15 - 2	1	0
Reserved	BUSY	ENABLE
R-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 134. Peripheral Control Register (PCR) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	BUSY		Histogram busy.
		0	Histogram is not busy.
		1	Histogram is busy.
0	ENABLE		Histogram enable.
		0	Histogram is disabled.
		1	Histogram is enabled.

6.4.3 Histogram Control Register (HIST_CNT)

The histogram control register (HIST_CNT) is shown in Figure 144 and described in Table 135.

Figure 144. Histogram Control Register (HIST_CNT)

31 16
Reserved
R-0

15 9	8	7	6	5 4	3	2 0
Reserved	DATSIZ	CLR	CFA	BINS	SOURCE	SHIFT
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 135. Histogram Control Register (HIST_CNT) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved
8	DATSIZ		Input data width.
		0	9 to 14 bits
		1	8 bits
7	CLR		Clear data after read.
		0	Do not clear data after read.
		1	Clear data after read.
6	CFA		CFA pattern
		0	2D (2 × 2) conventional pattern
		1	Foveon sensor pattern
5-4	BINS	0-3h	Number of bins per histogram.
		0	32 bins - Regions 0, 1, 2, and 3 are active.
		1h	64 bins - Regions 0, 1, 2, and 3 are active.
		2h	128 bins - Only regions 0 and 1 are active.
		3h	256 bins - Only region 0 is active.
3	SOURCE		Input source.
		0	CCD controller
		1	SDRAM
2-0	SHIFT	0-7h	Number of bits to right shift data before accessing bins.

6.4.4 White/Channel Balance Settings Register (WB_GAIN)

The white/channel balance settings register (WB_GAIN) is shown in Figure 145 and described in Table 136.

Figure 145. White/Channel Balance Settings Register (WB_GAIN)

31 — 24	23 — 16
WG00	WG01
R/W-20h	R/W-20h

15 — 8	7 — 0
WG02	WG03
R/W-20h	R/W-20h

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 136. White/Channel Balance Settings Register (WB_GAIN) Field Descriptions

Bit	Field	Value	Description
31-24	WG00	0-7.1Fh	White balance gain 00. WG00 is set within the range of 0 to 7.96875. The decimal point is located between WG00[4] and WG00[5]. See Table 137.
23-16	WG01	0-7.1Fh	White balance gain 01. WG01 is set within the range of 0 to 7.96875. The decimal point is located between WG01[4] and WG01[5]. See Table 137.
15-8	WG02	0-7.1Fh	White balance gain 02. WG02 is set within the range of 0 to 7.96875. The decimal point is located between WG02[4] and WG02[5]. See Table 137.
7-0	WG03	0-7.1Fh	White balance gain 03. WG03 is set within the range of 0 to 7.96875. The decimal point is located between WG03[4] and WG03[5]. See Table 137.

Table 137. White Balance Gain Values

WG0*n* Bits								
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Value**
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1.0
0	0	1	0	0	0	0	1	1.03125
0	0	1	0	0	0	1	0	1.0625
0	0	1	0	0	0	1	1	1.09375
0	0	1	0	0	1	0	0	1.125
0	0	1	0	0	1	0	1	1.15625
0	0	1	0	0	1	1	0	1.1875
0	0	1	0	0	1	1	1	1.21875
0	0	1	0	1	0	0	0	1.25
0	0	1	0	1	0	0	1	1.28125
0	0	1	0	1	0	1	0	1.3125
0	0	1	0	1	0	1	1	1.34375
0	0	1	0	1	1	0	0	1.375
0	0	1	0	1	1	0	1	1.40625
0	0	1	0	1	1	1	0	1.4375
0	0	1	0	1	1	1	1	1.46875
0	0	1	1	0	0	0	0	1.5
0	0	1	1	0	0	0	1	1.53125
0	0	1	1	0	0	1	0	1.5625
0	0	1	1	0	0	1	1	1.59375
0	0	1	1	0	1	0	0	1.625
0	0	1	1	0	1	0	1	1.65625
0	0	1	1	0	1	1	0	1.6875
0	0	1	1	0	1	1	1	1.71875
0	0	1	1	1	0	0	0	1.75
0	0	1	1	1	0	0	1	1.78125
0	0	1	1	1	0	1	0	1.8125
0	0	1	1	1	0	1	1	1.84375
0	0	1	1	1	1	0	0	1.875
0	0	1	1	1	1	0	1	1.90625
0	0	1	1	1	1	1	0	1.9375
0	0	1	1	1	1	1	1	1.96875
0	1	0	x	x	x	x	x	2.0 to 2.96875
0	1	1	x	x	x	x	x	3.0 to 3.96875
1	0	0	x	x	x	x	x	4.0 to 4.96875
1	0	1	x	x	x	x	x	5.0 to 5.96875
1	1	0	x	x	x	x	x	6.0 to 6.96875
1	1	1	x	x	x	x	x	7.0 to 7.96875

6.4.5 Region *n* Horizontal Information Registers (R0_HORZ-R3_HORZ)

The region *n* horizontal information register (R0_HORZ-R3_HORZ) is shown in Figure 146 and described in Table 138.

