VC0568

Mobile Phone Video Processor

Datasheet

Version 1.1

Mar 11th, 2005

Notes1: The information is subject to change without notice. Before using this document, please confirm that this is the latest version.
Notes2: Not all products and/or types are available in every country. Please check with a Vimicro sales representative availability and additional information.

Important Notice

Vimicro Corporation and its subsidiaries 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 Vimicro' terms and conditions of sale supplied at the time of order acknowledgment.

Vimicro does not warrant or represent that any license, either express or implied, is granted under any Vimicro patent right, copyright, mask work right, or other Vimicro intellectual property right relating to any combination, machine, or process in which Vimicro products or services are used. Information published by Vimicro regarding third-party products or services does not constitute a license from Vimicro to use such products or service 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 Vimicro under the patents or other intellectual property of Vimicro.

Reproduction of information in Vimicro 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. Vimicro is not responsible or liable for such altered documentation.

Resale of Vimicro products or services with statements different from or beyond the parameters stated by Vimicro for that product or service voids all express and any implied warranties for the associated Vimicro product or service and is an unfair and deceptive business practice. Vimicro is not responsible or liable for any such statements.

Table of Contents

1. Overview

VC0568 is a mega-pixel camera controller chip for mobile application. It enables camera phone to capture and display still image and video clip at real-time on mobile phone screen. By using VC0568, the burden on mobile phone host CPU to process image and graphics can be reduced dramatically.

With direct interface to CMOS/CCD image sensor and LCD display module, VC0568 can provide high-quality, low-latency video stream to most popular LCD panels with desired resolution and color depth. VC0568's image pipeline integrates Image Signal Processing unit, Image Post Processing unit, JPEG codec as well as JPEG buffer allocated from the on-chip 384KB SRAM for storing 1.3M-resolution JPEG images. A powerful yet flexible image sizer embedded in VC0568 further allows mobile phone users to perform up to 16x digital zoom and pan function in viewfinder, capture and display mode. The on-chip JPEG codec may also be employed by host CPU for various encoding/decoding tasks.

Within its display sub-module, VC0568 provides Graphics RAM allocated from the on-chip 384KB SRAM for host CPU to write to LCD panel directly. VC0568 supports overlay and alpha-blending functions between video stream and the graphics from host CPU, along with an integrated graphics engine delivering graphics acceleration for clipping, line drawing, BitBLT, etc. VC0568 also provides a direct path from host CPU to LCD panel.

VC0568 comes with 100-pin fpBGA package at 7mm x 7mm footprint.

VC0568 has the following applications

- Mobile Phones
- Personal Digital Assistants
- Wireless Communicators
- Video Phones

2. Feature

- **System Architecture**
  - Provide a transparent, flexible physical bridge between mobile phone baseband processor, image sensor and LCD panel
  - With camera function off, host CPU can refresh LCD directly
  - To add camera function, mobile phone design only need minimal change with the help of VC0568 and a API set provided by Vimicro

- **Low Power Design**
  - Power Consumption is optimized from Architecture to Circuit Design
  - 100uA in sleep mode
  - 20mA in preview mode in 160x120 @ 30fps
  - 1.8V for core
  - 1.5, 1.8, 2.5, 2.85, 3.0, 3.3 V for I/O

- **Small Form Factor**
  - 7 mm x 7 mm

- **Host Interface**
  - Support ARM, 80-type and 68-type bus
  - In-direct addressing mode:
    - 1-bit address bus and 16-bit data bus
    - 1-bit address bus and 8-bit data bus

- **Sensor Interface**
  - Support off-shelf CMOS and CCD sensors up to 1.3-mega pixels
  - 8/10 bit RGB Bayer or 8 bit YUV input
  - Strobe flash timing control

- **Image Signal Processing**
  - Real-time dead-pixel detection and compensation
  - Auto exposure
  - Auto white balance
  - Auto gain control
  - Auto focus control
  - Programmable AE/AWB windows
  - Auto flicker detection and cancellation
  - Edge-adaptive CFA interpolation (5x5 matrix)
  - Configurable gamma correction
  - Configurable color correction
  - Configurable white balancing
  - Configurable brightness, color saturation and hue

- Back light compensation
- Configurable image noise reduction
- Configurable image sharpness control

- **JPEG Codec**
  - Standard baseline JPEG codec with dynamic bit-rate control
  - Encoding: 4:2:2, 4:1:1, 4:2:0
  - Decoding: 4:4:4, 4:2:2, 4:1:1, 4:2:0
  - Configurable Huffman table
  - Configurable quantization table

- **Image Sizer and Digital Zoom**
  - Programmable image sub-sampling with anti-aliasing filtering
  - Programmable image dithering
  - Up to 16x digital zoom during preview, capture, postview
    - Up to 2x for SXGA
    - Up to 4x for VGA
    - Up to 8x for SIF
    - Up to 16x for QSIF

- **Programmable Image Resolution and Data Format**
  - Programmable output image resolution including standard resolutions:
    - 1280 x 1024
    - 1280 x 960
    - 1024 x 768
    - 800 x 600
    - 640 x 480
    - 320 x 240
    - 160 x 120
    - Thumbnail (40x30)
  - Multiple output image formats
    - JPEG
    - RGB: 888, 666, 565, 444
    - Gray

- **LCD Controller Interface**
  - Support dual LCD panels
  - Support TFT or STN LCD panels
  - Supports LCD with following bit-depth: 8/12/16/18/24
  - Supports LCD with following size: 160 x 120 ~ 320 x 240
  - Support Picture-in-Picture function
  - Separate controls for layer A and layer B
  - Backlight white LED control
  - Illumination LED control

- **2-D Hardware Acceleration Engine**
    - Enhanced BitBLT
    - ROPs
    - All-angle (Bresenham) line drawing
    - Clip

- **Video Post-processing**
    - Overlay between video and host graphics
    - Alpha-blending between video and host graphics
    - Image rotation (90, 180 and 270 degree)
    - Mirror (horizontal and vertical)
    - Special image effects:
        - Monochrome
        - Sepia
        - Special color
        - Negative
        - Relief
        - Sketch

3. Pin Layout

3.1 VC0568 Pin-out Diagram

	1	2	3	4	5	6	7	8	9	10	11	12
A	CONF[2]	LCD_ DATA[0]	OVDD8	RSTN	CS_ CLK	CS_ VSYNC	LCD_ RDN	LCD_ DATA [14]	LCD_ DATA [12]	OVDD7	CPU_ ADDR[5]	CPU_ A8
B	LCD_ DATA[1]	OVSS8	CONF[1]	INT	CS_ RSTN	DVSS4	DVDD4	LCD_ DATA [13]	CPU_ ADDR[7]	CPU_ ADDR[6]	CPU_ ADDR[4]	CPU_ RS
C	CPU_ DATA[1]	CPU_ DATA[0]									CS_ ENB	OVDD6
D	OVDD1	CPU_ DATA[2]		CONF[0]	CS_ HSYNC	LCD_ WRN	LCD_ DATA [15]	OVSS7	OVSS6		LCD_ RS	CS_ PWDN
E	LCD_ DATA[2]	CPU_ DATA[3]		OVSS1					LCD_ RST		CS_ SCK	CS_ SDA
F	GPIO[1]	GPIO[0]		LCD_ DATA[3]					DVSS3		DVDD3	XCLK
G	CPU_ WBEN[0]	DVDD1		DVSS1					LCD_ DATA [11]		VDDA	VSSA
H	CPU_ WEN	CPU_ OEN		CPU_ WBEN[1]					OVSS5		LCD_ DATA[9]	LCD_ DATA [10]
J	CS_ DATA[6]	CS_ DATA[7]		CS_ DATA[5]	OVSS3	LCD_ DATA[7]	GPIO[4]	LCD_ CS1N	OVSS4		LCD_ DATA[8]	OVDD5
K	OVDD2	CS_ DATA[4]									CPU_ ADDR[2]	CPU_ ADDR[3]
L	LCD_ DATA[4]	OVSS2	CPU_ DATA[4]	CPU_ DATA[6]	CPU_ DATA[7]	DVDD2	GPIO[2]	CPU_ CS1N	CPU_ CS2N	CS_ DATA[2]	CPU_ ADDR[0]	CPU_ ADDR[1]
M	TEST	LCD_ DATA[5]	CPU_ DATA[5]	OVDD3	LCD_ DATA[6]	DVSS2	GPIO[3]	LCD_ CS2N	OVDD4	CS_ DATA[3]	CS_ DATA[1]	CS_ DATA[0]

Figure 3-1 VC0568 fpBGA-100 Pinout Diagram – Top View

3.2 Pin description

Pin Name	I/O TYPE	Ball Pad	Remark
LCD_DATA[0]	B	A2	Data bus to the LCD panel, bit [0]
LCD_DATA[1]	B	B1	Data bus to the LCD panel, bit [1]
CPU_DATA[0]	B	C2	Data bus between host CPU and VC0568, bit [0]
CPU_DATA[1]	B	C1	Data bus between host CPU and VC0568, bit [1]
OVSS1	G	E4	IO ground
OVDD1	P	D1	IO power
CPU_DATA[2]	B	D2	Data bus between host CPU and VC0568, bit [2]
CPU_DATA[3]	B	E2	Data bus between host CPU and VC0568, bit [3]
LCD_DATA[2]	B	E1	Data bus to the LCD panel, bit [2]
LCD_DATA[3]	B	F4	Data bus to the LCD panel, bit [3]
GPIO[0]	B, PU	F2	Global purpose IO pin, bit [0]
GPIO[1]	B, PU	F1	Global purpose IO pin, bit [1]
DVSS1	G	G4	Core ground
DVDD1	P	G2	Core power
CPU_WBEN[0]	I	G1	Write byte enable, bit 0
CPU_WBEN[1]	I	H4	Write byte enable, bit 1
CPU_WEN	I	H1	Write signal, active low
CPU_OEN	I	H2	Output enable, active low
CS_DATA[7]	B, PD	J2	Video data input from camera module, bit [7]
CS_DATA[6]	B, PD	J1	Video data input from camera module, bit [6]
CS_DATA[5]	B, PD	J4	Video data input from camera module, bit [5]
CS_DATA[4]	B, PD	K2	Video data input from camera module, bit [4]
OVDD2	P	K1	IO power
OVSS2	G	L2	IO ground
TEST	I	M1	Test mode enable
LCD_DATA[4]	B	L1	Data bus to the LCD panel, bit [4]
LCD_DATA[5]	B	M2	Data bus to the LCD panel, bit [5]
CPU_DATA[4]	B	L3	Data bus between host CPU and VC0568, bit [4]
CPU_DATA[5]	B	M3	Data bus between host CPU and VC0568, bit [5]
OVSS3	G	J5	IO ground
OVDD3	P	M4	IO power
CPU_DATA[6]	B	L4	Data bus between host CPU and VC0568, bit

			[6]
CPU_DATA[7]	B,	L5	Data bus between host CPU and VC0568, bit [7]
LCD_DATA[6]	B	M5	Data bus to the LCD panel, bit [6]
LCD_DATA[7]	B	J6	Data bus to the LCD panel, bit [7]
DVSS2	G	M6	Core ground
DVDD2	P	L6	Core power
GPIO[2]	B, PU	L7	Global purpose IO pin, bit [2]
GPIO[3]	B, PU	M7	Global purpose IO pin, bit [3]
GPIO[4]	B, PU	J7	Global purpose IO pin, bit [4]
LCD_CS1N	B	J8	LCD Chip Select for main panel
LCD_CS2N	B	M8	LCD Chip Select for sub panel
CPU_CS1N	I	L8	Chip select 1, active low
CPU_CS2N	I	L9	Chip select 2, active low
OVDD4	P	M9	IO power
OVSS4	G	J9	IO ground
CS_DATA[3]	B, PD	M10	Video data input from camera module, bit [3]
CS_DATA[2]	B, PD	L10	Video data input from camera module, bit [2]
CS_DATA[1]	B, PD	M11	Video data input from camera module, bit [1]
CS_DATA[0]	B, PD	M12	Video data input from camera module, bit [0]
CPU_ADDR[0]	B	L11	Address bus between host CPU and VC0568, bit [0]
CPU_ADDR[1]	B	L12	Address bus between host CPU and VC0568, bit [1]
CPU_ADDR[2]	B	K11	Address bus between host CPU and VC0568, bit [2]
CPU_ADDR[3]	B	K12	Address bus between host CPU and VC0568, bit [3]
OVSS5	G	H9	IO ground
OVDD5	P	J12	IO power
LCD_DATA[8]	B	J11	Data bus between host CPU and VC0568, bit [8]
LCD_DATA[9]	B	H11	Data bus between host CPU and VC0568, bit [9]
LCD_DATA[10]	B	H12	Data bus between host CPU and VC0568, bit [10]
LCD_DATA[11]	B	G9	Data bus between host CPU and VC0568, bit [11]
VDDA	P	G11	PLL power
VSSA	G	G12	PLL ground
DVSS3	G	F9	Core ground
DVDD3	P	F11	Core power
CS_SCK	B	E11	Serial clock for camera module control

CS_SDA	B	E12	Serial data for camera module control
LCD_RST	B	E9	LCD panel reset
LCD_RS	B	D11	LCD Register Select index and command register
CS_PWDN	B	D12	Power down signal of camera module
CS_ENB	B	C11	Camera module enable signal
OVDD6	P	C12	IO power
XCLK	I	F12	Clock from the host CPU or soxillator
OVSS6	G	D9	IO ground
CPU_RS	I, PD	B12	Register select
CPU_A8	I, PD	A12	Address bus / HOST register select
CPU_ADDR[4]	B	B11	Address bus between host CPU and VC0568, bit [4]
CPU_ADDR[5]	B	A11	Address bus between host CPU and VC0568, bit [5]
CPU_ADDR[6]	B	B10	Address bus between host CPU and VC0568, bit [6]
CPU_ADDR[7]	B	B9	Address bus between host CPU and VC0568, bit [7]
OVDD7	P	A10	IO power
OVSS7	G	D8	IO ground
LCD_DATA[12]	B	A9	Data bus between host CPU and VC0568, bit [12]
LCD_DATA[13]	B	B8	Data bus between host CPU and VC0568, bit [13]
LCD_DATA[14]	B	A8	Data bus between host CPU and VC0568, bit [14]
LCD_DATA[15]	B	D7	Data bus between host CPU and VC0568, bit [15]
LCD_WRN	B	D6	LCD Write strobe to be asserted in write operation
LCD_RDN	B	A7	LCD Read strobe to be asserted in read operation
DVDD4	P	B7	Core power
DVSS4	G	B6	Core ground
CS_VSYNC	B, PD	A6	Vertical synchronous signal to/from camera module
CS_HSYNC	B, PD	D5	Horizontal synchronous signal to/from camera module
CS_CLK	B,	A5	Synchronous clock of camera data transfer (max.27MHz)
CS_RSTN	B,	B5	Camera module reset signal
INT	B,	B4	Interrupt to the host CPU

RSTN	I, S	A4	Reset signal, active low
OVDD8	P	A3	IO power
OVSS8	G	B2	IO ground
CONF[0]	I	D4	Configuration pin for CPU bus, bit 0
CONF[1]	I	B3	Configuration pin for CPU bus, bit 1
CONF[2]	I	A1	Configuration pin for CPU bus, bit 2

I = Input
O = Output
B = Bidirectional
P = Power
G = Ground
PD = Pull down
PU = Pull up
S = Schmitt trigger

4. System block diagram

Figure 4-1 shows the system block diagram of a typical camera phone. VC0568 is the bridge between the baseband processor and LCD module, enabling mobile phones to capture and display still images and video clips at real-time. Most of the display and camera functions are conducted by VC0568, which will greatly reduce the burden on host CPU for computation-intensive image and graphics processing.

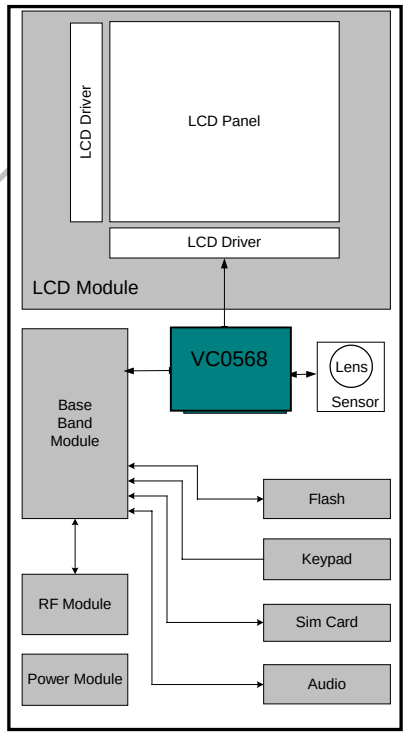


Figure 4-1 VC0568 system block diagram

5. Chip block diagram

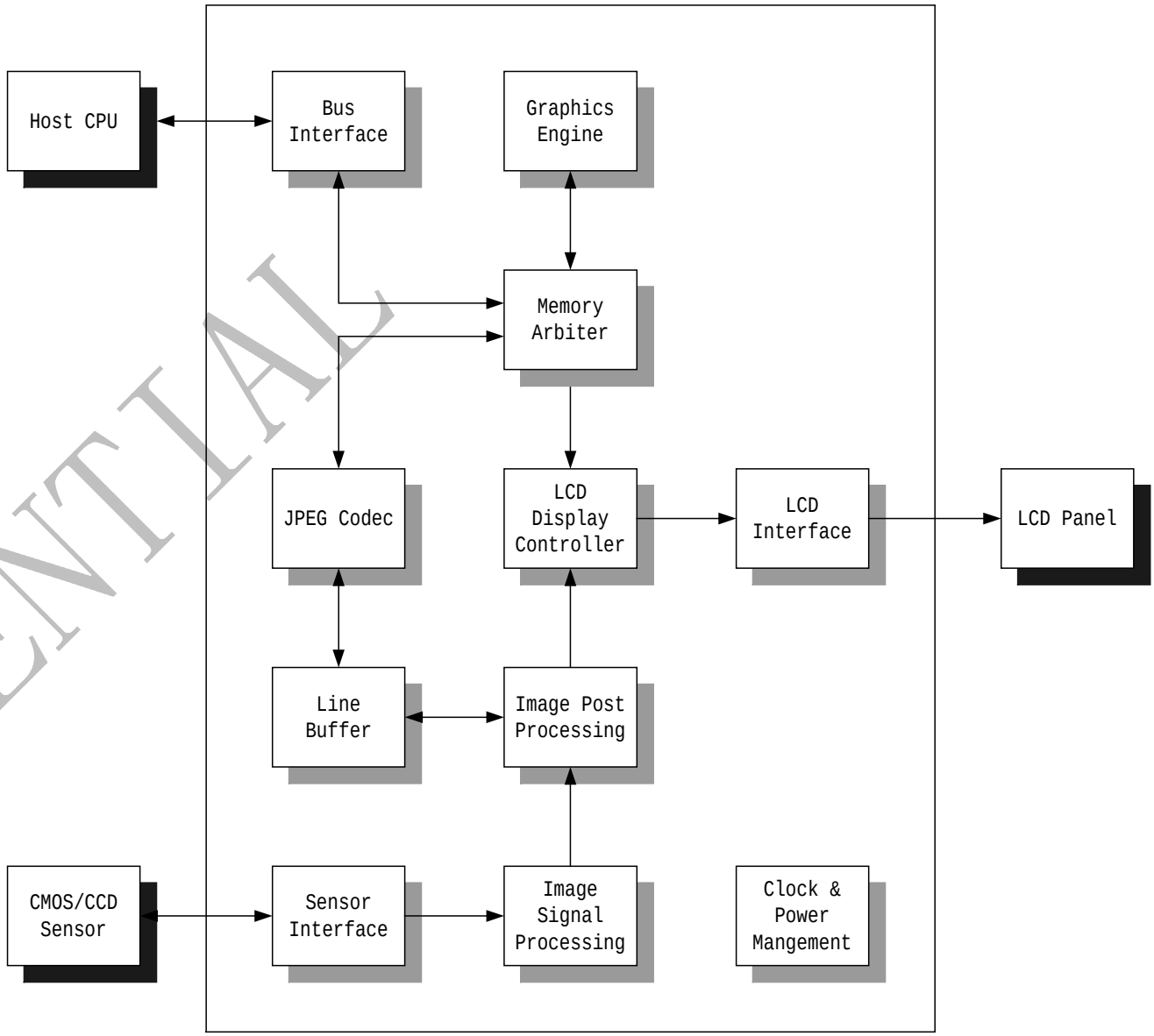


Figure 5-1 VC0568 Block Diagram

The main internal blocks in VC0568 are:

- **Sensor interface (SIF)**

Image data in YUV422 format or RGB Bayer pattern is captured synchronously. The sensor clock is generated by VC0568 and it can be adjusted accordingly up to 27 MHz, while the pixel rate can reach 13.5 MHz. VC0568 supports up to 1280 by 1024 image resolution from the sensor. Serial bus interface is used for controlling the sensor.

- **Image signal processing (ISP)**

If the image data from the sensor is in RGB Bayer format, the ISP unit is used to convert it into YUV422 format. The ISP unit includes CFA-interpolation, noise reduction, sharpness enhancement, color correction, gamma correction, etc.

- **Image post processing (IPP)**

This module can add special effects to the image during capture and display, such as relief, sephia, color range, negative, etc. Also, when the resolution of the input image from the sensor or host CPU does not match display or capture requirement, IPP can be used to convert these images into appropriate size, i.e., screen display size or the capture resolution set by users.

- **LCD display controller (LCDC)**

This module generates the correct data for LCD panel to display. Alpha blending and overlay function between video and graphics from host CPU are supported. Rotation (90, 180 and 270 degree), mirror (horizontal and vertical) and gamma correction on video plane are all supported in this module.

- **LCD panel interface (LCDIF)**

VC0568 supports TFT and STN-type LCD panels up to CIF resolution (352x288). It communicates with the LCD panel through standard asynchronous 8/16 bit data bus with the color depth of 4/8/12/16/18 bit per pixel. Direct interface between host CPU and LCD panel is also supported.

- **JPEG codec (JPEG)**

VC0568 has an embedded JPEG codec for both image compression and decompression. The input of the encoder comes from either the sensor or the host CPU under different operation modes, and the output of the encoder is saved in on-chip SRAM (64KB) for host CPU to read. The input of the decoder is from the on-chip SRAM which can be accessed by the host CPU and the decompressed data is sent back either to host CPU or LCD display. The JPEG CODEC supports YUV422, YUV411, YUV420, YUV400 data format.

- **Memory Arbiter (MARB)**

The built-in 384Byte SRAM is used for storing the JPEG image data and graphics data. The size and position of graphics buffer and JPEG buffer can be configured dynamically. Some modules, such as JPEG, BIU, GE, STO, LCDC and IPP will address the on-chip SRAM. The memory arbiter will arrange the request from these modules and then response correctly.

- **Line buffer (LBUF)**

Two sets of 8-line buffer (1280x8x2 Bytes) are built in VC0568 for converting the data from line sequence to block sequence in capture mode and from block sequence to line sequence in display mode and decoder mode. The line buffer unit controls data read/write access to 8-Line Buffer. The two sets of 8-line buffer is configured as ping-pang architecture in capture and display modes. In decoder mode, one set is used as the FIFO to change block sequence to line sequence and the other is used for data storage after resizing.

- **Bus interface (BIU)**

The host CPU can read/write the control registers and on-chip SRAM of VC0568 via host interface unit. VC0568 supports most popular CPUs such as ARM, XScale, and other 60/80-series CPUs. The data bus is 8bit/16bit (synchronous or asynchronous). The internal control bus is AMBA.

- **Graphics Engine (GE)**

VC0568 comes with a powerful graphics engine along with on-chip graphics buffer for 2D graphics acceleration operations such as BitBLT, ROPs, and line drawing. The on-chip SRAM is used as the display memory. The input to the graphics engine either comes from the host CPU or the display memory. With the built-in graphics engine, the host CPU is freed from most of the display function.

- **System Controller (CPM, including clock, reset and power management)**

VC0568 has a variety of power management mechanisms to reduce power consumption. The internal clock can be stopped or modified by the clock control registers, and the internal PLL oscillation may be stopped under PLL bypass mode (note that the performance depends on the input clock frequency in PLL bypass mode). Also, the gated clock function stops the clock of non-operation block automatically. This unit also controls the action of the on-chip analog phase-locked-loop (PLL).

6. Main Building Block

6.1 Host Interface

6.1.1 Host interface Overview

The host interface has following features:

- Asynchronous CPU bus
- 16/8 bit data bus
- 16/8 bit access for registers and SRAM (may be byte enable)
- Internal AMBA interface
- Serve as external bus master function
- Configurable timing parameter

The host CPU can control the vc0568 chip via the host interface

6.1.2 Address Mapping

The following table is the address map of VC0568. The host CPU can access them all.

Address allocation	Space name
200000h~FFFFFFh	Reserved
100000h~1FFFFFh	On-chip 1T SRAM
002C00h~0FFFFFh	Reserved Note: 0XFFFE and 0XFFFF reserved for Host Interface.
002800h~002BFFh	LCDC unit control register
002400h~0027FFh	MARB unit control register
002000h~0023FFh	ISP unit control register
001C00h~001FFFh	IPP unit control register
001800h~001BFFh	BIU unit control register
001400h~0017FFh	CPM unit control register
001000h~0013FFh	GE unit control register
000C00h~000fFFh	LBUF unit control register
000800h~000BFFh	SIF unit control register
000400h~0007FFh	LCDIF unit control register
000000h~0003FFh	JPEG unit control register

IA[23:0]*	Area name
	Reserved
00_18b6H	MEM_8_FLG
00_18b4H	REG_8_HIGH_WORD
00_18b2H	REG_8_LOW_WORD
00_18b0H	REG_8_FLG
00_1890H	BYPASS_SEL
00_188CH	SEL_8_16
00_1888H	SEL_PORT
00_1886H	SEL_WRITE_READ
00_1880H	MUL_U2IA
00_1850H	MEM_FLG
00_184CH	Reserved
00_1848H	MEM_HIGH_WORD
00_1844H	MEM_LOW_WORD
00_1802H	CMD

There are also some registers which are used only in the separate bus. They can be accessed by setting both CPU_A8 and CPU_RS to '1' at the same time.

CPU_ADDR[7:0]*	Register name	Description
FFh 81h	Reserved	
80H	MEMDAT	Memory port at normal memory access mode
7Fh 11h	Reserved	
10H	REGUIAH	
0Fh 04h	Reserved	
03h	MEMLIA0	Memory address (m_address), MEMUIA = m_address[17:10]
02h	MEMLIA	Memory address (m_address), MEMLIA = m_address[9:2]
01h	MEMUIA	Memory address (m_address), MEMLIA0 = m_address[1:0];
00h	REGUIA	high 8 bit of address, that is address[15:8];

6.1.3 Description of Host CPU Interface

6.1.3.1 16bit Multiplex Interface

In the 16bit multiplex mode, the CPU_A8 pin is used as the address and data select pin. When CPU_A8 is set to '0', DATA[15:0] is the address of the memory or register to be accessed; when A8 is set to '1', DATA[15:0] is the data of the accessed cell. CPU_RS pin is used in directly through mode by mapping CPU_RS to LCD_RS when host CPU directly accesses the LCD panel.

In normal operation mode, the internal register and memory can be accessed according to the following table:

VC0568 side	CPU_CS1N	CPU_A8	CPU_RS	CPU_DATA	CPU_ADDR	CPU_WEN	CPU_RDN	Operation mode
CPU side	H	-	-	-	-	-	-	Inactive mode
	L	L	-	DATA[7:0]	DATA[15:8]	L	H	Index address Write mode
	L	H	-	DATA[7:0]	DATA[15:8]	L	H	data Write mode
	L	H	-	DATA[7:0]	DATA[15:8]	H	L	Data Read mode

In through mode, the following table is the mapping between the LCD panel and host CPU:

VC0568 side	CPU_CS1N	CPU_A8	CPU_RS	CPU_DATA	CPU_ADDR	CPU_WEN	CPU_RDN	Operation mode
CPU side	H	-	-	-	-	-	-	Inactive mode
	L	L	LCD_RS	LCD_DATA [7:0]	LCD_DATA [15:8]	L	H	Index address Write mode
	L	L	LCD_RS	LCD_DATA [7:0]	LCD_DATA [15:8]	L	H	data Write mode

	L	L	LCD_RS	LCD_DATA [7:0]	LCD_DATA [15:8]	H	L	Data Read mode

There are several methods for writing data into a register or memory cell:

1. When writing data to a register:
   1) Write index address register (A8=0);
   2) Write control data (A8=1).
2. Write writing data to the memory:
   1) Write index address of register (MEM_LOW_WORD, MEM_HIGH_WORD);
   2) Write port address (MEM_FLG);
   3) Write control data (A8 = 1);

   For the meaning of the above register, please see the register table.
3. Address increment automatic access (memory access)
   1) Set sel_port and sel_write (SEL_PORT, SEL_WRITE);
   2) Write index address of register (MEM_LOW_WORD, MEM_HIGH_WORD);
   3) Write port address (MEM_FLG);
   4) Write control data (A8 = 1);
   5) Write control data (A8 = 1) ….

Normal register read cycle:

1. When reading data from a register
   1) Write index address of register (A8 = 0);
   2) Control data is outputted (A8 = 1)
2. When reading data from memory
   1) Write index address of register (MEM_LOW_WORD, MEM_HIGH_WORD);
   2) Write port address (MEM_FLG);
   3) Read control data (A8 = 1);
3. Address increment automatic access (memory access)
   1. Set the sel_port and sel_read register (SEL_PORT, SEL_READ);
   2. Write index address of register (MEM_LOW_WORD, MEM_HIGH_WORD);
   3. Write port address (MEM_FLG);
   4. Write control data (A8 = 1);
   5. Write control data (A8 = 1) ….

6.1.3.2 8bit Multiplex Interface

In the 8-bit multiplex mode, the A8 pin is used as address and data select pin.

Operation mode for normal register

CSN	A8	RS	DATA	ADDR	WEN	RDN	Operation mode
H	-	-	-	-	-	-	Inactive mode
L	L	-	DATA[7:0]	-	L	H	Index address Write mode
L	H	-	DATA[7:0]	-	L	H	Data Write mode
L	H	-	DATA[7:0]	-	H	L	Data Read mode

Normal register/memory writing:

1. Data/memory writing:
    1) Write register high address index (REG_8_HIGH_WORD);
    2) Write address high 8 bits;
    3) Write register low address index (REG_8_LOW_WORD);
    4) Write address low 8 bits;
    5) Write register flag index (REG_8_FLG);
    6) Write control data (A8 = 1).
2. Write data to the same index address successively (memory access)
    1) Write index address of memory register according to above register write method (MEM_LOW_WORD, MEM_HIGH_WORD);
    2) Write port index address (MEM_8_FLG) ;
    3) Write control data (A8 = 1);
3. Address increment automatic access (memory access)
    1) Set the increment register according (INCREMENT);
    2) Write index address of register (MEM_LOW_WORD, MEM_HIGH_WORD);
    3) Write port address (MEM_8_FLG);
    4) Write control data (A8 = 1);
    5) Write control data (A8 = 1) ….

    Note: in the access mode, the address is increased automatically

Normal register read cycle:

1. Data/register read:
    1) Write register high address index (REG_8_HIGH_WORD);
    2) Write address high 8 bits;
    3) Write register low address index (REG_8_LOW_WORD);
    4) Write address low 8 bits;
    5) Write register flag index (REG_8_FLG);
    6) Read control data (A8 = 1);
2. Memory read:
    1) Write index address of register (MEM_LOW_WORD, MEM_HIGH_WORD);

2) Write port address (MEM_8_FLG);
3) Read control data (A8 = 1);
4) Read control data (A8 = 1)….

3. Address increment automatic access (memory access)
1) Set the increment register (INCREMENT);
2) Write index address of register (MEM_LOW_WORD, MEM_HIGH_WORD);
3) Write port address (MEM_8_FLG);
4) Read control data (A8 = 1);
5) Read control data (A8 = 1) ….

6.1.3.5 Separate Page Bus Interface

In the access from the HOST CPU, there are two operation modes (HOST register access, register access). One of those modes is selected by the condition of CPU_RS and CPU_A8.

CPU_RS=1 and CPU_A8=1	HOST register access
CPU_RS=0and CPU_A8=0	Register access
CPU_RS=0 and CPU_A8=1	Register access
CPU_RS=1 and CPU_A8=0	Register access

In register access, according to the set up to the HOST registers, HOST CPU accesses each internal modules and internal memory of the VC0568. Internal address map (IA) of VC0568 is shown as follows. Internal address bus of VC0568 is 24bit wide.

In memory access there are two options. One is the same as the register access shown above. The IA configuration is IA[23:16] equals to REGUIAH, and the IA[15:8] equals to the REGUIA, and the IA[7:0] equals to the ADDR. The other option is automatic address count mode. In this mode once the initial target address is set, HOST I/F automatically increments the target address and consecutively accesses internal memories. IA[23:0] does not need to be set at each access when this option is applied.

Operation mode

CSN	CPU_RS	CPU_A8	Addr7-0	DATA	WEN	RDN	Operation mode
H	-	-	-	-	-	-	Inactive mode
L	H	H	Addr7-0	DATA	L	H	host register write mode
L	H	H	Addr7-0	DATA	H	L	host register read mode
L	H	L	Addr7-0	DATA	L	H	Register write mode
L	L	H	Addr7-0	DATA	L	H	Register write mode
L	L	L	Addr7-0	DATA	L	H	Register write mode
L	H	L	Addr7-0	DATA	H	L	Register read mode
L	L	H	Addr7-0	DATA	H	L	Register read mode
L	L	L	Addr7-0	DATA	H	L	Register read mode

Write cycle:

1. Register /normal memory write access
    1) Write host register;
    2) Write register/normal memory.
2. Write data to the same index address successively (memory access)
    1) Write host register (3 registers);
    2) Write the port write select register;
    3) Write data to the special register (host register, MEMDAT);
    4) Write data to the special register (host register, MEMDAT)….

    Note: in the access mode, the address is incremented automatically

Read cycle:

1. Register/normal memory read access
    1) Write host register;
    2) Read register/normal memory
2. Write data to the memory port address (memory access)
    1) Write host register (3 registers);
    2) Write the port select register;
    3) Read data from the special register (host register, MEMPORT)
    4) Read data from the special register (host register, MEMPORT)….

6.1.4 Chip Configuration

The CONF pin is used to configure the chip and should be connected to the VSS or VDD before chip reset.

Configuration input	Description
CONF [2:0]	Host bus interface type “000”: type 0; “001”: type 1; “010”: type 2; “011”: type 3; “100”: type 4; “101”: type 5; “110”: type 6; “111”: type 7;

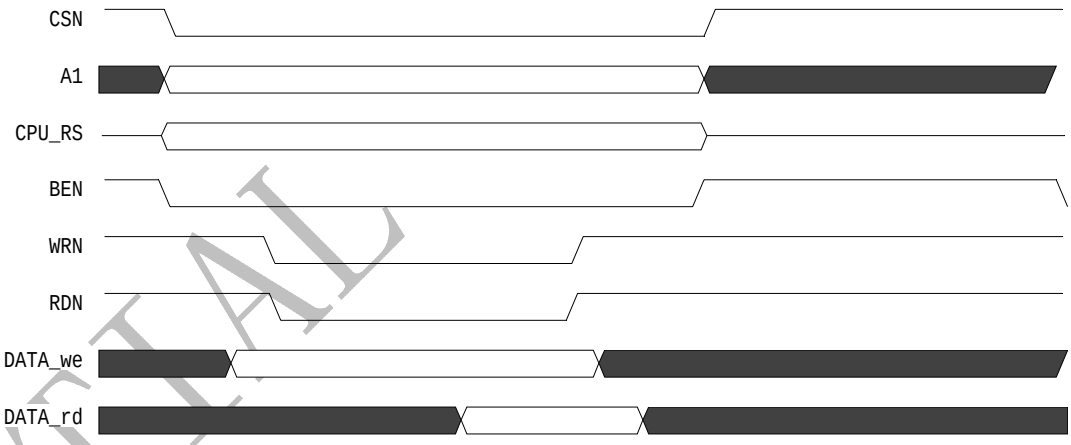
For different bus type, the CPU interface has different configuration, which can be seen in the following table.

Pin name	Type0	Type1	Type2	Type3	Type4	Type5	Type6	Type7
ADDR [7:0]	DATA [15:8]	DATA [15:8]	DATA [15:8]	DATA [15:8]	ADDR [7:0]	ADDR [7:0]	ADDR [7:0]	ADDR [7:0]
CPU_A8	A1	A1	A1	A1	CPU_A8	CPU_A8	CPU_A8	CPU_A8
CPU_RS	Low	Low	Low	Low	CPU_RS	CPU_RS	CPU_RS	CPU_RS
DATA	DATA [7:0]	DATA [7:0]	DATA [7:0]	DATA [7:0]	DATA [7:0]	DATA [7:0]	DATA [7:0]	DATA [7:0]
CSN	CS#	CS#	CS#	CS#	CS#	CS#	CS#	CS#
RDN	RD#	RD#	RDL#	high	RD#	RD#	RDL#	high
WEN	WR#	high	WRL#	R/W#	WR#	high	WRL#	R/W#
BEN[1]	UBE#	WRU#	RDU#	UDS#	UBE#	WRU#	RDU#	UDS#
BEN[0]	LBE#	WRL#	WRU#	LDS#	LBE#	WRL#	WRU#	LDS#

A1: address/data select pin
RDL#: read data low byte enable
RDU#: read data high byte enable
WRL#: write data low byte enable
WRU#: write data high byte enable
R/W#: CPU write/read signal(68 type)
UBE#: upper byte enable
LBE#: low byte enable

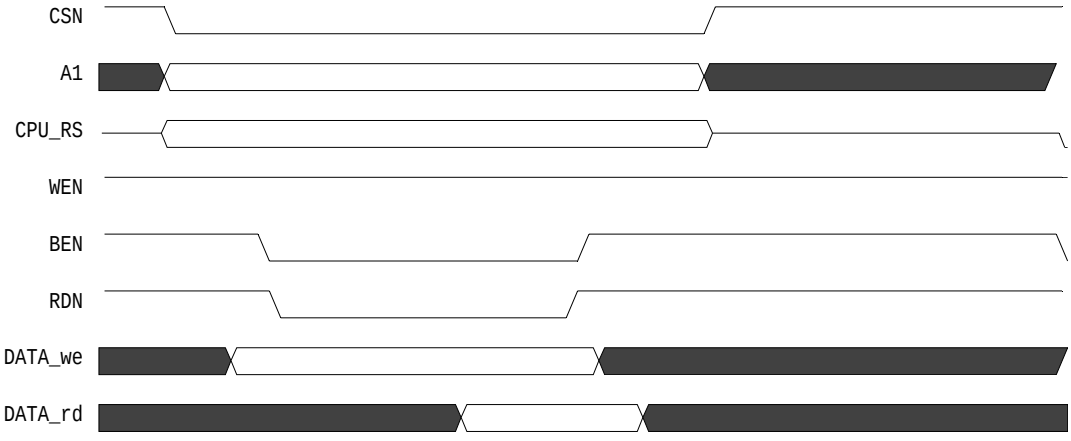
LDS#: low data select pin
UDS#: upper data select pin

6.1.4.1 Multiplex 80 type0

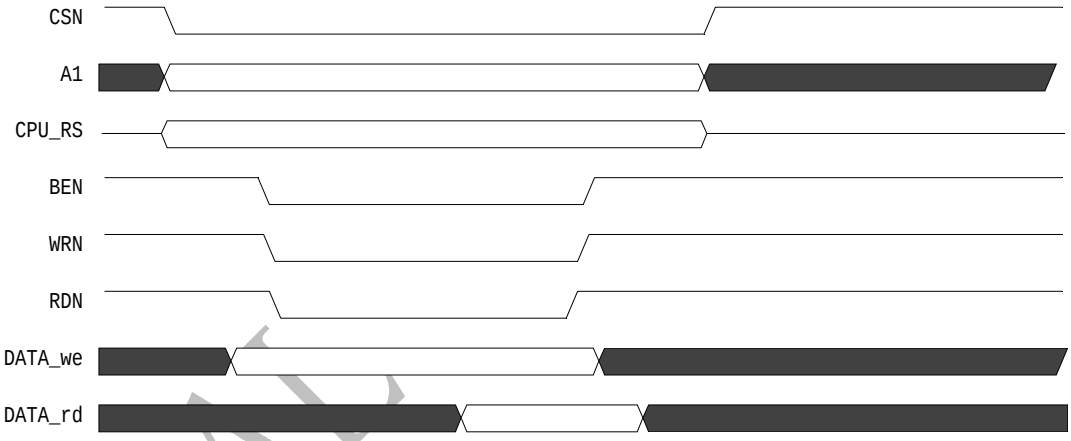


Note: WRN and RDN share using the BEN and don't care the CPU_RS value.

