SSD1928 Application Note

CONTENTS

TABLES

FIGURES

1 Display Data Formats

The following diagrams show the display mode data formats.

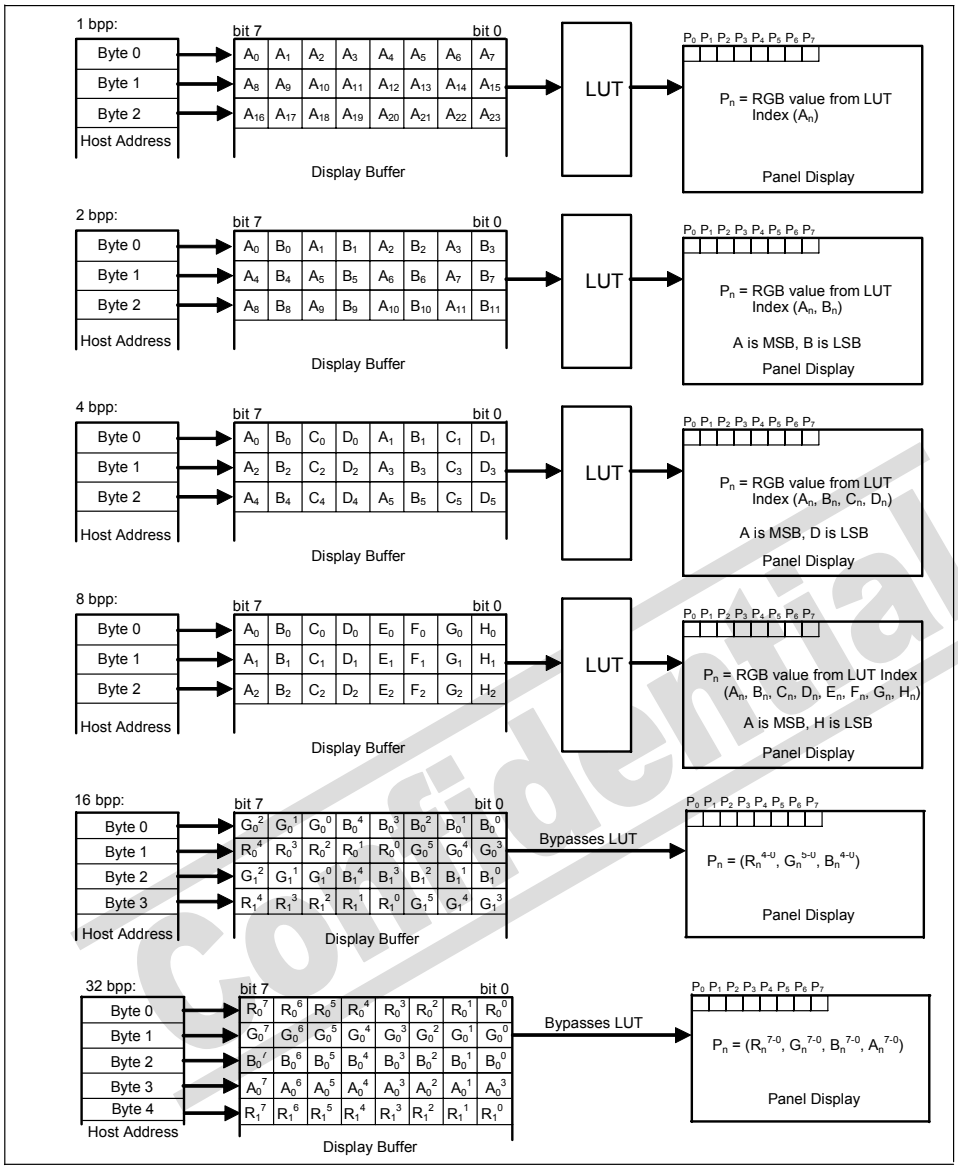


Figure 1-1 : 1/2/4/8/16/32 Bit-Per-Pixel Display Data Memory Organization

Note

1. For 16 bpp format, Rn, Gn, Bn represent the red, green, and blue color components.
2. For 32 bpp format, Rn, Gn, Bn, An represent the red, green, blue color and alpha blending components.

2 REGISTERS TABLE

This document discusses how and where to access the SSD1928 registers. It also provides detailed information about the layout and usage of each register.

2.1 Register Mapping

The SSD1928 registers are memory-mapped. When the system decodes the input pins as CS# = 0 and M/R# = 0, the registers may be accessed. The register space is decoded by A[18:0].

Unless specified otherwise, all register bits are set to 0 during power-on or software reset (REG[A2h] bit 0 = 1). All bits marked "0" should be programmed as zero. All bits marked "1" should be programmed as one.

Key :

RO : Read Only. Writes to these bits are ignored.
ROC : Read Only and initialized to zero at reset. Writes to these bits are ignored.
WO : Write Only
RW : Read / Write
RW1C : Read Only. Write 1 to clear status. Writing a 0 to these bits has no effect.
RWAC : Read-Write, automatic clear. Writing a 0 to these bits has no effect.
NA : Not Applicable
X : Don't care

2.1.1 Read-Only Configuration Registers

Debug Code Register **REG[00h]**

Bit	7	6	5	4	3	2	1	0
	Debug Code Bit 7	Debug Code Bit 6	Debug Code Bit 5	Debug Code Bit 4	Debug Code Bit 3	Debug Code Bit 2	Debug Code Bit 1	Debug Code Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	1	0	1	0	0	0

Bits 7-0 **Debug Code**
These bits show the dummy value for debug purpose. The readback code is 00101000.

Read Dummy Register **REG[01h]**

Bit	7	6	5	4	3	2	1	0
	Debug Code Bit 7	Debug Code Bit 6	Debug Code Bit 5	Debug Code Bit 4	Debug Code Bit 3	Debug Code Bit 2	Debug Code Bit 1	Debug Code Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	1	0	0	0	0	0	0	0

Bits 7-0 **Debug Code**
These bits show the dummy value for debug purpose. The readback code is 10000000.

Configuration Readback Register **REG[02h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	CNF6	Reserved	CNF4	CNF3	CNF2	CNF1	CNF0

		Status		Status	Status	Status	Status	Status
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	X	X	X	X	X	X	X	X

Bits 7-0 **CNF[6:0] Status**

These status bits return the status of the configuration pins CNF[6:0]. CNF[4:0] are latched at the rising edge of RESET# or software reset (REG[A2h] bit 0 = 1).
Bit 7 = Bit 6
Bit 5 = Bit 4

Product Code Register **REG[03h]**

Bit	7	6	5	4	3	2	1	0
	Product Code Bit 5	Product Code Bit 4	Product Code Bit 3	Product Code Bit 2	Product Code Bit 1	Product Code Bit 0	Revision Code Bit 1	Revision Code Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	1	0	0	0	0	0	0	0

Bits 7-2 **Product Code Bits [5:0]**

These are read-only bits that indicate the product code.
The product code of SSD1928 is 100000.

Bits 1-0 **Revision Code Bits [1:0]**

These are read-only bits that indicate the revision code which readback value is 00.

Memory status Register **REG[05h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	AD_MODE Status	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	1	X	X	0	0	0	0	0

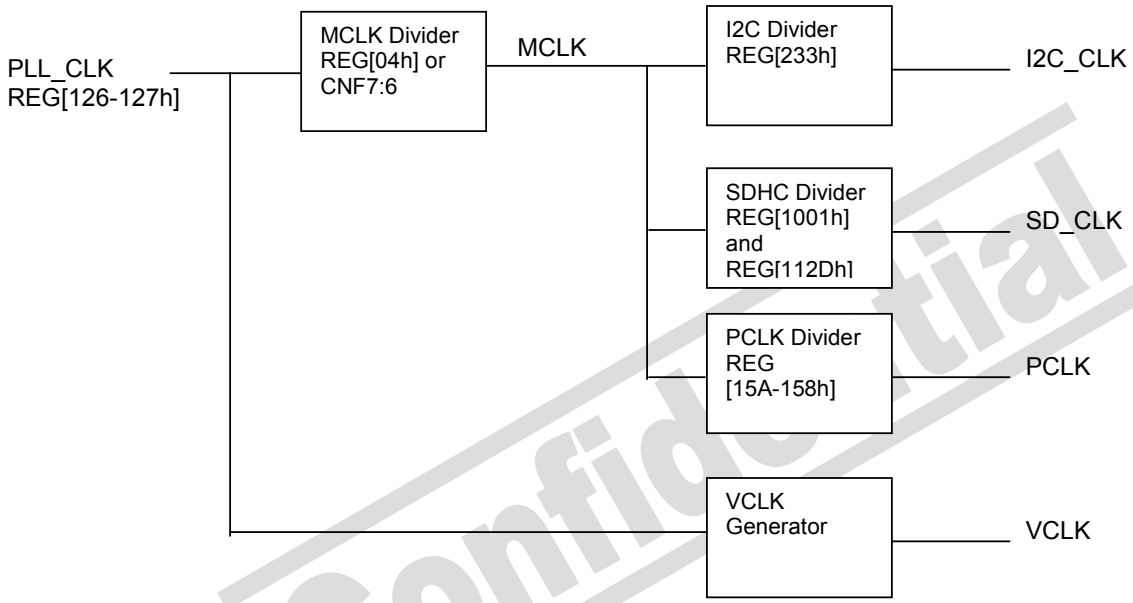
Bits 7, 4-0 **Reserved bits**

Bits 6-5 **AD_MODE Status**

These status bits return the status of the configuration pin AD_MODE which is latched at the rising edge of RESET# or software reset (REG[A2h] bit 0 = 1).
Bit 6 = Bit 5

2.1.2 Clock Configuration Registers

Figure 2-1: Clock configuration



PLL Clock Setting Register 0 **REG[126h]**

Bit	7	6	5	4	3	2	1	0
	PLL enable bit	Reserved	Reserved	N value bit 4	N value bit 3	N value bit 2	N value bit 1	N value bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	1	0	0	0	0

Bit 7 **PLL enable bit**
1 Enable PLL. PLL_DIS should tie to IOVSS and clock source should be provided through CLKI and CLKO(optional).
0 Disable PLL. PLL_DIS should tie to IOVDD and clock source should be provided through CLKI2.

Bits 6-5 **Reserved bits**

Bits 4-0 **N value bits [4:0]**
This register is used to program the N value for clock frequency

Note : The value of N should be greater than 1.

PLL Clock Setting Register 1 **REG[127h]**

Bit	7	6	5	4	3	2	1	0
	M value bit 7	M value bit 6	M value bit 5	M value bit 4	M value bit 3	M value bit 2	M value bit 1	M value bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	1	0	0	0	0	0	0	0

Bits 7-0 **M value bits [7:0]**
This register is used to program the M value for clock frequency

Note : The value of M should be greater than 1.

PLL Clock Setting Register 2 **REG[12Bh]**

Bit	7	6	5	4	3	2	1	0
	PLL config value bit 7	PLL config bit 6	PLL config bit 5	PLL config bit 4	PLL config bit 3	PLL config bit 2	PLL config bit 1	PLL config bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	1	0	0	0	1	1	1	0

Bits 7-0 **PLL config value bits [7:0]**
This register is used to config the PLL setting. This register should be programmed to 0xAE.
Bits 7:5 : control the internal reference clock value when PLL is not locked
Bits 4:0 : control the bias current of PLL (b'0x00000 is not allowed)

Program sequence (example input clock frequency = 2MHz) :
1. Write the N value (REG[126h] = 0x05)
2. Write the M value (REG[127h] = 0xC8)
3. Write the PLL Conf value (REG[12Bh] = 0xAE)
4. Enable the PLL (REG[126h] = 0x85)

Then, the PLL output clock frequency = Input clock frequency * (M / N) = 80MHz

Maximum output clock frequency = 85MHz

Table 2-1: Suggested M & N value

Input clock frequency	N value	M value
2MHz	0x05	0xC8
2.5MHz	0x06	0xC0
3MHz	0x08	0xD5
3.5MHz	0x09	0xCE
4Mz	0x0A	0xC8

Memory Clock Configuration Register **REG[04h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	MCLK Divide Select Bit 4	MCLK Divide Select Bit 3	MCLK Divide Select Bit 2	MCLK Divide Select Bit 1	MCLK Divide Select Bit 0
Type	RO	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5 **Reserved bits**

Bits 4-0 **MCLK Divide Select Bits [4:0]**
These bits determine the divide used to generate the Memory Clock (MCLK) from the PLL output frequency. Refer to REG[126h], REG[127h] for the PLL output frequency.
MCLK Frequency = PLL output frequency / (MCLK divide value + 1)

VCLK Divide Ratio Register **REG[06h]**

Bit	7	6	5	4	3	2	1	0
	VCLK Divide Ratio Bit 3	VCLK Divide Ratio Bit 2	VCLK Divide Ratio Bit 1	VCLK Divide Ratio Bit 0	VCLKO_S RC Divide Ratio Bit 3	VCLKO_S RC Divide Ratio Bit 2	VCLKO_S RC Divide Ratio Bit 1	VCLKO_S RC Divide Ratio Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-4 **VCLK Divide Ratio Bits [3:0]**
VCLK = VCLK_SRC / (VCLK Divide +1)

Bits 3-0 **VCLKO_SRC Divide Ratio Bits [3:0]**
VCLKO_SRC = PLL_CLK / (VCLKO_SRC Divide +1)

VCLK Select Register **REG[07h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	VCLK_SRC Select	VCLK Invert	VCLKO INV	Reserved	VCLK_SRC Enable	VCLKO output Enable	VCLKO_SRC Enable
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **Reserved bit**
Must program to 1

Bit 6 **VCLK_SRC select**
This bit to select the source of VCLK_SRC
1 : VCLK_SRC input from internal VCLKO_SRC
0 : VCLK_SRC input from VCLKI

Bit 5 VCLK INV select
1 : Invert the polarity of VCLK
0 : No change for polarity of VCLK

Bit 4 VCLKO_SRC invert select
1 : Input the polarity of VCLKO_SRC
0 : No change for polarity of VCLKO_SRC

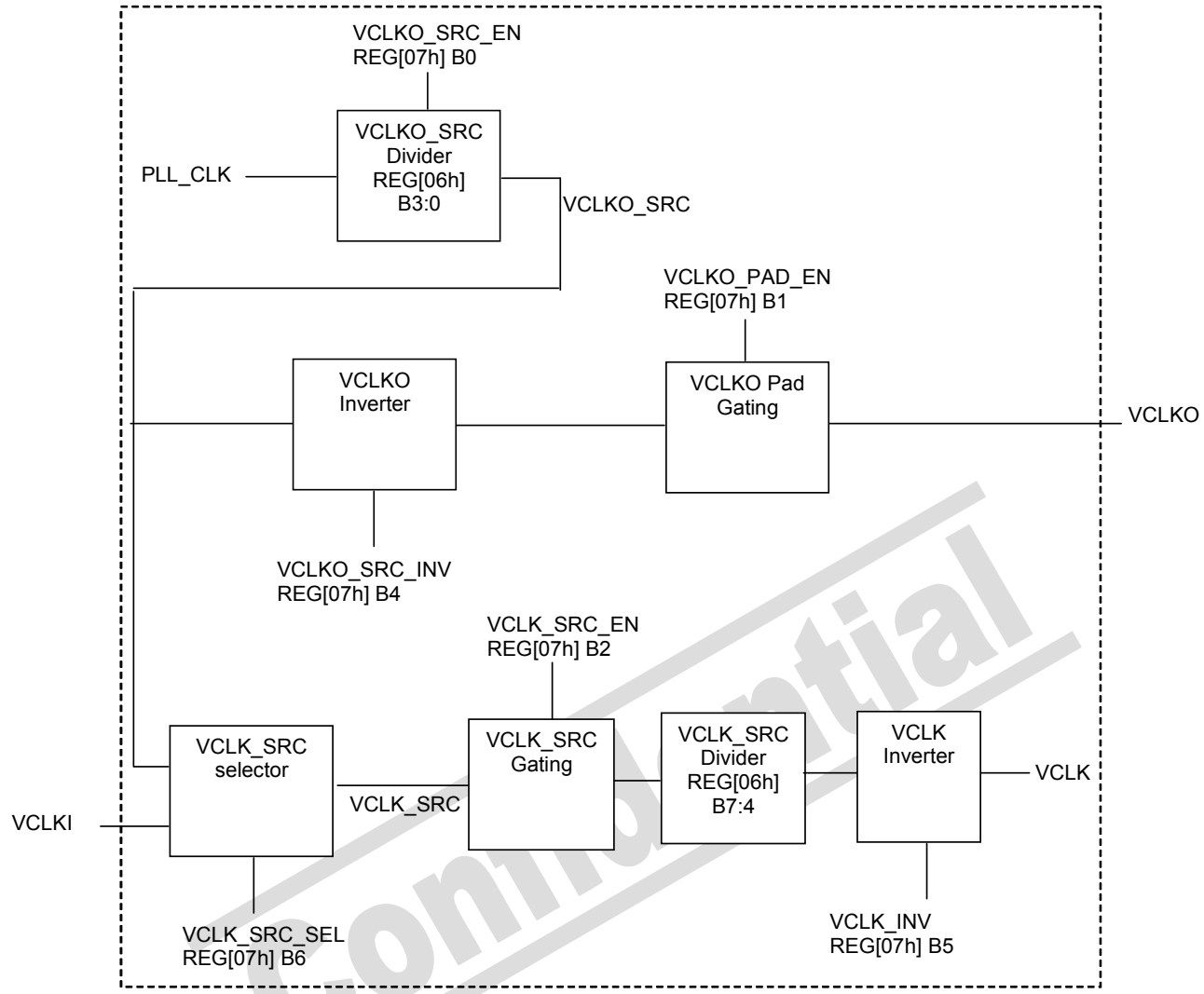
Bit 3 **Reserved bit**
Must program to 0

Bit 2 VCLK_SRC Enable
1 : Enable the VCLK_SRC input for VCLK
0 : Disable the VCLK_SRC input for VCLK

Bit 1 VCLKO Output Enable
1: Enable VCLKO_SRC output to VCLKO
0 : No output on VCLKO

Bit 0 VCLKO_SRC Enable
1: Enable the VCLKO_SRC which means the clock source of VCLK from internal PLL_CLK
0 :Disable the VCLKO_SRC which means the video source provide the VCLK

Figure 2-2: Block diagram for VCLK Generator



PCLK Frequency Ratio Register 0 **REG[158h]**

Bit	7	6	5	4	3	2	1	0
	PCLK Ratio bit 7	PCLK Ratio bit 6	PCLK Ratio bit 5	PCLK Ratio bit 4	PCLK Ratio bit 3	PCLK Ratio bit 2	PCLK Ratio bit 1	PCLK Ratio bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

PCLK Frequency Ratio Register 1 **REG[159h]**

Bit	7	6	5	4	3	2	1	0
	PCLK Ratio bit 15	PCLK Ratio bit 14	PCLK Ratio bit 13	PCLK Ratio bit 12	PCLK Ratio bit 11	PCLK Ratio bit 10	PCLK Ratio bit 9	PCLK Ratio bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

PCLK Frequency Ratio Register 2 **REG[15Ah]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	PCLK	PCLK	PCLK	PCLK

Bit	7	6	5	4	3	2	1	0
					Ratio bit 19	Ratio bit 18	Ratio bit 17	Ratio bit 16
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[15Ah] Bits 3-0,
REG[159h] Bits 7-0,
REG[158h] Bits 7-0

PCLK Frequency Ratio [19:0]
These bits determine the Frequency for PCLK.
PCLK frequency = MCLK frequency * (PCLK Frequency Ratio + 1) / (2^{20})

Note
(1) Bit[19:0] are used for non Serial-TFT panel type (REG[10h] bit 2:0 not equal to 010)
(2) Bit[17:0] are used for Serial-TFT panel type (REG[10h] bit 2:0 equal to 010)
(3) PCLK = MCLK (Bit[19:0] = 0xFFFFF) for smart panel interface.

2.1.3 Look-Up Table Registers

The following sections are the Look-up Table architecture which shows the display data output path only.

2.1.3.1 Monochrome Modes

The green Look-Up Table (LUT) is used for all monochrome modes.

2.1.3.1.1 1 Bit-per-pixel Monochrome Mode

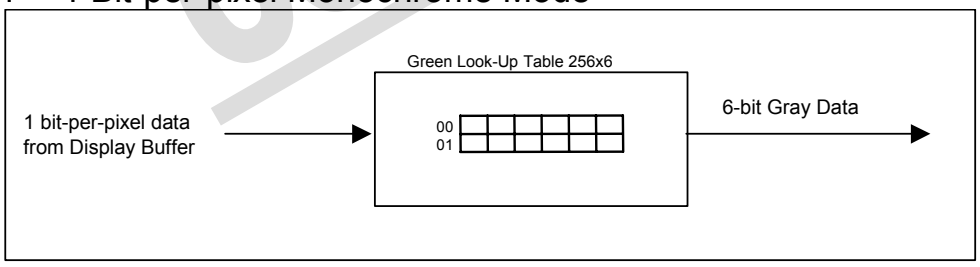


Figure 2-3 : 1 Bit-per-pixel Monochrome Mode Data Output Path

2.1.3.1.2 2 Bit-per-pixel Monochrome Mode

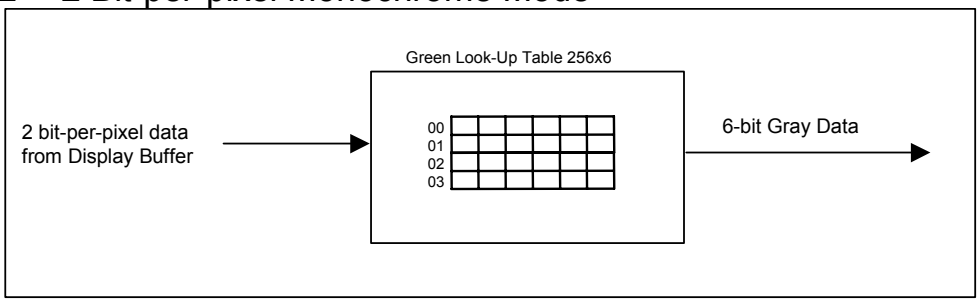


Figure 2-4 : 2 Bit-per-pixel Monochrome Mode Data Output Path

2.1.3.1.3 4 Bit-per-pixel Monochrome Mode

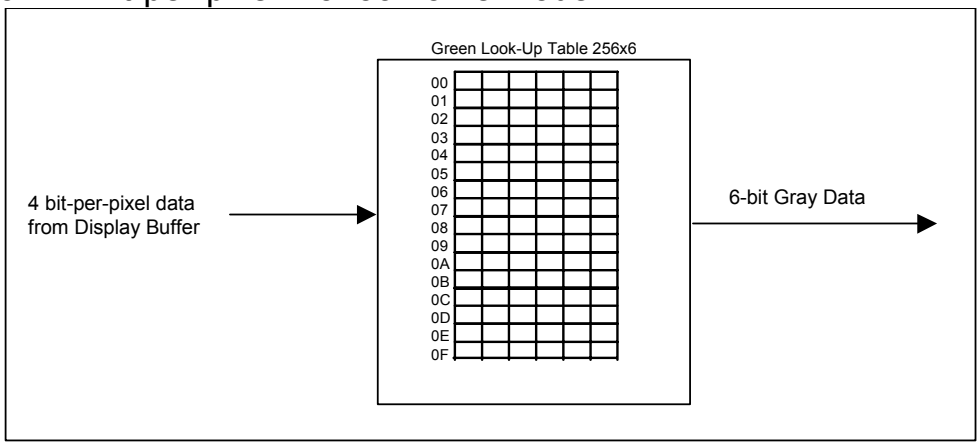


Figure 2-5 : 4 Bit-per-pixel Monochrome Mode Data Output Path

2.1.3.1.4 8 Bit-per-pixel Monochrome Mode

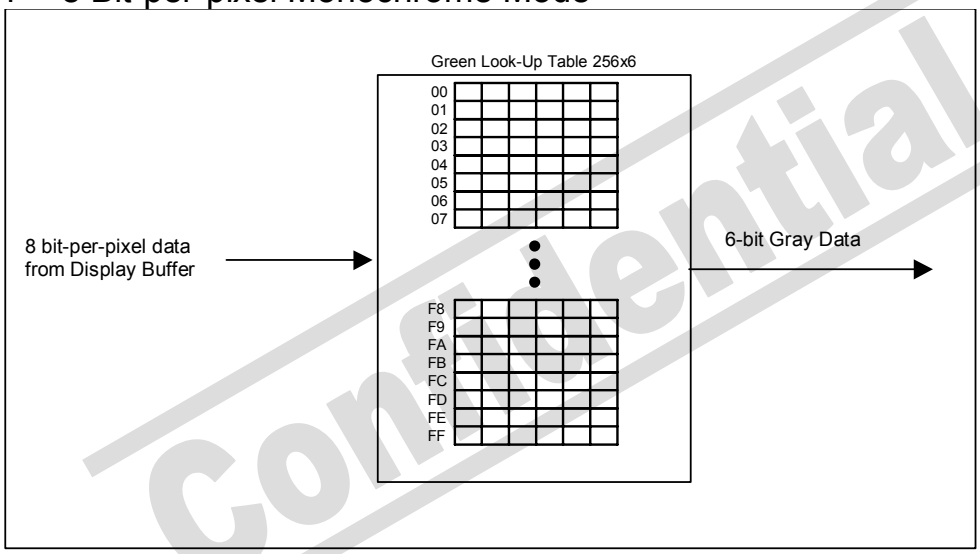


Figure 2-6 : 8 Bit-per-pixel Monochrome Mode Data Output Path

2.1.3.1.5 16/32 Bit-Per-Pixel Monochrome Mode

The LUT is bypassed and the green data is directly mapped for this color depth– See Figure 1-1 : 1/2/4/8/16/32 Bit-Per-Pixel Display Data Memory Organization.

2.1.3.2 Color Modes

2.1.3.2.1 1 Bit-Per-Pixel Color

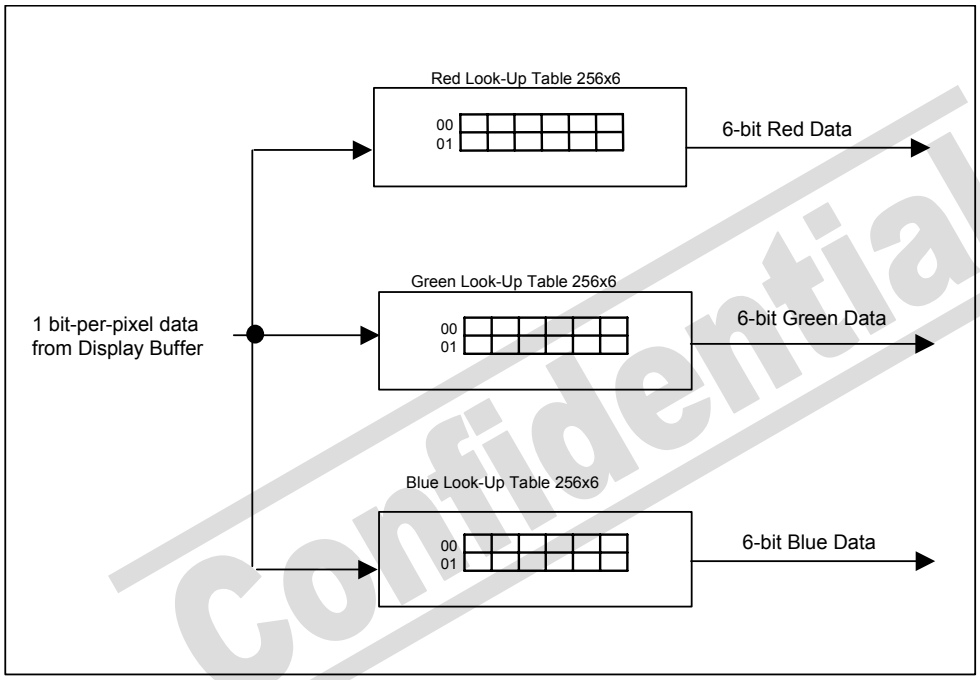


Figure 2-7 : 1 Bit-Per-Pixel Color Mode Data Output Path

2.1.3.2.2 2 Bit-Per-Pixel Color

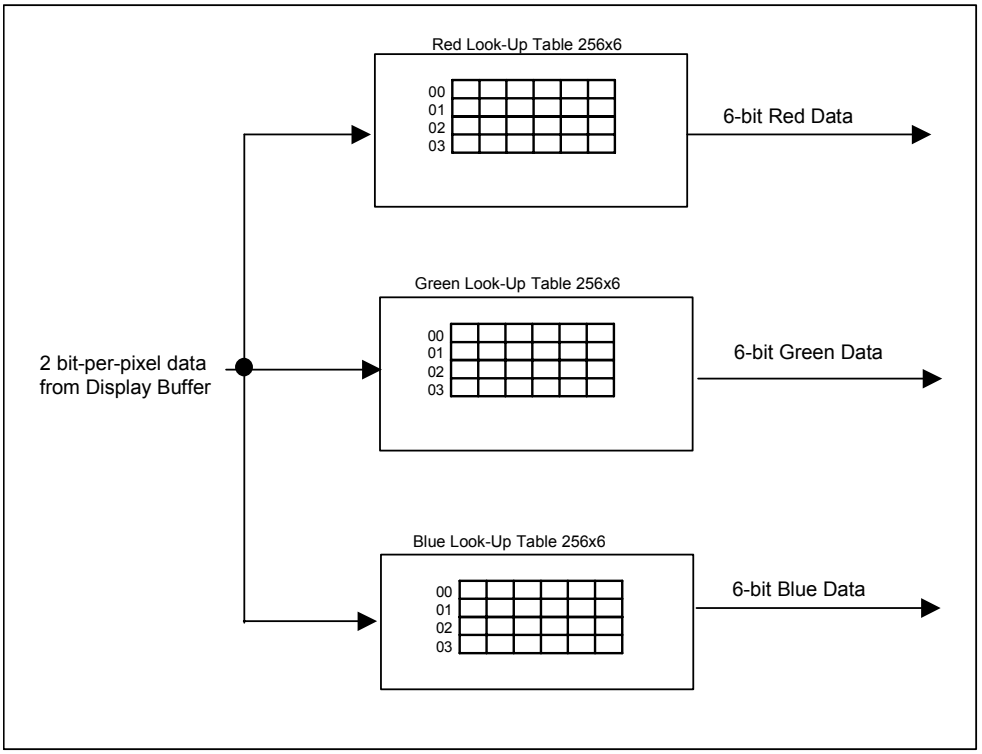


Figure 2-8 : 2 Bit-Per-Pixel Color Mode Data Output Path

2.1.3.2.3 4 Bit-Per-Pixel Color

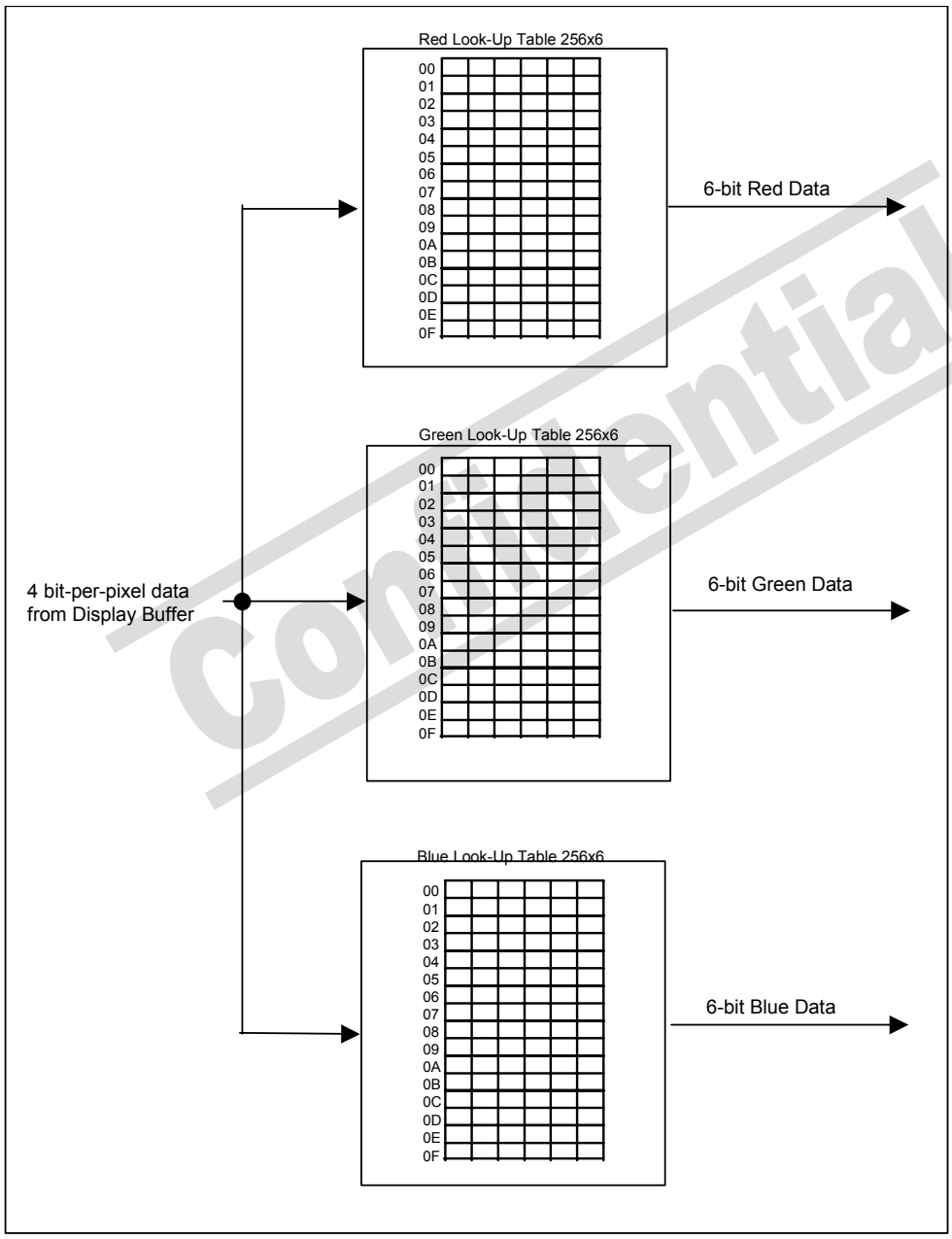


Figure 2-9 : 4 Bit-Per-Pixel Color Mode Data Output Path

2.1.3.2.4 8 Bit-per-pixel Color Mode

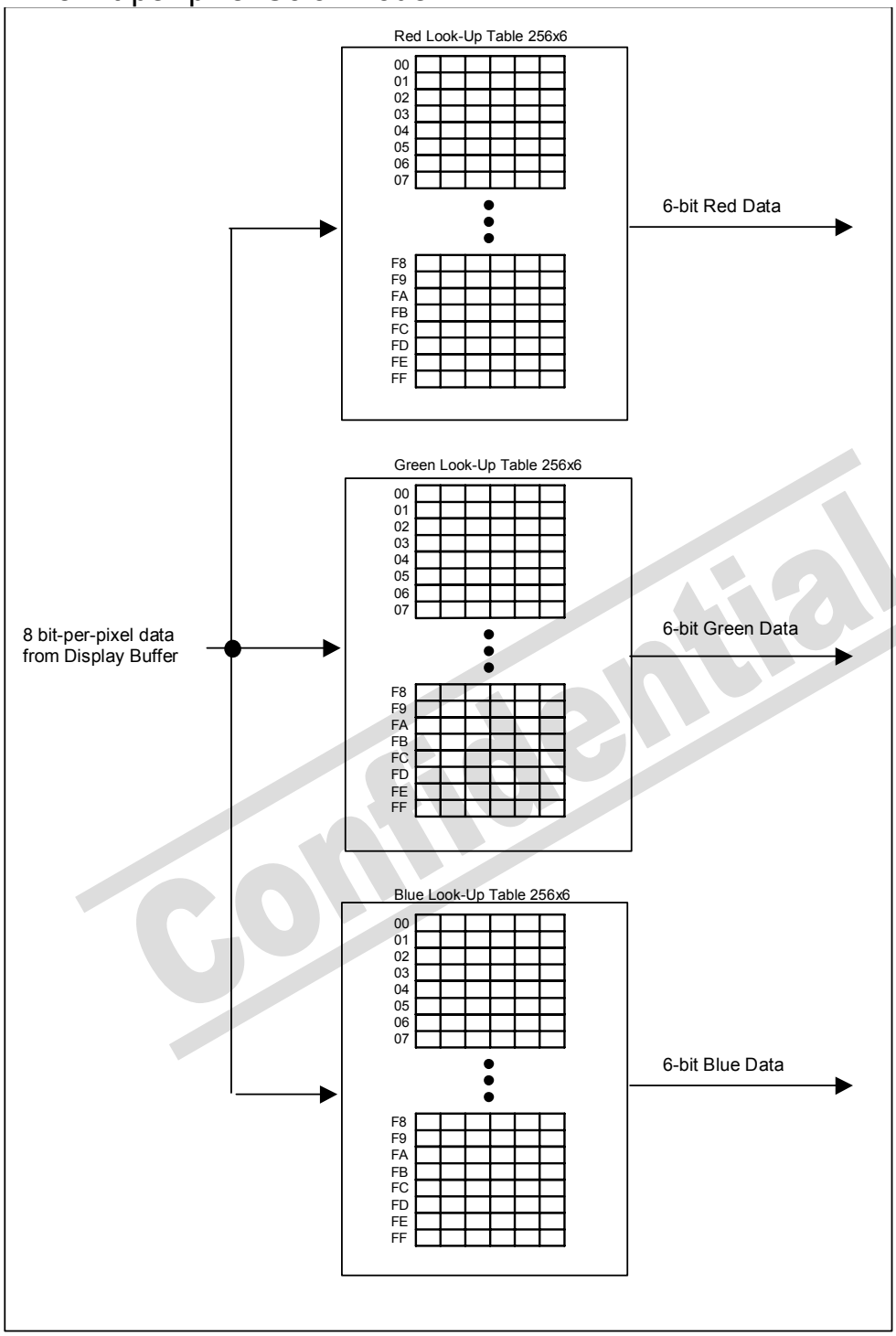


Figure 2-10 : 8 Bit-per-pixel Color Mode Data Output Path

2.1.3.2.5 16/32 Bit-Per-Pixel Color Mode

The LUT is bypassed and the color data is directly mapped for this color depth– See Figure 1-1 : 1/2/4/8/16/32 Bit-Per-Pixel Display Data Memory Organization.

Look-Up Table Blue Write Data Register **REG[08h]**

Bit	7	6	5	4	3	2	1	0
	LUT Blue Write Data Bit 7	LUT Blue Write Data Bit 6	LUT Blue Write Data Bit 5	LUT Blue Write Data Bit 4	LUT Blue Write Data Bit 3	LUT Blue Write Data Bit 2	LUT Blue Write Data Bit 1	LUT Blue Write Data Bit 0
Type	WO	WO	WO	WO	WO	WO	WO	WO
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LUT Blue Write Data Bits [7:0]**
This register contains the data to be written to the blue component of the Look-Up Table.

The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

Note
The LUT entry is updated only when the LUT Write Address Register (REG[0Bh]) is written.

Look-Up Table Green Write Data Register **REG[09h]**

Bit	7	6	5	4	3	2	1	0
	LUT Green Write Data Bit 7	LUT Green Write Data Bit 6	LUT Green Write Data Bit 5	LUT Green Write Data Bit 4	LUT Green Write Data Bit 3	LUT Green Write Data Bit 2	LUT Green Write Data Bit 1	LUT Green Write Data Bit 0
Type	WO	WO	WO	WO	WO	WO	WO	WO
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LUT Green Write Data Bits [7:0]**
This register contains the data to be written to the green component of the Look-Up Table.

The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

Note
The LUT entry is updated only when the LUT Write Address Register (REG[0Bh]) is written.

Look-Up Table Red Write Data Register **REG[0Ah]**

Bit	7	6	5	4	3	2	1	0
	LUT Red Write Data Bit 7	LUT Red Write Data Bit 6	LUT Red Write Data Bit 5	LUT Red Write Data Bit 4	LUT Red Write Data Bit 3	LUT Red Write Data Bit 3	LUT Red Write Data Bit 1	LUT Red Write Data Bit 0
Type	WO	WO	WO	WO	WO	WO	WO	WO
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LUT Red Write Data Bits [7:0]**
This register contains the data to be written to the red component of the Look-Up Table.

The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

Note
The LUT entry is updated only when the LUT Write Address Register (REG[0Bh]) is written.

Look-Up Table Write Address Register **REG[0Bh]**

Bit	7	6	5	4	3	2	1	0
	LUT Write Address Bit	LUT Write Address Bit	LUT Write Address Bit	LUT Write Address Bit	LUT Write Address Bit	LUT Write Address Bit	LUT Write Address Bit	LUT Write Address Bit

Bit	7	6	5	4	3	2	1	0
	7	6	5	4	3	2	1	0
Type	WO	WO	WO	WO	WO	WO	WO	WO
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LUT Write Address Bits [7:0]**

This register is a pointer to the Look-Up Table (LUT) which is used to write LUT data stored in REG[08h], REG[09h], and REG[0Ah]. **The data is updated to the LUT only with the completion of a write to this register**. This is a write-only register and returns 00h if read.

Note

The SSD1928 has three 256-entry, 8-bit-wide LUTs, one for each of red, green and blue (see Section "Look-Up Table Architecture" in datasheet).

Look-Up Table Blue Read Data Register **REG[0Ch]**

Bit	7	6	5	4	3	2	1	0
	LUT Blue Read Data Bit 7	LUT Blue Read Data Bit 6	LUT Blue Read Data Bit 5	LUT Blue Read Data Bit 4	LUT Blue Read Data Bit 3	LUT Blue Read Data Bit 2	LUT Blue Read Data Bit 1	LUT Blue Read Data Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LUT Blue Read Data Bits [7:0]**

This register contains the data from the blue component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]).

Note

This register is updated only when the LUT Read Address Register (REG[0Fh]) is written.

Look-Up Table Green Read Data Register **REG[0Dh]**

Bit	7	6	5	4	3	2	1	0
	LUT Green Read Data Bit 7	LUT Green Read Data Bit 6	LUT Green Read Data Bit 5	LUT Green Read Data Bit 4	LUT Green Read Data Bit 3	LUT Green Read Data Bit 2	LUT Green Read Data Bit 1	LUT Green Read Data Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LUT Green Read Data Bits [7:0]**

This register contains the data from the green component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]).

Note

This register is updated only when the LUT Read Address Register (REG[0Fh]) is written.

Look-Up Table Red Read Data Register **REG[0Eh]**

Bit	7	6	5	4	3	2	1	0
	LUT Red Read Data Bit 7	LUT Red Read Data Bit 6	LUT Red Read Data Bit 5	LUT Red Read Data Bit 4	LUT Red Read Data Bit 3	LUT Red Read Data Bit 2	LUT Red Read Data Bit 1	LUT Red Read Data Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LUT Red Read Data Bits [7:0]**

This register contains the data from the red component of the Look-Up Table. The LUT entry

read is controlled by the LUT Read Address Register (REG[0Fh]).

Note
This register is updated only when the LUT Read Address Register (REG[0Fh]) is written.

Look-Up Table Read Address Register **REG[0Fh]**

Bit	7	6	5	4	3	2	1	0
	LUT Read Address Bit 7	LUT Read Address Bit 6	LUT Read Address Bit 5	LUT Read Address Bit 4	LUT Read Address Bit 3	LUT Read Address Bit 2	LUT Read Address Bit 1	LUT Read Address Bit 0
Type	WO	WO	WO	WO	WO	WO	WO	WO
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LUT Read Address Bits [7:0]**
This register is a pointer to the Look-Up Table (LUT) which is used to read LUT data and store it in REG[0Ch], REG[0Dh], REG[0Eh]. **The data is read from the LUT only when a write to this register is completed**. This is a write-only register and returns 00h if read.

Note
The SSD1928 has three 256-entry, 8-bit-wide LUTs, one for each of red, green and blue (see Section "Look-Up Table Architecture" in datasheet).

2.1.4 Panel Configuration Registers

Figure 2-11: Panel Timing Parameters shows the timing parameters required to drive a flat panel display. Timing details for each supported panel types are provided in the remainder of this section.

Figure 2-11: Panel Timing Parameters

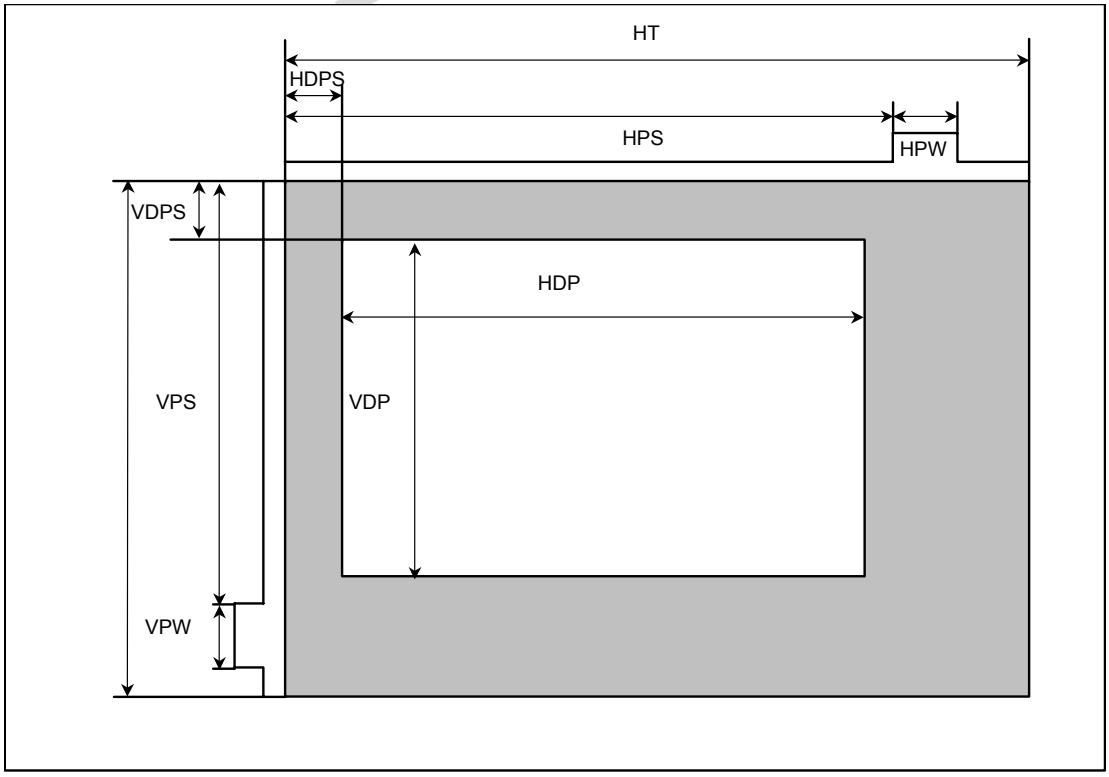


Table 2-2: Panel Timing Parameter Definition and Register Summary

Symbol	Description	Derived From	Units
HT	Horizontal Total	((REG[12h] bits 7-0) + 1) x 8 + (REG[13h] bits 2-0)	Ts [1]
HDP [2]	Horizontal Display Period [2]	((REG[14h] bits 6-0) + 1) x 8	
HDPS	Horizontal Display Period Start Position	((REG[17h]bits 2-0,REG[16h]bits7-0))	
HPS	LLINE Pulse Start Position	(REG[23h] bits 2-0, REG[22h] bits 7-0) + 1	
HPW	LLINE Pulse Width	(REG[20h] bits 6-0) + 1	
VT	Vertical Total	((REG[19h] bits 2-0, REG[18h] bits 7-0) + 1) x HT	Ts [1]
VDP [3]	Vertical Display Period [3]	((REG[1Dh] bits 1-0, REG[1Ch] bits 7-0) + 1) x HT	
VDPS	Vertical Display Period Start Position	(REG[1Fh] bits 2-0, REG[1Eh] bits 7-0) x HT	
VPS	LFRAME Pulse Start Position	(REG[27h] bits 2-0, REG[26h] bits 7-0) x HT + (REG[31h] bits 2-0, REG[30h] bits 7-0)	
VPW	LFRAME Pulse Width	((REG[24h] bits 2-0) + 1) x HT + (REG[35h] bits 2-0, REG[34h] bits 7-0) – (REG[31h] bits 2-0, REG[30h] bits 7-0)	

The following conditions must be fulfilled for all panel timings:

HDPS + HDP < HT

VDPS + VDP < VT

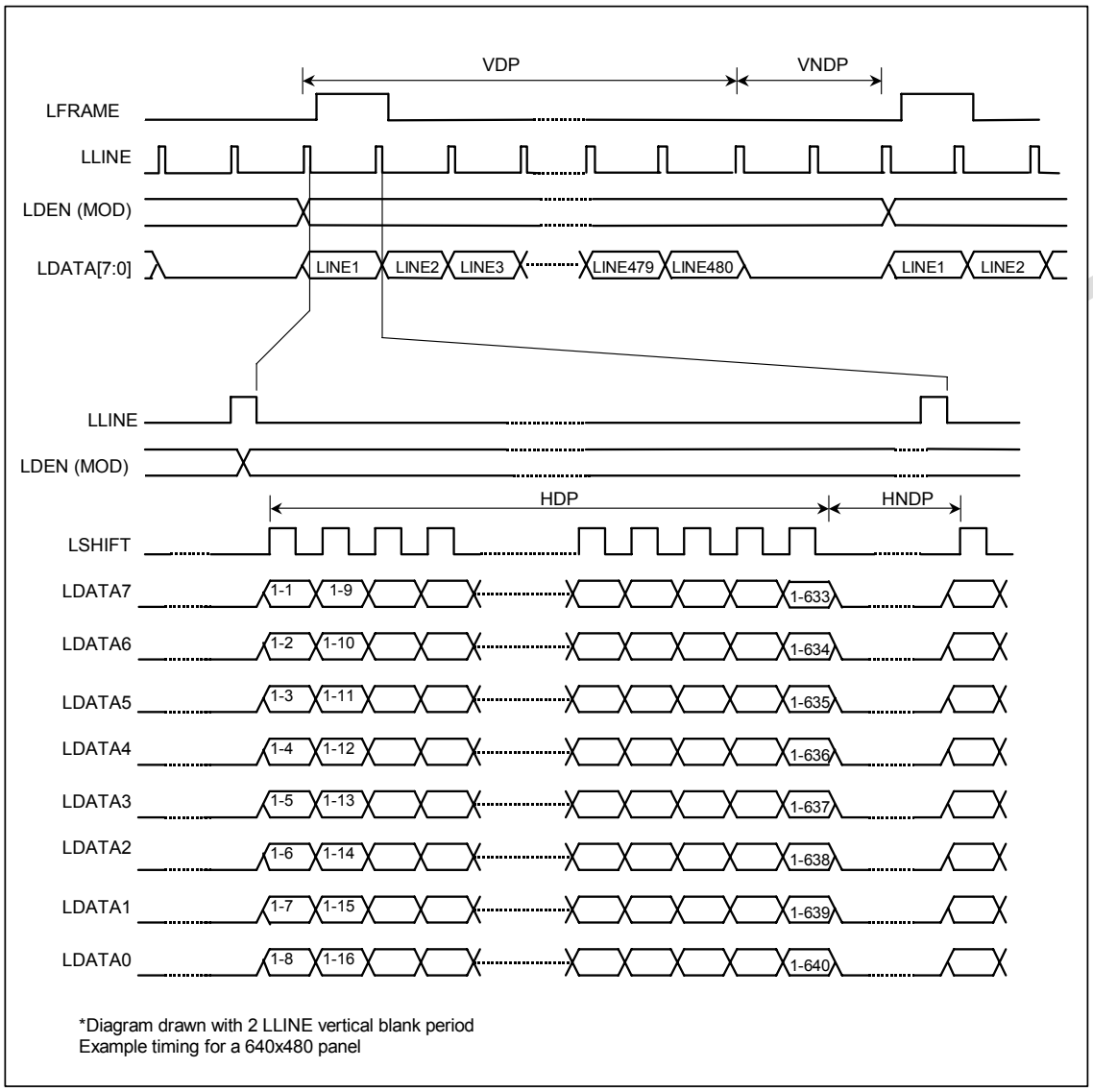
1 Ts = pixel clock period

2 The HDP must be a minimum of 32 pixels and can be increased by multiples of 8.

3 The VDP must be a minimum of 2 lines.

2.1.4.1 Monochrome 8-Bit Panel Timing

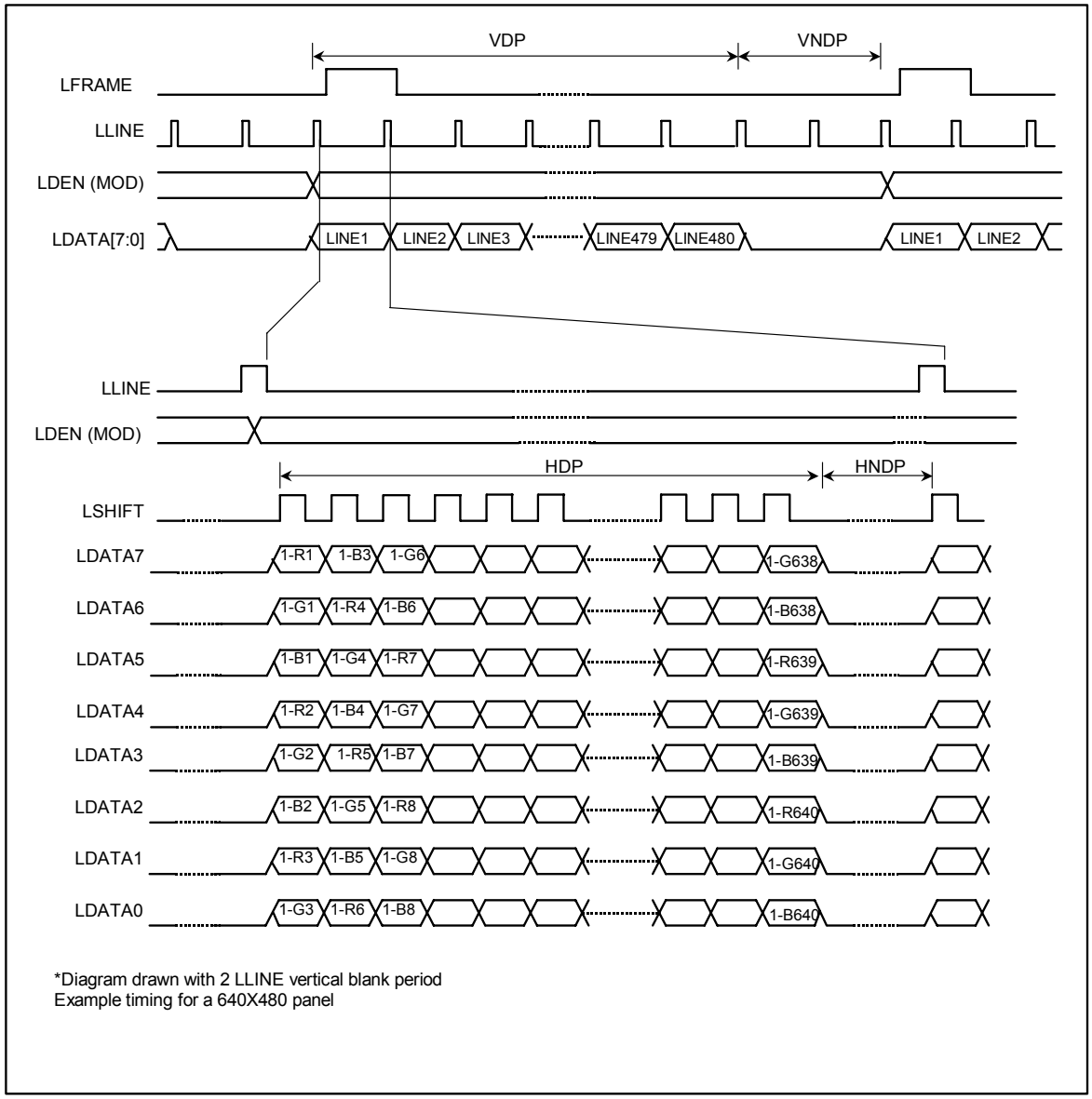
Figure 2-12: Monochrome 8-Bit Panel Timing



VDP = Vertical Display Period
= (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) + 1 Lines
VNDP = Vertical Non-Display Period
= VT-VDP
= (REG[19h] bits 2:0, REG[18h] bits 7:0) - (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) Lines
HDP = Horizontal Display Period
= ((REG[14h] bits 6:0) + 1) x 8Ts
HNDP = Horizontal Non-Display Period
= HT - HDP
= (((REG[12h] bits 7:0) + 1) x 8Ts + (REG[13h] bits 2-0)) - (((REG[14h] bits 6:0) + 1) x 8Ts)

2.1.4.2 Color 8-Bit Panel Timing (Format stripe)

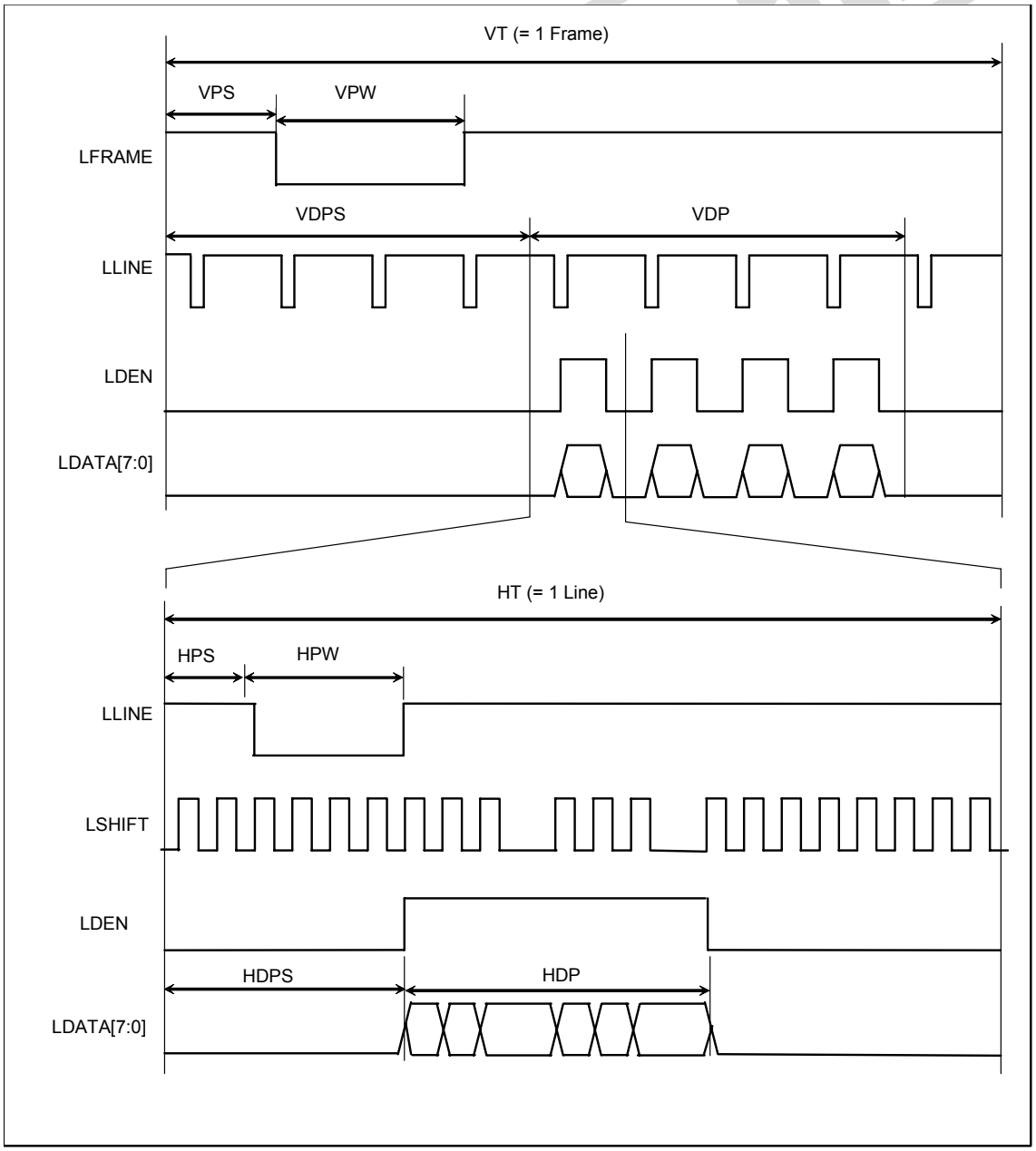
Figure 2-13: Color 8-Bit Panel Timing (Format stripe)



VDP = Vertical Display Period
= (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) + 1 Lines
VNDP = Vertical Non-Display Period
= VT-VDP
= (REG[19h] bits 2:0, REG[18h] bits 7:0) - (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) Lines
HDP = Horizontal Display Period
= ((REG[14h] bits 6:0) + 1) x 8Ts
HNDP = Horizontal Non-Display Period
= HT - HDP
= (((REG[12h] bits 7:0) + 1) x 8Ts + (REG[13h] bits 2-0)) - (((REG[14h] bits 6:0) + 1) x 8Ts)

2.1.4.3 Serial TFT Panel Timing

Figure 2-14: Serial TFT Panel Timing



VT = Vertical Total
= [(REG[19h] bits 2-0, REG[18h] bits 7-0) + 1] lines
VPS = LFRAME Pulse Start Position
= [(REG[27h] bits 2-0, REG[26h] bits 7-0)] x HT + (REG[31h] bits 2-0, REG[30h] bits 7-0) pixels
VPW = LFRAME Pulse Width
= [(REG[24h]bits2-0)+ 1] x HT + (REG[35h] bits 2-0, REG[34h] bits 7-0) – (REG[31h] bits 2-0, REG[30h] bits 7-0) pixels
VDPS = Vertical Display Period Start Position
= [(REG[1Fh]bits2-0,REG[1Eh]bits7-0)] lines
VDP = Vertical Display Period
= [(REG[1Dh]bits1-0,REG[1Ch]bits7-0)+ 1] lines
* The VDP must be a minimum of 2 lines
HT = Horizontal Total
= [(REG[12h] bits 7-0) x 8 + (REG[13h] bits 2-0) + 1] pixels
HPS = LLINE Pulse Start Position
= [(REG[23h] bits 2-0, REG[22h] bits 7-0) + 1] pixels
HPW = LLINE Pulse Width
= [(REG[20h] bits 6-0)+ 1] pixels
HDPS = Horizontal Display Period Start Position
= [(REG[17h] bits 2-0, REG[16h] bits 7-0)] pixels
HDP = Horizontal Display Period
= [((REG[14h] bits 6-0) + 1) x 8] pixels
The HDP must be a minimum of 32 pixels and can be increased by multiples of 8.