Figure 146. Region *n* Horizontal Information Registers (R0_HORZ-R3_HORZ)

31 30	29 16
Reserved	HSTART
R-0	R/W-0

15 14	13 0
Reserved	HEND
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 138. Region *n* Horizontal Information Registers (R0_HORZ-R3_HORZ) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-16	HSTART	0-3FFFh	Horizontal start location for region *n*.
15-14	Reserved	0	Reserved
13-0	HEND	0-3FFFh	Horizontal end for region *n*. If the end is programmed as the same value of start, then the region size is treated as 0. In all cases, the total number of pixels processed is End - Start + 1. Therefore, the minimum region dimension is at least 2.

6.4.6 Region *n* Vertical Information Registers (R0_VERT-R3_VERT)

The region *n* vertical information register (R0_VERT-R3_VERT) is shown in Figure 147 and described in Table 139.

Figure 147. Region *n* Vertical Information Registers (R0_VERT-R3_VERT)

31 30	29 16
Reserved	VSTART
R-0	R/W-0

15 14	13 0
Reserved	VEND
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 139. Region *n* Vertical Information Registers (R0_VERT-R3_VERT) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-16	VSTART	0-3FFFh	Vertical start location for region *n*.
15-14	Reserved	0	Reserved
13-0	VEND	0-3FFFh	Vertical end for region *n*. If the end is programmed as the same value of start, then the region size is treated as 0. In all cases, the total number of pixels processed is End - Start + 1. Therefore, the minimum region dimension is at least 2.

6.4.7 Histogram Address Register (HIST_ADDR)

The histogram address register (HIST_ADDR) contains the histogram address for the data to be read. HIST_ADDR is shown in Figure 148 and described in Table 140.

Figure 148. Histogram Address Register (HIST_ADDR)

31	16
Reserved	
R-0	

15	10	9	0
Reserved		ADDR	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 140. Histogram Address Register (HIST_ADDR) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	ADDR	0-3FFh	Address of histogram entry (note that each data entry is 20-bits wide).

6.4.8 Histogram Data Register (HIST_DATA)

The histogram data register (HIST_DATA) contains the histogram data that is read from memory. HIST_DATA is shown in Figure 149 and described in Table 141.

Figure 149. Histogram Data Register (HIST_DATA)

31	20	19	16
Reserved		RDATA	
R-0		R/W-0	

15	0
RDATA	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 141. Histogram Data Register (HIST_DATA) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved
19-0	RDATA	0-F FFFFh	Histogram data that is read from memory.

6.4.9 Read Address Register (RADD)

The read address register (RADD) specifies the address on the DMA port that should be initially read from the SDRAM/DDRAM. The 5 least-significant bits are always treated as zeroes. The starting address must be aligned on a 32-byte boundary. RADD can be altered even when the histogram is busy. RADD is shown in Figure 150 and described in Table 142.

Figure 150. Read Address Register (RADD)

31 0
RADD
R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 142. Read Address Register (RADD) Field Descriptions

Bit	Field	Value	Description
31-0	RADD	0-FFFF FFFFh	Read address for each line in SDRAM/DDRAM. Specifies the SDRAM address that should be initially read. The 5 LSB are always treated as 0 (the starting address must be aligned on a 32-byte boundary). RADD can be altered even when the histogram is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

6.4.10 Read Address Offset Register (RADD_OFF)

The read address offset register (RADD_OFF) specifies the address offset for each line in the SDRAM/DDRAM. The 5 least-significant bits are always treated as zeroes. The offset must be aligned on a 32-byte boundary. RADD_OFF can be altered even when the histogram is busy, but the change will take place only for the next frame. RADD_OFF is shown in Figure 151 and described in Table 143.

Figure 151. Read Address Offset Register (RADD_OFF)

31 16
Reserved
R-0

15 0
OFFSET
R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 143. Read Address Offset Register (RADD_OFF) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	Reserved	Reserved
15-0	OFFSET	0-FFFFh	Read address offset for each line in SDRAM/DDRAM. The 5 LSB are always treated as 0 (the address offset must be aligned on a 32-byte boundary). OFFSET can be altered even when the histogram is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

6.4.11 Horizontal/Vertical Information Register (H_V_INFO)

The horizontal/vertical information register (H_V_INFO) contains the number of horizontal pixels and vertical lines when the input data source is the SDRAM/DDRAM. H_V_INFO is shown in Figure 152 and described in Table 144.