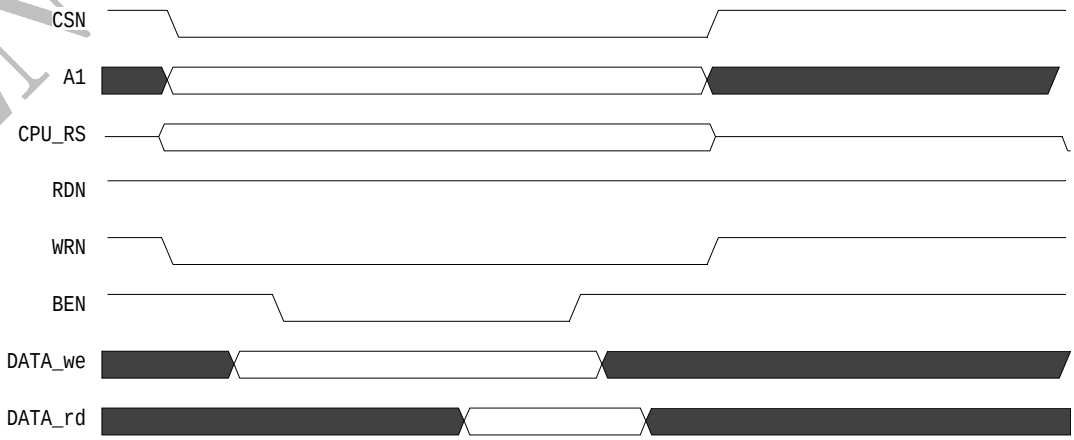
6.1.4.2 Multiplex 80 type1



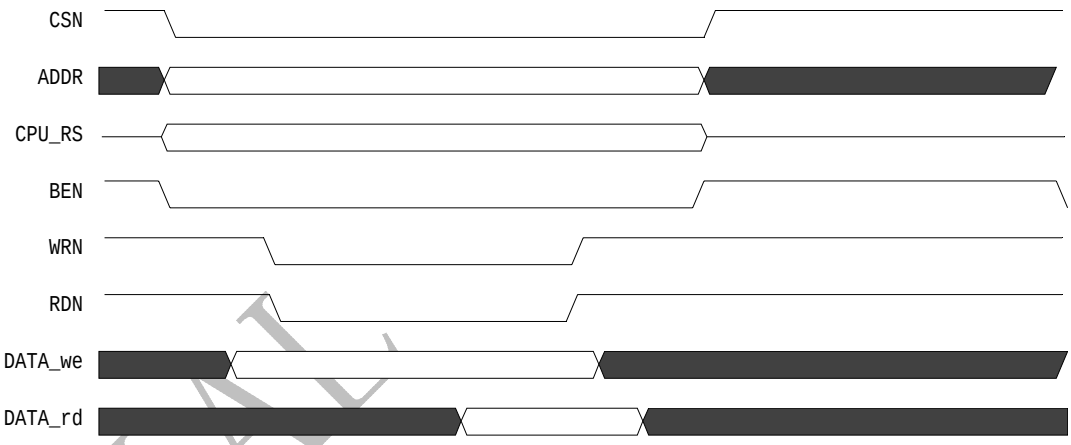
6.1.4.3 Multiplex 80 type2



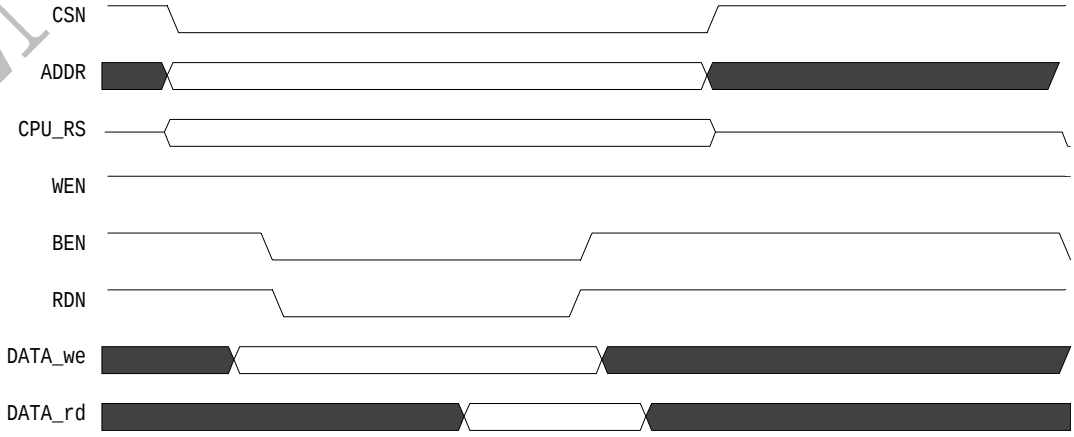
6.1.4.4 Multiplex 68 type3



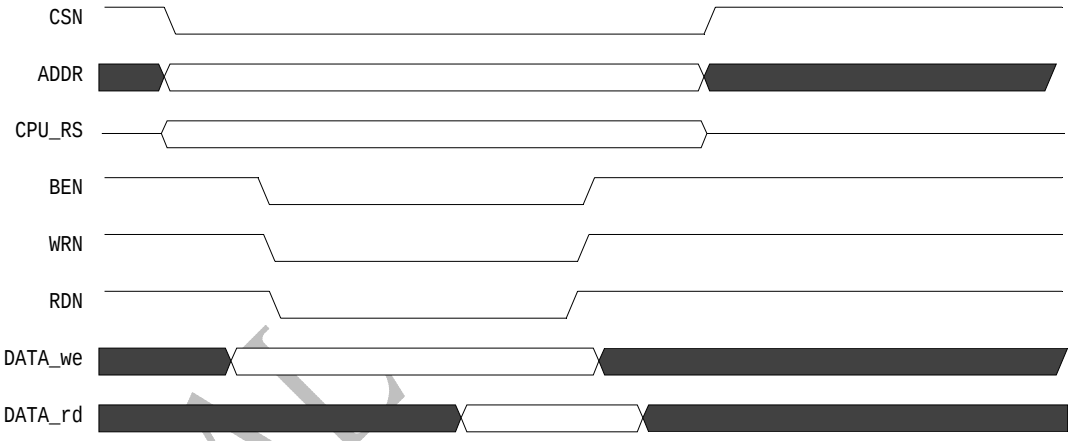
6.1.4.5 Separate page 80 type4



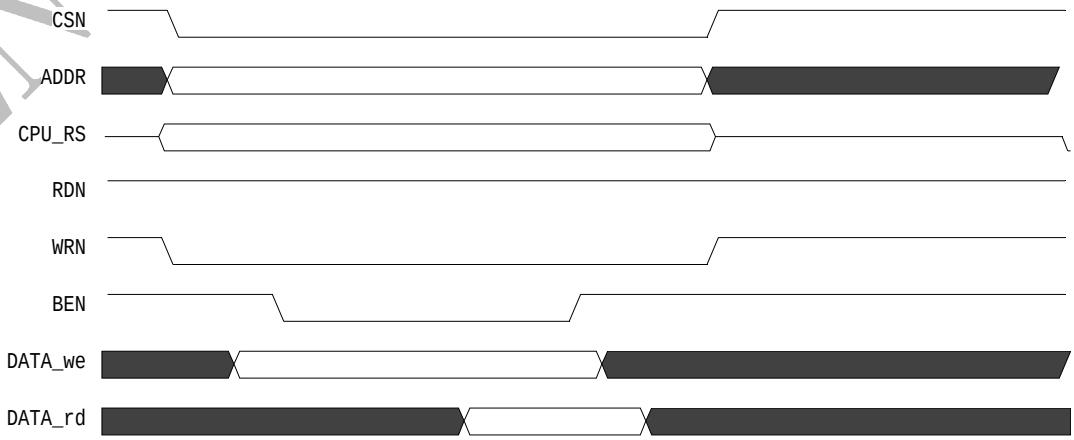
6.1.4.6 Separate page 80 type5



6.1.4.7 Separate page 80 type6



6.1.4.8 Separate page 68 type7



6.2 Sensor Interface

The sensor interface (SIF) acts as the bridge between the external sensor and the image processing block inside the VC0568 chip. On the sensor side, the SIF receives 10/8-bit image data and synchronous signals from the sensor and outputs control signals to it. If the sensor works in a slave mode, i.e., can't generate synchronous signals by itself, the SIF will generate them for the sensor. The synchronous signals for different sensors may vary. The SIF uniforms the synchronous signals and forwards them together with image data to image processing block. If the image is in Bayer pattern, the data will be sent to ISP module; if the image is in YUV422 format, the data will be sent to Special Effect module.

The basic features of SIF:

- Synchronous signals (to sensor or to ISP) programmable;
- Support 2 kinds of sensor control buses: 2-wired serial bus and 3-wired serial bus;
- Support batch writing sensor registers;
- Internal image window adjustable;

6.3 Image Signal Processing

6.3.1 ISP overivew

The image data from the sensor can be Bayer pattern or YUV422 data. For either Bayer pattern and YUV image data, the image data should be processed by the Image Signal Processing (ISP) unit. Using the ISP unit, many kinds of sensor can be supported by VC0568. The ISP unit supports the following features:

- Test bar
- Auto dead pixel detection and concealment (DPD/DPC)
- Noise cleaning
- Digital gain
- Lens fall off compensation
- 5x5 CFA Interpolation
- Color correction
- Gamma correction
- Edge detection and enhancement
- Color space converter (RGB2YCbCr)
- Auto white balance
- Auto exposure
- Auto flicker detection and removal
- 10/8 bit data processing
- Auto focus
- Histogram statistic and bright compensation based on histogram

6.3.2 Image Processing Function

Noise Cleaning and DPC

Dead pixel concealment (DPC) is used for finding the pixel that does not respond correctly to light input and hiding it from user's eye. Real time DPD/DPC algorithm is adopted for this purpose. Aside from DPD/DPC, noise may exist in the image signals. VC0568 supports the auto noise cleaning as well. Note that for YUV pattern data from SIF module, only DPD/DPC feature is available.

Digital Gain

There are two types of digital gain: global digital gain and RGB digital gain. The digital gain control is done on the input RGB Bayer pattern stream. All the pixel data (R, G and B) are multiplied by a digital global gain. The higher the global gain is, the brighter the image is. The global gain is used to adjust image brightness in auto exposure function.

Different from digital global gain, digital RGB gains affect RGB pixel separately. Different color channel uses different gain value, so there is three gain values for R, G and B pixel, respectively. Digital RGB gains are used when hardware auto white balance (AWB) process function is disabled.

Lens Falloff

Lenses produce non-uniform exposure at the focal plane when imaging a uniformly illuminated surface. When a lens is modeled as a thin lens, the ratio of the intensity of the light of the image at any point is the function of the angle between the optical axes, the lens, and the point in the image plane. This causes the corners of an image to be darker than desired, especially for systems with short focal length lens. In addition, flashlight will produce similar falloff effect if the subject is centrally located within the image frame.

The lens falloff effect can be corrected by applying certain digital gain to each pixel. The digital gain for a specific pixel can be determined from the focal length of the lens as well as the current pixel location within the image frame (distance to image center).

Interpolation

If the color data from the sensor is Bayer pattern, each pixel only has one color value that may be one of R, G or B. To display correctly, the Bayer pattern data must be converted to RGB space, where one pixel has R, G and B colors. This processing is called interpolation. VC0568 uses a fine interpolation algorithm to guarantee good image quality.

Edge Enhancement

The edge enhancement function will improve image sharpness.

Color Correction

Color correction is done using a color matrix and it compensates color deviation due to color filtering and sensing circuits.

Gamma Correction

Piecewise linear gamma approximation method is implemented in VC0568. There are three gamma curves for R, G and B channel respectively. Each gamma curve has sixteen pieces of linear segment that are user-programmable.

Color Space conversion

This converts image data from RGB to YUV color space for JPEG compression.

Auto exposure

The luminance of the image changes when the environment varies. In order to keep the luminance in the given range, we must adjust exposure time of the sensor. When the image is too bright, exposure time will be decreased. When the image is too dark, exposure time

will be increased. In outdoor environment, just changing exposure time may satisfy the requirement because the exposure time can be set to any value and the luminance of the image is continuous.

But in indoor environment, just changing exposure time may not satisfy the requirement. The illumination system flickers several times per second due to AC power fluctuation. The AC power frequency varies, too. Some region uses 60Hz, and some uses 50Hz. The human eye cannot detect the flickering. However, the sensor can, and the flickering appears as banding noise.

If the exposure time is very small, changing exposure time one unit can make dramatic change in the brightness of image. In order to make the brightness of the image relatively continuous, we can change the global digital gain of the image first. If the image is too bright or too dark, we can first adjust global gain and then adjust exposure. Thus the brightness of the image is relatively continuous.

For a given frame, VC0568 will calculate the average of the luminance and send it to the host CPU. The host CPU will adjust the exposure time of the sensor and the digital global gain accordingly.

Auto White Balance

The human eye and brain are capable of “white balancing.” If a person takes a white card outside, it looks white. If he takes it inside under fluorescent lights, it also looks white. Getting a machine to do the same thing is harder. When the white card moves from one light source to another, an image sensor “sees” different colors under different conditions. Consequently, when a digital camera is moved from outdoors (sunlight) to indoor fluorescent or incandescent light conditions, the color in the image shifts. If the white card looks good indoors, for example, it might look bluish outside. If it looks good under fluorescent light, it might look yellowish under an incandescent lamp.

VC0568 will calculate the average R, G and B values of every frame. The gain value of each color channels is then adjusted accordingly.

Auto Flicker Detection and Removal

This function will constantly monitor the environment and determine the right light source (outdoor, 60Hz indoor or 50Hz indoor). The host CPU will then adjust sensor exposure time to ensure that flickering is eliminated.

6.4 Image Post Processing

6.4.1 IPP Overview

The image post-processing unit is used to add some special effects on the image and resize the image for display and capture. The input of the image post-processing unit comes from the sensor interface unit (SIF) or image-processing unit (ISP), depends on the sensor type. The resized image is sent to the LCD I/F unit in viewfinder mode and display mode and the 8-Line buffer unit in capture mode. When capture video clip, the resized image should be decompressed and at the same time displayed on the LCD screen, so we need two sizers: display sizer for resizing image on LCD screen and capture sizer for capturing image. To avoid potential aliasing artifacts, an image pre-prefilter is employed to band-limit the original image before resizing. The IPP module supports the following features:

- 16x digital zoom in viewfinder mode and capture mode
- Smooth switch from viewfinder mode to capture mode
- Support the following special effect
  - Monochrome
  - Sepia
  - Special color
  - Negative
  - Relief
  - Sketch
- Flexible 7-tap prefilter for image sizer
- Configurable coefficient for prefilter
- Three sizer for display and capture function
  - Capture sizer
  - Display sizer
  - Thumbnail sizer

6.4.2 Function Description

6.4.2.1 Digital Zoom

Digital zoom means that the user can change the view which displayed on the LCD panel via a pushbutton. The size of the displayed image on LCD panel is unchanged, but the view can be changed by zooming in/out so that the image comprises more or less objects. To do digital zoom, truncating a window from the source image for resizing to an image with fixed size should be supported. Also the window size should be configurable.

The input image is come from the image sensor. When do digital zoom, the user first configure the sizes of the capture image and display image. Then he can push the button of zoom in/out to select the right view for capture. During this phase, the view displayed on the LCD panel changes as the source window changes. After the user has chosen the right view, he can push the 'capture button' and then a frame of still image or video will be captured. The view of the captured image or video should be the same as that of the image which displayed on the panel, i.e., the two sizer – capture sizer and display have the same source image.

According to the above description, for digital zoom we should have the following parameter:

1) The width and height of the input image: IMAGE_WIDTH, IMAGE_HEIGHT;
2) The width and height of the image window: WINDOW_WIDTH, WINDOW_HEIGHT;
3) The coordinate of the start pixel of the image window: WINDOW_START_X, WINDOW_START_Y;
4) The target width and height of the displayed image: DISPLAY_TARGET_WIDTH, DISPLAY_TARGET_HEIGHT;
5) The horizontal and vertical resize ratio of the display sizer: DISPLAY_HORIZONTAL_RATIO, DISPLAY_VERTICAL_RATIO;
6) The target width and height of the captured image: CAPTURE_TARGET_WIDTH, CAPTURE_TARGET_HEIGHT;
7) The horizontal and vertical resize ratio of the capture sizer: CAPTURE_HORIZONTAL_RATIO, CAPTURE_VERTICAL_RATIO

Let's take an example to show how to use parameters. If the width and height of the input image is VGA (640x480), the image displayed on the LCD screen is 128x96, and the captured image is 160x120, then we can set the initial value to the above registers:

IMAGE_WIDTH=640, IMAGE_HEIGHT=480
WINDOW_WIDTH=640, WINDOW_HEIGHT=480
WINDOW_START_X=0, WINDOW_START_Y=0
DISPLAY_TARGET_WIDTH=128, DISPLAY_TARGET_HEIGHT=96
CAPTURE _TARGET_WIDTH=160, CAPTURE _TARGET_HEIGHT=120

Generally, the display window and the capture window should be just the same because what you see, what you get (WYSWYG). So we have only one window for capture and display, and only one set of parameters to describe the size and start pixel of the window. Note there are vertical sizer ratio and horizontal sizer ratio for one sizer, but it is recommended that the horizontal sizer ratio should be equal to the vertical sizer ratio; otherwise the image would be distorted. Because the initial image window is 640x480, i.e., the whole input image, and the display size is 128x96, we can simplify 640/128 to 5/1, and the floating point is 000101_0000000000 (1400h), i.e., for the pixels in one row, we should take one pixel from every five for the resizing ration of 5/1. Then the following parameter is set to:

DISPLAY_HORIZONTAL_RATIO=1400h (5), DISPLAY_VERTICAL_RATIO=1400h (5)

Similarly, the following parameter is set to:

CAPTURE_HORIZONTAL_RATIO=1000h (4), CAPTURE_VERTICAL_RATIO=1000h (4)

When we want to zoom in the image displayed on the LCD panel, the image window for resizing should be less than the previous one but at the same time the display image keep unchanged. Because the same display region has fewer objects in time, we can notice the result of the zoom in. For example, we can take the digital zoom step as 32x24, i.e., the image window is (640-32)x(480-24)=608x456, the following register should be set again:

WINDOW_WIDTH=608, WINDOW_HEIGHT=456

WINDOW_START_X=16, WINDOW_START_Y=12

DISPLAY_HORIZONTAL_RATIO=1300h (4.75), DISPLAY_VERTICAL_RATIO=1300h (4.75)

CAPTURE_HORIZONTAL_RATIO=1000h(3.8), CAPTURE _VERTICAL_RATIO=1000h (3.8)

When such process of digital zoom is going, these parameters will change according to the above rules. It is similar for the process of zoom out.

For the above digital zoom process, there are (640-128)/32=16 steps and it is a 640/128=5x digital zoom. During each step of zoom, the above 12 registers should be configured. We use the writing operation of high byte of DISPLAY_VERTICAL_RATIO for updating all the registers related to digital zoom. So this byte should be written later than other registers, otherwise the zoom operation will not work right because the registers have not taken into effect.

6.4.2.2 Special Effect

Special effects can be added on the image for display and capture in viewfinder mode, capture mode and display mode. VC0568 supports the following special effects:

1) Monochrome pattern: this kind of effect is done on the YUV422 space. When set to the monochrome pattern, Y keep unchanged, U and V is set to zero;

2) Sephia: this kind of effect is done on the YUV space. When set to sephia effect, Y keep unchanged and U and V are set to fixed values. Sephia and monochrome can be implemented together. If we want to use this effect, we should set the special effect enable/disable bit of SPECIAL_CONTROL to ‘1’ and the special mode bits to 3’b000. The fixed U and V values are defined by the registers of U_OFFSET and V_OFFSET;

3) Special Color: this effect is also done on the YUV space. When set to this effect, Y keep unchanged; U and V is set to be in a certain region and the values outside the region are set to zero. For this kind of special effect, the special effect enable/disable bit is set to ‘1’ and special effect mode bits to 3’b001. The region of U and V and defined by U_UP_THRESHOLD, U_DOWN_THRESHOLD, V_UP_THRESHOLD and V_DOWN_THRESHOLD;

4) Negative: this effect is done on the RGB space. When set to this effect, R, G and B is defined by the following formula:

   New R = (256 – old R)
   New G = (256 – old G)
   New B = (256 – old B)

   When set special effect enable bit to ‘1’ and special effect mode bits to 3’b010, the negative effect will work.

5) Relief: this effect is also done on the RGB space. When set to this effect, R, G and B is defined by the following formula:

   New R = old R[x][y] – old R[x+2][y+2] + 128
   New G = old G[x][y] – old G[x+2][y+2] + 128
   New B = old B[x][y] – old B[x+2][y+2] + 128

   When set special effect enable bit to ‘1’ and special effect mode bits to 3’b100, the negative effect will work.

6) Sketch: this effect is done on YUV422 space. When set this effect, the extracted edge value (EDGE) from ISP module is set to Y channel. U and V are set to fixed color.

   NEW Y = EDGE + Y_OFFSET
   NEW U = U_OFFSET
   NEW V = V_OFFSET

For the special effect, please see figure 6-1:

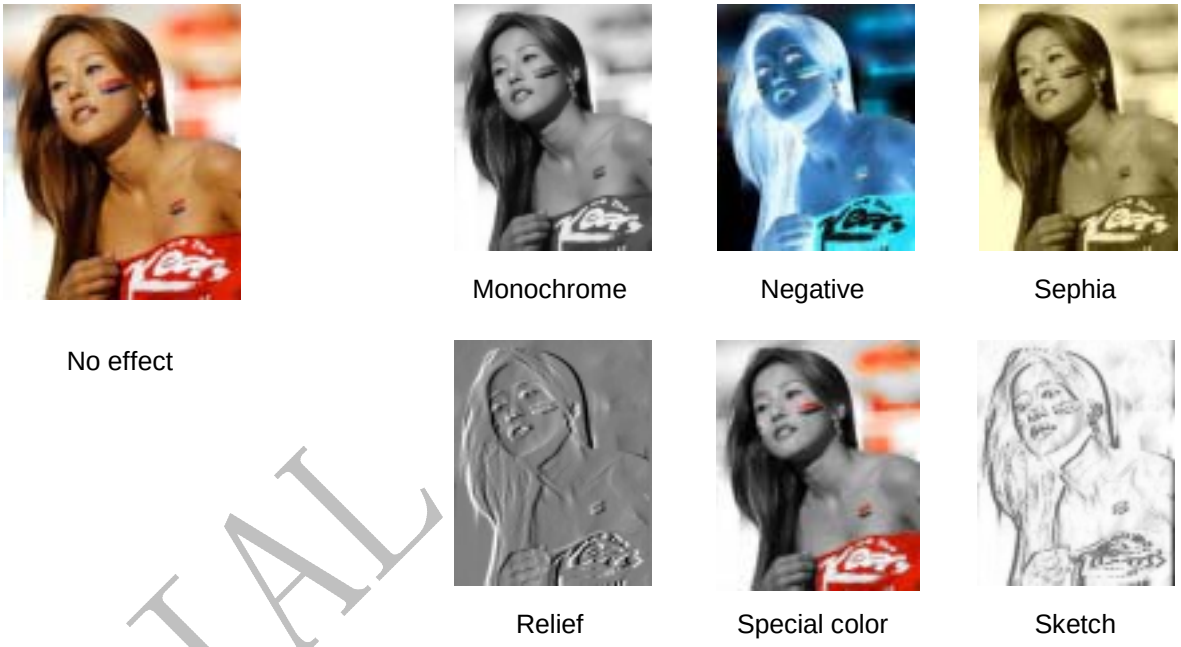


Figure 6-1: Special effects

6.4.2.3 Prefilter

The image prefilter in VC0568L we used in image downsizing is shown in Table 1, along with the associated range of downsizing factor for which a given filter is applied. All these filters are symmetric, odd-tap, and one-dimensional.

Down-sizing Factor (F)	Prefilter Kernel Coefficients
0.85 < F <= 1.0	No Prefiltering
0.75 < F <=0.85	0.25 0.5 0.25
0.5 < F <= 0.75	-0.1016 0.2578 0.6875 0.2578 -0.1016
0.25 < F <= 0.5	0.0625 0.25 0.375 0.25 0.0625
0.15 < F <= 0.25	0.0516 0.1221 0.2047 0.2432 0.2047 0.1221 0.0516
F <= 0.15	Either a prefilter with long taps or multiple-stage prefiltering

Table 1: Prefilters for different down-sizing factors

The prefilter should be done in horizontal direction and vertical direction. So 6 line buffers is needed for Y and another 6 line buffers for Cb and Cr because the tap of the prefilter is up to 7. When the image data coming, we should first do the horizontal filter and then save it in the line buffers. When the seventh line data comes, we will read out the previous 6 lines from the line buffer and do the vertical filter.

6.4.2.4 Bilinear Interpolation

In VC0568, bilinear interpolation algorithm is employed to do image resizing. Bilinear interpolation algorithm is commonly used in image resizing. It uses four nearest neighboring

pixels to interpolate one pixel (see Figure 6-2).

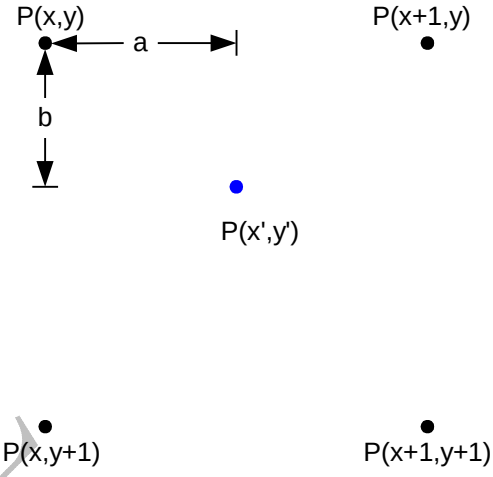


Figure 6-2: Bilinear interpolation algorithm

In Figure 6-2, the position (x,y), (x,y+1), (x+1,y), (x+1,y+1) is an original sample point in the original image. The position (x',y') represents a new sample location in the resized image that lies somewhere between the four surrounding pixel locations. The weight that a certain pixel has on the new sample point is directly related to the distance away from that point. The closer the new sample location is to one of the neighboring pixels, the more weight that pixel's gray level value should have. The values of a and b represent the vertical and horizontal distances respectively from the top left corner original sample pixel.

Hence, the bilinear formula is:

$$P(x^{'}, y^{'}) = (1-a)(1-b)P(x, y) + (1-a)bP(x, y+1) + a(1-b)P(x+1, y) + abP(x+1, y+1)\ \{0 \le a, b \le 1\}$$

The formula can be changed to the following

$$P(x^{'}, y^{'}) = (1-a)[(1-b)P(x, y) + bP(x, y+1)] + a[(1-b)P(x+1, y) + bP(x+1, y+1)]$$

From this formula, we can see that the bilinear interpolation algorithm can be decomposed to horizontal bilinear and vertical bilinear. In real implementation, we will first to the horizontal sizer and saved the image in the line buffers. For scaling down, 1 line buffer is enough. But for scaling up, there may be two or more new lines between two old lines, so these two old lines must be saved in line buffers. At the same time, the third line is coming, so we totally need three line buffers to do the sizer. Consider that Y and Cb/Cr, totally 6 line buffers are needed to do the vertical sizer.

6.5 Line Buffer

6.5.1 Overview

The JPEG compression and decompression deal with the image data block by block. But the other image pipe deals with the image data line by line. The line buffer unit converts the image data between line mode and block mode in capture mode, display mode and decode mode.

6.5.2 Role of 8-Line Buffer Unit in Each Mode

8-Line Buffer is used in capture mode, display mode and JPEG decode mode. The role of 8-Line Buffer Unit at each mode is summarized below.

Capture mode

In Capture mode, the image from the IPP unit is converted to block scan from line scan in LBUF module and YUV422 data of 8x8 block is sent to JPEG module.

The role of 8- Line Buffer Unit at the capture mode is as following.

1 Writing of image data input from IPP module or LCDC module.
2 Reading of data passed to JPEG module.

Display mode

In Display mode, the decoded image data (in YUV formats) in the JPEG unit can be stored back to 8-line buffers in LBUF module.

The role of 8- Line Buffer Unit at the display mode is as following.

1 Writing of data output from JPEG Decoder.
2 Reading of data passed to IPP module.

JPEG decode mode

In JPEG decode mode, the decoded YUV data is saved in the line buffer module. When 8(16 when yuv420) lines data are decoded, VC0568 will send out an interrupt signal to the Host and the Host will read the YUV data from the marb by biu. When data of 8(16 when yuv420) lines are read out, the Host will restart the JPEG to begin decoding the next 8 lines.

The role of LBUF Unit at the JPEG decode mode is as following.

1 Writing data output from JPEG to MARB.
2 Writing data output from IPP Unit to MARB.
3 Reading data from MARB to IPP.
4 Writing data output from LCDC unit.

There are 3 data paths pass 8-Line Buffer in JPEG decode mode, Step 1, Step 1&2&3, Step 1&2&4. decoder mode[1:0] in LBUF mode register choose the active pass among the three passes. Step 2&3, Step 4&3 are used in saving IPP data to MARB. The image data must be in YUV444 or YUV422 or YUV420 or YUV400 or YUV411 mode. One buffer is used for decoded data from JPEG Unit, the other buffer is used for storing data from IPP Unit. JPEG decoder is restarted when MARB unit read out data in line-buffer.

Summary

The mode is distinguished according to the Lbuf mode register in this module. 8-Line Buffer Unit controls writing/reading the necessary above mentioned after confirming a present mode according to the Lbuf mode register at each mode.

The data that comes from other modules to 8-Line Buffer and the data that sends to other modules from 8-Line Buffer are follows:
Data to 8-Line Buffer when write: IPP unit, LCDC unit, JPEG unit(JPEG Decoder).
Data from 8-Line Buffer when read: JPEG unit(JPEG Encoder), IPP unit, MARB unit.

When writing, 8-Line Buffer selects the data appropriately and writes to 8-Line Buffer.
When reading, the data are shared by all modules, and only the module that needs the data takes it.

6.6 Memory Arbiter

6.6.1 Memory Arbiter Overview

In VC0568, there are up-to three blocks of 1Mbit (32Kx32) embedded SRAM. These SRAM are used as graphic buffer, video buffer for LCD, JPEG buffer, thumbnail image buffer and JPEG 8-line buffer. Total three blocks of 1T-SRAM have continuous address from 0x000000 to 0x05FFFF. Address range from 0x060000 to 0x7FFFFF is not used. During each working mode, three blocks of SRAM can be configured to shutdown mode separately according to the requirement.

6.6.2 Memory Arbiter in Each Mode

6.6.2.1 Viewfinder Mode

In viewfinder mode, the following buffers is in active:

1) Video buffer. The video data is saved in the video buffer by LCDC module and then read out for refreshing LCD panel continuously. The video buffer can be used as frame buffer or line buffer;
2) Graphics buffer. The graphics data in the graphics buffer will update the LCD panel at the same time of video refreshing. The graphics data can be updated by host CPU and graphics engine. Layer B can be updated by host CPU and graphic engine. Host will write/read memory to update Layer B. Graphic Engine can also read/write memory to update Layer B.

6.6.2.2 Capture Still Image without Frame

When capture still image without frame, the following buffers are in active:

1) JPEG buffer. The compressed JPEG image data is stored in the JPEG buffer. The host interface will access memory to read captured image from SRAM through DMA mode. JPEG module will write compressed image data to memory;
2) Thumbnail buffer. IPP module will write uncompressed thumbnail image data to memory.
3) In this mode, there is no video or image to be shown in LCD. So the video buffer and graphics buffer are inactive;
4) Line buffer. The line buffer is a temporary buffer for storing YUV data of 16 lines to convert the image data from line mode to block mode for JPEG compression.

6.6.2.3 Capture Still Image with Frame

When capture still image with frame, the following buffers are in active:

1) JPEG buffer. The compressed JPEG image data is stored in the JPEG buffer. The host interface will access memory to read captured image from SRAM through DMA mode. JPEG module will write compressed image data to memory;
2) Video buffer. In this mode, the YUV data before compression is come from LCDC module, so the video buffer will store the YUV data;
3) Graphics buffer. The frame data is stored in the graphics buffer, which is written by the host CPU;
4) Thumbnail buffer. There's no thumbnail when capture still image with frame, so the thumbnail buffer doesn't work;
5) Line buffer. The line buffer is a temporary buffer for storing YUV data of 16 lines to convert the image data from line mode to block mode for JPEG compression.

6.6.2.4 Capture Video

When capture video, the following buffers are in active:

1) JPEG buffer. The compressed JPEG image data is stored in the JPEG buffer. The host interface will access memory to read captured image from SRAM through DMA mode. JPEG module will write compressed image data to memory. After a number of frames are saved to JPEG buffer, a save done interrupt is generated. The number of frame is set in register. The total frame size must be less than half JPEG buffer. The frame size is added before each JPEG frame automatically;
2) Video buffer. In this mode, the YUV data before compression is come from LCDC module, so the video buffer will store the YUV data;
3) Graphics buffer. The frame data is stored in the graphics buffer, which is written by the host CPU;
4) Thumbnail buffer. There's no thumbnail when capture still image with frame, so the thumbnail buffer doesn't work;
5) Line buffer. The line buffer is a temporary buffer for storing YUV data of 16 lines to convert the image data from line mode to block mode for JPEG compression.

6.6.2.5 Display Mode

In display mode, the following buffers are in active:

1) JPEG buffer. The JPEG buffer can be configured as a FIFO or a buffer. When JPEG buffer is a FIFO, JPEG data is written by host. When JPEG buffer is a pure buffer, JPEG data is already in buffer;
2) Line buffer. The line buffer is a temporary buffer for storing YUV data of 16 lines to convert the image data from block mode to line mode after JPEG decompression.

6.6.2.5 Decoder Mode

In this mode, memory usage is same as display mode.

6.7 LCD Display Control

6.7.1 LCDC Overview

The following features are supported in the LCDC module in VC0568:

- Programmable image dithering
- Support Picture-in-Picture function
- Separate controls for layer A and layer B
- Hardware cursor
- Alpha blending
- Overlay
- 90/180/270 degree image rotation
- Vertical and horizontal mirror
- Configurable background color
- Configurable data rate

6.7.2 LCDC Datapath

LCDC module is the display controller of LCD panel. It combines the image, graphics and background together and then sends to the LCD panel.

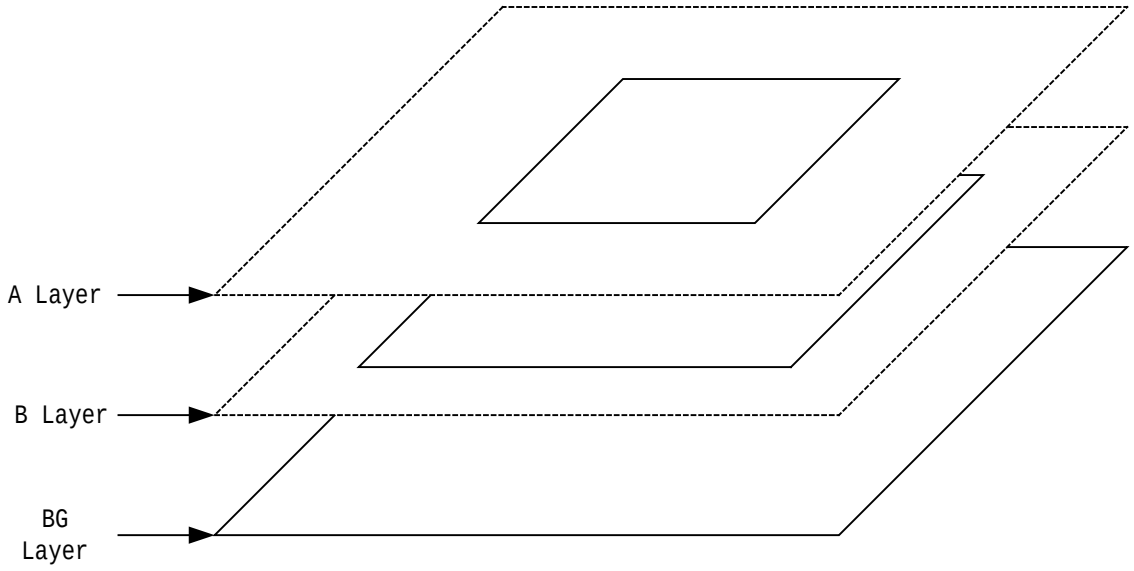


Figure 6-3 LCDC display layer

Figure 6-3 is the display layer for LCDC module. There are three layers to display on the LCD panel: layer A, layer B and background layer. Layer A is image data from the camera module, which is output from IPP module. Layer B is the graphics from the host CPU, which is stored in the on-chip graphics buffer and can be accessed via MARB module. The

background color is monochrome and it is a register which can be set by the host CPU.