*Panel Type Bits (REG[10h] bits 2-0) = 010 (Serial 8-bit TFT)
*LLINE Pulse Polarity Bit (REG[24h] bit 7) = 0 (active low)
*LFRAME Polarity Bit (REG[20h] bit 7) = 0 (active low)

In horizontal display period, one cycle out of every four LSHIFT clock is off. In horizontal non-display period, all LSHIFT clock cycles are on.
So, Horizontal Total (REG[13h] bits 2-0, REG[12h] bits 7-0) = [(no of subpixel clock of horizontal total – no of subpixel clock of horizontal display period)/4 + (no of subpixel clock of horizontal display period)/3] - 1

LLINE Pulse Start Position (REG[23h] bits 2-0, REG[22h] bits 7-0) = (no of subpixel clock of LLINE Pulse Start Position)/4 - 1

LLINE Pulse Width (REG[20h] bits 6-0) = (no of subpixel clock of LLINE Pulse Width)/4 – 1

Horizontal Display Period Start Position (REG[17h] bits 2-0, REG[16h] bits 7-0) = (no of subpixel clock of Horizontal Display Period Start Position)/4 - 1

Horizontal Display Period (REG[14h] bits 6-0) = [(no of subpixel clock of horizontal display period)/3]/8 – 1

The frequency of LSHIFT was different during display and non-display period.
During horizontal display period:
LSHIFT frequency = (¾) * MCLK frequency * (PCLK Frequency Ratio + 1) / (2^{18})
During horizontal non-display period:
LSHIFT frequency = MCLK frequency * (PCLK Frequency Ratio + 1) / (2^{18})

Panel Type Register **REG[10h]**

Bit	7	6	5	4	3	2	1	0
	Color STN Panel Select	Color/Mono Panel	Panel Data Width Bit 1	Panel Data Width Bit 0	Reserved	Panel Type Bit 2	Panel Type Bit 1	Panel Type Bit 0

Bit	7	6	5	4	3	2	1	0
		Select						
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **Color STN Panel Select**
When this bit = 0, non CSTN LCD panel is selected.
When this bit = 1, CSTN LCD panel is selected.

Bit 6 **Color/Mono Panel Select**
When this bit = 0, monochrome LCD panel is selected.
When this bit = 1, color LCD panel is selected.

Bits 5-4 **Panel Data Width Bits [1:0]**
These bits are determined by the data width of the LCD panel. Refer to Table 2-3: Panel Data Width Selection for the selection.

Note : These 2 bits are not effective for Serial TFT panel.

Table 2-3: Panel Data Width Selection

Panel Data Width Bits [1:0]	Passive Panel Data Width
00	4-bit
01	8-bit
10	Reserved
11	Reserved

Bit 3 **Reserved bit**
This bit should be programmed by 0.

Bits 2-0 **Panel Type Bit [2:0]**
This bit selects the panel type.

Table 2-4: LCD Panel Type Selection

Panel Type Bit[2:0]	Panel Type
000	STN
010	Serial-TFT
001	Reserved
100	Smart TFT
101	Smart CSTN
110	Smart OLED
111	Reserved

MOD Rate Register **REG[11h]**

Bit	7	6	5	4	3	2	1	0
	0	0	MOD Rate Bit 5	MOD Rate Bit 4	MOD Rate Bit 3	MOD Rate Bit 2	MOD Rate Bit 1	MOD Rate Bit 0
Type	RO	RO	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 5-0 **MOD Rate Bits [5:0]**
When these bits are all 0, the MOD output signal (LDEN) toggles every LFRAME.
For any non-zero value n, the MOD output signal (LDEN) toggles every n LLINE.

Note: These bits are for passive LCD panels and REG[340h] = 0 only.

Horizontal Total Register 1 **REG[12h]**

Bit	7	6	5	4	3	2	1	0
	Horizontal Total Bit 10	Horizontal Total Bit 9	Horizontal Total Bit 8	Horizontal Total Bit 7	Horizontal Total Bit 6	Horizontal Total Bit 5	Horizontal Total Bit 4	Horizontal Total Bit 3
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Horizontal Total Register 0 **REG[13h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	Horizontal Total Bit 2	Horizontal Total Bit 1	Horizontal Total Bit 0
Type	RO	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	1	1	1

REG[12h], Bits 6-0
REG[13h], Bits 2-0

Horizontal Total Bits [10:0]
This register is used for both dumb and smart panel interface.

For dumb panel interface, these bits specify the LCD panel Horizontal Total period. The Horizontal Total is the sum of the Horizontal Display period and the Horizontal Non-Display period.

The maximum Horizontal Total is 2048 pixels.

Horizontal Total in number of pixels = REG[12h] Bits [7:0] x 8 + REG[13h] Bits [2:0] + 1

Note
(1) This register must be programmed such that the following condition is fulfilled.
HDPS + HDP < HT

(2) For panel AC timing and timing parameter definitions, see Section "Display Interface" in datasheet.

(3) For smart panel interface (i.e. REG[10h] bit 2 = 1 and REG[250h] bit 5 = 1), REG[12h] will be used as horizontal number of pixels and REG[13h] = 0x07.
Horizontal width of smart panel = REG[12h] Bits[7:0] x 8 + 8

Horizontal Display Period Register **REG[14h]**

Bit	7	6	5	4	3	2	1	0
	0	Horizontal Display Period Bit 6	Horizontal Display Period Bit 5	Horizontal Display Period Bit 4	Horizontal Display Period Bit 3	Horizontal Display Period Bit 2	Horizontal Display Period Bit 1	Horizontal Display Period Bit 0
Type	RO	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7
Bits 6-0

Reserved bit
Horizontal Display Period Bits [6:0]
This register is used for both dumb and smart panel interface.

For dumb panel interface, these bits specify the LCD panel Horizontal Display period, in 8 pixel resolution. The Horizontal Display period should be less than the Horizontal Total to allow for a sufficient Horizontal Non-Display period.

Horizontal Display Period in number of pixels = (Bits [6:0] + 1) x 8

Note
(1) Maximum value of REG[14h] ≤ 0x3F when Display Rotate Mode (90° or 270°) is selected.

(2) For panel AC timing and timing parameter definitions, see Section "Display Interface" in

datasheet.

(3) For smart panel interface (i.e. REG[10h] bit 2 = 1 and REG[250h] bit 5 = 1), these bits should be set as same as REG[12h].

Horizontal Display Period Start Position Register 0 **REG[16h]**

Bit	7	6	5	4	3	2	1	0
	Horizontal Display Period Start Position Bit 7	Horizontal Display Period Start Position Bit 6	Horizontal Display Period Start Position Bit 5	Horizontal Display Period Start Position Bit 4	Horizontal Display Period Start Position Bit 3	Horizontal Display Period Start Position Bit 2	Horizontal Display Period Start Position Bit 1	Horizontal Display Period Start Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Horizontal Display Period Start Position Register 1 **REG[17h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	Horizontal Display Period Start Position Bit 10	Horizontal Display Period Start Position Bit 9	Horizontal Display Period Start Position Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[17h] bits1-0, REG[16h] bits 7-0

Horizontal Display Period Start Position Bits [10:0]
These bits specify the Horizontal Display Period Start Position in 1 pixel resolution.

Note
(1) For panel AC timing and timing parameter definitions, see Section "Display Interface" in datasheet.
(2) These bit can not be updated for smart panel interface (REG[250h] bit 5 = 1)

Vertical Total Register 0 **REG[18h]**

Bit	7	6	5	4	3	2	1	0
	Vertical Total Bit 7	Vertical Total Bit 6	Vertical Total Bit 5	Vertical Total Bit 4	Vertical Total Bit 3	Vertical Total Bit 2	Vertical Total Bit 1	Vertical Total Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Vertical Total Register 1 **REG[19h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	Vertical Total Bit 10	Vertical Total Bit 9	Vertical Total Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[19h] bits 1-0, REG[18h] bits 7-0

Vertical Total Bits [10:0]
This register is used for both dumb and smart panel interface.

For dumb panel interface, these bits specify the LCD panel Vertical Total period, in 1 line resolution. The Vertical Total is the sum of the Vertical Display Period and the Vertical Non-Display Period.

The maximum Vertical Total is 2048 lines. See "Display Interface" in datasheet.

Vertical Total in number of lines = Bits [10:0]+ 1

Note

(1) This register must be programmed such that the following condition is fulfilled.
VDPS + VDP < VT

(2) For panel AC timing and timing parameter definitions, see Section "Display Interface".

(3) For smart panel interface (i.e. REG[10h] bit 2 = 1 and REG[250h] bit 5 = 1), these bits will be used as vertical number of lines in smart panel.
Vertical height of smart panel = REG[19h] 2:0, REG[18h] 7:0 + 1

Vertical Display Period Register 0 **REG[1Ch]**

Bit	7	6	5	4	3	2	1	0
	Vertical Display Period Bit 7	Vertical Display Period Bit 6	Vertical Display Period Bit 5	Vertical Display Period Bit 4	Vertical Display Period Bit 3	Vertical Display Period Bit 2	Vertical Display Period Bit 1	Vertical Display Period Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Vertical Display Period Register 1 **REG[1Dh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Vertical Display Period Bit 9	Vertical Display Period Bit 8
Type	RO	Ro	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1Dh] bits 1-0, REG[1Ch] bits 7-0

Vertical Display Period Bits [9:0]
This register is used for both main and smart panel interface.

For dumb panel interface, these bits specify the LCD panel Vertical Display period, in 1 line resolution. The Vertical Display period should be less than the Vertical Total to allow for a sufficient Vertical Non-Display period.

Vertical Display Period in number of lines = Bits [9:0] + 1

Note

(1) For panel AC timing and timing parameter definitions, see Section "Display Interface" in datasheet.

(2) For smart panel interface (i.e. REG[10h] bit 2 = 1 and REG[250h] bit 5 = 1), these bits should be set as same as REG[19h-18h].

Vertical Display Period Start Position Register 0 **REG[1Eh]**

Bit	7	6	5	4	3	2	1	0
	Vertical Display Period Start Position Bit 7	Vertical Display Period Start Position Bit 6	Vertical Display Period Start Position Bit 5	Vertical Display Period Start Position Bit 4	Vertical Display Period Start Position Bit 3	Vertical Display Period Start Position Bit 2	Vertical Display Period Start Position Bit 1	Vertical Display Period Start Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Vertical Display Period Start Position Register 1 **REG[1Fh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	Vertical Display	Vertical Display	Vertical Display

Bit	7	6	5	4	3	2	1	0
						Start Position Period Bit 10	Start Position Period Bit 9	Start Position Period Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1Fh] bits 1-0, REG[1Eh] bits 7-0 **Vertical Display Period Start Position Bits [10:0]**
These bits specify the Vertical Display Period Start Position in 1 line resolution.

Note
For panel AC timing and timing parameter definitions, see Section "Display Interface" in datasheet.

LLINE Pulse Width Register **REG[20h]**

Bit	7	6	5	4	3	2	1	0
	LLINE Pulse Polarity	LLINE Pulse Width Bit 6	LLINE Pulse Width Bit 5	LLINE Pulse Width Bit 4	LLINE Pulse Width Bit 3	LLINE Pulse Width Bit 2	LLINE Pulse Width Bit 1	LLINE Pulse Width Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LLINE Pulse Polarity**
This bit determines the polarity of the horizontal sync signal. The horizontal sync signal is typically named as LLINE or LP, depending on the panel type.
When this bit = 0, the horizontal sync signal is active low.
When this bit = 1, the horizontal sync signal is active high.

Note
(1) These bit can not be updated for smart panel interface (REG[250h] bit 5 = 1)

Bits 6-0 **LLINE Pulse Width Bits [6:0]**
These bits specify the width of the panel horizontal sync signal, in number of PCLK. The horizontal sync signal is typically named as LLINE or LP, depending on the panel type.

LLINE Pulse Width in PCLK = Bits [6:0] + 1

Note
(1) These bit can not be updated for smart panel interface (REG[250h] bit 5 = 1)

Note
(1) For panel AC timing and timing parameter definitions, see Section "Display Interface" in datasheet.

LLINE Pulse Start Sub-pixel Position Register **REG[21h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Sub-pixel Position Bit 1	Sub-pixel Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-2 **Reserved bits**

Bits 1-0 **Sub-pixel Position Bits [1:0]**
00 : No sub-pixel delay
01 : 1 sub-pixel clock delay

10 : 2 sub-pixel clock delay
11 : 3 sub-pixel clock delay

Note
(1) This register is effective for Serial-TFT panel only.

LLINE Pulse Start Position Register 0 **REG[22h]**

Bit	7	6	5	4	3	2	1	0
	LLINE Pulse Start Position Bit 7	LLINE Pulse Start Position Bit 6	LLINE Pulse Start Position Bit 5	LLINE Pulse Start Position Bit 4	LLINE Pulse Start Position Bit 3	LLINE Pulse Start Position Bit 2	LLINE Pulse Start Position Bit 1	LLINE Pulse Start Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LLINE Pulse Start Position Register 1 **REG[23h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	LLINE Pulse Start Position Bit 10	LLINE Pulse Start Position Bit 9	LLINE Pulse Start Position Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[23h] bits 1-0, REG[22h] bits 7-0 **LLINE Pulse Start Position Bits [10:0]**
These bits specify the start position of the horizontal sync signal, in number of PCLK.

LLINE Pulses Start Position in PCLK = Bits [10:0] + 1

Note
(1) These bit can not be updated for smart panel interface (REG[250h] bit 5 = 1)

Note
(1) For panel AC timing and timing parameter definitions, see Section “Display Interface” in datasheet”.

LFRAME Pulse Width Register **REG[24h]**

Bit	7	6	5	4	3	2	1	0
	LFRAME Pulse Polarity	0	0	0	0	LFRAME Pulse Width Bit 2	LFRAME Pulse Width Bit 1	LFRAME Pulse Width Bit 0
Type	RW	NA	NA	NA	NA	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LFRAME Pulse Polarity**
This bit selects the polarity of the vertical sync signal. The vertical sync signal is typically named as LFRAME or SPS, depending on the panel type.

When this bit = 0, the vertical sync signal is active low.
When this bit = 1, the vertical sync signal is active high.

Note
(1) These bit can not be updated for smart panel interface (REG[250h] bit 5 = 1)

Bits 2-0 **LFRAME Pulse Width Bits [2:0]**
These bits specify the width of the panel vertical sync signal, in 1 line resolution. The vertical sync signal is typically named as LFRAME or SPS, depending on the panel type.

LFRAME Pulse Width in number of pixels = (Bits [2:0] + 1) x Horizontal Total + offset

Note
(1) These bit can not be updated for smart panel interface (REG[250h] bit 5 = 1)

Note
(1) For panel AC timing and timing parameter definitions, see Section "Display Interface" in datasheet.

LFRAME Pulse Start Position Register 0 **REG[26h]**

Bit	7	6	5	4	3	2	1	0
	LFRAME Pulse Start Position Bit 7	LFRAME Pulse Start Position Bit 6	LFRAME Pulse Start Position Bit 5	LFRAME Pulse Start Position Bit 4	LFRAME Pulse Start Position Bit 3	LFRAME Pulse Start Position Bit 2	LFRAME Pulse Start Position Bit 1	LFRAME Pulse Start Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LFRAME Pulse Start Position register 1 **REG[27h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	LFRAME Pulse Start Position Bit 10	LFRAME Pulse Start Position Bit 9	LFRAME Pulse Start Position Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[27h] bits 1-0
REG[26h] bits 7-0 **LFRAME Pulse Start Position Bits [10:0]**
These bits specify the start position of the vertical sync signal, in 1 line resolution.

LFRAME Pulse Start Position in number of pixels = (Bits [10:0]) x Horizontal Total + offset

Note
(1) These bit can not be updated for smart panel interface (REG[250h] bit 5 = 1)

Note
(1) For panel AC timing and timing parameter definitions, see Section "Display Interface" in datasheet.

Display Post-processing Saturation Control Register **REG[2Ch]**

Bit	7	6	5	4	3	2	1	0
	Display Post-proc Saturation Bit 7	Display Post-proc Saturation Bit 6	Display Post-proc Saturation Bit 5	Display Post-proc Saturation Bit 4	Display Post-proc Saturation Bit 3	Display Post-proc Saturation Bit 2	Display Post-proc Saturation Bit 1	Display Post-proc Saturation Bit 0
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	1	0	0	0	0	0	0

Bits 7-0 **Display Post-processing Saturation Control [7:0]**
These bits control the saturation of the display.

Table 2-5: The setting for display post-processing saturation

Control Bits [7:0]	Saturation Control
0x00	Gain = 0
0x01	Gain = 1/64
…	

0x40	(Default) Gain = 1
…	
0x7F	Gain = 127/64

Display Post-processing Brightness Control Register **REG[2Dh]**

Bit	7	6	5	4	3	2	1	0
	Display Post-proc Brightness Bit 7	Display Post-proc Brightness Bit 6	Display Post-proc Brightness Bit 5	Display Post-proc Brightness Bit 4	Display Post-proc Brightness Bit 3	Display Post-proc Brightness Bit 2	Display Post-proc Brightness Bit 1	Display Post-proc Brightness Bit 0
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	1	0	0	0	0	0	0	0

Bits 7-0 **Display Post-processing Brightness Control [7:0]**
These bits control the brightness of the display.

Table 2-6: The setting for Display Post-processing brightness

Control Bits [7:0]	Brightness Control
0x00	Value = 0
…	
0x80	(Default) Value = 128
…	
0xFF	Value = 255

Display Post-processing Contrast Control Register **REG[2Eh]**

Bit	7	6	5	4	3	2	1	0
	Display Post-proc Contrast Bit 7	Display Post-proc Contrast Bit 6	Display Post-proc Contrast Bit 5	Display Post-proc Contrast Bit 4	Display Post-proc Contrast Bit 3	Display Post-proc Contrast Bit 2	Display Post-proc Contrast Bit 1	Display Post-proc Contrast Bit 0
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	1	0	0	0	0	0	0

Bits 7-0 **Display Post-processing Contrast Control [7:0]**
These bits control the contrast of the display.

Table 2-7: The setting for Display Post-processing contrast

Control Bits [7:0]	Contrast Control
0x00	Gain = 0
0x01	Gain = 1/64
…	
0x40	(Default) Gain = 1
…	
0x7F	Gain = 127/64

Display Post-processing Control Register **REG[2Fh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Display Post-proc Enable
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **Display Post-processing Enable**
When this bit = 1, display post-processing is enabled.
When this bit = 0, display post-processing is disabled.

LFRAME Pulse Start Offset Register 0 **REG[30h]**

Bit	7	6	5	4	3	2	1	0
	LFRAME Start Offset Bit 7	LFRAME Start Offset Bit 6	LFRAME Start Offset Bit 5	LFRAME Start Offset Bit 4	LFRAME Start Offset Bit 3	LFRAME Start Offset Bit 2	LFRAME Start Offset Bit 1	LFRAME Start Offset Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LFRAME Pulse Start Offset Register 1 **REG[31h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	LFRAME Start Offset Bit 10	LFRAME Start Offset Bit 9	LFRAME Start Offset Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[31h] bits 2-0 **LFRAME Pulse Start Offset [10:0]**
REG[30h] bits 7-0 These bits specify the start offset of the vertical sync signal within a line, in 1 pixel resolution.

Note
(1) For panel AC timing and timing parameter definitions, see Section "Display Interface" in datasheet.

LFRAME Pulse Stop Offset Register 0 **REG[34h]**

Bit	7	6	5	4	3	2	1	0
	LFRAME Stop Offset Bit 7	LFRAME Stop Offset Bit 6	LFRAME Stop Offset Bit 5	LFRAME Stop Offset Bit 4	LFRAME Stop Offset Bit 3	LFRAME Stop Offset Bit 2	LFRAME Stop Offset Bit 1	LFRAME Stop Offset Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LFRAME Pulse Stop Offset Register 1 **REG[35h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	LFRAME Stop Offset Bit 10	LFRAME Stop Offset Bit 9	LFRAME Stop Offset Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[35h] bits 2-0 **LFRAME Pulse Stop Offset [10:0]**
REG[34h] bits 7-0 These bits specify the stop offset of the vertical sync signal within a line, in 1 pixel resolution.

Note
(1) For panel AC timing and timing parameter definitions, see Section "Display Interface" in datasheet.

RGB sequence Register **REG[42h]**

Bit	7	6	5	4	3	2	1	0
	0	0	Even line RGB sequence Bit	Even line RGB sequence Bit	Even line RGB sequence Bit	Odd line RGB sequence Bit	Odd line RGB sequence Bit	Odd line RGB sequence Bit

Bit	7	6	5	4	3	2	1	0
			2	1	0	2	1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-6 **Reserved bits**

Bits 5-3 **Even line RGB sequence Bits [2:0]**

Bits 2-0 **Odd line RGB sequence Bits [2:0]**

The first display line is the odd line.

Note

This register is effective for Serial-TFT panel only. The first line is the even line.

Table 2-8: The RGB sequence for Serial-TFT interface

RGB sequence bits [2:0]	RGB output sequence
000	RGB
001	RBG
010	GRB
011	GBR
100	BRG
101	BGR
11x	Reserved

MOD Time Period Register 0 **REG[340h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	MOD Time Period Bit 3	MOD Time Period Bit 2	MOD Time Period Bit 1	MOD Time Period Bit 0
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[340h] bits 3-0 **MOD Period position Register [3:0]**

This register define the period of MOD output signal, in terms of display line.

0 : Disable and MOD output will follow the setting of REG[11h]

n : MOD polarity will repeat every n frames with the patterns defined in REG[342-343h]. The setting of REG[11h] will be ignored.

Note: These bits are for passive LCD panels only.

MOD Time Pattern Register 0 **REG[342h]**

Bit	7	6	5	4	3	2	1	0
	MOD Time Pattern Bit 7	MOD Time Pattern Bit 6	MOD Time Pattern Bit 5	MOD Time Pattern Bit 4	MOD Time Pattern Bit 3	MOD Time Pattern Bit 2	MOD Time Pattern Bit 1	MOD Time Pattern Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

MOD Time Pattern Register 1 **REG[343h]**

Bit	7	6	5	4	3	2	1	0
	MOD Time Pattern Bit 15	MOD Time Pattern Bit 14	MOD Time Pattern Bit 13	MOD Time Pattern Bit 12	MOD Time Pattern Bit 11	MOD Time Pattern Bit 10	MOD Time Pattern Bit 9	MOD Time Pattern Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
state								

REG[343h] bits 7-0,
REG[342h] bits 7-0

MOD Time pattern register [15:0]
This register defines MOD pattern.
1 : Positive polarity
0 : Negative polarity

For example,
REG[340h] = 0x4
REG[343-342h] = 0x0001 (i.e. b0001)
Then the frame #1 is positive polarity, frame #2-4 are negative polarity. And such pattern will be repeated for every 4 frames.

Note
(1) These bits are for passive LCD panels only.

LSHIFT signal start position Register 0 **REG[350h]**

Bit	7	6	5	4	3	2	1	0
	LSHIFT start Bit 7	LSHIFT start Bit 6	LSHIFT start Bit 5	LSHIFT start Bit 4	LSHIFT start Bit 3	LSHIFT start Bit 2	LSHIFT start Bit 1	LSHIFT start Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LSHIFT signal start position Register1 **REG[351h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LSHIFT start Bit 10	LSHIFT start Bit 9	LSHIFT start Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

LSHIFT signal end position Register 0 **REG[354h]**

Bit	7	6	5	4	3	2	1	0
	LCD shift end Bit 7	LCD shift end Bit 6	LCD shift end Bit 5	LCD shift end Bit 4	LCD shift end Bit 3	LCD shift end Bit 2	LCD shift end Bit 1	LCD shift end Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LSHIFT signal end position Register1 **REG[355h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD shift end Bit 10	LCD shift end Bit 9	LCD shift end Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[351h] bits 1-0, REG[350h] bits 7-0 **LSHIFT signal start position Register [10:0]**
These bits define the start position of LSHIFT within a line for active lcd signal.

REG[355h] bits 1-0, REG[354h] bits 7-0 **LSHIFT signal end position Register [10:0]**
These bit defines the end position of LSHIFT within a line for active lcd signal.

If REG[351-350h] and REG[355-354h] are 0, LSHIFT will be enabled on the time (i.e. both display and non-display period).
If REG[351-350h] and REG[355-354h] are not equal to 0, LSHIFT will be enabled within the range defined by start and end position within each line.

The end position > The start position

For example,
Horizontal display period = 16
Horizontal total = 24
Horizontal period start = 4
LSHIFT start = 3 and LSHIFT end = 20
Then the LSHIFT will enabled between horizontal position pixel 3 and 20 within a line.

2.1.5 Smart Panel Configuration Registers

To operate with smart panel module, first is to enable the smart display (REG[250h] bit 5 = 1), then select the operation mode (Write Through or Auto Reflesh Mode), panel type and panel interface (REG[260h] & REG[10h]). The maximum resolution for main smart panel is 2048 x 2048.

Then next step is to configure the type of data to be transferred for Write Through Mode. Detail will be covered in section 2.1.5.1.

Data transfer is initiated by writing data to REG[26Ch] and REG[26Dh] for Write Through Mode while by setting REG[A0h] to enable the Auto Reflesh Mode of smart panel interface.

2.1.5.1 Write Through Mode

There are three types of data for Write Through Mode :

- Display pixel data
- Command
- Command argument

Write Through Mode will send either pixel display data or command or command argument to the panel driver at each time by issuing an enable pulse together with the 16-bit data or command.

Table 2-9: Data type for parallel interface of all panel types

Data Type	Data / Command Selection REG[26Eh] Bit 0	Data / Argument Selection REG[26Eh] Bit 7
Command	0	x
Command Argument	1	0
Pixel Data	1	1

2.1.5.1.1 Input Format

MCU data for Write Through mode are stored in REG[26Ch-26Dh]. Input data format from MCU in Write Through Mode as Table 2-10: Input data format for Write Through Mode.

Table 2-10: Input data format for Write Through Mode

Input Data Type	Smart Display Input Data [15:0] (REG[26Dh-26Ch])
Command/Argument (8 bit)	XXXXXXXXDDDDDDDD
Pixel Data (16bpp)	RRRRRGGGGGGBBBBB

2.1.5.1.2 Output Format

Considering parallel output interface, only TFT will support 8-bit and 9-bit interface while CSTN and OLED will only have 8-bit interface (SUB_DATA8 will be 0). Refer to Table 2-11: Data output format for display pixel in Parallel interface for output data format of different panel interface.

Table 2-11: Data output format for display pixel in Parallel interface

Panel Type	Parallel Interface Width REG[260h] Bit 5	Byte per pixel REG[261h] Bits 1-0	Pixel Data Description	Parallel Pixel Data Format (SUB_DATA[8:0])
TFT	0	xx	2 transfers per pixel (18bpp)	RRRRRRGGG GGGBBBBBB
	1	10	2 transfers per pixel(16bpp)	RRRRRGGG0 GGGBBBBB0
	1	11	3 transfers per pixel (18bpp)	RRRRRR000 GGGGGG000 BBBBBB000
CSTN	x	00	1 transfer per pixel (8bpp)	0RRRGGGBB
	x	01	3 transfers per 2 pixels (12bpp)	0RRRRGGGG 0BBBBRRRR 0GGGGBBBB
	x	10	2 transfers per pixel (16bpp)	0RRRRRGGG 0GGGBBBBB
	x	11	3 transfers per pixel (18bpp)	0RRRRRR00 0GGGGGG00 0BBBBBB00
OLED	x	00	1 transfer per pixel (8bpp)	0RRRGGGBB
	x	01	3 transfers per 2 pixels (12bpp)	0RRRRGGGG 0BBBBRRRR 0GGGGBBBB
	x	10	2 transfers per pixel (16bpp)	0RRRRRGGG 0GGGBBBBB
	x	11	3 transfers per pixel (18bpp)	000RRRRRR 000GGGGGG 000BBBBBB

Table 2-12: Data output format for 4-wire serial interface (REG[260h] Bit 6= 1)

Panel Type	Output Data Type	Byte per pixel REG[261h] Bits 1-0	Data Length per burst	Serial Output Format (SDA)

TFT/CSTN	Command/Argument	xx	8 bit	DDDDDDDD
	Pixel (8bpp)	00	8 bit	RRRGGGBB
	Pixel (12bpp)	01	8 bit, 16 bit	RRRRGGGG, BBBBRRRRGGGGBBBB
	Pixel (16bpp)	10	16 bit	RRRRRGGGGGGBBBBB
	Pixel (18bpp)	11	24 bit	RRRRRR00GGGGGG00BBBBBB00
OLED	Command/Argument	xx	8 bit	DDDDDDDD
	Pixel (8bpp)	00	8 bit	RRRGGGBB
	Pixel (12bpp)	01	8 bit, 16 bit	RRRRGGGG BBBBRRRRGGGGBBBB
	Pixel (16bpp)	10	16 bit	RRRRRGGGGGGBBBBB
	Pixel (18bpp)	11	24 bit	00RRRRRR00GGGGGG00BBBBBB

Table 2-13: Data output format for 3-wire serial interface (REG[260h] Bit 6= 0)

Panel Type	Output Data Type	Byte per pixel REG[261h] Bits 1-0	Data Length per burst	Serial Output Format (SDA)
TFT/CSTN	Command	xx	9 bit	0DDDDDDDD
	Argument	xx	9 bit	1DDDDDDDD
	Pixel (8bpp)	00	9 bit	1RRRGGGBB
	Pixel (12bpp)	01	9 bit, 18 bit	1RRRRGGGG, 1BBBBRRRR1GGGGBBBB
	Pixel (16bpp)	10	18 bit	1RRRRRGGG1GGGBBBBB
	Pixel (18bpp)	11	27 bit	1RRRRRR001GGGGGG001BBBBBB00
OLED	Command	xx	9 bit	0DDDDDDDD
	Argument	xx	9 bit	1DDDDDDDD
	Pixel (8bpp)	00	9 bit	1RRRGGGBB
	Pixel (12bpp)	01	9 bit, 18 bit	1RRRRGGGG 1BBBBRRRR1GGGGBBBB
	Pixel (16bpp)	10	18 bit	1RRRRRGGG1GGGBBBBB
	Pixel (18bpp)	11	27 bit	100RRRRRR100GGGGGG100BBBBBB

2.1.5.2 Auto Reflesh Mode

Auto Reflesh Mode will accept 32-bit (32 bits per pixel or 16 bits per pixel) display data from display memory and transfer them to panel driver continuously.

There is only display pixel data transfer (no command or argument transfer) for Auto Reflesh Mode. So there is no need to configure the data type.

Before transfer the pixel data to smart panel, it should configure the panel parameter with [12-13h, 18-19h, 74-76h, 78-79h].

The data input is set according to the format below in memory:

Table 2-14: Data input format for Auto Reflesh Mode

Bit per pixel REG[70h] Bits 2:0	Data [31:0]
100 (16 bit per pixel)	RRRRRGGGGGGBBBBBRRRRRGGGGGGBBBBB (2 pixels)
101 (32 bit per pixel)	RRRRRRRRGGGGGGGGBBBBBBBBXXXXXXXX (1 pixel)

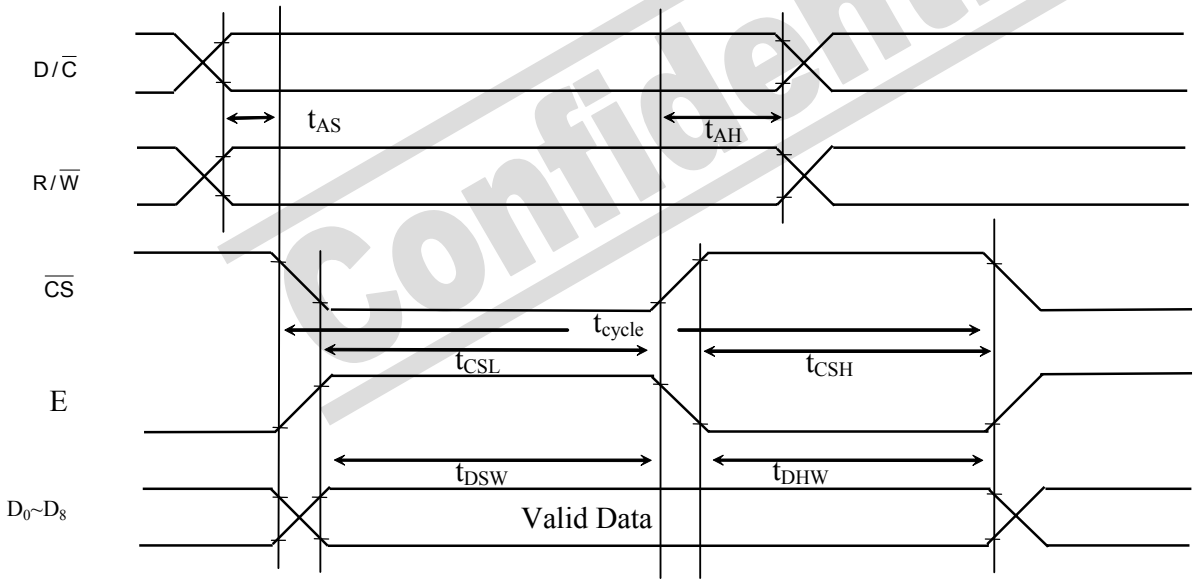
The output data format is as same as the output pixel data format in Write Through Mode.

2.1.5.3 Output Interface Timing

Table 2-15: Output Timing for 6800 Parallel interface

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time (write cycle) (note2)	2	-	36	Ts (note1)
t_{CSL}	Chip Select Low Width (note3)	1	-	16	Ts
t_{CSH}	Chip Select High Width (note4)	3	-	20	Ts
t_{AS}	Address Setup Time	-	1	-	Ts
t_{AH}	Address Hold Time	-	1	-	Ts
t_{DS}	Data Setup Time (note5)	1	-	16	Ts
t_{DH}	Data Hold Time (note6)	3	-	20	Ts

Figure 2-15: 6800 Timing Diagram



Note

(1) Ts = 1/(MCLK frequency / $2^{(REG[252h]\ bit\ 2\text{-}0)}$)

(2) $t_{cycle} = t_{CSL} + t_{CSH}$

(3) t_{CSL} = REG [270h] bit 3-0 + 1

(4) t_{CSH} = REG [271h] bit 3-0 + 1 + t during a burst or
t_{CSH} = REG [271h] bit 3-0 + 1 + t*2 at the end of the burst, t = 2

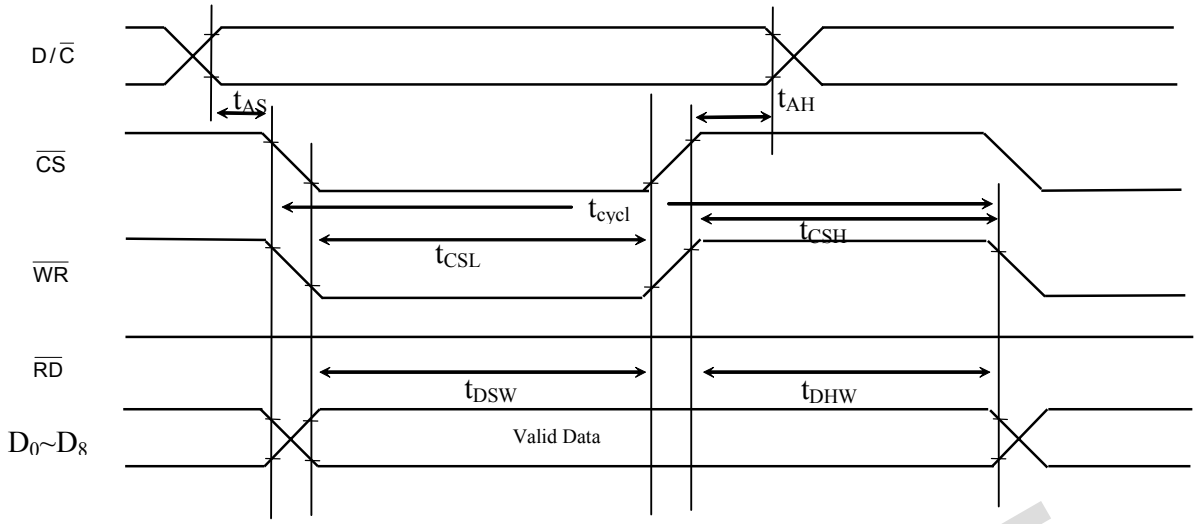
(5) $t_{DSW} = t_{CSL}$

(6) $t_{DHW} = t_{CSH}$

Table 2-16: Output Timing for 8080 Parallel interface

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time (write cycle) (note2)	2	-	36	Ts (note1)
t_{CSL}	Control Pulse Low Width (note3)	1	-	16	Ts
t_{CSH}	Control Pulse High Width (note4)	3	-	20	Ts
t_{AS}	Address Setup Time	-	1	-	Ts
t_{AH}	Address Hold Time	-	1	-	Ts
t_{DSW}	Data Setup Time (note5)	1	-	16	Ts
t_{DHW}	Data Hold Time (note6)	3	-	20	Ts

Figure 2-16: 8080 Timing Diagram



Note

(1) Ts = 1/(MCLK frequency / $2^{(REG[252h] bit 2-0)}$)

(2) $t_{cycle} = t_{CSL} + t_{CSH}$

(3) t_{CSL} = REG [270h] bit 3-0 + 1

(4) t_{CSH} = REG [271h] bit 3-0 + 1 + t during a burst or
t_{CSH} = REG [271h] bit 3-0 + 1 + t*2 at the end of the burst, t = 2

(5) $t_{DSW} = t_{CSL}$

(6) $t_{DHW} = t_{CSH}$

Table 2-17: Output Timing for Serial interface

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time (write cycle) (note2)	2	-	32	Ts (note1)
t_{CLKL}	SCK Low Width (note3)	1	-	16	Ts
t_{CLKH}	SCK High Width (note4)	1	-	16	Ts
t_{CSS}	Chip Select Setup Time (note5)	2	-	17	Ts
t_{CSH}	Chip Select Hold Time (note6)	2	-	17	Ts
t_{DSW}	Data Setup Time (note7)	1	-	16	Ts
t_{DHW}	Data Hold Time (note8)	1	-	16	Ts

Figure 2-17: 4 Wires Timing Diagram

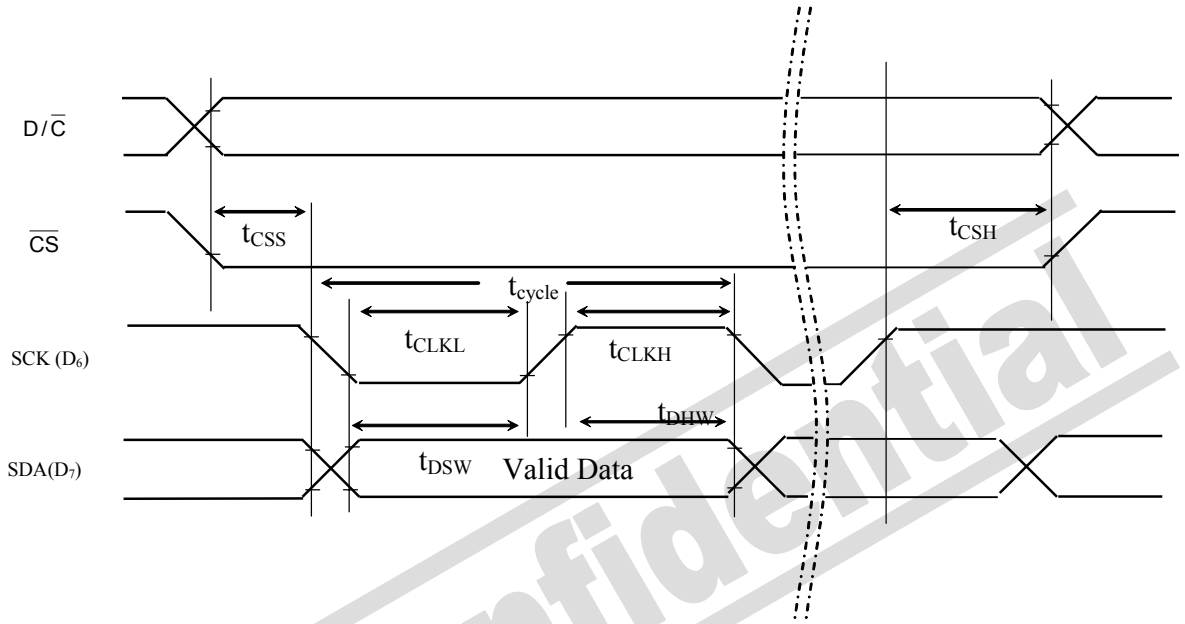
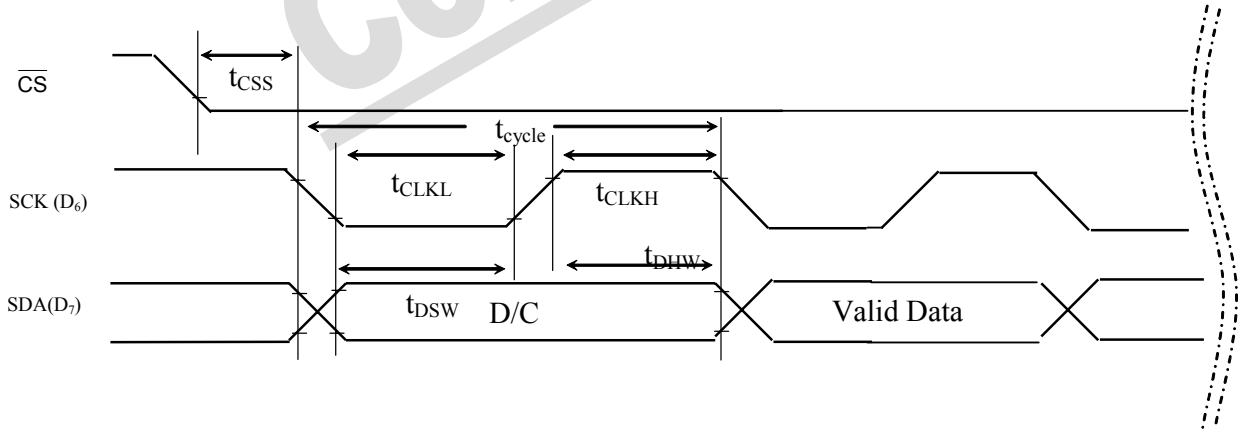


Figure 2-18: 3 Wires Timing Diagram



Note

(1) Ts = 1/(MCLK frequency / $2^{(REG[252h]\ bit\ 2\text{-}0)}$)

(2) $t_{cycle} = t_{CLKL} + t_{CLKH}$

(3) t_{CLKL} = REG [263h] bit 3-0 + 1

(4) t_{CLKH} = REG [263h] bit 3-0 + 1

(5) t_{CSS} = REG [270h] bit 3-0 + 2

(6) t_{CSH} = REG [271h] bit 3-0 + 2

(7) $t_{DSW} = t_{CLKL}$

(8) $t_{DHW} = t_{CLKH}$

Table 2-18: Register for smart panel interface

Register Description	Smart Display
Display reset	250h[7]
Display Enable	A0h[0]
Smart Select Enable	250h[5]

Display mode [1] (YUV/RGB)	1A4h[7:6]
Display mode [0] (32/16 bpp)	70h[2:0] (*1)
Divide input clock source Ration [2:0]	252h[2:0]
Display horizontal panel size [7:0]	13h[2:0]-12h[7:0]
Display vertical panel size [7:0]	19h[2:0]-18h[7:0]
Display Operation Mode Selection	260h[7]
Serial Input Interface Width	260h[6]
Parallel Input Interface Width	260h[5]
Parallel/Serial Output Interface Selection [1:0]	260h[3:2]
Panel Type Selection [1:0]	10h[2:0] (*2)
Byte per pixel Bit [1:0]	261h[1:0]
Serial Clock divide ratio [3:0]	263h[3:0]
Hold count1 Bit [3:0]	270h[3:0]
Hold count2 Bit [3:0]	271h[3:0]
Display CSC Mode	1A8h[7]
Display Y Offset Registers [7:0]	1A9h[7:0]
Display CB Offset Registers [7:0]	1AAh[7:0]
Display CR Offset Registers [7:0]	1ABh[7:0]
Display Write Through Mode input data bit [15:8]	26Ch[7:0]
Display Write Through Mode input data bit [7:0]	26Dh[7:0]
Pixel data/Argument Selection	26Eh[7]
Data/Command Selection	26Eh[0]
Window Display Start Address Bit [7:0]	74h[7:0]
Window Display Start Address Bit [15:8]	75h[7:0]
Window Display Start Address Bit [16]	76h[0]
Window Line Address Offset Bit [7:0]	78h[7:0]
Window Line Address Offset Bit [9:8]	79h[1:0]
Display ready	27Dh[0]

Note

(1) For Smart panel interface, 1/2/4/8/16/32 bit per pixel will be supported, 1/2/4/8 bpp will use lookup table.

(2) REG[10h] bit 2 should be set to 1 to enable Smart panel interface.

Smart Display Mode Register **REG[250h]**

Bit	7	6	5	4	3	2	1	0
	Smart Display reset	Reserved	Smart Select Enable	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RW	RW	RW	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG [250h] Bit 7 **Smart Display Reset**

1 : Reset
0 : Normal
0 -> 1 : transmission stop immediately
1 -> 0 : start transmission from beginning of display frame
This bit should be set to 0 before smart panel command send.

REG [250h] Bit 6 **Reserved bits**
These bits should be programmed by 0.

REG [250h] Bit 5 **Smart Enable**
1: Enable
0 : Disable

REG [250h] Bits 4-0 **Reserved bits**
These bits should be programmed by 0.

Clock Divide Register **REG[252h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Divide input clock source Ratio Bit 2	Divide input clock source Ratio Bit 1	Divide input clock source Ratio Bit 0
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG [252h] Bits 7-3 **Reserved bits**
These bits should be programmed by 0.

REG [252h] Bits 2-0 **Divide input clock source ratio register [2:0]**
This register is used for both main and smart panel interface.

Sub_CLK frequency = MCLK frequency / $2^{REG[252h]\ bit\ 2:0}$

000 : MCLK divide by 1
001 : MCLK divide by 2
010 : MCLK divide by 4
011 : MCLK divide by 8
100 : MCLK divide by 16
101 : MCLK divide by 32
11x : reserved

Smart Display Control Register **REG[260h]**

Bit	7	6	5	4	3	2	1	0
	Smart panel Operation Mode Selection	Serial interface Width	Parallel Interface Width	Reserved	Output Interface Selection Bit 1	Output Interface Selection Bit 0	Reserved	Reserved
Type	RW	RW	RW	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

This register is used for both main and smart panel interface.

REG [260h] Bit 7 **Smart panel Operation Mode Selection register**
1 : Write-Through Mode
0 : Auto-Refresh Mode
0 -> 1 : Stop immediately
1 -> 0 : continue

REG [260h] Bit 6 **Serial interface Width**
Define the width of serial interface mode
1 : 4 wire serial interface
0 : 3 wire serial interface

REG [260h] Bit 5 **Parallel Interface width**
Define the width of parallel interface for TFT
1 : 8-bit parallel interface
0 : 9-bit parallel interface
Note
(1) This bit is effective for 6800/8080 parallel TFT interface only

REG [260h] Bit 4 **Reserved bit**

This bit should be programmed by 0

REG [260h] Bits 3-2 **Output Interface Selection bits [1:0]**
00 : 6800 parallel interface
01 : 8080 parallel interface
10 : serial interface
11 : reserved

REG [260h] Bits 1-0 **Reserved bit**
This bit should be programmed by 0

Display Byte Per Pixel Register **REG[261h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Byte per pixel Bit 1	Byte per pixel Bit 0
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 4 **Byte Per Pixel register [1:0]**

00 : 8 bits per pixel output color resolution
01 : 12 bits per pixel output color resolution
10 :16 bits per pixel output color resolution
11 : 18 bits per pixel output color resolution

Serial Clock divide Register **REG[263h]**

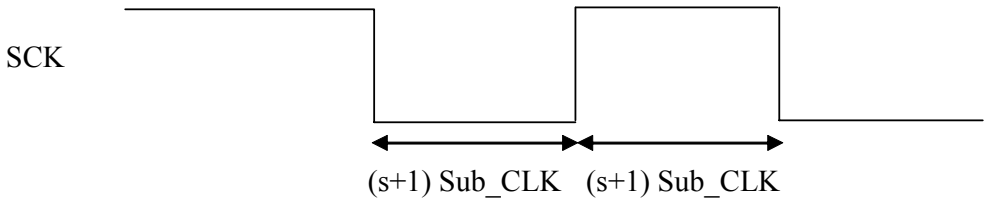
Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Serial Clock divide ratio Bit 3	Serial Clock divide ratio Bit 2	Serial Clock divide ratio Bit 1	Serial Clock divide ratio Bit 0
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-4 **Reserved bits**
These bits should be programmed by 0.

Bits 3-0 **Serial Clock divide register [3:0]**
This register is used for both main and smart panel interface.

SCK frequency = Sub_CLK frequency / (REG[263h] * 2 +2)

0000 : Sub_CLK divide by 2
0001 : Sub_CLK divide by 4
0010 : Sub_CLK divide by 6
0011 : Sub_CLK divide by 8
…
1101 : Sub_CLK divide by 28
1110 : Sub_CLK divide by 30
1111 : Sub_CLK divide by 32



Note

(1) Sub_CLK refer to REG[252h] bits 2-0

(2) s : Serial clock divide value, REG[263h] bits 3-0

Hold count for Count 1 Register **REG[270h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Hold count 2 Bit 3	Hold count 2 Bit 2	Hold count 2 Bit 1	Hold count 2 Bit 0
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-4 **Reserved bits**

Bits 3-0 **Hold count register [3:0]**

If 6800 interface, Hold count for E high register [3:0]
E will hold high for 1-16 extra clock cycles
If 8080 interface, Hold count for WR low register [3:0]
WR will hold low for 1-16 extra clock cycles
If SPI interface, CS to SCK cycle count register [3:0]
2-17 extra clock cycles for CS hold low before SCK start

Smart Display Hold Count 2 Register **REG[271h]**

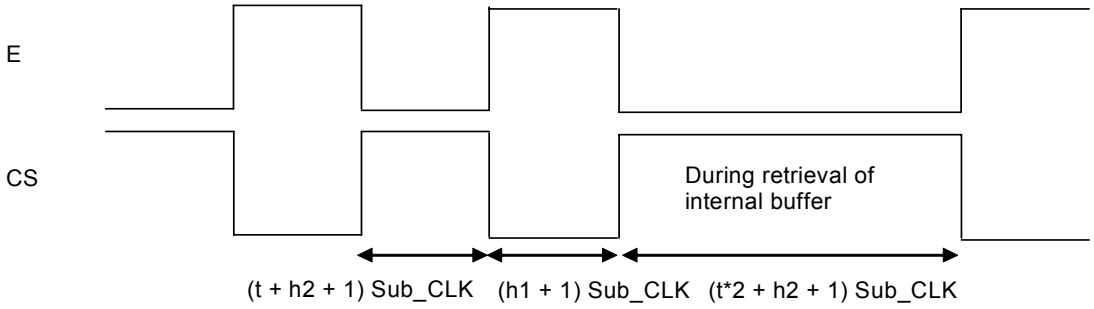
Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Hold count 1 Bit 3	Hold count 1 Bit 2	Hold 1 Bit 1	Hold 1Bit 0
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-4 **Reserved bits**

Bits 3-0 **Hold count register [3:0]**

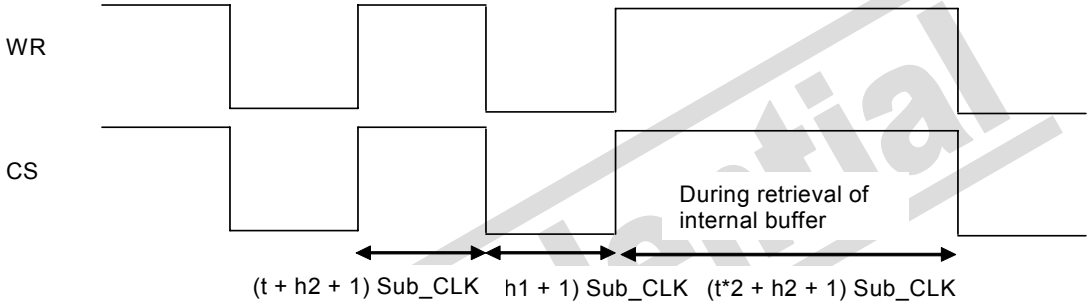
If 6800 interface, Hold count for E low register [3:0]
E will hold low for 3-20 extra clock cycles
If 8080 interface, Hold count for WR high register [3:0]
WR will hold high for 3-20 extra clock cycles
If SPI interface, SCK to CS cycle count register [3:0]
2-17 extra clock cycles for SCK hold high before CS return

For 6800 interface



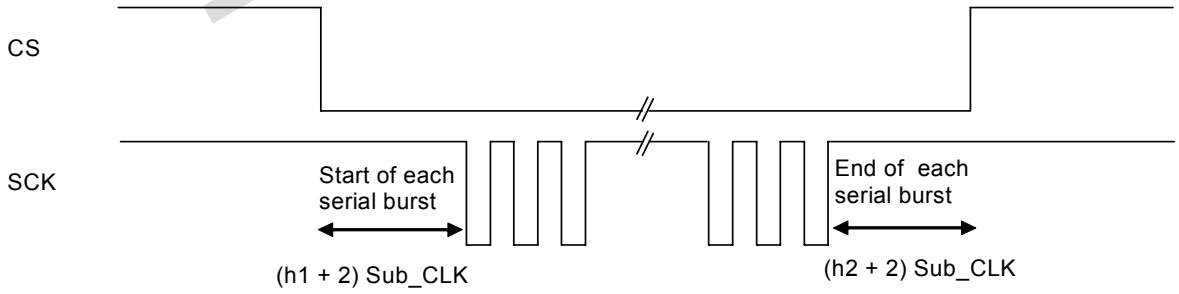
Note :
t = 2
Sub_CLK refer to REG[252] bits 2-0
h1 : Hold count for Count 1, REG[270h] bits 3:0
h2 : Hold count for Count 2, REG[271h] bits 3:0

For 8080 interface



Note :
t = 2
Sub_CLK refer to REG[252] bits 2-0
h1 : Hold count for Count 1, REG[270h] bits 3:0
h2 : Hold count for Count 2, REG[271h] bits 3:0

For SPI interface



Note :
Sub_CLK refer to REG[252] bits 2-0
h1 : Hold count for Count 1, REG[270h] bits 3:0
h2 : Hold count for Count 2, REG[271h] bits 3:0

Smart Display Write Through Mode input data Register 0 **REG[26Ch]**

Bit	7	6	5	4	3	2	1	0
	Smart Display Write Through Mode input data Bit 7	Smart Display Write Through Mode input data Bit 6	Smart Display Write Through Mode input data Bit 5	Smart Display Write Through Mode input data Bit 4	Smart Display Write Through Mode input data Bit 3	Smart Display Write Through Mode input data Bit 2	Smart Display Write Through Mode input data Bit 1	Smart Display Write Through Mode input data Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Smart Display Write Through Mode input data Register 1 **REG[26Dh]**

Bit	7	6	5	4	3	2	1	0
	Smart Display Write Through Mode input data Bit 15	Smart Display Write Through Mode input data Bit 14	Smart Display Write Through Mode input data Bit 13	Smart Display Write Through Mode input data Bit 12	Smart Display Write Through Mode input data Bit 11	Smart Display Write Through Mode input data Bit 10	Smart Display Write Through Mode input data Bit 9	Smart Display Write Through Mode input data Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[26Dh] Bits 7-0 **Write Through Mode input data register [15:0]**
REG[26Ch] Bits 7-0 NB : for byte access,
Please update REG[26Ch], then REG[26Dh]

Smart Display Output Data Format Register **REG[26Eh]**

Bit	7	6	5	4	3	2	1	0
	Pixel data/Argument Selection	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Data/Command Selection
Type	RW	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

REG[26Eh] Bit 7 **Pixel data/Argument Selection**
1 : Pixel Data
0 : Argument
Note :
This bit is effective for Write Through mode only.

REG[26Eh] Bits 6-1 **Reserved bits**
These bits should be programmed by 0.

REG[26Eh] Bit 0 **Data/Command Selection**
1 : Data
0 : Command
Note :
This bit is effective for Write Through mode only.

Smart Display Ready Register **REG[27Dh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Smart Display ready

Bit	7	6	5	4	3	2	1	0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG[27Dh] Bit 0 **Smart Display ready register**
0 : idle
1 : busy

2.1.6 Interrupt Registers

Interrupt Status Flag Register **REG[48h]**

Bit	7	6	5	4	3	2	1	0
	SDHC Interrupt Status Flag	0	I2C Interrupt Status Flag	0	0	JPEG Interrupt Status Flag	DRAW2D Interrupt Status Flag	Reserved
Type	RW	RW	RW	RO	RO	RW	RW	R
Reset state	0	0	0	0	0	0	0	0

Bit 7 **SDHC Interrupt Status Flag**
When this bit = 1, SDHC Interrupt Status detected. Write 1 to clear the status flag. Write 0 has no hardware effect.
When this bit = 0, SDHC Interrupt Status is not detected.

Bits 6, 4, 3, 0 **Reserved bits**

Bit 5 **I2C Interrupt Status Flag**
When this bit = 1, I2C Interrupt Status detected. Write 1 to clear the status flag. Write 0 has no hardware effect.
When this bit = 0, I2C Interrupt Status is not detected.

Bit 2 **JPEG Interrupt Status Flag**
When this bit = 1, JPEG Interrupt Status detected. Write 1 to clear the status flag. Write 0 has no hardware effect.
When this bit = 0, JPEG Interrupt Status is not detected.