Figure 152. Horizontal/Vertical Information Register (H_V_INFO)

31 30	29 16
Reserved	HSIZE
R-0	R/W-0

15 14	13 0
Reserved	VSIZE
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 144. Horizontal/Vertical Information Register (H_V_INFO) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-16	HSIZE	0-3FFFh	Horizontal size. Number of horizontal pixels when the input data source is SDRAM/DDRAM.
15-14	Reserved	0	Reserved
13-0	VSIZE	0-3FFFh	Vertical size. Number of vertical lines when the input data source is SDRAM/DDRAM.

6.5 Hardware 3A (H3A) Registers

Table 145 lists the memory-mapped registers for the hardware 3A (H3A). See the device-specific data manual for the memory address of these registers.

Table 145. Hardware 3A (H3A) Registers

Offset	Acronym	Register Description	Section
1400h	PID	Peripheral revision and class information register	Section 6.5.1
1404h	PCR	Peripheral control register	Section 6.5.2
1408h	AFPAX1	Setup for the AF engine paxel configuration register	Section 6.5.3
140Ch	AFPAX2	Setup for the AF engine paxel configuration register	Section 6.5.4
1410h	AFPAXSTART	Start position for AF engine paxels register	Section 6.5.5
1414h	AFIIRSH	Start position for IIRSH register	Section 6.5.6
1418h	AFBUFST	SDRAM/DDRAM start address for AF engine register	Section 6.5.7
141Ch	AFCOEFF010	IIR filter coefficient data for SET 0 register	Section 6.5.8
1420h	AFCOEFF032	IIR filter coefficient data for SET 0 register	Section 6.5.9
1424h	AFCOEFF054	IIR filter coefficient data for SET 0 register	Section 6.5.10
1428h	AFCOEFF076	IIR filter coefficient data for SET 0 register	Section 6.5.11
142Ch	AFCOEFF098	IIR filter coefficient data for SET 0 register	Section 6.5.12
1430h	AFCOEFF0010	IIR filter coefficient data for SET 0 register	Section 6.5.13
1434h	AFCOEFF110	IIR filter coefficient data for SET 1 register	Section 6.5.14
1438h	AFCOEFF132	IIR filter coefficient data for SET 1 register	Section 6.5.15
143Ch	AFCOEFF154	IIR filter coefficient data for SET 1 register	Section 6.5.16
1440h	AFCOEFF176	IIR filter coefficient data for SET 1 register	Section 6.5.17
1444h	AFCOEFF198	IIR filter coefficient data for SET 1 register	Section 6.5.18
1448h	AFCOEFF1010	IIR filter coefficient data for SET 1 register	Section 6.5.19
144Ch	AEWWIN1	Configuration for AE/AWB windows register	Section 6.5.20
1450h	AEWINSTART	Start position for AE/AWB windows register	Section 6.5.21
1454h	AEWINBLK	Start position and height for black line of AE/AWB windows register	Section 6.5.22
1458h	AEWSUBWIN	Configuration for subsample data in AE/AWB window register	Section 6.5.23
145Ch	AEWBUFST	SDRAM/DDRAM start address for AE/AWB engine register	Section 6.5.24

6.5.1 Peripheral Revision and Class Information Register (PID)

The peripheral revision and class information register (PID) is shown in Figure 153 and described in Table 146.

Figure 153. Peripheral Revision and Class Information Register (PID)

31 – 24	23 – 16
Reserved	TID
R-0	R-8h

15 – 8	7 – 0
CID	PREV
R-FEh	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 146. Peripheral Revision and Class Information Register (PID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	TID		Peripheral Identification
		8h	Hardware 3A
15-8	CID		Class identification
		FEh	Video processing front end module
7-0	PREV		Peripheral Revision Number
		0	Current revision

6.5.2 Peripheral Control Register (PCR)

The peripheral control register (PCR) is shown in Figure 154 and described in Table 147.

Figure 154. Peripheral Control Register (PCR)

31 – 24
AVE2LMT
R/W-3FFh

23 – 22	21 – 19	18	17	16
AVE2LMT	Reserved	BUSYAEAWB	AEW_ALAW_EN	AEW_EN
R/W-3FFh	R-0	R-0	R/W-0	R/W-0

15	14	13 – 11	10 – 8
BUSYAF	FVMODE	RGBPOS	MED_TH
R-0	R/W-0	R/W-0	R/W-FFh

7 – 3	2	1	0
MED_TH	AF_MED_EN	AF_ALAW_EN	AF_EN
R/W-FFh	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 147. Peripheral Control Register (PCR) Field Descriptions