The display region for each layer can be set by the host CPU. The BG layer always has the biggest region and layer A and layer B are parts of it. Layer BG has the lowest priority to display and it won't be displayed in the region of layer A or layer B. There are two kinds of region relationship for layer A and B: 1) layer A is one part of layer B; 2) layer A and layer B have no common part. For case a) in figure 6-4, the host CPU can set the priority layer for layer A and layer B for LCD panel to display. If one layer is set to the priority layer, this layer will be displayed on the LCD panel and the pixels of another layer in this region won't be displayed. For example, if layer A is set to priority layer in case a), the region of layer A will display the pixel data of layer A; but if layer B is set to priority layer, the region of layer A will display the pixel data of layer B. Transparent color can be set for the priority layer. If the pixel data of the priority layer is equal to the transparent color, then this pixel will display the combination of layer A and layer B, and this is called overlay. For detailed information, please check the chapter for overlay.

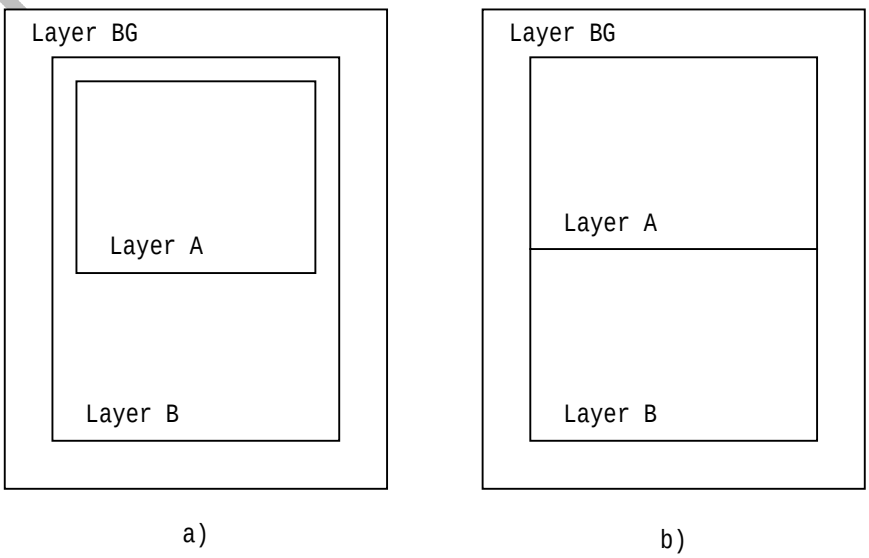


Figure 6-4 Layer relationship: a) layer A is one part of layer B;
b) layer A and layer B have no common part

6.7.2.1 Rotation

The image data of layer A can do some geometry transformations, such as rotation (90, 180 and 270 degree) and mirror (vertical and horizontal mirror), see figure 9-4. But when writing to the LCD panel, header should be added for the continuous display data to describe the address for writing LCD. For rotation, two bordering pixels in the same line from IPP module will be displayed in two lines of the LCD panel, so each pixel should be added a header before writing to the LCD panel. The bandwidth of the data is limited to this issue and it may lead to that there's no enough time to display all the video data on the LCD panel.

To do these transformations with no extra data bandwidth, the coming YUV422 data should be first saved in the rotation buffers. These two rotation buffers, each of which can hold 8 lines of YUV422 data with maximum 510 pixels per line, work in a ping-pong style. When image data from the IPP module is coming, they are stored in the line buffers. If eight line buffers are full, 8 pixels in the same row in the 8-line buffer will be sent to the LCD panel with only one header. Thus the bandwidth can be enlarged.

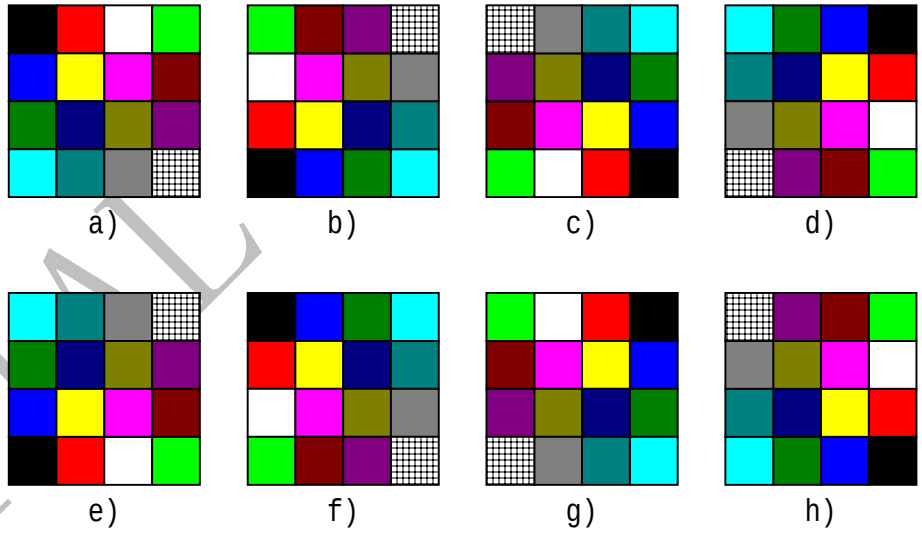


Figure 6-5 Rotation and mirror

a) Normal; b) Rotate 90^{o}; c) Rotate 180^{o}; d) Rotate 270^{o};
e) Mirror; f) Rotate 90o+Mirror; g) Rotate 180o+Mirror; h) Rotate 270o+Mirror

When there's no geometry transformation, such as supporting LCD panel with pixel interface, or the rotation function is disabled, the rotation buffer will be also employed.

6.7.2.2 YUV422 to RGB888 Conversion

This submodule converts the YUV422 image data to RGB888 for LCD panel to display.

6.7.2.3 GBUF Interface

The graphics data is stored in the graphics buffer, which can be accessed via the MARB module. The graphics buffer is one block of the 4 blocks of 1-T SRAM. This submodule is the interface between LCD display control module and graphics buffer. It generates the request signal to the MARB module for reading the graphics data from the buffer for displaying on LCD panel. When MARB response to the request, it will send a DWORD data to LCD module. The graphics data in the buffer can be one of the following types:

1) 24 bit direct color: 24 direct color is specified by R G B 8:8:8 format. A DWORD can store 1 and 1/3 pixels.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B1								R0								G0								B0							

2) 24 bit direct color with alpha bit: 24 direct color is specified by R G B 8:8:8 format and an extra bit for alpha blending. A DWORD can store 1 pixel.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a								R0								G0								B0							

3) 18 bit direct color (RGB6:6:6): R, G and B all have 6 bits. A DWORD can store data of 1 pixel and all RG bits and the most 3 significant bits of B.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1				G1				B1						R0						G0						B0					

[1:0]

4) 16 bit direct color (RGB5:6:5): R and B both have 5 bits and G has 6 bits for this color format. A DWORD can store data of 2 pixels.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1								G1						B1						R0						G0					

5) 15 bit direct color with alpha bit (RGB5:5:5): R, G and B all have 5 bits and one extra bit for alpha blending. A DWORD can store data of 2 pixels.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a	R1					G1					B1					a	R0					G0					B0				

6) 12 bit direct color (RGB4:4:4): R, G and B all have 4 bits. A DWORD can store data of 2 pixels and RG data of another pixel.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G2				B2				R1				G1				B1				R0				G0				B0			

7) 8 bit indirect color: The host CPU writes the color code to the graphics buffer for each pixel. But the color code is not the real color data and it should be converted to RGB565 by the 256 entries before display on the LCD screen. An SRAM of 256x16bit is used for the color palette. There are only 256 kinds of color for display for this format. A DWORD can save data of 4 pixels.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P3								P2								P1								P0							

8) 4 bit indirect color: Each pixel only has 4bits to depict its color and the color data should be converted to RGB565 before displaying on the LCD screen. The color palette is also used and only 16 entries are used for this format. A DWORD can save data of 8 pixels.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P7				P6				P5				P4				P3				P2				P1				P0			

9) 2 bit indirect color: Each pixel only has 2bits to depict its color and the color data should be converted to RGB565 before displaying on the LCD screen. The color palette is also used and only 4 entries are used for this format. A DWORD can save data of 16 pixels.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15		P14		P13		P12		P11		P10		P9		P8		P7		P6		P5		P4		P3		P2		P1		P0	

10) 1bit indirect color: Each pixel only has 1bit to depict its color and the color data should be converted to RGB565 before displaying on the LCD screen. The color palette is also used and only 2 entries are used for this format. A DWORD can save data of 32 pixels.

When data of layer A is to be updated with overlay or alpha blending, GBUF interface will request to the MARB module and then the graphics data from MARB will merged with layer A data. At the space of updating layer A, pure layer B data (i.e., the region of layer B outside layer A) can be updated. The timing between GBUF interface and MARB can be referred in 8.5.4.

When video is continuous updating the LCD panel, graphics data can only be updated during the idle time of video stream. The idle information is told by the rotation submodule. If the video of the next 8 lines is coming again, the graphics data will be stopped to update the panel when updating of current line is completed.

6.7.2.4 Alpha Blending

Alpha blending can be done by mixing the color of two images when two screens are overlapped, then the blended result can be shown on the LCD screen or to be captured. The blend coefficient (alpha) is set by the host CPU.

Result image = layer A * alpha + layer B * (1-alpha)

The alpha blending function is controlled by the control register. One bit register is used to enable/disable this function. Alpha blending function can be enabled by setting ‘1’ to the alpha blending enable bit of the control register. For the color depth with alpha bit (such as 15 bit direct color with alpha bit), alpha blending is enabled by setting ‘1’ to both the alpha blending enable bit of the control register and the alpha bit of the color data. This submodule won’t output if the control bit of register is disabled.

6.7.2.5 Overlay

Two screens can be displayed by using overlay. In case of using overlay, a lower screen is displayed in a transparent color or the logical combination between two layers. For using overlay function, an overlay key color register should be specified with the overlay function enable bit set.

Generally, the block of Layer B is greater than Layer A and Layer B is the prioritized layer. When overlay function is enabled, for the pixels which the color code is equal to the overlay key color in layer B, the pixels may shows the color of layer A or the combination of layer A and layer B; else the pixels shows the color of layer B. In VC0568, we support the following kinds of overlay:

1) Transparent overlay. If the color data of pixels in priority layer is equal to the overlay key color, the pixel shows the color of non-priority layer; else shows that of priority layer;
2) And overlay. If the color data of pixels in priority layer is equal to the overlay key color, the pixel shows the color of (non-priority layer AND priority layer); else shows that of priority layer;
3) Or overlay. If the color data of pixels in priority layer is equal to the overlay key color, the pixel shows the color of (non-priority layer Or priority layer); else shows that of priority layer;
4) Invert overlay. If the color data of pixels in priority layer is equal to the overlay key color, the pixel shows the color of (INV non-priority layer); else shows that of priority layer.

The transparent color of can be set for layer A and layer B, and the priority layer can also be set by the host CPU, i.e., the priority layer can layer A or layer B. The overlay function is enabled/disabled by the control register. This submodule won't output pixel data if the function is disabled.

6.7.2.6 Color Palette

For 6.7.2.3, we know that the pixel data in the graphics buffer may be index color, such as 8bpp, 4bpp, 2bpp and 1pp. For these kinds of color type, the color should be converted to direct color. There is an LUT which has 256 entries to convert indirect color to direct color. It is called color palette. The color palette is stored in a block of SRAM which is 256x24bits.

The palette is NOT used for modes with color depths greater than 8 bits per pixel. In these cases the data stored in the graphics buffer is the actual color data, not an index. The appropriate bits describing the intensities of the red, green and blue components are retrieved from the graphics buffer and routed to the output is being used. The palette is entirely bypassed, and so these are referred to as direct-color modes. Note that this submodule should also output 24 bits color data for direct color.

The palette is used for modes with color depths of 8 bits per pixel or less. The color data stored in the graphics buffer and received by the palette is actually an index that selects a location within the palette in which the components of a color is specified. The 3-bytes of the selected color are sent from the palette to the output is being used. Due to this use of an index into a palette, these modes are referred to as indexed modes.

Using indexed modes allows the main display image to take up less space in the frame buffer and allows the actual displayed colors to be specified independently. The latter feature is used in such applications as video games.

6.7.2.7 Background Layer

Except layer A and layer B, there's another layer – background layer. The background layer consists only one color and the background color is stored in a 3-bytes register that can be written by the host CPU. The background layer can only be updated when there's no layer A or during the space of updating layer A.

When video is continuous updating the LCD panel, background layer can only be updated during the idle time of video stream. The idle information is told by the rotation submodule. If the video of the next 8 lines is coming again, the background layer will be stopped to update the panel when updating of current line is completed.

6.7.2.8 Layer Multiplexer

There are five kinds of pixel data: pure video from YUV to RGB conversion submodule, video and graphics merged by alpha blending from alpha blending submodule, video and graphics merged by overlay from overlay submodule, pure graphics from color palette submodule and BG layer from background layer submodule. The layer multiplexer chooses among these five kinds of pixel data. The five kinds of pixel data don't come at the same. Their output timing of this submodule is just as the figure 9-6. If data of continuous pixel are written to the LCDIF module, the row and column address of the pixel are written first; then the continuous pixel data is written to the LCDIF module one after another. The timing of the three layers is generated in rotation submodule, GBUF interface submodule and background layer submodule in figure 9-3, and is only delayed by some cycles before arriving at the layer multiplexer submodule. The pixel rate and the time from address to the data of the first pixel can be configured by the host CPU.

The pixel rate can be configured by the host CPU. The pixel data can be RGB565, RGB444, RGB666 and RGB888. For any kind of data to the LCDIF module, this submodule will generate correct data from the pixel data from the previous submodule. Because the pixel_data is [23:0], the unused bit is set zero, such as:

RGB565: {R[4:0], 3'b000, G[5:0], 2'b00, B[4:0], 3'b000, }
RGB444: {R[3:0], 4'b0000, G[3:0], 4'b0000, B[3:0], 4'b0000, }
RGB666: {R[5:0], 2'b00, G[5:0], 2'b00, B[5:0], 2'b00, }
RGB888: {R[7:0], G[7:0], B[7:0]}

6.7.2.9 Gamma Correction

Gamma correction is done on layer A to adjust the quality of the image which displayed on LCD panel.

6.7.2.10 Dithering

Now all the image data are RGB888 format, but there's no LCD panel of RGB888 type. The data type of prevail LCD panel are RGB565, RGB444 and RGB666. So before sending to the GRAM of the LCD panel or on-chip frame buffer, it is necessary to convert RGB888 to the above format. The quality is bad when cutting the least significant bits from the image data directly, so we should do some image algorithm processing. To convert to these types from RGB888, we should add some high frequency noise before truncating because truncating might remove the high frequency part from the previous signal.

6.8 LCD Interface

This submodule is the interface of VC0568 and LCD module. It receives the image and graphics data, and then sends to the LCD panel for update according to some predefined data format.

6.9 JPEG Codec

6.9.1 Overview

This JPEG Unit contains the following two functions:

- Compresses (encode) the input image, generates stream data and stores it in the JPEG buffer
- Reads the stream data from JPEG buffer, extracts (decode) the stream and outputs the image data to 8-line buffer.

6.9.2 JPEG Codec Core

JPEG unit contains a codec core including both encoder and decoder. The codec is to provide high performance for real-time video and still image compression and decompression. It can compress images up to 1.3Mega pixels. And it supports different YUV type: YUV422, YUV420, YUV411 and YUV400 (gray). There are several main sub-blocks in codec core: FDCT_IDCT, Quantization, Zigzag, Huffman coder and decoder.

The features of JPEG module are listed below.

- JPEG baseline algorithm
- Encoder and decoder (non-simultaneous)
- Variable size of image
- Support different YUV type when encoding: 422, 420, 411 and gray
- Support different YUV type when decoding: 444, 422, 420, 411 and gray
- 128 quantization tables for both Y and C in compression
- JPEG file header
- Real-time video compression and decompression
- Single clock Huffman coding and decoding
- Up to four programmable quantization tables in decoding
- Fully programmable Huffman tables (two AC and two DC) in decoding
- Fully programmable Minimum Coded Unit (MCU)
- Support up to four channels of component color
- Support restart marker when decoding
- Auto bit rate control

6.9.3 Input/Output Data

JPEG unit cannot do encoding and decoding at the same time.

Encoding function is used in capture mode and JPEG encode mode. In capture mode, input data is from 8-line buffers and the format is YUV422. In JPEG encode mode, input data is from host interface. In this case, host interface writes data into a register of jpeg module. The format of data can be YUV422, YUV420, YUV411 and gray. In both modes, output data is compressed jpeg, which is written to 64K JPEG buffer.

Decoding function is used in display mode and JPEG decode mode. In both modes, input data is from 64K JPEG buffer, and output data is written to 8-line buffer. YUV format can be YUV444/YUV422/YUV420/YUV411/YUV400.

For both encoder and decoder, the uncompressed data is 8-bit wide, and compressed data is 32-bit wide.

6.10 Graphics Engine

6.10.1 Graphics Engine Overview

The graphics engine is a specialized logic processor for 2D graphics operation such as BitBLTs and ROPs, line draw, etc. With the combination of these functions, the chip can do a lot of interesting work, such as pattern fill, area fill and drawing a line.

With the built-in graphics engine, the host CPU is freed from most of the display function with the following benefits:

- o Accelerated graphics operations produce smooth screen updates without visible start-and-stop or slow-down during heavy CPU usage by an application;
- o The CPU is freed to perform time-critical or real-time operations, while the graphics engine renders the display.

6.10.2 Bresenham Line Drawing

Bresenham line drawing algorithm is a good algorithm for hardware implementation. VC0568 supports line drawing of arbitrary angle using Bresenham algorithm.

6.10.3 BitBLT Engine

The graphics controller provides a hardware-based BitBLT engine to shift the workload of moving blocks of graphics data from the host CPU. Although the BitBLT is often used simply to copy a block of graphics data from the source to the destination, it also has the capability of more complex functions. The BitBLT engine is capable of receiving as input three different blocks of graphics data: the source, the destination and the pattern. The source data may exist in the display memory (on-chip graphics buffer). The pattern data always represents an 8x8 block of pixels that is from the pattern buffer. The input destination data is the data already residing at the destination in the display memory prior to a BitBLT operation. The output destination data is the data written to the destination as a result of a BitBLT operation.

Before programming the graphics engine, we should first read the status register to see if the graphics engine is currently busy. The graphics engine should not be programmed in any way until all operations are complete and the engine is idle.

The BitBLT engine may be configured to take various combination of the source, pattern, and input destination as operands, and perform either bit-wise logical operations or per-pixel operations to generate the output destination data. It is intended that the BitBLT engine should perform these bit-wise or per-pixel operations on color graphics data whose color

depth matches that of the rest of the graphics system, i.e., the bit-depth of the graphics in the on-chip graphics SRAM. The format of the graphics data in the graphics buffer can be 4bpp, 8bpp, 16bpp (with or without alpha bit), 24bpp and 32bpp (with alpha bit), so the graphics should also support these formats. So, if either the source or pattern is monochrome, the BitBLT engine has the ability to put either block of graphics data through a process called “color expansion”, which converts the monochrome graphics data to color data. Since the destination is always somewhere in the display memory, it is assumed that any data already in the destination block will be of the appropriate color depth.

6.11 Chip Controller

6.11.1 Chip Controller Oerview

The chip controller has the following functions:

- Reset scheme
- Clock scheme
- Power management
- Interrupt controller
- PLL control
- Pad control

6.11.2 Reset Scheme

VC0568 has four kinds of reset signal.

Power on reset

Power on reset is the external asynchronous reset sync by host bus clock, and it resets the CPM registers.

Hardware reset

Hardware reset is delayed 15-20us after the power on reset. During hardware reset the PLL is bypassed, the internal clock is switched to bus clock instead of the output of PLL and the internal logic can be reset. After hardware reset the internal clock is switched to output of PLL. The software should delay 512 input clocks before starting other operation.

Global software reset

The global software reset will reset the whole chip functional module except some registers of the chip controller.

Block software reset

Each functional block has one reset which can be activated by setting the corresponding reset bit in control register.

6.11.3 Clock Scheme

The original clock input is from the CLKXIN pin. During power on reset the PLL is bypassed, all clocks are running and the MCLK is switched to BCLK instead of the output clock of PLL, so the internal logic can be reset. After power on reset the MCLK is switched back to PLL output clock and into idle state.

6.11.4 Power Management

When the VC0568 based system is not operating, the host can reduce the systems to power-down mode or stand-by mode so as to save power. The following are the possible modes:

1) Work mode
2) Standby mode
3) Power down mode

When the VC0568 based system is not operating, the firmware can put the systems into power saving modes. So there is another mode:

4) Power saving mode

Also, while in operation, the power can be saved with the following method:

1) Clocks for some blocks can be turned off, under firmware control
2) Programmable module working clock rate. (6MHz, 12MHz, 24Mhz)
3) Peripheral device's power can be turned off, like sensor interface.

7. Work mode

The following operation modes are supported in VC0568 accompanied with Camera module and LCD panel:

7.1 Viewfinder Mode

In viewfinder mode, continuous video stream (up to 30fps) from sensor is shown on LCD panel with no frame delay. The user can adjust the zooming factor and image of different resolution will be displayed on LCD. When the user is happy with the image frame on display, he or she can push a button and capture it. Then the chip will go to the capture mode.

In viewfinder mode, the following features are supported:

- Video streaming at high frame rate (up to 30fps)
- No frame delay (What you see is what you capture!)
- Zoom in/out
- Add frame on the video
- Configurable display window

7.2 Capture Mode

VC0568 can capture still image, multi-shot and video clip.

When capture still image, the user first chose an appropriate frame in viewfinder mode, then he can press a capture button and the next frame from the image sensor will be captured. The JPEG compression image is stored in the internal SRAM and the host CPU can read the data. The user can choose the image size to be captured and whether there is a ‘frame’ outside the captured image or not. VC0568 have on-chip buffer for capturing high-quality still image up to 1.3M pixels in JPEG format. Thumbnail with configurable size can be generated accompanied with the JPEG image data.

When capture multi-shot, the user first push the capture button, then VC0568 will capture consecutive TBD images at TBD internal from image sensor. The images will be resized to VGA size and compressed as JPEG files along with a thumbnail version, and stored in on-chip buffer. Each image will be assigned a unique number internally for reference. The thumbnail images will be sent to LCD for display side by side. When user selects one of the thumbnail images, the corresponding VGA sized image will be sent to LCD for display at right size and bit depth. 2-D graphics functions (including special image effect) are supported in capture mode.

After the user pushes the capture button, VC0568 will capture the video stream from image sensor, until he/she pushes the stop button. The video stream will be compressed as AVI file, and then sent to baseband processor. At the same time, video frames of right size and bit-depth will be sent to LCD for display. 2-D graphics functions (including special image effect) are supported in capture mode. VC0568 will allow user to capture video up to 320 x 240.

The captured still image, multi-shot and video clip can be sent to the host CPU or the storage card.

In capture mode, the following features are supported

- Capture still images up to 1.3Mega resolution
- Capture SIF (320x240), QSIF (160x120) and QQSIF (80x60) MJEPG video clip
- Capture still image with various resolution of thumbnail
- Capture still image with overlay
- Capture still image with alpha-blending
- Capture multi-shot up to VGA resolution

7.3 Display Mode

Under display mode, JPEG image or AVI saved in host memory or the storage card will be downloaded to VC0568, where it will be prepared (right size and bit depth) for LCD display. 2-D graphics functions are supported in display mode.

In display mode, the following features are supported:

- Receive JPEG still image or MJPEG video clip from host CPU
- Decode and display on LCD with right size and color depth
- Support various image format: 4:4:4, 4:2:2, 4:1:1, 4:2:0, 4:0:0
- Support JPEG image up to resolution 1.3Mega pxiels
- Resize the image with anti-aliasing pre-filtering
- Add frame to the image
- Add special effect to the image

7.4 JPEG Decode Mode

It is possible to use VC0568 as a JPEG decoding engine. When it is used as a JPEG decoder, the JPEG data is written to internal 64KB SRAM similar to the display mode and then decompressed by JPEG codec. But the decompressed data is sent back to the host CPU, other than LCD panel. The decoding result (YUV444/YUV422/YUV411/YUV420) is written in 8-line buffer. When data of 8 lines (YUV422/YUV411/YUV444) or 16 lines (YUV420) is decompressed, an interrupt is issued and the host CPU will read the data. Then

the host CPU will restart the decompression. Then ‘Interruption-read-restart’ repeats until decompression of one frame complete.

In JPEG decoder mode, the following features are supported:

- Receive JPEG image from host CPU
- Decode the JPEG image to YUV image data
- Add optional special effect to the image
- Resize the image data to specific resolution
- Send the YUV image data back to host CPU
- Support 4:4:4, 4:2:2, 4:1:1, 4:2:0
- Support JPEG image up to VGA resolution

7.5 JPEG Encode Mode

VC0568 can also be used as a JPEG encoder. At this mode, image data can be encoded into JPEG format by writing the image data in the specific registers one by one. Data of YUV422/YUV411/YUV420/YUV400 is written to the register in block sequence (8x8). The compressed JPEG data is saved in the internal 64KB SRAM and then is read by the host CPU after one frame is completed.

In JPEG encoder mode, the following features are supported:

- Receive YUV image data from host CPU
- Encode the image to JPEG format
- Send the JPEG image back to host CPU
- Support 4:2:2, 4:1:1, 4:2:0
- Support JPEG image up to VGA resolution

7.6 Through Mode

In this mode, the camera functions are disabled. The host CPU will read/write the LCD module directly. VC0568 support two ways for interfacing LCD panel in this mode: i) through registers for forwarding the index and the data to the LCD panel. Conversion from 8bit to 16bit of can be supported; ii) directly mapping the data and address from host CPU to LCD panel and there is no registers between host CPU and LCD panel, just like the host CPU interface LCD panel directly.

In through mode, the following features are supported:

- Video function is disabled
- Host CPU update LCD panel directly
- Directly pin mapping between host CPU and LCD module
- Index register for address and data for host CPU to update LCD

- Low power

7.7 Direct Display Mode

In this mode, the camera functions are disabled. The host CPU will writes the graphics data to the LCD panel through the on-chip graphics SRAM. 2-D graphics engine is supported at this mode. Direct data (16bpp, 24bpp and 32bpp) and indirect data (4 and 8bpp) are supported to write to the SRAM. For indirect data, a color palette table is used for converting to 16-bit direct color data.

In direct display mode, the following features are supported:

- Video function is disabled
- Host CPU reads/writes the frame buffer
- Optional BitBLT operation
- Optional 256 Raster operation
- Optional all angels line draw

8.Register Map

8.1 BIU register

MEM_LOW_WORD

Register	BIU_BASE_ADDR+45h								BIU_BASE_ADDR+44h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MEM_LOW_WORD															
R/W	RW															
De fault value	0															

Memory low address ([15:0]) when multiplex bus is used.

MEM_HIGH_WORD

Register	BIU_BASE_ADDR+49h								BIU_BASE_ADDR+48h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								MEM_HIGH_WORD							
R/W	-								RW							
De fault value	-								0							

Memory high address ([23:16]) when multiplex bus is used.

MEM_FLG

Register	BIU_BASE_ADDR+50h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	MEM_FLG							
R/W	-							
De fault value	-							

Memory port when multiplex bus is used.

ADDR_MEM_LOW

Register	BIU_BASE_ADDR+53h								BIU_BASE_ADDR+52h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	ADDR_MEM_LOW															
R/W	RO															
De fault value	-															

Memory address [15:0] index. This is a read only register.

ADDR_MEM_HIGH

Register	BIU_BASE_ADDR+55h								BIU_BASE_ADDR+54h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								ADDR_MEM_HIGH							
R/W	-								RO							
De fault value	-															

Memory address [23:16] index. This is a read only register.

MUL_U2IA

Register	BIU_BASE_ADDR+80h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved						MUL_U2IA	
R/W	-						RW	
De fault value	-						2'b0	

High 2 bits of register address, it is reserved in the current chip.

SEL_WRITE_READ

Register	BIU_BASE_ADDR+86h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							SEL_WRITE_READ
R/W	-							RW
De fault value	-							1'b0

Select read/write memory.
0: select read memory.
1: select write memory.

SEL_PORT

Register	BIU_BASE_ADDR+88h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							SEL_PORT
R/W	-							RW
De fault value	-							1'b1

Select port access mode.

SEL_8_16

Register	BIU_BASE_ADDR+8ch							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							SEL_8_16
R/W	-							RW
De fault value	-							1’b0

This register express data bus width of host.

0: it is 8 bit data width.

1: it is 16 bit data width.

BYPASS_SEL

Register	BIU_BASE_ADDR+90h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							BYPASS_SEL
R/W	-							RW
De fault value	-							1’b0

Bypass mode flag.

0: normal mode.

1: it is bypass mode or direct through mode.

REG_8_FLG

Register	BIU_BASE_ADDR+b0h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	REG_8_FLG							
R/W	-							
De fault value	-							

Register port when 8 bit multiplex bus is used.

REG_8_LOW_WORD

Register	BIU_BASE_ADDR+b2h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	REG_8_LOW_WORD							
R/W	RW							
De fault value	8’b0							

Register low address when 8 bit multiplex bus is used.

REG_8_HIGH_WORD

Register	BIU_BASE_ADDR+b4h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	REG_8_HIGH_WORD							
R/W	RW							
De fault value	8'b0							

Register high address when 8 bit multiplex bus is used.

MEM_8_FLG

Register	BIU_BASE_ADDR+b6h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	MEM_8_FLG							
R/W	-							
De fault value	-							

Memory port when 8 bit multiplex bus is used.

PLL_M

Register	BIU_BASE_ADDR+c1h								BIU_BASE_ADDR+c0h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								PLL_M							
R/W	-								RW							
De fault value	-								9'h03							

Feedback 9-bit divider control pins

PLL_TST

Register	BIU_BASE_ADDR+c2h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved						PLL_TST	
R/W	-						RW	
De fault value	-						2'b0	

PLL_M and PLL_N divider test mode
PLL_TST[0:1] = 0:0 N divider test and normal operation TST_OUT = XIN/N
PLL_TST[0:1] = 1:1 M divider test TST_OUT = XIN/M

PLL_N

Register	BIU_BASE_ADDR+c3h							
Bit number	15	14	13	12	11	10	9	8
Bit field name				PLL_N				
R/W				RW				
De fault value				5'h3				

Input 5-bit divider control pins

PLL_BP

Register	BIU_BASE_ADDR+c4h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							PLL_BP
R/W	-							RW
De fault value	-							1'b0

Bypass the PLL ; Active high

PLL_PD

Register	BIU_BASE_ADDR+c5h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							PLL_PD
R/W	-							RW
De fault value	-							1'b0

Power down control Active high
When PLL_PD = PLL_OE =PLL_BP. the PLL work normal

PLL_OE

Register	BIU_BASE_ADDR+c6h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							PLL_OE
R/W	-							RW
De fault value	-							1'b0

CLK_OUT enable
When PLL_PD=1&PLL_OE = 1 CLK_OUT =0

PLL_OD

Register	BIU_BASE_ADDR+c7h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved						PLL_OD	
R/W	-						RW	
De fault value	-						2'b0	

Output divider control

PLL_CLKCTRL

Register	BIU_BASE_ADDR+c8h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	PLL_CLKCTRL							
R/W	RW							
De fault value	8'h01							

Pll clock output divider

ADDR_DELAY_SEL

Register	BIU_BASE_ADDR+f0h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	-				ADDR_DELAY_SEL			
R/W	-				RW			
De fault value	-				4'h8			

Delay cell amount used in address bus.

DATA_OUT_DELAY_SEL

Register	BIU_BASE_ADDR+f2h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	-				DATA_OUT_DELAY_SEL			
R/W	-				RW			
De fault value	-				4'h8			

Delay cell amount used in data_out bus.

DATA_IN_DELAY_SEL

Register	BIU_BASE_ADDR+f2h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	-				DATA_IN_DELAY_SEL			
R/W	-				RW			
De fault value	-				4'h3			

Delay cell amount used in data_in bus.

DATA_OEN_DELAY_SEL

Register	BIU_BASE_ADDR+f4h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	-				DATA_OEN_DELAY_SEL			
R/W	-				RW			
De fault value	-				4'h3			

Delay cell amount used in data_oen bus.

CS_DELAY_SEL

Register	BIU_BASE_ADDR+f6h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	-				CS_DELAY_SEL			
R/W	-				RW			
De fault value	-				4'h3/4'h0			

Delay cell amount used in cs bus.
If CONF[1:0] = 2'b11, the default value is 4'h3, otherwise the default value is 4'h0.

8.2 SIF register

Sensor mode control register

Register	SIF_BASE_ADDR+00h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Sync_mode	Yuv_flag	Sync_en	Polarity _mode	Polarity_en	Snr_pwr	Snr_en	Snr_rst
R/W	RW							
De fault value	00h							

[7]: Sync_mode, sensor output data sync mode:

0: Normal Sync Mode;

1: CCIR656 Sync Mode;

[6]: yuv_flag, indicates the output data format from sensor:

1: YUV 4:2:2

0: bayer pattern

[5]: sync_en, enable the sync-generator . If this bit is “0”, syncgen module will stop at once.

[4]: Polarity_mode

1: pullup output pins to sensor when polarity_en =1;

0: pulldown output pins to sensor when polarity_en = 1.

[3]: Polarity_en

1: to meet some sensors requirements, pullup or pulldown output signal pins to sensor.

0: output normal signals to sensor.

[2]: snr_pwr, sensor power on

[1]: snr_en, sensor enable

[0]: snr_rst, sensor reset;

Sensor operator mode control register

Register	SIF_BASE_ADDR+01h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Capture	Sg_en	Serial_sel	SIClk_rv	Snrclk_rv	Serial_sel	Obreg_en	Sync_dir
R/W	RW							
De fault value	01h							

[7]: start capture
[6]: sg_en, enable output sync signals to ISP
1: Sycn generator operates and outputs normal sync signals to ISP;
0: Sync generator outputs 0 to ISP begin at next vsync coming.
[5]: serial_sel, select 2-wire serial bus or 3-wire serial bus.
1: 3-wire serial bus is selected.
0: 2-wire serial bus is selected. The pins map to 2-wire serial bus signals:
[4]: SICLK_rv
1: serial_bus clock reversed polarity
0: serial_bus clock common polarity
[3]: Snrclk_rv: sensor Clk reverse enable
1: sensor clk invert;
0: sensor normal;
[2]: serial_sel:
1: OV sensor used;
0: 2-wire serial bus normal;
[1]: Obreg_en: Obreg enable
1: Subtract one value from TASC data. The value is in the obreg register.
0: don’t subtract one value from TASC data
[0]: Sync direction:
1: sync from sensor;
0: output sync to sensor.

YUV data format register

Register	SIF_BASE_ADDR+02h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	tref_lh_blk		tref_lt_blk		Internal delay count		YUV data format	
R/W	RW							
De fault value	b0h							

[7:6] tref_lh_blk: Timing reference value for the head of blanking line in CCIR656, tref_lh_blk;

[5:4] tref_lt_blk: Timing reference value for the tail of blanking line in CCIR656, tref_lt_blk;

[3:2] Internal delay count (sif_clk) when handle YUV data

[1:0]: the YUV data format from sensor

00: sensor outputs 'UYVY';

01: sensor outputs 'VYUY';

10: sensor outputs 'YUYV';

11: sensor outputs 'YVYU';

Vertical synchronous signals control register

Register	SIF_BASE_ADDR+03h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserved		Vref_full	Vd_mode	Vsync_mode	V_bypass	V_polarity	reserved
R/W	RW							
De fault value	1eh							

[5]: Vref_full, internal verf is high constantly because sensor's Hsync is constant level between rows.

0: when sensor's Hsync has level change between rows.

1: when sensor's Hsync has not any level change between rows,

[4]: vd_mode – vertical sync signal for sensor width selection

0: vsync send to sensor more than 1 row

1: vsync send to sensor less than 1 row

[3]: vsync_mode – vertical sync signal for isp width selection

0 : vsync send to ips more than 1 row;

1: vsync send to isp less than 1 row

[2]: v_bypass – input vertical signal bypass selection.

0 : SIF generate vertical signal for ISP.

1: the input vertical signal is bypass to ISP

[1]: v_polarity – input vertical signal polarity flag

1: input vertical signal is active high;

0: input vertical signal is active low;

[0]: reserved

Horizontal synchronous control register

Register	SIF_BASE_ADDR+04h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	lh_vld		lt_vld		H_bypass	H_polarity	H_flag	reserved
R/W	RW							
De fault value	1dh							

[7:6] tref_lh_vld: Timing reference value for the head of valid line in CCIR656, tref_lh_vld;

[5:4] tref_lt_vld: Timing reference value for the tail of valid line in CCIR656, tref_lt_vld;

[3] h_bypass – input horizontal signal bypass selection.

0: SIF generate horizontal signal to ISP;

1: the input horizontal signal is bypass to ISP;

[2] h_polarity – input horizontal signal polarity flag,

0: input horizontal signal is active low;

1: input horizontal signal is active high;

[1] h_flag – input horizontal signal flag

0: input horizontal signal is href or hsync;

1: input horizontal signal is drdy;

[0] Reserved

Counter of master clock output to sensor

Register	SIF_BASE_ADDR+05h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	snrclk_cnt							
R/W	RW							
De fault value	01h							

This register is used to program the master clock output to sensor.

Snrclk = (SIF clock) / (snrclk_cnt +1)

Pixel rate select register

Register	SIF_BASE_ADDR+06h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserved		pixrate_sel					
R/W	RW							
De fault value	03h							

[5:0]: Pixel rate select—input data clk rate value.

This value is decided by input data rate. EXP: input data clock is 6Mhz, SIF clock is 24Mhz, then (SIF clock)/ (input data rate) = 24/6 = 4. Then this value is 4 - 1 = 3.

Serial bus clock factor register

Register	SIF_BASE_ADDR+07h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	SICLK_cnt							
R/W	RW							
De fault value	7fh							

This register is used to program the working frequency of the serial bus.

Frequency of SICLK = (SIF clock) / SICLK_factor

Serial bus clock counter register

Register	SIF_BASE_ADDR+08h								SIF_BASE_ADDR+09h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Serial_bus_clkcnt															
R/W	RW															
De fault value	010ch															

This register is to make sure that the working frequency of serial bus is 100k when the sensor interface clock changes.

Frequency of serial bus = (SIF clock) / (serial_bus_clkcnt)

Hsync output start/stop pointer registers

Register	SIF_BASE_ADDR+0ch								SIF_BASE_ADDR+0dh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	hd_start															
R/W	RW															
De fault value	0001h															
Register	SIF_BASE_ADDR+0eh								SIF_BASE_ADDR+0fh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	hd_stop															
R/W	RW															
De fault value	0024h															

These registers indicate the start and end position of hsync, which is outputted to sensor. These registers are used for those sensors which can't generate hsync and vsync signals.

Hsync start pointer registers

Register	SIF_BASE_ADDR+10h								SIF_BASE_ADDR+11h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	hsync_start															
R/W	RW															
De fault value	0001h															

These registers are used to indicate the 1st column position of 1st active line of verf when SIF output Vsync, Hsync to sensor. Verf is used to indicate the lines which dataen active for ISP or IPP. Vref_starth/l(reg 20,21) indicate the 1st line position of verf in one frame. Href_starth/l(reg14,15) indicate the 1st column position of href in one line. Hsync_starth/l must be set same as Href_starth/l, otherwise, for example, Hsync_start = 1, but Href_start=0, 1st dataen of 1st line of every frame's will be lost.