Bit 1 **DRAW2D Interrupt Status Flag**
When this bit = 1, DRAW2D Interrupt Status detected. Write 1 to clear the status flag. Write 0 has no hardware effect.
When this bit = 0, DRAW2D Interrupt Status is not detected.

Interrupt Enable Register **REG[4Ah]**

Bit	7	6	5	4	3	2	1	0
	SDHC Interrupt Enable	0	I2C Interrupt Enable	0	0	JPEG Interrupt Enable	DRAW2D Interrupt Enable	Reserved
Type	RW	RW	RW	RO	RO	RW	RW	RO
Reset state	0	0	0	0	0	0	0	0

Bit 7 **SDHC Interrupt Enable**
When this bit = 1, SDHC Interrupt Enable.
When this bit = 0, SDHC Interrupt Disable.

Bits 6, 4, 3, 0 **Reserved bits**
These bits should be programmed as 0

Bit 5 **I2C Interrupt Enable**
When this bit = 1, I2C Interrupt Enable.
When this bit = 0, I2C Interrupt Disable.

Bit 2 **JPEG Interrupt Enable**
When this bit = 1, JPEG Interrupt Enable.
When this bit = 0, JPEG Interrupt Disable.

Bit 1 **DRAW2D Interrupt Enable**
When this bit = 1, DRAW2D Interrupt Enable.
When this bit = 0, DRAW2D Interrupt Disable.

2.1.7 Power Up Registers

Power Saving Configuration Register **REG[A0h]**

Bit	7	6	5	4	3	2	1	0
	Vertical Non-Display Period Status	0	0	Memory Controller Power Saving Status Bit	Display Power Saving Status Bit	Power Saving Mode Enable Bit 2	Power Saving Mode Enable Bit 1	Power Saving Mode Enable Bit 0
Type	RO	NA	RO	RO	RO	RW	RW	RW
Reset state	1	0	0	0	0	0	0	1

Bit 7 **Vertical Non-Display Period Status**
When this bit = 0, the LCD panel is in Vertical Display Period.
When this bit = 1, the LCD panel is in Vertical Non-Display Period.

Bits 6-5 **Reserved bits**

Bit 4 **Memory Controller Power Saving Status Bit**
This bit indicates the Power Saving status of the Main Memory
When this bit = 0, the Main memory is on
When this bit = 1, the Main memory is off

Bit 3 **Display Power Saving Status Bit**
This bit indicates the Power Saving status of the Main display
When this bit = 0, the Main display is on
When this bit = 1, the Main display is off

Bit 2 **Power Saving Mode Enable Bit 2**
This bit control MCLK generation
When this bit = 1, all MCLK will be off.
When this bit = 0, all MCLK will be on.

Bit 1 **Power Saving Mode Enable Bit 1**
This bit control Display Memory Clock
When this bit = 1, MCLK for display SRAM will be off.
When this bit = 0, MCLK for display SRAM will be on.

Bit 0 **Power Saving Mode Enable Bit 0**
This bit control Main display power save mode.
When this bit = 1, Power Saving mode is enabled.
When this bit = 0, Power Saving mode is disabled.

Note :
Power saving mode sequence :
Set power saving mode bit 0 = 1 -> check the display power saving status bit until = 1 -> Set power saving mode bits 2:1 = 11

Memory can not be access if bits 2:1 = 11

Power save frame count Register **REG[A1h]**

Bit	7	6	5	4	3	2	1	0
	Power save frame count Bit 7	Power save frame count Bit 6	Power save frame count Bit 5	Power save frame count Bit 4	Power save frame count Bit 3	Power save frame count Bit 2	Power save frame count Bit 1	Power save frame count Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **Power save frame count Bits [7:0]**
These bit control main panel switch count once power save is enabled (REG[A0h] bit 0 = 1).
0 : Switch off immediately
n : Switch off at the end of n – 1 frames

Software Reset Register **REG[A2h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Software Reset
Type	RW	RW	RW	RW	RW	RW	RW	WO
Reset state	0	0	0	0	0	0	0	0

Bit 0 **Software Reset**
When a one is written to this bit, **the SSD1928 registers are reset**. This bit has no effect on the contents of the display buffer.

2.1.8 Display Mode Registers

STN Color Depth Control Register **REG[45h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	STN Color Depth Control / FRC Bit Select Bit 0
Type	NA	NA	NA	NA	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 0 **STN Color Depth control**
This bit controls the maximum number of color available for STN panels.

When this bit = 0, it allows maximum 32k color depth.
When this bit = 1, it allows maximum 256k color depth.

Refer Table 2-19: LCD Bit-per-pixel Selection for the color depth relationship.

Note
This register is effective for STN panel only (REG[10h] bits 2:0 = 000).
This register can be reset by the RESET signal pin only.

Dithering / FRC Control Register **REG[50h]**

Bit	7	6	5	4	3	2	1	0
	Dynamic Dithering Enable	0	0	FRC Seed Rotate Enable	Reserved	Reserved	FRC Period Select Bit 1	FRC Period Select Bit 0
Type	RW	NA	NA	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	1	0

Bit 7 **Dynamic Dithering Enable**
This bit will enable the dynamic dithering, the dithering mask will change after each 16 frames.

When this bit = 0, dynamic dithering is disabled.
When this bit = 1, dynamic dithering is enabled.

Note
This register is effective for both STN panel and dithering enabled (REG[10h] bits 2:0 = 000 and REG[70h] bit 6 = 0).

Bit 4 **FRC Seed Rotate Enable**
1 – Enable
0 – Disable

Bits 3, 2 **Reserved bits**
These bits should be programmed by 0.

Bits 1:0 **FRC Period Select**
00 – 14 frames
01 – 15 frames
10 – 16 frames
11 – 17 frames

Display Mode Register **REG[70h]**

Bit	7	6	5	4	3	2	1	0
	Display Blank	Dithering Disable	0	Software Color Invert	0	Bit-per-pixel Select Bit 2	Bit-per-pixel Select Bit 1	Bit-per-pixel Select Bit 0
Type	RW	RW	RW	RW	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **Display Blank**
When this bit = 0, the LCD display output is enabled.
When this bit = 1, the LCD display output is blank and all LCD data outputs are forced to zero (i.e., the screen is blanked).

Bit 6 **Dithering Disable**
SSD1928 use a combination of FRC and 4 pixel square formation dithering to achieve more colors per pixel.

In 256K mode (REG[45h] bit 0 = 1), the 4-bit MSB will be considered as FRC and the 2-bit LSB will be dithering. In the 32K mode (REG[45h] bit 0 = 0), the 3-bit MSB will be considered as FRC, the next 2-bit MSB will be dithering and the last LSB will be neglected.

When this bit = 0, dithering is enabled on the passive LCD panel. It allows maximum 64 intensity levels for each color component (RGB).
When this bit = 1, dithering is disabled on the passive LCD panel. It allows maximum 16 intensity levels for each color component (RGB).

Note
This bit does not refer to the number of simultaneously displayed colors but rather the maximum available colors (refer Table 2-19: LCD Bit-per-pixel Selection for the maximum number of displayed colors).

Bits 5, 3 **Reserved bits**

Bit 4 **Software Color Invert**
When this bit = 0, display color is normal.
When this bit = 1, display color is inverted.
This bit has no effect if REG[70h] bit 7 = 1.

Note
Display color is inverted after the Look-Up Table.

Bits 2-0 **Bit-per-pixel Select Bits [2:0]**
These bits select the color depth (bit-per-pixel) for the displayed data for both the main window and the floating window (if active).

Note

1, 2, 4 and 8 bpp modes use three 8-bit LUTs, allowing maximum 256 colors. 16 and 32 bpp mode bypasses the LUT, allowing 64K and 16M colors respectively.

Table 2-19: LCD Bit-per-pixel Selection

Bit-per-pixel Select Bits [2:0]	Color Depth (bpp)	Maximum Number of Colors/Shades			Max. No. Of Simultaneously Displayed Colors/Shades
		Passive Panel (Dithering On)		TFT Panel	
		REG[45h] bit 0 = 0	REG[45h] bit 0 = 1		
000	1 bpp	32K/32	256K/64	256K/64	2/2
001	2 bpp	32K/32	256K/64	256K/64	4/4
010	4 bpp	32K/32	256K/64	256K/64	16/16
011	8 bpp	32K/32	256K/64	256K/64	256/64
100	16 bpp	32K/32	64K/64	64K/64	64K/64
101	32 bpp	32K/32	256K/64	16M/256	16M/256
110, 111	Reserved	n/a	n/a	n/a	n/a

TFT FRC Enable Bit Register **REG[346h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TFT FRC Enable Bit
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

REG[346h] bits 7-1 **Reserved bits**
These bits should be programmed by 0.

REG[346h] bit 0 **TFT FRC Enable Bit Register**
SSD1928 use 2-bits Frame Rate Control (FRC) to achieve more colors for TFT. So 2 LSB are used for each RGB color component.

TFT FRC will be enabled if this bit is set to 1.

Note : This register is effective for 9/12/18 bit TFT only.

2.1.9 Main Window Registers

Main Window Display Start Address Register 0 **REG[74h]**

Bit	7	6	5	4	3	2	1	0
	Main window Display Start Address Bit 7	Main window Display Start Address Bit 6	Main window Display Start Address Bit 5	Main window Display Start Address Bit 4	Main window Display Start Address Bit 3	Main window Display Start Address Bit 2	Main window Display Start Address Bit 1	Main window Display Start Address Bit 0

Bit	7	6	5	4	3	2	1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Main Window Display Start Address Register 1 **REG[75h]**

Bit	7	6	5	4	3	2	1	0
	Main window Display Start Address Bit 15	Main window Display Start Address Bit 14	Main window Display Start Address Bit 13	Main window Display Start Address Bit 12	Main window Display Start Address Bit 11	Main window Display Start Address Bit 10	Main window Display Start Address Bit 9	Main window Display Start Address Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Main Window Display Start Address Register 2 **REG[76h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Main window Display Start Address Bit 16
Type	NA	NA	NA	NA	NA	NA	NA	RW
Reset state	0	0	0	0	0	0	0	0

REG[76h] bit 0,
REG[75h] bits 7-0,
REG[74h] bits 7-0

Main Window Display Start Address Bits [16:0]
These bits form the 17-bit address for the starting double-word of the LCD image in the display buffer for the main window.

Note that this is a double-word (32-bit) address. An entry of 00000h into these registers represents the first double-word of display memory, an entry of 00001h represents the second double-word of the display memory, and so on.

Calculate the Display Start Address as follows :

Main Window Display Start Address Bits 16:0
= Image address ÷ 4 (valid only for Display Rotate Mode 0°)

Note
(1) For information on setting this register for other Display Rotate Mode, see Section "Display Rotate Mode" in datasheet.

Main Window Line Address Offset Register 0 **REG[78h]**

Bit	7	6	5	4	3	2	1	0
	Main window Line Address Offset Bit 7	Main window Line Address Offset Bit 6	Main window Line Address Offset Bit 5	Main window Line Address Offset Bit 4	Main window Line Address Offset Bit 3	Main window Line Address Offset Bit 2	Main window Line Address Offset Bit 1	Main window Line Address Offset Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Main Window Line Address Offset Register 1 **REG[79h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Main window Line Address Offset Bit 9	Main window Line Address Offset Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[79h] bits 1-0,
REG[78h] bits 7-0

Main Window Line Address Offset Bits [9:0]
This register specifies the offset, in double words, from the beginning of one display line to the beginning of the next display line in the main window. **Note that this is a 32-bit address increment.**

Calculate the Line Address Offset as follows :
Main Window Line Address Offset bits 9-0
= Display Width in pixels ÷ (32 ÷ bpp)

Note
(1) A virtual display can be created by programming this register with a value greater than the formula requires. When a virtual display is created the image width is larger than the display width and the displayed image becomes a window into the larger virtual image.

2.1.10 Scratch bit Registers

Scratch bit Register 0 **REG[A4h]**

Bit	7	6	5	4	3	2	1	0
	Scratch Pad Bit 7	Scratch Pad Bit 6	Scratch Pad Bit 5	Scratch Pad Bit 4	Scratch Pad Bit 3	Scratch Pad Bit 2	Scratch Pad Bit 1	Scratch Pad Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Scratch bit Register 1 **REG[A5h]**

Bit	7	6	5	4	3	2	1	0
	Scratch Pad Bit 15	Scratch Pad Bit 14	Scratch Pad Bit 13	Scratch Pad Bit 12	Scratch Pad Bit 11	Scratch Pad Bit 10	Scratch Pad Bit 9	Scratch Pad Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[A5h] bits 7-0,
REG[A4h] bits 7-0

Scratch Pad Bits [15:0]
This register contains general purpose read/write bits. These bits have no effect on hardware configuration.

2.1.11 General IO Pins Registers

General Purpose I/O Pins Configuration Register 0 **REG[A8h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	GPIO4 I/O Configuration	GPIO3 I/O Configuration	GPIO2 I/O Configuration	GPIO1 I/O Configuration	GPIO0 I/O Configuration
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5 **Reserved bits**

Bit 4 **GPIO4 I/O Configuration**

When this bit = 0, GPIO4 is configured as an input pin.
When this bit = 1, GPIO4 is configured as an output pin.

Bit 3 **GPIO3 I/O Configuration**
When this bit = 0, GPIO3 is configured as an input pin.
When this bit = 1, GPIO3 is configured as an output pin.

Bit 2 **GPIO2 I/O Configuration**
When this bit = 0, GPIO2 is configured as an input pin.
When this bit = 1, GPIO2 is configured as an output pin.

Bit 1 **GPIO1 I/O Configuration**
When this bit = 0, GPIO1 is configured as an input pin.
When this bit = 1, GPIO1 is configured as an output pin.

Bit 0 **GPIO0 I/O Configuration**
When this bit = 0, GPIO0 is configured as an input pin.
When this bit = 1, GPIO0 is configured as an output pin.

Note
(1) The input functions of the GPIO pins are not enabled until REG[A9h] bit 7 is set to 1.

General Purpose IO Pins Configuration Register 1 **REG[A9h]**

Bit	7	6	5	4	3	2	1	0
	GPIO Pin Input Enable	0	0	0	0	0	0	0
Type	RW	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 7 **GPIO Pin Input Enable**
This bit is used to enable the input function of the GPIO[4:0] pins. It must be changed to a 1 after power-on reset to enable the input function of the GPIO[4:0] pins.

General Purpose IO Pins Status/Control Register **REG[ACh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	GPIO4 Pin IO Status	GPIO3 Pin IO Status	GPIO2 Pin IO Status	GPIO1 Pin IO Status	GPIO0 Pin IO Status
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5 **Reserved bits**

Bit 4 **GPIO4 Pin IO Status**
When GPIO4 is configured as an output, writing a 1 to this bit drives GPIO4 high and writing a 0 to this bit drives GPIO4 low or controlled by LCD Gen4 (REG[334h]).

When GPIO4 is configured as an input, a read from this bit returns the status of GPIO4.

Bit 3 **GPIO3 Pin IO Status**
When GPIO3 is configured as an output, writing a 1 to this bit drives GPIO3 high and writing a 0 to this bit drives GPIO3 low or controlled by LCD Gen3 (REG[333h]).

When GPIO3 is configured as an input, a read from this bit returns the status of GPIO3.

Bit 2 **GPIO2 Pin IO Status**
When GPIO2 is configured as an output, writing a 1 to this bit drives GPIO2 high and writing a 0 to this bit drives GPIO2 low or controlled by LCD Gen2 (REG[332h]).
When GPIO2 is configured as an input, a read from this bit returns the status of GPIO2.

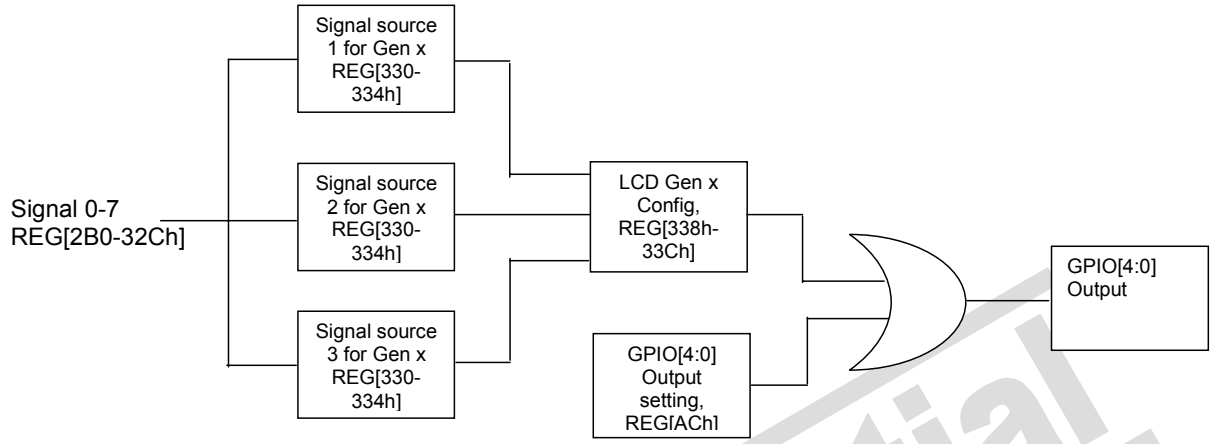
Bit 1 **GPIO1 Pin IO Status**
When GPIO1 is configured as an output, writing a 1 to this bit drives GPIO1 high and writing a 0 to this bit drives GPIO1 low or controlled by LCD Gen1 (REG[331h]).

When GPIO1 is configured as an input, a read from this bit returns the status of GPIO1.

Bit 0 **GPIO0 Pin IO Status**
When GPIO0 is configured as an output, writing a 1 to this bit drives GPIO0 high and writing a 0 to this bit drives GPIO0 low or controlled by LCD Gen0 (REG[330h]).

When GPIO0 is configured as an input, a read from this bit returns the status of GPIO0.

Figure 2-19: GPIO[4:0] output setup



LCD Power Control Register **REG[ADh]**

Bit	7	6	5	4	3	2	1	0
	LPOWER Control	0	0	0	0	0	0	0
Type	RW	NA	NA	NA	NA	NA	NA	NA
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LPOWER Control**
This bit controls the General Purpose Output pin.
Writing a 0 to this bit drives LPOWER to low.
Writing a 1 to this bit drives LPOWER to high.

Note
(1) Many implementations use the LPOWER pin to control the LCD bias power (see Section "LCD Power Sequencing" in datasheet).

LCD Signal0 Rise Location Register 0 **REG[2B0h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal0 Rise Location Bit 7	LCD Signal0 Rise Location Bit 6	LCD Signal0 Rise Location Bit 5	LCD Signal0 Rise Location Bit 4	LCD Signal0 Rise Location Bit 3	LCD Signal0 Rise Location Bit 2	LCD Signal0 Rise Location Bit 1	LCD Signal0 Rise Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal0 0 Rise Location Register 1 **REG[2B1h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal0 Rise Location Bit 10	LCD Signal0 Rise Location Bit 9	LCD Signal0 Rise Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2B1h] bits 2-0, REG[2B0h] bits 7-0 **LCD Signal0 Rise position Register [10:0]**
This register define the lcd Signal0 rise position.

LCD Signal0 Fall Location Register 0 **REG[2B4h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal0 Fall Location Bit 7	LCD Signal0 Fall Location Bit 6	LCD Signal0 Fall Location Bit 5	LCD Signal0 Fall Location Bit 4	LCD Signal0 Fall Location Bit 3	LCD Signal0 Fall Location Bit 2	LCD Signal0 Fall Location Bit 1	LCD Signal0 Fall Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal0 Fall Location Register 1 **REG[2B5h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal0 Fall Location Bit 10	LCD Signal0 Fall Location Bit 9	LCD Signal0 Fall Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2B5h] bits 2-0, REG[2B4h] bits 7-0 **LCD Signal0 Fall position Register [10:0]**
This register define the lcd Signal0 Fall position.

LCD Signal0 Period Location Register 0 **REG[2B8h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal0 Period Location Bit 7	LCD Signal0 Period Location Bit 6	LCD Signal0 Period Location Bit 5	LCD Signal0 Period Location Bit 4	LCD Signal0 Period Location Bit 3	LCD Signal0 Period Location Bit 2	LCD Signal0 Period Location Bit 1	LCD Signal0 Period Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal0 Period Location Register 1 **REG[2B9h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal0 Period Location Bit 10	LCD Signal0 Period Location Bit 9	LCD Signal0 Period Location Bit 8

Bit	7	6	5	4	3	2	1	0
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2B9h] bits 2-0,
REG[2B8h] bits 7-0

LCD Signal0 Period position Register [10:0]
This register define the lcd Signal0 Period position.

Note
(1) This register is ignored if toggle by frame (REG[2BCh] bit 1:0 = 11).

LCD Signal0 Control Register **REG[2BCh]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal0 Reset Bit	LCD Signal0 NDPOFF	LCD Signal0 Odd / Even Bit 1	LCD Signal0 Odd / Even Bit 0	Reserved	Reserved	LCD Signal0 Toggle Bit 1	LCD Signal0 Toggle Bit 0
Type	RW	RW	RW	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LCD Signal0 Reset Register**
This LCD Signal0 will be under frame reset state when this reset bit is set to 1.

Bit 6 **LCD Signal0 NDPOFF**
This bit enables signal output in non-display period.
0 : enable signal in non-display period
1 : disable signal in non-display period

Note
(1) This bit effective only when LCD signal toggle by PCLK.

Bits 5-4 **LCD Singal0 Odd / Even [1:0]**
These bits enable signal output in odd or even lines
00/11 : enable signal in all lines
01 : enable signal in even lines
10 : enable signal in odd lines

Note
(1) These bits effective only when NDPOFF = 1 and LCD signal toggle by PCLK.

Bits 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **LCD Signal0 Toggle Register [1:0]**
00 : Disable
01 : toggle by PCLK
10 : toggle by Line
11 : toggle by Frame

LCD Signal1 Rise Location Register 0 **REG[2C0h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal1 Rise Location Bit 7	LCD Signal1 Rise Location Bit 6	LCD Signal1 Rise Location Bit 5	LCD Signal1 Rise Location Bit 4	LCD Signal1 Rise Location Bit 3	LCD Signal1 Rise Location Bit 2	LCD Signal1 Rise Location Bit 1	LCD Signal1 Rise Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal1 Rise Location Register 1 **REG[2C1h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal1 Rise Location Bit 10	LCD Signal1 Rise Location Bit 9	LCD Signal1 Rise Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2C1h] bits 2-0, REG[2C0h] bits 7-0 **LCD Signal1 Rise position Register [10:0]**
This register define the lcd Signal1 rise position.

LCD Signal1 Fall Location Register 0 **REG[2C4h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal1 Fall Location Bit 7	LCD Signal1 Fall Location Bit 6	LCD Signal1 Fall Location Bit 5	LCD Signal1 Fall Location Bit 4	LCD Signal1 Fall Location Bit 3	LCD Signal1 Fall Location Bit 2	LCD Signal1 Fall Location Bit 1	LCD Signal1 Fall Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal1 Fall Location Register 1 **REG[2C5h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal1 Fall Location Bit 10	LCD Signal1 Fall Location Bit 9	LCD Signal1 Fall Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2C5h] bits 2-0, REG[2C4h] bits 7-0 **LCD Signal1 Fall position Register [10:0]**
This register define the lcd Signal1 Fall position.

LCD Signal1 Period Location Register 0 **REG[2C8h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal1 Period Location Bit 7	LCD Signal1 Period Location Bit 6	LCD Signal1 Period Location Bit 5	LCD Signal1 Period Location Bit 4	LCD Signal1 Period Location Bit 3	LCD Signal1 Period Location Bit 2	LCD Signal1 Period Location Bit 1	LCD Signal1 Period Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal1 Period Location Register 1 **REG[2C9h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal1 Period Location Bit 10	LCD Signal1 Period Location Bit 9	LCD Signal1 Period Location Bit 8

Bit	7	6	5	4	3	2	1	0
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2C9h] bits 2-0, REG[2C8h] bits 7-0 **LCD Signal1 Period position Register [10:0]**
This register define the lcd Signal1Period position.

Note
(1) This register is ignored if toggle by frame (REG[2CCh] bit 1:0 = 11).

LCD Signal1 Control Register **REG[2CCh]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal1 Reset Bit	LCD Signal1 NDPOFF	LCD Signal1 Odd / Even Bit 1	LCD Signal1 Odd / Even Bit 0	Reserved	Reserved	LCD Signal1 Toggle Bit 1	LCD Signal1 Toggle Bit 0
Type	RW	RW	RW	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LCD Signal1 Reset Register**
This LCD Signal1 will be under reset state when this reset bit is set to 1.

Bit 6 **LCD Signal1 NDPOFF**
This bit enables signal output in non-display period.
0 : enable signal in non-display period
1 : disable signal in non-display period

Note
(1) This bit effective only when LCD signal toggle by PCLK.

Bits 5-4 **LCD Singal1 Odd / Even [1:0]**
These bits enable signal output in odd or even lines
00/11 : enable signal in all lines
01 : enable signal in even lines
10 : enable signal in odd lines

Note
(1) These bits effective only when NDPOFF = 1 and LCD signal toggle by PCLK.

Bits 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **LCD Signal1 Toggle Register [1:0]**
00 : Disable
01 : toggle by PCLK
10 : toggle by Line
11 : toggle by Frame

LCD Signal2 Rise Location Register 0 **REG[2D0h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal2 Rise Location Bit 7	LCD Signal2 Rise Location Bit 6	LCD Signal2 Rise Location Bit 5	LCD Signal2 Rise Location Bit 4	LCD Signal2 Rise Location Bit 3	LCD Signal2 Rise Location Bit 2	LCD Signal2 Rise Location Bit 1	LCD Signal2 Rise Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal2 0 Rise Location Register 1 **REG[2D1h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal2 Rise Location Bit 10	LCD Signal2 Rise Location Bit 9	LCD Signal2 Rise Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2D1h] bits 2-0, REG[2D0h] bits 7-0 **LCD Signal2 Rise position Register [10:0]**
This register define the lcd Signal2 rise position.

LCD Signal2 Fall Location Register 0 **REG[2D4h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal2 Fall Location Bit 7	LCD Signal2 Fall Location Bit 6	LCD Signal2 Fall Location Bit 5	LCD Signal2 Fall Location Bit 4	LCD Signal2 Fall Location Bit 3	LCD Signal2 Fall Location Bit 2	LCD Signal2 Fall Location Bit 1	LCD Signal2 Fall Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal2 Fall Location Register 1 **REG[2D5h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal2 Fall Location Bit 10	LCD Signal2 Fall Location Bit 9	LCD Signal2 Fall Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2D5h] bits 2-0, REG[2D4h] bits 7-0 **LCD Signal2 Fall position Register [10:0]**
This register define the lcd Signal2 Fall position.

LCD Signal2 Period Location Register 0 **REG[2D8h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal2 Period Location Bit 7	LCD Signal2 Period Location Bit 6	LCD Signal2 Period Location Bit 5	LCD Signal2 Period Location Bit 4	LCD Signal2 Period Location Bit 3	LCD Signal2 Period Location Bit 2	LCD Signal2 Period Location Bit 1	LCD Signal2 Period Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal2 Period Location Register 1 **REG[2D9h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal2 Period Location Bit 10	LCD Signal2 Period Location Bit 9	LCD Signal2 Period Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
state								

REG[2D9h] bits 2-0, REG[2D8h] bits 7-0 **LCD Signal2 Period position Register [10:0]**
This register define the lcd Signal2 Period position.

Note
(1) This register is ignored if toggle by frame (REG[2DCh] bit 1:0 = 11).

LCD Signal2 Control Register **REG[2DCh]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal2 Reset Bit	LCD Signal2 NDPOFF	LCD Signal2 Odd / Even Bit 1	LCD Signal2 Odd / Even Bit 0	Reserved	Reserved	LCD Signal2 Toggle Bit 1	LCD Signal2 Toggle Bit 0
Type	RW	RW	RW	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LCD Signal2 Reset Register**
This LCD Signal2 will be under reset state when this reset bit is set to 1.

Bit 6 **LCD Signal2 NDPOFF**
This bit enables signal output in non-display period.
0 : enable signal in non-display period
1 : disable signal in non-display period

Note
(1) This bit effective only when LCD signal toggle by PCLK.

Bits 5-4 **LCD Singal2 Odd / Even [1:0]**
These bits enable signal output in odd or even lines
00/11 : enable signal in all lines
01 : enable signal in even lines
10 : enable signal in odd lines

Note
(1) These bits effective only when NDPOFF = 1 and LCD signal toggle by PCLK.

Bits 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **LCD Signal2 Toggle Register [1:0]**
00 : Disable
01 : toggle by PCLK
10 : toggle by Line
11 : toggle by Frame

LCD Signal3 Rise Location Register 0 **REG[2E0h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal3 Rise Location Bit 7	LCD Signal3 Rise Location Bit 6	LCD Signal3 Rise Location Bit 5	LCD Signal3 Rise Location Bit 4	LCD Signal3 Rise Location Bit 3	LCD Signal3 Rise Location Bit 2	LCD Signal3 Rise Location Bit 1	LCD Signal3 Rise Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal3 0 Rise Location Register 1 **REG[2E1h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal3 Rise Location Bit 10	LCD Signal3 Rise Location Bit 9	LCD Signal3 Rise Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2E1h] bits 2-0, REG[2E0h] bits 7-0 **LCD Signal3 Rise position Register [10:0]**
This register define the lcd Signal3 rise position.

LCD Signal3 Fall Location Register 0 **REG[2E4h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal3 Fall Location Bit 7	LCD Signal3 Fall Location Bit 6	LCD Signal3 Fall Location Bit 5	LCD Signal3 Fall Location Bit 4	LCD Signal3 Fall Location Bit 3	LCD Signal3 Fall Location Bit 2	LCD Signal3 Fall Location Bit 1	LCD Signal3 Fall Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal3 Fall Location Register 1 **REG[2E5h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal3 Fall Location Bit 10	LCD Signal3 Fall Location Bit 9	LCD Signal3 Fall Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2E5h] bits 2-0, REG[2E4h] bits 7-0 **LCD Signal3 Fall position Register [10:0]**
This register define the lcd Signal3 fall location.

LCD Signal3 Period Location Register 0 **REG[2E8h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal3 Period Location Bit 7	LCD Signal3 Period Location Bit 6	LCD Signal3 Period Location Bit 5	LCD Signal3 Period Location Bit 4	LCD Signal3 Period Location Bit 3	LCD Signal3 Period Location Bit 2	LCD Signal3 Period Location Bit 1	LCD Signal3 Period Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal3 Period Location Register 1 **REG[2E9h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal3 Period Location Bit 10	LCD Signal3 Period Location Bit 9	LCD Signal3 Period Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW

Bit	7	6	5	4	3	2	1	0
Reset state	0	0	0	0	0	0	0	0

REG[2E9h] bits 2-0,
REG[2E8h] bits 7-0

LCD Signal3 Period position Register [10:0]
This register define the lcd Signal3 Period position.

Note
(1) This register is ignored if toggle by frame (REG[2ECh] bit 1:0 = 11).

LCD Signal3 Control Register **REG[2ECh]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal3 Reset Bit	LCD Signal3 NDPOFF	LCD Signal3 Odd / Even Bit 1	LCD Signal3 Odd / Even Bit 0	Reserved	Reserved	LCD Signal3 Toggle Bit 1	LCD Signal3 Toggle Bit 0
Type	RW	RW	RW	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LCD Signal3 Reset Register**
This LCD Signal3 will be under reset state when this reset bit is set to 1.

Bit 6 **LCD Signal3 NDPOFF**
This bit enables signal output in non-display period.
0 : enable signal in non-display period
1 : disable signal in non-display period

Note
(1) This bit effective only when LCD signal toggle by PCLK.

Bits 5-4 **LCD Singal3 Odd / Even [1:0]**
These bits enable signal output in odd or even lines
00/11 : enable signal in all lines
01 : enable signal in even lines
10 : enable signal in odd lines

Note
(1) These bits effective only when NDPOFF = 1 and LCD signal toggle by PCLK.

Bits 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **LCD Signal3 Toggle Register [1:0]**
00 : Disable
01 : toggle by PCLK
10 : toggle by Line
11 : toggle by Frame

LCD Signal4 Rise Location Register 0 **REG[2F0h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal4 Rise Location Bit 7	LCD Signal4 Rise Location Bit 6	LCD Signal4 Rise Location Bit 5	LCD Signal4 Rise Location Bit 4	LCD Signal4 Rise Location Bit 3	LCD Signal4 Rise Location Bit 2	LCD Signal4 Rise Location Bit 1	LCD Signal4 Rise Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal4 Rise Location Register 1 **REG[2F1h]**

Bit	7	6	5	4	3	2	1	0

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal4 Rise Location Bit 10	LCD Signal4 Rise Location Bit 9	LCD Signal4 Rise Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2F1h] bits 2-0, REG[2F0h] bits 7-0 **LCD Signal4 Rise position Register [10:0]**
This register define the lcd Signal4 rise position.

LCD Signal4 Fall Location Register 0 **REG[2F4h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal4 Fall Location Bit 7	LCD Signal4 Fall Location Bit 6	LCD Signal4 Fall Location Bit 5	LCD Signal4 Fall Location Bit 4	LCD Signal4 Fall Location Bit 3	LCD Signal4 Fall Location Bit 2	LCD Signal4 Fall Location Bit 1	LCD Signal4 Fall Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal4 Fall Location Register 1 **REG[2F5h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal4 Fall Location Bit 10	LCD Signal4 Fall Location Bit 9	LCD Signal4 Fall Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2F5h] bits 2-0, REG[2F4h] bits 7-0 **LCD Signal4 Fall position Register [10:0]**
This register define the lcd Signal4 Fall position.

LCD Signal4 Period Location Register 0 **REG[2F8h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal4 Period Location Bit 7	LCD Signal4 Period Location Bit 6	LCD Signal4 Period Location Bit 5	LCD Signal4 Period Location Bit 4	LCD Signal4 Period Location Bit 3	LCD Signal4 Period Location Bit 2	LCD Signal4 Period Location Bit 1	LCD Signal4 Period Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal4 Period Location Register 1 **REG[2F9h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal4 Period	LCD Signal4 Period	LCD Signal4 Period

Bit	7	6	5	4	3	2	1	0
						Location Bit 10	Location Bit 9	Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[2F9h] bits 2-0, REG[2F8h] bits 7-0

LCD Signal4 Period position Register [10:0]
This register define the lcd Signal4 Period position.

Note
(1) This register is ignored if toggle by frame (REG[2FCh] bit 1:0 = 11).

LCD Signal4 Control Register **REG[2FCh]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal4 Reset Bit	LCD Signal4 NDPOFF	LCD Signal4 Odd / Even Bit 1	LCD Signal4 Odd / Even Bit 0	Reserved	Reserved	LCD Signal4 Toggle Bit 1	LCD Signal4 Toggle Bit 0
Type	RW	RW	RW	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LCD Signal4 Reset Register**
This LCD Signal4will be under reset state when this reset bit is set to 1.

Bit 6 **LCD Signal4 NDPOFF**
This bit enables signal output in non-display period.
0 : enable signal in non-display period
1 : disable signal in non-display period

Note
(1) This bit effective only when LCD signal toggle by PCLK.

Bits 5-4 **LCD Singal4 Odd / Even [1:0]**
These bits enable signal output in odd or even lines
00/11 : enable signal in all lines
01 : enable signal in even lines
10 : enable signal in odd lines

Note
(1) These bits effective only when NDPOFF = 1 and LCD signal toggle by PCLK.

Bits 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **LCD Signal4 Toggle Register [1:0]**
00 : Disable
01 : toggle by PCLK
10 : toggle by Line
11 : toggle by Frame

LCD Signal5 Rise Location Register 0 **REG[300h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal5 Rise Location Bit 7	LCD Signal5 Rise Location Bit 6	LCD Signal5 Rise Location Bit 5	LCD Signal5 Rise Location Bit 4	LCD Signal5 Rise Location Bit 3	LCD Signal5 Rise Location Bit 2	LCD Signal5 Rise Location Bit 1	LCD Signal5 Rise Location Bit 0

Bit	7	6	5	4	3	2	1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal5 Rise Location Register 1 **REG[301h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal5 Rise Location Bit 10	LCD Signal5 Rise Location Bit 9	LCD Signal5 Rise Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[301h] bits 2-0, REG[300h] bits 7-0 **LCD Signal5 Rise position Register [10:0]**
This register define the lcd Signal5 rise position.

LCD Signal5 Fall Location Register 0 **REG[304h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal5 Fall Location Bit 7	LCD Signal5 Fall Location Bit 6	LCD Signal5 Fall Location Bit 5	LCD Signal5 Fall Location Bit 4	LCD Signal5 Fall Location Bit 3	LCD Signal5 Fall Location Bit 2	LCD Signal5 Fall Location Bit 1	LCD Signal5 Fall Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal5 Fall Location Register 1 **REG[305h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal5 Fall Location Bit 10	LCD Signal5 Fall Location Bit 9	LCD Signal5 Fall Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[305h] bits 2-0, REG[304h] bits 7-0 **LCD Signal5 Fall position Register [10:0]**
This register define the lcd Signal5 Fall position.

LCD Signal5 Period Location Register 0 **REG[308h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal5 Period Location Bit 7	LCD Signal5 Period Location Bit 6	LCD Signal5 Period Location Bit 5	LCD Signal5 Period Location Bit 4	LCD Signal5 Period Location Bit 3	LCD Signal5 Period Location Bit 2	LCD Signal5 Period Location Bit 1	LCD Signal5 Period Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal5 Period Location Register 1 **REG[309h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal5 Period Location Bit 10	LCD Signal5 Period Location Bit 9	LCD Signal5 Period Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[309h] bits 2-0, REG[308h] bits 7-0 **LCD Signal5 Period position Register [10:0]**
This register define the lcd Signal5 Period position.

Note
(1) This register is ignored if toggle by frame (REG[30Ch] bit 1:0 = 11).

LCD Signal5 Control Register **REG[30Ch]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal5 Reset Bit	LCD Signal5 NDPOFF	LCD Signal5 Odd / Even Bit 1	LCD Signal5 Odd / Even Bit 0	Reserved	Reserved	LCD Signal5 Toggle Bit 1	LCD Signal5 Toggle Bit 0
Type	RW	RW	RW	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LCD Signal5 Reset Register**
This LCD Signal5 will be under reset state when this reset bit is set to 1.

Bit 6 **LCD Signal5 NDPOFF**
This bit enables signal output in non-display period.
0 : enable signal in non-display period
1 : disable signal in non-display period

Note
(1) This bit effective only when LCD signal toggle by PCLK.

Bits 5-4 **LCD Singal5 Odd / Even [1:0]**
These bits enable signal output in odd or even lines
00/11 : enable signal in all lines
01 : enable signal in even lines
10 : enable signal in odd lines

Note
(1) These bits effective only when NDPOFF = 1 and LCD signal toggle by PCLK.

Bits 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **LCD Signal5 Toggle Register [1:0]**
00 : Disable
01 : toggle by PCLK
10 : toggle by Line
11 : toggle by Frame

LCD Signal6 Rise Location Register 0 **REG[310h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal6	LCD Signal6	LCD Signal6	LCD Signal6	LCD Signal6	LCD Signal6	LCD Signal6	LCD Signal6

Bit	7	6	5	4	3	2	1	0
	Rise Location Bit 7	Rise Location Bit 6	Rise Location Bit 5	Rise Location Bit 4	Rise Location Bit 3	Rise Location Bit 2	Rise Location Bit 1	Rise Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal6 0 Rise Location Register 1 **REG[311h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal6 Rise Location Bit 10	LCD Signal6 Rise Location Bit 9	LCD Signal6 Rise Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[311h] bits 2-0, REG[310h] bits 7-0 **LCD Signal6 Rise position Register [10:0]**
This register define the lcd Signal6 rise position.

LCD Signal6 Fall Location Register 0 **REG[314h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal6 Fall Location Bit 7	LCD Signal6 Fall Location Bit 6	LCD Signal6 Fall Location Bit 5	LCD Signal6 Fall Location Bit 4	LCD Signal6 Fall Location Bit 3	LCD Signal6 Fall Location Bit 2	LCD Signal6 Fall Location Bit 1	LCD Signal6 Fall Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal6 Fall Location Register 1 **REG[315h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal6 Fall Location Bit 10	LCD Signal6 Fall Location Bit 9	LCD Signal6 Fall Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[315h] bits 2-0, REG[314h] bits 7-0 **LCD Signal6 Fall position Register [10:0]**
This register define the lcd Signal6 Fall position.

LCD Signal6 Period Location Register 0 **REG[318h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal6 Period Location Bit 7	LCD Signal6 Period Location Bit 6	LCD Signal6 Period Location Bit 5	LCD Signal6 Period Location Bit 4	LCD Signal6 Period Location Bit 3	LCD Signal6 Period Location Bit 2	LCD Signal6 Period Location Bit 1	LCD Signal6 Period Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal6 Period Location Register 1 **REG[319h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal6 Period Location Bit 10	LCD Signal6 Period Location Bit 9	LCD Signal6 Period Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[319h] bits 2-0, REG[318h] bits 7-0 **LCD Signal6 Period position Register [10:0]**
This register define the lcd Signal6 Period position.

Note
(1) This register is ignored if toggle by frame (REG[31Ch] bit 1:0 = 11).

LCD Signal6 Control Register **REG[31Ch]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal6 Reset Bit	LCD Signal6 NDPOFF	LCD Signal6 Odd / Even Bit 1	LCD Signal6 Odd / Even Bit 0	Reserved	Reserved	LCD Signal6 Toggle Bit 1	LCD Signal6 Toggle Bit 0
Type	RW	RW	RW	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LCD Signal6 Reset Register**
This LCD Signal6 will be under reset state when this reset bit is set to 1.

Bit 6 **LCD Signal6 NDPOFF**
This bit enables signal output in non-display period.
0 : enable signal in non-display period
1 : disable signal in non-display period

Note
(1) This bit effective only when LCD signal toggle by PCLK.

Bits 5-4 **LCD Singal6 Odd / Even [1:0]**
These bits enable signal output in odd or even lines
00/11 : enable signal in all lines
01 : enable signal in even lines
10 : enable signal in odd lines

Note
(1) These bits effective only when NDPOFF = 1 and LCD signal toggle by PCLK.

Bits 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **LCD Signal6 Toggle Register [1:0]**
00 : Disable
01 : toggle by PCLK
10 : toggle by Line
11 : toggle by Frame

LCD Signal7 Rise Location Register 0 **REG[320h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal7 Rise	LCD Signal7 Rise	LCD Signal7 Rise	LCD Signal7 Rise	LCD Signal7 Rise	LCD Signal7 Rise	LCD Signal7 Rise	LCD Signal7 Rise

Bit	7	6	5	4	3	2	1	0
	Location Bit 7	Location Bit 6	Location Bit 5	Location Bit 4	Location Bit 3	Location Bit 2	Location Bit 1	Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal7 0 Rise Location Register 1 **REG[321h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal7 Rise Location Bit 10	LCD Signal7 Rise Location Bit 9	LCD Signal7 Rise Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[321h] bits 2-0, REG[320h] bits 7-0 **LCD Signal7 Rise position Register [10:0]**
This register define the lcd Signal7 rise position.

LCD Signal7 Fall Location Register 0 **REG[324h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal7 Fall Location Bit 7	LCD Signal7 Fall Location Bit 6	LCD Signal7 Fall Location Bit 5	LCD Signal7 Fall Location Bit 4	LCD Signal7 Fall Location Bit 3	LCD Signal7 Fall Location Bit 2	LCD Signal7 Fall Location Bit 1	LCD Signal7 Fall Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

LCD Signal7 Fall Location Register 1 **REG[325h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal7 Fall Location Bit 10	LCD Signal7 Fall Location Bit 9	LCD Signal7 Fall Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[325h] bits 2-0, REG[324h] bits 7-0 **LCD Signal7 Fall position Register [10:0]**
This register define the lcd Signal7 Fall position.

LCD Signal7 Period Location Register 0 **REG[328h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal7 Period Location Bit 7	LCD Signal7 Period Location Bit 6	LCD Signal7 Period Location Bit 5	LCD Signal7 Period Location Bit 4	LCD Signal7 Period Location Bit 3	LCD Signal7 Period Location Bit 2	LCD Signal7 Period Location Bit 1	LCD Signal7 Period Location Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
state								

LCD Signal7 Period Location Register 1 **REG[329h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	LCD Signal7 Period Location Bit 10	LCD Signal7 Period Location Bit 9	LCD Signal7 Period Location Bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[329h] bits 2-0, REG[328h] bits 7-0

LCD Signal7 Period position Register [10:0]
This register define the lcd Signal7 Period position.

Note
(1) This register is ignored if toggle by frame (REG[32Ch] bit 1:0 = 11).

LCD Signal7 Control Register **REG[32Ch]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal7 Reset Bit	LCD Signal7 NDPOFF	LCD Signal7 Odd / Even Bit 1	LCD Signal7 Odd / Even Bit 0	Reserved	Reserved	LCD Signal7 Toggle Bit 1	LCD Signal7 Toggle Bit 0
Type	RW	RW	RW	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **LCD Signal7 Reset Register**
This LCD Signal7 will be under reset state when this reset bit is set to 1.

Bit 6 **LCD Signal7 NDPOFF**
This bit enables signal output in non-display period.
0 : enable signal in non-display period
1 : disable signal in non-display period

Note
(1) This bit effective only when LCD signal toggle by PCLK.

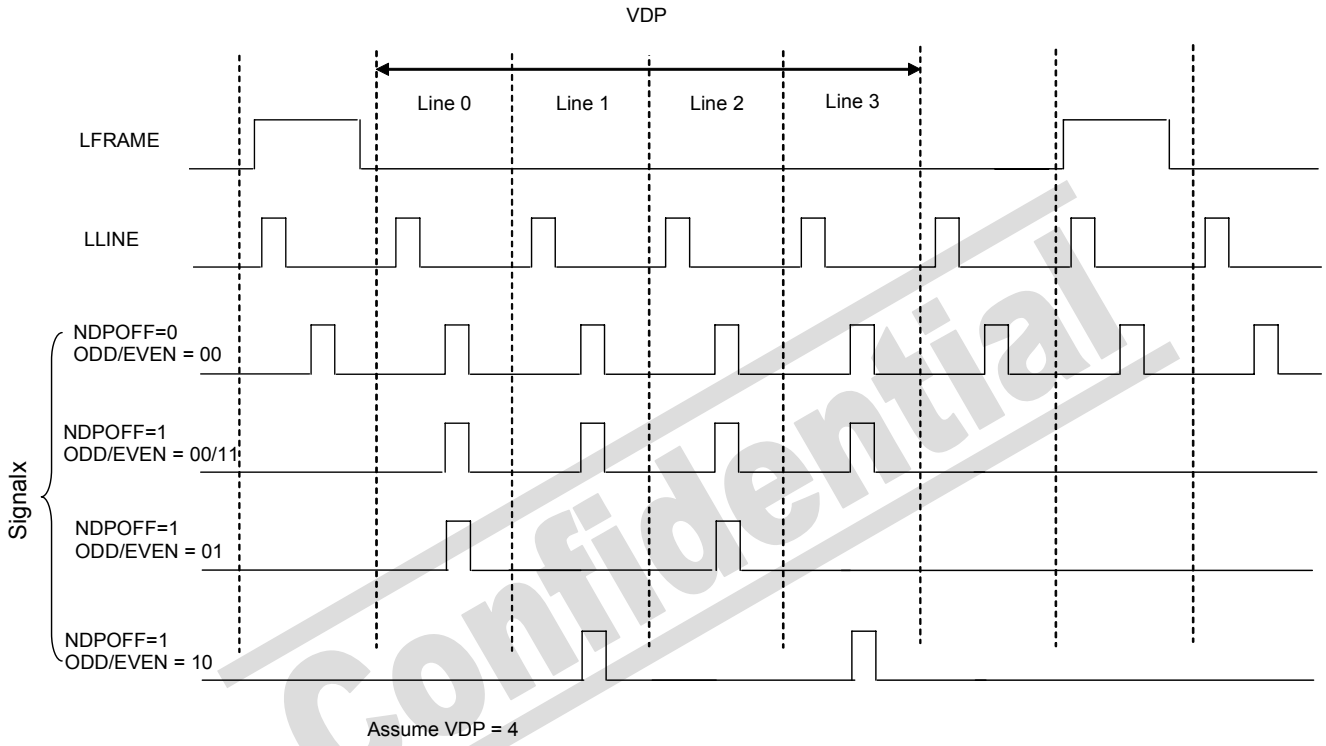
Bits 5-4 **LCD Singal7 Odd / Even [1:0]**
These bits enable signal output in odd or even lines
00/11 : enable signal in all lines
01 : enable signal in even lines
10 : enable signal in odd lines

Note
(1) These bits effective only when NDPOFF = 1 and LCD signal toggle by PCLK.

Bits 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **LCD Signal7 Toggle Register [1:0]**
00 : Disable
01 : toggle by PCLK
10 : toggle by Line
11 : toggle by Frame

Figure 2-20: Examples for LCD signalx by NDPOFF and ODD / Even bits



LCD Gen0 Configure Register **REG[330h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal Select for source 1 Bit 2	LCD Signal Select for source 1 Bit 1	LCD Signal Select for source 1 Bit 0	LCD Signal Select for source 2 Bit 2	LCD Signal Select for source 2 Bit 1	LCD Signal Select for source 2 Bit 0	LCD Signal Select for source 3 Bit 1	LCD Signal Select for source 3 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5 **LCD Signal Select for source 1 Register [2:0]**
This register select signal generator for source 1.

Bits 4-2 **LCD Signal Select for source 2 Register [2:0]**
This register select signal generator for source 2.

Bits 1-0 **LCD Signal Select for source 3 Register [1:0]**
This register select signal generator for source 3.

LCD Gen1 Configure Register **REG[331h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal Select for source 1 Bit 2	LCD Signal Select for source 1 Bit 1	LCD Signal Select for source 1 Bit 0	LCD Signal Select for source 2 Bit 2	LCD Signal Select for source 2 Bit 1	LCD Signal Select for source 2 Bit 0	LCD Signal Select for source 3 Bit 1	LCD Signal Select for source 3 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5 **LCD Signal Select for source 1 Register [2:0]**
This register select signal generator for source 1.

Bits 4-2 **LCD Signal Select for source 2 Register [2:0]**
This register select signal generator for source 2.

Bits 1-0 **LCD Signal Select for source 3 Register [1:0]**
This register select signal generator for source 3.

LCD Gen2 Configure Register **REG[332h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal Select for source 1 Bit 2	LCD Signal Select for source 1 Bit 1	LCD Signal Select for source 1 Bit 0	LCD Signal Select for source 2 Bit 2	LCD Signal Select for source 2 Bit 1	LCD Signal Select for source 2 Bit 0	LCD Signal Select for source 3 Bit 1	LCD Signal Select for source 3 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5 **LCD Signal Select for source 1 Register [2:0]**
This register select signal generator for source 1.

Bits 4-2 **LCD Signal Select for source 2 Register [2:0]**
This register select signal generator for source 2.

Bits 1-0 **LCD Signal Select for source 3 Register [1:0]**
This register select signal generator for source 3.

LCD Gen3 Configure Register **REG[333h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal Select for source 1 Bit 2	LCD Signal Select for source 1 Bit 1	LCD Signal Select for source 1 Bit 0	LCD Signal Select for source 2 Bit 2	LCD Signal Select for source 2 Bit 1	LCD Signal Select for source 2 Bit 0	LCD Signal Select for source 3 Bit 1	LCD Signal Select for source 3 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5 **LCD Signal Select for source 1 Register [2:0]**
This register select signal generator for source 1.

Bits 4-2 **LCD Signal Select for source 2 Register [2:0]**
This register select signal generator for source 2.

Bits 1-0 **LCD Signal Select for source 3 Register [1:0]**
This register select signal generator for source 3.

LCD Gen4 Configure Register **REG[334h]**

Bit	7	6	5	4	3	2	1	0
	LCD Signal Select for source 1 Bit 2	LCD Signal Select for source 1 Bit 1	LCD Signal Select for source 1 Bit 0	LCD Signal Select for source 2 Bit 2	LCD Signal Select for source 2 Bit 1	LCD Signal Select for source 2 Bit 0	LCD Signal Select for source 3 Bit 1	LCD Signal Select for source 3 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5 **LCD Signal Select for source 1 Register [2:0]**
This register select signal generator for source 1.

Bits 4-2 **LCD Signal Select for source 2 Register [2:0]**
This register select signal generator for source 2.

Bits 1-0 **LCD Signal Select for source 3 Register [1:0]**
This register select signal generator for source 3.

LCD Gen0 ROP Configure Register **REG[338h]**

Bit	7	6	5	4	3	2	1	0
	LCD Gen0 Configure Bit 7	LCD Gen0 Configure Bit 6	LCD Gen0 Configure Bit 5	LCD Gen0 Configure Bit 4	LCD Gen0 Configure Bit 3	LCD Gen0 Configure Bit 2	LCD Gen0 Configure Bit 1	LCD Gen0 Configure Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LCD Gen0 ROP Configure Register [1:0]**
This register configures signal generator operation for source 1, 2 and 3

LCD Gen1 ROP Configure Register **REG[339h]**

Bit	7	6	5	4	3	2	1	0
	LCD Gen1 Configure Bit 7	LCD Gen1 Configure Bit 6	LCD Gen1 Configure Bit 5	LCD Gen1 Configure Bit 4	LCD Gen1 Configure Bit 3	LCD Gen1 Configure Bit 2	LCD Gen1 Configure Bit 1	LCD Gen1 Configure Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LCD Gen1 ROP Configure Register [1:0]**
This register configures signal generator operation for source 1, 2 and 3

LCD Gen2 ROP Configure Register **REG[33Ah]**

Bit	7	6	5	4	3	2	1	0
	LCD Gen2 Configure Bit 7	LCD Gen2 Configure Bit 6	LCD Gen2 Configure Bit 5	LCD Gen2 Configure Bit 4	LCD Gen2 Configure Bit 3	LCD Gen2 Configure Bit 2	LCD Gen2 Configure Bit 1	LCD Gen2 Configure Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LCD Gen2 ROP Configure Register [1:0]**
This register configures signal generator operation for source 1, 2 and 3

LCD Gen3 ROP Configure Register **REG[33Bh]**

Bit	7	6	5	4	3	2	1	0
	LCD Gen3 Configure Bit 7	LCD Gen3 Configure Bit 6	LCD Gen3 Configure Bit 5	LCD Gen3 Configure Bit 4	LCD Gen3 Configure Bit 3	LCD Gen3 Configure Bit 2	LCD Gen3 Configure Bit 1	LCD Gen3 Configure Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LCD Gen3 ROP Configure Register [1:0]**
This register configures signal generator operation for source 1, 2 and 3

LCD Gen4 ROP Configure Register **REG[33Ch]**

Bit	7	6	5	4	3	2	1	0
	LCD Gen4	LCD Gen4	LCD Gen4	LCD Gen4	LCD Gen4	LCD Gen4	LCD Gen4	LCD Gen4

Bit	7	6	5	4	3	2	1	0
	Configure Bit 7	Configure Bit 6	Configure Bit 5	Configure Bit 4	Configure Bit 3	Configure Bit 2	Configure Bit 1	Configure Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **LCD Gen4 ROP Configure Register [1:0]**
This register configures signal generator operation for source 1, 2 and 3

Note
(1) Refer to WindowCE for the detailed Raster Operation (ROP).

2.1.12 Camera interface Registers

DV Mode Register **REG[160h]**

Bit	7	6	5	4	3	2	1	0
	JPEG start enable	0	0	Frame Generation Select	JPEG Still Picture Enable	DV display ready	Video Start enable	Video Still Picture Enable
Type	RW	RO	RO	RW	RW	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit7 **JPEG Start Enable**
This bit will enable the JPEG capture in camera interface

Bits 6-5 **Reserved bits**
These bits should be programmed by 0.

Bit 4 **Frame Generation Select**
This bit controls the frame start pulse.
When this bit = 1, DV_VValid signal is selected.
When this bit = 0, DV_Field signal is selected.

Bits 3 **JPEG Still Picture Enable**
This bit will enable the still picture mode which is disabled the update of the buffer.

Bits 2 **DV display ready flag**
These bits should be programmed by 0.

Bit 1 **Video Start Enable**
This bit will enable the preview in camera interface

Bit 0 **Video Still Picture Enable**
This bit will enable the still picture mode which is disabled the update of the JPEG and preview video buffer.

Frame Sampling Register **REG[161h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	Frame Sampling Control bit 2	Frame Sampling Control bit 1	Frame Sampling Control bit 0
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-3 **Reserved bits**
These bits should be programmed by 0.

Bit 2-0 **Frame Sampling Control bits [2:0]**
These bits control the camera data sampling rate in frames.

Table 2-20: Frame Sampling mode select

REG[161h] bits 2-0	Frame Sampling Mode
000	Every frame is sampled
001	1 Frame is sampled for every 2 Frames
010	1 Frame is sampled for every 3 Frames
...	...
111	1 Frame is sampled for every 6 Frames

New Frame Position Register 0 **REG[162h]**

Bit	7	6	5	4	3	2	1	0
	New Frame Position bit 7	New Frame Position bit 6	New Frame Position bit 5	New Frame Position bit 4	New Frame Position bit 3	New Frame Position bit 2	New Frame Position bit 1	New Frame Position bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

New Frame Position Register 1 **REG[163h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	New Frame Position bit 9	New Frame Position bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[163h] Bits 1-0, REG[162h] Bits 7-0 **New Frame Position [9:0]**
These bits determine the new frame position after DV_VValid signal

JPEG Line Buffer Horizontal Register 0 **REG[164h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Line Buffer Horizontal Size bit 7	JPEG Line Buffer Horizontal Size bit 6	JPEG Line Buffer Horizontal Size bit 5	JPEG Line Buffer Horizontal Size bit 4	JPEG Line Buffer Horizontal Size bit 3	JPEG Line Buffer Horizontal Size bit 2	JPEG Line Buffer Horizontal Size bit 1	JPEG Line Buffer Horizontal Size bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG Line Buffer Horizontal Register 1 **REG[165h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	JPEG Line Buffer Horizontal Size bit 8
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

REG[165h] Bit 0, **JPEG Line Buffer Horizontal Size [8:0]**

REG[164h] Bits 7-0 These bits determine the JPEG Line Buffer Horizontal capture Size.

REG[165-164h] bits 8-0 = (Video Horizontal Size / 2 ^ (2+ (REG[172h] bit 2-0)) - 1

JPEG Vertical Size Register 0 **REG[168h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Vertical Size bit 7	JPEG Vertical Size bit 6	JPEG Vertical Size bit 5	JPEG Vertical Size bit 4	JPEG Vertical Size bit 3	JPEG Vertical Size bit 2	JPEG Vertical Size bit 1	JPEG Vertical Size bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG Vertical Size Register 1 **REG[169h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	00	JPEG Vertical Size bit 10	JPEG Vertical Size bit 9	JPEG Vertical Size bit 8
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[169h] Bits 2-0, REG[168h] Bits 7-0 **JPEG Vertical Size [10:0]**
These bits determine the JPEG Vertical capture Size.

REG[168-169h] bits 9-0 = (Video Vertical Size / 2 ^ (REG[173h] bit 2-0)) - 1

DV Saturation Control Register **REG[16Ch]**

Bit	7	6	5	4	3	2	1	0
	DV Saturation Bit 7	DV Saturation Bit 6	DV Saturation Bit 5	DV Saturation Bit 4	DV Saturation Bit 3	DV Saturation Bit 2	DV Saturation Bit 1	DV Saturation Bit 0
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	1	0	0	0	0	0	0

Bits 7-0 **DV Saturation Control [7:0]**
These bits control the saturation of the camera interface.

Table 2-21: The example setting for DV Saturation

Control Bits [7:0]	Saturation Control
0x00	Gain = 0
0x01	Gain = 1/64
…	
0x40	(Default) Gain = 1
…	
0x7F	Gain = 127/64

DV Brightness Control Register **REG[16Dh]**

Bit	7	6	5	4	3	2	1	0
	DV Brightness Bit 7	DV Brightness Bit 6	DV Brightness Bit 5	DV Brightness Bit 4	DV Brightness Bit 3	DV Brightness Bit 2	DV Brightness Bit 1	DV Brightness Bit 0
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	1	0	0	0	0	0	0	0

Bits 7-0 **DV Brightness Control [7:0]**
These bits control the brightness of the camera interface.