Bit	Field	Value	Description
31-22	AVE2LMT	0-3FFh	AE/AWB saturation limit. This is the value that all subsampled pixels in the AE/AWB engine are compared. If the data is greater than or equal to this data then the block is considered saturated.
21-19	Reserved	0	Reserved
18	BUSYAEAWB	0-1	Busy bit for AE/AWB.
17	AEW_ALAW_EN		AE/AWB A-law table enable.
		0	AE/AWB A-law table is disabled.
		1	AE/AWB A-law table is enabled.
16	AEW_EN		AE/AWB engine enable
		0	AE/AWB engine is disabled.
		1	AE/AWB engine is enabled.
15	BUSYAF	0-1	Busy bit for AF.
14	FVMODE		Focus value accumulation mode.
		0	Sum mode
		1	Peak mode
13-11	RGBPOS	0-7h	Red, Green, and Blue pixel location in the AF windows.
		0	GR and GB as Bayer pattern.
		1h	RG and GB as Bayer pattern.
		2h	GR and BG as Bayer pattern.
		3h	RG and BG as Bayer pattern.
		4h	GG and RB as custom pattern.
		5h	RB and GG as custom pattern.
		6h-7h	Reserved
10-3	MED_TH	0-FFh	Median filter threshold
2	AF_MED_EN		Auto focus median filter enable. If the median filter is enabled, then the first two and the last two pixels in the frame are not in the valid region. Therefore, the paxel start/end and IIR filter start positions should not be set within the first two and the last two pixels.
		0	Disable auto focus median filter.
		1	Enable auto focus median filter.
1	AF_ALAW_EN		Auto focus A-law table enable.
		0	Auto focus A-law table is disabled.
		1	Auto focus A-law table is enabled.
0	AF_EN		Auto focus engine enable.
		0	Auto focus engine is disabled.
		1	Auto focus engine is enabled.

6.5.3 Setup for the AF Engine Paxel Configuration Register (AFPAX1)

The setup for the AF engine paxel configuration register (AFPAX1) is shown in Figure 155 and described in Table 148.

Figure 155. Setup for the AF Engine Paxel Configuration Register (AFPAX1)

31 23	22 16
Reserved	PAXW
R-0	R/W-0

15 7	6 0
Reserved	PAXH
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 148. Setup for the AF Engine Paxel Configuration Register (AFPAX1) Field Descriptions

Bit	Field	Value	Description
31-23	Reserved	0	Reserved
22-16	PAXW	0-7Fh	AF engine paxel width. The width of the paxel is (PAXW + 1) × 2. The minimum width is 6 pixels.
15-7	Reserved	0	Reserved
6-0	PAXH	0-7Fh	AF engine paxel height. The height of the paxel is (PAXH + 1) × 2. Valid values are 2 to 256 (even values).

6.5.4 Setup for the AF Engine Paxel Configuration Register (AFPAX2)

The setup for the AF engine paxel configuration register (AFPAX2) is shown in Figure 156 and described in Table 149.

Figure 156. Setup for the AF Engine Paxel Configuration Register (AFPAX2)

31 17	16
Reserved	AFINCV
R-0	R/W-0

15 13	12 6	5 0
AFINCV	PAXVC	PAXHC
R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 149. Setup for the AF Engine Paxel Configuration Register (AFPAX2) Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved
16-13	AFINCV	0-Fh	AF engine line increments. Number of lines to skip in a paxel plus 1 multiplied by 2. Incrementing the line in a paxel is always done on a line pair, due to the fact that the RGB pattern falls in two lines. If all the lines are to be processed, AFINCF should be cleared to 0, and thus, line count is incremented by 2 following a line pair. Thus, the number of lines that can be skipped between two processed line pairs is 0 to 30 (even). The starting two lines in a paxel are first processed before this field is applied.
12-6	PAXVC	0-7Fh	AF engine vertical paxel count. The number of paxels in the vertical direction plus 1. The maximum number of vertical paxels in a frame should not exceed 128. The value should be set to ensure that the bandwidth requirements and buffer size are not exceeded.
5-0	PAXHC	0-3Fh	AF engine horizontal paxel count. The number of paxels in the horizontal direction plus 1. It is invalid to set a number that is greater than 35 (total of 36 paxels in the horizontal direction).

6.5.5 Start Position for AF Engine Paxels Register (AFPAXSTART)

The start position for AF engine paxels register (AFPAXSTART) is shown in Figure 157 and described in Table 150.

Figure 157. Start Position for AF Engine Paxels Register (AFPAXSTART)

31 28	27 16
Reserved	PAXSH
R-0	R/W-0

15 12	11 0
Reserved	PAXSV
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 150. Start Position for AF Engine Paxels Register (AFPAXSTART) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	PAXSH	0-FFFh	AF engine paxel horizontal start position. PAXSH must be equal to or greater than (IIRSH + 1). This value must be even.
15-12	Reserved	0-Fh	Reserved
11-0	PAXSV	0-FFFh	AF engine paxel vertical start position. Sets the vertical line for the first paxel.