Href start/length pointer registers

Register	SIF_BASE_ADDR+14h								SIF_BASE_ADDR+15h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	href_start															
R/W	RW															
De fault value	0001h															
Register	SIF_BASE_ADDR+16h								SIF_BASE_ADDR+17h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	href_length															
R/W	RW															
De fault value	0280h															

These registers indicate the start position and length of the horizontal reference signal, which is outputted to other submodules.

Vsync output (VD) start/stop pointer registers

Register	SIF_BASE_ADDR+18h								SIF_BASE_ADDR+19h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	hsync_start															
R/W	RW															
De fault value	0031h															
Register	SIF_BASE_ADDR+1ah								SIF_BASE_ADDR+1bh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	hsync_start															
R/W	RW															
De fault value	0088h															

These registers indicate the position of vsync, which is output to sensor.

Vsync start/stop pointer registers

Register	SIF_BASE_ADDR+1ch								SIF_BASE_ADDR+1dh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	vsync_start															
R/W	RW															
De fault value	0000h															
Register	SIF_BASE_ADDR+1eh								SIF_BASE_ADDR+1fh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	vsync_stop															
R/W	RW															
De fault value	0001h															

These registers indicate the position of vsync, which is output to ISP/Special Effect.

Vref start/stop pointer registers

Register	SIF_BASE_ADDR+20h								SIF_BASE_ADDR+21h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	vref_start															
R/W	RW															
De fault value	0000h															
Register	SIF_BASE_ADDR+22h								SIF_BASE_ADDR+23h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	verf_stop															
R/W	RW															
De fault value	01e0h															

These registers indicate the position of vref, which is output to other submodules.

HD_fall start/stop pointer registers

Register	SIF_BASE_ADDR+24h								SIF_BASE_ADDR+25h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	HD_fall_start															
R/W	RW															
De fault value	0050h															
Register	SIF_BASE_ADDR+26h								SIF_BASE_ADDR+27h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	HD_fall_stop															
R/W	RW															
De fault value	0080h															

These registers indicate the position of AECNT pulse, which is output to sensor.

VD_fall start/stop pointer registers

Register	SIF_BASE_ADDR+28h								SIF_BASE_ADDR+29h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VD_fall_start															
R/W	RW															
De fault value	0000h															
Register	SIF_BASE_ADDR+2ah								SIF_BASE_ADDR+2bh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VD_fall_stop															
R/W	RW															
De fault value	0000h															

These registers indicate the position of serial_en, which is the enable signal for serial bus.

Colmax registers

Register	SIF_BASE_ADDR+2ch								SIF_BASE_ADDR+2dh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	colmax															
R/W	RW															
De fault value	0287h															

Thses registers indicate the maximum of columns in a frame.

Rowmax register

Register	SIF_BASE_ADDR+2eh								SIF_BASE_ADDR+2fh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	rowmax															
R/W	RW															
De fault value	01e7h															

Thses registers indicate the maximum of rows in a frame.

Exposure time register

Register	SIF_BASE_ADDR+30h								SIF_BASE_ADDR+31h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	extime															
R/W	RW															
De fault value	0000h															

These registers are sensor exposure time value, which is sent to the sensor through the AECNT pin. Pay attention this value can't beyond the value rowmax.

Obreg register

Register	SIF_BASE_ADDR+33h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	obreg							
R/W	RW							
De fault value	00h							

When obreg_en bit is 1, subtract this value from the image data of TASC.

Serial bus device address register

Register	SIF_BASE_ADDR+35h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	SerAckChk	Serial bus device address						
R/W	RW							
De fault value	91h							

[7] SerAckChk: check sensor (serial bus slave device) acknowledge bit.

0: don't check acknowledge bit. No matter sensor acknowledge or not, execute serial bus communication.

1: check acknowledge bit. If sensor does not acknowledge, stop serial bus communication at once.

[6:0] Serial bus device address: this 7 bit device will be connect 1bit "1" or "0" as the least bit according to read or write serial bus operation.

Sensor Write Data registers

Register	SIF_BASE_ADDR+36h								SIF_BASE_ADDR+37h								SIF_BASE_ADDR+38h							
Bit number	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	SnrWrtData																							
R/W	R/W																							
De fault value	000000h																							

These registers are the data to be written into sensor registers through 2-wire serial bus.

Sensor access control register

Register	SIF_BASE_ADDR+39h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved						Snr_rd	Snr_wrt
R/W	RW							
De fault value	00h							

[1]: read sensor trigger bit. Once this bit is set, the 2-wire serial bus will start to the read sensor register operation.

0: after read operation begin or later, set to "0" for next read operation;

1: let SIF to start read sensor register operation;

[0]: write sensor trigger bit. Once this bit is set, the 2-wire serial bus will start to the write sensor register operation.

0: after write operation begin or later, set to "0" for next write operation;

1 : let SIF to start write sensor register operation;

Sensor Address register

Register	SIF_BASE_ADDR+3ah							
Bit number	7	6	5	4	3	2	1	0
Bit field name	SnrAddr							
R/W	RW							
De fault value	00h							

This register is the sensor register address.

Sensor status register

Register	SIF_BASE_ADDR+3bh							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserved							Snr_status
R/W	RW							
De fault value	00h							

[0]: snr_status

0: Serial bus is idle;

1: The sensor is transferring data with the chip (read or write) through serial bus.

Sensor Read Data register

Register	SIF_BASE_ADDR+3ch								SIF_BASE_ADDR+3dh								SIF_BASE_ADDR+3eh							
Bit number	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	SnrRdData																							
R/W	R/W																							
De fault value	000000h																							

These registers are the data read from the sensor registers through serial bus.

Serial bus state register

Register	SIF_BASE_ADDR+3fh							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserved					serial_en	bus_idle	bus_noack
R/W	RW							
De fault value	00h							

[2]: serial_en
1: 3-wire Serial bus is ready to transfer data.
0: Can't read/write data through serial bus.
[1]: bus_idle
1: 2-wire serial bus master is ready to transfer data.
0: Can't read/write data through 2-wire serial bus.
[0]: bus_noack
1: 2-wire serial bus slave does not acknowledge.
0: acknowledge from serial bus slave device has arrived.

AE window start/ stop point registers

Register	SIF_BASE_ADDR+40h								SIF_BASE_ADDR+41h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	AEwin_start															
R/W	RW															
De fault value	0000h															
Register	SIF_BASE_ADDR+42h								SIF_BASE_ADDR+43h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	AEwin_stop															
R/W	RW															
De fault value	0001h															

These registers indicate the position of AE window, which is suitable time for SIF to set the sensor's AE registers.

AE pin control register

Register	SIF_BASE_ADDR+44h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserved				AE_sbus	AE_mode		AE-pol
R/W	RW							
De fault value	01h							

This register is set during the initialization.

[3]: ae_sbus, AE controller indicator

1: serial bus command controls the AE

0: AECNT pin controls the AE

[2:1]: AE_mode, AECNT output signal format when AECNT pin controls the AE

00: multi pulse

01: single pulse

10: level

[0]: AE_pol, AECNT output signal polarity

1: common polarity

0: reversed polarity

AE bus control register

Register	SIF_BASE_ADDR+45h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserved			Batch_done	AE_done	Addr_num		
R/W	RW							
De fault value	00h							

This register is set everytime firmware wants to batch write the sensor register.

[4]: batch_done.

1: Firmware has updated all the data for batch write sensor's none-AE registers. This bit is set by host and cleared by SIF, done clear mode.

[3]: AE_done.

1: Firmware has updated all the data for batch write sensor's AE registers. This bit is set by host and cleared by SIF, done clear mode.

[2:0]: the number of separate sensor register addresses in a batch. These bits are set by host and cleared by SIF, done clear mode.

For 2-wire serial bus, the maximum is 7; for 3-wire serial bus, the maximum is 2.

Byte/bit count registers for batch write

Register	SIF_BASE_ADDR+46h								SIF_BASE_ADDR+47h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	B_cnth								B_cntl							
R/W	RW															
De fault value	0000h															

For 2-wire serial bus, these registers are the byte count for each batch write. Since each serial bus operation supports to write 3 bytes at the most, 2 bits in the byte stand for the byte count of a serial bus write operation.

B_cntl [1:0]: the byte count for the serial bus to write for 1st sensor address;
B_cntl [3:2]: the byte count for the serial bus to write for 2nd sensor address;
B_cntl [5:4]: the byte count for the serial bus to write for 3rd sensor address;
B_cntl [7:6]: the byte count for the serial bus to write for 4th sensor address;
B_cnth [1:0]: the byte count for the serial bus to write for 5th sensor address;
B_cnth [3:2]: the byte count for the serial bus to write for 6th sensor address;
B_cnth [5:4]: the byte count for the serial bus to write for 7th sensor address;

For serial bus, these registers are the bit count for each batch write.

B_cntl [6:0]: the bit count for the serial bus to write for 1st sensor address;
B_cnth [6:0]: the bit count for the serial bus to write for 2nd sensor address.

AE data registers

There are 20 8-bit reigsters to hold the data for batch writing sensor registers. If the serial bus is used, the AE data register should set as the following format:

AE_data	Sensor register address 1
AE_data	Data1
AE_data	Data2
AE_data	Data3
AE_data	Sensor register address 2
AE_data	Data1
AE_data	Data2
……,	……,

If the serial bus is used, the AE data registers should be set as:

Serialbus_data	Serial bus data [7:0]
Serialbus_data	Serial bus data [15:8]
……,	……,
Serialbus_data	Serial bus data [7:0]
Serialbus_data	Serial bus data [15:8]
……,	……,

edge_chos

Choose posedge or negedge sample sensor inputs.

Register	SIF_BASE_ADDR+5ch							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							edge_chos
R/W	RW							
De fault value	00h							

sif counter reset

Register	SIF_BASE_ADDR+5dh							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							sif_cnt_reset
R/W	RW							
De fault value	00h							

This bit is used to reset SIF internal counter, but not reset SIF registers setting. Set by biu and reset by SIF self.

[0] : sif counter reset;

1’b1: reset counter;

1’b0: normal operation.

FL_PWM_EN

Register	SIF_BASE_ADDR+70h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							FL_PWM_EN
R/W	RW							
De fault value	00h							

[0] : Flash light charge circuit operation enable;

0: Flash power charge disable;

1: Flash power charge enable;

FL_POLARITY

Register	SIF_BASE_ADDR+71h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							FL_POLARITY
R/W	RW							
De fault value	01h							

[0] : Flash light charge signal polarity;
0: low active;
1: high active;

FL_PWM_CNT_TGT

Register	SIF_BASE_ADDR+72h								SIF_BASE_ADDR+73h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FL_PWM_CNT_TGT															
R/W	RW															
De fault value	053ch															

The FL_PWM_CNT_TGT = sif_clk frq / charge frq.

FL_PWM_CNT_DUTY

Register	SIF_BASE_ADDR+74h								SIF_BASE_ADDR+75h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FL_PWM_CNT_DUTY															
R/W	RW															
De fault value	04b6h															

FL_PWM_CNT_DUTY = FL_PWM_CNT_TGT * (1- charge duty)

FL_DUTY

Register	SIF_BASE_ADDR+76h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	FL_DUTY							
R/W	RW							
De fault value	0ah							

The fl_duty is used to control the charge pulse duty.

02H: 1/2 duty;
03H: 1/3 duty;
04H: 1/4 duty;
05H: 1/5 duty;
06H: 1/6 duty;
07H: 1/7 duty;
08H: 1/8 duty;
09H: 1/9 duty;
0aH: 1/10 duty;
0bH: 1/11 duty;
0cH: 1/12 duty;
0dH: 1/13 duty;
0eH: 1/14 duty;
0fH: 1/15 duty;
10H: 1/16 duty;

M10US_CNT_EN

Register offset	SIF_BASE_ADDR+80h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							M10US_CNT_EN
R/W	R/W							
Default	01h							

[0] : Enable internal counter to generate flg_m_10us_clk which period is 10us and is the base of flash light trigger.

0: disable internal counter to generate flg_m_10us_clk;
1 : enable internal counter to generate flg_m_10us_clk;

M10US_TGT

Register offset	SIF_BASE_ADDR+81h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	M10US_TGT							
R/W	R/W							
Default	78h							

This register's value = 10us/sif_clk_period; So if sif_clk is 24M, M10US_TGT = 10us/41.7ns=239(d)=ef(h);

DELAY_FLASH_TGT

Register	SIF_BASE_ADDR+83h								SIF_BASE_ADDR+84h								SIF_BASE_ADDR+85h							
Bit number	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DELAY_FLASH_TGT																							
R/W	R/W																							
De fault value	00060eh																							

Delay period after frame_end interrupt when reg01h[7] capture is high. This register value is base on SIF clock unit.

TRIGGER_TGT

Register	SIF_BASE_ADDR+86h								SIF_BASE_ADDR+87h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	TRIGGER_TGT															
R/W	RW															
De fault value	0032h															

Flash light trigger active period, base on 10 us unit.

8.3 ISP register

CTL_BAS control register

This register controls the basic ISP function of ISP module

Register	ISP_BASE_ADDR + 01h								ISP_BASE_ADDR + 00h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	CTL_ISP_BAS															
R/W	RW								RW							
Default value	38h								01h							

[15] Gray image mode
- 1 gray image
- 0 normal colorful image

[14] Digital RGB Gain mode
- 1 Enable
- 0 Disable

[13] Y Gamma Correction function
- 1 Enable
- 0 Disable

[12] Color Matrix Correction function
- 1 Enable
- 0 Disable

[11] RGB Gamma Correction function
- 1 Enable
- 0 Disable

[10] DPD/DPC and noise removal function
- 1 Enable
- 0 Disable

[9] Lens fall off function
- 1 Enable
- 0 Disable

[8] Sharpen edge function
- 1 Enable
- 0 Disable

[7:6] Reserved

[5] Bayer pattern mode
- 1 BG line first
- 0 RG line first

[4] Bayer pattern mode
- 1 G pixel first
- 0 non G pixel first

[3-0] Image pixel rate supported
- 0001: one pixel per 2 clock cycles

0011: one pixel per 4 clock cycles
0101: one pixel per 6 clock cycles
0111: one pixel per 8 clock cycles
1011: one pixel per 12 clock cycles
1111: one pixel per 16 clock cycles

image size width

Register	ISP_BASE_ADDR + 07h								ISP_BASE_ADDR + 06h							
Bit number	15~12				11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				IMG_Wh				IMG_WL							
R/W					RW											
Default value					280h											

[15:12] Reserved
[11-0] Image width

Image size height

Register	ISP_BASE_ADDR + 09h								ISP_BASE_ADDR + 08h							
Bit number	15~12				11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				IMG_hh				IMG_hL							
R/W					RW											
Default value					1e0h											

[15:12] Reserved
[11-0] Image height

ISP edge enhance parameters register

Register	ISP_BASE_ADDR + 0bh								ISP_BASE_ADDR + 0ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bit field name	FEM_P								FEM_M							
R/W	RW								RW							
Default value	60h								40h							

[15-8] 16*m*x2/(x2-x1) , recommended 60h
[7-0] 16*m , recommended 40h

ISP edge enhance parameters register 2

Register	ISP_BASE_ADDR + 0dh								ISP_BASE_ADDR + 0ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FEM_X2								FEM_X1							
R/W	RW								RW							
Default value	0fh								05h							

[15-8] x2 , recommended 0fh
[7-0] x1 , recommended 05h

ISP edge enhance parameters register 3

Register	ISP_BASE_ADDR + 0fh								ISP_BASE_ADDR + 0eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FEM_MAX								FEM_MIN							
R/W	RW								RW							
Default value	0Fh								05h							

[15-8] Maximum of edge enhance value, recommended ffh
[7-0] Minimum of edge enhance value, recommended ffh

Lens fall off center register 1

Register	ISP_BASE_ADDR + 11h					ISP_BASE_ADDR + 10h							
Bit number	15~12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved	CEN_Xh				CEN_XL							
R/W		RW											
Default value		0140h											

[15:12] Reserved
[11-0] Lens fall off center X coordinate, recommended 140h

Lens fall off center register 2

Register	ISP_BASE_ADDR + 13h					ISP_BASE_ADDR + 12h							
Bit number	15~12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved	CEN_Yh				CEN_YL							
R/W		RW											
Default value		00f0h											

[15:12] Reserved
[11-0] Lens fall off center Y coordinate, recommended f0h

Lens fall off focus register

Register	ISP_BASE_ADDR + 15h								ISP_BASE_ADDR + 14h							
Bit number	15~13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Bit field name	Reserved	F2_recproc														
R/W		RW														
Default value		43h														

[15:13] Reserved
[12-0] (1/F2)*224

Auto-flicker detection line control register

Register	ISP_BASE_ADDR + 17h			ISP_BASE_ADDR + 16h							
Bit number	15	14~9	8	7	6	5	4	3	2	1	0
Bit field name	AFlnmode	AFlncnt	AFlnstart								
R/W	RW	RW	RW								
Default value	FEh		D0h								

[15] Auto-filcker line mode, recommended 0h

[14-9] Auto-filcker line counter, recommended 3fh

[8-0] Auto-filcker line start, actual start position is AFlnstart [8:0] *8 ,recommended 08h

Auto-flicker detection statistics register

Register	ISP_BASE_ADDR + 1ah	ISP_BASE_ADDR + 19h	ISP_BASE_ADDR + 18h
Bit number	23-16	15-8	7-0
Bit field name	AF statistics		
R/W	R		
Default value			

[23-0] Auto-flicker detection statistics value

Auto-flicker detection line step register

Register	ISP_BASE_ADDR + 1bh							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved		AFlnstep				AF statistics [25:24]	
R/W			RW				R	
Default value			00h				00h	

[7:6] Reserved

[5-2] Auto-flicker detection line step value

[1:0] Auto-flicker detection statistics value [25:24]

Digital RGB gain register

Register	ISP_BASE_ADDR + 24h								ISP_BASE_ADDR + 23h								ISP_BASE_ADDR + 22h							
Bit number	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GGAIN								BGAIN								RGAIN							
R/W	RW								RW								RW							
Default value	40h								40h								40h							

[23-16] Digital G gain value

[15-8] Digital B gain value

[7-0] Digital R gain value

Tail blank control register between the last 5 rows

Register	ISP_BASE_ADDR + 25h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	GLB_GAIN							
R/W	RW							
Default value	02h							

Global gain register

Register	ISP_BASE_ADDR + 27h								ISP_BASE_ADDR + 26h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved		EN	DELAY			STEP		GLB_GAIN							
R/W			RW						RW							
Default value			04hh						40h							

The global gain control is different from the digital gain control. For digital gain control, three channels (R, G and B) have three gain values, which are set by digital gain registers; for global gain control, only one gain value is used for three channels.

[15:14] Reserved

[13] Global gain mode

1 enable

0 disable

[12-10] Global gain effect delay

000 0 frame delay

001 1 frame delay

010 2 frames delay

011 3 frames delay

…

111 7 frames delay

[9-8] Actual global gain effect fraction

00 GLB_GAIN/40h

01 GLB_GAIN/20h

10 GLB_GAIN/10h

11 GLB_GAIN/08h

[7-0] Global gain value

Color matrix registers

COLMAT11 COLMAT12 COLMAT13
COLMAT21 COLMAT22 COLMAT23
COLMAT31 COLMAT32 COLMAT33
R_OFFSET G_OFFSET B_OFFSET

Register	ISP_BASE_ADDR + 2dh								ISP_BASE_ADDR + 2ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	COLMAT12								COLMAT11							
R/W	RW								RW							
Default value	F3h								40h							

register	ISP_BASE_ADDR + 2fh								ISP_BASE_ADDR + 2eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	COLMAT21								COLMAT13							
R/W	RW								RW							
Default value	F3h								F3h							

Register	ISP_BASE_ADDR + 31h								ISP_BASE_ADDR + 30h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	COLMAT23								COLMAT22							
R/W	RW								RW							
Default value	F3h								40h							

Register	ISP_BASE_ADDR + 33h								ISP_BASE_ADDR + 32h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	COLMAT32								COLMAT31							
R/W	RW								RW							
Default value	F3h								F3h							

Register	ISP_BASE_ADDR + 35h								ISP_BASE_ADDR + 34h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	R_OFS								COLMAT33							
R/W	RW								RW							
Default value	00h								40h							

Register	ISP_BASE_ADDR + 37h								ISP_BASE_ADDR + 36h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	B_OFS								G_OFS							
R/W	RW								RW							
Default value	00h								00h							

These registers are used in color correction to compensate for color deviation due to color

filtering and sensing circuits.

Y gamma curve registers

When calculating the statistics for auto function (auto exposure, auto white balance and auto flicker detection), the Y channel of the color plane should be corrected by one curve. These registers define the curve for Y channel, approximated by eight lines. The first nine registers (Y_START0 to Y_START8) define the start and end point for the eight lines and the rest (Y_SLOPE0 to Y_SLOPE7) define the slope of these lines.

Register	ISP_BASE_ADDR + 39h								ISP_BASE_ADDR + 38h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Y_START1								Y_START0							
R/W	RW								RW							
Default value	15h								00h							

Register	ISP_BASE_ADDR + 3bh								ISP_BASE_ADDR + 3ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Y_START3								Y_START2							
R/W	RW								RW							
Default value	30h								20h							

Register	ISP_BASE_ADDR + 3dh								ISP_BASE_ADDR + 3ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Y_START5								Y_START4							
R/W	RW								RW							
Default value	5eh								49h							

Register	ISP_BASE_ADDR + 3fh								ISP_BASE_ADDR + 3eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Y_START7								Y_START6							
R/W	RW								RW							
Default value	a9h								6fh							

Register	ISP_BASE_ADDR + 41h								ISP_BASE_ADDR + 40h							
Bit number	Reserved	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Y_SLOPE0								Y_START8							
R/W	RW								RW							
Default value	30h								D7h							

Register	ISP_BASE_ADDR + 43h								ISP_BASE_ADDR + 42h							
Bit number	Reserved	14	13	12	11	10	9	8	Reserved	6	5	4	3	2	1	0
Bit field name	Y_SLOPE2								Y_SLOPE1							
R/W	RW								RW							

Default value	22h	2bh

Register	ISP_BASE_ADDR + 45h								ISP_BASE_ADDR + 44h							
Bit number	Reserved	14	13	12	11	10	9	8	Reserved-	6	5	4	3	2	1	0
Bit field name	Y_SLOPE4								Y_SLOPE3							
R/W	RW								RW							
Default value	14h								1ah							

Register	ISP_BASE_ADDR + 47h								ISP_BASE_ADDR + 46h							
Bit number	Reserved	14	13	12	11	10	9	8	Reserved-	6	5	4	3	2	1	0
Bit field name	Y_SLOPE6								Y_SLOPE5							
R/W	RW								RW							
Default value	0eh								11h							

Register	ISP_BASE_ADDR + 49h								ISP_BASE_ADDR + 48h							
Bit number	Reserved	14	13	12	11	10	9	8	Reserved-	6	5	4	3	2	1	0
Bit field name	Y_SLOPE8								Y_SLOPE7							
R/W	RW								RW							
Default value	0ah								0bh							

R G B gamma correction registers

Before converted to YCbCr color space for JPEG compression, the RGB color data should be corrected by Gamma curve. These registers define the Gamma curve for R channel. The gamma curve is approximated by 16 segments and these registers are the start and end point for these segments. The segments have the same length – 16, so the slope of one segment can be calculated by shifting the difference between the start point and end point. RGMAS0 to RGMAS16 define R gamma curve. GGMAS0 to GGMAS16 define R gamma curve. BGMAS0 to BGMAS16 define B gamma curve.

Register	ISP_BASE_ADDR + 4bh								ISP_BASE_ADDR + 4ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	RGMAS1								RGMAS0							
R/W	RW								RW							
Default value	04h								00h							

Register	ISP_BASE_ADDR + 4dh								ISP_BASE_ADDR + 4ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	RGMAS3								RGMAS2							
R/W	RW								RW							
Default value	30h								16h							

Register	ISP_BASE_ADDR + 4fh								ISP_BASE_ADDR + 4eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	RGMAS5								RGMAS4							
R/W	RW								RW							
Default value	68h								4Eh							

Register	ISP_BASE_ADDR + 51h								ISP_BASE_ADDR + 50h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	RGMAS7								RGMAS6							
R/W	RW								RW							
Default value	98h								81h							

Register	ISP_BASE_ADDR + 53h								ISP_BASE_ADDR + 52h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	RGMAS9								RGMAS8							
R/W	RW								RW							
Default value	BE								AC							

Register	ISP_BASE_ADDR + 55h								ISP_BASE_ADDR + 54h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	RGMAS11								RGMAS10							
R/W	RW								RW							
Default value	DAh								CDh							

Register	ISP_BASE_ADDR + 57h								ISP_BASE_ADDR + 56h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	RGMAS13								RGMAS12							
R/W	RW								RW							
Default value	EDh								E4h							

Register	ISP_BASE_ADDR + 59h								ISP_BASE_ADDR + 58h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	RGMAS15								RGMAS14							
R/W	RW								RW							
Default value	FBh								F5h							

Register	ISP_BASE_ADDR + 5bh								ISP_BASE_ADDR + 5ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GGMAS0								RGMAS16							
R/W	RW								RW							
Default value	00h								FFh							

Register	ISP_BASE_ADDR + 5dh								ISP_BASE_ADDR + 5ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GGMAS2								GGMAS1							
R/W	RW								RW							
Default value	16h								04h							

Register	ISP_BASE_ADDR + 5fh								ISP_BASE_ADDR + 5eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GGMAS4								GGMAS3							
R/W	RW								RW							
Default value	4Eh								30h							

Register	ISP_BASE_ADDR + 61h								ISP_BASE_ADDR + 60h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	GGMAS6								GGMAS5							
R/W	RW								RW							
Default value	81h								68h							

Register	ISP_BASE_ADDR + 63h								ISP_BASE_ADDR + 62h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GGMAS8								GGMAS7							
R/W	RW								RW							
Default value	ACh								98h							

Register	ISP_BASE_ADDR + 65h								ISP_BASE_ADDR + 64h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GGMAS10								GGMAS9							
R/W	RW								RW							
Default value	CDh								BEh							

Register	ISP_BASE_ADDR + 67h								ISP_BASE_ADDR + 66h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GGMAS12								GGMAS11							
R/W	RW								RW							
Default value	E4h								DAh							

Register	ISP_BASE_ADDR + 69h								ISP_BASE_ADDR + 68h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	GGMAS14								GGMAS13							
R/W	RW								RW							
Default value	F5hh								EDh							

Register	ISP_BASE_ADDR + 6bh	ISP_BASE_ADDR + 6ah

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GGMAS16								GGMAS15							
R/W	RW								RW							
Default value	FFh								FBh							

Register	ISP_BASE_ADDR + 6dh								ISP_BASE_ADDR + 6ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BGMAS1								BGMAS0							
R/W	RW								RW							
Default value	04h								00h							

Register	ISP_BASE_ADDR + 6fh								ISP_BASE_ADDR + 6eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BGMAS3								BGMAS2							
R/W	RW								RW							
Default value	30h								16h							

Register	ISP_BASE_ADDR + 71h								ISP_BASE_ADDR + 70h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BGMAS5								BGMAS4							
R/W	RW								RW							
Default value	68h								4Eh							

Register	ISP_BASE_ADDR + 73h								ISP_BASE_ADDR + 72h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BGMAS7								BGMAS6							
R/W	RW								RW							
Default value	98h								81h							

Register	ISP_BASE_ADDR + 75h								ISP_BASE_ADDR + 74h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BGMAS9								BGMAS8							
R/W	RW								RW							
Default value	BEh								ACh							

Register	ISP_BASE_ADDR + 77h								ISP_BASE_ADDR + 76h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BGMAS11								BGMAS10							
R/W	RW								RW							
Default value	DAh								CDh							

Register	ISP_BASE_ADDR + 79h								ISP_BASE_ADDR + 78h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit field name	BGMAS13	BGMAS12
R/W	RW	RW
Default value	EDh	E4h

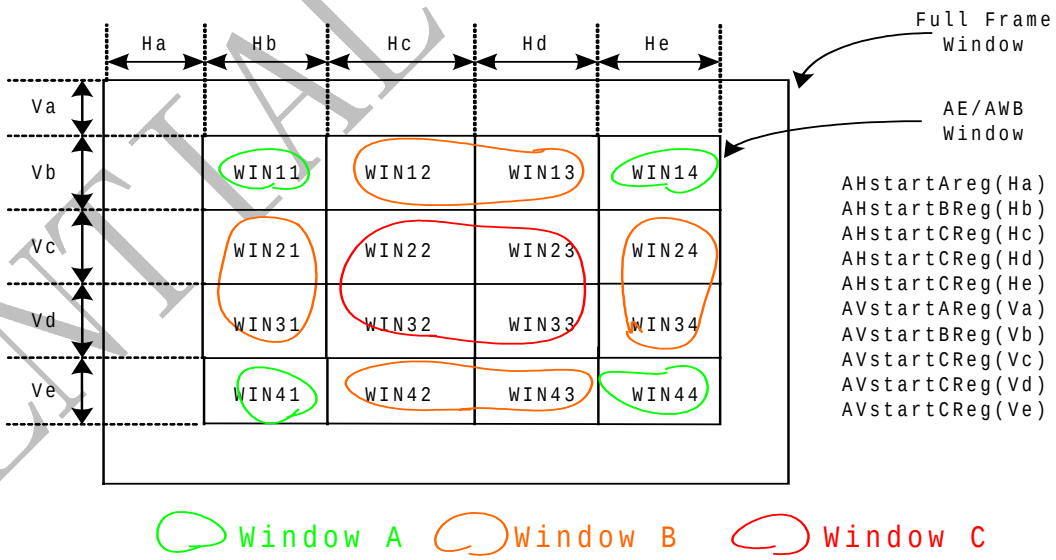
Register	ISP_BASE_ADDR + 7bh								ISP_BASE_ADDR + 7ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BGMAS15								BGMAS14							
R/W	RW								RW							
Default value	FBh								F5h							

Register	ISP_BASE_ADDR + 7dh								ISP_BASE_ADDR + 7ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserved								BGMAS16							
R/W									RW							
Default value									FFh							

Auto-function control register

Register	ISP_BASE_ADDR + 81h								ISP_BASE_ADDR + 80h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserved					Awbw	Awbr	Awben	AeWmode		Resv.	AeWw	AutoIntFrq			AntiEn
R/W			RW						RW							
Default value			0h						0h							

AE window shown below is used for AWB fully or partially also.



[15:11] reserved

[10] AWB window control

Selecting AWB windows in a frame used for calculating the average value

0 16 windows used all

1 center window(window C) used only

[9] AWB reverse mode

1 reverse mode

0 normal mode

[8] AWB function on/off

1 AWB function enable

0 AWB function disable

[7:6] AE weight mode

Selecting weight mode of AE Function. When block weight mode is selected, 16 AE windows have different weight. When window weight mode is selected, nine AE windows have same weight and all the pixels in the center window (window-c) and window-b will be used for AE but pixels in the window-a are limited by bit 4 Wweight. When block and window mixed mode, AE windows have different weights and pixel limit function is on too by

bit 4 Wweight. When Weightless mode is selected, all the original pixel value in the nine AE windows will be used for AE.

00 Bweight (Block weight) Mode only
01 Wweight (Window weight) Mode only
10 Bweight + Wweight
11 Weightless Mode

windows mode means that the pixel in windows C is filtered
block mode means that weight is adopted for caculation

[4] AE W Weight mode
Windows Weight Selector for AE when Wweight Mode Valid.
0 Larger Y range is used.
1 Smaller Y range is used

[3:1] auto function interrupt frequency which is adopted for anti-flicker int and auto_focus int

000 interrupt per 1 frame
001 interrupt per 3 frame
010 interrupt per 5 frame
011 interrupt per 9 frame
100 interrupt per 17 frame
101 interrupt per 33 frame
110 interrupt per 65 frame
111 interrupt per 129 frame

[0] Anti-flicker function on/off
1 enable
0 disable

Y limit of AWB register

Register	ISP_BASE_ADDR + 8bh								ISP_BASE_ADDR + 8ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	YTOP								YBOT							
R/W	RW								RW							
Default value	D0h								00h							

[15-8] Y top value
[7-0] Y bottom value

IQ limit of AWB register

Register	ISP_BASE_ADDR + 8dh								ISP_BASE_ADDR + 8ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	ITOP								QTOP							
R/W	RW								RW							
Default value	18h								10h							

[15-8] I top value
[7-0] Q top value

RB gain limit of AWB register

Register	ISP_BASE_ADDR + 8fh								ISP_BASE_ADDR + 8eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GTOP								GLOW							
R/W	RW								RW							
Default value	3Fh								14h							

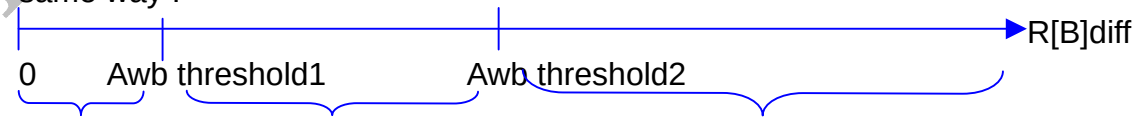
[15-8] RB gain value maximum
[7-0] RB gain value minmum

AWB range control register

Register	ISP_BASE_ADDR + 91h								ISP_BASE_ADDR + 90h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	AwbThd2						AwbThd1				AwbStep					
R/W	RW						RW				RW					
Default value	0fh						03h				04h					

[15-10] Awb threshold2
[9-6] Awb threshold1
[5-0] Awb adjust in larger step
Rdiff = | Rmean – Gmean |
Bdiff = | Bmean – Gmean |

The special R gain adjustment step length is determined by the relationship among Rdiff and Awb threshold1 and Abw threshold2 as below , B gain adjustment step length is in the same way .



R[B]step = 0 R[B]step = 1 R[B]step = AwbStep

Awb means register

Register	ISP_BASE_ADDR + 94h								ISP_BASE_ADDR + 93h								ISP_BASE_ADDR + 92h							
Bit number	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BMEAN								GMEAN								RMEAN							
R/W	R								R								R							
Default value	00h								00h								00h							

[23-16] B mean value
[15-8] G mean value
[7-0] R mean value

Y mean register

Register	ISP_BASE_ADDR + 95h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	YMEAN							
R/W	R							
Default value	00h							

Auto-function status register

Register	ISP_BASE_ADDR + 96h	
Bit number	7~4	3~0
Bit field name	reserved	AwbRBCon
R/W	R	
Default value		

[7:4] reserved

[3] 1 awb valid

[2] 1 B mean > G mean

0 B mean < G mean

[1] 1 R mean > G mean

0 R mean < G mean

[0] 1 RG gain reach limitation both.

0 NOT both RG gain reach limitation

DPD threshold register

Register	ISP_BASE_ADDR + a0h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	DPCThD							
R/W	RW							
Default value	20h							

Noise table register

Register	ISP_BASE_ADDR + a1h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	NT0							
R/W	RW							
Default value	20h							

Register	ISP_BASE_ADDR + a3h								ISP_BASE_ADDR + a2h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT2								NT1							
R/W	RW								RW							
Default value	30h								20h							

Register	ISP_BASE_ADDR + a5h								ISP_BASE_ADDR + a4h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT4								NT3							
R/W	RW								RW							
Default value	40h								38h							

Register	ISP_BASE_ADDR + a7h								ISP_BASE_ADDR + a6h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT6								NT5							
R/W	RW								RW							
Default value	60h								50h							

Register	ISP_BASE_ADDR + a9h								ISP_BASE_ADDR + a8h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT8								NT7							
R/W	RW								RW							
Default value	80h								70h							

Register	ISP_BASE_ADDR + abh								ISP_BASE_ADDR + aah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT10								NT9							
R/W	RW								RW							
Default value	A0h								90h							

Register	ISP_BASE_ADDR +ad9h								ISP_BASE_ADDR + ach							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT12								NT11							
R/W	RW								RW							
Default value	C0h								B0h							

Register	ISP_BASE_ADDR + afh								ISP_BASE_ADDR + aeh							
Bit mi,ber	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT14								NT13							
R/W	RW								RW							
Default value	E0h								d0h							

Register	ISP_BASE_ADDR + b1h								ISP_BASE_ADDR + b0h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT16								NT15							
R/W	RW								RW							
Default value	Ffh								F0h							

DPD_UV threshold register

Register	ISP_BASE_ADDR + b2h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	DPCThD_UV							
R/W	RW							
Default value	20h							

Noise table register

Register	ISP_BASE_ADDR + b3h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	NT0_UV							
R/W	RW							
Default value	20h							

Register	ISP_BASE_ADDR + b5h								ISP_BASE_ADDR + b4h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT2_UV								NT1_UV							
R/W	RW								RW							
Default value	30h								20h							

Register	ISP_BASE_ADDR + b7h								ISP_BASE_ADDR + b6h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT4_UV								NT3_UV							
R/W	RW								RW							
Default value	40h								38h							

Register	ISP_BASE_ADDR + b9h	ISP_BASE_ADDR + b8h

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT6_UV								NT5_UV							
R/W	RW								RW							
Default value	60h								50h							

Register	ISP_BASE_ADDR + bbh								ISP_BASE_ADDR + bah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT8_UV								NT7_UV							
R/W	RW								RW							
Default value	80h								70h							

Register	ISP_BASE_ADDR + bdh								ISP_BASE_ADDR + bch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT10_UV								NT9_UV							
R/W	RW								RW							
Default value	A0h								90h							

Register	ISP_BASE_ADDR +bfh								ISP_BASE_ADDR + beh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT12_UV								NT11_UV							
R/W	RW								RW							
Default value	C0h								B0h							

Register	ISP_BASE_ADDR + c1h								ISP_BASE_ADDR + c0h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT14_UV								NT13_UV							
R/W	RW								RW							
Default value	E0h								d0h							

Register	ISP_BASE_ADDR + c3h								ISP_BASE_ADDR + c2h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	NT16_UV								NT15_YV							
R/W	RW								RW							
Default value	Ffh								F0h							

YUV Interface Sensor Enable Register AND HIS_CTRL Register

Register	ISP_BASE_ADDR + c5h								ISP_BASE_ADDR + c4h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						histogram_effect	histogram_en	Reserved							YUV_SIF_EN
R/W	RW						RW		RW							RW
Default value	00h								00h							0h

[15:10] reserved

[9] histogram_effect .

1 when this bit is 1,the histogram compensation takes effect on the Y channel of YUV format data stream.

0 when this bit is 0 ,the histogram compensation does not take effect on the Y channel of YUV format data stream

[8] histogram_en.