Table 2-22: The example setting for DV Brightness

Control Bits [7:0]	Brightness Control
0x00	Value = 0
…	
0x80	(Default) Value = 128
…	
0xFF	Value = 255

DV Contrast Control Register **REG[16Eh]**

Bit	7	6	5	4	3	2	1	0
	DV Contrast Bit 7	DV Contrast Bit 6	DV Contrast Bit 5	DV Contrast Bit 4	DV Contrast Bit 3	DV Contrast Bit 2	DV Contrast Bit 1	DV Contrast Bit 0
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	1	0	0	0	0	0	0

Bits 7-0 **DV Contrast Control [7:0]**
These bits control the contrast of the camera interface.

Table 2-23: The example setting for DV Contrast

Control Bits 7 to 0	Contrast Control
0x00	Gain = 0
0x01	Gain = 1/64
…	
0x40	(Default) Gain = 1
…	
0x7F	Gain = 127/64

DV Control Register **REG[16Fh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	DV YUV Data Range
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **DV YUV Data Range**
This bit specifies the YUV data range of the camera interface input.

Table 2-24: DV YUV Data Range Selection

REG[16Fh] bit 0	YUV Data Range
0	0 =< Y =< 255 0 =< U =< 255 0 =< V =< 255
1	0 =< Y =< 255 -128 =< U =< 127 -128 =< V =< 127

Preview Horizontal Decimation Ratio Register **REG[170h]**

Bit	7	6	5	4	3	2	1	0

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	Preview Horizontal Decimation Ratio bit 2	Preview Horizontal Decimation Ratio bit 1	Preview Horizontal Decimation Ratio bit 0
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-3 **Reserved bits**
These bits should be programmed by 0.

Bits 2-0 **Preview Horizontal Decimation Ratio [2:0]**
Decimation Ratio = 1 : 2 ^ (REG[170h] bit 2-0)
The maximum value = b100

Note
(1) The horizontal size is align with 64-bit.

Preview Vertical Decimation Ratio Register **REG[171h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	Preview Vertical Decimation Ratio bit 2	Preview Vertical Decimation Ratio bit 1	Preview Vertical Decimation Ratio bit 0
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-3 **Reserved bits**
These bits should be programmed by 0.

Bits 2-0 **Preview Vertical Decimation Ratio [2:0]**
Decimation Ratio = 1 : 2 ^ (REG[171h] bit 2-0)
The maximum value = b100

Note
(1) The vertical size is align with 64-bit.

JPEG Horizontal Decimation Ratio Register **REG[172h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	JPEG Horizontal Decimation Ratio bit 2	JPEG Horizontal Decimation Ratio bit 1	JPEG Horizontal Decimation Ratio bit 0
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-3 **Reserved bits**
These bits should be programmed by 0.

Bits 2-0 **JPEG Horizontal Decimation Ratio [2:0]**
Decimation Ratio = 1 : 2 ^ (REG[172h] bit 2-0)

JPEG Vertical Decimation Ratio Register **REG[173h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	JPEG Vertical Decimation	JPEG Vertical Decimation	JPEG Vertical Decimation

Bit	7	6	5	4	3	2	1	0
						Ratio bit 2	Ratio bit 1	Rati bit 0
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-3 **Reserved bits**
These bits should be programmed by 0.

Bits 2-0 **JPEG Vertical Decimation Ratio [2:0]**
Decimation Ratio = 1 : 2 ^ (REG[173h] bit 2-0)

DV Horizontal Period Register 0 **REG[174h]**

Bit	7	6	5	4	3	2	1	0
	DV Horizontal Period bit 7	DV Horizontal Period bit 6	DV Horizontal Period bit 5	DV Horizontal Period bit 4	DV Horizontal Period bit 3	DV Horizontal Period bit 2	DV Horizontal Period bit 1	DV Horizontal Period bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

DV Horizontal Period Register 1 **REG[175h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	DV Horizontal Period bit 11	DV Horizontal Period bit 10	DV Horizontal Period bit 9	DV Horizontal Period bit 8
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[175h] bits 3-0, REG[174h] bits 7-0 **DV Horizontal Period [11:0]**

Preview Horizontal Crop Start Register 0 **REG[184h]**

Bit	7	6	5	4	3	2	1	0
	Preview Horizontal Crop Start bit 7	Preview Horizontal Crop Start bit 6	Preview Horizontal Crop Start bit 5	Preview Horizontal Crop Start bit 4	Preview Horizontal Crop Start bit 3	Preview Horizontal Crop Start bit 2	Preview Horizontal Crop Start bit 1	Preview Horizontal Crop Start bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Preview Horizontal Crop Start Register 1 **REG[185h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	Preview Horizontal Crop Start bit 11	Preview Horizontal Crop Start bit 10	Preview Horizontal Crop Start bit 9	Preview Horizontal Crop Start bit 8
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[185h] Bits 3-0, REG[184h] Bits 7-0 **Preview Horizontal Crop Start [11:0]**
These bits control the horizontal crop start position in the input video signal.

Preview Vertical Crop Start Register 0 **REG[188h]**

Bit	7	6	5	4	3	2	1	0
	Preview Vertical Crop Start bit 7	Preview Vertical Crop Start bit 6	Preview Vertical Crop Start bit 5	Preview Vertical Crop Start bit 4	Preview Vertical Crop Start bit 3	Preview Vertical Crop Start bit 2	Preview Vertical Crop Start bit 1	Preview Vertical Crop Start bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Preview Vertical Crop Start Register 1 **REG[189h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Preview Vertical Crop Start bit 9	Preview Vertical Crop Start bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[189h] Bits 1-0, REG[188h] Bits 7-0 **Preview Vertical Crop Start [9:0]**
These bits control the Vertical crop start position in the input video signal.

REG[189-188h] bits 11-0 = Preview Vertical Crop Start - 1

Preview Horizontal Crop Size Register 0 **REG[18Ch]**

Bit	7	6	5	4	3	2	1	0
	Preview Horizontal Crop Size bit 7	Preview Horizontal Crop Size bit 6	Preview Horizontal Crop Size bit 5	Preview Horizontal Crop Size bit 4	Preview Horizontal Crop Size bit 3	Preview Horizontal Crop Size bit 2	Preview Horizontal Crop Size bit 1	Preview Horizontal Crop Size bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Preview Horizontal Crop Size Register 1 **REG[18Dh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	Preview Horizontal Crop Size bit 11	Preview Horizontal Crop Size bit 10	Preview Horizontal Crop Size bit 9	Preview Horizontal Crop Size bit 8
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[18Dh] Bits 3-0, REG[18Ch] Bits 7-0 **Preview Horizontal Crop Size [11:0]**
These bits control the horizontal crop size in the input video signal.

Preview Vertical Crop Size Register 0 **REG[190h]**

Bit	7	6	5	4	3	2	1	0
	Preview Vertical Crop Size bit 7	Preview Vertical Crop Size bit 6	Preview Vertical Crop Size bit 5	Preview Vertical Crop Size bit 4	Preview Vertical Crop Size bit 3	Preview Vertical Crop Size bit 2	Preview Vertical Crop Size bit 1	Preview Vertical Crop Size bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
state								

Preview Vertical Crop Size Register 1 **REG[191h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Preview Vertical Crop Size bit 9	Preview Vertical Crop Size bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[191h] Bits 1-0, REG[190h] Bits 7-0 **Preview Vertical Crop Size [9:0]**
These bits control the Vertical crop size in the input video signal.

DV Misc Register **REG[194h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	Frame Pulse Width bit 4	Frame Pulse Width bit 3	Frame Pulse Width bit 2	Frame Pulse Width bit 1	Frame Pulse Width bit 0
Type	RO	RO	RO	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 4-0 **Frame Pulse Width Register**
These bits control Frame Pulse Width.

Preview Video Memory Start 1 Address Register 0 **REG[19Ch]**

Bit	7	6	5	4	3	2	1	0
	Preview Video Memory Start 1 Address bit 7	Preview Video Memory Start 1 Address bit 6	Preview Video Memory Start 1 Address bit 5	Preview Video Memory Start 1 Address bit 4	Preview Video Memory Start 1 Address bit 3	Preview Video Memory Start 1 Address bit 2	Preview Video Memory Start 1 Address bit 1	0
Type	RW	RW	RW	RW	RW	RW	RW	RO
Reset state	0	0	0	0	0	0	0	0

Preview Video Memory Start 1 Address Register 1 **REG[19Dh]**

Bit	7	6	5	4	3	2	1	0
	Preview Video Memory Start 1 Address bit 15	Preview Video Memory Start 1 Address 14	Preview Video Memory Start 1 Address bit 13	Preview Video Memory Start 1 Address bit 12	Preview Video Memory Start 1 Address bit 11	Preview Video Memory Start 1 Address bit 10	Preview Video Memory Start 1 Address bit 9	Preview Video Memory Start 1 Address bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Preview Video Memory Start 1 Address Register 2 **REG[19Eh]**

Bit	7	6	5	4	3	2	1	0

Bit	7	6	5	4	3	2	1	0
	Preview Video Memory Start 1 Address bit 23	Preview Video Memory Start 1 Address 22	Preview Video Memory Start 1 Address bit 21	Preview Video Memory Start 1 Address bit 20	Preview Video Memory Start 1 Address bit 19	Preview Video Memory Start 1 Address bit 18	Preview Video Memory Start 1 Address bit 17	Preview Video Memory Start 1 Address bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Preview Video Memory Start 1 Address Register 3 **REG[19Fh]**

Bit	7	6	5	4	3	2	1	0
	Preview Video Memory Start 1 Address bit 31	Preview Video Memory Start 1 Address 30	Preview Video Memory Start 1 Address bit 29	Preview Video Memory Start 1 Address bit 28	Preview Video Memory Start 1 Address bit 27	Preview Video Memory Start 1 Address bit 26	Preview Video Memory Start 1 Address bit 25	Preview Video Memory Start 1 Address bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[19Fh] Bits 7-0,
REG[19Eh] Bits 7-0,
REG[19Dh] Bits 7-0,
REG[19Ch] Bits 7-1

Preview Video Memory Start 1 Address [31:0]
These bits determine Preview Video Start Address 1 in Main Memory
Note
(1) This is a double-word (32-bit) address.

REG[19Ch] Bit 0

Reserved bit
This is should be programmed by 0

Preview Video Memory Start 2 Address Register 0 **REG[1A0h]**

Bit	7	6	5	4	3	2	1	0
	Preview Video Memory Start 2 Address bit 7	Preview Video Memory Start 2 Address bit 6	Preview Video Memory Start 2 Address bit 5	Preview Video Memory Start 2 Address bit 4	Preview Video Memory Start 2 Address bit 3	Preview Video Memory Start 2 Address bit 2	Preview Video Memory Start 2 Address bit 1	0
Type	RW	RW	RW	RW	RW	RW	RW	RO
Reset state	0	0	0	0	0	0	0	0

Preview Video Memory Start 2 Address Register 1 **REG[1A1h]**

Bit	7	6	5	4	3	2	1	0
	Preview Video Memory Start 2 Address bit 15	Preview Video Memory Start 2 Address 14	Preview Video Memory Start 2 Address bit 13	Preview Video Memory Start 2 Address bit 12	Preview Video Memory Start 2 Address bit 11	Preview Video Memory Start 2 Address bit 10	Preview Video Memory Start 2 Address bit 9	Preview Video Memory Start 2 Address bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Preview Video Memory Start 2 Address Register 2 **REG[1A2h]**

Bit	7	6	5	4	3	2	1	0
	Preview	Preview	Preview	Preview	Preview	Preview	Preview	Preview

Bit	7	6	5	4	3	2	1	0
	Video Memory Start 2 Address bit 23	Video Memory Start 2 Address 22	Video Memory Start 2 Address bit 21	Video Memory Start 2 Address bit 20	Video Memory Start 2 Address bit 19	Video Memory Start 2 Address bit 18	Video Memory Start 2 Address bit 17	Video Memory Start 2 Address bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Preview Video Memory Start 2 Address Register 3 **REG[1A3h]**

Bit	7	6	5	4	3	2	1	0
	Preview Video Memory Start 2 Address bit 31	Preview Video Memory Start 2 Address 30	Preview Video Memory Start 2 Address bit 29	Preview Video Memory Start 2 Address bit 28	Preview Video Memory Start 2 Address bit 27	Preview Video Memory Start 2 Address bit 26	Preview Video Memory Start 2 Address bit 25	Preview Video Memory Start 2 Address bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1A3h] Bits 7-0,
REG[1A2h] Bits 7-0,
REG[1A1h] Bits 7-0,
REG[1A0h] Bits 7-1

Preview Video Memory Start 2 Address [31:0]
These bits determine Preview Video Start Address 2 in Main Memory

In video preview mode, preview video data are fit starting from Preview Video Memory Address 1 or Address 2 alternatively at rate set by Frame Sampling Register, REG[161h].

When single preview window is used, both value of Address 1 and Address 2 should be the same.

Note
(1) This is a double-word (32-bit) address.

REG[1A0h] Bit 0

Reserved bit
This bit should be programmed by 0

RGB/YUV Setting Register **REG[1A4h]**

Bit	7	6	5	4	3	2	1	0
	Floating Window RGB	Main Window RGB	0	0	0	0	0	0
Type	RW	RW	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 7

Floating Window RGB
1 : RGB
0 : YUV

Bit 6

Main Window RGB
1 : RGB
0 : YUV

Bits 5-0

Reserved bits
These bits should be programmed by 0.

Video Control Register **REG[1A6h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	Video Delta Scale Format	Video Delta Scale Enable	Video Vertical Scale 6 to 5

Bit	7	6	5	4	3	2	1	0
						Mode		
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 2 **Video Delta Scale Format Mode**
0: First row inside. Second row outside
1: First row outside. Second row inside
Refer to Figure 2-21: Implementation for REG[1A6h] Bit 2

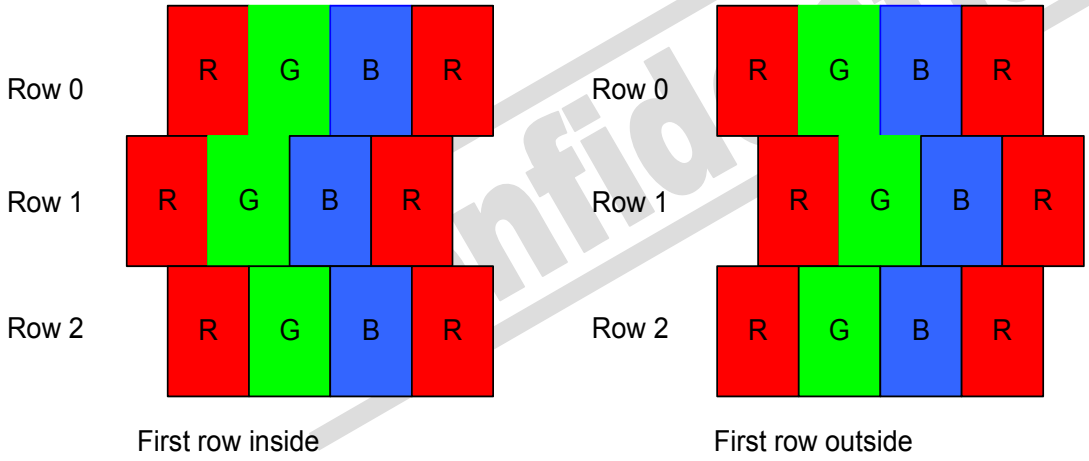
Bit 1 **Video Delta Scale Enable**
0: Normal
1: Enable

Note
(1) This bit is effective for RGB mode only

Bit 0 **Video Vertical Scale 6 to 5**
0: Normal (for NTSC)
1: Enable (for PAL)

Note
(1) Bits 2 and 1 for Delta-type Serial-TFT only

Figure 2-21: Implementation for REG[1A6h] Bit 2



YUV Offset Setting Registers **REG[1A8h]**

Bit	7	6	5	4	3	2	1	0
	CSC Mode	0	0	0	0	0	0	0
Type	RW	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Y Offset Registers **REG[1A9h]**

Bit	7	6	5	4	3	2	1	0
	Y Offset Registers Bit 7	Y Offset Registers Bit 6	Y Offset Registers Bit 5	Y Offset Registers Bit 4	Y Offset Registers Bit 3	Y Offset Registers Bit 2	Y Offset Registers Bit 1	Y Offset Registers Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

CB Offset Registers **REG[1AAh]**

Bit	7	6	5	4	3	2	1	0
	CB Offset Registers Bit 7	CB Offset Registers Bit 6	CB Offset Registers Bit 5	CB Offset Registers Bit 4	CB Offset Registers Bit 3	CB Offset Registers Bit 2	CB Offset Registers Bit 1	CB Offset Registers Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	1	0	0	0	0	0	0	0

CB Offset Registers **REG[1ABh]**

Bit	7	6	5	4	3	2	1	0
	CB Offset Registers Bit 7	CB Offset Registers Bit 6	CB Offset Registers Bit 5	CB Offset Registers Bit 4	CB Offset Registers Bit 3	CB Offset Registers Bit 2	CB Offset Registers Bit 1	CB Offset Registers Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	1	0	0	0	0	0	0	0

REG [1A8h] Bit 7 **Color Space Conversion (CSC) mode select**
This bit is used to select the conversion mode for RGB and YUV
1 : Equation 1
0 : Equation 2

REG [1A9h] Bit 7-0 **Y Offset Registers bits [7:0]**
These bits control Y Offset

REG [1AAh] Bit 7-0 **CB Offset Registers bits [7:0]**
These bits control CB Offset

REG [1ABh] Bit 7-0 **CR Offset Registers bits [7:0]**
These bits control CR Offset

2.1.13 Digital Video Interface Configuration Registers

TV0 Register **REG[1ACh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	DV_HValid select Bit	DV_Field Invert Bit	Reserved	1	Reserved
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5, 2, 0 **Reserved bits**
These bits should programmed as 0

Bit 4: **DV_HValid select**
If there is no DV_HValid signals in vertical blanking period, set this bit = 1
If there are DV_HValid signals in vertical blanking period, set this bit = 0

Bit 3: **DV_Field Invert**
When this bit = 1, invert the DV_Field signal
When this bit = 0, normal

Bit 1 **Reserved bit**
This bit should programmed as 1

TV1 Register **REG[1ADh]**

Bit	7	6	5	4	3	2	1	0

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Active Video Select Bit 0	Field Select Bit	Reserved	1
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	1	0	0	0	0	0	0	0

Bits 7-4, 1 **Reserved bits**
These bits should programmed as 0

Bit 3: Active Video Select
When this bit = 1, select BT656 active video signal
When this bit = 0, select external DV_VVALID and DV_HVALID input signal

Bit 2: Field Select
When this bit = 1, selects BT656 field signal
When this bit = 0, selects external DV_Field input signal.

Bit 1 **Reserved bit**
This bit should programmed as 1

TV3 Register **REG[1AEh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Vsync Invert Bit	Hsync Invert Bit	Sync Select Bit
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	1	0	0	0	0	0	0	0

Bits 7-3 **Reserved bits**
These bits should programmed as 0

Bit 2 Vsync Invert
When this bit = 1, invert the DV_VValid
When this bit = 0, normal

Bit 1 Hsync Invert
When this bit = 1, invert the DV_HValid
When this bit = 0, normal

Bit 0 Sync Select
When this bit = 1, sync select BT656 signal
When this bit = 0, normal

For BT656 format :
REG[1ACh] = 0x02
REG[1ADh] = 0x0D
REG[1AEh] = 0x01

External Camera Module Control Register **REG[1AFh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Enable DV Control	Reset DV Control
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 1 **Enable DV Control**
When this bit = 1, DV_ENB = high
When this bit = 1, DV_ENB = low

Bit 0 **Reset DV Control**
When this bit = 1, DV_RESET = high
When this bit = 0, DV_RESET = low

Digital Video Window Display Start Address 0 Register 0 **REG[1B0h]**

Bit	7	6	5	4	3	2	1	0
	Digital Video Window Display Start Address 0 Bit 7	Digital Video Window Display Start Address 0 Bit 6	Digital Video Window Display Start Address 0 Bit 5	Digital Video Window Display Start Address 0 Bit 4	Digital Video Window Display Start Address 0 Bit 3	Digital Video Window Display Start Address 0 Bit 2	Digital Video Window Display Start Address 0 Bit 1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Digital Video Window Display Start Address 0 Register 1 **REG[1B1h]**

Bit	7	6	5	4	3	2	1	0
	Digital Video Window Display Start Address 0 Bit 15	Digital Video Window Display Start Address 0 Bit 14	Digital Video Window Display Start Address 0 Bit 13	Digital Video Window Display Start Address 0 Bit 12	Digital Video Window Display Start Address 0 Bit 11	Digital Video Window Display Start Address 0 Bit 10	Digital Video Window Display Start Address 0 Bit 9	Digital Video Window Display Start Address 0 Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Digital Video Window Display Start Address 0 Register 2 **REG[1B2h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Digital Video Window Display Start Address 0 Bit 16
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

REG[1B2h] bit 0, REG[1B1h] bits 7-0, REG[1B0h] bits 7-1

Digital Video Window Display Start Address 0 Bits [16:0]
These bits form the 17-bit address for the starting double-word of the Digital Video window.

Note
(1) This is a double-word (32-bit) address. An entry of 00000h into these registers represents the first double-word of display memory, an entry of 00001h represents the second double-word of the display memory, and so on.

Note
(1) These bits will not effective until the Digital Video Window Enable bit is set to 1 (REG[160h] bit 1 = 1).

REG[1B0h] bit 0 **Reserved bit**
This bit should be programmed by 0.

Digital Video Window Display Start Address 1 Register 0 **REG[1B4h]**

Bit	7	6	5	4	3	2	1	0
	Digital Video Window Display Start Address 1 Bit 7	Digital Video Window Display Start Address 1 Bit 6	Digital Video Window Display Start Address 1 Bit 5	Digital Video Window Display Start Address 1 Bit 4	Digital Video Window Display Start Address 1 Bit 3	Digital Video Window Display Start Address 1 Bit 2	Digital Video Window Display Start Address 1 Bit 1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Digital Video Window Display Start Address 1 Register 1 **REG[1B5h]**

Bit	7	6	5	4	3	2	1	0
	Digital Video Window Display Start Address 1 Bit 15	Digital Video Window Display Start Address 1 Bit 14	Digital Video Window Display Start Address 1 Bit 13	Digital Video Window Display Start Address 1 Bit 12	Digital Video Window Display Start Address 1 Bit 11	Digital Video Window Display Start Address 1 Bit 10	Digital Video Window Display Start Address 1 Bit 9	Digital Video Window Display Start Address 1 Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Digital Video Window Display Start Address 1 Register 2 **REG[1B6h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Digital Video Window Display Start Address 1 Bit 16
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

REG[1B6h] bit 0,
REG[1B5h] bits 7-0,
REG[1B4h] bits 7-1

Digital Video Window Display Start Address 1 Bits [16:0]
These bits form the 17-bit address for the starting double-word of the Digital Video window.

Note
(1) This is a double-word (32-bit) address. An entry of 00000h into these registers represents the first double-word of display memory, an entry of 00001h represents the second double-word of the display memory, and so on.

Note
(1) These bits will not effective until the Digital Video Window Enable bit is set to 1 (REG[160h] bit 1 = 1).

REG[1B4h] bit 0 **Reserved bit**
This bit should be programmed by 0.

View Resizer Control Register **REG[360h]**

Bit	7	6	5	4	3	2	1	0

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	View Resizer Enable
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **View Resizer Enable**
This bit controls the view resizer logic.
When this bit = 1, the view resizer logic is enabled.
When this bit = 0, the view resizer logic is disabled.

View Resizer Start X Position Register 0 **REG[364h]**

Bit	7	6	5	4	3	2	1	0
	View Resizer Start X Position Bit 7	View Resizer Start X Position Bit 6	View Resizer Start X Position Bit 5	View Resizer Start X Position Bit 4	View Resizer Start X Position Bit 3	View Resizer Start X Position Bit 2	View Resizer Start X Position Bit 1	View Resizer Start X Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

View Resizer Start X Position Register 1 **REG[365h]**

Bit	7	6	5	4	3	2	1	0
	View Resizer Start X Position Bit 15	View Resizer Start X Position Bit 14	View Resizer Start X Position Bit 13	View Resizer Start X Position Bit 12	View Resizer Start X Position Bit 11	View Resizer Start X Position Bit 10	View Resizer Start X Position Bit 9	View Resizer Start X Position Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[365h] Bits 7-0 **View Resizer Start X Position bits [15:0]**
REG[364h] Bits 7-0 These bits determine the X start position for the view resizer. These bits cannot be changed during the JPEG decode progress and cannot be set larger than the image size.

View Resizer Start Y Position Register 0 **REG[366h]**

Bit	7	6	5	4	3	2	1	0
	View Resizer Start Y Position Bit 7	View Resizer Start Y Position Bit 6	View Resizer Start Y Position Bit 5	View Resizer Start Y Position Bit 4	View Resizer Start Y Position Bit 3	View Resizer Start Y Position Bit 2	View Resizer Start Y Position Bit 1	View Resizer Start Y Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

View Resizer Start Y Position Register 1 **REG[367h]**

Bit	7	6	5	4	3	2	1	0
	View	View	View	View	View	View	View	View

Bit	7	6	5	4	3	2	1	0
	Resizer Start Y Position Bit 15	Resizer Start Y Position Bit 14	Resizer Start Y Position Bit 13	Resizer Start Y Position Bit 12	Resizer Start Y Position Bit 11	Resizer Start Y Position Bit 10	Resizer Start Y Position Bit 9	Resizer Start Y Position Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[367h] Bits 7-0 **View Resizer Start Y Position bits [15:0]**
REG[366h] Bits 7-0 These bits determine the Y start position for the view resizer. These bits cannot be changed during the JPEG decode progress and cannot be set larger than the image size.

View Resizer End X Position Register 0 **REG[368h]**

Bit	7	6	5	4	3	2	1	0
	View Resizer End X Position Bit 7	View Resizer End X Position Bit 6	View Resizer End X Position Bit 5	View Resizer End X Position Bit 4	View Resizer End X Position Bit 3	View Resizer End X Position Bit 2	View Resizer End X Position Bit 1	View Resizer End X Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	1	1	1	1	1	1	1

View Resizer End X Position Register 1 **REG[369h]**

Bit	7	6	5	4	3	2	1	0
	View Resizer End X Position Bit 15	View Resizer End X Position Bit 14	View Resizer End X Position Bit 13	View Resizer End X Position Bit 12	View Resizer End X Position Bit 11	View Resizer End X Position Bit 10	View Resizer End X Position Bit 9	View Resizer End X Position Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	1	0

REG[369h] Bits 7-0 **View Resizer End X Position bits [15:0]**
REG[368h] Bits 7-0 These bits determine the X End position for the view resizer. These bits cannot be changed during the JPEG decode progress and cannot be set larger than the image size.

View Resizer End Y Position Register 0 **REG[36Ah]**

Bit	7	6	5	4	3	2	1	0
	View Resizer End Y Position Bit 7	View Resizer End Y Position Bit 6	View Resizer End Y Position Bit 5	View Resizer End Y Position Bit 4	View Resizer End Y Position Bit 3	View Resizer End Y Position Bit 2	View Resizer End Y Position Bit 1	View Resizer End Y Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	1	1	0	1	1	1	1	1

View Resizer End Y Position Register 1 **REG[36Bh]**

Bit	7	6	5	4	3	2	1	0
	View Resizer End Y Position	View Resizer End Y Position	View Resizer End Y Position	View Resizer End Y Position	View Resizer End Y Position	View Resizer End Y Position	View Resizer End Y Position	View Resizer End Y Position

Bit	7	6	5	4	3	2	1	0
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	1

REG[36Bh] bits 7-0 **View Resizer End Y Position bits [15:0]**
REG[36Ah] bits 7-0 These bits determine the Y end position for the view resizer. These bits cannot be changed during the JPEG decode progress and cannot be set larger than the image size.

View Resizer Operation Setting Register 0 **REG[36Ch]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	View Resizer Scaling Rate Bit 3	View Resizer Scaling Rate Bit 2	View Resizer Scaling Rate Bit 1	View Resizer Scaling Rate Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-4 **Reserved bits**
These bits should be programmed by 0.
Bits 3-0 **View Resizer Scaling Rate bits [3:0]**
These bits determine the view resizer scaling rate.

Table 2-25: View Resizer Scaling Rate Selection

REG[036ch] bits 3-0	View Resizer Scaling Rate
0000	Reserved
0001	Reserved
0010	1/2
0011	Reserved
0100	1/4
0101	Reserved
0110	Reserved
0111	Reserved
1000	1/8
1001-1111	Reserved

View Resizer Operation Setting Register 1 **REG[36Eh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	View Resizer Scaling Mode Bit 1	View Resizer Scaling Mode Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-2 **Reserved bits**
These bits should be programmed by 0.
Bits 1-0 **View Resizer Scaling Mode bits [1:0]**
These bits determine the view resizer scaling mode.

Table 2-26: View Resizer Scaling Mode Selection

REG[036Eh] bits 1-0	View Resizer Scaling Mode
00	no resizer scaling
01	V/H Reduction
10	V: Reduction, H: Average
11	Reserved

2.1.14 2D Engine Registers

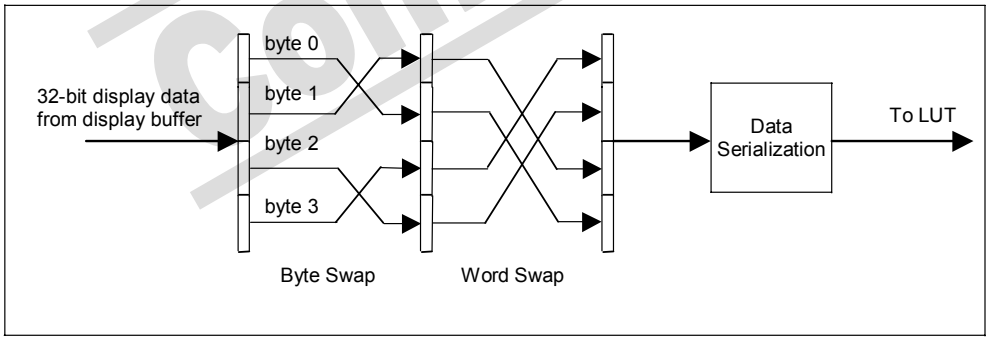
Special Effects Register **REG[71h]**

Bit	7	6	5	4	3	2	1	0
	Display Data Word Swap	Display Data Byte Swap	0	Floating Window Enable	0	Display Mirror Mode	Display Rotate Mode Select Bit 1	Display Rotate Mode Select Bit 0
Type	RW	RW	RO	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **Display Data Word Swap**
The display pipe fetches 32-bit of data from the display buffer. This bit enables the lower 16-bit word and the upper 16-bit word to be swapped before sending them to the LCD display. If the Display Data Byte Swap bit is also enabled, then the byte order of the fetched 32-bit data is reversed.

Bit 6 **Display Data Byte Swap**
The display pipe fetches 32-bit of data from the display buffer. This bit enables swapping of byte 0 and byte 1, byte 2 and byte 3, before sending them to the LCD. If the Display Data Word Swap bit is also set, then the byte order of the fetched 32-bit data is reversed.

Figure 2-22: Display Data Byte/Word Swap



Bit 4 **Floating Window Enable**
This bit enables the floating window within the main window used for the Floating Window feature. The location of the floating window within the main window is determined by the Floating Window Position X registers (REG[84h], REG[85h], REG[8Ch], REG[8Dh]) and Floating Window Position Y registers (REG[88h], REG[89h], REG[90h], REG[91h]). The floating window has its own Display Start Address register (REG[7Ch, REG[7Dh], REG[7Eh]) and Memory Address Offset register (REG[80h], REG[81h]). The floating window shares the same color depth and display orientation as the main window.

When this bit = 1, Floating Window is enabled.
When this bit = 0, Floating Window is disabled.

Bit 2 **Display Mirror Mode**
When this bit = 1,
When this bit = 1, Mirror Mode is enabled.
When this bit = 0, Mirror Mode is disable.

Bits 1-0 **Display Rotate Mode Select Bits [1:0]**
These bits select different display orientations:

Table 2-27: Display Rotate Mode Select Options

Display Rotate Mode Select Bits [1:0]	Display Orientation
00	0° (Normal)
01	90°
10	180°
11	270°

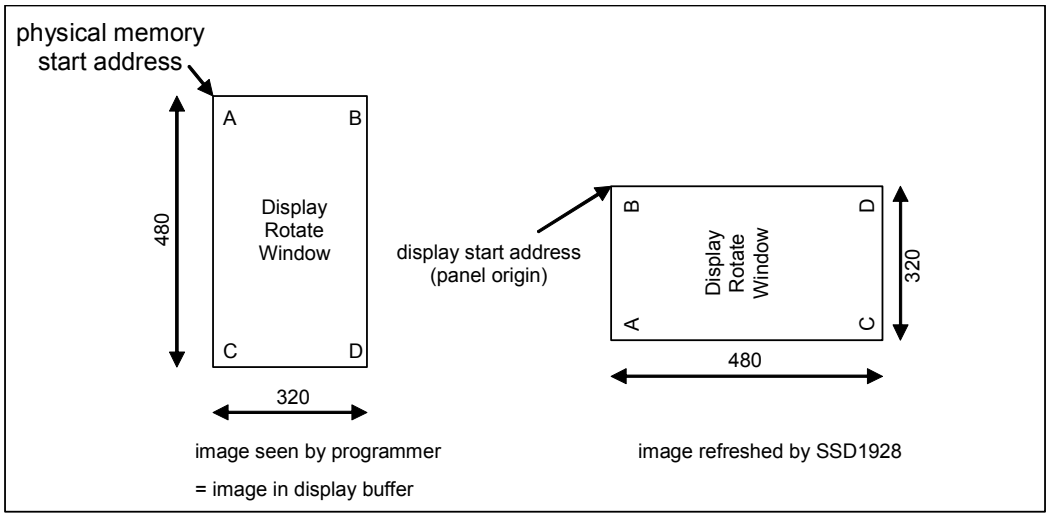
2.1.15 Display Rotate Mode

The image is not actually rotated in the display buffer since there is no address translation during CPU read/write. The image is rotated during display refresh.

2.1.15.1 90° Display Rotate Mode

The following figure shows how the programmer sees a 320x480 rotated image and how the image is being displayed. The application image is written to the SSD1928 in the following sense: A–B–C–D. The display is refreshed by the SSD1928 in the following sense: B-D-A-C.

Figure 2-23: Relationship Between The Screen Image and the Image Refreshed in 90° Display Rotate Mode.



Enable 90° Display Rotate Mode
Set Display Rotate Mode Select bits to 01 (REG[71h] bits 1:0 = 01).

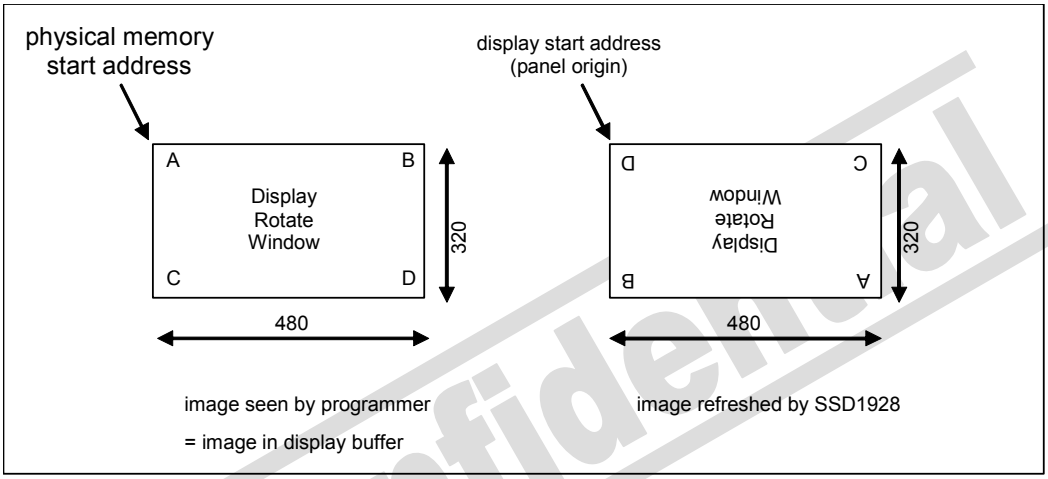
Display Start Address
The display refresh circuitry starts at pixel "B", therefore the Main Window Display Start Address registers (REG[74h], REG[75h], REG[76h]) must be programmed with the address of pixel "B".
To calculate the value of the address of pixel "B" use the following formula (assumes 8bpp color depth).
Main Window Display Start Address bits 16-0
= ((Image address + (panel height x bpp ÷ 8)) ÷ 4) –1
= ((0 + (320 pixels x 8 bpp ÷ 8)) ÷ 4) – 1
= 79 (4Fh)

Line Address Offset
The Main Window Line Address Offset register (REG[78h], REG[79h]) is based on the display width and programmed using the following formula.
Main Window Line Address Offset bits 9-0
= Display width in pixels ÷ (32 ÷ bpp)
= 320 pixels ÷ (32 ÷ 8 bpp)
= 80 (50h)

2.1.15.2 180° Display Rotate Mode

The following figure shows how the programmer sees a 480x320 landscape image and how the image is being displayed. The application image is written to the SSD1928 in the following sense: A–B–C–D. The display is refreshed by the SSD1928 in the following sense: D-C-B-A.

Figure 2-24: Relationship Between The Screen Image and the Image Refreshed in 180° Display Rotate Mode.



Enable 180° Display Rotate Mode
Set Display Rotate Mode Select bits to 10 (REG[71h] bits 1:0 = 10).

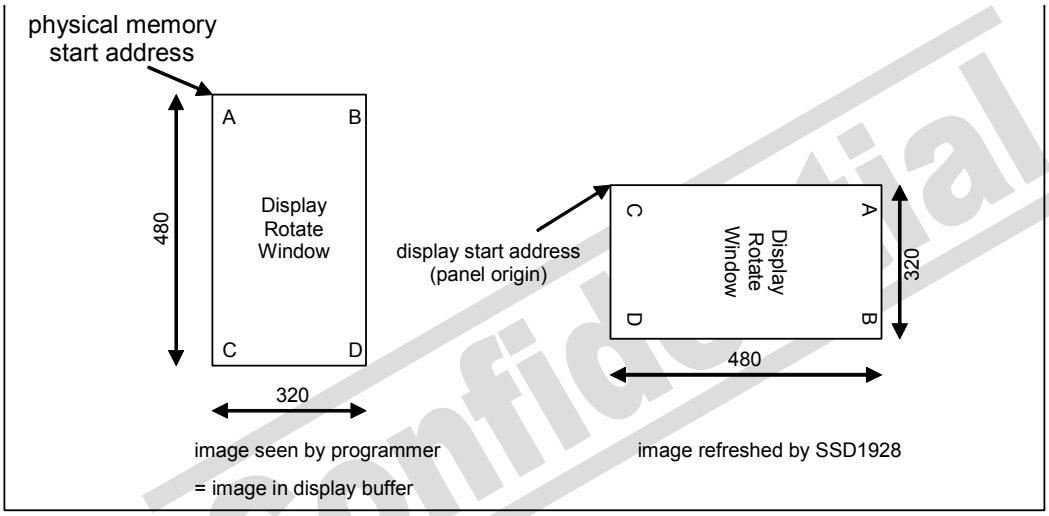
Display Start Address
The display refresh circuitry starts at pixel "D", therefore the Main Window Display Start Address registers (REG[74h], REG[75h], REG[76h]) must be programmed with the address of pixel "D".
To calculate the value of the address of pixel "D" use the following formula (assumes 8bpp color depth).
Main Window Display Start Address bits 16-0
= ((Image address + (image width x (panel height – 1) + panel width) x bpp ÷ 8) ÷ 4) –1
= ((0 + (480 pixels x 319 pixels + 480 pixels) x 8 bpp ÷ 8) ÷ 4) – 1
= 38399 (95FFh)

Line Address Offset
The Main Window Line Address Offset register (REG[78h], REG[79h]) is based on the display width and programmed using the following formula.
Main Window Line Address Offset bits 9-0
= Display width in pixels ÷ (32 ÷ bpp)
= 480 pixels ÷ (32 ÷ 8 bpp)
= 120 (78h)

2.1.15.3 270° Display Rotate Mode

The following figure shows how the programmer sees a 320x480 rotated image and how the image is being displayed. The application image is written to the SSD1928 in the following sense: A–B–C–D. The display is refreshed by the SSD1928 in the following sense: C-A-D-B.

Figure 2-25: Relationship Between The Screen Image and the Image Refreshed in 270° Display Rotate Mode.



Enable 270° Display Rotate Mode
Set Display Rotate Mode Select bits to 11 (REG[71h] bits 1:0 = 11).

Display Start Address
The display refresh circuitry starts at pixel "C", therefore the Main Window Display Start Address registers (REG[74h], REG[75h], REG[76h]) must be programmed with the address of pixel "C".
To calculate the value of the address of pixel "C" use the following formula (assumes 8bpp color depth).
Main Window Display Start Address bits 16-0
= (Image address + ((panel width – 1) x image width x bpp ÷ 8) ÷ 4)
= (0 + ((480 pixels – 1) x 320 pixels x 8 bpp ÷ 8) ÷ 4)
= 38320 (95B0h)

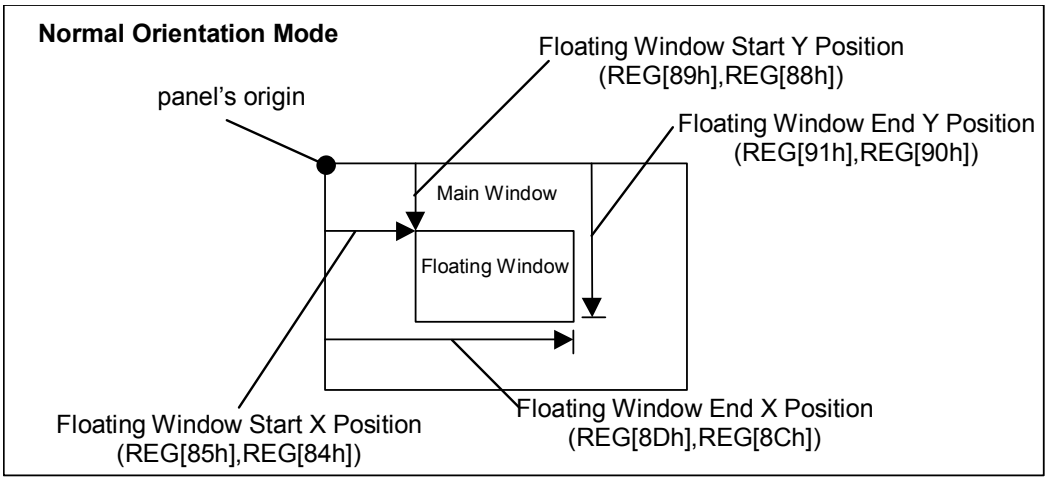
Line Address Offset
The Main Window Line Address Offset register (REG[78h], REG[79h]) is based on the display width and programmed using the following formula.
Main Window Line Address Offset bits 9-0
= Display width in pixels ÷ (32 ÷ bpp)
= 320 pixels ÷ (32 ÷ 8 bpp)
= 80 (50h)

2.1.16 Floating Window Mode

This mode enables a floating window within the main display window. The floating window can be positioned anywhere within the virtual display and is controlled through the Floating Window control registers (REG[7Ch] through REG[91h]). The floating window retains the same color depth and display orientation as the main window.

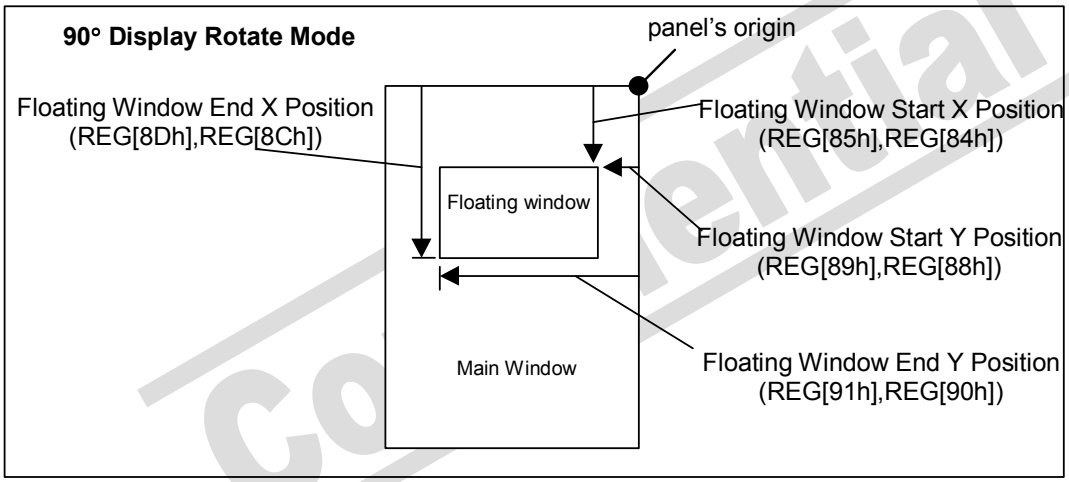
The following diagram shows an example of a floating window within a main window and the registers used to position it.

Figure 2-26: Floating Window with Display Rotate Mode disabled



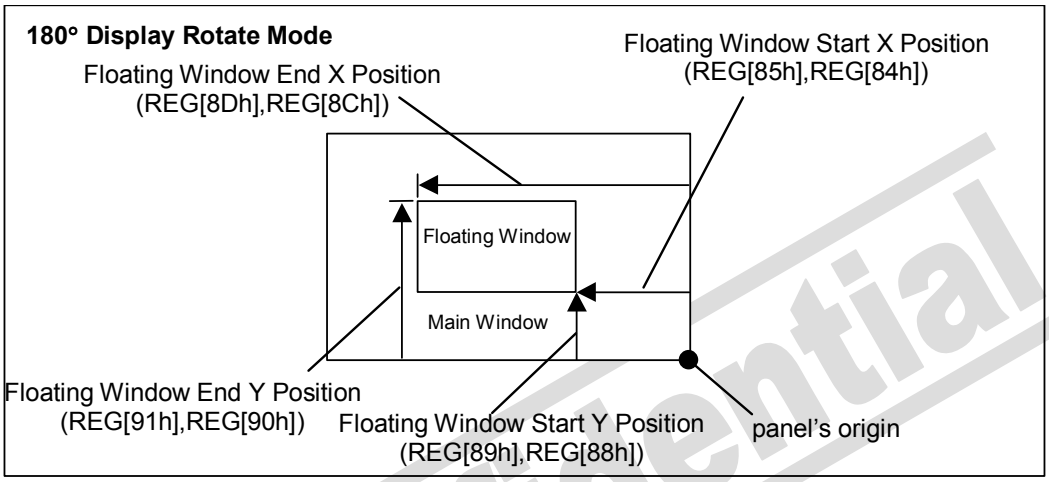
2.1.16.1 Floating window under 90°Display Rotate Mode

Figure 2-27: Floating Window with Display Rotate Mode 90° enabled



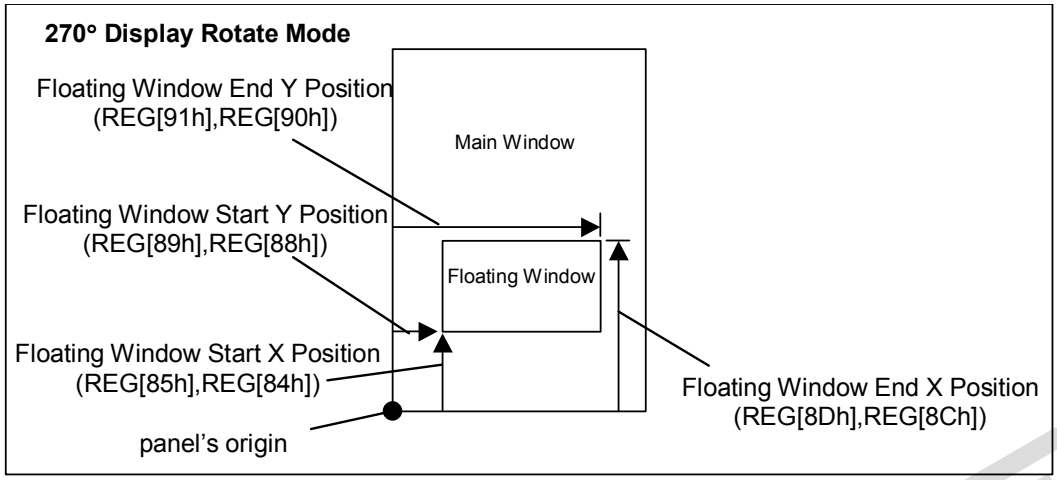
2.1.16.2 Floating window under 180°Display Rotate Mode

Figure 2-28: Floating Window with Display Rotate Mode 180° enabled



2.1.16.3 Floating window under 270°Display Rotate Mode

Figure 2-29: Floating Window with Display Rotate Mode 270° enabled



Floating Window Display Start Address Register 0 **REG[7Ch]**

Bit	7	6	5	4	3	2	1	0
	Floating Window Display Start Address Bit 7	Floating Window Display Start Address Bit 6	Floating Window Display Start Address Bit 5	Floating Window Display Start Address Bit 4	Floating Window Display Start Address Bit 3	Floating Window Display Start Address Bit 2	Floating Window Display Start Address Bit 1	Floating Window Display Start Address Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Floating Window Display Start Address Register 1 **REG[7Dh]**

Bit	7	6	5	4	3	2	1	0
	Floating Window Display Start Address Bit 15	Floating Window Display Start Address Bit 14	Floating Window Display Start Address Bit 13	Floating Window Display Start Address Bit 12	Floating Window Display Start Address Bit 11	Floating Window Display Start Address Bit 10	Floating Window Display Start Address Bit 9	Floating Window Display Start Address Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Floating Window Display Start Address Register 2 **REG[7Eh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Floating Window Display Start Address Bit 16
Type	NA	NA	NA	NA	NA	NA	NA	RW
Reset state	0	0	0	0	0	0	0	0

REG[7Eh] bit 0,
REG[7Dh] bits 7-0,
REG[7Ch] bits 7-0

Floating Window Display Start Address Bits [16:0]
These bits form the 17-bit address for the starting double-word of the floating window.

Note that this is a double-word (32-bit) address. An entry of 00000h into these registers represents the first double-word of display memory, an entry of 00001h represents the second double-word of the display memory, and so on.

Note
These bits will not effective until the Floating Window Enable bit is set to 1 (REG[71h] bit 4=1).

Floating Window Line Address Offset Register 0 **REG[80h]**

Bit	7	6	5	4	3	2	1	0
	Floating Window Line Address Offset Bit 7	Floating Window Line Address Offset Bit 6	Floating Window Line Address Offset Bit 5	Floating Window Line Address Offset Bit 4	Floating Window Line Address Offset Bit 3	Floating Window Line Address Offset Bit 2	Floating Window Line Address Offset Bit 1	Floating Window Line Address Offset Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Floating Window Line Address Offset Register 1 **REG[81h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Floating Window Line Address Offset Bit 9	Floating Window Line Address Offset Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[81h] bits 1-0,
REG[80h] bits 7-0

Floating Window Line Address Offset Bits [9:0]
These bits are the LCD display's 10-bit address offset from the starting double-word of line "n" to the starting double-word of line "n + 1" for the floating window.
Note
(1) This is a 32-bit address increment.

Note
(1) These bits will not effective until the Floating Window Enable bit is set to 1 (REG[71h] bit 4=1).

Floating Window Start Position X Register 0 **REG[84h]**

Bit	7	6	5	4	3	2	1	0
	Floating Window Start X Position Bit 7	Floating Window Start X Position Bit 6	Floating Window Start X Position Bit 5	Floating Window Start X Position Bit 4	Floating Window Start X Position Bit 3	Floating Window Start X Position Bit 2	Floating Window Start X Position Bit 1	Floating Window Start X Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Floating Window Start Position X Register 1 **REG[85h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Floating Window Start X Position Bit 9	Floating Window Start X Position Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[85h] bits 1-0, REG[84h] bits 7-0

Floating Window Start Position X Bits [9:0]
These bits determine the start position X of the floating window in relation to the origin of the panel. Due to the SSD1928 Display Rotate feature, the start position X may not be a horizontal position value (only true in 0° and 180° rotation). For further information on defining the value of the Start Position X register, see Section "Floating Window Mode" in datasheet.

The value of register is also increased differently based on the display orientation. For 0° and 180° Display Rotate Mode, the start position X is incremented by x pixels where x is relative to the current color depth. Refer to Table 2-28: 32-bit Address X Increments for Various Color Depths. For 90° and 270° Display Rotate Mode, the start position X is incremented by 1 line.

Depending on the color depth, some of the higher bits in this register are unused because the maximum horizontal display width is 1024 pixels.

Note
(1) These bits will not effective until the Floating Window Enable bit is set to 1 (REG[71h] bit 4=1).

Table 2-28: 32-bit Address X Increments for Various Color Depths

Color Depth (bpp)	Pixel Increment (x)
1	32
2	16
4	8
8	4
16	2
32	1

Floating Window Start Position Y Register 0 **REG[88h]**

Bit	7	6	5	4	3	2	1	0
	Floating Window Start Y Position Bit 7	Floating Window Start Y Position Bit 6	Floating Window Start Y Position Bit 5	Floating Window Start Y Position Bit 4	Floating Window Start Y Position Bit 3	Floating Window Start Y Position Bit 2	Floating Window Start Y Position Bit 1	Floating Window Start Y Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Floating Window Start Position Y Register 1 **REG[89h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Floating Window Start Y Position Bit 9	Floating Window Start Y Position Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[89h] bits 1-0, REG[88h] bits 7-0

Floating Window Start Position Y Bits [9:0]
These bits determine the start position Y of the floating window in relation to the origin of the panel. Due to the SSD1928 Display Rotate feature, the start position Y may not be a vertical position value (only true in 0° and 180° Floating Window). For further information on defining the value of the Start Position Y register, see Section "Floating Window Mode" in datasheet.

The register is also incremented according to the display orientation. For 0° and 180° Display Rotate Mode, the start position Y is incremented by 1 line. For 90° and 270° Display Rotate Mode, the start position Y is incremented by y pixels where y is relative to the current color depth. Refer to Table 2-29: 32-bit Address Y Increments for Various Color Depths.

Depending on the color depth, some of the higher bits in this register are unused because the maximum vertical display height is 1024 pixels.

Note
(1) These bits will not effective until the Floating Window Enable bit is set to 1 (REG[71h] bit 4=1).

Table 2-29: 32-bit Address Y Increments for Various Color Depths

Color Depth (bpp)	Pixel Increment (y)
1	32
2	16
4	8
8	4
16	2
32	1

Floating Window End Position X Register 0 **REG[8Ch]**

Bit	7	6	5	4	3	2	1	0
	Floating Window End X Position Bit 7	Floating Window End X Position Bit 6	Floating Window End X Position Bit 5	Floating Window End X Position Bit 4	Floating Window End X Position Bit 3	Floating Window End X Position Bit 2	Floating Window End X Position Bit 1	Floating Window End X Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Floating Window End Position X Register 1 **REG[8Dh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Floating Window End X Position Bit 9	Floating Window End X Position Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[8Dh] bits 1-0, REG[8Ch] bits 7-0

Floating Window End Position X Bits [9:0]
These bits determine the end position X of the floating window in relation to the origin of the panel. Due to the SSD1928 Display Rotate feature, the end position X may not be a horizontal position value (only true in 0° and 180° rotation). For further information on defining the value of the End Position X register, see "Floating Window Mode" in datasheet.

The value of register is also increased according to the display orientation. For 0° and 180° Display Rotate Mode, the end position X is incremented by x pixels where x is relative to the current color depth. Refer to Table 2-30: 32-bit Address X Increments for Various Color Depths. For 90° and 270° Display Rotate Mode, the end position X is incremented by 1 line.

Depending on the color depth, some of the higher bits in this register are unused because the maximum horizontal display width is 1024 pixels.

Note
(1) These bits will not effective until the Floating Window Enable bit is set to 1 (REG[71h] bit 4=1).

Table 2-30: 32-bit Address X Increments for Various Color Depths

Color Depth (bpp)	Pixel Increment (x)
1	32
2	16
4	8
8	4
16	2
32	1

Floating Window End Position Y Register 0 **REG[90h]**

Bit	7	6	5	4	3	2	1	0
	Floating Window End Y Position Bit 7	Floating Window End Y Position Bit 6	Floating Window End Y Position Bit 5	Floating Window End Y Position Bit 4	Floating Window End Y Position Bit 3	Floating Window End Y Position Bit 2	Floating Window End Y Position Bit 1	Floating Window End Y Position Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Floating Window End Position Y Register 1 **REG[91h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Floating Window End Y Position Bit 9	Floating Window End Y Position Bit 8
Type	NA	NA	NA	NA	NA	NA	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[91h] bits 1-0, REG[90h] bits 7-0

Floating Window End Position Y Bits [9:0]
These bits determine the end position Y of the floating window in relation to the origin of the panel. Due to the SSD1928 Display Rotate feature, the end position Y may not be a vertical position value (only true in 0° and 180° Display Rotate Mode). For further information on defining the value of the End Position Y register, see Section "Floating Window Mode" in datasheet.

The value of register is also increased according to the display orientation. For 0° and 180° Display Rotate Mode, the end position Y is incremented by 1 line. For 90° and 270° Display Rotate Mode, the end position Y is incremented by y pixels where y is relative to the current color depth. Refer to Table 2-31: 32-bit Address Y Increments for Various Color Depths.

Depending on the color depth, some of the higher bits in this register are unused because the maximum vertical display height is 1024 pixels.

Note
(1) These bits will not effective until the Floating Window Enable bit is set to 1 (REG[71h] bit 4=1).

Table 2-31: 32-bit Address Y Increments for Various Color Depths

Color Depth (bpp)	Pixel Increment (y)
1	32
2	16
4	8
8	4
16	2
32	1

2.1.17 Cursor Mode

This mode enables two cursors on the main display window. The cursors can be positioned anywhere within the display and are controlled through Cursor Mode registers (REG[C0h] through REG[111h]). Cursor support is available only at 4/8/16/32-bpp display modes.

Each cursor pixel is 2-bit and the indexing scheme is as follows:

Table 2-32: Indexing scheme for Hardware Cursor

Value		Color of Cursor 1 / Cursor 2
00		Transparent
01	Content of color index 1 register	(REG[E31h-E0h] / REG[10Bh-108h])
10	Content of color index 2 register	(REG[E7h-E4h] / REG[10Fh-10Ch])
11	Content of color index 3 register	(REG[EBh-E8h] / REG[1131h-110h])

Three 16-bit color index registers (REG[E0h] through REG[E9h] and REG[108h] through REG[111h]) have been implemented for each cursor. Only the lower portion of the color index register is used in 4/8-bpp display modes. The LUT is bypassed and the color data is directly mapped for 16/32-bpp display mode.

4 Bit-per-pixel

15 ... 12	11 ... 8	7 ... 4	3 ... 0
Don't Care			4-bit Color Index

8 Bit-per-pixel

15 ... 12	11 ... 8	7 ... 4	3 ... 0
Don't Care		8-bit Color Index	

16 Bit-per-pixel (the index registers represents the 16-bit color component)

15 ... 13	12 ... 8	7 ... 3	2 ... 0
Green Component Bits 2-0	Blue Component Bits 4-0	Red Component Bits 4-0	Green Component Bits 5-3

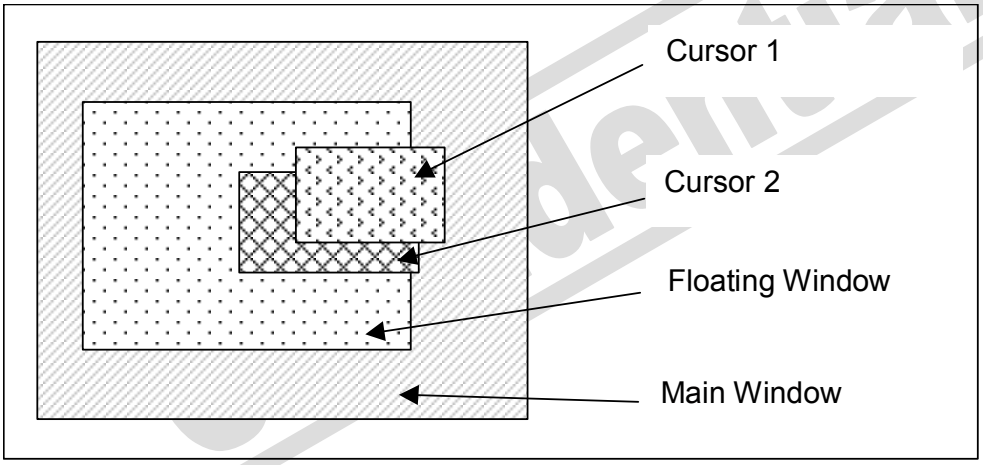
32 Bit-per-pixel (the index registers represents the 32-bit color component)

31 ... 24	23 ... 16
Alpha Component Bits 7-0	Blue Component Bits 5-3

15 ... 8	7 ... 0
Green Component Bits 7-0	Red Component Bits 5-3

The display precedence is Cursor1 > Cursor2 > Floating window > Main Window.