6.5.6 Start Position for IIRSH Register (AFIIRSH)

The start position for IIRSH register (AFIIRSH) is shown in Figure 158 and described in Table 151.

Figure 158. Start Position for IIRSH Register (AFIIRSH)

31 16
Reserved
R-0

15 12	11 0
Reserved	IIRSH
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 151. Start Position for IIRSH Register (AFIIRSH) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reserved
11-0	IIRSH	0-FFFh	AF engine IIR horizontal start position. When the horizontal position of a line equals this value, the shift registers are cleared on the next pixel. This value must be even.

6.5.7 SDRAM/DDRAM Start Address for AF Engine Register (AFBUFST)

The SDRAM/DDRAM start address for AF engine register (AFBUFST) is shown in Figure 159 and described in Table 152.

Figure 159. SDRAM/DDRAM Start Address for AF Engine Register (AFBUFST)

31 — 0
AFBUFST
R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 152. SDRAM/DDRAM Start Address for AF Engine Register (AFBUFST) Field Descriptions

Bit	Field	Value	Description
31-0	AFBUFST	0-FFFF FFFFh	AF engine SDRAM/DDRAM start address. The starting location in SDRAM/DDRAM. The 6 LSBs are ignored, address should be on a 64-byte boundary. AFBUFST can be altered even when the AF engine is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

6.5.8 IIR Filter Coefficient Data for Set 0 Register (AFCOEFF010)

The IIR filter coefficient data for set 0 register (AFCOEFF010) is shown in Figure 160 and described in Table 153.

Figure 160. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF010)

31 — 28	27 — 16
Reserved	COEFF1
R-0	R/W-0

15 — 12	11 — 0
Reserved	COEFF0
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 153. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF010) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF1	0-FFFh	AF engine IIR filter coefficient #1 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF0	0-FFFh	AF engine IIR filter coefficient #0 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.9 IIR Filter Coefficient Data for Set 0 Register (AFCOEFF032)

The IIR filter coefficient data for set 0 register (AFCOEFF032) is shown in Figure 161 and described in Table 154.

Figure 161. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF032)

31 - 28	27 - 16
Reserved	COEFF3
R-0	R/W-0

15 - 12	11 - 0
Reserved	COEFF2
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 154. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF032) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF3	0-FFFh	AF engine IIR filter coefficient #3 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF2	0-FFFh	AF engine IIR filter coefficient #2 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.10 IIR Filter Coefficient Data for Set 0 Register (AFCOEFF054)

The IIR filter coefficient data for set 0 register (AFCOEFF054) is shown in Figure 162 and described in Table 155.

Figure 162. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF054)

31 - 28	27 - 16
Reserved	COEFF5
R-0	R/W-0

15 - 12	11 - 0
Reserved	COEFF4
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 155. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF054) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF5	0-FFFh	AF engine IIR filter coefficient #5 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF4	0-FFFh	AF engine IIR filter coefficient #4 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.11 IIR Filter Coefficient Data for Set 0 Register (AFCOEFF076)

The IIR filter coefficient data for set 0 register (AFCOEFF076) is shown in Figure 163 and described in Table 156.

Figure 163. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF076)

31 28	27 16
Reserved	COEFF7
R-0	R/W-0

15 12	11 0
Reserved	COEFF6
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 156. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF076) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF7	0-FFFh	AF engine IIR filter coefficient #7 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF6	0-FFFh	AF engine IIR filter coefficient #6 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.12 IIR Filter Coefficient Data for Set 0 Register (AFCOEFF098)

The IIR filter coefficient data for set 0 register (AFCOEFF098) is shown in Figure 164 and described in Table 157.

Figure 164. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF098)

31 28	27 16
Reserved	COEFF9
R-0	R/W-0

15 12	11 0
Reserved	COEFF8
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 157. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF098) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF9	0-FFFh	AF engine IIR filter coefficient #9 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF8	0-FFFh	AF engine IIR filter coefficient #8 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.13 IIR Filter Coefficient Data for Set 0 Register (AFCOEFF0010)

The IIR filter coefficient data for set 0 register (AFCOEFF0010) is shown in Figure 165 and described in Table 158.

Figure 165. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF0010)

31 16
Reserved
R-0

15 12	11 0
Reserved	COEFF10
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 158. IIR Filter Coefficient Data for Set 0 Register (AFCOEFF0010) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reserved
11-0	COEFF10	0-FFFh	AF engine IIR filter coefficient #10 (set 0). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.14 IIR Filter Coefficient Data for Set 1 Register (AFCOEFF110)

The IIR filter coefficient data for set 1 register (AFCOEFF110) is shown in Figure 166 and described in Table 159.