1 When this bit is 1,histogram statistic result is available.

0 When this bit is 0,histogram statistic result is not available.

[7:1] reserved

[0] YUV_SIF_EN

1 When YUV_SIF_EN is set to 1,the YUV format data is input to the isp module .

0 When YUV_SIF_EN is set to 0,the YUV format data is input to the isp module .

histogram Gain That Is Set Manually

Register	ISP_BASE_ADDR + c6h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	histogram_gain_manual[7:0]							
R/W	RW							
Default value	80h							

The number which is 1% of the number of all pixels (Q_histogram)
the X value of current fame (X_CUR)

Register	ISP_BASE_ADDR + c9h								ISP_BASE_ADDR + c8h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Q_histogram[15:8]								Q_histogram[7:0]							
R/W	RW								RW							
Default value	00h								00h							

Register	ISP_BASE_ADDR + cbh							ISP_BASE_ADDR + CAh							
Bit number	15	14	13	12	11	10	9	8	6	5	4	3	2	1	0
Bit field name	X_CUR							Reserved	Q_histogram[21:16]						
R/W	R							RW							
Default value								0ch							

Q_histogram is the number which is 1% or so of the pixel number of whole frame.
Q_histogram is set by host .
X_CUR : this is the histogram compensation point of last frame data.

The X value of last three frames

Register	ISP_BASE_ADDR +cch							
Bit number	7	6	5	4	3	2	1	0
Bit field name	X_AVG							
R/W	R							
Default value								

X_AVG :this is the weighted average value of last three frame data. This value will take effect if histogram_effect bit is enabled.

Black level offset registers

Register	ISP_BASE_ADDR + d1h							ISP_BASE_ADDR + d0h							
Bit number	15	14	13	12	11	10	9	8	6	5	4	3	2	1	0
Bit field name	BL_R_OFFSET							BL_B_OFFSET							
R/W	RW							RW							
Default value	00h							00h							

Register	ISP_BASE_ADDR + d3h							ISP_BASE_ADDR + d2h							
Bit number	15	14	13	12	11	10	9	8	6	5	4	3	2	1	0
Bit field name	BL_G2_OFFSET							BL_G1_OFFSET							
R/W	RW							RW							
Default value	00h							00h							

auto focus enable register (AUTO_FOCUS_EN)

Register	ISP_BASE_ADDR + d4h	
Bit number	7:1	0
Bit field name	Reserved	AUTO_FOCUS_EN
R/W	RW	
Default value	01h	

Auto focus statistic results

Register	ISP_BASE_ADDR +d7h								ISP_BASE_ADDR + d6h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	EDGE_SUM[15:8]								EDGE_SUM[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + d9h								ISP_BASE_ADDR + d8h							
Bit number	reserved	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	EDGE_SUM[30:24]								EDGE_SUM[23:16]							
R/W	R								R							
Default value																

Edge_sum is used to do auto focus, adjust the lens to the position where you can get the biggest edge_sum of a frame.

If a picture is in focus, the sum of it's edge-map data is biggest (compare with the sum of the same picture if it is out of focus).

The weight value for 16 AE windows

Register	ISP_BASE_ADDR + dbh								ISP_BASE_ADDR + dah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN14_WT				WIN13_WT				WIN12_WT				WIN11_WT			
R/W	RW								RW							
Default value	13								31							

Register	ISP_BASE_ADDR + ddh								ISP_BASE_ADDR + dch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN24_WT				WIN23_WT				WIN22_WT				WIN21_WT			
R/W	RW								RW							
Default value	24								42							

Register	ISP_BASE_ADDR + dfh								ISP_BASE_ADDR + deh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN34_WT				WIN33_WT				WIN32_WT				WIN31_WT			
R/W	RW								RW							
Default value	24								42							

Register	ISP_BASE_ADDR + e1h								ISP_BASE_ADDR + e0h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN44_WT				WIN43_WT				WIN42_WT				WIN41_WT			
R/W	RW								RW							
Default value	13								31							

Y_MEAN values for 16 windows

Register	ISP_BASE_ADDR + e3h								ISP_BASE_ADDR + e2h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN12_YMEAN								WIN11_YMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + e5h								ISP_BASE_ADDR +e4h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN14_YMEAN								WIN13_YMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + e7h								ISP_BASE_ADDR + e6h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN22_YMEAN								WIN21_YMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + e9h								ISP_BASE_ADDR + e8h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN24_YMEAN								WIN23_YMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + ebh								ISP_BASE_ADDR + eah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN32_YMEAN								WIN31_YMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + edh								ISP_BASE_ADDR + ech							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN34_YMEAN								WIN33_YMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + efh								ISP_BASE_ADDR + eeh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN42_YMEAN								WIN41_YMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + f1h								ISP_BASE_ADDR + f0h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN44_YMEAN								WIN43_YMEAN							
R/W	R								R							
Default value	NA								NA							

I mean values for 16 windows

Register	ISP_BASE_ADDR + f3h								ISP_BASE_ADDR + f2h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN12_IMEAN								WIN11_IMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + f5h								ISP_BASE_ADDR +f4h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN14_IMEAN								WIN13_IMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + f7h								ISP_BASE_ADDR + f6h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN22_IMEAN								WIN21_IMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + f9h								ISP_BASE_ADDR + f8h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN24_IMEAN								WIN23_IMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + fbh								ISP_BASE_ADDR + fah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN32_IMEAN								WIN31_IMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + fdh								ISP_BASE_ADDR + fch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN34_IMEAN								WIN33_IMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + ffh								ISP_BASE_ADDR + feh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN42_IMEAN								WIN41_IMEAN							
R/W	R								R							

Default value	NA	NA

Register	ISP_BASE_ADDR + 101h								ISP_BASE_ADDR + 100h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN44_IMEAN								WIN43_IMEAN							
R/W	R								R							
Default value	NA								NA							

Q mean values for 16 windows

Register	ISP_BASE_ADDR + 103h								ISP_BASE_ADDR + 102h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN12_QMEAN								WIN11_QMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + 105h								ISP_BASE_ADDR +104h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN14_QMEAN								WIN13_QMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + 107h								ISP_BASE_ADDR + 106h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN22_QMEAN								WIN21_QMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + 109h								ISP_BASE_ADDR + 108h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN24_QMEAN								WIN23_QMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + 10bh								ISP_BASE_ADDR + 10ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN32_QMEAN								WIN31_QMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + 10dh	ISP_BASE_ADDR + 10ch

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN34_QMEAN								WIN33_QMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + 10fh								ISP_BASE_ADDR + 10eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN42_QMEAN								WIN41_QMEAN							
R/W	R								R							
Default value	NA								NA							

Register	ISP_BASE_ADDR + 111h								ISP_BASE_ADDR + 110h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WIN44_QMEAN								WIN43_QMEAN							
R/W	R								R							
Default value	NA								NA							

Windows position definition registers

Register	ISP_BASE_ADDR + 113h								ISP_BASE_ADDR + 112h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	AhStartBReg								AhStartAReg							
R/W	RW								RW							
Default value	OFh								05h							

Register	ISP_BASE_ADDR + 115h								ISP_BASE_ADDR + 114h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	AhStartDReg								AhStartCReg							
R/W	RW								RW							
Default value	28h								28h							

Register	ISP_BASE_ADDR + 117h								ISP_BASE_ADDR +116h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								AhStartEReg							
R/W									RW							
Default value									28h							

Register	ISP_BASE_ADDR + 119h								ISP_BASE_ADDR + 118h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	AVStartBReg								AVStartAReg							
R/W	RW								RW							

Default value	0Ah	05h

Register	ISP_BASE_ADDR + 11bh								ISP_BASE_ADDR + 11ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	AVStartDReg								AVStartCReg							
R/W	RW								RW							
Default value	1Eh								1Eh							

Register	ISP_BASE_ADDR + 11dh								ISP_BASE_ADDR + 11ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								AVStartEReg							
R/W									RW							
Default value									1Eh							

16 times of ha hb hc hd he are used as the actual values of window configuration
8 times of Va Vb Vc Vd Ve are used as the actual values of window configration

histogram statistic results

Register	ISP_BASE_ADDR + 121h								ISP_BASE_ADDR + 120h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	histogram_sum_result0[15:8]								histogram_sum_result0[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 123h								ISP_BASE_ADDR + 122							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	histogram_sum_result1[7:0]								reserved	histogram_sum_result0[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 125h								ISP_BASE_ADDR + 124h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	reserved	histogram_sum_result1[21:16]							histogram_sum_result1[15:8]							
R/W		R							R							
Default value																

Register	ISP_BASE_ADDR + 127h								ISP_BASE_ADDR + 126h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result2[15:8]								histogram_sum_result2[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 129h								ISP_BASE_ADDR + 128							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result3[7:0]								reserved	histogram_sum_result2[21:16]						

R/W	R		R
Default value			

Register	ISP_BASE_ADDR + 12bh								ISP_BASE_ADDR + 12ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_result3[21:16]						histogram_sum_result3[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 12dh								ISP_BASE_ADDR + 12ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result4[15:8]								histogram_sum_result4[7:0]							
R/W	R								R							
Default value																
Register	ISP_BASE_ADDR + 12fh								ISP_BASE_ADDR + 12eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result5[7:0]								reserved	histogram_sum_result4[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 131h								ISP_BASE_ADDR + 130h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_result5[21:16]						histogram_sum_result5[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 133h								ISP_BASE_ADDR + 132h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result6[15:8]								histogram_sum_result6[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 135h								ISP_BASE_ADDR + 134							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result7[7:0]								reservd	histogram_sum_result6[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 137h								ISP_BASE_ADDR + 136h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_result7[21:16]						histogram_sum_result7[15:8]							

R/W		R	R
Default value			

Register	ISP_BASE_ADDR + 139h								ISP_BASE_ADDR + 138h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result8[15:8]								histogram_sum_result8[7:0]							
R/W	R								R							
Default value																
Register	ISP_BASE_ADDR + 13bh								ISP_BASE_ADDR + 13ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result9[7:0]								reserved	histogram_sum_result8[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 13dh								ISP_BASE_ADDR + 13ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reservd		histogram_sum_result19[21:16]						histogram_sum_result19[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 13fh								ISP_BASE_ADDR + 13eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result10[15:8]								histogram_sum_result10[7:0]							
R/W	R								R							
Default value																
Register	ISP_BASE_ADDR + 141h								ISP_BASE_ADDR + 140							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_result11[7:0]								reserved	histogram_sum_result10[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 143h								ISP_BASE_ADDR + 142h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_result11[21:16]						histogram_sum_result11[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 145h								ISP_BASE_ADDR + 144h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultc0[15:8]								histogram_sum_resultc0[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 147h								ISP_BASE_ADDR + 146							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultc4[7:0]								reserved	histogram_sum_resultc0[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 149h								ISP_BASE_ADDR + 148h							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reservd		histogram_sum_resultc4[21:16]						histogram_sum_resultc4[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 14bh								ISP_BASE_ADDR + 14ah							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultc8[15:8]								histogram_sum_resultc8[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 14d								ISP_BASE_ADDR + 14c							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultcc[7:0]								reserved	histogram_sum_resultc8[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 14fh								ISP_BASE_ADDR + 14eh							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_resultcc[21:16]						histogram_sum_resultcc[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 151h								ISP_BASE_ADDR + 150h							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultd0[15:8]								histogram_sum_resultd0[7:0]							
R/W	R								R							
Default value																
Register	ISP_BASE_ADDR + 153h								ISP_BASE_ADDR + 152							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultd4[7:0]								reserved	histogram_sum_resultd0[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 155h								ISP_BASE_ADDR + 154h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_resultd4[21:16]						histogram_sum_resultd4[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 157h								ISP_BASE_ADDR + 156h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultd8[15:8]								histogram_sum_resulted8[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 159h								ISP_BASE_ADDR + 158							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultdc[7:0]								reserved		histogram_sum_resultd8[21:16]					
R/W	R										R					
Default value																

Register	ISP_BASE_ADDR + 15bh								ISP_BASE_ADDR + 15ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_resultdc[21:16]						histogram_sum_resultdc[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 15dh								ISP_BASE_ADDR + 15ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resulte0[15:8]								histogram_sum_resulte0[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 15fh								ISP_BASE_ADDR + 15e							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resulte4[7:0]								reserved		histogram_sum_resulte0[21:16]					
R/W	R										R					
Default value																

Register	ISP_BASE_ADDR + 161h								ISP_BASE_ADDR + 160h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_resulte4[21:16]						histogram_sum_resulte4[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 163h								ISP_BASE_ADDR + 162h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resulte8[15:8]								histogram_sum_resulte8[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 165h								ISP_BASE_ADDR + 164							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultec[7:0]								reserved	histogram_sum_resulte8[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 167h								ISP_BASE_ADDR + 166h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_resultec[21:16]						histogram_sum_resultec[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 169h								ISP_BASE_ADDR + 168h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultf0[15:8]								histogram_sum_resultf0[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 16bh								ISP_BASE_ADDR + 16a							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultf4[7:0]								reserved	histogram_sum_resultf0[21:16]						
R/W	R									R						
Default value																

Register	ISP_BASE_ADDR + 16dh								ISP_BASE_ADDR + 16ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_resultf4[21:16]						histogram_sum_resultf4[15:8]							
R/W			R						R							
Default value																

Register	ISP_BASE_ADDR + 16fh								ISP_BASE_ADDR + 16eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultf8[15:8]								histogram_sum_resultf8[7:0]							
R/W	R								R							
Default value																

Register	ISP_BASE_ADDR + 171h								ISP_BASE_ADDR + 170							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	histogram_sum_resultfc[7:0]								reserved		histogram_sum_resultf8[21:16]					
R/W	R										R					
Default value																

Register	ISP_BASE_ADDR + 173h								ISP_BASE_ADDR + 172h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit fieldname	reserved		histogram_sum_resultfc[21:16]						histogram_sum_resultfc[15:8]							
R/W			R						R							
Default value																

8.4 IPP register

Special effect control

Register	IPP_BASE_ADDR+00h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	default		IPP_MODE		SPESEL			SPEEN
R/W	RW							
De fault value	00h							

The register is used to select one of five special effects for capture and display.

[5:4] IPP_MODE: IPP working mode
2'b00: viewfinder mode
2'b01: capture mode
2'b10: decoder mode
2'b11: display mode

[3:0] SPESEL: select between the following 5 special effects
3'b000: sephia
3'b001: special color
3'b010: negative
3'b011: sketch
3'b100: relief

[0] SPEEN: special effect enable bit
1: enable
0: disable

To select one kind of special effect, SPPEN must be set to 1'b1 and SPESEL is used to select the effect type. If SPPEN is set to 1'b0, the input image data is sent to next submodule directly without any processing.

Sizer Control

Register	IPP_BASE_ADDR+01h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	FRAME	DATASEL		IPP_RESET	FIREN	DPSIEN	CPSIEN	TMSIEN
R/W	RW							
De fault value	4Ch							

[7] FRAME: frame enable bit
1’b0: capture or decoder without frame
1’b1: capture or decoder with frame

[6:5] DATASEL: input image data select
2’b00: select LBUF data
2’b01: select SIF data
2’b10: select ISP data
2’b11: reserved

[4] IPP_RESET: software reset signal of IPP module
1’b1: reset all the internal registers except control registers when written to 1’b1
1’b0: dose not reset internal registers

[3] FIREN: prefilter enable bit
1’b1: prefilter enable
1’b0: prefilter disable

[2] DPSIEN: display sizer enable bit
1’b1: display sizer enable
1’b0: display sizer disable

[1] CPSIEN: capture sizer enable bit
1’b1: capture sizer enable
1’b0: capture sizer disable

[0] TMSIEN: thumbnail sizer enable bit
1’b1: thumbnail sizer enable
1’b0: thumbnail sizer disable

Image width

Register	IPP_BASE_ADDR+03h								IPP_BASE_ADDR+02h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				IMAGE_WIDTH											
R/W	NA				RW											
De fault value	02h								80h							

This register defines the width of the input image.

[15:12] Reserved

[11:0] IMAGE_WIDTH: width of the input image

Image height

Register	IPP_BASE_ADDR+05h								IPP_BASE_ADDR+04							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				IMAGE_WIDTH											
R/W	NA				RW											
De fault value	01h								E0h							

This register defines the height of the input image.

[15:12] Reserved

[11:0] IMAGE_HEIGHT: height of the input image

The above two registers (IMAGE_WIDTH and IMAGE_HEIGHT) define the size of the input image from SIF, ISP or LBUF module.

Window width

Register	IPP_BASE_ADDR+07h								IPP_BASE_ADDR+06h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				WINDOW_WIDTH											
R/W	NA				RW											
De fault value	02h								80h							

This register defines the width of image window.

[15:12] Reserved

[11:0] WINDOW_WIDTH: width of the image window

Window height

Register	IPP_BASE_ADDR+09h								IPP_BASE_ADDR+08h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				WINDOW_WIDTH											
R/W	NA				RW											
De fault value	01h								E0h							

This register defines the height of image window.

[15:12] Reserved

[11:0] WINDOW_HEIGHT: height of the image window

The above two registers (WINDOW_WIDTH and WINDOW_HEIGHT) define the size of the image window truncating from input image.

Horizontal coordinate of window start pixel

Register	IPP_BASE_ADDR+0Bh								IPP_BASE_ADDR+0Ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				WIDTH_START_X											
R/W	NA				RW											
De fault value	00h								00h							

This register defines the horizontal coordinate of window start pixel.

[15:12] Reserved

[11:0] WINDOW_START_X: horizontal coordinate of window start pixel

Vetical coordinate of window start pixel

Register	IPP_BASE_ADDR+0Dh								IPP_BASE_ADDR+0Ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				WIDTH_START_Y											
R/W	NA				RW											
De fault value	00h								00h							

This register defines the vertical coordinate of window start pixel.

[15:12] Reserved

[11:0] WINDOW_HEIGHT: height of the image window

The above two registers (WINDOW_START_X and WINDOW_START_Y) define the coordinate of the start pixel of the image window truncating from input image.

For the relationship of input image and image window, please refer to figure 5:

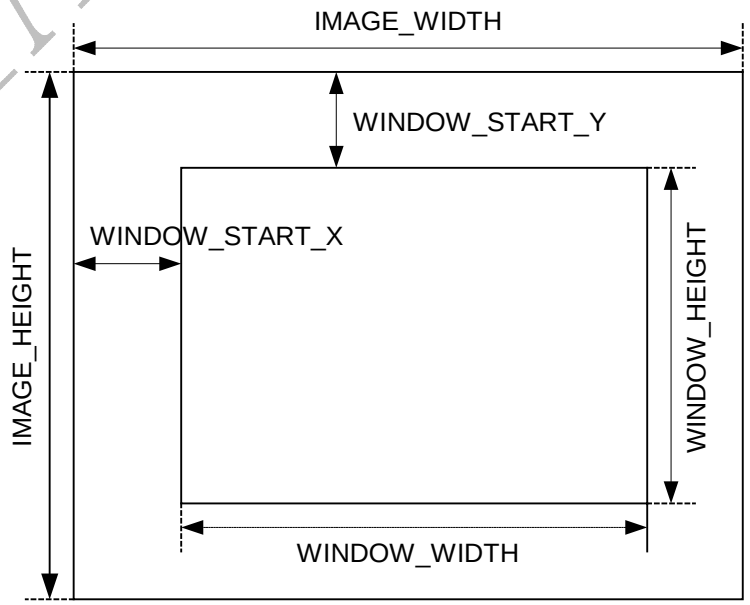


Figure 5: Image window related registers

Generally, the above six registers have the following relationship:

WINDOW_WIDTH <= IMAGE_WIDTH

WINDOW_HEIGHT <= IMAGE_HEIGHT

WINDOW_START_X <= IMAGE_WIDTH – WINDOW_WIDTH

WINDOW_START_Y <= IMAGE_HEIGHT – WINDOW_HEIGHT

But usually the window is in the center of the input image, so we have:

WINDOW_START_X = (IMAGE_WIDTH – WINDOW_WIDTH)/2

WINDOW_START_Y = (IMAGE_HEIGHT – WINDOW_HEIGHT)/2

Output image width of display sizer

Register	IPP_BASE_ADDR+0Fh								IPP_BASE_ADDR+0Eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								DISPLAY_TARGET_WIDTH							
R/W	NA								RW							
De fault value	00h								80h							

This register defines the width of output image from the display sizer.

[15:10] Reserved

[9:0] DISPLAY_TARGET_WIDTH: width of the output image from the display sizer

Output image height of display sizer

Register	IPP_BASE_ADDR+11h								IPP_BASE_ADDR+10h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								DISPLAY_TARGET_HEIGHT							
R/W	NA								RW							
De fault value	00h								60h							

This register defines the height of output image from the display sizer.

[15:10] Reserved

[9:0] DISPLAY_TARGET_HEIGHT: height of the output image from the display sizer

The above two registers (DISPLAY_TARGET_WIDTH and DISPLAY_TARGET_HEIGHT) define the size of the image which sends to LCDC module for updating panel.

Horizontal ratio of display sizer

Register	IPP_BASE_ADDR+013h								IPP_BASE_ADDR+12h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	HRDS_INTEGER						HRDS_FRACTION									
R/W	RW						RW									
De fault value	14h								00h							

This register defines the horizontal sizer ratio for display sizer.

[15:10] HRDS_INTEGER: integer part of the display horizontal ratio

[9:0] HRDS_FRACTION: fraction part of the display horizontal ratio

Vertical ratio of display sizer

Register	IPP_BASE_ADDR+15h								IPP_BASE_ADDR+14h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VRDS_INTEGER						VRDS_FRACTION									
R/W	RW						RW									
De fault value	14h								00h							

This register defines the vertical sizer ratio for display sizer.

[15:10] VRDS_INTEGER: integer part of the display horizontal ratio

[9:0] VRDS_FRACTION: fraction part of the display horizontal ratio

Let's take an example to show how to calculate the sizer ratio. If the size of the image window is VGA (640x480), and the output image size from the display sizer is 128x96, then the downscaling ratio is 5. For DISPLAY_HORIZONTAL_RATIO,

HRDS_INTEGER = 000101b

HRDS_FRACTION = 0000000000b

And

DISPLAY_HORIZONTAL_RATIO = 000101_0000000000b = 1400h

For above image downscaling, DISPLAY_HORIZONTAL_RATIO is equal to DISPLAY_VERTICAL_RATIO. Note that although there are two registers for horizontal resizing and vertical resizing ratio, but generally they are set to the same value so that the image won't change its shape.

Another example shows the case when the ratio has its fraction part, e.g., 608x456 to 160X120 and the sizer ratio is 3.8. For this case, HRDS_INTEGER= 000011b. Because HRDS_FRACTION is a 10-bit binary, so we can calculate it as following:

HRDS_FRACTION = 0.8x1024=819d=1100110011b

Output image width of capture sizer

Register	IPP_BASE_ADDR+17h								IPP_BASE_ADDR+16h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						CAPTURE_TARGET_WIDTH									
R/W	NA						RW									
De fault value	02h								80h							

This register defines the width of output image from the capture sizer.

[15:10] Reserved

[9:0] CAPTURE_TARGET_WIDTH: width of the output image from the capture sizer

Output image height of capture sizer

Register	IPP_BASE_ADDR+19h								IPP_BASE_ADDR+18h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						CAPTURE_TARGET_HEIGHT									
R/W	NA						RW									
De fault value	01h								E0h							

This register defines the height of output image from the capture sizer.

[15:10] Reserved

[9:0] CAPTURE_TARGET_HEIGHT: height of the output image from the capture sizer

The above two registers (CAPTURE_TARGET_WIDTH and CAPTURE_TARGET_HEIGHT) define the size of the image which sends to LBUF module to be compressed.

Horizontal ratio of display sizer

Register	IPP_BASE_ADDR+1Bh								IPP_BASE_ADDR+1Ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	HRCS_INTEGER						HRCS_FRACTION									
R/W	RW						RW									
De fault value	04h								00h							

This register defines the horizontal sizer ratio for capture sizer.

[15:10] HRCS_INTEGER: integer part of the capture horizontal ratio

[9:0] HRDS_FRACTION: fraction part of the capture horizontal ratio

Vertical ratio of display sizer

Register	IPP_BASE_ADDR+1Dh								IPP_BASE_ADDR+1Ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VRCS_INTEGER						VRCS_FRACTION									
R/W	RW						RW									
De fault value	04h						00h									

This register defines the vertical sizer ratio for capture sizer.

[15:10] VRCS_INTEGER: integer part of the capture horizontal ratio

[9:0] VRCS_FRACTION: fraction part of the capture horizontal ratio

The ratio computation of capture sizer is just the same as that of display sizer.

Thumbnail size

Register	IPP_BASE_ADDR+1Fh								IPP_BASE_ADDR+1Eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	THUMBNAIL_HEIGHT								THUMBNAIL_WIDTH							
R/W	RW								RW							
De fault value	1Eh								28h							

This register is used to define the size of thumbnail. The thumbnail can be up to 256x256.

[15:8] THUMBNAIL_HEIGHT: Thumbnail height

[7:0] THUMBNAIL_WIDTH: Thumbnail width

Thumbnail sizer ratio

Register	IPP_BASE_ADDR+20h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	THUMBNAIL_RATIO							
R/W	RW							
De fault value	10h							

This register defines the downscaling ratio of thumbnail sizer. The ratio can up to 255

[7:0] THUMBNAIL_RATIO: Thumbnail sizer ratio

The thumbnail sizer is used to generate the thumbnail when capturing still image, multi shot or video clip. Its input is the output of capture sizer. So there's the following relationship:

THUMBNAIL_RATIO = CAPTURE_WIDTH/THUMBNAIL_WIDTH

Prefilter coefficients

Register	IPP_BASE_ADDR+23h								IPP_BASE_ADDR+22h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FIR_COEFFICIENT1								FIR_COEFFICIENT0							
R/W	RW								RW							
De fault value	10h								07h							
Register	IPP_BASE_ADDR+25h								IPP_BASE_ADDR+24h							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	FIR_COEFFICIENT3								FIR_COEFFICIENT2							
R/W	RW								RW							
De fault value	1Eh								1Ah							

The coefficients are described as signed magnitude binary number. The most significant bit is the sign bit and the reset seven bits are magnitude. So the range of the range of each coefficient is –0.1111111b to 0.1111111b, i.e., -0.9921875d to 0.9921875d.

UV offset

Register	IPP_BASE_ADDR+2Bh								IPP_BASE_ADDR+2Ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	V_OFFSET								U_OFFSET							
R/W	RW								RW							
De fault value	00h								00h							

The register defines the YUV offset for sephia effect.

[15:8] V_OFFSET: V offset for sephia effect

[7:0] U_OFFSET: U offset for sephia effect

When sephia effect is enabled, U and V of all the pixels in one frame are set to the fixed values. These registers are also used in special color effect. When U and V values of the pixels exceed the defined range, they are set to U_OFFSET and V_OFFSET.

UV down threshold

Register	IPP_BASE_ADDR+2Dh								IPP_BASE_ADDR+2Ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	U_UP_THRESHOLD								U_DOWN_THRESHOLD							
R/W	RW								RW							
De fault value	FFh								00h							
Register	IPP_BASE_ADDR+2Fh								IPP_BASE_ADDR+2Eh							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	V_UP_THRESHOLD								V_DOWN_THRESHOLD							
R/W	RW								RW							
De fault value	FFh								00h							

This register defines the UV down threshold value.

[31:23] U_UP_THRESHOLD: U up threshold value for special color effect
[23:16] U_DOWN_THRESHOLD: U down threshold value for special color effect
[15:8] V_UP_THRESHOLD: V up threshold value for special color effect
[7:0] V_DOWN_THRESHOLD: V down threshold value for special color effect

For special color effect, if U and V of the pixel exceed a certain range, they are set to fixed values. If U_DOWN_THRESHOLD<=U<=U_UP_THRESHOLD, U keeps unchanged; otherwise it is set to U_OFFSET. It is the same for V. Figure 3 shows the special color effect when both U_OFFSET and V_OFFSET are set to zero.

Frame drop (FRAME_DROP)

Register	IPP_BASE_ADDR+31h								IPP_BASE_ADDR+30h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	FRAME_DROP															
R/W	RW															
De fault value	FFh								FFh							

When user wants to drop some frames, he/she can use this register. In IPP module, there's a counter *frame_counter* to count the frame, which has 4-bit and can count from 0 to 15. If *frame_counter*[3:0]=4'b0000 and bit 0 of this register is set to 1'b0, the current frame is dropped; else if bit 0 is set to 1'b1, the frame is kept. It is just the same when *frame_counter* is other value.

8.5 LBUF register

LBUF Status register

Register	LBUF_BASE_ADDR+00h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	hblank_don	Line number					BUF1_RD_	BUF0_RD_O
R/W	RO	RO					RO	RO
Default value	00h							

[7] hblank_done

Used in decoder_mode = 2'b01 && 2'b10, indicate all lbuf0 data have send to ipp or lcdc

[6:2] Line number

These four bits are used in jpeg decoder mode and decoder mode != 2'b00 for informing host the number of lines to read out; the number is form 0 to 16 and refreshed before each interrupt signal lbuf_cpm_read_int occur

[1] BUF1_RD_OK

This bit set when 8-Line Buffer-1 can be read in the JPEG decode mode
The result of this bit or Bit0 is connected to the interrupt of MARB Unit
This bit reset when JPEG decoder send data into 8_line_buffer again
If buf1 is not used in operation, this bit will be reset

[0] BUF0_RD_OK

This bit set when 8-Line Buffer-0 can be read in the JPEG decode mode
The result of this bit or Bit1 is connected to the interrupt of MARB Unit
This bit reset when JPEG decode send data into 8_line_buffer again

YUV mode register

Register	LBUF_BASE_ADDR +01h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Pixrate				lcd	YUV_mode		
R/W	RW				RW	RW		
Default value	30h							

[7:4] pixrate

Pixrate is used for setting lbuf_ipp_data pixrate in dsplay and decoder mode
0001: 1 pixel/2 clk
0011: 1 pixel/4 clk
0111: 1 pixel/8 clk
1011: 1 pixel/12 clk
1111: 1 pixel/16 clk

[3] lcd

This bit is used in capture mode. When this bit is set by host, YUV data is from LCDC Unit. If this bit is 0, YUV data is from IPP Unit

[2:0] YUV_mode

000: YUV422 , support in all lbuf_mode
001: YUV420, lbuf_mode
010: YUV411, lbuf_mode
011: YUV400, lbuf_mode
100: YUV444, only support in disp or deco mode

LBUF mode register

Register	LBUF_BASE_ADDR +02h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserved		vsyncbypas	decoder mode		Lbuf_mode		
R/W			RW	RW		RW		
Default value	00h							

[7:6] Reserved

[5] vsyncbypass
The bit is used in decoder mode.
1'b1 : lbuf_ipp_vsync = jpeg_lbuf_vsync
1'b0 : transfer vsync signal from jpeg_lbuf_vsync to lbuf_ipp_vsync need delay

[4:3] decoder mode
These 2 bits are used in decoder mode. Data write into line buffer from jpeg encoder. MARB unit read data from buf0 or buf1 in line buffer following decoder mode setting
2'b00: data doesn't pass to IPP unit, MARB read data from buf0 in line buffer unit
2'b01: data pass to IPP unit, return to buf1 in line buffer, MARB read data from buf1 in line buffer unit
2'b10: data pass to IPP, LCDC, return to buf1 in line buffer, MARB read data from buf1 in line buffer unit

[2:0] Lbuf_mode
000: Viewfinder mode
001: Capture mode
010: Display mode
100: JPEG decoder mode

Line buffer internal counter reset register

Register	LBUF_BASE_ADDR +03h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved					preshift_dec	preshift_enc	lbuf_cnt_rese
R/W						RW	RW	RW
Default value	00h							

[7:3] Reserved

[2] preshift_dec
This bit is used to control preshift data in display and decoder mode
1'b1 : do not do preshift
1'b0 : preshift data from –128 - +128 -> 0 – 255

[1] preshift_enc
1'b1 : do not do preshift
1'b0 : preshift data from 0 – 255 -> -128 - +128

[0] lbuf_cnt_reset
The bis is used for reset lbuf internal counter. Set by biu and reset by lbuf self
1'b1 : reset lbuf internal counter
1'b0 : normal operation

Image width register

Register	LBUF_BASE_ADDR +04h								LBUF_BASE_ADDR +05h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				Width											
R/W					RW											
Default value	500h															

[15:12] Reserved
[11:0] Image width
Image width register is used in each mode. The default value for image width is 1280d

Hblank value register

Register	LBUF_BASE_ADDR +06h								LBUF_BASE_ADDR +07h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						Hblank									
R/W							RW									
Default value	00h															

[15:10] Reserved

[9:0] Hblank

Hblank time is set according to pixrate count value

Hblank time = (pixrate+1)*hblank value

Initial address register for lbuf in SRAM

Register	LBUF_BASE_ADDR +09h								LBUF_BASE_ADDR +0ah								LBUF_BASE_ADDR +0bh							
Bit number	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								Ini_addr															
R/W									RW															
Default value	00h																							

[23:17] Reserved

[16:0] ini_addr

Indicate the initial address for lbuf in Marb unit

8.6 MARB register

MARB Control

Register	MARB_BASE_ADDR+00h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved			Update_pt	Start_dec	Stop_cap	Start_cap	Reserved
De fault value				0	0	0	0	-
R/W				RW				-

This register controls the status in different working modes. They are used for clearing the read or write address in the memory. All the bits are cleared by hardware.

[7:5] reserved

[4] update_pt: update priority table.

[3] start_dec: MARB starts decoder for JPEG. MARB read data from memory to JPEG. Data source is defined by “mode_ctrl” bit3

[2] stop_cap: In capture AVI and pc camera mode, it is used for stop video. MARB will finish saving current frame and then stop capturing. This bit must be set when sensor is still working.

[1] start_cap: In capture still image or video or multi-shot or PC camera mode, it is used for start video. In capture still image mode, the next frame will be captured. In capture AVI or multi-shot or PC camera mode, the videos after next frame will be captured.

[0] Reserved

Note: in decoder mode, if JPEG buffer is used as FIFO, host must write data to SRAM after start_dec is set.

MARB Reset Register

Register	MARB_BASE_ADDR+01h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserved			Rst_mem2	Rst_mem 1	Rst_mem 0	Rst_jpeg	Rst_ipp
De fault value				1	1	1	0	0
R/W				RW				

This register reset internal registers in sub-modules. ‘1’ means reset, ‘0’ means not reset. It is controlled by software. For memory, default is reset. Before using memory, set rst_mem* to ‘0’.

[7:5] reserved

[4] rst_mem2: reset 1T-SRAM 2

[3] rst_mem1: reset 1T-SRAM 1

[2] rst_mem0: reset 1T-SRAM 0

[1] rst_jpeg: reset marb_jpeg module
[0] rst_ipp: reset marb_ipp module

Mode Control

Register	MARB_BASE_ADDR+02h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Jbuf_intv_type	Wo_jpeg	Wo_avi_idx	Video_hd	Wo_thumbnail	Enc_dec	Cap_mode	
De fault value	0	0	0	0	0	0	00	
R/W	RW							

This register controls the working mode of MARB.
[7] jbuf_intv_type: JPEG buffer interval interrupt type
0: use jbuf_intv, 1: use frame_intv
[6] wo_jpeg: when capturing, with or without JPEG file. This bit is used for only capturing thumbnail.
0: with JPEG, 1: without JPEG
[5] wo_avi_idx: when capturing JPEG video, write AVI index to avi index buffer. Each JPEG file has 16-bit index to indicate file size. The unit is 4 bytes.
0: write avi index, 1: don't write avi index
[4] video_hd: write video header before each JPEG file. The header is "00dc" + video size
0: write to memory, 1: not write
[3] wo_thumbnail: write thumbnail to thumbnail buffer
0: write, 1: not write
This bit is set before "start_cap" is set. If host set "start_cap" to 1 and "wo_thumbnail" is set to 0, thumbnail image from IPP will saved to thumbnail buffer.
[2] enc_dec: encoder or decoder for marb_jpeg
0: encoder, 1: decoder
This bit determines read or write JPEG buffer. If encoder, then MARB writes JPEG buffer; if decoder, then reads JPEG buffer.
[1:0] cap_mode: capture mode
00: capture still image
01: capture multi-shot
10: capture avi
11: pc camera

Register	MARB_BASE_ADDR+03h							
Bit number	7	6	5	4	3	2	1	0
Bit field name							Tbuf_type	Jbuf_type
De fault value							0	0
R/W							RW	

This register controls the working mode of MARB.

[7:2] reserved

[1] tbuf_type: thumbnail buffer type
0: FIFO, care host read/write point
1: buffer, doesn't care host read/write point

[0] jbuf_type: JPEG buffer type
0: FIFO, care host read/write point
1: buffer, doesn't care host read/write point

JPEG Buffer Start Address

Register	MARB_BASE_ADDR+04h								MARB_BASE_ADDR+05h								MARB_BASE_ADDR+06h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Jbuf_start_0								Jbuf_start_1								Jbuf_start_2							
De fault value																							00	
R/W	R/W								R/W								R/W						RO	

These registers set JPEG buffer start address, in byte unit. The last two bits cannot be written.
Jbuf_start_0: JPEG buffer start address [23:16]
Jbuf_start_1: JPEG buffer start address [15:8]
Jbuf_start_2: JPEG buffer start address [7:0]

JPEG Buffer End Address

Register	MARB_BASE_ADDR+07h								MARB_BASE_ADDR+08h								MARB_BASE_ADDR+09h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Jbuf_end_0								Jbuf_end_1								Jbuf_end_2							
De fault value																							00	
R/W	R/W								R/W								R/W						RO	

These registers set JPEG buffer end address, in byte unit. The last two bits cannot be written.
Jbuf_end_0: JPEG buffer end address [23:16]
Jbuf_end_1: JPEG buffer end address [15:8]
Jbuf_end_2: JPEG buffer end address [7:0]

The size of JPEG buffer is (jbuf_end[23:2] – jbuf_start[23:2]+1)*4

Thumbnail Buffer Start Address

Register	MARB_BASE_ADDR+0Ah								MARB_BASE_ADDR+0Bh								MARB_BASE_ADDR+0Ch							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	tbuf_start_0								tbuf_start_1								tbuf_start_2							
De fault value																							00	
R/W	R/W								R/W								R/W						RO	

These registers set thumbnail buffer start address, in byte unit. The last two bits cannot be written.
tbuf_start_0: thumbnail buffer start address [23:16]
tbuf_start_1: thumbnail buffer start address [15:8]
tbuf_start_2: thumbnail buffer start address [7:0]

Thumbnail Buffer End Address

Register	MARB_BASE_ADDR+0Dh								MARB_BASE_ADDR+0Eh								MARB_BASE_ADDR+0Fh							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	tbuf_end_0								tbuf_end_1								tbuf_end_2							
De fault value																								00
R/W	R/W								R/W								R/W							RO

These registers set thumbnail buffer end address, in byte unit. The last two bits cannot be written.
tbuf_end_0: thumbnail buffer end address [23:16]
tbuf_end_1: thumbnail buffer end address [15:8]
tbuf_end_2: thumbnail buffer end address [7:0]

The size of thumbnail buffer is (tbuf_end[23:2] – tbuf_start[23:2]+1)*4

AVI Index start Address

Register	MARB_BASE_ADDR+1Ch								MARB_BASE_ADDR+1Dh								MARB_BASE_ADDR+1Eh							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Idx_start_0								Idx_start_1								idx_start_2							
De fault value																								
R/W	R/W								R/W								R/W							RO

These registers set AVI index buffer start address, in byte unit. The last two bits cannot be written.
idx_start_0: AVI index buffer start address [23:16]
idx_start_1: AVI index buffer start address [15:8]
idx_start_2: AVI index buffer start address [7:0]

AVI Index End Address

Register	MARB_BASE_ADDR+1Fh								MARB_BASE_ADDR+20h								MARB_BASE_ADDR+21h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Idx_end_0								Idx_end_1								idx_end_2							
De fault value																								
R/W	R/W								R/W								R/W							RO

These registers set AVI index buffer end address, in byte unit. The last two bits cannot be written.
idx_end_0: AVI index buffer end address [23:16]

idx_end_1: AVI index buffer end address [15:8]
idx_end_2: AVI index buffer end address [7:0]

The size of AVI index buffer is (idx_end[23:2] – idx_start[23:2]+1)*4

JPEG Buffer Interval

Register	MARB_BASE_ADDR+23h								MARB_BASE_ADDR+24h								MARB_BASE_ADDR+25h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Jbuf_intv_0								Jbuf_intv_1								Jbuf_intv_2							
De fault value																								00
R/W	R/W								R/W								R/W							RO

These registers set JPEG buffer interval, in byte unit. The last two bits cannot be written. It is actual interval -1.
Jbuf_intv_0: JPEG buffer interval [23:16]
Jbuf_intv_1: JPEG buffer interval [15:8]
Jbuf_intv_2: JPEG buffer interval [7:0]

During JPEG encoder mode, JPEG data is written by JPEG module. After every jbuf_intv number of data is written to buffer, an interrupt is generated. During JPEG decoder mode, JPEG data is read by JPEG module. After every jbuf_intv number of data is read by JPEG, an interrupt is generated.