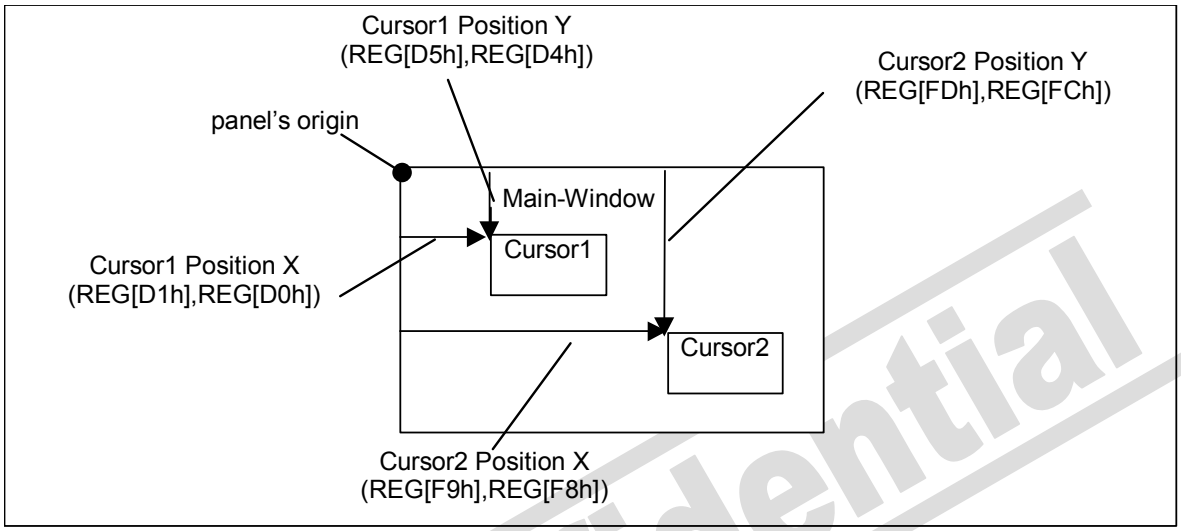
Figure 2-30: Display Precedence in Hardware Cursor



Note
(1) The minimum size varies for different color depths and display orientations.

The cursors retains the same color depth and display orientation as the main window. The following diagram shows an example of two cursors within a main window and the registers used to position it.

Figure 2-31: Cursors on the main window



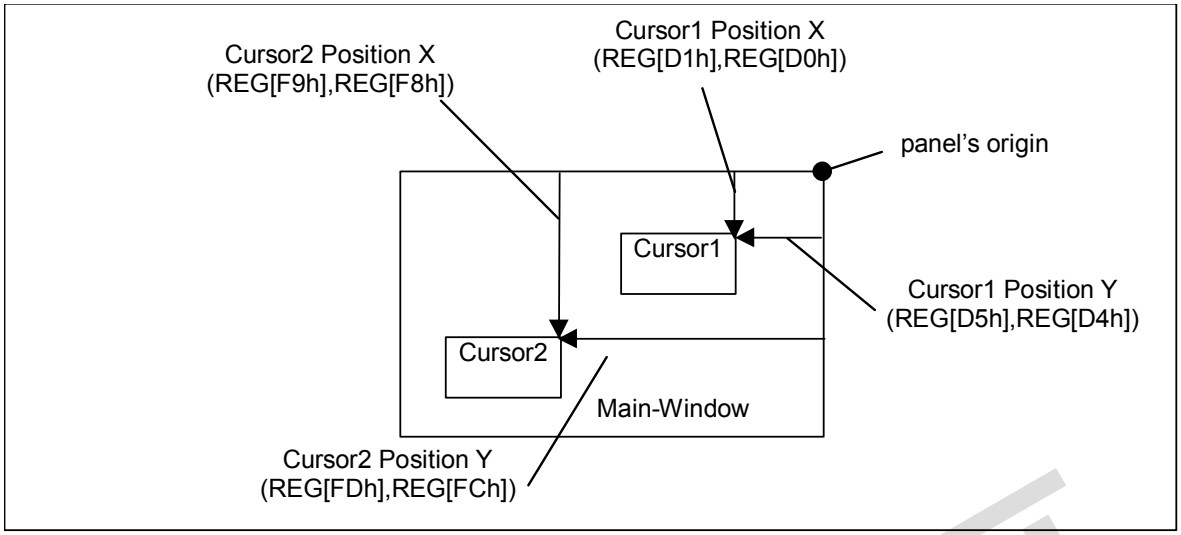
Assume the pixel data stores start at address n, which n must be divisible by 4 (i.e. aligned to 32-bit boundary). In this example, a 16x16 cursor is displayed which each cursor index is defined by x and y coordinate, C(y,x). Following are the pixel format.

For 4/8/16 Bit-per-pixel

	7 6	5 4	3 2	1 0
Addr. n	C(0,0)	C(0,1)	C(0,2)	C(0,3)
Addr. n + 1	C(0,4)	C(0,5)	C(0,6)	C(0,7)
Addr. n + 2	C(0,8)	C(0,9)	C(0,10)	C(0,11)
Addr. n + 3	C(0,12)	C(0,13)	C(0,14)	C(0,15)
Addr. n + 4	C(1,0)	C(1,1)	C(1,2)	C(1,3)
	•			
	•			
	•			
Addr. n + 60	C(15,0)	C(15,1)	C(15,2)	C(15,3)
Addr. n + 61	C(15,4)	C(15,5)	C(15,6)	C(15,7)
Addr. n + 62	C(15,8)	C(15,9)	C(15,10)	C(15,11)
Addr. n + 63	C(15,12)	C(15,13)	C(15,14)	C(15,15)

2.1.17.1 Cursor with 90° Display Rotate Mode

Figure 2-32: Cursors with Display Rotate Mode 90° enabled



Assume the pixel data stores start at address n, which n must be divisible by 4 (i.e. aligned to 32-bit boundary). In this example, a 16x16 cursor is displayed which each cursor index is defined x and y coordinate, C(y,x). Following are the pixel format.

For 4 Bit-per-pixel

	7 6	5 4	3 2	1 0
Addr. n	C(0,8)	C(0,9)	C(0,10)	C(0,11)
Addr. n + 1	C(0,12)	C(0,13)	C(0,14)	C(0,15)
Addr. n + 2	C(1,8)	C(1,9)	C(1,10)	C(1,11)
Addr. n + 3	C(1,12)	C(1,13)	C(1,14)	C(1,15)
•				
•				
•				
Addr. n + 28	C(14,8)	C(14,9)	C(14,10)	C(14,11)
Addr. n + 29	C(14,12)	C(14,13)	C(14,14)	C(14,15)
Addr. n + 30	C(15,8)	C(15,9)	C(15,10)	C(15,11)
Addr. n + 31	C(15,12)	C(15,13)	C(15,14)	C(15,15)
Addr. n + 32	C(0,0)	C(0,1)	C(0,2)	C(0,3)
Addr. n + 33	C(0,4)	C(0,5)	C(0,6)	C(0,7)
Addr. n + 34	C(1,0)	C(1,1)	C(1,2)	C(1,3)
Addr. n + 35	C(1,4)	C(1,5)	C(1,6)	C(1,7)
•				
•				
•				
Addr. n + 60	C(14,0)	C(14,1)	C(14,2)	C(14,3)
Addr. n + 61	C(14,4)	C(14,5)	C(14,6)	C(14,7)
Addr. n + 62	C(15,0)	C(15,1)	C(15,2)	C(15,3)
Addr. n + 63	C(15,4)	C(15,5)	C(15,6)	C(15,7)

For 8 Bit-per-pixel

7	6	5	4	3	2	1	0

Addr. n	C(0,12)	C(0,13)	C(0,14)	C(0,15)
Addr. n + 1	C(1,12)	C(1,13)	C(1,14)	C(1,15)
Addr. n + 2	C(2,12)	C(2,13)	C(2,14)	C(2,15)
Addr. n + 3	C(3,12)	C(3,13)	C(3,14)	C(3,15)

•
•
•

Addr. n + 12	C(12,12)	C(12,13)	C(12,14)	C(12,15)
Addr. n + 13	C(13,12)	C(13,13)	C(13,14)	C(13,15)
Addr. n + 14	C(14,12)	C(14,13)	C(14,14)	C(14,15)
Addr. n + 15	C(15,12)	C(15,13)	C(15,14)	C(15,15)
Addr. n + 16	C(0,8)	C(0,9)	C(0,10)	C(0,11)
Addr. n + 17	C(1,8)	C(1,9)	C(1,10)	C(1,11)
Addr. n + 18	C(2,8)	C(2,9)	C(2,10)	C(2,11)
Addr. n + 19	C(3,8)	C(3,9)	C(3,10)	C(3,11)

•
•
•

Addr. n + 60	C(12,0)	C(12,1)	C(12,2)	C(12,3)
Addr. n + 61	C(13,0)	C(13,1)	C(13,2)	C(13,3)
Addr. n + 62	C(14,0)	C(14,1)	C(14,2)	C(14,3)
Addr. n + 63	C(15,0)	C(15,1)	C(15,2)	C(15,3)

For 16 Bit-per-pixel

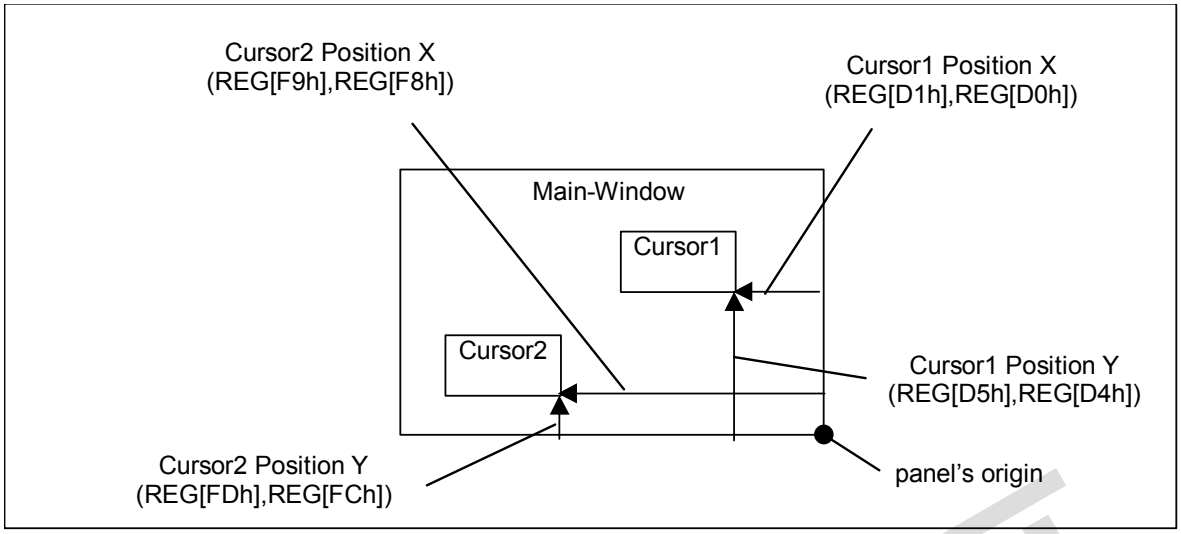
	7 6	5 4	3 2	1 0
Addr. n	C(0,14)	C(0,15)	C(1,14)	C(1,15)
Addr. n + 1	C(2,14)	C(2,15)	C(3,14)	C(3,15)
Addr. n + 2	C(4,14)	C(4,15)	C(5,14)	C(5,15)
Addr. n + 3	C(6,14)	C(6,15)	C(7,14)	C(7,15)
Addr. n + 4	C(8,14)	C(8,15)	C(9,14)	C(9,15)
Addr. n + 5	C(10,14)	C(10,15)	C(11,14)	C(11,15)
Addr. n + 6	C(12,14)	C(12,15)	C(12,14)	C(12,15)
Addr. n + 7	C(14,14)	C(14,15)	C(15,14)	C(15,15)
Addr. n + 8	C(0,12)	C(0,13)	C(1,12)	C(1,13)
Addr. n + 9	C(2,12)	C(2,13)	C(3,12)	C(3,13)
Addr. n + 10	C(4,12)	C(4,13)	C(5,12)	C(5,13)
Addr. n + 11	C(6,12)	C(6,13)	C(7,12)	C(7,13)

•
•
•

Addr. n + 60	C(8,0)	C(8,1)	C(9,0)	C(9,1)
Addr. n + 61	C(10,0)	C(10,1)	C(11,0)	C(11,1)
Addr. n + 62	C(12,0)	C(12,1)	C(12,0)	C(12,1)
Addr. n + 63	C(14,0)	C(14,1)	C(15,0)	C(15,1)

2.1.17.2 Cursor with 180° Display Rotate Mode

Figure 2-33: Cursors with Display Rotate Mode 180° enabled



Assume the pixel data stores start at address n, which n must be divisible by 4 (i.e. aligned to 32-bit boundary). In this example, a 16x16 cursor is displayed which each cursor index is defined by x and y coordinate, C(y,x). Following are the pixel format.

For 4 Bit-per-pixel

	7 6	5 4	3 2	1 0
Addr. n	C(15,8)	C(15,9)	C(15,10)	C(15,11)
Addr. n + 1	C(15,12)	C(15,13)	C(15,14)	C(15,15)
Addr. n + 2	C(15,0)	C(15,1)	C(15,2)	C(15,3)
Addr. n + 3	C(15,4)	C(15,5)	C(15,6)	C(15,7)
Addr. n + 4	C(14,8)	C(14,9)	C(14,10)	C(14,11)
•				
•				
•				
Addr. n + 60	C(0,8)	C(0,9)	C(0,10)	C(0,11)
Addr. n + 61	C(0,12)	C(0,13)	C(0,14)	C(0,15)
Addr. n + 62	C(0,0)	C(0,1)	C(0,2)	C(0,3)
Addr. n + 63	C(0,4)	C(0,5)	C(0,6)	C(0,7)

For 8 Bit-per-pixel

	7 6	5 4	3 2	1 0
Addr. n	C(15,12)	C(15,13)	C(15,14)	C(15,15)
Addr. n + 1	C(15,8)	C(15,9)	C(15,10)	C(15,11)
Addr. n + 2	C(15,4)	C(15,5)	C(15,6)	C(15,7)
Addr. n + 3	C(15,0)	C(15,1)	C(15,2)	C(15,3)
Addr. n + 4	C(14,12)	C(14,13)	C(14,14)	C(14,15)
•				
•				
•				

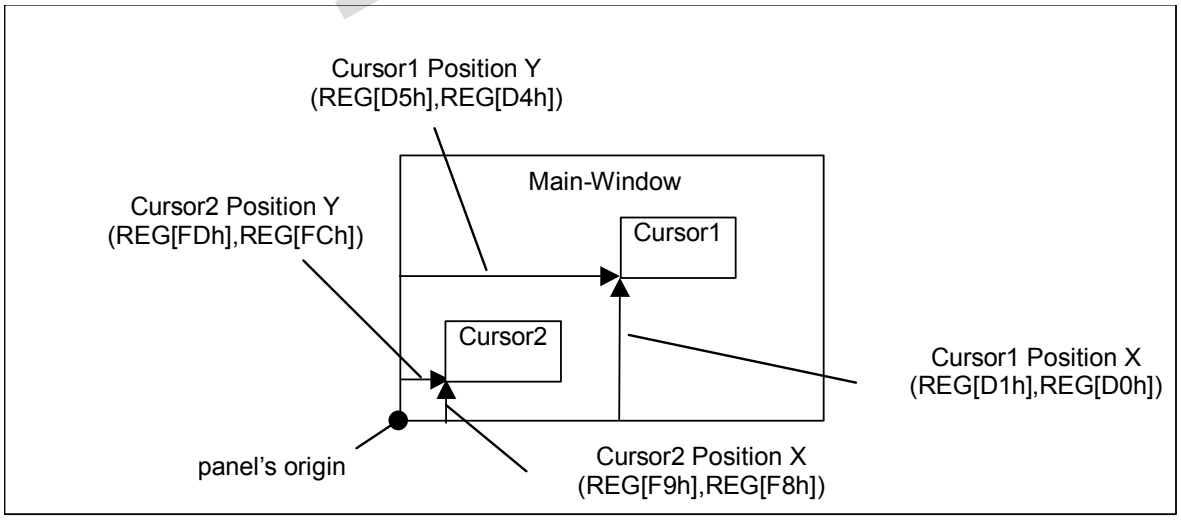
Addr. n + 60	C(0,12)	C(0,13)	C(0,14)	C(0,15)
Addr. n + 61	C(0,8)	C(0,9)	C(0,10)	C(0,11)
Addr. n + 62	C(0,4)	C(0,5)	C(0,6)	C(0,7)
Addr. n + 63	C(0,0)	C(0,1)	C(0,2)	C(0,3)

For 16 Bit-per-pixel

	7 6	5 4	3 2	1 0
Addr. n	C(15,14)	C(15,15)	C(15,12)	C(15,13)
Addr. n + 1	C(15,10)	C(15,11)	C(15,8)	C(15,9)
Addr. n + 2	C(15,6)	C(15,7)	C(15,4)	C(15,5)
Addr. n + 3	C(15,2)	C(15,3)	C(15,0)	C(15,1)
Addr. n + 4	C(14,14)	C(14,15)	C(14,12)	C(14,13)
•				
•				
•				
Addr. n + 60	C(0,14)	C(0,15)	C(0,12)	C(0,13)
Addr. n + 61	C(0,10)	C(0,11)	C(0,8)	C(0,9)
Addr. n + 62	C(0,6)	C(0,7)	C(0,4)	C(0,5)
Addr. n + 63	C(0,2)	C(0,3)	C(0,0)	C(0,1)

2.1.17.3 Cursor with 270° Display Rotate Mode

Figure 2-34: Cursors with Display Rotate Mode 270° enabled



Assume the pixel data stores start at address n, which n must be divisible by 4 (i.e. aligned to 32-bit boundary). In this example, a 16x16 cursor is displayed which each cursor index is defined by x and y coordinate, C(y,x). Following are the pixel format.

Fro 4 Bit-per-pixel

	7 6	5 4	3 2	1 0
Addr. n	C(15,0)	C(15,1)	C(15,2)	C(15,3)
Addr. n + 1	C(15,4)	C(15,5)	C(15,6)	C(15,7)
Addr. n + 2	C(14,0)	C(14,1)	C(14,2)	C(14,3)

Addr. n + 3	C(14,4)	C(14,5)	C(14,6)	C(14,7)

•
•
•

Addr. n + 28	C(1,0)	C(1,1)	C(1,2)	C(1,3)
Addr. n + 29	C(1,4)	C(1,5)	C(1,6)	C(1,7)
Addr. n + 30	C(0,0)	C(0,1)	C(0,2)	C(0,3)
Addr. n + 31	C(0,4)	C(0,5)	C(0,6)	C(0,7)
Addr. n + 32	C(15,8)	C(15,9)	C(15,10)	C(15,11)
Addr. n + 33	C(15,12)	C(15,13)	C(15,14)	C(15,15)
Addr. n + 34	C(14,8)	C(14,9)	C(14,10)	C(14,11)
Addr. n + 35	C(14,12)	C(14,13)	C(14,14)	C(14,15)

•
•
•

Addr. n + 60	C(1,8)	C(1,9)	C(1,10)	C(1,11)
Addr. n + 61	C(1,12)	C(1,13)	C(1,14)	C(1,15)
Addr. n + 62	C(0,8)	C(0,9)	C(0,10)	C(0,11)
Addr. n + 63	C(0,12)	C(0,13)	C(0,14)	C(0,15)

For 8 Bit-per-pixel

	7	6	5	4	3	2	1	0
Addr. n	C(15,0)		C(15,1)		C(15,2)		C(15,3)	
Addr. n + 1	C(14,0)		C(14,1)		C(14,2)		C(14,3)	
Addr. n + 2	C(13,0)		C(13,1)		C(13,2)		C(13,3)	
Addr. n + 3	C(12,0)		C(12,1)		C(12,2)		C(12,3)	

•
•
•

Addr. n + 12	C(3,0)	C(3,1)	C(3,2)	C(3,3)
Addr. n + 13	C(2,0)	C(2,1)	C(2,2)	C(2,3)
Addr. n + 14	C(1,0)	C(1,1)	C(1,2)	C(1,3)
Addr. n + 15	C(0,0)	C(0,1)	C(0,2)	C(0,3)
Addr. n + 16	C(15,4)	C(15,5)	C(15,6)	C(15,7)
Addr. n + 17	C(14,4)	C(14,5)	C(14,6)	C(14,7)
Addr. n + 18	C(13,4)	C(13,5)	C(13,6)	C(13,7)
Addr. n + 19	C(12,4)	C(12,5)	C(12,6)	C(12,7)

•
•
•

Addr. n + 60	C(3,12)	C(3,13)	C(3,14)	C(3,15)
Addr. n + 61	C(2,12)	C(2,13)	C(2,14)	C(2,15)
Addr. n + 62	C(1,12)	C(1,13)	C(1,14)	C(1,15)
Addr. n + 63	C(0,12)	C(0,13)	C(0,14)	C(0,15)

For 16 Bit-per-pixel

	7	6	5	4	3	2	1	0
Addr. n	C(15,0)		C(15,1)		C(14,0)		C(14,1)	
Addr. n + 1	C(13,0)		C(13,1)		C(12,0)		C(12,1)	
Addr. n + 2	C(11,0)		C(11,1)		C(10,0)		C(10,1)	

Addr. n + 3	C(9,0)	C(9,1)	C(8,0)	C(8,1)
Addr. n + 4	C(7,0)	C(7,1)	C(6,0)	C(6,1)
Addr. n + 5	C(5,0)	C(5,1)	C(4,0)	C(4,1)
Addr. n + 6	C(3,0)	C(3,1)	C(2,0)	C(2,1)
Addr. n + 7	C(1,0)	C(1,1)	C(0,0)	C(0,1)
Addr. n + 8	C(15,2)	C(15,3)	C(14,2)	C(14,3)
Addr. n + 9	C(13,2)	C(13,3)	C(12,2)	C(12,3)
Addr. n + 10	C(11,2)	C(11,3)	C(10,2)	C(10,3)
Addr. n + 11	C(9,2)	C(9,3)	C(8,2)	C(8,3)

•
•
•

Addr. n + 60	C(7,14)	C(7,15)	C(6,14)	C(6,15)
Addr. n + 61	C(5,14)	C(5,15)	C(4,14)	C(4,15)
Addr. n + 62	C(3,14)	C(3,15)	C(2,14)	C(2,15)
Addr. n + 63	C(1,14)	C(1,15)	C(0,14)	C(0,15)

Cursor Feature Register **REG[C0h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Enable	Cursor2 Enable	0	0	0	0	0	0
Type	RW	RW	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 7 **Cursor1 Enable**
When this bit = 0 Cursor1 is disabled.
When this bit = 1 Cursor1 is enabled.

Bit 6 **Cursor2 Enable**
When this bit = 0, Cursor2 is disabled.
When this bit = 1, Cursor2 is enabled.

Note
(1) This register is effective for 4/8/16/32 bpp (REG[70h] Bits 2:0 = 010/011/100/101)

Cursor1 Blink Total Register 0 **REG[C4h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Blink Total Bit 7	Cursor1 Blink Total Bit 6	Cursor1 Blink Total Bit 5	Cursor1 Blink Total Bit 4	Cursor1 Blink Total Bit 3	Cursor1 Blink Total Bit 2	Cursor1 Blink Total Bit 1	Cursor1 Blink Total Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Blink Total Register 1 **REG[C5h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor1 Blink Total Bit 9	Cursor1 Blink Total Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[C5h] bits 1-0, REG[C4h] bits 7-0 **Cursor1 Blink Total Bits [9:0]**
This is the total blinking period per frame for cursor1. This register must be set to a non-zero

value in order to make the cursor visible.

Note
(1) These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

Cursor1 Blink On Register 0 **REG[C8h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Blink On Bit 7	Cursor1 Blink On Bit 6	Cursor1 Blink On Bit 5	Cursor1 Blink On Bit 4	Cursor1 Blink On Bit 3	Cursor1 Blink On Bit 2	Cursor1 Blink On Bit 1	Cursor1 Blink On Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Blink On Register 1 **REG[C9h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor1 Blink On Bit 9	Cursor1 Blink On Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[C9h] bits 1-0, REG[C8h] bits 7-0

Cursor1 Blink On Bits [9:0]
This is the blink on frame period for Cursor1.

This register must be set to a non-zero value in order to make the cursor1 visible. Also, cursor1 will start to blink if the following conditions are fulfilled :

Cursor1 Blink Total Bits [9:0] > Cursor1 Blink On Bits [9:0] > 0

To enable cursor1 without blinking, user must program cursor1 blink on register with a non-zero value, and this value must be greater than or equal to Cursor1 Blink Total Register.
Cursor1 Blink On Bits [9:0] > Cursor1 Blink Total Bits [9:0] > 0

These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

Cursor1 Memory Start Register 0 **REG[CCh]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Memory Start Bit 7	Cursor1 Memory Start Bit 6	Cursor1 Memory Start Bit 5	Cursor1 Memory Start Bit 4	Cursor1 Memory Start Bit 3	Cursor1 Memory Start Bit 2	Cursor1 Memory Start Bit 1	Cursor1 Memory Start Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Memory Start Register 1 **REG[CDh]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Memory Start Bit 15	Cursor1 Memory Start Bit 14	Cursor1 Memory Start Bit 13	Cursor1 Memory Start Bit 12	Cursor1 Memory Start Bit 11	Cursor1 Memory Start Bit 10	Cursor1 Memory Start Bit 9	Cursor1 Memory Start Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Memory Start Register 2 **REG[CEh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Cursor1

Bit	7	6	5	4	3	2	1	0
								Memory Start Bit 16
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

REG[CEh] bit 0,
REG[CDh] bits 7-0,
REG[CCh] bits 7-0

Cursor1 Memory Start Bits [16:0]
These bits form the 17-bit address for the starting double-word of the LCD image in the display buffer for the Cursor1 image.

Note
(1) That this is a double-word (32-bit) address.
An entry of 00000h into these registers represents the first double-word of display memory, an entry of 00001h represents the second double-word of the display memory, and so on.
Calculate the Cursor1 Start Address as follows :
Cursor1 Memory Start Bits 16:0
= Cursor Image address ÷ 4 (valid only for Display Rotate Mode 0°)

Note
(1) These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

Cursor1 Position X Register 0 **REG[D0h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Position X Bit 7	Cursor1 Position X Bit 6	Cursor1 Position X Bit 5	Cursor1 Position X Bit 4	Cursor1 Position X Bit 3	Cursor1 Position X Bit 2	Cursor1 Position X Bit 1	Cursor1 Position X Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Position X Register 1 **REG[D1h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor1 Position X Bit 9	Cursor1 Position X Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[D1h] bits 1-0,
REG[D0h] bits 7-0

Cursor1 Position X Bits [9:0]
This is starting position X of Cursor1 image. The definition of this register is same as Floating Window Start Position X Register.

Note
(1) These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

Cursor1 Position Y Register 0 **REG[D4h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Position Y Bit 7	Cursor1 Position Y Bit 6	Cursor1 Position Y Bit 5	Cursor1 Position Y Bit 4	Cursor1 Position Y Bit 3	Cursor1 Position Y Bit 2	Cursor1 Position Y Bit 1	Cursor1 Position Y Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Position Y Register 1 **REG[D5h]**

Bit	7	6	5	4	3	2	1	0

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor1 Position Y Bit 9	Cursor1 Position Y Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[D5h] bits 1-0,
REG[D4h] bits 7-0

Cursor1 Position Y Bits [9:0]
This is starting position Y of Cursor1 image. The definition of this register is same as Floating Window Y Start Position Register.

Note
(1) These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

Cursor1 Horizontal Size Register 0 **REG[D8h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Horizontal Size Bit 7	Cursor1 Horizontal Size Bit 6	Cursor1 Horizontal Size Bit 5	Cursor1 Horizontal Size Bit 4	Cursor1 Horizontal Size Bit 3	Cursor1 Horizontal Size Bit 2	Cursor1 Horizontal Size Bit 1	Cursor1 Horizontal Size Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Horizontal Size Register 1 **REG[D9h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor1 Horizontal Size Bit 9	Cursor1 Horizontal Size Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[D9h] bits 1-0,
REG[D8h] bits 7-0

Cursor1 Horizontal Size Bits [9:0]
These bits specify the horizontal size of Cursor1.

Note
The definition of this register various under different panel orientation and color depth settings. These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

Table 2-33: X Increment Mode for Various Color Depths

Orientation	Main window Color Depths (bpp)	Increment (x)
0˚	4	16 pixels increment e.g. 0000h = 16 pixels; 0001h = 32 pixels
	8	
	16	
	32	
90˚	4	2 lines increment
	8	4 lines increment
	16	8 lines increment
	32	16 lines increment
180˚	4	16 pixels increment
	8	
	16	
	32	
270˚	4	2 lines increment
	8	4 lines increment

	16	8 lines increment
	32	16 lines increment

Cursor1 Vertical Size Register 0 **REG[DCh]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Vertical Size Bit 7	Cursor1 Vertical Size Bit 6	Cursor1 Vertical Size Bit 5	Cursor1 Vertical Size Bit 4	Cursor1 Vertical Size Bit 3	Cursor1 Vertical Size Bit 2	Cursor1 Vertical Size Bit 1	Cursor1 Vertical Size Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Vertical Size Register 1 **REG[DDh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor1 Vertical Size Bit 9	Cursor1 Vertical Size Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[DDh] bits 1-0, REG[DCh] bits 7-0

Cursor1 Vertical Size Bits [9:0]
These bits specify the vertical size of Cursor1.

Note
(1) The definition of this register various under different panel orientation and color depth settings.
These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

Table 2-34: Y Increment Mode for Various Color Depths

Orientation	Main window Color Depths (bpp)	Increment (y)
0˚	4	1 line increment e.g. 0000h = 1 line; 0001h = 2 lines
	8	
	16	
	32	
90˚	4	8 pixels increment
	8	4 pixels increment
	16	2 pixels increment
	32	1 pixel increment
180˚	4	1 line increment
	8	
	16	
	32	
270˚	4	8 pixels increment
	8	4 pixels increment
	16	2 pixels increment
	32	1 pixel increment

Cursor1 Color Index1 Register 0 **REG[E0h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index1 Bit 7	Cursor1 Color Index1 Bit 6	Cursor1 Color Index1 Bit 5	Cursor1 Color Index1 Bit 4	Cursor1 Color Index1 Bit 3	Cursor1 Color Index1 Bit 2	Cursor1 Color Index1 Bit 1	Cursor1 Color Index1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	7	6	5	4	3	2	1	0
Reset state	0	0	0	0	0	0	0	0

Cursor1 Color Index1 Register 1 **REG[E1h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index1 Bit 15	Cursor1 Color Index1 Bit 14	Cursor1 Color Index1 Bit 13	Cursor1 Color Index1 Bit 12	Cursor1 Color Index1 Bit 11	Cursor1 Color Index1 Bit 10	Cursor1 Color Index1 Bit 9	Cursor1 Color Index1 Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Color Index1 Register 2 **REG[E2h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index1 Bit 23	Cursor1 Color Index1 Bit 22	Cursor1 Color Index1 Bit 21	Cursor1 Color Index1 Bit 20	Cursor1 Color Index1 Bit 19	Cursor1 Color Index1 Bit 18	Cursor1 Color Index1 Bit 17	Cursor1 Color Index1 Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Color Index1 Register 3 **REG[E3h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index1 Bit 31	Cursor1 Color Index1 Bit 30	Cursor1 Color Index1 Bit 29	Cursor1 Color Index1 Bit 28	Cursor1 Color Index1 Bit 27	Cursor1 Color Index1 Bit 26	Cursor1 Color Index1 Bit 25	Cursor1 Color Index1 Bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[E3h] bits 7-0,
REG[E2h] bits 7-0,
REG[E1h] bits 7-0,
REG[E0h] bits 7-0

Cursor1 Color Index1 Bits [31:0]
Each cursor pixel is represented by 2 bits. This register stores the color index for pixel value 01 of Cursor1.

Note
(1) These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

For Hardware Cursors operation, see Section "Hardware Cursor Mode" in datasheet.

Cursor1 Color Index2 Register 0 **REG[E4h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index2 Bit 7	Cursor1 Color Index2 Bit 6	Cursor1 Color Index2 Bit 5	Cursor1 Color Index2 Bit 4	Cursor1 Color Index2 Bit 3	Cursor1 Color Index2 Bit 2	Cursor1 Color Index2 Bit 1	Cursor1 Color Index2 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Color Index2 Register 1 **REG[E5h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index2 Bit	Cursor1 Color Index2 Bit	Cursor1 Color Index2 Bit	Cursor1 Color Index2 Bit	Cursor1 Color Index2 Bit	Cursor1 Color Index2 Bit	Cursor1 Color Index2 Bit	Cursor1 Color Index2 Bit

Bit	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Color Index2 Register 2 **REG[E6h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index2 Bit 23	Cursor1 Color Index2 Bit 22	Cursor1 Color Index2 Bit 21	Cursor1 Color Index2 Bit 20	Cursor1 Color Index2 Bit 19	Cursor1 Color Index2 Bit 18	Cursor1 Color Index2 Bit 17	Cursor1 Color Index2 Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Color Index2 Register 3 **REG[E7h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index2 Bit 31	Cursor1 Color Index2 Bit 30	Cursor1 Color Index2 Bit 29	Cursor1 Color Index2 Bit 28	Cursor1 Color Index2 Bit 27	Cursor1 Color Index2 Bit 26	Cursor1 Color Index2 Bit 25	Cursor1 Color Index2 Bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[E7h] bits 7-0,
REG[E6h] bits 7-0,
REG[E5h] bits 7-0,
REG[E4h] bits 7-0

Cursor1 Color Index2 Bits [31:0]
Each cursor pixel is represented by 2 bits. This register stores the color index for pixel value 10 of Cursor1.

Note
(1) These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

For Hardware Cursors operation, see Section "Hardware Cursor Mode" in datasheet.

Cursor1 Color Index3 Register 0 **REG[E8h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index3 Bit 7	Cursor1 Color Index3 Bit 6	Cursor1 Color Index3 Bit 5	Cursor1 Color Index3 Bit 4	Cursor1 Color Index3 Bit 3	Cursor1 Color Index3 Bit 2	Cursor1 Color Index3 Bit 1	Cursor1 Color Index3 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Color Index3 Register 1 **REG[E9h]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index3 Bit 15	Cursor1 Color Index3 Bit 14	Cursor1 Color Index3 Bit 13	Cursor1 Color Index3 Bit 12	Cursor1 Color Index3 Bit 11	Cursor1 Color Index3 Bit 10	Cursor1 Color Index3 Bit 9	Cursor1 Color Index3 Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Color Index3 Register 2 **REG[EAh]**

Bit	7	6	5	4	3	2	1	0
	Cursor1	Cursor1	Cursor1	Cursor1	Cursor1	Cursor1	Cursor1	Cursor1

Bit	7	6	5	4	3	2	1	0
	Color Index3 Bit 23	Color Index3 Bit 22	Color Index3 Bit 21	Color Index3 Bit 20	Color Index3 Bit 19	Color Index3 Bit 18	Color Index3 Bit 17	Color Index3 Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor1 Color Index3 Register 3 **REG[EBh]**

Bit	7	6	5	4	3	2	1	0
	Cursor1 Color Index3 Bit 31	Cursor1 Color Index3 Bit 30	Cursor1 Color Index3 Bit 29	Cursor1 Color Index3 Bit 28	Cursor1 Color Index3 Bit 27	Cursor1 Color Index3 Bit 26	Cursor1 Color Index3 Bit 25	Cursor1 Color Index3 Bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[EBh] bits 7-0,
REG[EAh] bits 7-0,
REG[E9h] bits 7-0,
REG[E8h] bits 7-0

Cursor1 Color Index3 Bits [31:0]
Each cursor pixel is represented by 2 bits. This register stores the color index for pixel value 11 of Cursor1.

Note
(1) These bits will not effective until the Cursor1 Enable bit is set to 1 (REG[C0h] bit 7=1).

For Hardware Cursors operation, see Section "Hardware Cursor Mode" in datasheet.

Cursor2 Blink Total Register 0 **REG[ECh]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Blink Total Bit 7	Cursor2 Blink Total Bit 6	Cursor2 Blink Total Bit 5	Cursor2 Blink Total Bit 4	Cursor2 Blink Total Bit 3	Cursor2 Blink Total Bit 2	Cursor2 Blink Total Bit 1	Cursor2 Blink Total Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Blink Total Register 1 **REG[EDh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor2 Blink Total Bit 9	Cursor2 Blink Total Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[EDh] bits 1-0,
REG[ECh] bits 7-0

Cursor2 Blink Total Bits [9:0]
This is the total blinking period per frame for Cursor2. This register must be set to a non-zero value in order to make the cursor visible.

Note
(1) These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

Cursor2 Blink On Register 0 **REG[F0h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Blink On Bit 7	Cursor2 Blink On Bit 6	Cursor2 Blink On Bit 5	Cursor2 Blink On Bit 4	Cursor2 Blink On Bit 3	Cursor2 Blink On Bit 2	Cursor2 Blink On Bit 1	Cursor2 Blink On Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	7	6	5	4	3	2	1	0
Reset state	0	0	0	0	0	0	0	0

Cursor2 Blink On Register 1 **REG[F1h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor2 Blink On Bit 9	Cursor2 Blink On Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[F1h] bits 1-0, REG[F0h] bits 7-0

Cursor2 Blink On Bits [9:0]

This is the blink on frame period for Cursor2. This register must be set to a non-zero value in order to make the Cursor2 visible. Also, Cursor2 will start to blink if the following conditions are fulfilled:

Cursor2 Blink Total Bits [9:0] > Cursor2 Blink On Bits [9:0] > 0

To enable Cursor2 without blinking, user must program Cursor2 Blink On Register with a non-zero value, and this value must be greater than or equal to Cursor2 Blink Total Register.

Cursor2 Blink On Bits [9:0] > Cursor2 Blink Total Bits [9:0] > 0

These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

Cursor2 Memory Start Register 0 **REG[F4h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Memory Start Bit 7	Cursor2 Memory Start Bit 6	Cursor2 Memory Start Bit 5	Cursor2 Memory Start Bit 4	Cursor2 Memory Start Bit 3	Cursor2 Memory Start Bit 2	Cursor2 Memory Start Bit 1	Cursor2 Memory Start Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Memory Start Register 1 **REG[F5h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Memory Start Bit 15	Cursor2 Memory Start Bit 14	Cursor2 Memory Start Bit 13	Cursor2 Memory Start Bit 12	Cursor2 Memory Start Bit 11	Cursor2 Memory Start Bit 10	Cursor2 Memory Start Bit 9	Cursor2 Memory Start Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Memory Start Register 2 **REG[F6h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Cursor2 Memory Start Bit 16
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

REG[F6h] bit 0, REG[F5h] bits 7-0, REG[F4h] bits 7-0

Cursor2 Memory Start Bits [16:0]

These bits form the 17-bit address for the starting double-word of the LCD image in the display buffer for the Cursor2 image.

Note that this is a double-word (32-bit) address. An entry of 00000h into these registers represents the first double-word of display memory, an entry of 00001h represents the second double-word of the display memory, and so on.
Calculate the Cursor2 Start Address as follows :

Cursor2 Memory Start Bits 16:0
= Cursor Image address ÷ 4 (valid only for Display Rotate Mode 0°)

Note
(1) These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

Cursor2 Position X Register 0 **REG[F8h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Position X Bit 7	Cursor2 Position X Bit 6	Cursor2 Position X Bit 5	Cursor2 Position X Bit 4	Cursor2 Position X Bit 3	Cursor2 Position X Bit 2	Cursor2 Position X Bit 1	Cursor2 Position X Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Position X Register 1 **REG[F9h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor2 Position X Bit 9	Cursor2 Position X Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[F9h] bits 1-0, REG[F8h] bits 7-0 **Cursor2 Position X Bits [9:0]**
This is starting position X of Cursor2 image. The definition of this register is same as Floating Window Start Position X Register.

Note
(1) These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

Cursor2 Position Y Register 0 **REG[FCh]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Position Y Bit 7	Cursor2 Position Y Bit 6	Cursor2 Position Y Bit 5	Cursor2 Position Y Bit 4	Cursor2 Position Y Bit 3	Cursor2 Position Y Bit 2	Cursor2 Position Y Bit 1	Cursor2 Position Y Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Position Y Register 1 **REG[FDh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor2 Position Y Bit 9	Cursor2 Position Y Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[FDh] bits 1-0, REG[FCh] bits 7-0 **Cursor2 Position Y Bits [9:0]**
This is starting position Y of Cursor2 image. The definition of this register is same as Floating Window Y Start Position Register.

Note

(1) These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

Cursor2 Horizontal Size Register 0 **REG[100h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Horizontal Size Bit 7	Cursor2 Horizontal Size Bit 6	Cursor2 Horizontal Size Bit 5	Cursor2 Horizontal Size Bit 4	Cursor2 Horizontal Size Bit 3	Cursor2 Horizontal Size Bit 2	Cursor2 Horizontal Size Bit 1	Cursor2 Horizontal Size Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Horizontal Size Register 1 **REG[101h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor2 Horizontal Size Bit 9	Cursor2 Horizontal Size Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[101h] bits 1-0, REG[100h] bits 7-0 **Cursor2 Horizontal Size Bits [9:0]**
These bits specify the horizontal size of Cursor2.

Note
(1) The definition of this register various under different panel orientation and color depth settings. Refer to Table 2-33: X Increment Mode for Various Color Depths.
These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

Cursor2 Vertical Size Register 0 **REG[104h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Vertical Size Bit 7	Cursor2 Vertical Size Bit 6	Cursor2 Vertical Size Bit 5	Cursor2 Vertical Size Bit 4	Cursor2 Vertical Size Bit 3	Cursor2 Vertical Size Bit 2	Cursor2 Vertical Size Bit 1	Cursor2 Vertical Size Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Vertical Size Register 1 **REG[105h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Cursor2 Vertical Size Bit 9	Cursor2 Vertical Size Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[105h] bits 1-0, REG[104h] bits 7-0 **Cursor2 Vertical Size Bits [9:0]**
These bits specify the vertical size of Cursor2.

Note
(1) The definition of this register various under different panel orientation and color depth settings. Refer to Table 2-34: Y Increment Mode for Various Color Depths.
These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

Cursor2 Color Index1 Register 0 **REG[108h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2	Cursor2	Cursor2	Cursor2	Cursor2	Cursor2	Cursor2	Cursor2

Bit	7	6	5	4	3	2	1	0
	Color Index1 Bit 7	Color Index1 Bit 6	Color Index1 Bit 5	Color Index1 Bit 4	Color Index1 Bit 3	Color Index1 Bit 2	Color Index1 Bit 1	Color Index1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Color Index1 Register 1 **REG[109h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index1 Bit 15	Cursor2 Color Index1 Bit 14	Cursor2 Color Index1 Bit 13	Cursor2 Color Index1 Bit 12	Cursor2 Color Index1 Bit 11	Cursor2 Color Index1 Bit 10	Cursor2 Color Index1 Bit 9	Cursor2 Color Index1 Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Color Index1 Register 2 **REG[10Ah]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index1 Bit 23	Cursor2 Color Index1 Bit 22	Cursor2 Color Index1 Bit 21	Cursor2 Color Index1 Bit 20	Cursor2 Color Index1 Bit 19	Cursor2 Color Index1 Bit 18	Cursor2 Color Index1 Bit 17	Cursor2 Color Index1 Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Color Index1 Register 3 **REG[10Bh]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index1 Bit 31	Cursor2 Color Index1 Bit 30	Cursor2 Color Index1 Bit 29	Cursor2 Color Index1 Bit 28	Cursor2 Color Index1 Bit 27	Cursor2 Color Index1 Bit 26	Cursor2 Color Index1 Bit 25	Cursor2 Color Index1 Bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[10Bh] bits 7-0
REG[10Ah] bits 7-0
REG[109h] bits 7-0
REG[108h] bits 7-0

Cursor2 Color Index1 Bits [31:0]
Each cursor pixel is represented by 2 bits. This register stores the color index for pixel value 01 of Cursor2.

Note
(1) These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

For Hardware Cursors operation, see Section "Hardware Cursor Mode" in datasheet.

Cursor2 Color Index2 Register 0 **REG[10Ch]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index2 Bit 7	Cursor2 Color Index2 Bit 6	Cursor2 Color Index2 Bit 5	Cursor2 Color Index2 Bit 4	Cursor2 Color Index2 Bit 3	Cursor2 Color Index2 Bit 2	Cursor2 Color Index2 Bit 1	Cursor2 Color Index2 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Color Index2 Register 1 **REG[10Dh]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index2 Bit 15	Cursor2 Color Index2 Bit 14	Cursor2 Color Index2 Bit 13	Cursor2 Color Index2 Bit 12	Cursor2 Color Index2 Bit 11	Cursor2 Color Index2 Bit 10	Cursor2 Color Index2 Bit 9	Cursor2 Color Index2 Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Color Index2 Register 0 **REG[10Eh]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index2 Bit 23	Cursor2 Color Index2 Bit 22	Cursor2 Color Index2 Bit 21	Cursor2 Color Index2 Bit 20	Cursor2 Color Index2 Bit 19	Cursor2 Color Index2 Bit 18	Cursor2 Color Index2 Bit 17	Cursor2 Color Index2 Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Color Index2 Register 1 **REG[10Fh]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index2 Bit 31	Cursor2 Color Index2 Bit 30	Cursor2 Color Index2 Bit 29	Cursor2 Color Index2 Bit 28	Cursor2 Color Index2 Bit 27	Cursor2 Color Index2 Bit 26	Cursor2 Color Index2 Bit 25	Cursor2 Color Index2 Bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[10Fh] bits 7-0
REG[10Eh] bits 7-0
REG[10Dh] bits 7-0
REG[10Ch] bits 7-0

Cursor2 Color Index2 Bits [31:0]
Each cursor pixel is represented by 2 bits. This register stores the color index for pixel value 10 of Cursor2.

Note
(1) These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

For Hardware Cursors operation, see Section "Hardware Cursor Mode" in datasheet.

Cursor2 Color Index3 Register 0 **REG[110h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index3 Bit 7	Cursor2 Color Index3 Bit 6	Cursor2 Color Index3 Bit 5	Cursor2 Color Index3 Bit 4	Cursor2 Color Index3 Bit 3	Cursor2 Color Index3 Bit 2	Cursor2 Color Index3 Bit 1	Cursor2 Color Index3 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Color Index3 Register 1 **REG[111h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index3 Bit 15	Cursor2 Color Index3 Bit 14	Cursor2 Color Index3 Bit 13	Cursor2 Color Index3 Bit 12	Cursor2 Color Index3 Bit 11	Cursor2 Color Index3 Bit 10	Cursor2 Color Index3 Bit 9	Cursor2 Color Index3 Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
state								

Cursor2 Color Index3 Register 2 **REG[112h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index3 Bit 23	Cursor2 Color Index3 Bit 22	Cursor2 Color Index3 Bit 21	Cursor2 Color Index3 Bit 20	Cursor2 Color Index3 Bit 19	Cursor2 Color Index3 Bit 18	Cursor2 Color Index3 Bit 17	Cursor2 Color Index3 Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Cursor2 Color Index3 Register 3 **REG[113h]**

Bit	7	6	5	4	3	2	1	0
	Cursor2 Color Index3 Bit 31	Cursor2 Color Index3 Bit 30	Cursor2 Color Index3 Bit 29	Cursor2 Color Index3 Bit 28	Cursor2 Color Index3 Bit 27	Cursor2 Color Index3 Bit 26	Cursor2 Color Index3 Bit 25	Cursor2 Color Index3 Bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[113h] bits 7-0
REG[112h] bits 7-0
REG[111h] bits 7-0
REG[110h] bits 7-0

Cursor2 Color Index3 Bits [31:0]
Each cursor pixel is represented by 2 bits. This register stores the color index for pixel value 11 of Cursor2.

Note
(1) These bits will not effective until the Cursor2 Enable bit is set to 1 (REG[C0h] bit 6=1).

For Hardware Cursors operation, see Section "Hardware Cursor Mode" in datasheet.

2.1.18 Draw2D Mode

Draw 2D Command Register 1 **REG[1D0h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Command Bit 7	Draw 2D Command Bit 6	Draw 2D Command Bit 5	Draw 2D Command Bit 4	Draw 2D Command Bit 3	Draw 2D Command Bit 2	Draw 2D Command Bit 1	Draw 2D Command Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Command Register 2 **REG[1D1h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Command Bit 15	Draw 2D Command Bit 14	Draw 2D Command Bit 13	Draw 2D Command Bit 12	Draw 2D Command Bit 11	Draw 2D Command Bit 10	Draw 2D Command Bit 9	Draw 2D Command Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Command Registers [15:0]

REG[1D1h] Bits 7-4 These bits are rotation extension decoded as below:
Rotation extension is used for bitblt, alpha blending, rop and stretch blt operations.

Bits 7:6 is the x orientation
Bits 5:4 is y orientation
2'b00: original orientation
2'b01: mirror in own axis
2'b10: mirror in opposite axis
2'b11: original orientation using opposite axis

Note
(1) Restriction: when x axis orientation setting is opposite axis, y axis orientation cannot be own axis. Same restriction applies when x axis orientation setting is own axis (mirror or not mirror), y axis orientation cannot be opposite axis.

Some example settings:
0 degree: 4'b0000
horizontal mirror: 4'b0100
vertical mirror: 4'b0001
rotate CW 90: 4'b1110
rotate CCW 90: 4'b1011

REG[1D1h] Bits 3-0 These bits are the command decode:
0x00: nop
0x01: line draw
0x02: rect draw
0x03: ellipse draw
0x04: transparent blt
0x05: bitblt
0x06: clut setup
0x07: alpha blending
0x08: stretchblt
0x09: rop

REG[1D0h] Bits 7:0 In the color expansion mode, these 8 bits specifies the default alpha value.
In the CLUT operation, these 8 bits specifies the color index to be set.
In the alpha blending operation, these 8 bits is the default alpha in 16bpp modes.

Draw 2D Command FIFO Status **REG[1D2h]**

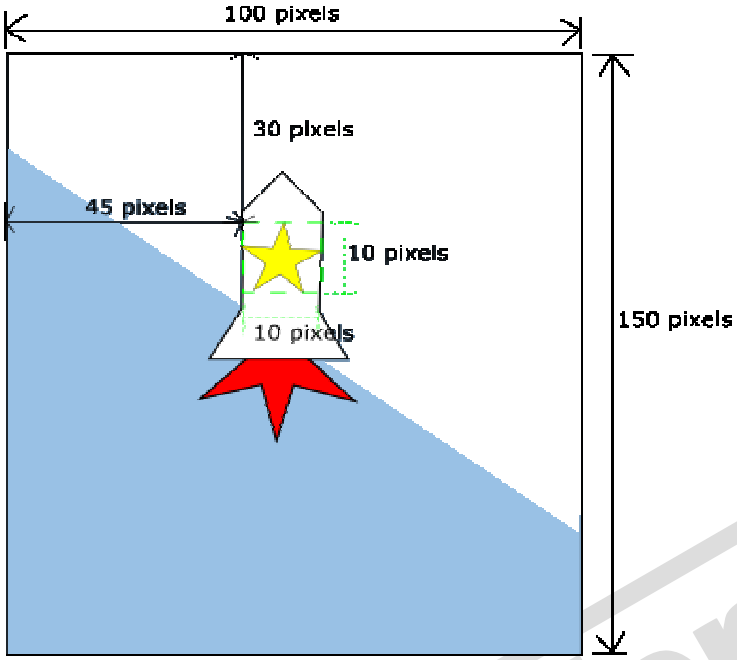
Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw2D AutoMode In	Draw 2D Command start
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 1 **Draw2D AutoMode in**
After setting up the appropriate parameters, set this bit to 1 and then set the draw2d command start bit (REG[1D2h] Bit 0) to 1. The draw2d operation's start timing will switch to automode. In automode, the 2D engine will start only once after DV stream in non display period.
There are some limitations in automode, 1) only a single command is possible, 2) draw2d_brush_win_start will be behave as designed. This is because DV window is double buffered, and we will have to find some way to specify a secondary destination address to 2D engine. We've chosen draw2d_brush_win_start as the secondary destination address pointer.

Note : ROP256 operations are not allowed in automode.

Bit 0 **Draw 2D Command start Bit**
Write this bit = 1, draw2d operation will be started.

Say, this graph is 32bpp, located @ physical location 0x3000. It is 100x150 pixels. Assume the "star" is needed for bitblt, then the source window configuration is:
SRC_WIN_START = 0x3000/(32bpp/8bit)+30*100+45
SRC_WIN_OFFSET = 100
SRC_WIN_WIDTH = 10
SRC_WIN_HEIGHT = 10



Draw 2D Source Window Start Address Register 0 **REG[1D4h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Window Src Start Address Bit 7	Draw 2D Window Src Start Address Bit 6	Draw 2D Window Src Start Address Bit 5	Draw 2D Window Src Start Address Bit 4	Draw 2D Window Src Start Address Bit 3	Draw 2D Window Src Start Address Bit 2	Draw 2D Window Src Start Address Bit 1	Draw 2D Window Src Start Address Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Source Window Start Address Register 1 **REG[1D5h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Window Src Start Address Bit 15	Draw 2D Window Src Start Address Bit 14	Draw 2D Window Src Start Address Bit 13	Draw 2D Window Src Start Address Bit 12	Draw 2D Window Src Start Address Bit 11	Draw 2D Window Src Start Address Bit 10	Draw 2D 1 Window Src Start Address Bit 9	Draw 2D Window Src Start Address Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Source Window Start Address Register 2 **REG[1D6h]**

Bit	7	6	5	4	3	2	1	0

	x	Draw 2D Window Src Start Address Bit 22	Draw 2D Window Src Start Address Bit 21	Draw 2D Window Src Start Address Bit 20	Draw 2D Window Src Start Address Bit 19	Draw 2D Window Src Start Address Bit 18	Draw 2D Window Src Start Address Bit 17	Draw 2D Window Src Start Address Bit 16
Type	RO	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1D6h] bits6-0,
REG[1D5h] bits 7-0,
REG[1D4h] bits 7-0

Draw 2D Source Window Start Address Bits [22:0]
These bits form the 16bit internal start address of draw2d source window in SSD1928 memory with 64bit width (in term of bit per pixel).
E.g. to specify a 1bpp source buffer located physically at ram address 0x0001, the value of this register should be 0x0040.

This is a multi-usage register
In bitBlt, alpha-blend, stretchblt, transparent blt operations, this register is the source window start address. When operations do not require a source window, it will not be interpreted as an address.

In CLUT setup command, bits 15:0 is the {red, green} color of the table being set.

In ellipse draw operation, bits 8:0 is ellipse's center x coordinate and bits 17:9 is ellipse's y clipping boundary.

Draw 2D Source Window Line Address Offset Register 0 **REG[1D8h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Src Window Line Address Offset Bit 7	Draw 2D Src Window Line Address Offset Bit 6	Draw 2D Src Window Line Address Offset Bit 5	Draw 2D Src Window Line Address Offset Bit 4	Draw 2D Src Window Line Address Offset Bit 3	Draw 2D Src Window Line Address Offset Bit 2	Draw 2D Src Window Line Address Offset Bit 1	Draw 2D Src Window Line Address Offset Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Source Window Line Address Offset Register 1 **REG[1D9h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Src Window Line Address Offset Bit 9	Draw 2D Src Window Line Address Offset Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1D9h] bits 1-0,
REG[1D8h] bits 7-0

Draw 2D Source Window Line Address Offset Bits [9:0]
This register is the screen pixel width, (not necessarily same as draw2d window width)

Draw 2D Source Window Color Mode **REG[1DCh**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	SRC Color Bit 2	SRC Color Bit 1	SRC Color Bit 0
Type	RO	RO	RO	RO	RO	RW	RW	RW

Bit	7	6	5	4	3	2	1	0
Reset state	0	0	0	0	0	0	0	0

REG[1DCh] Bits 2-0 **Source Window Color Bits [2:0]**
000 : 16 bpp
001 : 32 bpp
010 : YUV
100 : 1 bpp
101 : 8 bpp

Draw 2D Destination Window Color Mode Register **REG[1DDh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Dest Window Color Mode Bit 1	Draw 2D Dest Window Color Mode Bit 0
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1DDh] Bits 1-0 **Destination Window Color Bits [1:0]**
00 : 16 bpp
01 : 32 bpp
10 : YUV

Draw 2D Source Window Width Register 0 **REG[1E4h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Src Window Width Bit 7	Draw 2D Src Window Width Bit 6	Draw 2D Src Window Width Bit 5	Draw 2D Src Window Width Bit 4	Draw 2D Src Window Width Bit 3	Draw 2D Src Window Width Bit 2	Draw 2D Src Window Width Bit 1	Draw 2D Src Window Width Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Source Window Width Register 1 **REG[1E5h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Src Window Width Bit 9	Draw 2D Src Window Width Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1E5h] bits 1-0, REG[1E4h] bits 7-0 **Draw 2D Source Window Width Register [9:0]**
This is a multi purpose register
In line draw mode, this register is the starting x coordinate of the line drawn

For "blting" operations (alpha blend, stretchblt, bitblt etc), this register is the width of the source window (in pixel). Alpha blend is for RGB mode only. In YUV mode, this value should be a multiple of 2 (This limitation is not applicable in RGB mode).

In ellipse draw mode, this register is the y coordinate of the ellipse's center.

Draw 2D Source Window Height Register 0 **REG[1E8h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Src Window Height Bit 7	Draw 2D Src Window Height Bit 6	Draw 2D Src Window Height Bit 5	Draw 2D Src Window Height Bit 4	Draw 2D Src Window Height Bit 3	Draw 2D Src Window Height Bit 2	Draw 2D Src Window Height Bit 1	Draw 2D Src Window Height Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Source Window Height Register 1 **REG[1E9h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Src Window Height Bit 9	Draw 2D Src Window Height Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1E9h] bits 1-0, REG[1E8h] bits 7-0 **Draw 2D Source Window Height Register [9:0]**
This is a multi purpose register

For "blting" operations (alpha blend, stretchblt, bitblt etc), this register is the height of the source window (in pixel).

In line draw mode, this register is the starting y coordinate

In ellipse draw mode, this register is the "radius" of the ellipse in x axis.

Draw 2D Destination Window Width Register 0 **REG[1ECh]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Dest Window Width Bit 7	Draw 2D Dest Window Width Bit 6	Draw 2D Dest Window Width Bit 5	Draw 2D Dest Window Width Bit 4	Draw 2D Dest Window Width Bit 3	Draw 2D Dest Window Width Bit 2	Draw 2D Dest Window Width Bit 1	Draw 2D Dest Window Width Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Destination Window Width Register 1 **REG[1EDh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Dest Window Width Bit 9	Draw 2D Dest Window Width Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1EDh] bits 1-0, **Draw 2D Destination Window Width Register [9:0]**

REG[1ECh] bits 7-0 **This is a multi purpose register**

For stretchblt operation, this is the width of destination window in pixel. In YUV mode, this value should be a multiple of 2 (This limitation is not applicable in RGB mode).