Figure 166. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF110)

31 28	27 16
Reserved	COEFF1
R-0	R/W-0

15 12	11 0
Reserved	COEFF0
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 159. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF110) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF1	0-FFFh	AF engine IIR filter coefficient #1 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF0	0-FFFh	AF engine IIR filter coefficient #0 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.15 IIR Filter Coefficient Data for Set 1 Register (AFCOEFF132)

The IIR filter coefficient data for set 1 register (AFCOEFF132) is shown in Figure 167 and described in Table 160.

Figure 167. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF132)

31-28	27-16
Reserved	COEFF3
R-0	R/W-0

15-12	11-0
Reserved	COEFF2
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 160. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF132) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF3	0-FFFh	AF engine IIR filter coefficient #3 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF2	0-FFFh	AF engine IIR filter coefficient #2 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.16 IIR Filter Coefficient Data for Set 1 Register (AFCOEFF154)

The IIR filter coefficient data for set 1 register (AFCOEFF154) is shown in Figure 168 and described in Table 161.

Figure 168. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF154)

31-28	27-16
Reserved	COEFF5
R-0	R/W-0

15-12	11-0
Reserved	COEFF4
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 161. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF154) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF5	0-FFFh	AF engine IIR filter coefficient #5 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF4	0-FFFh	AF engine IIR filter coefficient #4 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.17 IIR Filter Coefficient Data for Set 1 Register (AFCOEFF176)

The IIR filter coefficient data for set 1 register (AFCOEFF176) is shown in Figure 169 and described in Table 162.

Figure 169. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF176)

31 28	27 16
Reserved	COEFF7
R-0	R/W-0

15 12	11 0
Reserved	COEFF6
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 162. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF176) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF7	0-FFFh	AF engine IIR filter coefficient #7 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF6	0-FFFh	AF engine IIR filter coefficient #6 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.18 IIR Filter Coefficient Data for Set 1 Register (AFCOEFF198)

The IIR filter coefficient data for set 1 register (AFCOEFF198) is shown in Figure 170 and described in Table 163.

Figure 170. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF198)

31 28	27 16
Reserved	COEFF9
R-0	R/W-0

15 12	11 0
Reserved	COEFF8
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 163. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF198) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	COEFF9	0-FFFh	AF engine IIR filter coefficient #9 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).
15-12	Reserved	0	Reserved
11-0	COEFF8	0-FFFh	AF engine IIR filter coefficient #8 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.19 IIR Filter Coefficient Data for Set 1 Register (AFCOEFF1010)

The IIR filter coefficient data for set 1 register (AFCOEFF1010) is shown in Figure 171 and described in Table 164.

Figure 171. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF1010)

31 16
Reserved
R-0

15 12	11 0
Reserved	COEFF10
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 164. IIR Filter Coefficient Data for Set 1 Register (AFCOEFF1010) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reserved
11-0	COEFF10	0-FFFh	AF engine IIR filter coefficient #10 (set 1). The range is signed -32 ≤ value ≤ 31 (+63/64).

6.5.20 Configuration for AE/AWB Windows Register (AEWWIN1)

The configuration for AE/AWB windows register (AEWWIN1) is shown in Figure 172 and described in Table 165.

Figure 172. Configuration for AE/AWB Windows Register (AEWWIN1)

31	30 24	23 20	19 16
Rsvd	WINH	Reserved	WINW
R-0	R/W-0	R-0	R/W-0

15 13	12 6	5 0
WINW	WINVC	WINHC
R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 165. Configuration for AE/AWB Windows Register (AEWWIN1) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-24	WINH	0-7Fh	AE/AWB engine window height. Specifies the window height in an even number of pixels, the window height is (WINH + 1) × 2. The final value can be from 2 to 256 (even value).
23-20	Reserved	0	Reserved
19-13	WINW	0-7Fh	AE/AWB engine window width. This specifies the window width in an even number of pixels, the window width is (WINW + 1) × 2. The minimum width is 6 pixels.
12-6	WINVC	0-7Fh	AE/AWB engine vertical window count. The number of windows in the vertical direction plus 1. The maximum number of vertical windows in a frame should not exceed 128. The value should be set to ensure that the bandwidth requirements and buffer size are not exceeded.
5-0	WINHC	0-3Fh	AE/AWB engine horizontal window count. The number of horizontal windows plus 1. The maximum number of horizontal windows is 35 plus 1 (36). The minimum number of windows should be 2 (valid range for the field is 1 to 35).

6.5.21 Start Position for AE/AWB Windows Register (AEWINSTART)

The start position for AE/AWB windows register (AEWINSTART) is shown in Figure 173 and described in Table 166.