JPEG Buffer FIFO count threshold

Register	MARB_BASE_ADDR+26h								MARB_BASE_ADDR+27h								MARB_BASE_ADDR+28h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Jbuf_up_0								Jbuf_up_1								Jbuf_up_2							
De fault value																								00
R/W	R/W								R/W								R/W							RO

Register	MARB_BASE_ADDR+29h								MARB_BASE_ADDR+2Ah								MARB_BASE_ADDR+2Bh							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Jbuf_low_0								Jbuf_low_1								Jbuf_low_2							
De fault value																								00
R/W	R/W								R/W								R/W							RO

These registers set JPEG buffer FIFO count threshold, in byte unit. The last two bits cannot be written. Upper threshold is more than lower threshold.
Jbuf_up_0: JPEG buffer FIFO count upper threshold [23:16]
Jbuf_up_1: JPEG buffer FIFO count upper threshold [15:8]
Jbuf_up_2: JPEG buffer FIFO count upper threshold [7:0]

Jbuf_low_0: JPEG buffer FIFO count lower threshold [23:16]
Jbuf_low_1: JPEG buffer FIFO count lower threshold [15:8]

Jbuf_low_2: JPEG buffer FIFO count lower threshold [7:0]

During JPEG encoder mode, JPEG data is written by JPEG module. If FIFO count is more than upper threshold, FIFO status is from less to more. If FIFO count is less than lower threshold, FIFO status is from more to less. When FIFO status changes from less to more, an interrupt is generated. Host will read data from JPEG buffer after the interrupt.

During JPEG decoder mode, JPEG data is read by JPEG module. If (JPEG buffer size - FIFO count) is more than upper threshold, FIFO status is from more to less. If (JPEG buffer size - FIFO count) is less than lower threshold, FIFO status is from less to more. When FIFO status changes from more to less, an interrupt is generated. Host will write data to JPEG buffer after the interrupt

Thumbnail Buffer Interval

Register	MARB_BASE_ADDR+2Ch								MARB_BASE_ADDR+2Dh								MARB_BASE_ADDR+2Eh							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	tbuf_intv_0								tbuf_intv_1								tbuf_intv_2							
De fault value																							00	
R/W	R/W								R/W								R/W						RO	

These registers set thumbnail buffer interval, in byte unit. The last two bits cannot be written. It is actual interval -1.
tbuf_intv_0: thumbnail buffer interval [23:16]
tbuf_intv_1: thumbnail buffer interval [15:8]
tbuf_intv_2: thumbnail buffer interval [7:0]

During capturing mode, thumbnail data is written by IPP module. After every tbuf_intv number of data is written to buffer, an interrupt is generated.

Thumbnail Buffer FIFO count threshold

Register	MARB_BASE_ADDR+2Fh								MARB_BASE_ADDR+30h								MARB_BASE_ADDR+31h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	tbuf_up_0								tbuf_up_1								tbuf_up_2							
De fault value																							00	
R/W	R/W								R/W								R/W						RO	

Register	MARB_BASE_ADDR+32h								MARB_BASE_ADDR+33h								MARB_BASE_ADDR+34h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	tbuf_low_0								tbuf_low_1								tbuf_low_2							
De fault value																							00	
R/W	R/W								R/W								R/W						RO	

These registers set thumbnail buffer FIFO count threshold, in byte unit. The last two bits cannot be written. Upper threshold is more than lower threshold.
tbuf_up_0: thumbnail buffer FIFO count upper threshold [23:16]
tbuf_up_1: thumbnail buffer FIFO count upper threshold [15:8]
tbuf_up_2: thumbnail buffer FIFO count upper threshold [7:0]

tbuf_low_0: thumbnail buffer FIFO count lower threshold [23:16]
tbuf_low_1: thumbnail buffer FIFO count lower threshold [15:8]
tbuf_low_2: thumbnail buffer FIFO count lower threshold [7:0]

During capture mode, thumbnail data is written by IPP module. If FIFO count is more than upper threshold, FIFO status is from less to more. If FIFO count is less than lower threshold, FIFO status is from more to less. When FIFO status changes from less to more, an interrupt is generated. Host will read data from JPEG buffer after the interrupt.

Frame Interval

Register	MARB_BASE_ADDR+4Ah							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Frame_interval							
De fault value	1BH							
R/W	RW							

This register set frame interval for AVI or multi-shot file. AVI file contains a number of video frames and audio streams. For capturing AVI, frame rate must be a constant. Frame interval can be set to 1 second frame number. If the frame rate is 30f/s, frame interval will be set to 30. The total frame size must be less than half of JPEG buffer. Thus host can read jpeg data in enough time.

Memory Control

Register	MARB_BASE_ADDR+4Bh							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserverd		Mem_ss		reserved	Mem2_on	Mem1_on	Mem0_on
De fault value			2’b11			1	1	1
R/W		RW						

This register controls the memory statu
[7:6] reserved
[5:4] mem_ss: memory speed select
00: 70-83 MHz
01: 50-70 MHz
10: 30-50 MHz
11: 20-30 MHz
These two bits control the speed select pins of 1T-SRAM

[3] reserved
[2] mem2_on: the third block of SRAM power on
0: power down, 1: power on
[1] mem1_on: the second block of SRAM power on
0: power down, 1: power on
[0] mem0_on: first block of SRAM power on
0: power down, 1: power on
These four bits set memory to shutdown mode to save power. If the memory is not used, set memn_on to 0 to save power.

Memory Frequency

Register	MARB_BASE_ADDR+4Ch							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Mem_freq							
De fault value	1BH							
R/W	RW							

This register controls the memory frequency
[7] reserved
[6:0] mem_freq: memory frequency, the unit is MHz

It is used for mem_ctrl module. When memory is in shutdown mode, memory access may resume 16us after resumption of a stable clock signal after exiting non-retentive standby. This register is used for counting 16us.

Multi-shot Frame Count

Register	MARB_BASE_ADDR+4Dh							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Mshot_fcnt							
De fault value	0FH							
R/W	RW							

This register is used in capturing multi-shot images. After start capturing, the number of image to be captured is set by this register. The value is actual frame count – 1.

Priority Registers

Register	MARB_BASE_ADDR+4Eh								MARB_BASE_ADDR+4Fh								MARB_BASE_ADDR+50h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	IPP				JPEG				lcd_wr_vb				Lcd_rd_vb				ge				Lcd_rd_gb			
De fault value	6				5				3				2				7				4			
R/W	R/W																							

This registers set user-defined priority for each source. “0” and “1” are not allowed to the table because “0” is reserved for host. “1” is reserved for LBUF. These registers take effect only after update priority is written.

JPEG Write/Read Pointer

Register	MARB_BASE_ADDR+60h								MARB_BASE_ADDR+61h								MARB_BASE_ADDR+62h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Jbuf_pt_0								Jbuf_pt_1								Jbuf_pt_2							
De fault value																								
R/W	RO								RO								RO							

These registers show JPEG write/read pointer in JPEG buffer, next point will be written/read.
Jbuf_pt_0: JPEG write/read pointer [23:16]
Jbuf_pt_1: JPEG write/read pointer [15:8]
Jbuf_pt_2: JPEG write/read pointer [7:0]

Note: before read these registers, write jbuf_pt_0 first, write data is any. This is used for lock three bytes.

Thumbnail Buffer Write Pointer

Register	MARB_BASE_ADDR+63h								MARB_BASE_ADDR+64h								MARB_BASE_ADDR+65h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	tbuf_pt_0								tbuf_pt_1								Tbuf_pt_2							
De fault value																								
R/W	RO								RO								RO							

These registers show IPP write pointer in thumbnail buffer
tbuf_pt_0: IPP write pointer [23:16]
tbuf_pt_1: IPP write pointer [15:8]
tbuf_pt_2: IPP write pointer [7:0]

Note: before read these registers, write tbuf_pt_0 first, write data is any. This is used for lock three bytes.

Last JPEG Write/Read Pointer

Register	MARB_BASE_ADDR+66h								MARB_BASE_ADDR+67h								MARB_BASE_ADDR+68h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	last_pt_0								last_pt_1								Last_pt_2							
De fault value																								
R/W	RO								RO								RO							

These registers show Last JPEG write pointer in JPEG buffer
last_pt_0: Last JPEG write/read pointer [23:16]
last_pt_1: Last JPEG write/read pointer [15:8]
last_pt_2: Last JPEG write/read pointer [7:0]

Frame Counter Registers

Register	MARB_BASE_ADDR+7 0h								MARB_BASE_ADDR+7 1h								MARB_BASE_ADDR+7 2h								MARB_BASE_ADDR+7 3h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Frame_cnt_0								Frame_cnt_1								Frame_cnt_2								Frame_cnt_3							
De fault value	-								-								-								-							
R/W	RO																															

These registers shows total frame counter, it is reset by “start_cap” in marb_ctrl register
Frame_cnt_0: Total frame counter [31:24]
Frame_cnt_1: Total frame counter [23:16]
Frame_cnt_2: Total frame counter [15:8]
Frame_cnt_3: Total frame counter [7:0]

Video Size Registers

Register	MARB_BASE_ADDR+7 4h								MARB_BASE_ADDR+7 5h								MARB_BASE_ADDR+7 6h								MARB_BASE_ADDR+7 7h							
Bit number	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bit field name	Video_size_0								Video_size_1								Video_size_2								Video_size_3							
De fault value	-								-								-								-							
R/W	RO																															

These registers shows total JPEG file size, it is reset by “start_cap” in marb_ctrl register, in double word unit
Video_size_0: Total jpeg file size [31:24]
Video_size_1: Total jpeg file size [23:16]
Video_size_2: Total jpeg file size [15:8]
Video_size_3: Total jpeg file size [7:0]

JPEG file is from ffd8 to ffd9

MARB Status Register

Register	MARB_BASE_ADDR+78h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Tfifo_err	Thum_done	reserved	Tbuf_cnt_sta	Jfifo_err	Jpeg_done	reserved	Jbuf_cnt_stat
De fault value	-	-	-	-	-	-	-	-
R/W	RO							

This register shows MARB status

[7] tfifo_err: Thumbnail FIFO overflow

[6] thum_done: all thumbnail image saved to SRAM
[5] Reserved
[4] tbuf_cnt_stat: thumbnail FIFO count more than threshold
[3] jfifo_err: JPEG FIFO overflow
[2] jpeg_done: all JPEG frames saved to SRAM
[1] Reserved
[0] jbuf_cnt_stat: JPEG FIFO count more than threshold

8.7 LCDC register

DC (Display Configuration)

Register	LCD_BASE_ADDR+01h							
Bit number	15	14	13	12	11	10	9	8
Bit field name	VBUFM	SFBUF	GBUF_SEL	DECM				LPM
R/W	RW	RW	RW	RW	RW		RW	RW
De fault value	0	0	0	0	0		0	0
Register	LCD_BASE_ADDR+00h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	BLDEN	OVLEN	OM		GAMEN	DITSEL		DITEN
R/W	RW	RW	RW		RW	RW		RW
De fault value	0	0	00		0	00		0

This register setups the basic operation.

[0] DITEN (dither enable)

0 : disable

1 : enable

[2:1] DITSEL (dither select)

DITSEL R:G:B

00 : 4:4:4

01 : 5:6:5

10 : 6:6:6

[3] GAMEN (Gamma Correction enable)

0 : disable

1 : enable

[5:4] OM (Overlay mode)

2'b00 : Transparent overlay. If the color data of pixels in layer B is equal to the overlay key color, the pixel shows the color of layer A; else shows that of layer B;

2'b01: And overlay. If the color data of pixels in layer B is equal to the overlay key color, the pixel shows the color of (layer A AND layer B); else shows that of layer B;

2'b10: Or overlay. If the color data of pixels in layer B is equal to the overlay key color, the pixel shows the color of (layer A Or layer B); else shows that of layer B;

2'b11: Invert overlay. If the color data of pixels in layer B is equal to the overlay key color, the pixel shows the color of (INV layer A); else shows that of layer B.

[6] OVLEN (overlay enable)

0: disable
1: enable

[7] BLDEN (Alpha blending enable)
0: disable
1: enable
[8] LPM (layer priority mode)
0: layer A has higher priority, in front of layer B
1: layer B has higher priority, in front of layer A

[12] DECM (decode mode)
0: normal mode
1: decode mode
[13] GBUF_SEL (graphic buffer select)
0: select graphic buffer 0
1: select graphic buffer 1
[14] SFBUF (single frame buffer enable)
0: disable
1: enable (video buffer must act as frame buffer)

[15] VBUFM (video buffer mode)
0: video buffer acts as line buffer
1: video buffer acts as frame buffer, this bit should be set in decode with frame mode

DE (Display Enable)

Register	LCD_BASE_ADDR+2h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved						AE	Reserved
R/W					RW	RW	RW	RW
De fault value					0	0	0	0

This register controls the display signal output of each layer.

[0] Reserved
[1] AE (Layer-A Enable)
0 Disable A layer display
1 Enable A layer display
[7:2] Reserved

DRST (Display Reset)

Register	LCD_BASE_ADDR+04h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Lcdc_reset							
R/W	W							
De fault value	0							

Lcdc_reset : module level reset. Reset all logic excepting register configuration.

UPDATE(UPDATE)

Register	LCD_BASE_ADDR+06h							
Bit number	7	6	5	4	3	2	1	0
Bit field name					reserved	reserved	Update_bg	Update_b
R/W					w	w	w	w
De fault value					0	0	0	0

[0] update_b
mcu send out a pulse signal to update b layer, this bit is self-clear
[1] update_bg
mcu send out a pulse signal to update bg layer, this bit is self-clear
[7:3] reserved

RM (Rotation Mode)

Register	LCD_BASE_ADDR+08h							
Bit number	7	6	5	4	3	2	1	0
Bit field name			LBRM			LARM		
R/W			RW			rw		
De fault value			000			3'b000		

[2:0] Layer A Rotation Mode
000 : 0 degree rotation
001 : 90 degree rotation
010: 180 degree rotation
011 : 270 degree rotation
100 : 0 degree rotation then mirror
101 : 90 degree rotation then mirror
110: 180 degree rotation then mirror
111 : 270 degree rotation then mirror

[5:3] Layer B Rotation Mode

000 : 0 degree rotation
001 : 90 degree rotation
010: 180 degree rotation
011 : 270 degree rotation
100 : 0 degree rotation then mirror
101 : 90 degree rotation then mirror
110: 180 degree rotation then mirror
111 : 270 degree rotation then mirror

VBBA0 (Video Buffer 0 Base Address)

Register	LCD_BASE_ADDR+0bh								LCD_BASE_ADDR+0ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VBBA0															
R/W	RW															
De fault value	0															

Register	LCD_BASE_ADDR+0ch							
Bit number	23	22	21	20	19	18	17	16
Bit field name	VBBA0							
R/W	RW							
De fault value	0							

Set the starting address of video buffer 0 in the 1-T SRAM space.

VBBA1 (Video Buffer 1 Base Address)

Register	LCD_BASE_ADDR+0fh								LCD_BASE_ADDR+0eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VBBA1															
R/W	RW															
De fault value	0															

Register	LCD_BASE_ADDR+10h							
Bit number	23	22	21	20	19	18	17	16

Bit field name	VBBA1
R/W	RW
De fault value	0

Set the starting address of video buffer 1 in the 1-T SRAM space.

GBBA0 (Graphic Buffer 0 Base Address)

Register	LCD_BASE_ADDR+13h								LCD_BASE_ADDR+12h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GBBA0															
R/W	RW															
De fault value	0															

Register	LCD_BASE_ADDR+14h							
Bit number	23	22	21	20	19	18	17	16
Bit field name	GBBA0							
R/W	RW							
De fault value	0							

Set the starting address of graphic buffer 0 in the 1-T SRAM space.

GBBA1 (Graphic Buffer 1 Base Address)

Register	LCD_BASE_ADDR+17h								LCD_BASE_ADDR+16h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	GBBA1															
R/W	RW															
De fault value	0															

Register	LCD_BASE_ADDR+18h							
Bit number	23	22	21	20	19	18	17	16

Bit field name	GBBA1
R/W	RW
De fault value	0

Set the starting address of graphic buffer 1 in the 1-T SRAM space.

AX (Layer-A Position X)

Register	LCD_BASE_ADDR+1bh								LCD_BASE_ADDR+1ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								AX							
R/W									RW							
De fault value									0							

Set the X coordinates of Layer-A display position on the LCD screen.

9'h000 0pixel

|

9'h1FF 511pixel

AY (Layer-A Position Y)

Register	LCD_BASE_ADDR+1dh								LCD_BASE_ADDR+1ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								AY							
R/W									RW							
De fault value									0							

Set the Y coordinates of Layer-A display position on the LCD screen.

9'h000 0pixel

|

9'h1FF 511pixel

AW (Layer-A Width)

Register	LCD_BASE_ADDR+1fh								LCD_BASE_ADDR+1eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								AW							
R/W									RW							
De fault value									0							

Set the horizontal width of Layer-A window on the LCD screen.

9'h000 1pixel
|
9'h1FD 510pixel

AH (Layer-A Height)

Register	LCD_BASE_ADDR+21h								LCD_BASE_ADDR+20h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								AH							
R/W									RW							
De fault value									0							

Set the height of Layer-A window on the LCD screen.

9'h000 1pixel
|
9'h1FF 512pixel

ADX0 (Layer-A Display Position X0)

Register	LCD_BASE_ADDR+23h								LCD_BASE_ADDR+22h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								ADX0							
R/W									RW							
De fault value									0							

Set the X coordinates of display start position on the logical frame of Layer-A0. Position is set in pixel unit from the base address of the logical frame.

9'h000 0pixel
|
9'h1FF 511pixel

ADY0 (Layer-A Display Position Y0)

Register	LCD_BASE_ADDR+25h								LCD_BASE_ADDR+24h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								ADY0							
R/W									RW							
De fault value									0							

Set the Y coordinates of display start position on the logical frame of Layer-A0. Position is set in pixel unit from the base address of the logical frame

9’h000 1pixel
|
9’h1FF 512pixel

AMW (Layer-A Memory Width)

Register	LCD_BASE_ADDR+27h								LCD_BASE_ADDR+26h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								AMW							
R/W									RW							
Default value									0							

Set the width of the logical frame of Layer-A in pixel unit.

9’h000 1 pixel
|
9’h1FF 512 pixel

AMH (Layer-A Memory Height)

Register	LCD_BASE_ADDR+29h								LCD_BASE_ADDR+28h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								AMH							
R/W									RW							
Default value									0							

Set the height of the logical frame of Layer-A in pixel unit.

9’h000 0pixel
|
9’h1FF 511pixel

BX (Layer-B Position X)

Register	LCD_BASE_ADDR+2bh								LCD_BASE_ADDR+2ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BX							
R/W									RW							
Default value									0							

Set the X coordinates of Layer-B display position on the LCD screen.

9'h000 0pixel
|
9'h1FF 511pixel

BY (Layer-B Position Y)

Register	LCD_BASE_ADDR+2dh								LCD_BASE_ADDR+2ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BY							
R/W									RW							
Default value									0							

Set the Y coordinates of Layer-B display position on the LCD screen.

9'h000 0pixel
|
9'h1FF 511pixel

BW (Layer-B Width)

Register	LCD_BASE_ADDR+2fh								LCD_BASE_ADDR+2eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BW							
R/W									RW							
Default value									0							

Set the width of Layer-B window on the LCD screen.

9'h000 1pixel
|
9'h1FF 512pixel

BH (Layer-B Height)

Register	LCD_BASE_ADDR+31h								LCD_BASE_ADDR+30h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BW							
R/W									RW							
De fault value									0							

Set the height of Layer-B window on the LCD screen.

9'h001 1pixel
|
9'h1FF 512pixel

BDX0 (Layer-B Display Position X0)

Register	LCD_BASE_ADDR+33h								LCD_BASE_ADDR+32h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BDX0							
R/W									RW							
De fault value									0							

Set the X coordinates of display start position on the logical frame of Layer-B0. Position is set in pixel unit from the base address of the logical frame.

9’h000 0pixel

|

9’h1FF 511pixel

BDY0 (Layer-B Display Position Y0)

Register	LCD_BASE_ADDR+35h								LCD_BASE_ADDR+34h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BDY0							
R/W									RW							
De fault value									0							

Set the Y coordinates of display start position on the logical frame of Layer-B0. Position is set in pixel unit from the base address of the logical frame

9’h000 0pixel

|

9’h1FF 511pixel

BDX1 (Layer-B Display Position X1)

Register	LCD_BASE_ADDR+37h								LCD_BASE_ADDR+36h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BDX1							
R/W									RW							
De fault value									0							

Set the X coordinates of display start position on the logical frame of Layer-B1. Position is set in pixel unit from the base address of the logical frame.

9’h000 0pixel

|

9’h1FF 511pixel

BDY1 (Layer-B Display Position Y1)

Register	LCD_BASE_ADDR+39h								LCD_BASE_ADDR+38h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BDY1							
R/W									RW							
De fault value									0							

Set the Y coordinates of display start position on the logical frame of Layer-B1. Position is set in pixel unit from the base address of the logical frame

9’h000 0pixel
|
9’h1FF 511pixel

BMW0 (Layer-B Memory Width in Graphic Buffer 0)

Register	LCD_BASE_ADDR+3bh								LCD_BASE_ADDR+3ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BMW0							
R/W									RW							
De fault value									0							

Set the width of the logical frame of Layer-B in pixel unit.

9’h000 1pixel
|
9’h1FF 512pixel

BMW1 (Layer-B Memory Width in Graphic Buffer 1)

Register	LCD_BASE_ADDR+3dh								LCD_BASE_ADDR+3ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BMW1							
R/W									RW							
De fault value									0							

Set the height of the logical frame of Layer-B in pixel unit.

9’h000 1pixel
|
9’h1FF 512pixel

BRX (X coordinate of Layer-B Refresh Block)

Register	LCD_BASE_ADDR+3fh								LCD_BASE_ADDR+3eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BRX							
R/W									RW							
De fault value									0							

Set the horizontal coordinate of the top-left corner of the block to be refreshed in Layer-B in pixel unit. The original point of this coordinate is the top-left corner of rotated layer B on the LCD screen.

9'h000 0pixel
|
9'h1FF 511pixel

BRY (Y coordinate of Layer-B Refresh Block)

Register	LCD_BASE_ADDR+41h								LCD_BASE_ADDR+40h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BRY							
R/W									RW							
De fault value									0							

Set the vertical coordinate of the top-left corner of the block to be refreshed in Layer-B in pixel unit. The original point of this coordinate is the top-left corner of rotated layer B on the LCD screen.

9'h000 0pixel
|
9'h1FF 511pixel

BRW (Width of Layer-B Refresh Block)

Register	LCD_BASE_ADDR+43h								LCD_BASE_ADDR+42h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BRW							
R/W									RW							
De fault value									0							

Set the width of the Layer-B refresh block in pixel unit.

9’h000 1pixel

|

9’h1FE 511pixel

BRH (Height of Layer-B Refresh Block)

Register	LCD_BASE_ADDR+45h								LCD_BASE_ADDR+44h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BRH							
R/W									RW							
De fault value									0							

Set the height of the Layer-B refresh block in pixel unit.

9’h000 1pixel

|

9’h1FF 512pixel

BGX (X coordinate of Background Layer Refreshing Block)

Register	LCD_BASE_ADDR+47h								LCD_BASE_ADDR+46h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BGX							
R/W									RW							
De fault value									0							

Set the horizontal coordinate of the top-left corner of the block to be refreshed in background layer in pixel unit. The original point of this coordinate is the top-left corner of the LCD screen.

9’h000 0pixel

|

9’h1FF 511pixel

BGY (Y coordinate of Background Layer Refresh Block)

Register	LCD_BASE_ADDR+49h								LCD_BASE_ADDR+48h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BRY							
R/W									RW							
De fault value									0							

Set the vertical coordinate of the top-left corner of the block to be refreshed in background layer in pixel unit. The original point of this coordinate is the top-left corner of the LCD screen.

9'h000 0pixel
|
9'h1FF 511pixel

BGW (Width of Background Layer Refreshing Block)

Register	LCD_BASE_ADDR+4bh								LCD_BASE_ADDR+4ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BGW							
R/W									RW							
De fault value									0							

Set the width of the background layer refreshing block in pixel unit.

9'h000 1pixel
|
9'h1FE 511pixel

BGH (Height of Background Layer Refreshing Block)

Register	LCD_BASE_ADDR+4dh								LCD_BASE_ADDR+4ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BGH							
R/W									RW							
De fault value									0							

Set the height of the background layer refreshing block in pixel unit.

9'h000 1pixel
|
9'h1FF 512pixel

OKC & OKCM(Overlay Key Color & Overlay Key Color Mode)

Register	LCD_BASE_ADDR+51h								LCD_BASE_ADDR+50h							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved							OKCM	OKC							

R/W								RW			RW						
De fault value								0			0						
Register	LCD_BASE_ADDR+4fh								LCD_BASE_ADDR+4eh								
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field name	OKC																
R/W	RW																
De fault value	0																

[23-0] OKC (overlay key color)

Define the overlay color code adopted for the front layer. Compare the bit field

Corresponding to the definition of input data format. In case of indirect color

mode, bit[7-0] are applied.

[24] OKCM (overlay key color mode)

Set the definition of overlay key color.

0 use 24'h000000 as overlay key color

1 use [23:0] color in this register as overlay key color

BR (Blend Ratio)

Register	LCD_BASE_ADDR+53h								LCD_BASE_ADDR+52h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								BRATIO							
R/W																
De fault value																

[7-0] BRATIO (Blend Ratio)

Set alpha value (transparent ratio) for blending

00000000 Alpha value = 0

00000001 Alpha value = 1/256

| |

11111111 Alpha value = 255/256

GBUFFMT (Graphic buffer data format)

Register	LCD_BASE_ADDR+54h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	reserved		reserved		bfm			
R/W					RW			

De fault value			0000

[3-0] BFM (Graphic buffer data Format)

Set the format of Layer-B input data

1001 24 bit direct color with alpha bit(RGB888 & Alpha bit);

1000 24 bit direct color(RGB888);

0111 18 bit direct color (RGB6:6:6)

0110 Direct color (16bpp, RGB 5:6:5)

0101 Direct color (16bpp, Alpha+RGB 5:5:5)

0100 12 bit direct color (RGB4:4:4)

0011 Indirect color (8bpp)

0010 Indirect color (4bpp)

0001 Indirect color (2bpp)

0000 Indirect color (1bpp)

BGD (Back Ground Data)

Register	LCD_BASE_ADDR+57h							LCD_BASE_ADDR+56h								
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BGD															
R/W	RW															
De fault value	0															

Register	LCD_BASE_ADDR+58h							
Bit number	23	22	21	20	19	18	17	16
Bit field name	BGD							
R/W	RW							
De fault value	0							

Set the background color

[23-0]

Set the background color in direct color format (RGB 8:8:8 no alpha bit). Truncating the least significant bits can get the other format.

PD & PA (PALETTE DATA and PALETTE ADDRESS)

Register	LCD_BASE_ADDR+5dh	LCD_BASE_ADDR+5ch

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	color_data(G,B)															
R/W	RW															
De fault value	0															
Register	LCD_BASE_ADDR+5fh								LCD_BASE_ADDR+5eh							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Color_address								color_data(R)							
R/W	RW								RW							
De fault value	0								0							

[23:0] the color table value we want to send to ram . it is include R value ,G value , B value
[31:24] the address of the 24-bit color data, write operation on this byte will write the 24-bit data into the RAM

GLT_R0 – GLT_R17 (gamma look up table R0 – R16)

Register	LCD_BASE_ADDR+8ch – 9ch							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Lut_r0-r16							
R/W	RW							
De fault value	0							

Gamma look up table R value

GLT_G0 – GLT_G17 (gamma look up table G0 – G16)

Register	LCD_BASE_ADDR+9eh-aeh							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Lut_g0-g16							
R/W	RW							
De fault value	0							

Gamma look up table G value

GLT_B0 – GLT_B17 (gamma look up table B0 –B16)

Register	LCD_BASE_ADDR+b0h – c0h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Lut_b0-b16							
R/W	RW							
De fault value	0							

Gamma look up table B value

DEDLY (data enable delay)

Register	LCD_BASE_ADDR+C8h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	DEDLY							
R/W	RW							
De fault value	00000000							

[7:0] DEDLY (data enable delay)

The number of clock cycles from address enable assertion to data enable assertion. The min value is 15 and the max value is 256.

DEINTV (data enable interval)

Register	LCD_BASE_ADDR+CAh							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved		DEINTV					
R/W	RW		RW					
De fault value	00		000000					

[5:0] DEINTV (data enable interval)

The minimum number of clock cycles between the two data enable signals. The min value is 2 and the max value is 63.

LINEINTV (line interval)

Register	LCD_BASE_ADDR+CCh							
Bit number	7	6	5	4	3	2	1	0
Bit field name	LINEINTV							
R/W	RW							
De fault value	00000000							

[7:0] LINEINTV (line interval)

The number of clock cycles from the last data enable assertion to the address enable assertion in the next line. The min value is 15 and the max value is 256.

8.8 LCDIF register

HEAD_RS_FLAG (head RS sign)

Register	LCD_BASE_ADDR+01h								LCD_BASE_ADDR+00h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Head_rs_flag															
R/W	RW															
De fault value	0															

[15:0] when sending head data . RS signal value equal to the head_rs_flag.

HEAD_NUM (head number register)

Register	LCD_BASE_ADDR+03h								LCD_BASE_ADDR+02h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Head_num															
R/W	RW															
De fault value	0															

[15:0] set the head number.

START_X_POS (start x pixel position)

Register	LCD_BASE_ADDR+05h								LCD_BASE_ADDR+04h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Start_x_pos															
R/W	RW															
De fault value	0															

[15:0] set start x point position in the HEAD_DATA register group..

START_Y_POS (start y pixel position)

Register	LCD_BASE_ADDR+07h								LCD_BASE_ADDR+06h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Start_y_pos															
R/W	RW															
De fault value	0															

[15:0] set start y point position in the HEAD_DATA register group..

END_X_POS (end x pixel position)

Register	LCD_BASE_ADDR+09h								LCD_BASE_ADDR+08h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	end_x_pos															
R/W	RW															
De fault value	0															

[15:0] set end x point position in the HEAD_DATA register group..

END_Y_POS (end y pixel position)

Register	LCD_BASE_ADDR+0bh								LCD_BASE_ADDR+0ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	end_y_pos															
R/W	RW															
De fault value	0															

[15:0] set end y point position in the HEAD_DATA register group..

HEAD_CONFIG (head config register)

Register	LCD_BASE_ADDR+0dh								LCD_BASE_ADDR+0ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Head_config															
R/W	RW															
De fault value	0															

[0] hen head enable when set 1

[3] update the all position of x and y to the register. It will set 1 after setting all other register

CONFIG (config register)

Register	LCD_BASE_ADDR+0fh								LCD_BASE_ADDR+0eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	config															
R/W	RW															
De fault value	0															

[3] drs set the rs high enable

[4] cs_sel select the main lcd or sub lcd . when set 0 selcet the main lcd panel

[5] sel_68 select the 68 serial bus

LCDRST (LCD Reset)

Register	LCD_BASE_ADDR+10h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							LCDRST
R/W								RW
De fault value								1

Register	LCD_BASE_ADDR+13h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Lcd_test							
R/W	W							
De fault value	0							

Lcd_reset : lcd reset signal. Low active

Lcd_test : when the lcd_test set one. The A layer register update at once

DPAD (PAD)

Register	LCD_BASE_ADDR+15h								LCD_BASE_ADDR+12h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved							PDE	DPAD							
R/W								RW	RW							
De fault value								1	0							

Specify the data to be output in the PAD field. (Refer 5.5.1)

[7-0] DPAD Data PAD)

Set the output data in PAD field

[8] PDE Pad Enable)

Select data to be output in the PAD field.

0 Data of field **[1]** is output (Refer 5.5.1)

1 Data specified in DPAD is output

WCONT (Write Control)

Register	LCD_BASE_ADDR+17h								LCD_BASE_ADDR+14h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved	DHC					WRC					WSC				

R/W		RW	RW	RW
De fault value		1	1	0

Set the write access parameter for built-on SRAM type LCD access

[4-0] WSC (Write Set up Cycle)

Specify data set up cycle count

00000 0cycle

|

11111 31cycle

[9-5] WRC (Write Data Cycle)

Specify write access cycle. DIS_WRn signal assertion period is specified by the parameter.

00000 1cycle

|

11111 32cycle

[14-10] DHC (Data Hold Cycle)

Specify the hold time of data in valid.

00000 0cycle

|

11111 31cycle

RCONT (Read Control)

Register	LCD_BASE_ADDR+19h								LCD_BASE_ADDR+16h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved	ROC					RAC					RSC				
R/W		RW					RW					RW				
De fault value		1					1					0				

Set the read access parameter for built-on SRAM type LCD access

[4-0] RSC (Read Set up Cycle)

Specify set up cycle count of LCD_RDn signal

00000 0cycle

|

11111 31cycle

[9-5] RAC (Read Access Cycle)

Specify read access cycle. LCD_RDn signal assertion period is specified by this parameter.

00000 1cycle

|

11111 32cycle

[14-10] ROC (Read Off Cycle)

Specify the access interval period between read cycle and next read cycle.

Or specify the transition period from read cycle to idle state.

00000 0cycle

|

11111 31cycle

EXWEN (External Write ebable)

Register	LCD_BASE_ADDR+18h							
Bit number	7	6	5	4	3	2	1	0
Bit field name								
R/W								
De fault value								0

[0] when set 1. send EXW0 to LCD bus

[1] when set 1 send EXW1 to LCD bus.

EXW0 (External Write0)

Register	LCD_BASE_ADDR+1ch								LCD_BASE_ADDR+1ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	EXW0															
R/W	RW															
De fault value	0															

Set output data to this register. According to the timing sequence described in 5.6.2, the data put in this register is output through LCD_D[15-0]. During this process LCD_RS signal is kept in Low.

5.8.71 EXW1 (External Write1)

Register	LCD_BASE_ADDR+21h								LCD_BASE_ADDR+1eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	EXW1															
R/W	RW															
De fault value	0															

Set output data to this register. According to the timing sequence described in 5.6.2, the data put in this register is output through LCD_D[15-0]. During this process LCD_RS signal is kept in High.

EXR0 (External Read0)

Register	LCD_BASE_ADDR+23h								LCD_BASE_ADDR+22h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	EXR0															
R/W	R															
De fault value	0															

Read current data on DIS_D[15-0] from this register. From the register of LCD panel directed by EXW0 and EXW1, current data put in that register is read from ERD0 field.

EXRS (External Read RS select)

Register	LCD_BASE_ADDR+26h							
Bit number	7	6	5	4	3	2	1	0
Bit field name								exrs
R/W								rw
De fault value								0

After sending the LCD address we want to read . when LCD_RS signal want to be high. Set the exrs one. Or when LCD_RS signal want to be low .Set the exrs zero. The read value of the LCD can be read from EXR0 register

SER_COMM (serial transfers command)

Register	LCD_BASE_ADDR+29h								LCD_BASE_ADDR+28h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Ser_comm															
R/W	R															
De fault value	0															

[15:0] ser_comm. Serial transfers command.

SER_DATA (serial transfers DATA)

Register	LCD_BASE_ADDR+2bh								LCD_BASE_ADDR+2ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Ser_comm															
R/W	R															
De fault value	0															

[15:0] ser_comm. Serial transfers command.

SER_NUM (serial transfers NUM)

Register	LCD_BASE_ADDR+2dh								LCD_BASE_ADDR+2ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Ser_num															
R/W	R															
De fault value	0															

[7:0] serial transfers data bit number

[15:8] serial transfers command bit number..

SER_SEND (serial transfers send)

Register	LCD_BASE_ADDR+2fh								LCD_BASE_ADDR+2eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Ser_send															
R/W	R															
De fault value	0															

[0] when sent 1 .send the serial data.

SCK_WIDTH (SCK width)

Register	LCD_BASE_ADDR+31h								LCD_BASE_ADDR+30h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Ser_send															
R/W	R															
De fault value	0															

[7:0] set the SCK width . unit is the main clock width..

A0_SEL (A0 select)

Register	LCD_BASE_ADDR+33h								LCD_BASE_ADDR+32h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	A0_sel															
R/W	R															
De fault value	0															

[0] when set 0 . the A0 is low level when sending the serial command .else is high level

HEAD_DATA0---HEAD_DATA15

Register	LCD_BASE_ADDR+34h-52h							
Bit number	15-7	6	5	4	3	2	1	0
Bit field name	Head_data0 – head_data_15							
R/W	RW							
De fault value	0							

[15:0] set the head value according to the head request.

BIAS_X

Register	LCD_BASE_ADDR+54h							
Bit number	15-7	6	5	4	3	2	1	0
Bit field name	Bias_x							
R/W	RW							
De fault value	0							

[7:0] set the x bias value.
[8] div_x one half of x start address is sent .

BIAS_Y

Register	LCD_BASE_ADDR+56h							
Bit number	15-7	6	5	4	3	2	1	0
Bit field name	Bias_y							
R/W	RW							
De fault value	0							

[7:0] set the y bias value.
[8] div_y one half of y start address is sent.