In line draw mode, this register is the ending x coordinate

In ellipse draw mode, this register is the starting angle of the ellipse (360 < register value < 0)

Draw 2D Destination Window Height Register 0 **REG[1F0h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Dest Window Height Bit 7	Draw 2D Dest Window Height Bit 6	Draw 2D Dest Window Height Bit 5	Draw 2D Dest Window Height Bit 4	Draw 2D Dest Window Height Bit 3	Draw 2D Dest Window Height Bit 2	Draw 2D Dest Window Height Bit 1	Draw 2D Dest Window Height Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Destination Window Y Height Register 1 **REG[1F1h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Dest Window Height Bit 9	Draw 2D Dest Window Height Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1F1h] bits 1-0, REG[1F0h] bits 7-0 **Draw 2D Destination Window Height Register [9:0]**

This is a multi-purpose register

In stretch blit operation, this register is the height of destination window in pixel.

In line draw operation, this register is the ending y coordinate.

In ellipse draw operation, this register is the “radius” of the ellipse in y axis.

Draw 2D Destination Window Start Address Register 0 **REG[1F4h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Dest Window Start Address Bit 7	Draw 2D Dest Window Start Address Bit 6	Draw 2D Dest Window Start Address Bit 5	Draw 2D Dest Window Start Address Bit 4	Draw 2D Dest Window Start Address Bit 3	Draw 2D Dest Window Start Address Bit 2	Draw 2D Dest Window Start Address Bit 1	Draw 2D Dest Window Start Address Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Destination Window Start Address Register 1 **REG[1F5h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Dest Window	Draw 2D Dest Window	Draw 2D Dest Window	Draw 2D Dest Window	Draw 2D Dest Window	Draw 2D Dest Window	Draw 2D Dest Window	Draw 2D Dest Window

Bit	7	6	5	4	3	2	1	0
	Start Address Bit 15	Start Address Bit 14	Start Address Bit 13	Start Address Bit 12	Start Address Bit 11	Start Address Bit 10	Start Address Bit 9	Start Address Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Destination Window Start Address Register 2 **REG[1F6h]**

Bit	7	6	5	4	3	2	1	0
	0	Draw 2D Dest Window Start Address Bit 22	Draw 2D Dest Window Start Address Bit 21	Draw 2D Dest Window Start Address Bit 20	Draw 2D Dest Window Start Address Bit 19	Draw 2D Dest Window Start Address Bit 18	Draw 2D Dest Window Start Address Bit 17	Draw 2D Dest Window Start Address Bit 16
Type	RO	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1F6h] bits 6-0,
REG[1F5h] bits 7-0,
REG[1F4h] bits 7-0

Draw 2D Destination Window Start Address Bits [22:0]
These bits form the 16bit internal start address of draw2d destination window in SSD1928 memory with 64bit width (in term of bit per pixel).
E.g. to specify a 1bpp destination buffer located at physically ram address 0x0001, the value of this register should be 0x0040.

This is a multi-usage register
This register is the destination window start address of all draw2d operations, expect CLUT setup.
In the CLUT setup command, bits 15:0 is the {blue, alpha} color of the table being set.

Draw 2D Destination Window Line Address Offset Register 0 **REG[1f8h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Line Address Offset Bit 7	Draw 2D Line Address Offset Bit 6	Draw 2D Line Address Offset Bit 5	Draw 2D Line Address Offset Bit 4	Draw 2D Line Address Offset Bit 3	Draw 2D Line Address Offset Bit 2	Draw 2D Line Address Offset Bit 1	Draw 2D Line Address Offset Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Destination Window Line Address Offset Register 1 **REG[1f9h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Line Address Offset Bit 9	Draw 2D Line Address Offset Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1D9h] bits 1-0,

Draw Destination Window Line Address Offset Bits [9:0]

REG[1D8h] bits 7-0 This register is the number of pixels between lines inside the display screen size.

Draw 2D Write Value Pattern (Command Parameter) Register 0 **REG[1FCh]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Window Write Pattern Bit 7	Draw 2D Window Write Pattern Bit 6	Draw 2D Window Write Pattern Bit 5	Draw 2D Window Write Pattern Bit 4	Draw 2D Window Write Pattern Bit 3	Draw 2D Window Write Pattern Bit 2	Draw 2D Window Write Pattern Bit 1	Draw 2D Window Write Pattern Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Window Write Pattern (Command Parameter) Register 1 **REG[1FDh]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Window Write Pattern Bit 15	Draw 2D Window Write Pattern Bit 14	Draw 2D Window Write Pattern Bit 13	Draw 2D Window Write Pattern Bit 12	Draw 2D Window Write Pattern Bit 11	Draw 2D Window Write Pattern Bit 10	Draw 2D Window Write Pattern Bit 9	Draw 2D Window Write Pattern Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Window Write Pattern (Command Parameter) Register 2 **REG[1FEh]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Window Write Pattern Bit 23	Draw 2D Window Write Pattern Bit 22	Draw 2D Window Write Pattern Bit 21	Draw 2D Window Write Pattern Bit 20	Draw 2D Window Write Pattern Bit 19	Draw 2D Window Write Pattern Bit 18	Draw 2D Window Write Pattern Bit 17	Draw 2D Window Write Pattern Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1FEh] bits 7-0,
REG[1FDh] bits 7-0,
REG[1FCh] bits 7-0

Draw 2D Window Write Pattern Bits [23:0]
Multi-purpose register
In ROP mode, [7:0] marks the ROP code specified by Microsoft's ROP3 operation.
In transparent blt mode, this register represents the transparent color's BGR value. However, since this register is 24bit in width, when a 16bit color is compared to, the 16bit red color will be shifted left by 3, Green is shifted left by 2, and blue shifted left by 2.

In alpha blending mode, [1:0] is 2 different modes:
2'b00 => Microsoft pre-blend:
dst = (src * const_alpha + dst * (255-const_alpha)) / 255
2'b01 => Microsoft pre-blend2:
if (draw2d_cmd[7:0] == 7'hff)
dst = src + (1 – src.alpha/255) * dst
else // 2 cycle blending
dst = (src + (1 – src.alpha) * dst) * const_alpha
2'b10, 2'b11 => reserved mode

In drawing lines, ellipse, this register is the {R,G,B} color

Draw 2D Brush Window Start Address Register 0 **REG[204h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Brush Window Start Address Bit 7	Draw 2D Brush Window Start Address Bit 6	Draw 2D Brush Window Start Address Bit 5	Draw 2D Brush Window Start Address Bit 4	Draw 2D Brush Window Start Address Bit 3	Draw 2D Brush Window Start Address Bit 2	Draw 2D Brush Window Start Address Bit 1	Draw 2D Brush Window Start Address Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Brush Window Start Address Register 1 **REG[205h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Brush Window Start Address Bit 15	Draw 2D Brush Window Start Address Bit 14	Draw 2D Brush Window Start Address Bit 13	Draw 2D Brush Window Start Address Bit 12	Draw 2D Brush Window Start Address Bit 11	Draw 2D Brush Window Start Address Bit 10	Draw 2D Brush Window Start Address Bit 9	Draw 2D Brush Window Start Address Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Brush Window Start Address Register 2 **REG[206h]**

Bit	7	6	5	4	3	2	1	0
	0	Draw 2D Brush Window Start Address Bit 22	Draw 2D Brush Window Start Address Bit 21	Draw 2D Brush Window Start Address Bit 20	Draw 2D Brush Window Start Address Bit 19	Draw 2D Brush Window Start Address Bit 18	Draw 2D Brush Window Start Address Bit 17	Draw 2D Brush Window Start Address Bit 16
Type	RO	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

6REG[206h] bits 6-0,
REG[205h] bits 7-0,
REG[204h] bits 7-0

Draw 2D Brush Window Start Address Bits [22:0]
These bits form the 16bit internal start address of draw2d brush window in SSD1928 memory for ROP operation (in term of bit per pixel).
E.g. to specify a 1bpp destination buffer located at physically ram address 0x0001, the value of this register should be 0x0040.

Draw 2D Brush Window Line Address Offset Register 0 **REG[208h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Brush Line Address Offset Bit 7	Draw 2D Brush Line Address Offset Bit 6	Draw 2D Brush Line Address Offset Bit 5	Draw 2D Brush Line Address Offset Bit 4	Draw 2D Brush Line Address Offset Bit 3	Draw 2D Brush Line Address Offset Bit 2	Draw 2D Brush Line Address Offset Bit 1	Draw 2D Brush Line Address Offset Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D brush Window Line Address Offset Register 1 **REG[209h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Brush Line Address Offset Bit 9	Draw 2D Brush Line Address Offset Bit 8

Bit	7	6	5	4	3	2	1	0
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[209h] bits 1-0, REG[208h] bits 7-0 **Draw Brush Window Line Address Offset Bits [9:0]**
This register is the width of the pattern in memory in terms of number of pixels

Draw 2D Brush Window width Register 0 **REG[214h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Brush Window Width Bit 7	Draw 2D Brush Window Width Bit 6	Draw 2D Brush Window Width Bit 5	Draw 2D Brush Window Width Bit 4	Draw 2D Brush Window Width Bit 3	Draw 2D Brush Window Width Bit 2	Draw 2D Brush Window Width Bit 1	Draw 2D Brush Window Width Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D Brush Window Width Register 1 **REG[215h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Brush Window Width Bit 9	Draw 2D Brush Window Width Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[215h] bits 1-0, REG[214h] bits 7-0 **Draw 2D Brush Window Width[9:0]**
This register is the operation region width. Note that "operation width" is not necessarily equal to Brush Window line offset (i.e. line offset is related to the brush's layout in memory, while one might choose a smaller part of the brush to operate on)

Draw 2D Brush Window Height Register 0 **REG[[218h]**

Bit	7	6	5	4	3	2	1	0
	Draw 2D Brush Window Height Bit 7	Draw 2D Brush Window Height Bit 6	Draw 2D Brush Window Height Bit 5	Draw 2D Brush Window Height Bit 4	Draw 2D Brush Window Height Bit 3	Draw 2D Brush Window Height Bit 2	Draw 2D Brush Window Height Bit 1	Draw 2D Brush Window Height Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Draw 2D brush Window Y Height Register 1 **REG[219h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Draw 2D Brush Window Height Bit 9	Draw 2D Brush Window Height Bit 8
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
state								

REG[219h] bits 1-0, REG[218h] bits 7-0 **Draw 2D Brush Window Height Register [9:0]**
This register is Draw2D Brush Window height.

Draw 2D Command FIFO Interrupt enable **REG[21Ch]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Reserved	Draw 2D Command FIFO Interrupt Enable
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 1 **Reserved bit**
This bit should be programmed as 0.

Bit 0 **Draw 2D Command FIFI Interrupt Enable**
When this bit = 1 : Enable Command FIFO Ready.
When this bit = 0 : Enable Command FIFO Busy.

Draw 2D Command FIFO Interrupt Status **REG[21Eh]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Reserved	Draw 2D cmd fifo status flag
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 1 **Reserved bit**

Bit 0 **Draw 2D Cmd FIFO Status Flag**
When this bit = 1: Cmd FIFO ready. Write 1 to clear the flag .Write 0 has not hardware effect
When this bit = 0 : Busy

Raw Draw 2D Command FIFO FLAG **REG[220h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Raw Draw2D Command Status flag
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	1

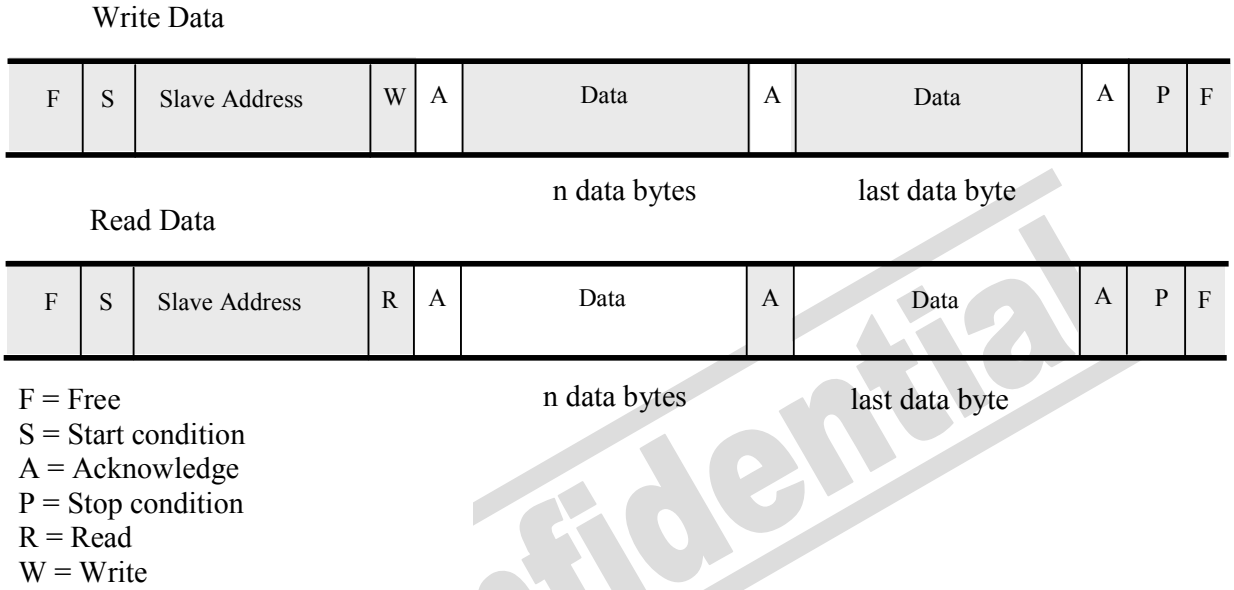
Bit 1 **Reserved bit**

Bit 0 **RAW Draw 2D CMD status flag**
When this bit = 1 : Ready
When this bit = 0 : Busy

2.1.19 I2C Control Registers

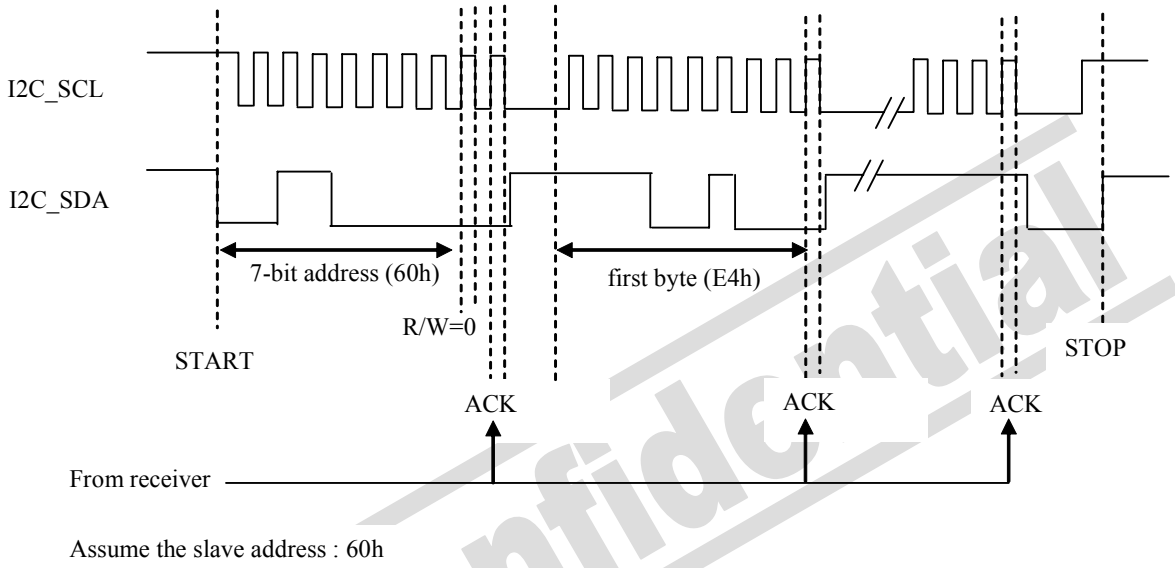
Each I2C device has a unique 7-bit I2C address so that the master knows specifically whom they are communicating with. This address is sent in the 1st byte and the least significant bit of this initial byte indicates if the master is going to send (write) or receive (read) data from the receiver, called the slave device. Each transmission sequence must begin with the Start condition and end with the Stop or ReStart condition.

Figure 2-35: Data format for I2C interface



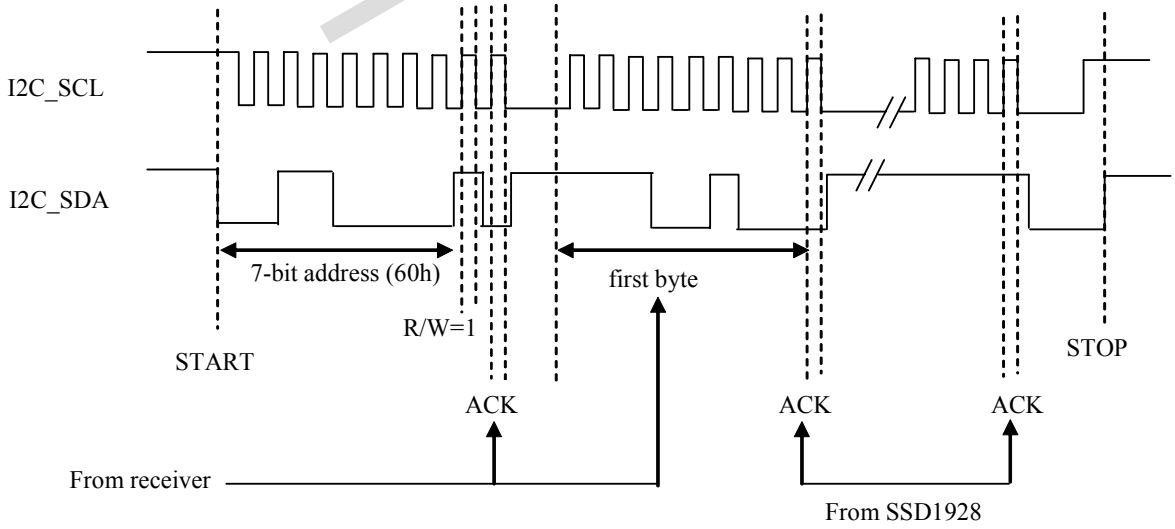
F (FREE) - the bus is free or idle; SSD1928 will release the data line I2C_SDA and the I2C_SCL clock and causes both in the high state.
S (START) or R (RESTART) - data transfer begins with a Start condition. The level of the I2C_SDA data line changes from high to low, while the I2C_SCL clock line remains high. When this occurs, the bus becomes 'busy'.
D (DATA) - a high or low bit of information on the I2C_SDA data line is valid during the high level of the I2C_SCL clock line. This level will be kept stable during the entire time that the clock remains high to avoid misinterpretation as a Start or Stop condition.
P (STOP) - data transfer is terminated by a Stop condition. This occurs when the level on the I2C_SDA data line passes from the low state to the high state, while the I2C_SCL clock line remains high. When the data transfer has been terminated, the bus is free once again.
A (Acknowledge) - The 9th clock pulse is the acknowledge bit. The bit must be reset (to logic 0) when the I2C-bus controller is operating in master/receiver mode and requires no further data to be sent from the slave transmitter. This causes a negative acknowledge on the I2C-bus, which halts further transmission from the slave device.

Figure 2-36: Bus diagram to write data to receiver



Assume the slave address : 60h

Figure 2-37: Bus diagram to read data from receiver



Assume the slave address : 60h

I2C Data Out registers **REG[230h]**

Bit	7	6	5	4	3	2	1	0
	I2C Data Bit 7	I2C Data Bit 6	I2C Data Bit 5	I2C Data Bit 4	I2C Data Bit 3	I2C Data Bit 2	I2C Data Bit 1	I2C Data Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
state								

Bits 7-0 **I2C Data Out Registers [7:0]**
This register drive data to I2C Slave.

I2C R/W Control registers **REG[231h]**

Bit	7	6	5	4	3	2	1	0
	I2C Start	0	0	I2C Stop	0	0	I2C CMD Bit 1	I2C CMD bit 0
Type	RW	RO	RO	RW	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **I2C Start bit**
1 : Send out the Start condition
Bit 4 **I2C Stop bit**
1 : Send out the Stop condition
Bits 1-0 **I2C Command Bits [1:0]**
00 : Idle
01 : Write
10 : Read and acknowledgement
11 : Read and without acknowledgement

I2C R/W Control registers **REG[232h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	I2C bus clear	I2C Enable out
Type	RO	RO	RO	RO	RO	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 1 **I2C Bus Clear bit**
1 : Clear
0 : Normal
Bit 0 **I2C Enable Out bit**
1 : Enable
0 : Disable

I2C Baud Rate registers **REG[233h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	I2C Baud Rate Bit 4	I2C Baud Rate Bit 3	I2C Baud Rate Bit 2	I2C Baud Rate Bit 1	I2C Baud Rate Bit 0
Type	RO	RO	RO	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 4-0, **I2C Baud Rate Registers [4:0]**
00 : Slowest
1F : Fastest
I2C CLK period = $[2^9 - (\text{baud rate}) \times 2^4] \times \text{MCLK period} \times 2$

I2C Status registers **REG[234h]**

Bit	7	6	5	4	3	2	1	0

Bit	7	6	5	4	3	2	1	0
	I2C Module Ready	I2C Slave No Ack	I2C Bus Busy	I2C Arbitration Lost	0	0	0	0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 7 **I2C Module Ready**
1 : Ready
0 Busy

Bit 6 **I2C Slave No Acknowledge**
1 : No acknowledge
0 : With acknowledge

Bit 5 **I2C Bus Busy**
1 : Busy
0 : Idle

Bit 4 **I2C Arbitration lost**
1 : Arbitration lost
0 : normal

I2C Interrupt Status registers **REG[235h]**

Bit	7	6	5	4	3	2	1	0
	0	0	I2C bus busy interrupt enable	I2C Module Ready Interrupt enable	0	0	I2C bus busy Flag	I2C Module Ready Flag
Type	RO	RO	RO	RO	RO	RO	RW	R/W
Reset state	0	0	0	0	0	0	0	0

Bit 5 **I2C Bus Busy Interrupt Enable Bit**
1 : enable interrupt
0 : disable interrupt

Bit 4 **I2C Module Ready Interrupt Enable Bit.**
1 : enable interrupt
0 : disable interrupt

Bit 1 **I2C BUS BUSY flag,**
Write 1 to this bit clear the interrupt flag.

Bit 0 **I2C Module Ready flag,**
Write 1 to this bit clear the interrupt flag.

I2C Data In ready registers **REG[236h]**

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	I2C Data in Ready
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 0 **I2C Data In Ready Flag**
This bit indicates I2C Data in is ready.

I2C Data in registers **REG[237h]**

Bit	7	6	5	4	3	2	1	0
	I2C Data In Bit 7	I2C Data In Bit 6	I2C Data In Bit 5	I2C Data In Bit 4	I2C Data In Bit 3	I2C Data In Bit 2	I2C Data In Bit 1	I2C Data In Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bits 7-0 **I2C Data in Registers [7:0]**
This register latch up data from I2C Slave.

2.1.20 JPEG Registers

2.1.20.1 Encode Procedure

1) Enable the JPEG module (REG[380h] bit 0 = 1).
2) Initialize the JPEG codec registers.
   a) Software reset the JPEG codec (REG[402h] bit 7 = 1).
   b) Set operation mode to encode (REG[400h] bit 2).
   c) Set JPEG MJPEG Mode (REG[400h] bit 5).
   d) Set YUV format (REG[400h] bit 1-0).
   e) Set Marker Insert Enable (REG[400h] bit 3).
   f) Generate JPEG Header:
      i) Set JPEG markers (REG[468h] – REG[4F8h], REG[9B0h] – REG[9ECh]).
      ii) Set DRI Setting (REG[9F8h] – REG[9FAh]).
      iii) Set Vertical Pixel Size (REG[9F0h] – REG[9F2h]).
      iv) Set Horizontal Pixel Size (REG[9F4h] – REG[9F6h]).
      v) Set Insertion Marker Data, if any (REG[420h] – REG[466h]).
      vi) Set Quantization Table No. 0 (REG[500h] – REG[57Eh]) & Quantization Table No. 1 (REG[580h] – REG[5FEh]) according to the following zigzag sequence:

1	2	6	7	15	16	28	30
3	5	8	14	17	27	31	44
4	9	13	18	26	32	43	45
10	12	19	25	33	42	46	55
11	20	24	34	41	47	54	56
21	23	35	40	48	53	57	62
22	36	39	49	52	58	61	63

37	38	50	51	59	60	64	65

vii) Set Huffman Table according to the following sequence:

(1) DC Huffman Table No. 0 Register 0 (REG[600h] – REG[61Eh]).

(2) DC Huffman Table No. 0 Register 1 (REG[620h] – REG[636h]).

(3) AC Huffman Table No. 0 Register 0 (REG[640h] – REG[65Eh]).

(4) AC Huffman Table No. 0 Register 1 (REG[660h] – REG[7A2h]).

(5) DC Huffman Table No. 1 Register 0 (REG[800h] – REG[81Eh]).

(6) DC Huffman Table No. 1 Register 1 (REG[820h] – REG[836h]).

(7) AC Huffman Table No. 1 Register 0 (REG[840h] – REG[85Eh]).

(8) AC Huffman Table No. 1 Register 1 (REG[860h] – REG[9A2h]).

Note

(1) The default Huffman tables for luminous and chrominance color components is specified in ISO/IEC 10918 attachment K.

3) Initialize the JPEG module registers (i.e. JPEG Line Buffer and FIFO).

a) Set the JPEG Source Start Address (REG[410h] – REG[412h]).

b) Set the JPEG Destination Address (REG[414h] – REG[416h]).

c) Set the JPEG FIFO Size (REG[3A4h]).

d) Set the Encode Size Limit (REG[3B0h] – REG[3B2h]).

e) Set JPEG FIFO Threshold Trigger (REG[3A0h] bit 5-4).

f) Set JPEG YUV Output Data Range Select (REG[380h] bit 4).

4) Initialize the DV Interface.

a) Set JPEG Still Picture Enable (REG[160h] bit 3).

b) Set JPEG Line Buffer Horizontal (REG[164h] – REG[165h]).

c) Set JPEG Vertical Size (REG[168h] – REG[169h]).

d) Set JPEG Horizontal Decimation Ratio (REG[172h]).

e) Set JPEG Vertical Decimation Ratio (REG[173h]).

Note

(1) The size of the image to be encoded must comply with the following specification:

YUV format	YUV420	YUV422	YUV411	YUV444
Horizontal Pixel Size	Multiples of 16	Multiples of 16	Multiples of 32	Multiples of 8
Vertical Pixel Size	Multiples of 16	Multiples of 8	Multiples of 8	Multiples of 8

5) Start the encode process:

a) Clear all status of JPEG Line Buffer and FIFO (REG[382h] = 00h & REG[383h] = 00h).

b) Enable required interrupts (REG[386h] – REG[387h]).

c) Start JPEG operation (REG[402h] bit 0 = 1).

d) Start capturing (REG[38Ah] bit 0 = 1).

Note: Capturing is not start until the JPEG codec has generated the markers.

6) Use the JPEG FIFO Threshold Trigger Flag (REG[382h] bit 2) or poll the JPEG FIFO Threshold Status (REG[382h] bit 5-4) to check if JPEG FIFO Threshold condition is met.
7) Read data from JPEG FIFO (REG[414h] – REG[416h])
8) Repeat steps 1 to 3 until FIFO Valid Data Size is 0 and JPEG Operation Status is idle (REG[404h] bit 0 = 0).
9) Verify the actual file size with the Encode Size Result (REG[3B4h] – REG[3B6h]).
10) Confirm the encode process is complete with the JPEG Codec Interrupt Flag (REG[383h] bit 1).

2.1.20.2 Decode Procedure

1) Enable the JPEG Codec (REG[380h] bit 0 = 1).
2) Initialize the JPEG Codec registers.
   a) Software reset the JPEG Codec (REG[402h] bit 7 = 1).
   b) Set operation mode to decode (REG[400h] bit 2).
   c) Set JPEG MJPEG Mode (REG[400h] bit 5).
   d) Set the RST Marker Operation Setting (REG[41Ch] bit 1-0).
3) Initialize the JPEG module registers (i.e. JPEG Line Buffer and FIFO).
   a) Set the JPEG Source Start Address (REG[410h] – REG[413h]).
   b) Set the JPEG Destination Address (REG[414h] – REG[417h]).
   c) Set the JPEG FIFO Size (REG[3A4h]).
   d) Set the JPEG File Size (REG[3B8h] – REG[3BAh]).
   e) Set JPEG YUV Output Data Range Select (REG[380h] bit 4).
4) Start the decode process:
   a) Clear all status of JPEG Line Buffer and FIFO (REG[382h] – REG[383h] = 0000h).
   b) Enable required interrupts (REG[386h] – REG[387h]).
   c) Start JPEG operation (REG[402h] bit 0 = 1).
   d) Start decoding (REG[38Ah] bit 0 = 1).
5) Write data to JPEG FIFO (REG[414h – REG[416h]]
6) Wait for FIFO condition is met (i.e Empty/Half-Full/Quad-Full) by interrupt or polling. If Decode Marker Read Interrupt is detected, read the Vertical Pixel Size (REG[3DCh] – REG[3DDh]) and Horizontal Pixel Size (REG[3D8h] – REG[3D9h]) and set the registers for display.
7) Repeat steps 5 and 6 until the end of file.
8) Wait for JPEG Decode Complete Flag (REG[383h] bit 5) .

9) Verify the decode process is complete with the JPEG Operation Status (REG[404h] bit 0 = 0).

Note: 64-bit unit data is required to write to the JPEG FIFO. Pad the end of the JPEG data stream with 8'h00s to complete the last 64-bit data for the last FIFO entry is necessary.

Followings description for Error Handling:

If JPEG Codec Interrupt Flag is high and both the Decode Marker Read Interrupt and JPEG Decode Complete Interrupt are low, the JPEG codec encountered error in the JPEG data stream and cannot finish decoding. Check the JPEG Operation Status and JPEG Error Status for debug.

If RST Marker Operation is set to enable the data revise function, JPEG codec may finish decoding even if the JPEG data stream is corrupted. In this case, the decode process is complete according to steps 1 and 9 but a corrupted JPEG image will be displayed. Check the RST Revise Code to confirm if a revise operation is done and the type of error is indicated by the JPEG Error Status.

JPEG Control Register **REG[380h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	YUV Output Data Range Select	Reserved	Reserved	Reserved	JPEG Module Enable
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-5 **Reserved bits**
These bits should be programmed by 0.

Bit 4 **YUV Output Data Range Select**
The YUV output range depends on the camera interface data range (Encode mode) or the display data range (Decode mode).

Table 2-35: YUV Output Range Selection

REG[0380h] bit 4	YUV Output Data Range
0	0 =< Y =< 255 -128 =< U =< 127 -128 =< V =< 127
1	0 =< Y =< 255 0 =< U =< 255 0 =< V =< 255

Bits 3-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **JPEG Module Enable**
This bit enables/disables the JPEG module and its associated registers.
When this bit = 1, the JPEG module is enabled and a clock source is supplied.
When this bit = 0, the JPEG module is disabled and the clock source is disabled.

JPEG Status Flag Register 0 **REG[382h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Encode Overflow Violation Flag	JPEG Codec File Out Status	JPEG FIFO Threshold Status Bit 1	JPEG FIFO Threshold Status Bit 0	JPEG Encode Size Limit Violation Flag	JPEG FIFO Threshold Trigger Flag	JPEG FIFO Full Flag	JPEG FIFO Empty Flag

Bit	7	6	5	4	3	2	1	0
Type	RW	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **JPEG Encode Overflow Violation Flag**

This bit indicates Encode overflow status.
For Reads:
When this bit = 1, an encode overflow violation has occurred.
When this bit = 0, no violation has occurred.
For Writes:
When a 1 is written to this bit, the JPEG Encode Overflow Violation Flag is cleared.
When a 0 is written to this bit, there is no hardware effect.

Bit 6 **JPEG Codec File Out Status**
This bit indicates the status of the JPEG Codec output.
When this bit = 1, the JPEG Codec is encoding or outputing encoded data.
When this bit = 0, the JPEG Codec is not outputing encoded data.

Bit 5-4 **JPEG FIFO Threshold Status bits [1:0]**
These bits indicate how much data is currently in the JPEG FIFO. See the JPEG FIFO Size register (REG[03A4h]) for information on setting the JPEG FIFO size.

Table 2-36: JPEG FIFO Threshold Status

REG[0382h] bits 5-4	JPEG FIFO Threshold Status
00	no data (same as empty)
01	more than 4 bytes of data exist
10	more than 1/4 of specified FIFO size data exists
11	more than 1/2 of specified FIFO size data exists

Bit 3 **JPEG Encode Size Limit Violation Flag**
This flag is asserted when the JPEG compressed data size is over the encode size limit as specified in the Encode Size Limit registers (REG[03B0h], REG[03B2h]). This flag is masked by the JPEG Encode Size Limit Violation Interrupt Enable bit and is only available when REG[0386h] bit 3 = 1.
For Reads:
When this bit = 1, an encode size limit violation has occurred.
When this bit = 0, no violation has occurred.
For Writes:
When a 1 is written to this bit after disabling the interrupt, this bit is cleared.
When a 0 is written to this bit, there is no hardware effect.

Bit 2 **JPEG FIFO Threshold Trigger Flag**
This flag is asserted when the amount of data in the JPEG FIFO meets the condition specified by the JPEG FIFO Trigger Threshold bits (REG[03A0h] bits 5-4). This flag is masked by the JPEG FIFO Threshold Trigger Interrupt Enable bit and is only available when REG[0386h] bit 2 = 1.
For Reads:
When this bit = 1, the amount of data in the JPEG FIFO has reached the JPEG FIFO Trigger Threshold.
When this bit = 0, the amount of data in the JPEG FIFO is less than the JPEG FIFO Trigger Threshold.
For Writes:
When a 1 is written to this bit, the FIFO Threshold Trigger Flag is cleared.
When a 0 is written to this bit, there is no hardware effect.

Bit 1 **JPEG FIFO Full Flag**
This flag is asserted when the JPEG FIFO is full. This flag is masked by the JPEG FIFO Full Interrupt Enable bit and is only available when REG[0386h] bit 1 = 1.
For Reads:
When this bit = 1, the JPEG FIFO is full.
When this bit = 0, the JPEG FIFO is not full.

For Writes:
When a 1 is written to this bit, the JPEG FIFO Full Flag is cleared.
When a 0 is written to this bit, there is no hardware effect.

Bit 0 **JPEG FIFO Empty Flag**
This flag is asserted when the JPEG FIFO is empty. This flag is masked by the JPEG FIFO Empty Interrupt Enable bit and is only available when REG[0386h] bit 0 = 1.
For Reads:
When this bit = 1, the JPEG FIFO is empty.
When this bit = 0, the JPEG FIFO is not empty.
For Writes:
When a 1 is written to this bit, the JPEG FIFO Empty Flag is cleared.
When a 0 is written to this bit, there is no hardware effect.

JPEG Status Flag Register 1 **REG[383h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	JPEG Decode Complete Flag	JPEG Decode Marker Read Flag	Reserved	JPEG Line Buffer Overflow Flag	JPEG Codec Interrupt Flag	Reserved
Type	RW	RW	RW	RW	RW	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-6 **Reserved bits**
These bits should be programmed by 0.

Bit 5 **JPEG Decode Complete Flag**
This flag is asserted when the JPEG decode operation is finished. This flag is masked by the JPEG Decode Complete Interrupt Enable bit and is only available when REG[0387h] bit 5 = 1.
For Reads:
When this bit = 1, the JPEG decode operation is finished.
When this bit = 0, the JPEG decode operation is not finished yet.
For Writes:
When a 1 is written to this bit after disabling the interrupt, this bit is cleared.
When a 0 is written to this bit, there is no hardware effect.

Bit 4 **JPEG Decode Marker Read Flag**
This flag is asserted during the JPEG decoding process when decoded marker information is read from the JPEG file. This flag is masked by the JPEG Decode Marker Read Interrupt Enable bit and is only available when REG[0387h] bit 4 = 1.
For Reads:
When this bit = 1, a JPEG decode marker has been read.
When this bit = 0, a JPEG decode marker has not been read.
For Writes:
When a 1 is written to this bit after disabling the interrupt, this bit is cleared.
When a 0 is written to this bit, there is no hardware effect.

Bit 3 **Reserved bit**
This bit should be programmed by 0.

Bit 2 **JPEG Line Buffer Overflow Flag**
This flag is asserted when a JPEG Line Buffer overflow occurs. This flag is masked by the JPEG Line Buffer Overflow Interrupt Enable bit and is only available when REG[0387h] bit 2 = 1.
When this bit = 1, a JPEG Line Buffer overflow has occurred.
When this bit = 0, a JPEG Line Buffer overflow has not occurred.
To clear this flag, perform a JPEG Codec Software Reset (REG[402h] bit 7 = 1).

Bit 1 **JPEG Codec Interrupt Flag**
This flag is asserted when the JPEG codec generates an interrupt. This flag is masked by the JPEG Codec Interrupt Enable bit and is only available when REG[0387h] bit 1 = 1).
When this bit = 1, the JPEG codec has generated an interrupt.
When this bit = 0, the JPEG codec has not generated an interrupt.

To clear this flag, perform a JPEG Codec Software Reset (REG[402h] bit 7 = 1).

Bit 0 **Reserved bit**
This bit should be programmed by 0.

JPEG RAW Status Flag Register 0 **REG[384h]**

Bit	7	6	5	4	3	2	1	0
	Raw JPEG Encode Overflow Violation Flag	JPEG Codec File Out Status	JPEG FIFO Threshold Status Bit 1	JPEG FIFO Threshold Status Bit 0	Raw JPEG Encode Size Limit Violation Flag	Raw JPEG FIFO Threshold Trigger Flag	Raw JPEG FIFO Full Flag	Raw JPEG FIFO Empty Flag
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 7 **Raw JPEG Encode Overflow Violation Flag**
This bit indicates Encode overflow status.
When this bit = 1, JPEG Codec overwrite unread encode data.
When this bit = 0, JPEG Codec does not overwrite unread encode data.
To clear this flag, write a 1 to the **JPEG Encode Overflow Violation Flag (REG[382h] bit 7 = 1).**

Bit 6 **JPEG Codec File Out Status**
This bit provides the status of the JPEG Codec output.
When this bit = 1, the JPEG Codec is encoding or outputing encoded data.
When this bit = 0, the JPEG Codec is not outputing encoded data.

Bits 5-4 **JPEG FIFO Threshold Status bits [1:0]**
These bits indicate how much data is currently in the JPEG FIFO. See the JPEG FIFO Size Register (REG[3A4h] for information on setting the JPEG FIFO Size.

Table 2-37: JPEG FIFO Threshold Status

REG[0384h] bits 5-4	JPEG FIFO Threshold Status
00	no data (same as empty)
01	more than 4 bytes of data exist
10	more than 1/4 of specified FIFO size data exists
11	more than 1/2 of specified FIFO size data exists

Bit 3 **Raw JPEG Encode Size Limit Violation Flag**
This flag is asserted when the JPEG encoded data size is over the size limit as specified in the Encode Size Limit registers (REG[3B0h] - REG[3B2h]). This flag is not affected by the JPEG Encode Size Limit Violation Interrupt Enable bit (REG[386h] bit 3).
When this bit = 1, an encode size limit violation has occurred.
When this bit = 0, no violation has occurred.
To clear this flag, write a 1 to the JPEG Encode Size Limit Violation Flag (REG[382h] bit 3 = 1).

Bit 2 **Raw JPEG FIFO Threshold Trigger Flag**
This flag is asserted when the amount of data in the JPEG FIFO meets the condition specified by the JPEG FIFO Trigger Threshold bits (REG[3A0] bits 5-4). This flag is not affected by the JPEG FIFO Threshold Trigger Interrupt Enable bit (REG[386h] bit 2).
When this bit = 1, the amount of data in the JPEG FIFO has reached the JPEG FIFO Trigger Threshold.
When this bit = 0, the amount of data in the JPEG FIFO is less than the JPEG FIFO Trigger Threshold.
To clear this flag, write a 1 to the JPEG FIFO Threshold Trigger Flag (REG[382h] bit 2 = 1).

Bit 1 **Raw JPEG FIFO Full Flag**
This flag is asserted when the JPEG FIFO is full. This flag is not affected by the JPEG FIFO Full Interrupt Enable bit (REG[386h] bit 1).
When this bit = 1, the JPEG FIFO is full.
When this bit = 0, the JPEG FIFO is not full.
To clear this flag, write a 1 to the JPEG FIFO Full Flag (REG[382h] bit 1 = 1).

Bit 0 **Raw JPEG FIFO Empty Flag**
This flag is asserted when the JPEG FIFO is empty. This flag is not affected by the JPEG FIFO Empty Interrupt Enable bit (REG[386h] bit 0).
When this bit = 1, the JPEG FIFO is empty.
When this bit = 0, the JPEG FIFO is not empty.
To clear this flag, write a 1 to the JPEG FIFO Empty Flag (REG[382h] bit 0 = 1).

JPEG RAW Status Flag Register 1 **REG[385h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Raw JPEG Decode Complete Flag	Raw JPEG Decode Marker Read Flag	Reserved	Raw JPEG Line Buffer Overflow Flag	Raw JPEG Codec Interrupt Flag	Reserved
Type	RW	RW	RO	RO	RW	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-6 **Reserved bits**
These bits should be programmed by 0.

Bit 5 **Raw JPEG Decode Complete Flag**
This flag is asserted when the JPEG decode operation is finished and ready to display.
This flag is not affected by the JPEG Decode Complete Interrupt Enable bit (REG[387h] bit 5).
When this bit = 1, the JPEG decode operation is finished.
When this bit = 0, the JPEG decode operation is not finished yet.
To clear this flag, write a 1 to the JPEG Decode Complete Flag (REG[383h] bit 5 = 1).

Bit 4 **Raw JPEG Decode Marker Read Flag**
This flag is asserted during the JPEG decoding process when decoded marker information is read from the JPEG file.
When this bit = 1, a JPEG decode marker has been read.
When this bit = 0, a JPEG decode marker has not been read.
To clear this flag, write a 1 to the JPEG Decode Marker Read Flag (REG[383h] bit 4 = 1).

Bit 3 **Reserved bit**
This bit should be programmed by 0.

Bit 2 **Raw JPEG Line Buffer Overflow Flag**
This flag is asserted when a JPEG Line Buffer overflow occurs. This flag is not affected by the JPEG Line Buffer Overflow Interrupt Enable (REG[387h] bit 2).
When this bit = 1, a JPEG Line Buffer overflow has occurred.
When this bit = 0, a JPEG Line Buffer overflow has not occurred.
To clear this flag, perform a JPEG Codec Software Reset (REG[402h] bit 7 = 1).

Bit 1 **Raw JPEG Codec Interrupt Flag**
This flag is asserted when an interrupt is generated by the JPEG codec. This flag is not affected by the JPEG Codec Interrupt Enable bit (REG[387h] bit 1).
When this bit = 1, the JPEG codec has generated an interrupt.
When this bit = 0, no interrupt has been generated.
To clear this flag, perform a JPEG Codec Software Reset (REG[402h] bit 7 = 1).

Bit 0 **Reserved bit**
This bit should be programmed by 0.

JPEG Interrupt Control Register 0 **REG[386h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Encode Overflow Violation Interrupt Enable	Reserved	Reserved	Reserved	JPEG Encode Size Limit Violation Interrupt Enable	JPEG FIFO Threshold Trigger Interrupt Enable	JPEG FIFO Full Interrupt Enable	JPEG FIFO Empty Interrupt Enable
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7 **JPEG Encode Overflow Violation Interrupt Enable**
This bit controls the encode overflow violation interrupt. The status of this interrupt can be determined using the JPEG Encode Overflow Violation Flag bit (REG[382h] bit 7).
When this bit = 1, the interrupt is enabled.
When this bit = 0, the interrupt is disabled.

Bits 6-4 **Reserved bits**
These bits should be programmed by 0.

Bit 3 **JPEG Encode Size Limit Violation Interrupt Enable**
This bit controls the encode size limit violation interrupt. The status of this interrupt can be determined using the JPEG Encode Size Limit Violation Flag bit (REG[382h] bit 3).
When this bit = 1, the interrupt is enabled.
When this bit = 0, the interrupt is disabled.

Bit 2 **JPEG FIFO Threshold Trigger Interrupt Enable**
This bit controls the JPEG FIFO threshold trigger interrupt. The status of this interrupt can be determined using the JPEG FIFO Threshold Trigger Flag bit (REG[382h] bit 2).
When this bit = 1, the interrupt is enabled.
When this bit = 0, the interrupt is disabled.

Bit 1 **JPEG FIFO Full Interrupt Enable**
This bit controls the JPEG FIFO full interrupt. The status of this interrupt can be determined using the JPEG FIFO Full Flag bit (REG[382h] bit 1).
When this bit = 1, the interrupt is enabled.
When this bit = 0, the interrupt is disabled.

Bit 0 **JPEG FIFO Empty Interrupt Enable**
This bit controls the JPEG FIFO empty interrupt. The status of this interrupt can be determined using the JPEG FIFO Empty Flag bit (REG[382h] bit 0).
When this bit = 1, the interrupt is enabled.
When this bit = 0, the interrupt is disabled.

JPEG Interrupt Control Register 1 **REG[387h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	JPEG Decode Complete Interrupt Enable	JPEG Decode Marker Read Interrupt Enable	Reserved	JPEG Line Buffer Overflow Interrupt Enable	JPEG Codec Interrupt Enable	Reserved
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-6 **Reserved bits**
These bits should be programmed by 0.

Bit 5 **JPEG Decode Complete Interrupt Enable**
This bit controls the JPEG decode complete interrupt. The status of this interrupt can be determined using the JPEG Decode Complete Flag bit (REG[383h] bit 5).
When this bit = 1, the interrupt is enabled.
When this bit = 0, the interrupt is disabled.

Bit 4 **JPEG Decode Marker Read Interrupt Enable**
This bit controls the JPEG decode marker read interrupt. The status of this interrupt can be determined using the JPEG Decode Marker Read Flag (REG[383h] bit 4).
When this bit = 1, the interrupt is enabled.
When this bit = 0, the interrupt is disabled.

Bit 3 **Reserved bit**
This bit should be programmed by 0.

Bit 2 **JPEG Line Buffer Overflow Interrupt Enable**
This bit controls the JPEG Line Buffer overflow interrupt. The status of this interrupt can be determined using the JPEG Line Buffer Overflow Flag (REG[383h] bit 2).
When this bit = 1, the interrupt is enabled.
When this bit = 0, the interrupt is disabled.

Bit 1 **JPEG Codec Interrupt Enable**
This bit controls the JPEG codec interrupt. The status of this interrupt can be determined using the JPEG Codec Interrupt Flag (REG[383h] bit 1).
When this bit = 1, the interrupt is enabled.
When this bit = 0, the interrupt is disabled.

Bit 0 **Reserved bit**
This bit should be programmed by 0.

JPEG Code Start / Stop Control Register) **REG[38Ah]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	JPEG Start/Stop Control
Type	RW	RW	RW	RW	RW	RW	RW	WO
Reset state	0	0	0	0	0	0	0	0

Bits 7-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **JPEG Start/Stop Control**
This bit controls the JPEG codec for both JPEG encode mode and JPEG decode mode.
For JPEG Encode:
When this bit is set to 1, the JPEG codec starts capturing the next frame and then stops.
When this bit is set to 0, the JPEG codec will be ready to capture from the next frame.
For JPEG Decode:
When this bit is set to 1, the JPEG codec starts to decode the image.
When this bit is set to 0, the JPEG codec will be ready to read the JPEG markers.

JPEG FIFO Control Register **REG[3A0h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	JPEG FIFO Trigger Threshold Bit 1	JPEG FIFO Trigger Threshold Bit 0	Reserved	Reserved	JPEG Output select	Reserved
Type	RW	RW	RW	RW	RW	RW	RO	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-6 **Reserved bits**
These bits should be programmed by 0.

Bit 5-4 **JPEG FIFO Trigger Threshold bits [1:0]**
These bits set the JPEG FIFO Threshold Trigger Flag (REG[382h] bit 2) when the specified conditions are met.

Table 2-38: JPEG FIFO Trigger Threshold Selection

REG[03A0h] bits 5-4	JPEG FIFO Trigger Threshold
00	Never trigger
01	Trigger when the JPEG FIFO contains 4 bytes of data or more
10	Trigger when the JPEG FIFO contains more than 1/4 of the specified JPEG FIFO size (REG[3A4h] bits 6-0)
11	Trigger when the JPEG FIFO contains more than 1/2 of the specified JPEG FIFO size (REG[3A4h] bits 6-0)

Bits 3-2 **Reserved bit**
This bit should be programmed by 0.

Bit 1 **JPEG Output Select Bit**
This bit indicates the configuration of the JPEG FIFO.

When this bit = 0, the JPEG FIFO is configured to transmit (decode process).
When this bit = 1, the JPEG FIFO is configured to receive (encode process).

Bit 0 **Reserved bit**
This bit should be programmed by 0.

JPEG FIFO Status Register **REG[3A2h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	JPEG FIFO Threshold Status Bit 1	JPEG FIFO Threshold Status Bit 0	JPEG FIFO Full Status	JPEG FIFO Empty Status
Type	RW	RW	RW	RW	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bits 7-4 **Reserved bits**
These bits should be programmed by 0.

Bit 3-2 **JPEG FIFO Threshold Status bits [1:0]**
These bits indicate the amount of data in the JPEG FIFO.

Table 2-39: JPEG FIFO Threshold Status

REG[03A2h] bits 3-2	JPEG FIFO Threshold Status
00	No data (Same as Empty)
01	4 bytes of data or more exists
10	More than 1/4 of the specified JPEG FIFO size data exists (see REG[03A4h] bits 6-0)
11	More than 1/2 of the specified JPEG FIFO size data exists (see REG[03A4h] bits 6-0)

Bit 1 **JPEG FIFO Full Status**
This bit indicates whether the JPEG FIFO is full.
When this bit = 1, the JPEG FIFO is full.
When this bit = 0, the JPEG FIFO is not full.

Bit 0 **JPEG FIFO Empty Status**
This bit indicates that the JPEG FIFO is empty.
When this bit = 1, the JPEG FIFO is empty.
When this bit = 0, the JPEG FIFO is not empty.

JPEG FIFO Size Register **REG[3A4h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	JPEG FIFO Size Bit 6	JPEG FIFO Size Bit 5	JPEG FIFO Size Bit 4	JPEG FIFO Size Bit 3	JPEG FIFO Size Bit2	JPEG FIFO Size Bit 1	JPEG FIFO Size Bit 0
Type	RW	RW	RW	RW	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 7 **Reserved bit**
This bit should be programmed by 0.

Bits 6-0 **JPEG FIFO Size bits [6:0]**
These bits determine the JPEG FIFO size in 4K byte units. The maximum size of the JPEG FIFO is 512K bytes. These bits also specify the amount of memory reserved for the JPEG FIFO.
JPEG FIFO size = (REG[3A4h] bits 6-0 + 1) x 4K bytes

JPEG Encode Size Limit Register 0 **REG[3B0h]**

Bit	7	6	5	4	3	2	1	0
	Encode Size Limit Bit 7	Encode Size Limit Bit 6	Encode Size Limit Bit 5	Encode Size Limit Bit 4	Encode Size Limit Bit 3	Encode Size Limit Bit 2	Encode Size Limit Bit 1	Encode Size Limit Bit 0

Bit	7	6	5	4	3	2	1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG Encode Size Limit Register 1 **REG[3B1h]**

Bit	7	6	5	4	3	2	1	0
	Encode Size Limit Bit 15	Encode Size Limit Bit 14	Encode Size Limit Bit 13	Encode Size Limit Bit 12	Encode Size Limit Bit 11	Encode Size Limit Bit 10	Encode Size Limit Bit 9	Encode Size Limit Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG Encode Size Limit Register 2 **REG[3B2h]**

Bit	7	6	5	4	3	2	1	0
	Encode Size Limit Bit 23	Encode Size Limit Bit 22	Encode Size Limit Bit 21	Encode Size Limit Bit 20	Encode Size Limit Bit 19	Encode Size Limit Bit 18	Encode Size Limit Bit 17	Encode Size Limit Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[3B2h] Bits 7-0, REG[3B1h] Bits 7-0, REG[3B0h] Bits 7-0 — **JPEG Encode Size Limit bits [23:0]**
These bits are required for the JPEG encode process only. These bits specify the data size limit, in bytes, for the encoded JPEG file.

JPEG Encode Size Result Register 0 **REG[3B4h]**

Bit	7	6	5	4	3	2	1	0
	Encode Size Result Bit 7	Encode Size Result Bit 6	Encode Size Result Bit 5	Encode Size Result Bit 4	Encode Size Result Bit 3	Encode Size Result Bit 2	Encode Size Result Bit 1	Encode Size Result Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

JPEG Encode Size Result Register 1 **REG[3B5h]**

Bit	7	6	5	4	3	2	1	0
	Encode Size Result Bit 15	Encode Size Result Bit 14	Encode Size Result Bit 13	Encode Size Result Bit 12	Encode Size Result Bit 11	Encode Size Result Bit 10	Encode Size Result Bit 9	Encode Size Result Bit 8
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

JPEG Encode Size Result Register 2 **REG[3B6h]**

Bit	7	6	5	4	3	2	1	0
	Encode Size Result Bit 23	Encode Size Result Bit 22	Encode Size Result Bit 21	Encode Size Result Bit 20	Encode Size Result Bit 19	Encode Size Result Bit 18	Encode Size Result Bit 17	Encode Size Result Bit 16
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG[3B6h] Bits 7-0, REG[3B5h] Bits 7-0, REG[3B4h] Bits 7-0 — **JPEG Encode Size Result bits [23:0]**
These bits are required for the JPEG encode process only. These bits indicate the data size result, in bytes, for the encoded JPEG file.

JPEG File Size Register 0 **REG[3B8h]**

Bit	7	6	5	4	3	2	1	0
	JPEG File Size Bit 7	JPEG File Size Bit 6	JPEG File Size Bit 5	JPEG File Size Bit 4	JPEG File Size Bit 3	JPEG File Size Bit 2	JPEG File Size Bit 1	JPEG File Size Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG File Size Register 1 **REG[3B9h]**

Bit	7	6	5	4	3	2	1	0
	JPEG File Size Bit 15	JPEG File Size Bit 14	JPEG File Size Bit 13	JPEG File Size Bit 12	JPEG File Size Bit 11	JPEG File Size Bit 10	JPEG File Size Bit 9	JPEG File Size Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG File Size Register 2 **REG[3BAh]**

Bit	7	6	5	4	3	2	1	0
	JPEG File Size Bit 23	JPEG File Size Bit 22	JPEG File Size Bit 21	JPEG File Size Bit 20	JPEG File Size Bit 19	JPEG File Size Bit 18	JPEG File Size Bit 17	JPEG File Size Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[3BAh] Bits 7-0, REG[3B9h] Bits 7-0, REG[3B8h] Bits 7-0 — **JPEG File Size bits [23:0]**
These bits are required for the JPEG decode process only. These bits specify the JPEG file size in bytes and must be set before the Host begins writing decoded data to the JPEG FIFO.

JPEG Decode Horizontal Pixel Size Register 0 **REG[3D8h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Decode Horizontal Pixel Size Bit 7	JPEG Decode Horizontal Pixel Size Bit 6	JPEG Decode Horizontal Pixel Size Bit 5	JPEG Decode Horizontal Pixel Size Bit 4	JPEG Decode Horizontal Pixel Size Bit 3	JPEG Decode Horizontal Pixel Size Bit 2	JPEG Decode Horizontal Pixel Size Bit 1	JPEG Decode Horizontal Pixel Size Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

JPEG Decode Horizontal Pixel Size Register 1 **REG[3D9h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Decode Horizontal Pixel Size Bit 15	JPEG Decode Horizontal Pixel Size Bit 14	JPEG Decode Horizontal Pixel Size Bit 13	JPEG Decode Horizontal Pixel Size Bit 12	JPEG Decode Horizontal Pixel Size Bit 11	JPEG Decode Horizontal Pixel Size Bit 10	JPEG Decode Horizontal Pixel Size Bit 9	JPEG Decode Horizontal Pixel Size Bit 8
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG[3D9h] Bits 7-0, REG[3D8h] Bits 7-0 — **JPEG Decode Horizontal Pixel Size bits [15:0]**
These bits specify the horizontal image size during JPEG decode process.

JPEG Decode Vertical Pixel Size Register 0 **REG[3DCh]**

Bit	7	6	5	4	3	2	1	0
	JPEG	JPEG	JPEG	JPEG	JPEG	JPEG	JPEG	JPEG

Bit	7	6	5	4	3	2	1	0
	Decode Vertical Pixel Size Bit 7	Decode Vertical Pixel Size Bit 6	Decode Vertical Pixel Size Bit 5	Decode Vertical Pixel Size Bit 4	Decode Vertical Pixel Size Bit 3	Decode Vertical Pixel Size Bit 2	Decode Vertical Pixel Size Bit 1	Decode Vertical Pixel Size Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

JPEG Decode Vertical Pixel Size Register 1 **REG[3DDh]**

Bit	7	6	5	4	3	2	1	0
	JPEG Decode Vertical Pixel Size Bit 15	JPEG Decode Vertical Pixel Size Bit 14	JPEG Decode Vertical Pixel Size Bit 13	JPEG Decode Vertical Pixel Size Bit 12	JPEG Decode Vertical Pixel Size Bit 11	JPEG Decode Vertical Pixel Size Bit 10	JPEG Decode Vertical Pixel Size Bit 9	JPEG Decode Vertical Pixel Size Bit 8
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG[3DDh] Bits 7-0, REG[3DCh] Bits 7-0 **JPEG Decode Vertical Pixel Size bits [15:0]**
These bits specify the vertical image size during JPEG decode process.

JPEG Operation Mode Setting Register **REG[400h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	MJPEG Mode	Reserved	Marker Insert Enable	JPEG Operation Select	YUV Format Select Bit 1	YUV Format Select Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-6 **Reserved bits**
These bits should be programmed by 0.

Bit 5 **MJPEG mode**
This bit determines if Motion JPEG mode is used for encode or decode process.
When this bit = 1, Motion JPEG mode is used.
When this bit = 0, Still JPEG mode is used.

Bit 4 **Reserved bit**
This bit should be programmed by 0.

Bit 3 **Marker Insert Enable**
This bit determines if the marker (see REG[420h - 466h]) is inserted during JPEG encoding.
This bit is ignored during the JPEG decode process.
When this bit = 1, the marker is inserted.
When this bit = 0, the marker is not inserted.

Bit 2 **JPEG Operation Select**
This bit selects the JPEG operation.
When this bit = 1, JPEG codec operates in decode mode.
When this bit = 0, JPEG codec operates in encode mode.

Table 2-40: JPEG Operation Selection

REG[400h] bit 2	JPEG Operation
0	Encode
1	Decode

Bits 1-0 **YUV Encode Format Select bits [1:0]**
These bits are required for the JPEG encode process only.

These bits select the desired YUV format for the JPEG encode process.

Table 2-41: YUV Format Selection

REG[400h] bits 1-0	YUV Format
00	4:4:4
01	4:2:2
10	4:2:0
11	4:1:1

JPEG Operation Mode Setting Register **REG[401h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	YUV Decode Format Bit 1	YUV Decode Format Bit 0
Type	RW	RW	RW	RW	RW	RW	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bits 7-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **YUV Decode Format bits [1:0]**
These bits are required for the JPEG decode process only.
These bits indicate the YUV format of the data being decoded.

Table 2-42: YUV Format Selection

REG[401h] bits 1-0	YUV Format
00	4:4:4
01	4:2:2
10	4:2:0
11	4:1:1

JPEG Command Setting Register **REG[402h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Codec SW Reset	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	JPEG Operation Start
Type	WO	RW	RW	RW	RW	RW	RW	WO
Reset state	0	0	0	0	0	0	0	0

Bit 7 **JPEG Codec Software Reset**
This bit initiates a software reset of the JPEG Codec. The JPEG Codec registers (REG[400h]-[9A2h]) are not affected.
When a 1 is written to this bit, the JPEG Codec is reset.
When a 0 is written to this bit, there is no hardware effect.

Bits 6-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **JPEG Operation Start**
This bit is used to begin a JPEG operation.
When a 1 is written to this bit, the JPEG operation is started.
When a 0 is written to this bit, there is no hardware effect.

JPEG Operation Status Register **REG[404h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	JPEG Operation Status
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bits 7-1 **Reserved bit**

Bit 0 **JPEG Operation Status**
This bit indicates the state of the JPEG codec.
When this bit = 1, the JPEG codec is busy (a decode or encode operation is in progress).
When this bit = 0, the JPEG codec is idle.