Figure 173. Start Position for AE/AWB Windows Register (AEWINSTART)

31 28	27 16
Reserved	WINSV
R-0	R/W-0

15 12	11 0
Reserved	WINSH
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 166. Start Position for AE/AWB Windows Register (AEWINSTART) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	WINSV	0-FFFh	AE/AWB engine vertical window start position. Sets the first line for the first window.
15-12	Reserved	0	Reserved
11-0	WINSH	0-FFFh	AE/AWB engine horizontal window start position. Sets the horizontal position for the first window on each line.

6.5.22 Start Position and Height for Black Line of AE/AWB Windows Register (AEWINBLK)

The start position and height for black line of AE/AWB windows register (AEWINBLK) is shown in Figure 174 and described in Table 167.

Figure 174. Start Position and Height for Black Line of AE/AWB Windows Register (AEWINBLK)

31 28	27 16
Reserved	WINSV
R-0	R/W-0

15 7	6 0
Reserved	WINH
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 167. Start Position and Height for Black Line of AE/AWB Windows Register (AEWINBLK) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	WINSV	0-FFFh	AE/AWB engine vertical window start position for single black line of windows. Sets the first line for the single black line of windows. The horizontal start and the horizontal number of windows will be similar to the regular windows.
15-7	Reserved	0	Reserved
6-0	WINH	0-7Fh	AE/AWB engine window height for the single black line of windows. Specifies the window height in an even number of pixels, the window height is (WINH + 1) × 2. The final value can be from 2 to 256 (even value).

6.5.23 Configuration for Subsample Data in AE/AWB Window Register (AEWSUBWIN)

The configuration for subsample data in AE/AWB window register (AEWSUBWIN) is shown in Figure 175 and described in Table 168.

Figure 175. Configuration for Subsample Data in AE/AWB Window Register (AEWSUBWIN)

31 16
Reserved
R-0

15 12	11 8	7 4	3 0
Reserved	AEWINCV	Reserved	AEWINCH
R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 168. Configuration for Subsample Data in AE/AWB Window Register (AEWSUBWIN) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reserved
11-8	AEWINCV	0-Fh	AE/AWB engine vertical sampling point increment. Sets vertical distance between subsamples within a window plus 1 multiplied by 2. The final range is 2 to 32.
7-4	Reserved	0	Reserved
3-0	AEWINCH	0-Fh	AE/AWB engine horizontal sampling point increment. Sets horizontal distance between subsamples within a window plus 1 multiplied by 2. The final range is 2 to 32.

6.5.24 SDRAM/DDRAM Start Address for AE/AWB Engine Register (AEWBUFST)

The SDRAM/DDRAM start address for AE/AWB engine register (AEWBUFST) is shown in Figure 176 and described in Table 169.

Figure 176. SDRAM/DDRAM Start Address for AE/AWB Engine Register (AEWBUFST)

31 0
AEWBUFST
R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 169. SDRAM/DDRAM Start Address for AE/AWB Engine Register (AEWBUFST) Field Descriptions

Bit	Field	Value	Description
31-0	AEWBUFST	0-FFFF FFFFh	AE/AWB engine SDRAM/DDRAM start address. The starting location in SDRAM/DDRAM for the AE/AWB data. The 6 LSBs are ignored, address should be on a 64-byte boundary. AEWBUFST can be altered even when the AE/AWB engine is busy. The change will take place only for the next frame. Note that reading this register will always give the latest value.

7 Video Processing Subsystem (VPSS) Registers

This section discusses the registers in the video processor subsystem (VPSS). Table 170 lists the memory-mapped registers for the VPSS. See the device-specific data manual for the memory address of these registers.

Table 170. Video Processing Subsystem (VPSS) Registers

Offset	Acronym	Register Description	Section
3400h	PID	VPSS Peripheral Revision and Class Information	Section 7.1
3404h	PCR	VPSS Peripheral Control Register	Section 7.2
3508h	SDR_REQ_EXP	SDRAM Non-Real-Time Read Request Expand Register	Section 7.3

7.1 VPSS Peripheral Revision and Class Information Register (PID)

The VPSS peripheral revision and class information register (PID) is shown in Figure 177 and described in Table 171.

Figure 177. VPSS Peripheral Revision and Class Information Register (PID)

31 - 24	23 - 16
Reserved	TID
R-0	R-1

15 - 8	7 - 0
CID	PREV
R-FBh	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 171. VPSS Peripheral Revision and Class Information Register (PID) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-16	TID	0-FFh	Peripheral identification
		1	VPSS
15-8	CID	0-FFh	Class identification
		FBh	VPSS
7-0	PREV	0-FFh	Peripheral revision number
		0	Initial revision

7.2 VPSS Peripheral Control Register (PCR)

The VPSS peripheral control register (PCR) is shown in Figure 178 and described in Table 172.