DMAP (Display Mapping)

Register	LCD_BASE_ADDR+58h							
Bit number	7	6	5	4	3	2	1	0
Bit field name	conv	MPPAT			DFMT			
R/W	rw	RW			RW			
De fault value	0	0			0			

Set display format and mapping of the image data. Please refer to the related chapter for the detailed information.

[3-0] DFMT (Data Format)

Define display data bus format

0000	R:G:B=5:6:5 (16bit)
0001	R:G:B=5:5:5 (16bit)
0010	R:G:B=4:4:4 (16bit)
0011	R:G:B=6:6:6 (16bit)
0100	R:G:B=5:6:5 (8bit, 2byte/pixel)
0101	R:G:B=6:6:6 (18bit, 2byte/pixel, little endian)
0110	R:G:B=6:6:6 (18bit, 2byte/pixel, big endian)
0111	Reserved
1000	R:G:B=4:4:4 (8bit, 2byte/pixel, big endian)
1001	R:G:B=4:4:4 (8bit, 2byte/pixe, little endian)
1010	R:G:B=each 6, 8bit (16bit, 2hw/pixel, big endian)
1011	R:G:B=each 6, 8bit (16bit, 2hw/pixel, little endian)
1100	R:G:B=4:4:4 (8bit, 3byte/2pixel, big endian)
1101	R:G:B=4:4:4 (8bit, 3byte/2pixel, little endian)
1110	R:G:B=each 6, 8bit (16bit, 3hw/2pixel, big endian)
1111	R:G:B=each 6, 8bit (16bit, 3hw/2pixel, little endian)

[6-4] MPPAT (Mapping Pattern)

Set the RGB bit order in the data path according to the format definition of DFMT.

(1)-(2)-(3)	
000	R-G-B
001	R-B-G
010	G-R-B
011	G-B-R
100	B-R-G
102	B-G-R

[7] CONV

exchange the data[7:0] to data[15:8]

8.9 JPEG register

JPEG Mode Register

Register address	JPEG_BASE_ADDR+00H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved	AF	CH	SQ	CQ	SH	Mode	
R/W		R/W	R/W	R/W	R/W	R/W	R/W	
Default		0	0	0	0	0	0	0

Bit1-0 Working Mode

These two bits define working mode.

mode[2:0]=2'b00: capture mode
mode[2:0]=2'b01 : display mode
mode[2:0]=2'b10 : jpeg encode mode
mode[2:0]=2'b11 : jpeg decode mode

Bit2 SH: Simple header enable

When setting this bit, jpeg module generates simple header which only contains SOI and COM segments. This bit is used in capture mode and decode mode.

Bit3 CQ: Use custom quantization table.

This bit is used in JPEG decode mode or display mode. If JPEG file uses user defined quantization table, this bit is set to 1. In this case, jpeg module needs to calculate BRC table in different way. If quantization table is generated by Q factor, this bit can be set to 0. In this case, jpeg module calculates BRC table in simple way, and Q factor is defined in "brc" register.

Bit4 SQ: use same quantization table

This bit is used in encoding or decoding video. In the first frame, BRC table will be calculated. In the following frames, BRC table will not be calculated if this bit is set.

Bit5 CH: use custom Huffman table

This bit is used in JPEG decode mode or display mode.
JPEG file uses user defined Huffman table whenever this bit is set to 0 or 1.

Bit6 AF: avi format

This bit is used in any mode. If JPEG file uses avi_format, this bit is set to 1. And the JPEG file means Motion JPEG.

JPEG Control Register

Register address	JPEG_BASE_ADDR+01H							
Bit number	7	6	5	4	3	2	1	0

Bit field name	HS	WH	WQ	RM	RD	SD	SE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0

Bit0 SE: Start jpeg encode

This bit is used in JPEG encode mode. After setting this bit, jpeg unit begins to generate header. After header is written to JPEG buffer, header done status is set to “jpeg_status” register by hardware. After this, host write data to “dct_data” register byte by byte and jpeg begin to perform encode function. This bit is clear by hardware

Bit1 SD: start decode

This bit is used in display mode and JPEG decode mode. After setting this bit, jpeg module begins to calculate quantization brc table. After this, jpeg module reads data from JPEG buffer and perform decode function. Before setting this bit, host must write Huffman table and quantization table to jpeg module. After decoding 8 lines or 16 lines, jpeg module stops decoding and waits for restart signal from host (JPEG decode mode) or 8-line buffer module (display mode). This bit is clear by hardware.

Bit2 RD: Restart decode

This bit is used in JPEG decode mode. After host read 8/16 lines data from 8-line buffers, host restart decode process. JPEG module continues to read data from JPEG buffer after get restart signal from host until finishing decoding 8/16 lines, and then waiting for another restart signal. This bit is clear by hardware.

Bit3 RM: Reset jpeg state machine to idle state

This bit can be used in every mode. When there is something wrong with jpeg module, host can set this bit. This bit is clear by hardware.

Bit4 WQ: Start writing quantization table

This bit is used in JPEG decode mode and display mode. Before setting start decode bit, host writes quantization table to jpeg module. This bit is used for resetting the table address to 0. It is clear by hardware.

Bit5 WH: Start writing Huffman table

This bit is used in JPEG decode mode and display mode. Before setting start decode bit, host writes Huffman table to jpeg module. This bit is used for resetting the writing status to the beginning. It is clear by hardware.

Bit7-6 HS: Huffman table select

These bits are used in Huffman table selecting before custom Huffman table in JPEG decode mode and display mode.

huff_sel[1:0]=2’b00 : y_dc Huffman table select

huff_sel[1:0]=2’b01 : y_ac Huffman table select

huff_sel[1:0]=2’b10 : c_dc Huffman table select

huff_sel[1:0]=2'b11 : c_cc Huffman table select

JPEG status Register

Register address	JPEG_BASE_ADDR+02H							
Bit number	7	6	5	4	3	2	1	0
Bit field name		p_state			HD	ER	DD	ED
R/W		RO			RO	RO	RO	RO
Default					-	-	-	-

Bit0 ED: Encode done

This bit is used in capture mode and JPEG encode mode. When all the Huffman code is written to JPEG buffer, this bit is set by hardware. When jpeg begins encoding, this bit is reset to 0. In capture mode, vsync from 8-line buffer unit can reset this bit. In JPEG encode mode, SE bit in "jpeg_ctrl" register can reset this bit.

Bit1 DD: Decode done

This bit is used in JPEG decode mode and display mode. When all the data from decoder are written to 8-line buffers, this bit is set by hardware. When jpeg begins decoding, this bit is reset to 0. SD bit in "jpeg_ctrl" register can reset this bit.

Bit2 ER: Decode error flag

This bit is used in display and decode mode. It is set if there is something wrong with display and decode mode. It is reset by RM bit in "jpeg_ctrl" register.

Bit3 HD: Header done

This bit can be used in every mode. When display or decode mode, if setting CQ bit in "jpeg_mode" register it is set after user configure quantization tables and if setting SQ bit in "jpeg_mode" register it is set. When capture or encode mode, after header is written to JPEG buffer, this bit is set to 1 by hardware.

Bit6-4 p_state: JPEG control state machine

These bits can be used in every mode. It means current state of jpeg_ctrl state machine.

p_state[2:0] = 3'b000 IDLE state
p_state[2:0] = 3'b001 INIT state
p_state[2:0] = 3'b010 HEAD state
p_state[2:0] = 3'b011 FEEDTHROUGH_HEAD state
p_state[2:0] = 3'b100 ACT state
p_state[2:0] = 3'b101 DRAIN state

Bit Rate Control Register

Register address	JPEG_BASE_ADDR+03H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	ABRC	QF						

R/W	R/W	R/W
Default	1	78H

Bit6-0 QF: Initial Q factor.
When auto bit rate control is enabled, Q factor is used for compression. When auto bit rate control is disabled, it is an initial Q factor. In decode mode, Q factor is set when CQ in “jpeg_mode” register is 0.

Bit7 ABRC: Auto bit rate control enable
This bit is used in JPEG encode mode and capture mode. When this bit is set, initial Q factor uses bit6-0 of this register, and the Q factor of the following frames are calculated by bit rate control logic.

Target Compression Ratio Register

Register address	JPEG_BASE_ADDR+04H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	TCR							
R/W	R/W							
Default	29H							

Bit7-0 TCR: Target compression ratio
This bit is used when auto bit rate control is on. The actual ratio for compression is calculated by: ratio = (tcr-1)/4. For example, if the expected ratio is 10, this register will be set to 41(29H).

Q Factor Register

Register address	JPEG_BASE_ADDR+05H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved	QF						
R/W		RO						
Default		-						

Bit6-0 QF: Q factor of current image
This register is used when auto bit rate control is enabled. Q factor of each frame may be different. It is a read only register

Header Step Register

Register address	JPEG_BASE_ADDR+06H							
Bit number	7	6	5	4	3	2	1	0
Bit field name					Header_step			
R/W					R/W			
Default					03H			

Header step register is used in viewfinder, capture or encode mode. It is set by host. It can

control the output time from jpeg to 64K SRAM when jpeg_ctrl is on header state. The default value is 03H. It means that the output time is four jpeg clock.

Video Word Count Register

Register address	JPEG_BASE_ADDR+0CH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	VWC_0							
R/W	R/W							
Default	VIDEOWC _0							

Bit7-0 VWC_0: Video word count [31:24]

Register address	JPEG_BASE_ADDR+0DH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	VWC_1							
R/W	R/W							
Default	VIDEOWC _1							

Bit7-0 VWC_1: Video word count [23:16]

Register address	JPEG_BASE_ADDR+0EH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	VWC_2							
R/W	R/W							
Default	VIDEOWC _2							

Bit7-0 VWC_2: Video word count [15:8]

Register address	JPEG_BASE_ADDR+0FH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	VWC_3							
R/W	R/W							
Default	VIDEOWC _3							

Bit7-0 VWC_3: Video word count [7:0]

If the image size is 1280x1024 (1.3Meg):
VIDEOWC _0=8'h00, VIDEOWC _1=8'h09, VIDEOWC _2=8'h60, VIDEOWC _3=8'h00

Video word count register is used in each mode. The value set in the register is 1/4 of the

actual byte count of uncompressed image. For example, if the image size is 1280x960 and image format is YUV422, the image byte count before compression is 1280x960x2 = 2457600 bytes. In this case, the video word count register will be set to 2457600/4 = 96000 (H). If the image format is YUV420, the image byte count before compression is 1280x960x1.5 = 1843200 bytes. In this case, the register value will be 460800 (70800H).

Image Width Register

Register address	JPEG_BASE_ADDR+14H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved					Width_h		
R/W						R/W		
Default						WIDTH_H		

Bit1-0 width_h: Image width high byte [10:8]

Register address	JPEG_BASE_ADDR+15H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	width_l							
R/W	R/W							
Default	WIDTH_L							

Bit7-0 width_l: Image width low byte [7:0]

If the image size is 1280x960 (1.3Meg): WIDTH_H=3'b101, WIDTH_L=8'h00

Image width register is used in each mode. It is set by host. The maximum value for image width is 1280(1.3Meg).

Image Height Register

Register address	JPEG_BASE_ADDR+16H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved					Height_h		
R/W						R/W		
Default						03H		

Bit2-0 height_h: Image height high byte [10:8]

Register address	JPEG_BASE_ADDR+17H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	height_l							

R/W	R/W
Default	C0H

Bit7-0 height_l: Image height low byte [7:0]

If the image size is 1280x960 (1.3Meg): HEIGHT _H=3'b011, HEIGHT _L=8'hc0

Image height register is used in each mode. It is set by host. The maximum value for image height is 960(1.3Meg).

Frame Count Register

Register address	JPEG_BASE_ADDR+1AH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	FC_0							
R/W	RO							
Default	-							

Bit7-0 FC_0: Frame counter high byte [15:8]

Register address	JPEG_BASE_ADDR+1BH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	FC_0							
R/W	RO							
Default	-							

Bit7-0 FC_1: Frame counter high byte [7:0]

Frame count register is generated by hardware. The maximum value is CFFFH. After the counter reaches to CFFFH, it is reset to zero and then increased again.

DCT Data Register

Register address	JPEG_BASE_ADDR+1CH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Dct_data							
R/W	R/W							
Default	00H							

Bit7-0 dct_data: DCT data used in JPEG encode mode
This register is written by host byte by byte in JPEG encode mode. After header data is written to JPEG buffer, host begins to write DCT data to this register. The index of data is increased automatically. The data sequence must be block by block.

Quantization Table Register

Register address	JPEG_BASE_ADDR+1DH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Quan_t							
R/W	R/W							
Default	00H							

Bit7-0 Quan_t: Quantization table data used in decoder

This byte is used in JPEG decode mode and display mode. Before setting start decode bit, host writes quantization table to jpeg module. The index of table data is automatically increased. After setting WQ bit in “jpeg_ctrl” register, host writes data to this register byte by byte. The sequence of tables is quantization table 0 to 3. Not all the tables are needed. This is decided by jpeg header. And if possible, host needn’t write quantization table to jpeg module if the table is fixed.

Huffman Table Register

Register address	JPEG_BASE_ADDR+1EH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Huff_t							
R/W	R/W							
Default	00H							

Bit7-0 Huff_t: Huffman table data used in decoder

This byte is used in JPEG decode mode and display mode. Before setting start decode bit, host writes Huffman table to jpeg module. The index of table data is automatically increased. After setting WH bit in “jpeg_ctrl” register, host writes data to this register byte by byte. The sequence of tables is DC Luminance, DC Chrominance, AC Luminance and AC Chrominance.

Number of the Components Register

Register address	JPEG_BASE_ADDR+1FH							
Bit number	7	6	5	4	3	2	1	0
Bit field name		CN				QN		
R/W		R/W				R/W		
Default		03H				02H		

Bit2-0 QN: Number of quantization tables

This register indicates the number of quantization tables in the header. The value is from 1 to 4. It is used in JPEG decode mode and display mode. When calculating BRC RAM values, this register is used.

Bit6-4 CN: Number of components

This register indicates the number of components in the compressed image. The value is from 1 to 4. It is used in JPEG decode mode and display mode.

Component0 Parameter Register

Register address	JPEG_BASE_ADDR+20H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	TQ0		V0		Reserved	H0		
R/W	R/W		R/W			R/W		
Default	00		01		0	010		

Bit2-0 H1: Horizontal sampling factor of component 0
The range of H1 is from 1 to 4.

Bit5-4 V1: Vertical sampling ratio of component 0
The range of V1 is from 1 to 2.

Bit7-6 TQ1: Quantization table selector of component 0
The range of TQ1 is from 0 to 3

Register address	JPEG_BASE_ADDR+21H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved	N0					TA0	TD0
R/W		R/W					R/W	R/W
Default	0	00010					1	1

Bit0 TD0: Huffman table number of DC coefficient of component 0
The range of TD0 is from 0 to 1

Bit1 TA0: Huffman table number of AC coefficient of component 0
The range of TA0 is from 0 to 1

Bit6-2 N0: Block number of component 0 in one MCU
N0 is the result of H0*V0. The range is from 1 to 8

Component1 Parameter Register

Register address	JPEG_BASE_ADDR+22H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	TQ1		V1		Reserved	H1		
R/W	R/W		R/W			R/W		
Default	01		01		0	001		

Bit2-0 H1: Horizontal sampling factor of component 1
The range of H1 is from 1 to 4.

Bit5-4 V1: Vertical sampling ratio of component 1
The range of V1 is from 1 to 2.

Bit7-6 TQ1: Quantization table selector of component 1
The range of TQ1 is from 0 to 3

Register address	JPEG_BASE_ADDR+23H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved	N1					TA1	TD1
R/W		R/W					R/W	R/W
Default	0	00001					0	0

Bit0 TD1: Huffman table number of DC coefficient of component 1
The range of TD1 is from 0 to 1

Bit1 TA1: Huffman table number of AC coefficient of component 1
The range of TA1 is from 0 to 1

Bit6-2 N1: Block number of component 1 in one MCU
N1 is the result of H1*V1. The range is from 1 to 8

Component2 Parameter Register

Register address	JPEG_BASE_ADDR+24H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	TQ2		V2		Reserved	H2		
R/W	R/W		R/W			R/W		
Default	01		01		0	001		

Bit2-0 H2: Horizontal sampling factor of component 2
The range of H2 is from 1 to 4.

Bit5-4 V2: Vertical sampling ratio of component 2
The range of V2 is from 1 to 2.

Bit7-6 TQ2: Quantization table selector of component 2
The range of TQ2 is from 0 to 3

Register address	JPEG_BASE_ADDR+25H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved	N2					TA2	TD2
R/W		R/W					R/W	R/W
Default	0	00001					0	0

Bit0 TD2: Huffman table number of DC coefficient of component 2
The range of TD2 is from 0 to 1

Bit1 TA2: Huffman table number of AC coefficient of component 2
The range of TA2 is from0 to 1

Bit6-2 N2: Block number of component 2 in one MCU
N2 is the result of H2*V2. The range is from 1 to 8

Component3 Parameter Register

Register address	JPEG_BASE_ADDR+26H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	TQ3		V3		Reserved	H3		
R/W	R/W		R/W			R/W		
Default	00		00		0	000		

Bit2-0 H3: Horizontal sampling factor of component 3
The range of H3 is from 1 to 4.

Bit5-4 V3: Vertical sampling ratio of component 3
The range of V3 is from 1 to 2.

Bit7-6 TQ3: Quantization table selector of component 3
The range of TQ3 is from 0 to 3

Register address	JPEG_BASE_ADDR+27H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved	N3					TA3	TD3
R/W		R/W					R/W	R/W
Default	0	00000					0	0

Bit0 TD3: Huffman table number of DC coefficient of component 3
The range of TD3 is from 0 to 1

Bit1 TA3: Huffman table number of AC coefficient of component 3

Bit6-2 N3: Block number of component 3 in one MCU
N3 is the result of H3*V3. The range is from 1 to 8

Block Number Register

Register address	JPEG_BASE_ADDR+28H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved						num_blk_0	
R/W							R/W	
Default							NUM_BLK_0	

Bit0 num_blk_0: the high byte of the numbers of 8x8 blocks in 8 lines

Register address	JPEG_BASE_ADDR+29H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	num_blk_1							
R/W	R/W							
Default	NUM_BLK_1							

Bit6-0 num_blk_1: the low byte of the numbers of 8x8 blocks in 8 lines

If the image size is 1280x960 (1.3Meg): NUM_BLK_0=2'b01, NUM_BLK_1=8'h40

If the image size is 1280x960:

the color mode is YUV422, num_blk =1280*(2+1+1)/16 =320(140H).
the color mode is YUV420, num_blk= 1280*(4+1+1)/16 =480(1E0H).
the color mode is YUV411, num_blk= 1280*(4+1+1)/32 =240(F0H).
the color mode is YUV400, num_blk= 1280*(1+0+0)/8 =160(A0H).

JPEG Word Count Register

Register address	JPEG_BASE_ADDR+40H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	JPEG_WC_0							
R/W	RO							
Default	-							

Bit7-0 JPEG_WC_0: Huffman word count [31:24]

Register address	JPEG_BASE_ADDR+41H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	JPEG_WC_1							
R/W	RO							
Default	-							

Bit7-0 JPEG_WC_1: Huffman word count [23:16]

Register address	JPEG_BASE_ADDR+42H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	JPEG_WC_2							
R/W	RO							
Default	-							

Bit7-0 JPEG_WC_2: Huffman word count [15:8]

Register address	JPEG_BASE_ADDR+43H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	JPEG_WC_3							
R/W	RO							
Default	-							

Bit7-0 JPEG_WC_3: Huffman word count [7:0]

JPEG word count is 1/4 of the byte count of JPEG file. The JPEG word count equals to HWC*4 plus byte count of jpeg header.

Head Length Register

Register address	JPEG_BASE_ADDR+44H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Head_Length_h							
R/W	R/W							
Default	00H							

Bit7-0 head_length_h: jpeg file's header length[15:8]

Register address	JPEG_BASE_ADDR+45H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Head_Length_l							
R/W	R/W							
Default	9FH							

Bit7-0 head_length_l: jpeg file's header length [7:0]

The Head Length Regsiter is the length of the jpeg file subtract 1. And the default value of VIMICRO jpeg file is 9FH.

The Head Length Regsiter is only used in feedthrough_head enable. Which can be seen as below.

Feedthrough Head Register

Register address	JPEG_BASE_ADDR+46H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							Feedthrough_head
R/W								R/W
Default								00H

Bit0 feedthrough_head: Feedthrough head enable.

This bit is used in JPEG display mode or decode mode. When the bit is set 1, the jpeg

module can feed through the jpeg file's header length which is defined in the Head Length Register JPEG_BASE_ADDR+[44]H. Then the jpeg module start decoding the Huffman code of the jpeg file.

8.10 GE register

BitBLT control register

Register	GE_BASE_ADDR+03h								GE_BASE_ADDR+02h							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	GE state	Bitblt start	Line start	Bitblt state	Reserved				BitBLT color depth			Reserved			Pat Dep	Pat Mask
R/W	R	R/W		R	R -> 0h				R/W			R -> 0h			R/W	
De fault value	00h								00h							
Register	GE_BASE_ADDR+01h								GE_BASE_ADDR+00h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Tran Sel		Tran En	Src Mask	Src Dep	Src ext	Starting Select		Bit-wise operation selected							
R/W	R/W								R/W							
De fault value																

[31] Graphic engine status

This bit indicates whether the GE module is busy, if “1” the GE module is busy now, else the GE module is idle. If one operation is not completed, modify register will damage the current operation.

[30] Start BitBLT

When write “1” to this bit during the GE is idle, this will cause GE to start the bitblt operation. After the start, GE will clear this bit.

[29] Start line draw

When write “1” to this bit during the GE is idle, this will cause GE to start the line drawing operation. After the start, GE will clear this bit.

[28] BitBLT status

This bit indicates whether the BitBLT part of GE module is busy, if “1” the BitBLT is busy now, else the BitBLT is idle.

[27:24] Reserved

[23:21] BitBLT engine color depth

These 3 bits configure the color depth of the BitBLT engine, i.e., the color depth of the destination data.

Bit 23 22 21	BitBLT engine color depth selected
000	16-bit RGB565 direct color. This is the default after reset.
001	24-bit RGB888 direct color.
others	Reserved

The choice of color depth configures the BitBLT engine to work with 24, 18, 16, 12 bits per pixel. For the color depths with indirect color which are supported by VC0568, BitBLT operation doesn't work. This directly affects the number of bits of graphics data that the BitBLT engine will read and write for a given number of pixels. In case of monochrome source or pattern data, this setting directly affects the color depth into which such monochrome data will be converted during the color expansion process.

It is strongly recommend that, when possible, the color depth of the BitBLT engine be set to match the color depth to which the rest of graphics system has been set. However, if the rest of the graphics system has been set to a color depth that is not supported by the BitBLT engine, then it is strongly recommended that the BitBLT engine not be used. See the section on the BitBLT engine fore more information.

[20:18] Reserved

[17] Pattern Color Depth

0: Specifies that the pattern data is in color and therefore can have a color depth of 16 or 24 bits per pixel.
1: Specifies that the pattern data is monochrome and therefore has a color depth of 1 bit per pixel.

[16] Monochrome Pattern Write-Masking

Note: This bit applies only when the pattern data is monochrome (determined by bit 17 of this register).

This bit enables a form of per-pixel write-masking in which monochrome pattern data is used as a pixel mask that controls which pixels at the destination will be written to by the BitBLT engine.

0: This disables the use of monochrome pattern data as a write mask, allowing normal operation of the BitBLT engine with regard to the use of monochrome pattern data.
1: Wherever a bit in monochrome pattern data carries the value of 0 the bits of the corresponding pixel at the destination are NOT written, thereby preserving any data already carried by those bytes.

[15:14] Color Transparency Select

These 2 bits are used to select the type of color transparency to be performed. When color transparency is enabled by setting bit 13 of this register to 1, the color value carried within bits 23-0 of either Pattern/Source Expansion Background Color Register or Source Expansion Background Color Register is used as a key color to mask the writing of pixel data to the destination on a per-pixel basis. Before each pixel at the destination is written, a comparison is made between this key color and the destination data before or after bit-wise operation, and whether or not that given pixel at the destination will actually be written depends upon the result of that comparison.

Whether PAT_SRC_BG Register or SRC_BG Register is used to supply the key color depends on the setting of bit 10 of BitBlt_CTL Register since the same register that is used to supply the key color for color transparency also happens to be used to supply the background color for monochrome-to-color expansion. Also, depending on the type of color transparency selected, the other color value to which the key color is compared may be the color value resulting from the bit-wise operation selected via bits 7-0 of this register.

Bit 15 14	Form of Per-Pixel Color Comparison Selected
00	The color value carried by either PAT_SRC_BG or SRC_BG is compared to the color value resulting from the bit-wise operation being performed for the current pixel. If these two color values are NOT the same, then the bits for the current pixel at the destination will be written with the color value resulting from the bit-wise operation.
01	The color value carried by either PAT_SRC_BG or SRC_BG is compared to the color value already specified in the bits for the current pixel at the destination. If these two color values are NOT the same, then the bits for the current pixel at the destination will be written with the color value resulting from the bit-wise operation.
10	The color value carried by PAT_SRC_BG or SRC_BG is compared to the color value resulting from the bit-wise operation being performed for the current pixel. If these two color values ARE the same, then the bits for the current pixel at the destination will be written with the color value resulting from the bit-wise operation.
11	The color value carried by PAT_SRC_BG or SRC_BG is compared to the color value already specified in the bits for the current pixel at the destination. If these two color values ARE the same, then the bits for the current pixel at the destination will be written with the color value resulting from the bit-wise operation.

[13] Color Transparency Enable

This bit is used to enable or disable color transparency.

When color transparency is enabled, the color value carried within bits 23-0 of either PAT_SRC_BG or SRC_BG is used as a key color to mask the writing of pixel data to the destination on a per-pixel basis. Before each pixel at the destination is written, a comparison is made between this key color and another color, and whether or not that given pixel at the

destination will actually be written depends upon the result of that comparison.

Whether PAT_SRC_BG or SRC_BG is used to supply the key color depends on the setting of bit 10 of BitBlt_CTL Register since the same register that is used to supply the key color for color transparency also happens to be used to supply the background color for monochrome-to-color expansion. Also, depending on the type of color transparency selected via bits 15-14 of this register, the other color value to which the key color is compared may be the color value resulting from the bit-wise operation selected via bits 7-0 of this register.

0: Disables color transparency.
1: Enables color transparency.

[12] Monochrome Source Write-Masking

Note: This bit applies only when the source data is monochrome (determined by bit 11 of this register).

This bit enables a form of per-pixel write-masking in which monochrome source data is used as a pixel mask that controls which pixels at the destination will be written to by the BitBLT engine.

0: This disables the use of monochrome source data as a write mask, allowing normal operation of the BitBLT engine with regard to the use of monochrome source data.
1: Wherever a bit in monochrome source data carries the value of 0, the bits of the corresponding pixel at the destination are NOT written, thereby preserving any data already carried by those bytes.

[11] Source color depth

0: Specifies that the source data is in color, and therefore, can have a color depth of 12, 16, 18 or 24 bits per pixel.
1: Specifies that the source data is monochrome, and therefore, has a color depth of 1 bit per pixel.

[10] Source Extend

0: This causes the background and foreground colors for the color expansion of monochrome source data to be specified by PAT_SRC_BG and PAT_SRC_FG, respectively.
1: This causes the background and foreground colors for the color expansion of monochrome source data to be specified by SRC_BG and SRC_FG, respectively.

[9:8] Starting Point Select

These two bits are used to select which of the four corners to use as the starting point in reading and writing graphics data in a BitBLT operation. Normally, the upper left corner is used. However, situations involving an overlap of source and destination locations (this usually occurs when the source and destination locations are both on-screen) often require

the use of a different corner as a starting point. It should be remembered that the addresses specified for each piece of graphics data used in a BitBLT operation must point to the byte(s) corresponding to whichever pixel is at the selected starting point. If the starting point is changed, then these addresses must also be changed. See the chapter on the BitBLT engine for more information.

Bit 1	Bit 0	Corner Selected as the Starting Point
0	0	Upper Left Corner -- This is the default after reset.
0	1	Upper Right Corner
1	0	Bottom Left Corner
1	1	Bottom Right Corner

[7:0] Bit-Wise Operation Select

These 8 bits are meant to be programmed with an 8-bit code that selects which one of 256 possible bit-wise operations is to be performed by the BitBLT engine during a BitBLT operation. These 256 possible bit-wise operations and their corresponding 8-bit codes are designed to be compatible with the manner in which raster operations are specified in the standard BitBLT parameter block normally used in the Microsoft Windows environment, without translation. See the section on the BitBLT engine for more information.

Pattern Buffer Write Data Register

Register	GE_BASE_ADDR+07h								GE_BASE_ADDR+06h							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Pattern buffer write data															
R/W	R/W															
De fault value	0000h															
Register	GE_BASE_ADDR+05h								GE_BASE_ADDR+04h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Pattern buffer write data															
R/W	R/W															
De fault value	0000h															

[31:0] Pattern buffer write data

This register is used for storing the write data to pattern buffer, after write to this register, user should configure the write control register which is GE_BASE_ADDR+08h to the expected value, the write to the PAT_ADDR register which is GE_BASE_ADDR+09h the address of the pattern buffer. After writing to the PAT_ADDR register, the internal circuits will

move the patter buffer data to the pattern buffer according to the

Pattern Buffer Write Control Register

Register	GE_BASE_ADDR+08h				
Bit number	7..4	3	2	1	0
Bit field name	Reserved	Byte 3 write mask	Byte 2 write mask	Byte 1 write mask	Byte 0 write mask
R/W	N/A	R/W			
De fault value	N/A	0h			

[0] Byte 0 write mask

Write mask for the lowest byte of the 32 bits pattern buffer data, when “0”, the data in PAT_DATA[7:0] will be write to the pattern buffer when a write occurs; when “1”, the [7:0] of the corresponding address’s pattern data remains unchanged when a write occurs.

[1] Byte 1 write mask

Write mask for the byte 1 of the 32 bits pattern buffer data, when “0”, the data in PAT_DATA[15:8] will be write to the pattern buffer when a write occurs; when “1”, the [15:8] of the corresponding address’s pattern data remains unchanged when a write occurs.

[2] Byte 2 write mask

Write mask for the byte 2 of the 32 bits pattern buffer data, when “0”, the data in PAT_DATA[23:16] will be write to the pattern buffer when a write occurs; when “1”, the [23:16] of the corresponding address’s pattern data remains unchanged when a write occurs.

[3] Byte 3 write mask

Write mask for the most significant byte of the 32 bits pattern buffer data, when “0”, the data in PAT_DATA[31:24] will be write to the pattern buffer when a write occurs; when “1”, the [31:24] of the corresponding address’s pattern data remains unchanged when a write occurs.

Pattern Buffer Write Address Register

Register	GE_BASE_ADDR+09h						
Bit number	7:6	5	4	3	2	1	0
Bit field name	Reserved	Pattern buffer write address					
R/W	N/A	R/W					
De fault value	N/A	0h					

[5:0] Patter buffer write address

These bits are the address of the 64x32 pattern buffer SRAM when writing. After writing to these bits, one write will occur, and the address of the SRAM will be written to the data in

PAT_DATA, and the write mask is in PAT_WRT.

Pattern/Source Expansion Background Color & Transparency Key Register

Register	GE_BASE_ADDR+0dh								GE_BASE_ADDR+0ch							
Bit number									23	22	21	20	19	18	17	16
Bit field name									Pat/Src Expansion Background Color & Transparency Key Bits 23-16							
R/W									R/W							
De fault value									00h							
Register	GE_BASE_ADDR+0bh								GE_BASE_ADDR+0ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Pat/Src Expansion Background Color & Transparency Key Bits 15-0															
R/W	R/W															
De fault value	0000h															

[23:0] Pattern/Source Expansion Background Color Bits

These 24 bits are used to specify the foreground color for the color expansion of either monochrome pattern data only, or both monochrome pattern data and monochrome source data (depending on the setting of bit 10 of BitBlt_CTL register). When bit 10 of BitBlt_CTL register is set to 0 this register is used in the color expansion of monochrome pattern data only, SRC_FG is used to specify the background color for the color expansion of monochrome source data.

These 24 bits are also optionally used to specify the key color for whichever form of color transparency is selected via bits 15-14 of BITBLT_CTL register (depending on the setting of bit 10 of BitBlt_CTL register).

Whether bits 15-0 or 23-0 of this register are used in both the color expansion and color transparency processes depends upon the color depth to which the BitBLT engine has been set.

Pattern/Source Expansion Foreground Color

Register	GE_BASE_ADDR+0fh								GE_BASE_ADDR+0eh							
Bit number	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
Bit field name	Pat/Src Expansion Background Color Bits 23-8															
R/W	R/W															

De fault value	0000h								
Register	GE_BASE_ADDR+0dh								GE_BASE_ADDR+0ch
Bit number	7	6	5	4	3	2	1	0	
Bit field name	Pat/Src Expansion Background Color Bits7-0								
R/W	R/W								
De fault value	00h								

[23:0] Pattern/Source Expansion Background Color & Transparency Key Bits

These 24 bits are used to specify the background color for the color expansion of either monochrome pattern data only, or both monochrome pattern data and monochrome source data (depending on the setting of bit 10 of BitBlt_CTL register). When bit 10 of BitBlt_CTL register is set to 0 this register is used in the color expansion of monochrome pattern data only, SRC_BG register is used to specify the background color for the color expansion of monochrome source data.

Whether bits 15-0 or 23-0 of this register are used in both the color expansion and color transparency processes depends upon the color depth to which the BitBLT engine has been set.

Source Expansion Background Color & Transparency Key Register

Register	GE_BASE_ADDR+13h								GE_BASE_ADDR+12h							
Bit number	Reserved								23	22	21	20	19	18	17	16
Bit field name	n/a								Source Expansion Background Color & Transparency Key Bits 23-8							
R/W	R/W								R/W							
De fault value	R/W								00h							
Register	GE_BASE_ADDR+11h								GE_BASE_ADDR+10h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Source Expansion Background Color & Transparency Key Bits 15-0															
R/W	R/W															
De fault value	0000h															

[23:0] Source Expansion Background Color & Transparency Key Bits 23-0

These 24 bits are optionally used to specify the background color for the color expansion of monochrome source data (depending on the setting of bit 10 of BitBlt_CTL register). When

bit 10 of BitBlt_CTL register is set to 0 this register is used in the color expansion of monochrome source data, PAT_SRC_BG register is used to specify the background color for the color expansion of monochrome pattern data.

These 24 bits are also optionally used to specify the key color for whichever form of color transparency is selected by setting bits 15-14 of BITBLT_CTL register (depending on the setting of bit 10 of BitBlt_CTL register).

Whether bits 15-0 or 23-0 of this register are used in both the color expansion and color transparency processes depends upon the color depth to which the BitBLT engine has been set.

Source Expansion Foreground Color Register

Register	GE_BASE_ADDR+17h								GE_BASE_ADDR+16h							
Bit number									23	22	21	20	19	18	17	16
Bit field name	n/a								Source Expansion Background Color Bits 23-16							
R/W	n/a								R/W							
De fault value	n/a								00h							
Register	GE_BASE_ADDR+15h								GE_BASE_ADDR+14h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Source Expansion Background Color Bits 7-0															
R/W	R/W															
De fault value	0000h															

[23:0] Source Expansion Background Color & Transparency Key Bits 23-0

These 24 bits are optionally used to specify the foreground color for the color expansion of monochrome source data (depending on the setting of bit 10 of BitBlt_CTL register). When bit 10 of BitBlt_CTL register is set to 0 this register is used in the color expansion of monochrome source data, PAT_SRC_FG register is used to specify the background color for the color expansion of monochrome pattern data.

Whether bits 15-0 or 23-0 of this register are used in both the color expansion and color transparency processes depends upon the color depth to which the BitBLT engine has been set.

Destination Width & Height Register

Register	GE_BASE_ADDR+1bh	GE_BASE_ADDR+1ah

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				Destination Scanline Height											
R/W	n/a				R/W											
De fault value	n/a				000h											
Register	GE_BASE_ADDR+19h								GE_BASE_ADDR+18h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				Destination pixel Width											
R/W	n/a				R/W											
De fault value	n/a				000h											

[31:28] Reserved

These bits always return 0 when read.

[27:16] Destination Scanline Height

These 12 bits specify the height of the destination input and output data in terms of the number of scanlines for a BitBLT operation.

[15:12] Reserved

These bits always return 0 when read.

[11:0] Destination Byte Width

These 12 bits specify the width of the destination input and output data in terms of the number of pixels per scanline.

Source Address Register

Register	GE_BASE_ADDR+1fh								GE_BASE_ADDR+1eh							
Bit number	Reserved								23	22	21	20	19	18	17	16
Bit field name	n/a								Source Address Bits 23-16							
R/W	R/W								R/W							
De fault value	n/a								00h							
Register	GE_BASE_ADDR+1dh								GE_BASE_ADDR+1ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Source Address Bits 15-0															

R/W	R/W
De fault value	0000h

[23:0] Source Address

These bits are used to specify the starting address of the source data within the graphics buffer as an offset from the beginning of the graphics buffer to where the first byte of source data is located.

When the source data is not monochrome, then this address should be the just byte address of the start point of the source block. When the source data is monochrome, every scanline's data is double word continuous; this address should be the first double word's byte address.

Line color Register

Register	GE_BASE_ADDR+23h								GE_BASE_ADDR+22h							
Bit number	Reserved								23	22	21	20	19	18	17	16
Bit field name	n/a								Line color Bits 23-8							
R/W	R/W								R/W							
De fault value	n/a								00h							
Register	GE_BASE_ADDR+21h								GE_BASE_ADDR+20h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Line color Bits 15-0															
R/W	R/W															
De fault value	0000h															

[23:0] Pattern Address

These bits specify the line color when drawing line using Bresenham Algorithm.

Destination Address Register

Register	GE_BASE_ADDR+27h								GE_BASE_ADDR+26h							
Bit number	Reserved								23	22	21	20	19	18	17	16
Bit field name	n/a								Destination Address Bits 23-8							
R/W	n/a								R/W							
De fault value	n/a								00h							

Register	GE_BASE_ADDR+25h								GE_BASE_ADDR+24h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Destination Address Bits 15-0															
R/W	R/W															
De fault value	0000h															

[23:0] Destination Address

These bits are used to specify the starting address of the destination location within the graphics buffer .

When the start point is the left corner, then the address will be the start pixel's start byte address; when the start point is the right corner, then the address will be the end pixel's end byte address.

Source and Destination Span Register

Register	GE_BASE_ADDR+2bh								GE_BASE_ADDR+2ah							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				Destination span											
R/W	n/a				R/W											
De fault value	n/a				000h											
Register	GE_BASE_ADDR+29h								GE_BASE_ADDR+28h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				Source span											
R/W	n/a				R/W											
De fault value	n/a				000h											

[31:28] Reserved

These bits always return 0 when read.