JPEG Decode Quantization Table Number Register **REG[407h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Color 3 QTable Select	Color 2 QTable Select	Color 1 QTable Select
Type	RW	RW	RW	RW	RW	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 7-3 **Reserved bits**
These bits should be programmed by 0.

Bit 2 **Color 3 QTable Select**
This bit indicates the Quantization Table used by Color 3.

Bit 1 **Color 2 QTable Select**
This bit indicates the Quantization Table used by Color 2.

Bit 0 **Color 1 QTable Select**
This bit indicates the Quantization Table used by Color 1.

JPEG Decode Huffman Table Register **REG[408h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Color 3 AC HTable Select	Color 3 DC HTable Select	Color 2 AC HTable Select	Color 2 DC HTable Select	Color 1 AC HTable Select	Color 1 DC HTable Select
Type	RW	RW	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Bit 7-6 **Reserved bits**
These bits should be programmed by 0.

Bit 5 **Color 3 AC HTable Select**
This bit indicated the AC Huffman Table used by Color 3.

Bit 4 **Color 3 DC HTable Select**
This bit indicated the DC Huffman Table used by Color 3.

Bit 3 **Color 2 AC HTable Select**
This bit indicated the AC Huffman Table used by Color 2.

Bit 2 **Color 2 DC HTable Select**
This bit indicated the DC Huffman Table used by Color 2.

Bit 1 **Color 1 AC HTable Select**
This bit indicated the AC Huffman Table used by Color 1.

Bit 0 **Color 1 DC HTable Select**
This bit indicated the DC Huffman Table used by Color 1.

JPEG Decode DRI Setting Register 0 **REG[40Ah]**

Bit	7	6	5	4	3	2	1	0
	JPEG Decode DRI Setting bit 7	JPEG Decode DRI Setting bit 6	JPEG Decode DRI Setting bit 5	JPEG Decode DRI Setting bit 4	JPEG Decode DRI Setting bit 3	JPEG Decode DRI Setting bit 2	JPEG Decode DRI Setting bit 1	JPEG Decode DRI Setting bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

JPEG Decode DRI Setting Register 1 **REG[40Bh]**

Bit	7	6	5	4	3	2	1	0
	JPEG Decode DRI Setting bit 15	JPEG Decode DRI Setting bit 14	JPEG Decode DRI Setting bit 13	JPEG Decode DRI Setting bit 12	JPEG Decode DRI Setting bit 11	JPEG Decode DRI Setting bit 10	JPEG Decode DRI Setting bit 9	JPEG Decode DRI Setting bit 8
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG [40Bh] bits 7-0,
REG [40Ah] bits 7-0

JPEG Decode DRI Setting bits [15:0]
These bits indicate DRI Setting used for the JPEG decode process.

JPEG Line Buffer Start Address Register 0 **REG[410h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Line Buffer Start Address Bit 7	JPEG Line Buffer Start Address Bit 6	JPEG Line Buffer Start Address Bit 5	JPEG Line Buffer Start Address Bit 4	JPEG Line Buffer Start Address Bit 3	JPEG Line Buffer Start Address Bit 2	JPEG Line Buffer Start Address Bit 1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG Line Buffer Start Address Register 1 **REG[411h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Line Buffer Start Address Bit 15	JPEG Line Buffer Start Address Bit 14	JPEG Line Buffer Start Address Bit 13	JPEG Line Buffer Start Address Bit 12	JPEG Line Buffer Start Address Bit 11	JPEG Line Buffer Start Address Bit 10	JPEG Line Buffer Start Address Bit 9	JPEG Line Buffer Start Address Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG Line Buffer Start Address Register 2 **REG[412h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	JPEG Line Buffer Start Address Bit 17	JPEG Line Buffer Start Address Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[412h] Bits 1-0, **JPEG Line Buffer Start Address bits [17:0]**
REG[411h] Bits 7-0, **Bit 0 must be program as 0.**
REG[410h] Bits 7-0 These bits define the start address for the JPEG Line Buffer.
REG[412h] Bits 7-2 **Reserved bits**
These bits should be programmed by 0.

JPEG FIFO Start Address Register 0 **REG[414h]**

Bit	7	6	5	4	3	2	1	0
	JPEG FIFO Start Address Bit 7	JPEG FIFO Start Address Bit 6	JPEG FIFO Start Address Bit 5	JPEG FIFO Start Address Bit 4	JPEG FIFO Start Address Bit 3	JPEG FIFO Start Address Bit 2	JPEG FIFO Start Address Bit 1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG FIFO Start Address Register 1 **REG[415h]**

Bit	7	6	5	4	3	2	1	0
	JPEG FIFO Start Address Bit 15	JPEG FIFO Start Address Bit 14	JPEG FIFO Start Address Bit 13	JPEG FIFO Start Address Bit 12	JPEG FIFO Start Address Bit 11	JPEG FIFO Start Address Bit 10	JPEG FIFO Start Address Bit 9	JPEG FIFO Start Address Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

JPEG FIFO Start Address Register 2 **REG[416h]**

~~Bit~~	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	JPEG FIFO Start Address Bit 17	JPEG FIFO Start Address Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[416h] Bits 1-0, **JPEG FIFO Start Address bits [17:0]**
REG[415h] Bits 7-0, **Bit 0 must be program as 0**
REG[414h] Bits 7-0 These bits define the start address for the JPEG FIFO.
REG[416h] Bits 7-2 **Reserved bits**
These bits should be programmed by 0.

JPEG DNL Value Setting Register 0 **REG[418h]**

Bit	7	6	5	4	3	2	1	0
	DNL Value Bit 7	DNL Value Bit 6	DNL Value Bit 5	DNL Value Bit 4	DNL Value Bit 3	DNL Value Bit 2	DNL Value Bit 1	DNL Value Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

JPEG DNL Value Setting Register 1 **REG[419h]**

Bit	7	6	5	4	3	2	1	0
	DNL Value Bit 15	DNL Value Bit 14	DNL Value Bit 13	DNL Value Bit 12	DNL Value Bit 11	DNL Value Bit 10	DNL Value Bit 9	DNL Value Bit 8
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG[419h] Bits 7-0,
REG[418h] Bits 7-0

DNL Value bits [15:0]
For the JPEG decode process, these bits are read-only and indicate the DNL value.
For the JPEG encode process, these bits are not used.

JPEG RST Marker Operation Setting Register **REG[41Ch]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RST Marker Operation Select Bit 1	RST Marker Operation Select Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bits 7-2

Reserved bits
These bits should be programmed by 0.

Bit 1-0

RST Marker Operation Select bits [1:0]
For the JPEG decode process, these bits select the RST Marker Operation.
For the JPEG encode process, these bits are not used.

Table 2-43: RST Marker Selection

REG[41Ch] bits 1-0	RST Marker Operation
00	**Error detection and data revise function is turned off** This option should only be used when it is certain that the JPEG file to be decoded is correct and has no errors. If there is an error in the file, no error detection will take place and the decode process will not finish correctly.
01	**Error detection on** When an error is detected during the decode process, the decode process finishes and the JPEG interrupt is asserted (REG[48h] bit 2 = 1, to determine the exact nature of the error see REG[382h]). Because the decode process finished before normal completion, all data can not be displayed. If the JPEG file is to be decoded again with the Data Revise function on, a software reset is required (see REG[402h] bit 7).
10	**Data revise function on** When an error is detected during the decode process, data is skipped/added automatically and the decode process continues normally to the end of file. After the decode process finishes, a data revise interrupt is asserted. Because the decode process is finished completely, the next JPEG file can be decoded immediately.
11	Reserved

JPEG RST Marker Operation Status Register **REG[41Eh]**

Bit	7	6	5	4	3	2	1	0
	Revise Code	JPEG Error Status Bit 3	JPEG Error Status Bit 2	JPEG Error Status Bit 1	JPEG Error Status Bit 0	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Bit 7

Revise Code

For the JPEG decode process, this bit indicates whether a revise operation has been done.
For the JPEG encode process, this bit is not used.
When this bit = 1, a revise operation was done.
When this bit = 0, a revise operation was not done.

Bits 6-3 **JPEG Error Status [3:0]**
For the JPEG decode process, these bits indicate the type of JPEG error. If these bits return 0000, no error has occurred.
For the JPEG encode process, these bits are not used.

Table 2-44: JPEG Error Status

REG[41Eh] bits 6-3	JPEG Error Status
0000	No error
0001 – 1010	Reserved
1011	Restart interval error
1100	Image size error
1101 - 1111	Reserved

Bits 2-0 **Reserved bits**
These bits should be programmed by 0.

JPEG Insert Marker Data Register **REG[420-466h]**

Bit	7	6	5	4	3	2	1	0
	Insert Marker Value Bit 7	Insert Marker Value Bit 6	Insert Marker Value Bit 5	Insert Marker Value Bit 4	Insert Marker Value Bit 3	Insert Marker Value Bit 2	Insert Marker Value Bit 1	Insert Marker Value Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

These registers (36 bytes) contain data for marker insertion as follows.
REG[420-422h] These registers set the insertion marker code type.
REG[424-426h] These registers set the marker length (0002h - 0022h).
REG[428-466h] These registers set the marker data (maximum of 32 bytes).

Note
(1) REG[468h-500h] (Even Address only). These register are reserved, Must be write in special sequence.

SOI Marker Register 0 **REG[468h]**

Bit	7	6	5	4	3	2	1	0
	SOI Marker Register 0 Bit 7	SOI Marker Register 0 Bit 6	SOI Marker Register 0 Bit 5	SOI Marker Register 0 Bit 4	SOI Marker Register 0 Bit 3	SOI Marker Register 0 Bit 2	SOI Marker Register 0 Bit 1	SOI Marker Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

SOI Marker Register 1 **REG[46Ah]**

Bit	7	6	5	4	3	2	1	0
	SOI Marker Register 1 Bit 7	SOI Marker Register 1 Bit 6	SOI Marker Register 1 Bit 5	SOI Marker Register 1 Bit 4	SOI Marker Register 1 Bit 3	SOI Marker Register 1 Bit 2	SOI Marker Register 1 Bit 1	SOI Marker Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[468h] Bits 7-0, REG[46Ah] Bits 7-0 **SOI Marker (2 bytes)**
These registers specify the SOI marker and must be set to (hex) FF D8.

JFIF Marker Register 0 **REG[46Ch]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Register 0 Bit 7	JFIF Marker Register 0 Bit 6	JFIF Marker Register 0 Bit 5	JFIF Marker Register 0 Bit 4	JFIF Marker Register 0 Bit 3	JFIF Marker Register 0 Bit 2	JFIF Marker Register 0 Bit 1	JFIF Marker Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Register 1 **REG[46Eh]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Register 1 Bit 7	JFIF Marker Register 1 Bit 6	JFIF Marker Register 1 Bit 5	JFIF Marker Register 1 Bit 4	JFIF Marker Register 1 Bit 3	JFIF Marker Register 1 Bit 2	JFIF Marker Register 1 Bit 1	JFIF Marker Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[46Ch] Bits 7-0, REG[46Eh] Bits 7-0 **JFIF Marker (2 bytes)**
These registers specify the JFIF marker and must be set to (hex) FF E0.

JFIF Marker Size Register 0 **REG[470h]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Size Register 0 Bit 7	JFIF Marker Size Register 0 Bit 6	JFIF Marker Size Register 0 Bit 5	JFIF Marker Size Register 0 Bit 4	JFIF Marker Size Register 0 Bit 3	JFIF Marker Size Register 0 Bit 2	JFIF Marker Size Register 0 Bit 1	JFIF Marker Size Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Size Register 1 **REG[472h]**

Size Bit	7	6	5	4	3	2	1	0
	JFIF Marker Size Register 1 Bit 7	JFIF Marker Size Register 1 Bit 6	JFIF Marker Size Register 1 Bit 5	JFIF Marker Size Register 1 Bit 4	JFIF Marker Size Register 1 Bit 3	JFIF Marker Size Register 1 Bit 2	JFIF Marker Size Register 1 Bit 1	JFIF Marker Size Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[470h] Bits 7-0, REG[472h] Bits 7-0 **JFIF Marker Size (2 bytes)**
These registers specify the JFIF marker size and must be set to (hex) 00 10.

JFIF Marker Value Register 0 **REG[474h]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value	JFIF Marker Value	JFIF Marker Value	JFIF Marker Value	JFIF Marker Value	JFIF Marker Value	JFIF Marker Value	JFIF Marker Value

Bit	7	6	5	4	3	2	1	0
	Register 0 Bit 7	Register 0 Bit 6	Register 0 Bit 5	Register 0 Bit 4	Register 0 Bit 3	Register 0 Bit 2	Register 0 Bit 1	Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 1 **REG[476h]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 1 Bit 7	JFIF Marker Value Register 1 Bit 6	JFIF Marker Value Register 1 Bit 5	JFIF Marker Value Register 1 Bit 4	JFIF Marker Value Register 1 Bit 3	JFIF Marker Value Register 1 Bit 2	JFIF Marker Value Register 1 Bit 1	JFIF Marker Value Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 2 **REG[478h]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 2 Bit 7	JFIF Marker Value Register 2 Bit 6	JFIF Marker Value Register 2 Bit 5	JFIF Marker Value Register 2 Bit 4	JFIF Marker Value Register 2 Bit 3	JFIF Marker Value Register 2 Bit 2	JFIF Marker Value Register 2 Bit 1	JFIF Marker Value Register 2 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 3 **REG[47Ah]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 3 Bit 7	JFIF Marker Value Register 3 Bit 6	JFIF Marker Value Register 3 Bit 5	JFIF Marker Value Register 3 Bit 4	JFIF Marker Value Register 3 Bit 3	JFIF Marker Value Register 3 Bit 2	JFIF Marker Value Register 3 Bit 1	JFIF Marker Value Register 3 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 4 **REG[47Ch]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 4 Bit 7	JFIF Marker Value Register 4 Bit 6	JFIF Marker Value Register 4 Bit 5	JFIF Marker Value Register 4 Bit 4	JFIF Marker Value Register 4 Bit 3	JFIF Marker Value Register 4 Bit 2	JFIF Marker Value Register 4 Bit 1	JFIF Marker Value Register 4 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 5 **REG[47Eh]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 5	JFIF Marker Value Register 5	JFIF Marker Value Register 5	JFIF Marker Value Register 5	JFIF Marker Value Register 5	JFIF Marker Value Register 5	JFIF Marker Value Register 5	JFIF Marker Value Register 5

Bit	7	6	5	4	3	2	1	0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 6 **REG[480h]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 6 Bit 7	JFIF Marker Value Register 6 Bit 6	JFIF Marker Value Register 6 Bit 5	JFIF Marker Value Register 6 Bit 4	JFIF Marker Value Register 6 Bit 3	JFIF Marker Value Register 6 Bit 2	JFIF Marker Value Register 6 Bit 1	JFIF Marker Value Register 6 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 7 **REG[482h]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 7 Bit 7	JFIF Marker Value Register 7 Bit 6	JFIF Marker Value Register 7 Bit 5	JFIF Marker Value Register 7 Bit 4	JFIF Marker Value Register 7 Bit 3	JFIF Marker Value Register 7 Bit 2	JFIF Marker Value Register 7 Bit 1	JFIF Marker Value Register 7 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 8 **REG[484h]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 8 Bit 7	JFIF Marker Value Register 8 Bit 6	JFIF Marker Value Register 8 Bit 5	JFIF Marker Value Register 8 Bit 4	JFIF Marker Value Register 8 Bit 3	JFIF Marker Value Register 8 Bit 2	JFIF Marker Value Register 8 Bit 1	JFIF Marker Value Register 8 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 9 **REG[486h]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 9 Bit 7	JFIF Marker Value Register 9 Bit 6	JFIF Marker Value Register 9 Bit 5	JFIF Marker Value Register 9 Bit 4	JFIF Marker Value Register 9 Bit 3	JFIF Marker Value Register 9 Bit 2	JFIF Marker Value Register 9 Bit 1	JFIF Marker Value Register 9 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 10 **REG[488h]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 10 Bit 7	JFIF Marker Value Register 10 Bit 6	JFIF Marker Value Register 10 Bit 5	JFIF Marker Value Register 10 Bit 4	JFIF Marker Value Register 10 Bit 3	JFIF Marker Value Register 10 Bit 2	JFIF Marker Value Register 10 Bit 1	JFIF Marker Value Register 10 Bit 0

Bit	7	6	5	4	3	2	1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 11 **REG[48Ah]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 11 Bit 7	JFIF Marker Value Register 11 Bit 6	JFIF Marker Value Register 11 Bit 5	JFIF Marker Value Register 11 Bit 4	JFIF Marker Value Register 11 Bit 3	JFIF Marker Value Register 11 Bit 2	JFIF Marker Value Register 11 Bit 1	JFIF Marker Value Register 11 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 12 **REG[48Ch]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 12 Bit 7	JFIF Marker Value Register 12 Bit 6	JFIF Marker Value Register 12 Bit 5	JFIF Marker Value Register 12 Bit 4	JFIF Marker Value Register 12 Bit 3	JFIF Marker Value Register 12 Bit 2	JFIF Marker Value Register 12 Bit 1	JFIF Marker Value Register 12 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JFIF Marker Value Register 13 **REG[48Eh]**

Bit	7	6	5	4	3	2	1	0
	JFIF Marker Value Register 13 Bit 7	JFIF Marker Value Register 13 Bit 6	JFIF Marker Value Register 13 Bit 5	JFIF Marker Value Register 13 Bit 4	JFIF Marker Value Register 13 Bit 3	JFIF Marker Value Register 13 Bit 2	JFIF Marker Value Register 13 Bit 1	JFIF Marker Value Register 13 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[474h] Bits 7-0,
REG[476h] Bits 7-0,
REG[478h] Bits 7-0,
REG[47Ah] Bits 7-0,
REG[47Ch] Bits 7-0,
REG[47Eh] Bits 7-0,
REG[480h] Bits 7-0,
REG[482h] Bits 7-0,
REG[484h] Bits 7-0,
REG[486h] Bits 7-0,
REG[488h] Bits 7-0,
REG[48Ah] Bits 7-0,
REG[48Ch] Bits 7-0,
REG[48Eh] Bits 7-0

JFIF Marker Value (14 bytes)
These registers specify the JFIF marker value and must be set to (hex) 4A 46 49 46 00 01 02 01 00 48 00 48 00 00.

DHT Marker LUM DC Register 0 **REG[490h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker LUM DC	DHT Marker LUM DC	DHT Marker LUM DC	DHT Marker LUM DC	DHT Marker LUM DC	DHT Marker LUM DC	DHT Marker LUM DC	DHT Marker LUM DC

Bit	7	6	5	4	3	2	1	0
	Register 0 Bit 7	Register 0 Bit 6	Register 0 Bit 5	Register 0 Bit 4	Register 0 Bit 3	Register 0 Bit 2	Register 0 Bit 1	Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DHT Marker LUM DC Register 1 **REG[492h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker LUM DC Register 1 Bit 7	DHT Marker LUM DC Register 1 Bit 6	DHT Marker LUM DC Register 1 Bit 5	DHT Marker LUM DC Register 1 Bit 4	DHT Marker LUM DC Register 1 Bit 3	DHT Marker LUM DC Register 1 Bit 2	DHT Marker LUM DC Register 1 Bit 1	DHT Marker LUM DC Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[490h] Bits 7-0, REG[492h] Bits 7-0 **DHT Marker LUM DC (2 bytes)**
These registers specify the DHT marker for the Luminance DC Huffman table and must be set to (hex) FF C4.

DHT Marker Size LUM DC Register 0 **REG[494h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker Size LUM DC Register 0 Bit 7	DHT Marker Size LUM DC Register 0 Bit 6	DHT Marker Size LUM DC Register 0 Bit 5	DHT Marker Size LUM DC Register 0 Bit 4	DHT Marker Size LUM DC Register 0 Bit 3	DHT Marker Size LUM DC Register 0 Bit 2	DHT Marker Size LUM DC Register 0 Bit 1	DHT Marker Size LUM DC Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DHT Marker Size LUM DC Register 1 **REG[496h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker Size LUM DC Register 1 Bit 7	DHT Marker Size LUM DC Register 1 Bit 6	DHT Marker Size LUM DC Register 1 Bit 5	DHT Marker Size LUM DC Register 1 Bit 4	DHT Marker Size LUM DC Register 1 Bit 3	DHT Marker Size LUM DC Register 1 Bit 2	DHT Marker Size LUM DC Register 1 Bit 1	DHT Marker Size LUM DC Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[494h] Bits 7-0, REG[496h] Bits 7-0 **DHT Marker Size LUM DC (2 bytes)**
These registers specify the DHT marker size for the Luminance DC Huffman table and must be set to (hex) 00 1F.

DHT Marker LUM DC ID Register **REG[498h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker LUM DC ID Bit 7	DHT Marker LUM DC ID Bit 6	DHT Marker LUM DC ID Bit 5	DHT Marker LUM DC ID Bit 4	DHT Marker LUM DC ID Bit 3	DHT Marker LUM DC ID Bit 2	DHT Marker LUM DC ID Bit 1	DHT Marker LUM DC ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	NA	NA	NA	NA	NA	NA	NA	NA

Bit	7	6	5	4	3	2	1	0
state								

Bits 7-0 **DHT Marker LUM DC ID bits [7:0]**
These bits specify the DHT marker ID for the Luminance DC Huffman table and must be set to (hex) 00.

DHT Marker CHR DC Register 0 **REG[4A0h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker CHR DC Register 0 Bit 7	DHT Marker CHR DC Register 0 Bit 6	DHT Marker CHR DC Register 0 Bit 5	DHT Marker CHR DC Register 0 Bit 4	DHT Marker CHR DC Register 0 Bit 3	DHT Marker CHR DC Register 0 Bit 2	DHT Marker CHR DC Register 0 Bit 1	DHT Marker CHR DC Register 0 Bit 0
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DHT Marker CHR DC Register 1 **REG[4A2h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker CHR DC Register 1 Bit 7	DHT Marker CHR DC Register 1 Bit 6	DHT Marker CHR DC Register 1 Bit 5	DHT Marker CHR DC Register 1 Bit 4	DHT Marker CHR DC Register 1 Bit 3	DHT Marker CHR DC Register 1 Bit 2	DHT Marker CHR DC Register 1 Bit 1	DHT Marker CHR DC Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4A0h] Bits 7-0,
REG[4A2h] Bits 7-0 **DHT Marker CHR DC (2 bytes)**
These registers specify the DHT marker for the Chrominance DC Huffman table and must be set to (hex) FF C4.

DHT Marker Size CHR DC Register 0 **REG[4A4h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker Size CHR DC Register 0 Bit 7	DHT Marker Size CHR DC Register 0 Bit 6	DHT Marker Size CHR DC Register 0 Bit 5	DHT Marker Size CHR DC Register 0 Bit 4	DHT Marker Size CHR DC Register 0 Bit 3	DHT Marker Size CHR DC Register 0 Bit 2	DHT Marker Size CHR DC Register 0 Bit 1	DHT Marker Size CHR DC Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DHT Marker Size CHR DC Register 1 **REG[4A6h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker Size CHR DC Register 1 Bit 7	DHT Marker Size CHR DC Register 1 Bit 6	DHT Marker Size CHR DC Register 1 Bit 5	DHT Marker Size CHR DC Register 1 Bit 4	DHT Marker Size CHR DC Register 1 Bit 3	DHT Marker Size CHR DC Register 1 Bit 2	DHT Marker Size CHR DC Register 1 Bit 1	DHT Marker Size CHR DC Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4A4h] Bits 7-0,
REG[4A6h] Bits 7-0 **DHT Marker Size CHR DC (2 bytes)**
These registers specify the DHT marker size for the Chrominance DC Huffman table and must

be set to (hex) 00 1F.

DHT Marker CHR DC ID Register **REG[4A8h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker CHR DC ID Bit 7	DHT Marker CHR DC ID Bit 6	DHT Marker CHR DC ID Bit 5	DHT Marker CHR DC ID Bit 4	DHT Marker CHR DC ID Bit 3	DHT Marker CHR DC ID Bit 2	DHT Marker CHR DC ID Bit 1	DHT Marker CHR DC ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **DHT Marker CHR DC ID bits [7:0]**
These bits specify the DHT marker ID for the Chrominance DC Huffman table and must be set to (hex) 01.

DHT Marker LUM AC Register 0 **REG[4B0h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker LUM AC Register 0 Bit 7	DHT Marker LUM AC Register 0 Bit 6	DHT Marker LUM AC Register 0 Bit 5	DHT Marker LUM AC Register 0 Bit 4	DHT Marker LUM AC Register 0 Bit 3	DHT Marker LUM AC Register 0 Bit 2	DHT Marker LUM AC Register 0 Bit 1	DHT Marker LUM AC Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DHT Marker LUM AC Register 1 **REG[4B2h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker LUM AC Register 1 Bit 7	DHT Marker LUM AC Register 1 Bit 6	DHT Marker LUM AC Register 1 Bit 5	DHT Marker LUM AC Register 1 Bit 4	DHT Marker LUM AC Register 1 Bit 3	DHT Marker LUM AC Register 1 Bit 2	DHT Marker LUM AC Register 1 Bit 1	DHT Marker LUM AC Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4B0h] Bits 7-0, REG[4B2h] Bits 7-0 **DHT Marker LUM AC (2 bytes)**
These registers specify the DHT marker for the Luminance AC Huffman table and must be set to (hex) FF C4.

DHT Marker Size LUM AC Register 0 **REG[4B4h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker Size LUM AC Register 0 Bit 7	DHT Marker Size LUM AC Register 0 Bit 6	DHT Marker Size LUM AC Register 0 Bit 5	DHT Marker Size LUM AC Register 0 Bit 4	DHT Marker Size LUM AC Register 0 Bit 3	DHT Marker Size LUM AC Register 0 Bit 2	DHT Marker Size LUM AC Register 0 Bit 1	DHT Marker Size LUM AC Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DHT Marker Size LUM AC Register 1 **REG[4B6h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker	DHT Marker	DHT Marker	DHT Marker	DHT Marker	DHT Marker	DHT Marker	DHT Marker

Bit	7	6	5	4	3	2	1	0
	Size LUM AC Register 1 Bit 7	Size LUM AC Register 1 Bit 6	Size LUM AC Register 1 Bit 5	Size LUM AC Register 1 Bit 4	Size LUM AC Register 1 Bit 3	Size LUM AC Register 1 Bit 2	Size LUM AC Register 1 Bit 1	Size LUM AC Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4B4h] Bits 7-0, REG[4B6h] Bits 7-0 **DHT Marker Size LUM AC (2 bytes)**
These registers specify the DHT marker size for the Luminance AC Huffman table and must be set to (hex) 00 B5.

DHT Marker LUM AC ID Register **REG[4B8h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker LUM AC ID Bit 7	DHT Marker LUM AC ID Bit 6	DHT Marker LUM AC ID Bit 5	DHT Marker LUM AC ID Bit 4	DHT Marker LUM AC ID Bit 3	DHT Marker LUM AC ID Bit 2	DHT Marker LUM AC ID Bit 1	DHT Marker LUM AC ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **DHT Marker LUM AC ID bits [7:0]**
These bits specify the DHT marker ID for the Luminance AC Huffman table and must be set to (hex) 10.

DHT Marker CHR AC Register 0 **REG[4C0h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker CHR AC Register 0 Bit 7	DHT Marker CHR AC Register 0 Bit 6	DHT Marker CHR AC Register 0 Bit 5	DHT Marker CHR AC Register 0 Bit 4	DHT Marker CHR AC Register 0 Bit 3	DHT Marker CHR AC Register 0 Bit 2	DHT Marker CHR AC Register 0 Bit 1	DHT Marker CHR AC Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DHT Marker CHR AC Register 1 **REG[4C2h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker CHR AC Register 1 Bit 7	DHT Marker CHR AC Register 1 Bit 6	DHT Marker CHR AC Register 1 Bit 5	DHT Marker CHR AC Register 1 Bit 4	DHT Marker CHR AC Register 1 Bit 3	DHT Marker CHR AC Register 1 Bit 2	DHT Marker CHR AC Register 1 Bit 1	DHT Marker CHR AC Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4C0h] Bits 7-0, REG[4C2h] Bits 7-0 **DHT Marker CHR AC (2 bytes)**
These registers specify the DHT marker for the Chrominance AC Huffman table and must be set to (hex) FF C4.

DHT Marker Size CHR AC Register 0 **REG[4C4h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker	DHT Marker	DHT Marker	DHT Marker	DHT Marker	DHT Marker	DHT Marker	DHT Marker

Bit	7	6	5	4	3	2	1	0
	Size CHR AC Register 0 Bit 7	Size CHR AC Register 0 Bit 6	Size CHR AC Register 0 Bit 5	Size CHR AC Register 0 Bit 4	Size CHR AC Register 0 Bit 3	Size CHR AC Register 0 Bit 2	Size CHR AC Register 0 Bit 1	Size CHR AC Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DHT Marker Size CHR AC Register 1 **REG[4C6h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker Size CHR AC Register 1 Bit 7	DHT Marker Size CHR AC Register 1 Bit 6	DHT Marker Size CHR AC Register 1 Bit 5	DHT Marker Size CHR AC Register 1 Bit 4	DHT Marker Size CHR AC Register 1 Bit 3	DHT Marker Size CHR AC Register 1 Bit 2	DHT Marker Size CHR AC Register 1 Bit 1	DHT Marker Size CHR AC Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4C4h] Bits 7-0, REG[4C6h] Bits 7-0 **DHT Marker Size CHR AC (2 bytes)**
These registers specify the DHT marker size for the Chrominance AC Huffman table and must be set to (hex) 00 B5.

DHT Marker CHR AC ID Register **REG[4C8h]**

Bit	7	6	5	4	3	2	1	0
	DHT Marker CHR AC ID Bit 7	DHT Marker CHR AC ID Bit 6	DHT Marker CHR AC ID Bit 5	DHT Marker CHR AC ID Bit 4	DHT Marker CHR AC ID Bit 3	DHT Marker CHR AC ID Bit 2	DHT Marker CHR AC ID Bit 1	DHT Marker CHR AC ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **DHT Marker CHR AC ID bits [7:0]**
These bits specify the DHT marker ID for the Chrominance AC Huffman table and must be set to (hex) 11.

DQT Marker LUM Register 0 **REG[4D0h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker LUM Register 0 Bit 7	DQT Marker LUM Register 0 Bit 6	DQT Marker LUM Register 0 Bit 5	DQT Marker LUM Register 0 Bit 4	DQT Marker LUM Register 0 Bit 3	DQT Marker LUM Register 0 Bit 2	DQT Marker LUM Register 0 Bit 1	DQT Marker LUM Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DQT Marker LUM Register 1 **REG[4D2h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker LUM Register 1 Bit 7	DQT Marker LUM Register 1 Bit 6	DQT Marker LUM Register 1 Bit 5	DQT Marker LUM Register 1 Bit 4	DQT Marker LUM Register 1 Bit 3	DQT Marker LUM Register 1 Bit 2	DQT Marker LUM Register 1 Bit 1	DQT Marker LUM Register 1 Bit 0

Bit	7	6	5	4	3	2	1	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4D0h] Bits 7-0, REG[4D2h] Bits 7-0 **DQT Marker LUM (2 bytes)**
These registers specify the DQT marker for the Luminance quantization table and must be set to (hex) FF DB.

DQT Marker Size LUM Register 0 **REG[4D4h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker Size LUM Register 0 Bit 7	DQT Marker Size LUM Register 0 Bit 6	DQT Marker Size LUM Register 0 Bit 5	DQT Marker Size LUM Register 0 Bit 4	DQT Marker Size LUM Register 0 Bit 3	DQT Marker Size LUM Register 0 Bit 2	DQT Marker Size LUM Register 0 Bit 1	DQT Marker Size LUM Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DQT Marker Size LUM Register 1 **REG[4D6h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker Size LUM Register 1 Bit 7	DQT Marker Size LUM Register 1 Bit 6	DQT Marker Size LUM Register 1 Bit 5	DQT Marker Size LUM Register 1 Bit 4	DQT Marker Size LUM Register 1 Bit 3	DQT Marker Size LUM Register 1 Bit 2	DQT Marker Size LUM Register 1 Bit 1	DQT Marker Size LUM Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4D4h] Bits 7-0, REG[4D6h] Bits 7-0 **DQT Marker Size LUM (2 bytes)**
These registers specify the DQT marker size for the Luminance quantization table and must be set to (hex) 00 43.

DQT Marker LUM ID Register **REG[4D8h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker LUM ID Bit 7	DQT Marker LUM ID Bit 6	DQT Marker LUM ID Bit 5	DQT Marker LUM ID Bit 4	DQT Marker LUM ID Bit 3	DQT Marker LUM ID Bit 2	DQT Marker LUM ID Bit 1	DQT Marker LUM ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **DQT Marker LUM ID bits [7:0]**
These bits specify the DQT marker ID for the Luminance quantization table and must be set to (hex) 00.

DQT Marker CHR Register 0 **REG[4E0h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker CHR Register 0 Bit 7	DQT Marker CHR Register 0 Bit 6	DQT Marker CHR Register 0 Bit 5	DQT Marker CHR Register 0 Bit 4	DQT Marker CHR Register 0 Bit 3	DQT Marker CHR Register 0 Bit 2	DQT Marker CHR Register 0 Bit 1	DQT Marker CHR Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DQT Marker CHR Register 1 **REG[4E2h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker CHR Register 1 Bit 7	DQT Marker CHR Register 1 Bit 6	DQT Marker CHR Register 1 Bit 5	DQT Marker CHR Register 1 Bit 4	DQT Marker CHR Register 1 Bit 3	DQT Marker CHR Register 1 Bit 2	DQT Marker CHR Register 1 Bit 1	DQT Marker CHR Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4E0h] Bits 7-0, REG[4E2h] Bits 7-0 **DQT Marker CHR (2 bytes)**
These registers specify the DQT marker for the Chrominance quantization table and must be set to (hex) FF DB.

QT Marker Size CHR Register 0 **REG[4E4h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker Size CHR Register 0 Bit 7	DQT Marker Size CHR Register 0 Bit 6	DQT Marker Size CHR Register 0 Bit 5	DQT Marker Size CHR Register 0 Bit 4	DQT Marker Size CHR Register 0 Bit 3	DQT Marker Size CHR Register 0 Bit 2	DQT Marker Size CHR Register 0 Bit 1	DQT Marker Size CHR Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DQT Marker Size CHR Register 1 **REG[4E6h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker Size CHR Register 1 Bit 7	DQT Marker Size CHR Register 1 Bit 6	DQT Marker Size CHR Register 1 Bit 5	DQT Marker Size CHR Register 1 Bit 4	DQT Marker Size CHR Register 1 Bit 3	DQT Marker Size CHR Register 1 Bit 2	DQT Marker Size CHR Register 1 Bit 1	DQT Marker Size CHR Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4E4h] Bits 7-0, REG[4E6h] Bits 7-0 **DQT Marker Size CHR (2 bytes)**
These registers specify the DQT marker size for the Chrominance quantization table and must be set to (hex) 00 43.

DQT Marker CHR ID Register **REG[4E8h]**

Bit	7	6	5	4	3	2	1	0
	DQT Marker CHR ID Bit 7	DQT Marker CHR ID Bit 6	DQT Marker CHR ID Bit 5	DQT Marker CHR ID Bit 4	DQT Marker CHR ID Bit 3	DQT Marker CHR ID Bit 2	DQT Marker CHR ID Bit 1	DQT Marker CHR ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **DQT Marker CHR ID bits [7:0]**
These bits specify the DQT marker ID for the Chrominance quantization table and must be set to (hex) 01.

SOF Marker Register 0 **REG[4F0h]**

Bit	7	6	5	4	3	2	1	0
	SOF Marker Register 0 Bit 7	SOF Marker Register 0 Bit 6	SOF Marker Register 0 Bit 5	SOF Marker Register 0 Bit 4	SOF Marker Register 0 Bit 3	SOF Marker Register 0 Bit 2	SOF Marker Register 0 Bit 1	SOF Marker Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	7	6	5	4	3	2	1	0
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

SOF Marker Register 1 **REG[4F2h]**

Bit	7	6	5	4	3	2	1	0
	SOF Marker Register 1 Bit 7	SOF Marker Register 1 Bit 6	SOF Marker Register 1 Bit 5	SOF Marker Register 1 Bit 4	SOF Marker Register 1 Bit 3	SOF Marker Register 1 Bit 2	SOF Marker Register 1 Bit 1	SOF Marker Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4F0h] Bits 7-0, REG[4F2h] Bits 7-0 **SOF Marker (2 bytes)**
These registers specify the SOF marker and must be set to (hex) FF C0.

SOF Marker Size Register 0 **REG[4F4h]**

Bit	7	6	5	4	3	2	1	0
	SOF Marker Size Register 0 Bit 7	SOF Marker Size Register 0 Bit 6	SOF Marker Size Register 0 Bit 5	SOF Marker Size Register 0 Bit 4	SOF Marker Size Register 0 Bit 3	SOF Marker Size Register 0 Bit 2	SOF Marker Size Register 0 Bit 1	SOF Marker Size Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

SOF Marker Size Register 1 **REG[4F6h]**

Bit	7	6	5	4	3	2	1	0
	SOF Marker Size Register 1 Bit 7	SOF Marker Size Register 1 Bit 6	SOF Marker Size Register 1 Bit 5	SOF Marker Size Register 1 Bit 4	SOF Marker Size Register 1 Bit 3	SOF Marker Size Register 1 Bit 2	SOF Marker Size Register 1 Bit 1	SOF Marker Size Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[4F4h] Bits 7-0, REG[4F6h] Bits 7-0 **SOF Marker Size (2 bytes)**
These registers specify the DQT marker size and must be set to (hex) 00 11.

SOF Marker PR Register **REG[4F8h]**

Bit	7	6	5	4	3	2	1	0
	SOF Marker PR Bit 7	SOF Marker PR Bit 6	SOF Marker PR Bit 5	SOF Marker PR Bit 4	SOF Marker PR Bit 3	SOF Marker PR Bit 2	SOF Marker PR Bit 1	SOF Marker PR Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOF Marker PR bits [7:0]**
These bits specify the SOF marker sample precision and must be set to (hex) 08.

JPEG Quantization Table No. 0 Register **REG[500-57Eh]**

Bit	7	6	5	4	3	2	1	0
	Quantization Table No. 0 Bit 7	Quantization Table No. 0 Bit 6	Quantization Table No. 0 Bit 5	Quantization Table No. 0 Bit 4	Quantization Table No. 0 Bit 3	Quantization Table No. 0 Bit 2	Quantization Table No. 0 Bit 1	Quantization Table No. 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	7	6	5	4	3	2	1	0
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[500-57Eh] **Quantization Table No. 0**

These registers are used for the JPEG encode process only.

JPEG Quantization Table No. 1 Register **REG[580-5FEh]**

Bit	7	6	5	4	3	2	1	0
	Quantization Table No. 1 Bit 7	Quantization Table No. 1 Bit 6	Quantization Table No. 1 Bit 5	Quantization Table No. 1 Bit 4	Quantization Table No. 1 Bit 3	Quantization Table No. 1 Bit 2	Quantization Table No. 1 Bit 1	Quantization Table No. 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[580-5FEh] **Quantization Table No. 1**

These registers are used for the JPEG encode process only.

DC Huffman Table No. 0 Register 0 **REG[600-61Eh]**

Bit	7	6	5	4	3	2	1	0
	DC Huffman Table No. 0 Register 0 Bit 7	DC Huffman Table No. 0 Register 0 Bit 6	DC Huffman Table No. 0 Register 0 Bit 5	DC Huffman Table No. 0 Register 0 Bit 4	DC Huffman Table No. 0 Register 0 Bit 3	DC Huffman Table No. 0 Register 0 Bit 2	DC Huffman Table No. 0 Register 0 Bit 1	DC Huffman Table No. 0 Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[600-61Eh] **DC Huffman Table No. 0**

These registers are used for the JPEG encode process only and set the codes for code length.

DC Huffman Table No. 0 Register 1 **REG[620-636h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	DC Huffman Table No. 0 Register 1 Bit 3	DC Huffman Table No. 0 Register 1 Bit 2	DC Huffman Table No. 0 Register 1 Bit 1	DC Huffman Table No. 0 Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[620-636h] **DC Huffman Table No. 0**

These registers are used for the JPEG encode process only and set a group number based on the order of probability of occurrence. Only bits 3-0 are used (bits 7-4 must be set to 0).

AC Huffman Table No. 0 Register 0 **REG[640-65Eh]**

Bit	7	6	5	4	3	2	1	0
	AC Huffman Table No. 0 Register 0 Bit 7	AC Huffman Table No. 0 Register 0 Bit 6	AC Huffman Table No. 0 Register 0 Bit 5	AC Huffman Table No. 0 Register 0 Bit 4	AC Huffman Table No. 0 Register 0 Bit 3	AC Huffman Table No. 0 Register 0 Bit 2	AC Huffman Table No. 0 Register 0 Bit 1	AC Huffman Table No. 0 Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	NA	NA	NA	NA	NA	NA	NA	NA

Bit	7	6	5	4	3	2	1	0
state								

REG[640-65Eh] **AC Huffman Table No. 0**

These registers are used for the JPEG encode process only and set the codes for code length.

AC Huffman Table No. 0 Register 1 **REG[660-7A2h]**

Bit	7	6	5	4	3	2	1	0
	AC Huffman Table No. 0 Register 1 Bit 7	AC Huffman Table No. 0 Register 1 Bit 6	AC Huffman Table No. 0 Register 1 Bit 5	AC Huffman Table No. 0 Register 1 Bit 4	AC Huffman Table No. 0 Register 1 Bit 3	AC Huffman Table No. 0 Register 1 Bit 2	AC Huffman Table No. 0 Register 1 Bit 1	AC Huffman Table No. 0 Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[660-7A2h] AC Huffman Table No. 0

These registers are used for the JPEG encode process only and set a zero run length / group number based on the order of probability of occurrence.

DC Huffman Table No. 1 Register 0 **REG[800-81Eh]**

Bit	7	6	5	4	3	2	1	0
	DC Huffman Table No. 1 Register 0 Bit 7	DC Huffman Table No. 1 Register 0 Bit 6	DC Huffman Table No. 1 Register 0 Bit 5	DC Huffman Table No. 1 Register 0 Bit 4	DC Huffman Table No. 1 Register 0 Bit 3	DC Huffman Table No. 1 Register 0 Bit 2	DC Huffman Table No. 1 Register 0 Bit 1	DC Huffman Table No. 1 Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[800-81Eh] **DC Huffman Table No. 1**

These registers are used for the JPEG encode process only and set the codes for code length.

DC Huffman Table No. 1 Register 1 **REG[820-836h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	DC Huffman Table No. 1 Register 1 Bit 3	DC Huffman Table No. 1 Register 1 Bit 2	DC Huffman Table No. 1 Register 1 Bit 1	DC Huffman Table No. 1 Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[820-836h] **DC Huffman Table No. 1**

These registers are used for the JPEG encode process only and set a group number based on the order of probability of occurrence. Only bits 3-0 are used (bits 7-4 must be set to 0).

AC Huffman Table No. 1 Register 0 **REG[840-85Eh]**

Bit	7	6	5	4	3	2	1	0
	AC Huffman Table No. 1 Register 0	AC Huffman Table No. 1 Register 0	AC Huffman Table No. 1 Register 0	AC Huffman Table No. 1 Register 0	AC Huffman Table No. 1 Register 0	AC Huffman Table No. 1 Register 0	AC Huffman Table No. 1 Register 0	AC Huffman Table No. 1 Register 0

Bit	7	6	5	4	3	2	1	0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[840-85Eh] **AC Huffman Table No. 1**

These registers are used for the JPEG encode process only and set the codes for code length.

AC Huffman Table No. 1 Register 1 **REG[860-9A2h]**

Bit	7	6	5	4	3	2	1	0
	AC Huffman Table No. 1 Register 1 Bit 7	AC Huffman Table No. 1 Register 1 Bit 6	AC Huffman Table No. 1 Register 1 Bit 5	AC Huffman Table No. 1 Register 1 Bit 4	AC Huffman Table No. 1 Register 1 Bit 3	AC Huffman Table No. 1 Register 1 Bit 2	AC Huffman Table No. 1 Register 1 Bit 1	AC Huffman Table No. 1 Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[860-8A2h] **AC Huffman Table No. 0**

These registers are used for the JPEG encode process only and set a zero run length / group number based on the order of probability of occurrence.

SOF Marker NUM COMP Register **REG[9B0h]**

Bit	7	6	5	4	3	2	1	0
	SOF Marker NUM COMP Bit 7	SOF Marker NUM COMP Bit 6	SOF Marker NUM COMP Bit 5	SOF Marker NUM COMP Bit 4	SOF Marker NUM COMP Bit 3	SOF Marker NUM COMP Bit 2	SOF Marker NUM COMP Bit 1	SOF Marker NUM COMP Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOF Marker NUM COMP bits [7:0]**

These bits specify the SOF marker number of color components and must be set to (hex) 03.

SOF Marker Color 1 ID Register **REG[9B2h]**

Bit	7	6	5	4	3	2	1	0
	SOF Marker Color 1 ID Bit 7	SOF Marker Color 1 ID Bit 6	SOF Marker Color 1 ID Bit 5	SOF Marker Color 1 ID Bit 4	SOF Marker Color 1 ID Bit 3	SOF Marker Color 1 ID Bit 2	SOF Marker Color 1 ID Bit 1	SOF Marker Color 1 ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOF Marker Color 1 ID bits [7:0]**

These bits specify the SOF marker Color 1 ID and must be set to (hex) 01.

SOF Marker Color 2 ID Register **REG[9B4h]**

Bit	7	6	5	4	3	2	1	0
	SOF Marker Color 2 ID Bit 7	SOF Marker Color 2 ID Bit 6	SOF Marker Color 2 ID Bit 5	SOF Marker Color 2 ID Bit 4	SOF Marker Color 2 ID Bit 3	SOF Marker Color 2 ID Bit 2	SOF Marker Color 2 ID Bit 1	SOF Marker Color 2 ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	NA	NA	NA	NA	NA	NA	NA	NA

Bit	7	6	5	4	3	2	1	0
state								

Bits 7-0 **SOF Marker Color 2 ID bits [7:0]**
These bits specify the SOF marker Color 2 ID and must be set to (hex) 02.

SOF Marker Color 3 ID Register **REG[9B6h]**

Bit	7	6	5	4	3	2	1	0
	SOF Marker Color 3 ID Bit 7	SOF Marker Color 3 ID Bit 6	SOF Marker Color 3 ID Bit 5	SOF Marker Color 3 ID Bit 4	SOF Marker Color 3 ID Bit 3	SOF Marker Color 3 ID Bit 2	SOF Marker Color 3 ID Bit 1	SOF Marker Color 3 ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOF Marker Color 3 ID bits [7:0]**
These bits specify the SOF marker Color 3 ID and must be set to (hex) 03.

SOF Marker Color 1 HV Register **REG[9B8h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Color 1 Hsamp Bit 2	Color 1 Hsamp Bit 1	Color 1 Hsamp Bit 0	Reserved	Reserved	Color 1 Vsamp Bit 1	Color 1 Vsamp Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bit 7 **Reserved bit**
This bit should be programmed by 0.

Bits 6-4 **Color 1 Hsamp bits [2:0]**
These bits specify the horizontal sampling factor of Color 1. Only 1, 2 and 4 are supported.

Bit 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **Color 1 Vsamp bits [1:0]**
These bits specify the vertical sampling factor of Color 1. Only 1 and 2 are supported.

SOF Marker Color 1 QT Register **REG[9BAh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Color 1 QTable Select
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bit 7-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **Color 1 QTable Select**
This bit specify the quantization table used by Color 1.

SOF Marker Color 2 HV Register **REG[9BCh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Color 2 Hsamp Bit 2	Color 2 Hsamp Bit 1	Color 2 Hsamp Bit 0	Reserved	Reserved	Color 2 Vsamp Bit 1	Color 2 Vsamp Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bit 7 **Reserved bit**

This bit should be programmed by 0.

Bits 6-4 **Color 2 Hsamp bits [2:0]**
These bits specify the horizontal sampling factor of Color 2. Only 1, 2 and 4 are supported.

Bit 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **Color 2 Vsamp bits [1:0]**
These bits specify the vertical sampling factor of Color 2. Only 1 and 2 are supported.

SOF Marker Color 2 QT Register **REG[9BEh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Color 2 QTable Select
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bit 7-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **Color 2 QTable Select**
This bit specify the quantization table used by Color 2.

SOF Marker Color 3 HV Register **REG[9C0h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Color 3 Hsamp Bit 2	Color 3 Hsamp Bit 1	Color 3 Hsamp Bit 0	Reserved	Reserved	Color 3 Vsamp Bit 1	Color 3 Vsamp Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bit 7 **Reserved bit**
This bit should be programmed by 0.

Bits 6-4 **Color 3 Hsamp bits [2:0]**
These bits specify the horizontal sampling factor of Color 3. Only 1, 2 and 4 are supported.

Bit 3-2 **Reserved bits**
These bits should be programmed by 0.

Bits 1-0 **Color 3 Vsamp bits [1:0]**
These bits specify the vertical sampling factor of Color 3. Only 1 and 2 are supported.

SOF Marker Color 3 QT Register **REG[9C2h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Color 3 QTable Select
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bit 7-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **Color 3 QTable Select**
This bit specify the quantization table used by Color 3.

DRI Marker Register 0 **REG[9C4h]**

Bit	7	6	5	4	3	2	1	0
	DRI Marker Register 0 Bit 7	DRI Marker Register 0 Bit 6	DRI Marker Register 0 Bit 5	DRI Marker Register 0 Bit 4	DRI Marker Register 0 Bit 3	DRI Marker Register 0 Bit 2	DRI Marker Register 0 Bit 1	DRI Marker Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	NA	NA	NA	NA	NA	NA	NA	NA

Bit	7	6	5	4	3	2	1	0
state								

DRI Marker Register 1 **REG[9C6h]**

Bit	7	6	5	4	3	2	1	0
	DRI Marker Register 1 Bit 7	DRI Marker Register 1 Bit 6	DRI Marker Register 1 Bit 5	DRI Marker Register 1 Bit 4	DRI Marker Register 1 Bit 3	DRI Marker Register 1 Bit 2	DRI Marker Register 1 Bit 1	DRI Marker Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[9C4h] Bits 7-0, REG[9C6h] Bits 7-0 **DRI Marker (2 bytes)**
These registers specify the DRI marker and must be set to (hex) FF DD.

DRI Marker Size Register 0 **REG[9C8h]**

Bit	7	6	5	4	3	2	1	0
	DRI Marker Size Register 0 Bit 7	DRI Marker Size Register 0 Bit 6	DRI Marker Size Register 0 Bit 5	DRI Marker Size Register 0 Bit 4	DRI Marker Size Register 0 Bit 3	DRI Marker Size Register 0 Bit 2	DRI Marker Size Register 0 Bit 1	DRI Marker Size Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

DRI Marker Size Register 1 **REG[9CAh]**

Bit	7	6	5	4	3	2	1	0
	DRI Marker Size Register 1 Bit 7	DRI Marker Size Register 1 Bit 6	DRI Marker Size Register 1 Bit 5	DRI Marker Size Register 1 Bit 4	DRI Marker Size Register 1 Bit 3	DRI Marker Size Register 1 Bit 2	DRI Marker Size Register 1 Bit 1	DRI Marker Size Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[9C8h] Bits 7-0, REG[9CAh] Bits 7-0 **DRI Marker Size (2 bytes)**
These registers specify the DRI marker size and must be set to (hex) 00 04.

SOS Marker Register 0 **REG[9CCh]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker Register 0 Bit 7	SOS Marker Register 0 Bit 6	SOS Marker Register 0 Bit 5	SOS Marker Register 0 Bit 4	SOS Marker Register 0 Bit 3	SOS Marker Register 0 Bit 2	SOS Marker Register 0 Bit 1	SOS Marker Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

SOS Marker Register 1 **REG[9CEh]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker Register 1 Bit 7	SOS Marker Register 1 Bit 6	SOS Marker Register 1 Bit 5	SOS Marker Register 1 Bit 4	SOS Marker Register 1 Bit 3	SOS Marker Register 1 Bit 2	SOS Marker Register 1 Bit 1	SOS Marker Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	NA	NA	NA	NA	NA	NA	NA	NA

Bit	7	6	5	4	3	2	1	0
state								

REG[9CCh] Bits 7-0, REG[9CEh] Bits 7-0 **SOS Marker (2 bytes)**
These registers specify the SOS marker and must be set to (hex) FF DA.

SOS Marker Size Register 0 **REG[9D0h]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker Size Register 0 Bit 7	SOS Marker Size Register 0 Bit 6	SOS Marker Size Register 0 Bit 5	SOS Marker Size Register 0 Bit 4	SOS Marker Size Register 0 Bit 3	SOS Marker Size Register 0 Bit 2	SOS Marker Size Register 0 Bit 1	SOS Marker Size Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

SOS Marker Size Register 1 **REG[9D2h]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker Size Register 1 Bit 7	SOS Marker Size Register 1 Bit 6	SOS Marker Size Register 1 Bit 5	SOS Marker Size Register 1 Bit 4	SOS Marker Size Register 1 Bit 3	SOS Marker Size Register 1 Bit 2	SOS Marker Size Register 1 Bit 1	SOS Marker Size Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[9D0h] Bits 7-0, REG[9D2h] Bits 7-0 **SOS Marker Size (2 bytes)**
These registers specify the SOS marker size and must be set to (hex) 00 0C.

SOS Marker NUM COMP Register **REG[9D4h]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker NUM COMP Bit 7	SOS Marker NUM COMP Bit 6	SOS Marker NUM COMP Bit 5	SOS Marker NUM COMP Bit 4	SOS Marker NUM COMP Bit 3	SOS Marker NUM COMP Bit 2	SOS Marker NUM COMP Bit 1	SOS Marker NUM COMP Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOS Marker NUM COMP bits [7:0]**
These bits specify the SOS marker number of color components and must be set to (hex) 03.

SOS Marker Color 1 ID Register **REG[9D8h]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker Color 1 ID Bit 7	SOS Marker Color 1 ID Bit 6	SOS Marker Color 1 ID Bit 5	SOS Marker Color 1 ID Bit 4	SOS Marker Color 1 ID Bit 3	SOS Marker Color 1 ID Bit 2	SOS Marker Color 1 ID Bit 1	SOS Marker Color 1 ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOS Marker Color 1 ID bits [7:0]**
These bits specify the SOS marker Color 1 ID and must be set to (hex) 01.

SOS Marker Color 1 HT Register **REG[9DAh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Color 1 DC	Reserved	Reserved	Reserved	Color 1 AC

Bit	7	6	5	4	3	2	1	0
				HTable Select				HTable Select
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-5 **Reserved bits**
These bits should be programmed by 0.

Bit 4 **Color 1 DC HTable Select**
When this bit = 1, the DC Color 1 Table uses the DC Huffman Table No. 1 (REG[800-81Eh] and REG[820-836h]).
When this bit = 0, the DC Color 1 Table uses the DC Huffman Table No. 0 (REG[600-61Eh] and REG[620-636h]).

Bits 3-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **Color 1 AC HTable Select**
When this bit = 1, the AC Color 1 Table uses the AC Huffman Table No. 1 (REG[840-85Eh] and REG[860-9A2h]).
When this bit = 0, the AC Color 1 Table uses the AC Huffman Table No. 0 (REG[640-65Eh] and REG[660-7A2h]).

SOS Marker Color 2 ID Register **REG[9DCh]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker Color 2 ID Bit 7	SOS Marker Color 2 ID Bit 6	SOS Marker Color 2 ID Bit 5	SOS Marker Color 2 ID Bit 4	SOS Marker Color 2 ID Bit 3	SOS Marker Color 2 ID Bit 2	SOS Marker Color 2 ID Bit 1	SOS Marker Color 2 ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOS Marker Color 2 ID bits [7:0]**
These bits specify the SOS marker Color 2 ID and must be set to (hex) 02.

SOS Marker Color 2 HT Register **REG[9DEh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Color 2 DC HTable Select	Reserved	Reserved	Reserved	Color 2 AC HTable Select
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-5 **Reserved bits**
These bits should be programmed by 0.

Bit 4 **Color 2 DC HTable Select**
When this bit = 1, the DC Color 2 Table uses the DC Huffman Table No. 1 (REG[800-81Eh] and REG[820-836h]).
When this bit = 0, the DC Color 2 Table uses the DC Huffman Table No. 0 (REG[600-61Eh] and REG[620-636h]).

Bits 3-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **Color 2 AC HTable Select**
When this bit = 1, the AC Color 2 Table uses the AC Huffman Table No. 1 (REG[840-85Eh] and REG[860-9A2h]).
When this bit = 0, the AC Color 2 Table uses the AC Huffman Table No. 0 (REG[640-65Eh] and REG[660-7A2h]).

SOS Marker Color 3 ID Register **REG[9E0h]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker Color 3 ID Bit 7	SOS Marker Color 3 ID Bit 6	SOS Marker Color 3 ID Bit 5	SOS Marker Color 3 ID Bit 4	SOS Marker Color 3 ID Bit 3	SOS Marker Color 3 ID Bit 2	SOS Marker Color 3 ID Bit 1	SOS Marker Color 3 ID Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOS Marker Color 3 ID bits [7:0]**
These bits specify the SOS marker Color 3 ID and must be set to (hex) 03.

SOS Marker Color 3 HT Register **REG[9E2h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Color 3 DC HTable Select	Reserved	Reserved	Reserved	Color 3 AC HTable Select
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-5 **Reserved bits**
These bits should be programmed by 0.

Bit 4 **Color 3 DC HTable Select**
When this bit = 1, the DC Color 3 Table uses the DC Huffman Table No. 1 (REG[800-81Eh] and REG[820-836h]).
When this bit = 0, the DC Color 3 Table uses the DC Huffman Table No. 0 (REG[600-61Eh] and REG[620-636h]).

Bits 3-1 **Reserved bits**
These bits should be programmed by 0.

Bit 0 **Color 3 AC HTable Select**
When this bit = 1, the AC Color 3 Table uses the AC Huffman Table No. 1 (REG[840-85Eh] and REG[860-9A2h]).
When this bit = 0, the AC Color 3 Table uses the AC Huffman Table No. 0 (REG[600-61Eh] and REG[620-636h]).

SOS Marker SS Constant Register **REG[9E4h]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker SS Constant Bit 7	SOS Marker SS Constant Bit 6	SOS Marker SS Constant Bit 5	SOS Marker SS Constant Bit 4	SOS Marker SS Constant Bit 3	SOS Marker SS Constant Bit 2	SOS Marker SS Constant Bit 1	SOS Marker SS Constant Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOS Marker SS Constant bits [7:0]**
These bits specify the SOS marker start of spectral selection and must be set to (hex) 00.

SOS Marker SE Constant Register **REG[9E6h]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker SE Constant Bit 7	SOS Marker SE Constant Bit 6	SOS Marker SE Constant Bit 5	SOS Marker SE Constant Bit 4	SOS Marker SE Constant Bit 3	SOS Marker SE Constant Bit 2	SOS Marker SE Constant Bit 1	SOS Marker SE Constant Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOS Marker SE Constant bits [7:0]**
These bits specify the SOS marker end of spectral selection and must be set to (hex) 3F.

SOS Marker SA Constant Register **REG[9E8h]**

Bit	7	6	5	4	3	2	1	0
	SOS Marker SA Constant Bit 7	SOS Marker SA Constant Bit 6	SOS Marker SA Constant Bit 5	SOS Marker SA Constant Bit 4	SOS Marker SA Constant Bit 3	SOS Marker SA Constant Bit 2	SOS Marker SA Constant Bit 1	SOS Marker SA Constant Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

Bits 7-0 **SOS Marker SA Constant bits [7:0]**
These bits specify the SOS marker successive approximation bit position and must be set to (hex) 00.

EOI Marker Register 0 **REG[9EAh]**

Bit	7	6	5	4	3	2	1	0
	EOI Marker Register 0 Bit 7	EOI Marker Register 0 Bit 6	EOI Marker Register 0 Bit 5	EOI Marker Register 0 Bit 4	EOI Marker Register 0 Bit 3	EOI Marker Register 0 Bit 2	EOI Marker Register 0 Bit 1	EOI Marker Register 0 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

EOI Marker Register 1 **REG[9ECh]**

Bit	7	6	5	4	3	2	1	0
	EOI Marker Register 1 Bit 7	EOI Marker Register 1 Bit 6	EOI Marker Register 1 Bit 5	EOI Marker Register 1 Bit 4	EOI Marker Register 1 Bit 3	EOI Marker Register 1 Bit 2	EOI Marker Register 1 Bit 1	EOI Marker Register 1 Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[9EAh] Bits 7-0, REG[9ECh] Bits 7-0 **EOI Marker (2 bytes)**
These registers specify the EOI marker and must be set to (hex) FF D9.