Figure 178. VPSS Peripheral Control Register (PCR)

31 – 24
Reserved
R-0

23	22	21	20	19	18	17	16
CCDC_WBL_O	PRV_WBL_O	RSZ1_WBL_O	RSZ2_WBL_O	RSZ3_WBL_O	RSZ4_WBL_O	AF_WBL_O	AEW_WBL_O
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15 – 4	3 – 0
Reserved	DMA_PRI
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 172. VPSS Peripheral Control Register (PCR) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23	CCDC_WBL_O		Write buffer memory overflow (CCDC). All data units have been filled and not yet transferred to SDRAM before the next data unit is to be filled from the WBL. Software has to clear the bit.
		0	No overflow.
		1	Overflow (fail).
22	PRV_WBL_O		Write buffer memory overflow (Preview engine). All data units have been filled and not yet transferred to SDRAM before the next data unit is to be filled from the WBL. Software has to clear the bit.
		0	No overflow.
		1	Overflow (fail).
21	RSZ1_WBL_O		Write buffer memory overflow (Resizer line 1). All data units have been filled and not yet transferred to SDRAM before the next data unit is to be filled from the WBL. Software has to clear the bit.
		0	No overflow.
		1	Overflow (fail).
20	RSZ2_WBL_O		Write buffer memory overflow (Resizer line 2). All data units have been filled and not yet transferred to SDRAM before the next data unit is to be filled from the WBL. Software has to clear the bit.
		0	No overflow.
		1	Overflow (fail).
19	RSZ3_WBL_O		Write buffer memory overflow (Resizer line 3). All data units have been filled and not yet transferred to SDRAM before the next data unit is to be filled from the WBL. Software has to clear the bit.
		0	No overflow.
		1	Overflow (fail).
18	RSZ4_WBL_O		Write buffer memory overflow (Resizer line 4). All data units have been filled and not yet transferred to SDRAM before the next data unit is to be filled from the WBL. Software has to clear the bit.
		0	No overflow.
		1	Overflow (fail).
17	AF_WBL_O		Write buffer memory overflow (AF). All data units have been filled and not yet transferred to SDRAM before the next data unit is to be filled from the WBL. Software has to clear the bit.
		0	No overflow.
		1	Overflow (fail).

Table 172. VPSS Peripheral Control Register (PCR) Field Descriptions (continued)

Bit	Field	Value	Description
16	AEW_WBL_O		Write buffer memory overflow (AE/AWB). All data units have been filled and not yet transferred to SDRAM before the next data unit is to be filled from the WBL. Software has to clear the bit.
		0	No overflow.
		1	Overflow (fail).
15-4	Reserved	0	Reserved
3-0	DMI_PRI	0-Fh	VBUSM priority in the system to the DDR EMIF. The default should be the highest priority in the system.

7.3 SDRAM Non-Real-Time Read Request Expand Register (SDR_REQ_EXP)

The SDRAM non real-time read request expand (SDR_REQ_EXP) register is shown in Figure 179 and described in Table 173.

Figure 179. SDRAM Non-Real-Time Read Request Expand Register (SDR_REQ_EXP)

31	30	29	20	19	16
Reserved		PRV_EXP		RESZ_EXP	
R-0		R/W-0		R/W-0	

15	10	9	0
RESZ_EXP		HIST_EXP	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 173. SDRAM Non-Real-Time Read Request Expand Register (SDR_REQ_EXP) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-20	PRV_EXP	0-3FFh	Preview read request expand. Delay to allow between consecutive read requests from the preview module. Units are in VPSS clock cycles. Since the VPSS DMA priority is typically set to the highest for real-time requirements, this is for spreading any non-real-time reads with respect to the other traffic in the system. This minimizes the potential of locking out other requests for the duration of a frame being read from DDR/SDR.
19-10	RESZ_EXP	0-3FFh	Resizer read request expand. Delay to allow between consecutive read requests from the resizer module. Units are 32 VPSS clock cycles. The delay is RESZ_EXP × 32 VPSS clock cycles. Since the VPSS DMA priority is typically set to the highest for real-time requirements, this is for spreading any non-real-time reads with respect to the other traffic in the system. This minimizes the potential of locking out other requests for the duration of a frame being read from DDR/SDR.
9-0	HIST_EXP	0-3FFh	Histogram read request expand. Delay to allow between consecutive read requests from the histogram module. Units are in VPSS clock cycles. Since the VPSS DMA priority is typically set to the highest for real-time requirements, this is for spreading any non-real-time reads with respect to the other traffic in the system. This minimizes the potential of locking out other requests for the duration of a frame being read from DDR/SDR.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2008, Texas Instruments Incorporated