[27:16] Destination Span

These 12 bits specify the span from the first byte in a scanline of destination data to the first byte in the next scanline. In other words, these bits specify the amount by which the destination address specified in DES_ADDR should be incremented after a scanline of destination data has been written to the destination in order to point to where the first byte in the next scanline of destination data should be written.

This span value is byte length.

[15:12] Reserved

These bits always return 0 when read.

[11:0] Source Span

These 12 bits are used only when color source data is being used as an input in a BitBLT operation. If the source data is monochrome, or no source data is to be used, then the BitBLT engine will ignore the value carried by these bits.

When color source data is read from the graphics buffer, these 12 bits specify the span from the first byte in a scanline of color source data to the first byte in the next scanline. In other words, these bits specify the amount by which the source address specified in SRC_ADDR should be incremented after a scanline of color source data has been read from the graphic buffer in order to point to where the first byte in the next scanline of color source data should be read.

This span value is byte length.

Coordinate Register of Left Upper Corner of the Clipping Rectangle

Register	GE_BASE_ADDR+2fh								GE_BASE_ADDR+2eh							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				Y-coordinate of left upper corner of the clipping rectangle											
R/W	n/a				R/W											
De fault value	n/a				000h											
Register	GE_BASE_ADDR+2dh								GE_BASE_ADDR+2ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				X-coordinate of left upper corner of the clipping rectangle											
R/W	n/a				R/W											
De fault value	n/a				000h											

[31:28] Reserved

These bits always return 0 when read.

[27:16] Y-coordinate of left upper corner of the clipping rectangle

These 12 bits specify the Y-coordinate of left upper corner of the clipping rectangle.

[15:12] Reserved

These bits always return 0 when read.

[11:0] X-coordinate of left upper corner of the clipping rectangle

These 12 bits specify the X-coordinate of left upper corner of the clipping rectangle.

Coordinate Register of Right Bottom Corner of the Clipping Rectangle

Register	GE_BASE_ADDR+33h								GE_BASE_ADDR+32h							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				Y-coordinate of right bottom corner of the clipping rectangle											
R/W	n/a				R/W											
De fault value	n/a				fffh											
Register	GE_BASE_ADDR+31h								GE_BASE_ADDR+30h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				X-coordinate of right bottom corner of the clipping rectangle											
R/W	n/a				R/W											
De fault value	n/a				fffh											

[31:28] Reserved

These bits always return 0 when read.

[27:16] Y-coordinate of right bottom corner of the clipping rectangle

These 12 bits specify the Y-coordinate of right bottom corner of the clipping rectangle.

[15:12] Reserved

These bits always return 0 when read.

[11:0] X-coordinate of right bottom corner of the clipping rectangle

These 12 bits specify the X-coordinate of right bottom corner of the clipping rectangle.

Coordinate Register of Start point for Bresenham Line Draw

Register	GE_BASE_ADDR+37h								GE_BASE_ADDR+36h							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				Y-coordinate of start point of Bresenham line draw											
R/W	n/a				R/W											
De fault value	n/a				000h											
Register	GE_BASE_ADDR+35h								GE_BASE_ADDR+34h							

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				X-coordinate of start point of Bresenham line draw											
R/W	n/a				R/W											
De fault value	n/a				000h											

[31:28] Reserved

These bits always return 0 when read.

[27:16] Y-coordinate of start point of Bresenham Line Draw

These 12 bits specify the Y-coordinate of start point of Bresenham line draw.

[15:12] Reserved

These bits always return 0 when read.

[11:0] X-coordinate of start point of Bresenham line Draw

These 12 bits specify the X-coordinate of start point of Bresenham line draw.

Coordinate Register of End point for Bresenham Line Draw

Register	GE_BASE_ADDR+3bh								GE_BASE_ADDR+3ah							
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved				Y-coordinate of end point of Bresenham line draw											
R/W	n/a				R/W											
De fault value	n/a				000h											
Register	GE_BASE_ADDR+39h								GE_BASE_ADDR+38h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				X-coordinate of end point of Bresenham line draw											
R/W	n/a				R/W											
De fault value	n/a				000h											

[31:28] Reserved

These bits always return 0 when read.

[27:16] Y-coordinate of end point of Bresenham Line Draw

These 12 bits specify the Y-coordinate of end point of Bresenham line draw.

[15:12] Reserved

These bits always return 0 when read.

[11:0] X-coordinate of end point of Bresenham Line Draw

These 12 bits specify the X-coordinate of end point of Bresenham line draw.

8.11 CPM register

UMCTRL

Register	CPM_BASE_ADDR+01h								CPM_BASE_ADDR+00h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									RW							
De fault value									00							

The register is used to control the debug pin. The high 4bit assign the different modules

[7:4]

4’b0001 : open the cpm debug pin
4’b0010 : open the sif debug pin
4’b0011: open the isp debug pin
4’b0100: open the ipp debug pin
4’b0101: open the lcdc debug pin
4’b0110: open the lcdif debug pin
4’b0111: open the lbuf debug pin
4’b1000: open the jpeg debug pin
4’b1001: open the marb debug pin
4’b1010: open the ge debug pin

[3:0]

each unit will define

RSTCTRL1

Register	CPM_BASE_ADDR+03h								CPM_BASE_ADDR+02h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W	RW								RW							
De fault value									00							

This register is the software reset control register and can be cleared by the hardware reset.

[0] LCDIFRST bit
Software reset of LCD I/F unit

[1] LCDCRST bit
Software reset of LCDC unit

[2] MARBRST bit
Software reset of MARB codec unit

[3] GERST bit
Software reset GE unit

[4] JPEGRST bit
Software reset of JPEG unit

[5] LBUFRST bit

Software reset of LBUF unit

[6] IPPRST bit

Software reset of IPP unit.

[7] SIFRST bit

Software reset of SIF unit.

[8] ISPRST bit

Software reset of ISP unit.

[15:9] Reserved

RSTCTRL2

Register	CPM_BASE_ADDR+05h								CPM_BASE_ADDR+04h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W	RW								RW							
De fault value									00							

Description: Global software reset . can be cleared by the hardware reset.

[7]: GBLRST

CLKOFF

Register	CPM_BASE_ADDR+06h								CPM_BASE_ADDR+05h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W	RW								RW							
De fault value									00							

This register is used to control the module clock. If the bit of the module is set to ‘1’, then the clock of this module is gated off.

[11] SIFOFF

SIF unit clock off

[10] ISPOFF

ISP unit clock off

[9] IPPOFF

IPP unit clock off

[8] LBUFOFF

LBUF unit clock off

[7] JPEGOFF

JPEG unit clock off

[6] JPEGCOREOFF

JPEG core clock off

[5] JEOFF

GE unit clock off

[4] MARBOFF

MARB unit clock off

[3] MARB_FASTOFF

MARB unit fast clock off

[2] LCDCOFF

LCDC unit clock off

[1] LCDIFOFF

LCDIF unit clock off

[0] CPMOFF

CPM unit clock off

INTEN

Register	CPM_BASE_ADDR+0fh								CPM_BASE_ADDR+0eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W	RW								RW							
De fault value									ff							

Interrupt enable register. Enable interrupt request to first level flag, corresponding bit is “1”, “0” the request is masked.

[0] sif interrupt enable
[1] isp interrupt enable
[2] lbuf interrupt enable
[3] jpeg interrupt enable
[4] ge interrupt enable
[5] marb interrupt enable
[6] lcdc interrupt enable

INTCTRL

Register	CPM_BASE_ADDR+11h								CPM_BASE_ADDR+10h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W	RW								RW							
De fault value									00							

This register controls the interrupt of VC0568.

[7:3] INTWID

interrupt signal valid width

[2] DEGAF

Edge active fall (1)

[1] LVLAL

Level active low (1)

[0] TRIG

Trigger type, 1’b0 for edge trigger and 1’b1 for level trigger

INTSERV

Register	CPM_BASE_ADDR+13h								CPM_BASE_ADDR+12h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W	RW								RW							
De fault value									00							

This register is the interrupt serve flag, when corresponding bit is “1”, means the interrupt is under serving, “0” means interrupt service over.

[0] sif interrupt souce service
[1] isp interrupt souce service
[2] lbuf interrupt souce service
[3] jpeg interrupt souce service
[4] ge interrupt souce service
[5] marb interrupt souce service
[6] lcdc interrupt souce service

INTFLAG

Register	CPM_BASE_ADDR+15h								CPM_BASE_ADDR+14h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									R							
De fault value									00							

This register is the first level interrupt flag, cleared by set corresponding bit of INTSERV register.

[0] sif interrupt status flag
[1] isp interrupt status flag
[2] lbuf interrupt status flag
[3] jpeg interrupt status flag
[4] ge interrupt status flag
[5] marb interrupt status flag
[6] lcdc interrupt status flag

INTFLAG0

Register	CPM_BASE_ADDR+17h								CPM_BASE_ADDR+16h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									R							
De fault value									00							

The second level interrupt status flag
[0] sif_cpm_vsync_int status flag
[1] sif_cpm_frameend_int status flag

INTFLAG1

Register	CPM_BASE_ADDR+19h								CPM_BASE_ADDR+18h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									R							
De fault value									00							

The second level interrupt status flag
[1] y_min_int
[2] his_int
[3] antif_int
[4] auto_foc_int

INTFLAG2

Register	CPM_BASE_ADDR+1bh								CPM_BASE_ADDR+1ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									R							
De fault value									00							

The second level interrupt status flag
[0] lbuf_cpm_read_int status flag

INTFLAG3

Register	CPM_BASE_ADDR+1dh								CPM_BASE_ADDR+1ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									R							
De fault value									00							

The second level interrupt status flag
[0] jpeg_cpm_enc_done_int status flag
[1] jpeg_cpm_dec_done_int status flag
[2] jpeg_cpm_dec_err_int status flag

INTFLAG4

Register	CPM_BASE_ADDR+1fh								CPM_BASE_ADDR+1eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									R							
De fault value									00							

The second level interrupt status flag

[0] ge_cpm_cmd_done_int status flag

[1] ge_cpm_buf_done_int status flag

INTFLAG5

Register	CPM_BASE_ADDR+21h								CPM_BASE_ADDR+20h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									R							
De fault value									00							

The second level interrupt status flag

[0] marb_cpm_jbuf_fifocnt_int status flag

[1] marb_cpm_jbuf_intv_int status flag

[2] marb_cpm_jpeg_done_int status flag

[3] marb_cpm_jbuf_err_int status flag

[4] marb_cpm_tbuf_fifcnt_int status flag

[5] marb_cpm_tbuf_intv_int status flag

[6] marb_cpm_thumb_done_int status flag

[7] marb_cpm_thumb_err_int status flag

INTFLAG6

Register	CPM_BASE_ADDR+23h								CPM_BASE_ADDR+22h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									R							
De fault value									00							

Description: The second level interrupt status flag

[0] lcdc_cpm_layerb_done_int status flag

[1] lcdc_cpm_layerbg_done_int status flag

[2] lcdc_cpm_fifo_warning_int status flag

[3] lcdc_cpm_vb_timeout_int status flag

[4] lcdc_cpm_gb_timeout_int status flag

INTEN0 - INTEN6

Register	CPM_BASE_ADDR+25h-31h								CPM_BASE_ADDR+24h-30h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									R							
De fault value									ff							

High level enable 2nd level interrupt flag setting, low mask corresponding interrupt source.

INTEN0 : sif 2nd level interrupt enable
INTEN1 : isp 2nd level interrupt enable
INTEN2 : lbuf 2nd level interrupt enable
INTEN3 : jpeg 2nd level interrupt enable
INTEN4 : ge 2nd level interrupt enable
INTEN5 : marb 2nd level interrupt enable
INTEN6 : lcdc 2nd level interrupt enable

GPIO_CFG

Register	CPM_BASE_ADDR+3bh								CPM_BASE_ADDR+3ah							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									RW							
De fault value									ff							

This register is used to control the GPIO pins.

[7:5] Reserved
[4:0] GPIO configuration bits for 5 GPIO pins

GPIO_MODE

Register	CPM_BASE_ADDR+3dh								CPM_BASE_ADDR+3ch							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									RW							
De fault value									ff							

This register is mode control of the GPIO pins. One bit for one GPIO pin. If the bit is set to '1', then this pin is in PIO mode; otherwise open drain mode.

[7:5] Reserved
[4:0] GPIO mode bits for 5 GPIO pins

GPIO_DIR

Register	CPM_BASE_ADDR+3fh								CPM_BASE_ADDR+3eh							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit field name		
R/W		RW
De fault value		00

This register is directional control of the GPIO pins. One bit for one GPIO pin. If the bit is set to '1', then this pin is in output mode; otherwise input mode.

[7:5] Reserved

[4:0] GPIO direction control bits for 5 GPIO pins

GPIO_PO

Register	CPM_BASE_ADDR+41h								CPM_BASE_ADDR+40h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W									RW							
De fault value									00							

This register is data out of the GPIO pins. One bit for one GPIO pin.

[7:5] Reserved

[4:0] GPIO R/W value

CPM_SIF_FLASH

Register	CPM_BASE_ADDR+43h								CPM_BASE_ADDR+42h							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																
R/W															rw	
De fault value															0	

This register is configuration of SIF flash .

[1] set 1 .the sif_fl_charge send out by GPIO[4] pin

[0] set 1 the sif_fl_trigger send out by GPIO[3] pin

9. Electrical characteristics

9.1 Absolute Maximum Ratings

Permanent device damage may occur if absolute maximum ratings are exceeded, hence the maximum ratings must never be exceeded. Functional operation should be restricted to the conditions as detailed in recommended operating conditions. Exposure to the absolute maximum conditions for extended periods may affect device reliability.

PARAMETER	SYMBOL	MIN	MAX	UNIT
Input voltage	V_I	-0.5	4.6	V
Output voltage	V_O	-0.5	4.6	V
Pre-driver power supply voltage	-	-0.5	2.5	V
Post-driver power supply voltage	-	-0.5	4.6	V
Operation temperature	T_{OPT}	-40	125	℃
Storage temperature	T_{STG}	-65	150	℃

9.2 Recommended Operating Conditions for 3.3V IO Application

The recommended operating conditions are the recommended values for assuring normal logic operation. As long as the device is used within the recommended operating conditions, the electrical characteristics (DC and AC characteristics) described below are assured.

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT
Pre-driver supply voltage	VDD	1.62	1.8	1.98	V
I/O supply voltage	VDD33	2.6	3.3	3.63	V
Input high voltage	V_{IH}	2.0	-	3.6	V
Input low voltage	V_{IL}	-0.3	-	0.8	V
Threshold point	V_T	1.46	1.59	1.75	V
Schmitt trig low to high threshold point	V_{T+}	1.44	1.50	1.56	V
Schmitt trig high to low threshold point	V_{T-}	0.88	0.94	0.99	V
Junction temperature	T_J	0	25	125	^oC
Input leakage current	I_L	-	-	±10	uA
Tri-state output leakage current	I_{OZ}	-	-	±10	uA
Pull-up resistor	R_{PU}	50	65	100	Kohm
Pull-down resistor	R_{PD}	40	56	107	Kohm
Output low voltage @I_{OL}=4mA	V_{OL}	-	-	0.4	V
Output high voltage @I_{OH}=4mA	V_{OH}	2.4	-	-	V
Low level output current @V_{OL}=0.4V	I_{OL}	4.4	7.4	9.2	mA
High level output current @VOH=2.4V	I_{OH}	5.0	10.2	15.0	mA

*The value of the supply voltage is the typical case.

9.3 Recommended Operating Conditions for 1.8V IO Application

The recommended operating conditions are the recommended values for assuring normal logic operation. As long as the device is used within the recommended operating conditions, the electrical characteristics (DC and AC characteristics) described below are assured.

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT
Pre-driver supply voltage	VDD	1.62	1.8	1.98	V
I/O supply voltage	VDD33	1.62	1.8	1.98	V
Input high voltage	V_{IH}	0.65*VDD	-	VDD+0.3	V
Input low voltage	V_{IL}	-0.3	-	0.35*VDD	V
Threshold point	V_T	0.87	0.92	0.98	V
Schmitt trig low to high threshold	V_{T+}	0.95	0.99	1.00	V
Schmitt trig high to low threshold	V_{T-}	0.56	0.58	0.60	V
Junction temperature	T_J	0	25	125	^oC
Input leakage current	I_L	-	-	±10	uA
Tri-state output leakage current	I_{OZ}	-	-	±10	uA
Pull-up resistor	R_{PU}	94	148	261	Kohm
Pull-down resistor	R_{PD}	77	135	312	Kohm
Output low voltage @I_{OL}=4mA	V_{OL}	-	-	0.45	V
Output high voltage @I_{OH}=4mA	V_{OH}	VDD-0.45	-	-	V
Low level output current @V_{OL}=0.45V	I_{OL}	1.8	3.8	6.0	mA
High level output current @VOH=VDD33-0.45V	I_{OH}	1.8	3.1	4.5	mA

*The value of the supply voltage is the typical case.

9.4 Working Current

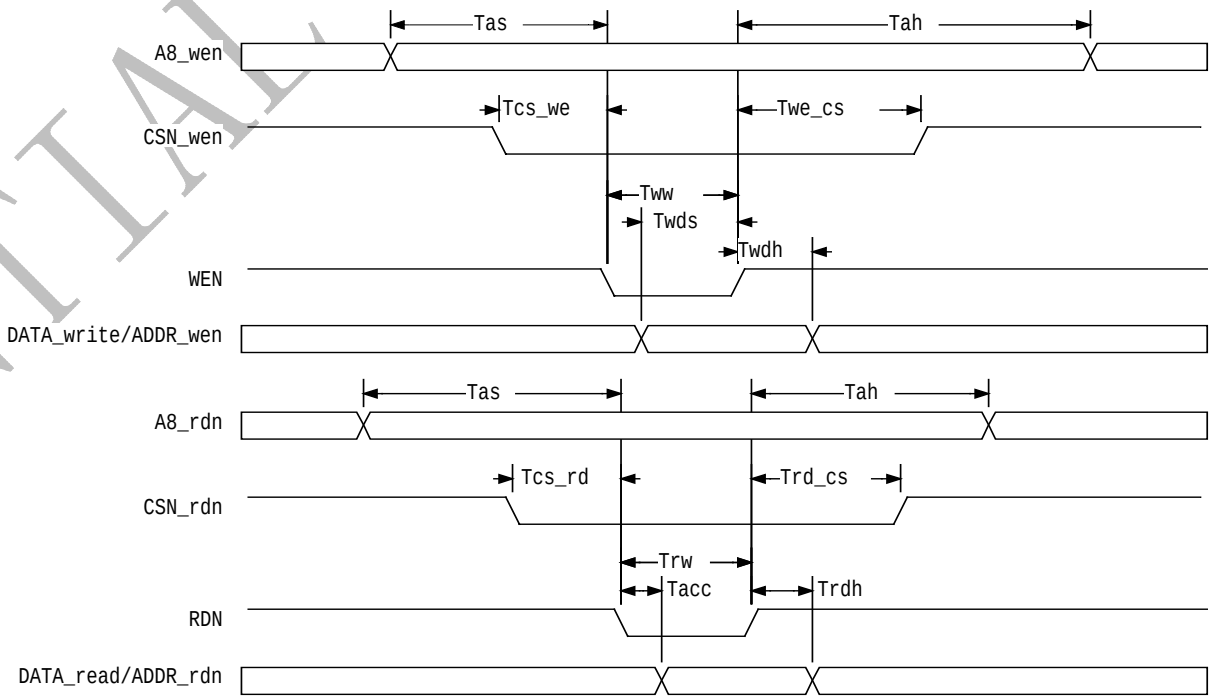
Current / Mode style	3.3V_IO (mA)	1.8V (mA)			P (mW)
		CORE	PLL	SUM	
Viewfinder Mode	2.8	17.94	5.836	23.776	52.04
Capture Mode	2.83	22.15	5.826	27.976	59.70
Display Mode	2.85	10.179	5.832	16.011	38.22
Bypass Mode	0.173	0.041	0.003	0.044	0.39

9.5 AC Characteristics

This section describes the VC0568 AC characteristics, which consist of output delays, input setup and hold times and all the interface timing.

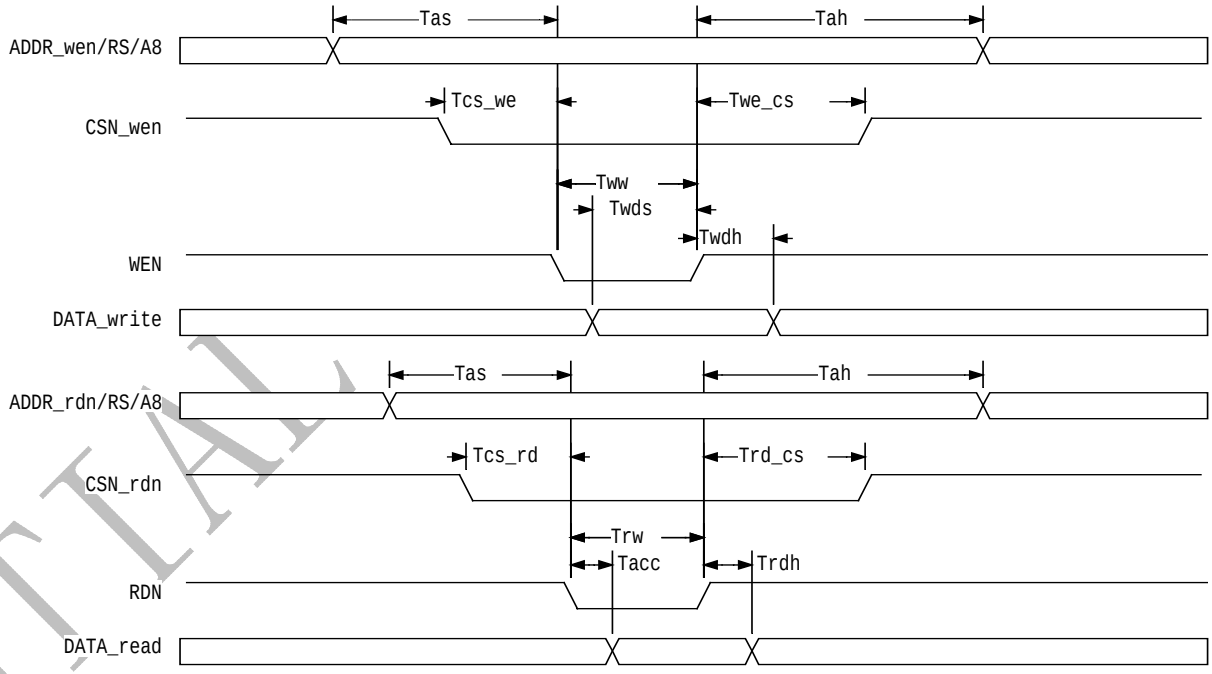
9.5.1 Host interface

9.5.1.1 Multiplex interface characteristics



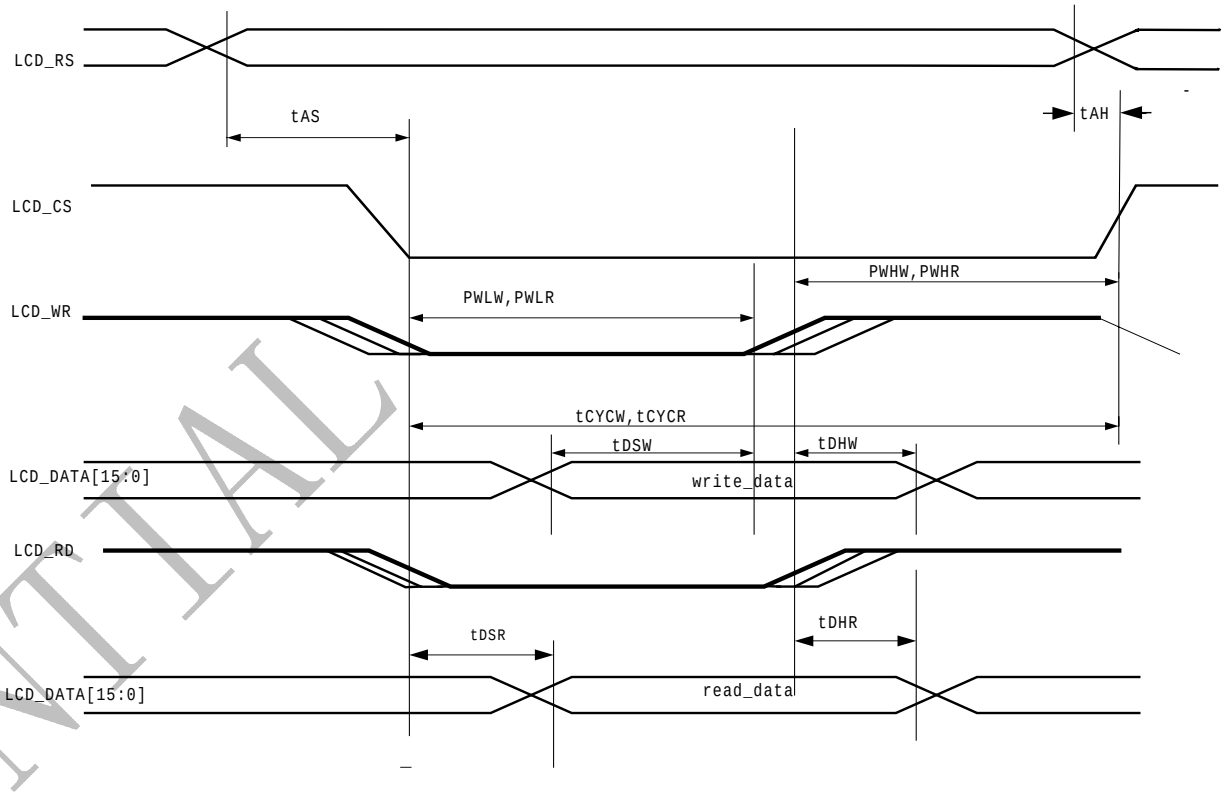
Item	symbol	Min	Typical	Max	Unit
Address setup time	Tas	0			ns
Address hold time	Tah	13.8	20		ns
Cs setup time at write status	Tcs_we	0			ns
Cs hold time at write status	Twe_cs	0			ns
Data setup time	Twds	12.2	24		ns
Data hold time	Twdh	8.8	10		ns
Write width	Tww	120			ns
Cs setup time at read status	Tcs_rd	0			ns
Cs hold time at read status	Trd_cs	0			ns
Read data access time	Tacc	120			ns
Read width	Trw	120			ns
Data hold time	Trdh	2.5			ns

9.5.1.2 Separate page interface characteristics



Item	symbol	Min	Typical	Max	Unit
Address setup time	Tas	0			ns
Address hold time	Tah	13.8	20		ns
Cs setup time at write status	Tcs_we	0			ns
Cs hold time at write status	Twe_cs	0			ns
Data setup time	Twds	12.2	24		ns
Data hold time	Twdh	8.8	10		ns
Write width	Tww	120			ns
Cs setup time at read status	Tcs_rd	0			ns
Cs hold time at read status	Trd_cs	0			ns
Read data access time	Tacc	120			ns
Read width	Trw	120			ns
Data hold time	Trdh	2.5			ns

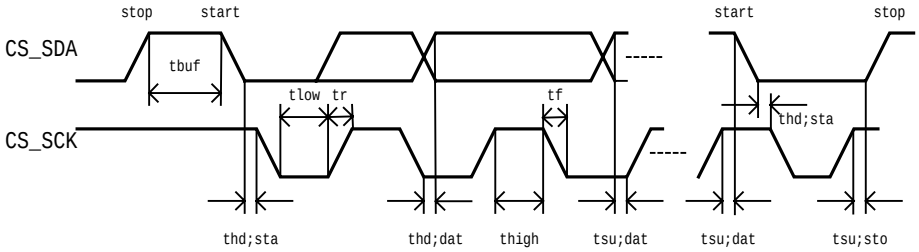
9.5.2 LCD panel interface



Parameter		sybmol	Min	Max	Unit
Bus cycle time	Write	t_{CYCW}	2		cycle *
	Read	t_{CYCR}	2		cycle
Write low-level pulse width		PW_{LW}	1		cycle
Read low-level pulse width		PW_{LR}	1		cycle
Write high-level pulse width		PW_{HW}	1		cycle
Read high-level pulse width		PW_{HR}	1		cycle
Setup time (LCD_RS to LCD_CS, LCD_WRN, LCD_RDN)		t_{AS}	0		ns
Hold time (LCD_RS to LCD_CS)		t_{AH}	0		ns
Write data setup time		t_{DSW}	1		cycle
Write data hold time		t_{DHW}	1		cycle
Read data setup time		t_{DSR}	0		ns
Read data hold time		t_{DHR}	0		ns

* The cycle width is according to the VC0568 system clock period

9.5.3 Sensor Interface

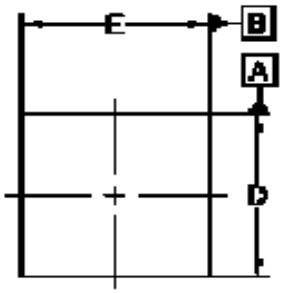


Parameter	sybmol	Min	Max	Unit
CS_SCK clock frequency	f_{sck}	0	375	kHz
Time that sensor serial bus must be free before a new transmission can start	t_{buf}	1.3	-	us
Hold time for a start	$t_{hd;sta}$	0.6	-	us
Low period of CS_SCK	t_{low}	1.3	-	us
High period of CS_SCK	t_{high}	1.3	-	us
Setup time for start	$t_{su;sta}$	0.6	-	us
Data hold time	$t_{hd;dat}$	0	-	us
Data setup time	$t_{su;dat}$	100	-	ns
Rise time of both CS_SDA and CS_SCK	t_r	-	1	us
Fall time of both CS_SDA and CS_SCK	t_f	-	300	ns
Setup time for stop	$t_{su;sto}$	0.6	-	us

Because this chip don’t use PCLK from sensor to sample sensor’s synchronous signal and data, the output from sensor must be generate in same clock domain and have same setup and hold timeing requirement .

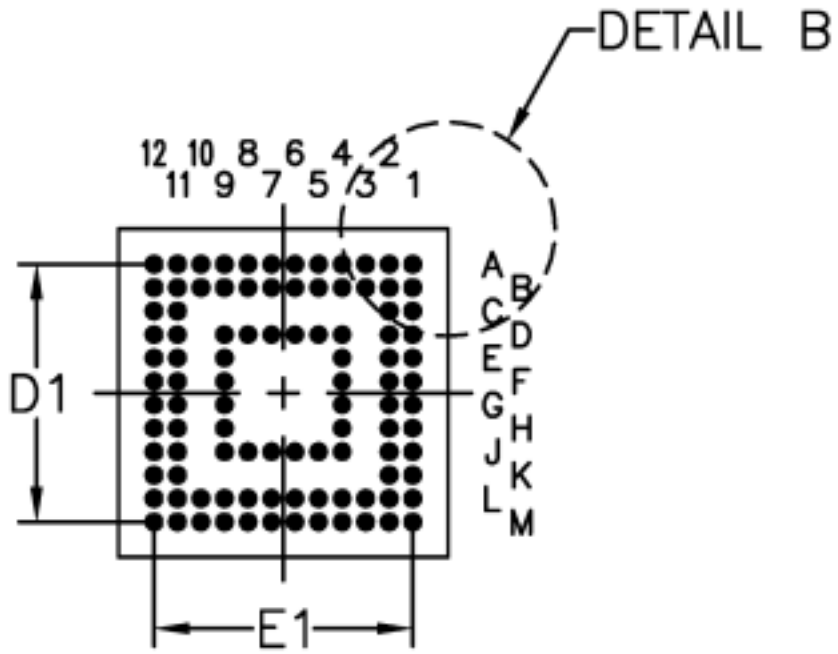
10. Mechanical dimensions

10.1 Top View



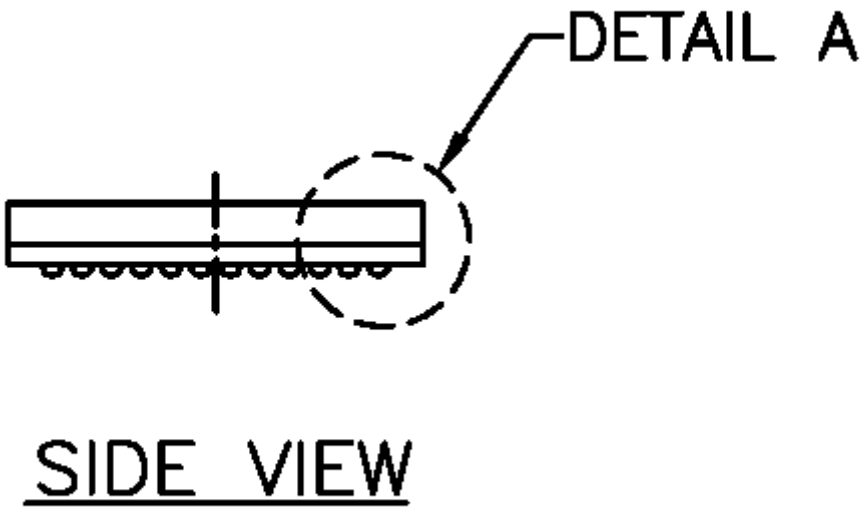
TOP VIEW

10.2 Bottom View

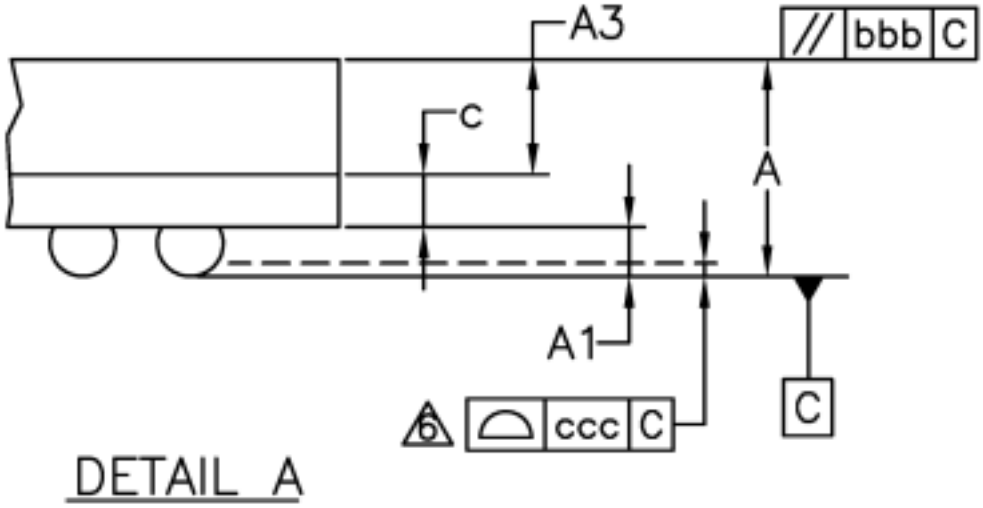


BOTTOM VIEW

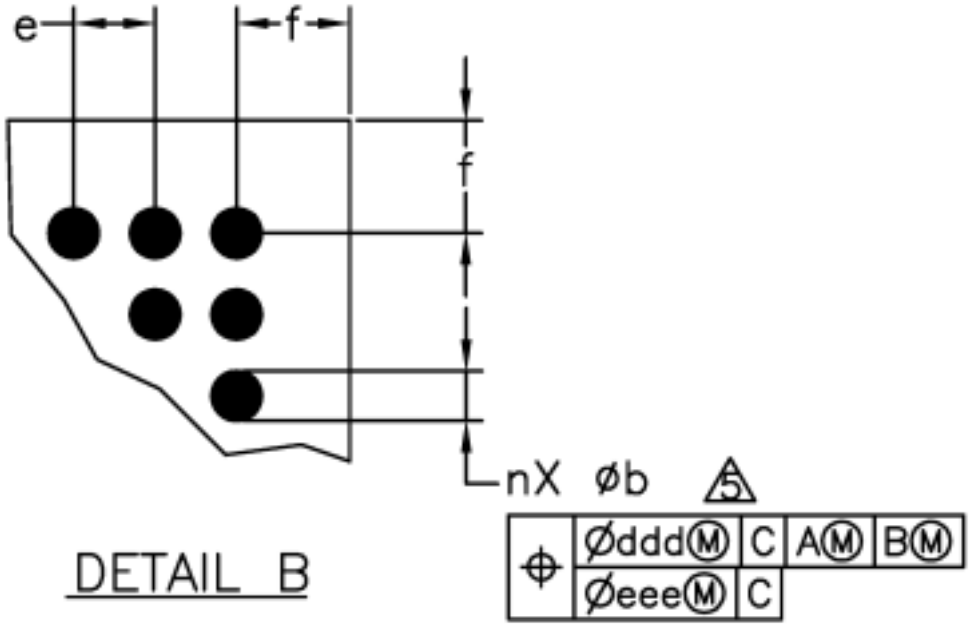
10.3 Side view



10.4 Delta A



10.5 Delta B



10.6 Dimensional Reference

DIMENSIONAL REFERENCES			
REF.	MIN.	NOM.	MAX.
A	0.94	1.09	1.24
A1	0.17	0.22	0.27
A3	0.50	0.55	0.60
D	6.80	7.00	7.20
D1		5.50 BSC.	
E	6.80	7.00	7.20
E1		5.50 BSC.	
b	0.25	0.30	0.35
bbb			0.20
c	0.27	0.32	0.37
ccc		0.08	
ddd			0.15
eee			0.05
e		0.50 BSC.	
f	0.65	0.75	0.85
M		12	
N		100	

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5-1994.
2. ALL DIMENSIONS ARE IN MILLIMETERS.
3. THE "e" REPRESENTS THE SOLDER BALL GRID PITCH.
4. "M" REPRESENTS THE BASIC SOLDER BALL MATRIX SIZE AND SYMBOL "N" IS THE ACTUAL NUMBER OF BALLS AFTER DEPOPULATING.
⚠ DIMENSION "b" IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.
⚠ PRIMARY DATUM C IS DEFINED BY THE CROWNS OF THE SOLDER BALLS.
7. PACKAGE SURFACE SHALL BE MATTE FINISH CHARMILLES 24 TO 27.
8. SUBSTRATE MATERIAL BASE IS BT RESIN.
9. THE OUTLINE DRAWING IS REFER TO JEDEC SPEC. MO-207 ISSUE G.
10. BALL MATRIX VARIATIONS CM-1.
11. FOR QUALIFICATION PURPOSE ONLY

11.Revision History

Version No.	Content of Revision	Corrector	Release Time
1.0	First version of VC0568 datasheet	Wally Li	2005-1-28
1.1	1) Update the IO voltage according to the test result; 2) Correct some error according to FAE/AE's suggestion	Wally Li	2005-3-11

12.Contact Information

Vimicro Corporation
15/F Shining Tower
No.35 Xueyuan Road
Haidian District, Beijing
P. R. China 100083
Tel: 86-10-68948888
Fax: 86-10-68944075
Web: http://www.vimicro.com

Vimicro Corporation – USA
1758 N. Shoreline
Mountain View, CA 94043
Tel: 1-650-966-1882
Fax: 1-650-966-1875