JPEG Encode Vertical Pixel Size Register 0 **REG[9F0h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Encode Vertical Pixel Size Bit 15	JPEG Encode Vertical Pixel Size Bit 14	JPEG Encode Vertical Pixel Size Bit 13	JPEG Encode Vertical Pixel Size Bit 12	JPEG Encode Vertical Pixel Size Bit 11	JPEG Encode Vertical Pixel Size Bit 10	JPEG Encode Vertical Pixel Size Bit 9	JPEG Encode Vertical Pixel Size Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JPEG Encode Vertical Pixel Size Register 1 **REG[9F2h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Encode Vertical Pixel Size Bit 7	JPEG Encode Vertical Pixel Size Bit 6	JPEG Encode Vertical Pixel Size Bit 5	JPEG Encode Vertical Pixel Size Bit 4	JPEG Encode Vertical Pixel Size Bit 3	JPEG Encode Vertical Pixel Size Bit 2	JPEG Encode Vertical Pixel Size Bit 1	JPEG Encode Vertical Pixel Size Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[9F0h] Bits 7-0, REG[9F2h] Bits 7-0 **JPEG Encode Vertical Pixel Size bits [15:0]**
These bits specify the vertical image size before encoding takes place.

JPEG Encode Horizontal Pixel Size Register 0 **REG[9F4h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Encode Horizontal Pixel Size Bit 15	JPEG Encode Horizontal Pixel Size Bit 14	JPEG Encode Horizontal Pixel Size Bit 13	JPEG Encode Horizontal Pixel Size Bit 12	JPEG Encode Horizontal Pixel Size Bit 11	JPEG Encode Horizontal Pixel Size Bit 10	JPEG Encode Horizontal Pixel Size Bit 9	JPEG Encode Horizontal Pixel Size Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JPEG Encode Horizontal Pixel Size Register 1 **REG[9F6h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Encode Horizontal Pixel Size Bit 7	JPEG Encode Horizontal Pixel Size Bit 6	JPEG Encode Horizontal Pixel Size Bit 5	JPEG Encode Horizontal Pixel Size Bit 4	JPEG Encode Horizontal Pixel Size Bit 3	JPEG Encode Horizontal Pixel Size Bit 2	JPEG Encode Horizontal Pixel Size Bit 1	JPEG Encode Horizontal Pixel Size Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[9F4h] Bits 7-0, REG[9F6h] Bits 7-0 **JPEG Encode Horizontal Pixel Size bits [15:0]**
These bits specify the horizontal image size before encoding takes place.

JPEG Encode DRI Setting Register 0 **REG[9F8h]**

Bit	7	6	5	4	3	2	1	0
	JPEG Encode DRI Setting Bit 15	JPEG Encode DRI Setting Bit 14	JPEG Encode DRI Setting Bit 13	JPEG Encode DRI Setting Bit 12	JPEG Encode DRI Setting Bit 11	JPEG Encode DRI Setting Bit 10	JPEG Encode DRI Setting Bit 9	JPEG Encode DRI Setting Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

JPEG Encode DRI Setting Register 1 **REG[9FAh]**

Bit	7	6	5	4	3	2	1	0
	JPEG Encode DRI Setting Bit 7	JPEG Encode DRI Setting Bit 6	JPEG Encode DRI Setting Bit 5	JPEG Encode DRI Setting Bit 4	JPEG Encode DRI Setting Bit 3	JPEG Encode DRI Setting Bit 2	JPEG Encode DRI Setting Bit 1	JPEG Encode DRI Setting Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	NA	NA	NA	NA	NA	NA	NA	NA

REG[9F8h] Bits 7-0, REG[9FAh] Bits 7-0 **JPEG Encode DRI Setting bits [15:0]**
These bits determine the MCU number for RST marker insertion during encoding. DRI marker is inserted only if the value of DRI setting is greater than 0.
During decoding, these bits are ignored.

2.1.21 MMC/SD/SDIO Registers

The detailed description for registers of SDHC module (REG[1001h-11FFh]) were referred to Part A2, SD Host Controller Standard Specification [i].

The detailed description for the SDHC were referred to Part 1, Physical Layer Specification [ii].

SD_CLK Divider Register **REG[1001h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SD_CLK Divisor
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	1

REG[1001h] Bits 7-1 **Reserved bits**

REG[1001h] Bit 0 **SD_CLK Divisor bit**
This bit divides the MCLK by 2 for SD_CLK. It is enabled by default.
Bit 0 = 1 : SD_CLK = MCLK / 2
Bit 0 = 0 : SD_CLK = MCLK.
The SD_CLK output will be further divides by REG[112Dh].

DMA System Address Register 0 **REG[1100h]**

Bit	7	6	5	4	3	2	1	0
	DMA Address Bit 7	DMA Address Bit 6	DMA Address Bit 5	DMA Address Bit 4	DMA Address Bit 3	DMA Address Bit 2	DMA Address Bit 1	DMA Address Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

DMA System Address Register 1 **REG[1101h]**

Bit	7	6	5	4	3	2	1	0
	DMA Address Bit 15	DMA Address Bit 14	DMA Address Bit 13	DMA Address Bit 12	DMA Address Bit 11	DMA Address Bit 10	DMA Address Bit 9	DMA Address Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

DMA System Address Register 2 **REG[1102h]**

Bit	7	6	5	4	3	2	1	0
	DMA Address Bit 23	DMA Address Bit 22	DMA Address Bit 21	DMA Address Bit 20	DMA Address Bit 19	DMA Address Bit 18	DMA Address Bit 17	DMA Address Bit 16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

DMA System Address Register 3 **REG[1103h]**

Bit	7	6	5	4	3	2	1	0
	DMA Address Bit 31	DMA Address Bit 30	DMA Address Bit 29	DMA Address Bit 28	DMA Address Bit 27	DMA Address Bit 26	DMA Address Bit 25	DMA Address Bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1100h] Bits 7-0,
REG[1101h] Bits 7-0,
REG[1102h] Bits 7-0,
REG[1103h] Bits 7-0

DMA System Address bits [31:0]

This register contains the system memory address for a DMA transfer. Bits [31:16] are reserved and should programmed as '0'.

When the Host Controller stops a DMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a DMA transaction.

After DMA has stopped, the next system address of the next contiguous data position can be read from this register.

The DMA transfer waits at the every boundary specified by the **Host DMA Buffer Boundary** in the *Block Size* register. The Host Controller generates **DMA Interrupt** to request the Host Driver to update this register. The Host Driver set the next system address of the next data position to this register. When the most upper byte of this register (REG[1103h]) is written, the Host Controller restart the DMA transfer.

When restarting DMA by the Resume command or by setting **Continue Request** in the *Block Gap Control* register, the Host Controller shall start at the next contiguous address stored here in the *System Address* register.

Block size Register 0 **REG[1104h]**

Bit	7	6	5	4	3	2	1	0
	Transfer Block Size Bit 7	Transfer Block Size Bit 6	Transfer Block Size Bit 5	Transfer Block Size Bit 4	Transfer Block Size Bit 3	Transfer Block Size Bit 2	Transfer Block Size Bit 1	Transfer Block Size Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Block size Register1 **REG[1105h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	DMA Buffer Boundary Bit 2	DMA Buffer Boundary Bit 1	DMA Buffer Boundary Bit 0	Transfer Block Size Bit 11	Transfer Block Size Bit 10	Transfer Block Size Bit 9	Transfer Block Size Bit 8
Type	RO	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1105h] Bit 7

Reserved bit

REG[1105h] Bits 6-4

Host DMA Buffer Boundary bits [2:0]

The large contiguous memory space may not be available in the virtual memory system. To perform long DMA transfer, *System Address* register shall be updated at every system memory boundary during DMA transfer.

These bits specify the size of contiguous buffer in the system memory in terms of byte granularity. This requires the Host Controller to break the last access according to the left over count of bytes.

The DMA transfer shall wait at the every boundary specified by these fields and the Host Controller generates the **DMA Interrupt** to request the Host Driver to update the *System Address* register.

In case of this register is set to 0 (buffer size = 4K bytes), lower 12-bit of byte address points data in the contiguous buffer and the upper 20-bit points the location of the buffer in the system memory. The DMA transfer stops when the Host Controller detects carry out of the address from bit 11 to 12.

These bits shall be supported when the **DMA Support** in the *Capabilities* register is set to 1 and this function is active when the **DMA Enable** in the *Transfer Mode* register is set to 1.

000b 4K bytes (Detects A11 carry out)
001b 8K bytes (Detects A12 carry out)

010b 16K Bytes (Detects A13 carry out)
011b 32K Bytes (Detects A14 carry out)
100b 64K bytes (Detects A15 carry out)
101b 128K Bytes (Detects A16 carry out)
110b 256K Bytes (Detects A17 carry out)
111b 512K Bytes (Detects A18 carry out)

REG[1104h] Bits 7-0, REG[1105h] Bits 3-0

Transfer Block Size bits [11:0]

This register specifies the block size for block data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53 in terms of byte granularity. This requires the Host Controller to break the last access according to the left over count of bytes. Values ranging from 1 up to the maximum buffer size can be set (Refer to SD Specification [i] 'Determining Buffer block length'). It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations shall be ignored.

0800h 2048 Bytes
…
0200h 512 Bytes
01FFh 511 Bytes
…
0004h 4 Bytes
0003h 3 Bytes
0002h 2 Bytes
0001h 1 Byte
0000h No data transfer

Block Count Register 0 **REG[1106h]**

Bit	7	6	5	4	3	2	1	0
	Block Count Bit 7	Block Count Bit 6	Block Count Bit 5	Block Count Bit 4	Block Count Bit 3	Block Count Bit 2	Block Count Bit 1	Block Count Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Block Count Register 1 **REG[1107h]**

Bit	7	6	5	4	3	2	1	0
	Block Count Bit 15	Block Count Bit 14	Block Count Bit 13	Block Count Bit 12	Block Count Bit 11	Block Count Bit 10	Block Count Bit 9	Block Count Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1106h] Bits 7-0, REG[1107h] Bits 7-0

Block Count bits [15:0]

This register is enabled when **Block Count Enable** in the *Transfer Mode* register is set to 1 and is valid only for multiple block transfers. The Host Driver shall set this register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred.

This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.

When saving transfer context as a result of a Suspend command, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, the Host Driver shall restore the previously saved block count.

FFFFh 65535 blocks
… …
0002h 2 blocks

0001h 1 block
0000h Stop Count

Argument Register 0 **REG[1108h]**

Bit	7	6	5	4	3	2	1	0
	Argument Bit 7	Argument Bit 6	Argument Bit 5	Argument Bit 4	Argument Bit 3	Argument Bit 2	Argument Bit 1	Argument Bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Argument Register 1 **REG[1109h]**

Bit	7	6	5	4	3	2	1	0
	Argument Bit 15	Argument Bit 14	Argument Bit 13	Argument Bit 12	Argument Bit 11	Argument Bit 10	Argument Bit 9	Argument Bit 8
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Argument Register 2 **REG[110Ah]**

Bit	7	6	5	4	3	2	1	0
	Argument Bit 23	Argument Bit 22	Argument Bit 21	Argument Bit 20	Argument Bit 19	Argument Bit 18	Argument Bit 17	Argument Bit16
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Argument Register 3 **REG[110Bh]**

Bit	7	6	5	4	3	2	1	0
	Argument Bit 31	Argument Bit 30	Argument Bit 29	Argument Bit 28	Argument Bit 27	Argument Bit 26	Argument Bit 25	Argument Bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1108h] Bits 7-0, **Argument bits [31:0]**
REG[1109h] Bits 7-0, This register contains the SD Command Argument.
REG[110Ah] Bits 7-0, The SD Command Argument is specified as bit39-8 of Command-Format in the SD Memory
REG[110Bh] Bits 7-0 Physical Layer Specification.

Transfer Mode Register 0 **REG[110Ch]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Multi / Single Block Select	Data Transfer Direction Select	Reserved	Auto CMD12 Enable	Block Count Enable	DMA Enable
Type	RO	RO	RW	RW	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Transfer Mode Register 1 **REG[110Dh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Transfer Mode bits [15:0]
This register is used to control the operation of data transfers. The Host Driver shall set this register before issuing a command which transfers data (see **Data Present Select** in the *Command* register), or before issuing a Resume command. The Host Driver shall save the value of this register when the data transfer is suspended (as a result of a Suspend command) and restore it before issuing a Resume command. To prevent data loss, the Host Controller shall implement write protection for this register during data transactions. Writes to this register shall be ignored when the **Command Inhibit (DAT)** in the *Present State* register is 1.

REG[110Ch] Bits 7,6,3
REG[110Dh] Bits 7:0 **Reserved bits**

REG[110Ch] Bit 5 **Multi / Single Block Select**
This bit enables multiple block DATA line data transfers. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the *Block Count* register. (Refer to Table 1-45)

1 Multiple Block
0 Single Block

REG[110Ch] Bit 4 **Data Transfer Direction Select**
This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the SD Host Controller and it is set to 0 for all other commands.
1 Read (Card to Host)
0 Write (Host to Card)

REG[110Ch] Bit 2 **Auto CMD12 Enable**
Multiple block transfers for memory require CMD12 to stop the transaction. When this bit is set to 1, the Host Controller shall issue CMD12 automatically when last block transfer is completed. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop data transfer. In particular, secure commands defined in the SD Specification [File Security] do not require CMD12.
1 Enable
0 Disable

REG[110Ch] Bit 1 **Block Count Enable**
This bit is used to enable the Block Count register, which is only relevant for multiple block transfers. When this bit is 0, the *Block Count* register is disabled, which is useful in executing an infinite transfer. (Refer to Table 1-45)
1 Enable
0 Disable

REG[110Ch] Bit 0 **DMA Enable**
This bit enables DMA functionality as described in SD Specification [i] 'Supporting DMA'. DMA can be enabled only if it is supported as indicated in the **DMA Support** in the *Capabilities* register. If DMA is not supported, this bit is meaningless and shall always read 0. If this bit is set to 1, a DMA operation shall begin when the Host Driver writes to the upper byte of *Command* register REG[110Fh].
1 Enable
0 Disable

Table 2-45 shows the summary of how register settings determine types of data transfer.

Table 2-45: Determination of Transfer Type

Multi/Single Block Select	Block Count Enable	*Block Count*	Function
0	Don't care	Don't care	Single Transfer
1	0	Don't care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

Command Register 0 **REG[110Eh]**

Bit	7	6	5	4	3	2	1	0

Bit	7	6	5	4	3	2	1	0
	Command Type Bit 1	Command Type Bit 0	Data Present Select	Command Index Check Enable	Command CRC Check Enable	Reserved	Response Type Select Bit 1	Response Type Select Bit 0
Type	RW	RW	RW	RW	RW	RO	RW	RW
Reset state	0	0	0	0	0	0	0	0

Command Register 1 **REG[110Fh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Command Index Bit 5	Command Index Bit 4	Command Index Bit 3	Command Index Bit 2	Command Index Bit 1	Command Index Bit 0
Type	RO	RO	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Command bits [15:0]
The Host Driver shall check the **Command Inhibit (DAT)** bit and **Command Inhibit (CMD)** bit in the *Present State* register before writing to this register. Writing to the upper byte of this register triggers SD command generation. The Host Driver has the responsibility to write this register because the Host Controller does not protect for writing when **Command Inhibit (CMD)** is set.

REG[110Eh] Bit 2, REG[110Fh] Bits 7-6 **Reserved bits**

REG[110Fh] Bits 5-0 **Command Index bits [5:0]**
These bits shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.

REG[110Eh] Bits 7-6 **Command Type bits [1:0]**
There are three types of special commands: Suspend, Resume and Abort. These bits **shall** be set to 00b for all other commands.
(1) Suspend Command
If the Suspend command succeeds, the Host Controller shall assume the SD Bus has been released and that it is possible to issue the next command which uses the ***DAT*** line. The Host Controller shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The interrupt cycle shall start, in 4-bit mode. If the Suspend command fails, the Host Controller shall maintain its current state, and the Host Driver shall restart the transfer by setting **Continue Request** in the *Block Gap Control* register. (Refer to SD Specification [i] 'Suspend Sequence')
(2) Resume Command
The Host Driver re-starts starts the data transfer by restoring the registers in the range of 000-00Dh. (Refer to SD Specification [i] 'Suspend and Resume mechanism' for the register map.) The Host Controller shall check for busy before starting write transfers.
(3) Abort Command
If this command is set when executing a read transfer, the Host Controller shall stop reads to the buffer. If this command is set when executing a write transfer, the Host Controller shall stop driving the ***DATA*** line. After issuing the Abort command, the Host Driver should issue a software reset. (Refer to SD Specification [i] 'Abort Transaction')

11b	Abort	CMD12, CMD52 for writing "I/O Abort" in CCCR
10b	Resume	CMD52 for writing "Function Select" in CCCR
01b	Suspend	CMD52 for writing "Bus Suspend" in CCCR
00b	Normal	Other commands

REG[110Eh] Bit 5 **Data Present Select**
This bit is set to 1 to indicate that data is present and shall be transferred using the ***DATA*** line. It is set to 0 for the following:

It is set to 0 for the following:
(1) Commands using only ***CMD*** line (ex. CMD52).
(2) Commands with no data transfer but using busy signal on ***SD_DATA***[0] line (R1b or R5b ex. CMD38)
(3) Resume command
1 Data Present
0 No Data Present

REG[110Eh] Bit 4 **Command Index Check Enable**
If this bit is set to 1, the Host Controller shall check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked.
1 Enable
0 Disable

REG[110Eh] Bit 3 **Command CRC Check Enable**
If this bit is set to 1, the Host Controller shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. (Refer to D01-00 and Table 2-47 below.)
1 Enable
0 Disable

REG[110Eh] Bits 1-0 **Response Type Select bits [1:0]**

Table 2-46: Command Register

00	No Response
01	Response Length 136
10	Response Length 48
11	Response Length 48 check Busy after response

These bits determine Response types.
Note
(1) In the SDIO specification, response type notation of R5b is not defined. R5 includes R5b in the SDIO specification. But R5b is defined in this specification to specify the Host Controller shall check busy after receiving response. For example, usually CMD52 is used as R5 but I/O abort command shall be used as R5b.

Implementation Note
(1) The CRC field for R3 and R4 is expected to be all "1" bits. The CRC check should be disabled for these response types.

Table 2-47: Relation Between Parameters and the Name of Response Type

Response Type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R6, R5
11	1	1	R1b, R5b

Response Register 15-0 **REG[1110h]-[111Fh]**

Bit	7	6	5	4	3	2	1	0
	Response Bit	Response Bit	Response Bit	Response Bit	Response Bit	Response Bit	Response Bit	Response Bit
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG[1110h] Bits 7-0
to
REG[111Fh] Bits 7-0

Response bits [127:0]

This register is used to store responses from SD cards.

The Table 2-48 describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the *Response* register.

The Response Field indicates bit positions of "Responses" defined in the SD Specification [ii]. The Table 2-48 shows that most responses with a length of 48 (R[47:0]) have 32 bits of the response data (R[39:8]) stored in the *Response* register at REP[31:0]. Responses of type R1b (Auto CMD12 responses) have response data bits R[39:8] stored in the *Response* register at REP[127:96]. Responses with length 136 (R[135:0]) have 120 bits of the response data (R[127:8]) stored in the *Response* register at REP[119:0].

To be able to read the response status efficiently, the Host Controller only stores part of the response data in the *Response* register. This enables the Host Driver to efficiently read 32 bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by the Host Controller (as specified by the **Command Index Check Enable** and the **Command CRC Check Enable** bits in the *Command* Register REG[110Eh-110Fh]) and generate an error interrupt if an error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the Host Controller shall check R[47:1], and if the response length is 136 the Host Controller shall check R[119:1].

Since the Host Controller may have a multiple block data DAT line transfer executing concurrently with a CMD_wo_DAT command, the Host Controller stores the Auto CMD12 response in the upper bits (REP[127:96]) of the *Response* register. The CMD_wo_DAT response is stored in REP[31:0]. This allows the Host Controller to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa.

When the Host Controller modifies part of the *Response* register, as shown in the Table 2-48, it shall preserve the unmodified bits.

Table 2-48: Response Bit Definition for Each Response Type

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R [39:8]	REP [31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R [39:8]	REP [127:96]
R2 (CID, CSD register)	CID or CSD reg. incl.	R [127:8]	REP [119:0]
R3 (OCR register)	OCR register for memory	R [39:8]	REP [31:0]
R4 (OCR register)	OCR register for I/O etc	R [39:8]	REP [31:0]
R5,R5b	SDIO response	R [39:8]	REP [31:0]
R6 (Published RCA response)	New published RCA[31:16] etc	R [39:8]	REP [31:0]

Data Port Register 0 **REG[1120h]**

Bit	7	6	5	4	3	2	1	0
	Data Port Bit 7	Data Port Bit 6	Data Port Bit 5	Data Port Bit 4	Data Port Bit 3	Data Port Bit 2	Data Port Bit 1	Data Port Bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Data Port Register 1 **REG[1121h]**

Bit	7	6	5	4	3	2	1	0
	Data Port Bit 15	Data Port Bit 14	Data Port Bit 13	Data Port Bit 12	Data Port Bit 11	Data Port Bit 10	Data Port Bit 9	Data Port Bit 8
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
state								

Data Port Register 2 **REG[1122h]**

Bit	7	6	5	4	3	2	1	0
	Data Port Bit 23	Data Port Bit 22	Data Port Bit 21	Data Port Bit 20	Data Port Bit 19	Data Port Bit 18	Data Port Bit 17	Data Port Bit 16
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Data Port Register 3 **REG[1123h]**

Bit	7	6	5	4	3	2	1	0
	Data Port Bit 31	Data Port Bit 30	Data Port Bit 29	Data Port Bit 28	Data Port Bit 27	Data Port Bit 26	Data Port Bit 25	Data Port Bit 24
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1120h] Bits 7-0, REG[1121h] Bits 7-0, REG[1122h] Bits 7-0, REG[1123h] Bits 7-0

Data Port bits [31:0]

The Data Port Register shall be accessible via the MCU interface in 8-bit read and write modes only.

Present State Register 0 **REG[1124h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	DATA Line Active	Command Inhibit (DAT)	Command Inhibit (CMD)
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Present State Register 1 **REG[1125h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Buffer Read Enable	Buffer Write Enable	Read Transfer Active	Write Transfer Active
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Present State Register 2 **REG[1126h]**

Bit	7	6	5	4	3	2	1	0
	DATA[3:0] Line Signal Level Bit 3	DATA[3:0] Line Signal Level Bit 2	DATA[3:0] Line Signal Level Bit 1	DATA[3:0] Line Signal Level Bit 0	Write Protect Switch Pin Level	Card Detect Pin Level	Card State Stable	Card Inserted
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Present State Register 3 **REG[1127h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CMD Line Single Level
Type	RO	RO	RO	RO	RO	RO	RO	RO

Bit	7	6	5	4	3	2	1	0
Reset state	0	0	0	0	0	0	0	0

Present State bits [31:0]
The Host Driver can get status of the Host Controller from this 32-bit read only register.

REG[1124h] Bits 7-3, REG[1125h] Bits 7-4, REG[1127h] Bits 7-1 **Reserved bits**

REG[1127h] Bit 0 **CMD Line Signal Level**
This status is used to check the ***SD_CMD*** line level to recover from errors, and for debugging.

REG[1126h] Bits 7-4 **DATA[3:0] Line Signal Level**
This status is used to check the ***DATA*** line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from ***SD_DATA***[0]
Bit 7 SD_DATA[3]
Bit 6 SD_DATA[2]
Bit 5 SD_DATA[1]
Bit 4 SD_DATA[0]

REG[1126h] Bit 3 **Write Protect Switch Pin Level**
The Write Protect Switch is supported for memory and combo cards.
This bit reflects the SD_WP pin
1 Write Enabled (SD_WP = 1)
0 Write Disabled (SD_WP = 0)

REG[1126h] Bit 2 **Card Detect Pin Level**
This bit reflects the inverse value of the ***SD_CD*** pin. Debouncing is not performed on this bit. This bit may be valid when **Card State Stable** is set to 1, but it is not guaranteed because of propagation delay. Use of this bit is limited to testing since it must be debounced by software

1 Card present (SD_CD = 0)
0 No card present (SD_CD = 1)

REG[1126h] Bit 1 **Card State Stable**
This bit is used for testing. If it is 0, the **Card Detect Pin Level** is not stable. If this bit is set to 1, it means the **Card Detect Pin Level** is stable. No Card state can be detected by this bit is set to 1 and **Card Inserted** is set to 0. The **Software Reset For All** in the *Software Reset* register shall not affect this bit.
1 No Card or Inserted
0 Reset or Debouncing

REG[1126h] Bit 0 **Card Inserted**
This bit indicates whether a card has been inserted. The Host Controller shall debounce this signal so that the Host Driver will not need to wait for it to stabilize. Changing from 0 to 1 generates a **Card Insertion** interrupt in the *Normal Interrupt Status* register and changing from 1 to 0 generates a **Card Removal** interrupt in the *Normal Interrupt Status* register. The **Software Reset For All** in the *Software Reset* register shall not affect this bit.
If a card is removed while its power is on and its clock is oscillating, the Host Controller shall clear **SD Bus Power** in the *Power Control* register (REG[1129h]) and **SD Clock Enable** in the *Clock Control* register (REG[112Ch]). In addition, the Host Driver should clear the Host Controller by the **Software Reset For All** in *Software Reset* register. The card detect is active regardless of the **SD Bus Power**.
1 Card Inserted
0 Reset or Debouncing or No Card

Implementation Note: The Host Controller starts in "Reset" state at power on and changes to the "Debouncing" state once the debouncing clock is valid. In the "Debouncing" state, if the Host Controller detects that the signal (***SD_CD***) is stable during the debounce period, the state shall change to "Card Inserted" or "No Card". If the card is removed while in the "Card Inserted" state, it will immediately change to the "Debouncing" state. Since the card detect signal is then not stable, the Host Controller will change to the "Debouncing" state.

REG[1125h] Bit 3 **Buffer Read Enable**

This status is used for non-DMA read transfers.
The Host Controller may implement multiple buffers to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer status. If this bit is 1, readable data exists in the buffer. A change of this bit from 1 to 0 occurs when all the block data is read from the buffer. A change of this bit from 0 to 1 occurs when block data is ready in the buffer and generates the **Buffer Read Ready** interrupt.

1 Read enable
0 Read disable

REG[1125h] Bit 2 **Buffer Write Enable**

This status is used for non-DMA write transfers.
The Host Controller can implement multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer. A change of this bit from 1 to 0 occurs when all the block data is written to the buffer. A change of this bit from 0 to 1 occurs when top of block data can be written to the buffer and generates the **Buffer Write Ready** interrupt.

1 Write enable
0 Write disable

REG[1125h] Bit 1 **Read Transfer Active**

This status is used for detecting completion of a read transfer. Refer to SD specification [i] 'Read transaction wait / continue timing' for sequence details.
This bit is set to 1 for either of the following conditions:
(1) After the end bit of the read command.
(2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a read transfer.
This bit is cleared to 0 for either of the following conditions::
(1) When the last data block as specified by block length is transferred to the System.
(2) When all valid data blocks have been transferred to the System and no current block transfers are being sent as a result of the **Stop At Block Gap Request** being set to 1.
A Transfer Complete interrupt is generated when this bit changes to 0.

1 Transferring data
0 No valid data

REG[1125h] Bit 0 **Write Transfer Active**

This status indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the Host Controller. Refer to SD Specification [i] 'Write transaction wait / continue timing' for more details on the sequence of events.
This bit is set in either of the following cases:
(1) After the end bit of the write command.
(2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a write transfer.
This bit is cleared in either of the following cases:
(1) After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple)
(2) After getting the CRC status of any block where data transmission is about to be stopped by a **Stop At Block Gap Request**.
During a write transaction, a **Block Gap Event** interrupt is generated when this bit is changed to 0, as result of the **Stop At Block Gap Request** being set. This status is useful for the Host Driver in determining when to issue commands during write busy.

1 Transferring data
0 No valid data

REG[1124h] Bit 2 **DATA Line Active**

This bit indicates whether one of the ***DATA*** line on SD Bus is in use.
(a) In the case of read transactions
This status indicates if a read transfer is executing on the SD Bus. Changes in this value from 1 to 0 between data blocks generates a **Block Gap Event** interrupt in the *Normal Interrupt Status* register. Refer to SD Specification [i] 'Read transaction wait/continue timing' for details on timing.
This bit shall be set in either of the following cases:
(1) After the end bit of the read command.

(2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a read transfer.
This bit shall be cleared in either of the following cases:
(1) When the end bit of the last data block is sent from the SD Bus to the Host Controller.
(2) When beginning a wait read transfer at a stop at the block gap initiated by a **Stop At Block Gap Request**.
The Host Controller shall wait at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), the Host Controller can wait for current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait in order to use the suspend / resume function.
(b) In the case of write transactions
This status indicates that a write transfer is executing on the SD Bus. Changes in this value from 1 to 0 generate a **Transfer Complete** interrupt in the *Normal Interrupt Status* register. Refer to SD Specification [i] 'Write transaction wait/continue timing' for sequence details.
This bit shall be set in either of the following cases:
(1) After the end bit of the write command.
(2) When writing to 1 to **Continue Request** in the *Block Gap Control* register to continue a write transfer.
This bit shall be cleared in either of the following cases:
(1) When the SD card releases write busy of the last data block the Host Controller shall also detect if output is not busy. If SD card does not drive busy signal for 8
SD Clocks, the Host Controller shall consider the card drive "Not Busy".
(2) When the SD card releases write busy prior to waiting for write transfer as a result of a **Stop At Block Gap Request**.
1 DATA Line Active
0 DATA Line Inactive

REG[1124h] Bit 1

Command Inhibit (DAT)
This status bit is generated if either the **DATA Line Active** or the **Read Transfer Active** is set to 1. If this bit is 0, it indicates the Host Controller can issue the next SD Command. Commands with busy signal belong to **Command Inhibit (DAT)** (ex. R1b, R5b type). Changing from 1 to 0 generates a **Transfer Complete** interrupt in the *Normal Interrupt Status* register.
Note
(1) The SD Host Driver can save registers in the range of 000-00Dh for a suspend transaction after this bit has changed from 1 to 0.
1 Cannot issue command which uses the ***DATA*** line
0 Can issue command which uses the ***DATA*** line

REG[1124h] Bit 0

Command Inhibit (CMD)
If this bit is 0, it indicates the ***CMD*** line is not in use and the Host Controller can issue a SD Command using the ***CMD*** line.
This bit is set immediately after the *Command* register (00Fh) is written. This bit is cleared when the command response is received. Even if the **Command Inhibit (DAT)** is set to 1, Commands using only the ***CMD*** line can be issued if this bit is 0. Changing from 1 to 0 generates a **Command Complete** interrupt in the *Normal Interrupt Status* register. If the Host Controller cannot issue the command because of a command conflict error (Refer to **Command CRC Error** (REG[1132h]) or because of **Command Not Issued By Auto CMD12 Error** (REG[113Ch]), this bit shall remain 1 and the **Command Complete** is not set.
Status issuing Auto CMD12 is not read from this bit.
1 Cannot issue command
0 Can issue command using only ***CMD*** line

Implementation Note
(1) The Host Driver can issue CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the **DATA** lines are busy during data transfer. These commands can be issued when **Command Inhibit (CMD)** is set to zero.
Other commands shall be issued when **Command Inhibit (DAT)** is set to zero. Possible changes to the SD

Physical Specification may add other commands to this list in the future.

Implementation Note
(1) Some fields defined in the Present State Register change values asynchronous to the system clock. The System reads these statuses through the System Bus Interface and it may require data stable period during bus cycle. The Host Controller should sample and hold values during reads from this register according to the timing required by the System Bus Interface specification.

Figure 2-38 to **Error! Reference source not found.** shows the timing of setting and clearing the **Command Inhibit (DAT)** and the **Command Inhibit (CMD)**.

Figure 2-38: Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with data transfer

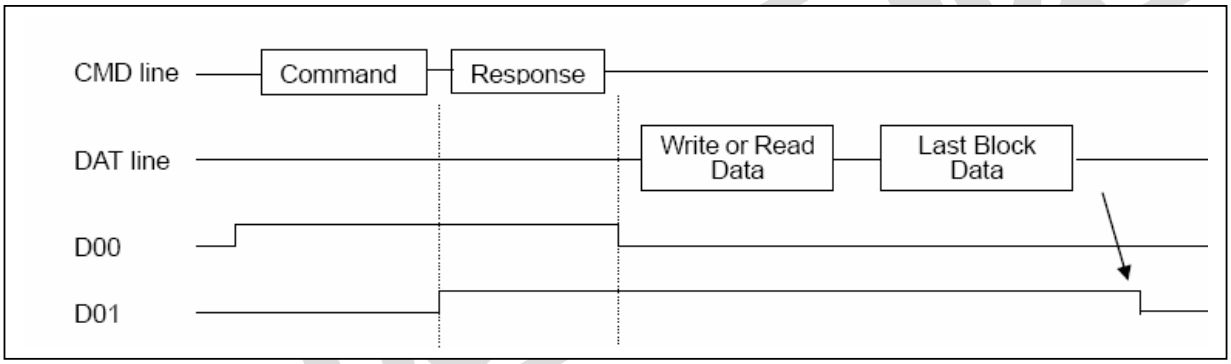


Figure 2-39: Timing of Command Inhibit (DAT) for the case of response with busy

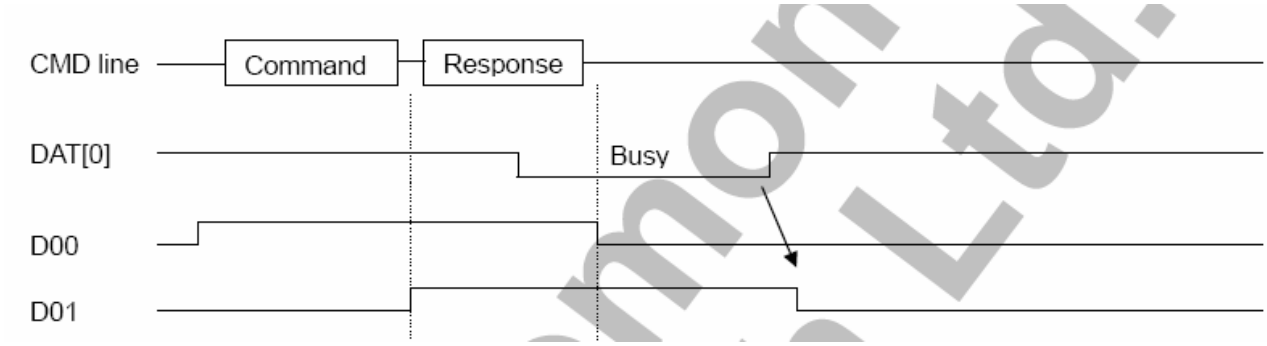
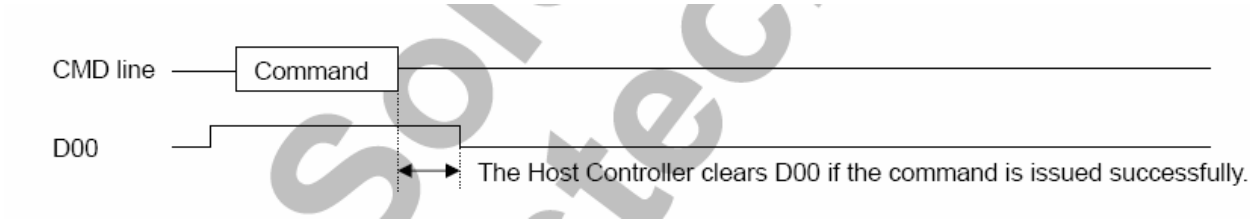


Figure 2-40: Timing of Command Inhibit (CMD) for the case of no response command



Host Control Register **REG[1128h]**

Bit	7	6	5	4	3	2	1	0
	Reserved Bit	Reserved Bit	Reserved Bit	Reserved Bit	Reserved Bit	Reserved Bit	Data Transfer Width	LED Control
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1128h] Bits 7-2 **Reserved bits**

REG[1128h] Bit 1 **Data Transfer Width bit**
This bit selects the data width of the Host Controller. The Host Driver shall set it to match the data width of the SD card.
1 4 bit mode
0 1 bit mode

REG[1128h] Bit 0 **LED Control bit**
This bit is used to caution the user not to remove the card while the SD card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction.
If this bit = 1, LED on
If this bit = 0, LED off

Power Control Register **REG[1129h]**

Bit	7	6	5	4	3	2	1	0
	Reserved Bit	Reserved Bit	Reserved Bit	Reserved Bit	SD Bus Voltage Select Bit 2	SD Bus Voltage Select Bit 1	SD Bus Voltage Select Bit 0	SD Bus Power Bit
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[1129h] Bits 7-4 **Reserved bits**

REG[1129h] Bits 3-1 **SD Bus Voltage bits [2:0]**
By setting these bits, the Host Driver selects the voltage level for the SD card. Before setting this register, the Host Driver shall check the **Voltage Support** bits in the *Capabilities* register. If an unsupported voltage is selected, the Host System shall not supply SD Bus voltage.

SD Bus Voltage bits [2:0]	SD Bus Voltage
111	3.3V
110-000	Reserved

REG[1129h] Bit 0 **SD Bus Power bit**
Before setting this bit, the SD Host Driver shall set **SD Bus Voltage Select**. If the Host Controller detects the No Card state, this bit shall be cleared.
If this bit = 1, Power on
If this bit = 0, Power off

Implementation Note
(1) Basically, the Host Driver has responsibility to supply SD Bus voltage by **SD Bus Power**, according to SD card OCR and supply voltage capabilities depend on the Host System.
If the Host Driver selects an unsupported voltage in the **SD Bus Voltage Select** field, the Host Controller may ignore writes to SD Bus Power and keep its value at zero.

Implementation Note
(1) The Host System shall not supply SD Bus power when **SD Bus Power** is set to 0 and can supply SD Bus power when **SD Bus Power** is set to 1 depending on the system conditions (ex. Left of the battery).

Reserved Register **REG[112Ah]**

Bit	7	6	5	4	3	2	1	0
	Reserved Bit	Reserved Bit	Reserved Bit	Reserved Bit	Reserved Bit	Reserved Bit	Reserved Bit	Reserved Bit
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[112Ah] Bits 7-0 **Reserved bits**

Wake Up Control Register **REG[112Bh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Wake Event Enable on SD Card Removal	Wake Event Enable on SD Card Insertion	Wake Event Enable on SD Card Interrupt
Type	RO	RO	RO	RO	RO	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[112Bh] Bits 7-3 **Reserved bit**

REG[112Bh] Bit 2 **Wake Event Enable on SD Card Removal bit**
This bit enables wakeup event via Card Removal assertion in the Normal Interrupt Status register.
1 Enable
0 Disable

REG[112Bh] Bit 1 Wake Event Enable on SD Card Insertion bit
This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register.
1 Enable
0 Disable

REG[112Bh] Bit 0 **Wake Event Enable on SD Card interrupt bit**
This bit enables wakeup event via Card Interrupt assertion in the Normal Interrupt Status register.
1 Enable
0 Disable

Clock Control Register 0 **REG[112Ch]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	SD Clock Enable	Internal Clock Stable	Internal Clock enable
Type	RO	RO	RO	RO	RO	RW	RO	RW
Reset state	0	0	0	0	0	0	0	0

Clock Control Register 1 **REG[112Dh]**

Bit	7	6	5	4	3	2	1	0
	SDCLK Frequency Select bit 7	SDCLK Frequency Select bit 6	SDCLK Frequency Select bit 5	SDCLK Frequency Select bit 4	SDCLK Frequency Select bit 3	SDCLK Frequency Select bit 2	SDCLK Frequency Select bit 1	SDCLK Frequency Select bit 0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[112Dh] Bits 7-0 **SDCLK Frequency Select bits [7:0]**

At the initialization of the Host Controller, the Host Driver shall set the **SDCLK Frequency Select**.

This register is used to select the frequency of ***SD_CLK*** pin. The frequency is not programmed directly; rather this register holds the divisor of the **MCLK Frequency**. Only the following settings are allowed.

80h MCLK divided by 256
40h MCLK divided by 128
20h MCLK divided by 64
10h MCLK divided by 32
08h MCLK divided by 16
04h MCLK divided by 8
02h MCLK divided by 4
01h MCLK divided by 2
00h MCLK (10MHz-63MHz)

Setting 00h specifies the highest frequency of the SD Clock. When setting multiple bits, the most significant bit is used as the divisor.

According to the SD Physical Specification Version 1.01 and the SDIO Card Specification Version 1.0, maximum SD Clock frequency is 25MHz, and shall never exceed this limit.

Note : The SD_CLK will default divided by 2 with REG[1001h] bit 0 = 1.

The frequency of SDCLK is set by the following formula:

SDCLK Frequency = MCLK / divisor

Thus, choose the smallest possible divisor which results in a clock frequency that is less than or equal to the target frequency.

For example, if the **MCLK** has the value 33MHz, and the target frequency is 25MHz, then choosing the divisor value of 01h will yield 16.5MHz (if REG[1001h] bit 0 = 0), which is the nearest frequency less than or equal to the target.

REG[112Ch] Bits 7-3 **Reserved bits**

REG[112Ch] Bit 2 **SD Clock Enable**

The Host Controller shall stop ***SD_CLK*** when writing this bit to 0. **SDCLK Frequency Select** can be changed when this bit is 0. Then, the Host Controller shall maintain the same clock frequency until ***SDCLK*** is stopped (Stop at ***SD_CLK***=0). If the Host Controller detects the No Card state, this bit shall be cleared.

1 Enable
0 Disable

REG[112Ch] Bit 1 **Internal Clock Stable**

This bit is set to 1 when SD Clock is stable after writing to **Internal Clock Enable** in this register to 1. The SD Host Driver shall wait to set **SD Clock Enable** until this bit is set to 1.

Note: This is useful when using PLL for a clock oscillator that requires setup time.

1 Ready
0 Not Ready

REG[112Ch] Bit 0 **Internal Clock Enable**

This bit is set to 0 when the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller should stop its internal clock to go very low power state. Still, registers shall be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller shall set **Internal Clock Stable** in this register to 1. This bit shall not affect card detection.

1 Oscillate
0 Stop

Timeout Control Register **REG[112Eh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Timeout Control Bit 3	Timeout Control Bit 2	Timeout Control Bit 1	Timeout Control Bit 0

Bit	7	6	5	4	3	2	1	0
Type	RO	RO	RO	RO	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

REG[112Eh] Bits 7-4 **Reserved bits**
These bits should be programmed by 0.

REG[112Eh] Bits 3-0 **Timeout Control bits [3:0]**
At the initialization of the Host Controller, the Host Driver shall set the **Data Timeout Counter Value** according to the *Capabilities* register.
This value determines the interval by which DATA line timeouts are detected. Refer to the **Data Timeout Error** in the *Error Interrupt Status* register for information on factors that dictate timeout generation. Timeout clock frequency will be generated by dividing the base clock MCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the **Data Timeout Error Status Enable** (in the *Error Interrupt Status Enable* register)
1111 Reserved
1110 MCLK x 2^{27}
…..
0001 MCLK x 2^{14}
0000 MCLK x 2^{13}

Software Reset Register **REG[112Fh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Software Reset for DATA Line	Software Reset for CMD Line	Software Reset for all
Type	RO	RO	RO	RO	RO	RWAC	RWAC	RWAC
Reset state	0	0	0	0	0	0	0	0

REG[112Fh] Bits 7-3 **Reserved bits**

REG[112Fh] Bit 2 **Software reset bits**
A reset pulse is generated when writing 1 to each bit of this register. After completing the reset, the Host Controller shall clear each bit. Because it takes some time to complete software reset, the SD Host Driver shall confirm that these bits are 0.
Software Reset for DATA Line
Only part of data circuit is reset. DMA circuit is also reset.
The following registers and bits are cleared by this bit:
Buffer Data Port register
Buffer is cleared and initialized.
Present State register
Buffer Read Enable
Buffer Write Enable
Read Transfer Active
Write Transfer Active
DATA Line Active
Command Inhibit (DAT)
Block Gap Control register
Continue Request
Stop At Block Gap Request
Normal Interrupt Status register
Buffer Read Ready
Buffer Write Ready
DMA Interrupt
Block Gap Event

Transfer Complete

1 Reset
0 Work

REG[112Fh] Bit 1 **Software Reset For CMD Line**

Only part of command circuit is reset.
The following registers and bits are cleared by this bit:
Present State register
Command Inhibit (CMD)
Normal Interrupt Status register
Command Complete

1 Reset
0 Work

REG[112Fh] Bit 0 **Software Reset For All**

This reset affects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared to 0. During its initialization, the Host Driver shall set this bit to 1 to reset the Host Controller. The Host Controller shall reset this bit to 0 when capabilities registers are valid and the Host Driver can read them.
Additional use of **Software Reset For All may** not affect the value of the C*apabilities* registers. If this bit is set to 1, the SD card shall reset itself and must be reinitialized by the Host Driver.

1 Reset
0 Work

Normal Interrupt Status Register 0 **REG[1130h]**

Bit	7	6	5	4	3	2	1	0
	Card Removal	Card Insertion	Buffer Read Ready	Buffer Write Ready	DMA Interrupt	Block Gap Event	Transfer Complete	Command Complete
Type	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
Reset state	0	0	0	0	0	0	0	0

Normal Interrupt Status Register 1 **REG[1131h]**

Bit	7	6	5	4	3	2	1	0
	Error Interrupt	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Card Interrupt
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Normal Interrupt Status bits [15:0]

The *Normal Interrupt Status Enable* affects reads of this register, but *Normal Interrupt Signal Enable* does not affect these reads. An interrupt is generated when the Normal Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. For all bits except **Card Interrupt** and **Error Interrupt**, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. The **Card Interrupt** is cleared when the card stops asserting the interrupt; that is, when the Card Driver services the interrupt condition.

REG[1131h] Bit 7 **Error Interrupt**

If any of the bits in the *Error Interrupt Status* register are set, then this bit is set. Therefore the Host Driver can efficiently test for an error by checking this bit first. This bit is read only.

1 Error
0 No Error

REG[1131h] Bits 6-1 **Reserved bits**

REG[1131h] Bit 0 **Card Interrupt**

Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the Host Controller shall detect the **Card Interrupt** without SD Clock to support

wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay.
When this status has been set and the Host Driver needs to start this interrupt service, **Card Interrupt Status Enable** in the *Normal Interrupt Status Enable* register shall be set to 0 in order to clear the card interrupt statuses latched in the Host Controller and to stop driving the interrupt signal to the Host System. After completion of the card interrupt service (It should reset interrupt factors in the SD card and the interrupt signal may not be asserted), set **Card Interrupt Status Enable** to 1 and start sampling the interrupt signal again.
1 Generate Card Interrupt
0 No Card Interrupt

REG[1130h] Bit 7 **Card Removal**
This status is set if the **Card Inserted** in the *Present State* register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated
1 Card removed
0 Card state stable or Debouncing

REG[1130h] Bit 6 **Card Insertion**
This status is set if the **Card Inserted** in the *Present State* register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated.
1 Card inserted
0 Card state stable or Debouncing

REG[1130h] Bit 5 **Buffer Read Ready**
This status is set if the **Buffer Read Enable** changes from 0 to 1. Refer to the **Buffer Read Enable** in the *Present State* register.
1 Ready to read buffer
0 Not ready to read buffer

REG[1130h] Bit 4 **Buffer Write Ready**
This status is set if the **Buffer Write Enable** changes from 0 to 1. Refer to the **Buffer Write Enable** in the *Present State* register.
1 Ready to write buffer
0 Not ready to write buffer

REG[1130h] Bit 3 **DMA Interrupt**
This status is set if the Host Controller detects the Host DMA Buffer boundary during transfer. Refer to the **Host DMA Buffer Boundary** in the *Block Size* register.
Other DMA interrupt factors may be added in the future. This interrupt shall not be generated after the **Transfer Complete**.
1 DMA Interrupt is generated
0 No DMA Interrupt

REG[1130h] Bit 2 **Block Gap Event**
If the **Stop At Block Gap Request** in the *Block Gap Control* register is set, this bit is set when both a read / write transaction is stopped at a block gap. If **Stop At Block Gap Request** is not set to 1, this bit is not set to 1.
(1) In the case of a Read Transaction
This bit is set at the falling edge of the **DATA Line Active** Status (When the transaction is stopped at SD Bus timing. The Read Wait must be supported in order to use this function. Refer to SD Specification [i] 'Read transaction wait / continue timing' about the detail timing.
(2) Case of Write Transaction
This bit is set at the falling edge of **Write Transfer Active** Status (After getting CRC status at SD Bus timing). Refer to SD Specification [i] 'Write transaction wait / continue timing' for more details on the sequence of events.

1 Transaction stopped at block gap
0 No Block Gap Event

REG[1130h] Bit 1

Transfer Complete
This bit is set when a read / write transfer is completed.
(1) In the case of a Read Transaction
This bit is set at the falling edge of **Read Transfer Active** Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by data length (After the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the **Stop At Block Gap Request** in the *Block Gap Control* register (After valid data has been read to the Host System). Refer to Section 3.10.3 of [i] for more details on the sequence of events.
(2) In the case of a Write Transaction
This bit is set at the falling edge of the **DATA Line Active** Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by data length and the busy signal released. The second is when data transfers are stopped at the block gap by setting **Stop At Block Gap Request** in the *Block Gap Control* register and data transfers completed. (After valid data is written to the SD card and the busy signal released). Refer to SD Specification [i] 'Write transaction wait / continue timing' for more details on the sequence of events.
The table below shows that **Transfer Complete** has higher priority than **Data Timeout Error**. If both bits are set to 1, the data transfer can be considered complete.

Relation between Transfer Complete and Data Timeout Error

Transfer Complete	**Data Timeout Error**	Meaning of the status
0	0	Interrupted by another factor
0	1	Timeout occur during transfer
1	Don't Care	Data transfer complete

1 Data transfer complete
0 No transfer complete

REG[1130h] Bit 0

Command Complete
This bit is set when get the end bit of the command response. (Except Auto CMD12)
Refer to **Command Inhibit (CMD)** in the *Present State* register.

The table below shows that **Command Timeout Error** has higher priority than **Command Complete**. If both bits are set to 1, it can be considered that the response was not received correctly.

Command Complete	**Command Timeout Error**	**Meanin of the status**
0	0	Interrupted by another factor
Don't Care	1	Response not received within 64 SDCLK cycles.
1	0	Response received

1 Command complete
0 No command complete

Error Interrupt Status Register 0 **REG[1132h]**

Bit	7	6	5	4	3	2	1	0
	Current limit Error	Data End Bit Error	Data CRC Error	Data Timeout Error	Command Index Error	Command End bit Error	Command CRC Error	Command Timeout Error
Type	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
Reset state	0	0	0	0	0	0	0	0

Error Interrupt Status Register 1 **REG[1133h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Block Size Bit	Data Line Conflict Bit	Response Direction Bit Error	Reserved	Reserved	Reserved	Auto CMD12 Error
Type	RO	RW1C	RW1C	RW1C	RO	RO	RO	RW1C
Reset state	0	0	0	0	0	0	0	0

Signals defined in this register can be enabled by the Error Interrupt Status Enable register, but not by the Error Interrupt Signal Enable register. The interrupt is generated when the Error Interrupt Signal Enable is enabled and at least one of the statuses is set to 1. Writing to 1 clears the bit and writing to 0 keeps the bit unchanged. More than one status can be cleared at the one register write.

REG[1133h] Bits 7, 3-1 **Reserved bit**

REG[1133h] Bit 6 **Block Size Error Bit**
This bit indicates that a Block Size limit error was detected. A Block Size limit error is detected when the programmed Block Size for a read or write transaction is set to 0 or greater than 2048. This bit is not set when the corresponding status enable bit is not enabled.

REG[1133h] Bit 5 **Data Line Conflict**
This bit indicates that a DATA Line Conflict was detected on SD_DATA0 or additionally on SD_DATA1-3 when the 4-bit Transfer Mode is selected. This bit is not set when the corresponding status enable bit is not enabled.

REG[1133h] Bit 4 **Response Direction Bit Error**
This bit indicates that the Card-to-Host Transmitter bit was not set to '0'. This bit is not set when the corresponding status enable bit is not enabled.

REG[1133h] Bit 0 **Auto CMD12 Error**
Occurs when detecting that one of the bits in *Auto CMD12 Error Status* register has changed from 0 to 1. This bit is set to 1,not only when the errors in Auto CMD12 occur but also when Auto CMD12 is not executed due to the previous command error.
1 Error
0 No Error

REG[1132h] Bit 7 **Current Limit Error**
By setting the **SD Bus Power** bit in the *Power Control* register, the Host Controller is requested to supply power for the SD Bus. If the Host Controller supports the Current Limit function, it can be protected from an illegal card by stopping power supply to the card in which case this bit indicates a failure status. Reading 1 means the Host Controller is not supplying power to SD card due to some failure. Reading 0 means that the Host Controller is supplying power and no error has occurred. The Host Controller may require some sampling time to detect the current limit. If the Host Controller does not support this function, this bit shall always be set to 0.
1 Power failed
0 No Error

REG[1132h] Bit 6 **Data End Bit Error**
Occurs either when detecting 0 at the end bit position of read data which uses the ***DATA*** line or at the end bit position of the CRC Status.
1 Error
0 No Error

REG[1132h] Bit 5 **Data CRC Error**
Occurs when detecting CRC error when transferring read data which uses the ***DATA*** line or when detecting the Write CRC status having a value of other than "010".
1 Error
0 No Error

REG[1132h] Bit 4 **Data Timeout Error**
Occurs when detecting one of following timeout conditions.
(1) Busy timeout for R1b,R5b type
(2) Busy timeout after Write CRC status
(3) Write CRC Status timeout

(4) Read Data timeout.
1 Time out
0 No Error

REG[1132h] Bit 3 **Command Index Error**
Occurs if a Command Index error occurs in the command response.
1 Error
0 No Error

REG[1132h] Bit 2 **Command End Bit Error**
Occurs when detecting that the end bit of a command response is 0.
1 End Bit Error Generated
0 No Error

REG[1132h] Bit 1 **Command CRC Error**
Command CRC Error is generated in two cases.
(1) If a response is returned and the **Command Timeout Error** is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response.
(2) The Host Controller detects a ***CMD*** line conflict by monitoring the ***CMD*** line when a command is issued. If the Host Controller drives the ***CMD*** line to 1 level, but detects 0 level on the ***CMD*** line at the next SDCLK edge, then the Host Controller shall abort the command (Stop driving ***CMD*** line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish ***CMD*** line conflict (Refer to Table 1-50).
1 CRC Error Generated
0 No Error

REG[1132h] Bit 0 **Command Timeout Error**
Occurs only if no response is returned within 64 ***SDCLK*** cycles from the end bit of the command. If the Host Controller detects a ***CMD*** line conflict, in which case **Command CRC Error shall** also be set as shown in Table 2-49, this bit **shall** be set without waiting for 64 ***SDCLK*** cycles because the command will be aborted by the Host Controller.
1 Time out
0 No Error

The relation between **Command CRC Error** and **Command Timeout Error** is shown in Table 1-50.

Table 2-49: The relation between Command CRC Error and Command Timeout Error

Command CRC Error	Command Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	***CMD*** line conflict

Normal Interrupt Status Enable Register 0 **REG[1134h]**

Bit	7	6	5	4	3	2	1	0
	Card Removal Status Enable	Card Insertion Status Enable	Buffer Read Ready Status Enable	Buffer Write Ready Status Enable	DMA Interrupt Status Enable	Block Gap Event Status Enable	Transfer Complete Status Enable	Command Complete Status Enable
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Normal Interrupt Status Enable Register 1 **REG[1135h]**

Bit	7	6	5	4	3	2	1	0
	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Card Interrupt Status Enable
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
state								

REG[1135h] Bits 7-1 **This bit should be readback as 0.**

REG[1135h] Bits 6-1 **Reserved bit**

REG[1135h] Bit 0 **Card Interrupt Status Enable**
If this bit is set to 0, the Host Controller shall clear interrupt request to the System. The **Card Interrupt** detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver should clear the **Card Interrupt Status Enable** before servicing the **Card Interrupt** and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.
1 Enabled
0 Masked

REG[1134h] Bit 7 **Card Removal Status Enable**
1 Enabled
0 Masked

REG[1134h] Bit 6 **Card Insertion Status Enable**
1 Enabled
0 Masked

REG[1134h] Bit 5 **Buffer Read Ready Status Enable**
1 Enabled
0 Masked

REG[1134h] Bit 4 **Buffer Write Ready Status Enable**
1 Enabled
0 Masked

REG[1134h] Bit 3 **DMA Interrupt Status Enable**
1 Enabled
0 Masked

REG[1134h] Bit 2 **Block Gap Event Status Enable**
1 Enabled
0 Masked

REG[1134h] Bit 1 **Transfer Complete Status Enable**
1 Enabled
0 Masked

REG[1134h] Bit 0 **Command Complete Status Enable**
1 Enabled
0 Masked

Implementation Note
(1) The Host Controller may sample the card interrupt signal during interrupt period and may hold its value in the flip-flop. If the **Card Interrupt Status Enable** is set to 0, the Host Controller shall clear all internal signals regarding **Card Interrupt**.

Error Interrupt Status Enable Register 0 **REG[1136h]**

Bit	7	6	5	4	3	2	1	0
	Current Limit Error Status Enable	Data End Bit Error Status Enable	Data CRC Error Status Enable	Data Timeout Error Status Enable	Command Index Error Status Enable	Command End Bit Error Status Enable	Command CRC Error Status Enable	Command Timeout Error Status Enable
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Error Interrupt Status Enable Register 1 **REG[1137h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Auto

Bit	7	6	5	4	3	2	1	0
								CMD12 Error Status Enable
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

REG[1137h] Bits 7-1 **Reserved bits**

REG[1137h] Bit 0 **Auto CMDI2 Error Status Enable**
1 Enabled
0 Masked

REG[1136h] Bit 7 **Current Limit Error Status Enable**
1 Enabled
0 Masked

REG[1136h] Bit 6 **Data End Bit Error Status Enable**
1 Enabled
0 Masked

REG[1136h] Bit 5 **Data CRC Error Status Enable**
1 Enabled
o Masked

REG[1136h] Bit 4 **Data Timeout Error Status Enable**
1 Enabled
0 Masked

REG[1136h] Bit 3 **Command Index Error Status Enable**
1 Enabled
0 Masked

REG[1136h] Bit 2 **Command End Bit Error Status Enable**
1 Enabled
0 Masked

REG[1136h] Bit 1 **Command CRC Error Status Enable**
1 Enabled
0 Masked

REG[1136h] Bit 0 **Command Timeout Error Status Enable**
1 Enabled
0 Masked

Implementation Note
(1) To detect SD_CMD line conflict, the Host Driver must set Status Enable and Command CRC Error Status Enable to 1.

Normal Interrupt Signal Enable Register 0 **REG[1138h]**

Bit	7	6	5	4	3	2	1	0
	Card Removal Signal Enable	Card Insertion Signal Enable	Buffer Read Ready Signal Enable	Buffer Write Ready Signal Enable	DMA Interrupt Signal Enable	Block Gap Event Signal Enable	Transfer Complete Signal Enable	Command Complete Signal Enable
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Normal Interrupt Signal Enable Register 1 **REG[1139h]**

Bit	7	6	5	4	3	2	1	0
	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Card Interrupt Signal Enable

Bit	7	6	5	4	3	2	1	0
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits all share the same bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

REG[1139h] Bit 7 **This bit should be readback as 0.**

REG[1139h] Bit 6-1 **Reserved bits**

REG[1139h] Bit 0 **Card Interrupt Signal Enable**
1 Enabled
0 Masked

REG[1138h] Bit 7 **Card Removal Signal Enable**
1 Enabled
0 Masked

REG[1138h] Bit 6 **Card Insertion Signal Enable**
1 Enabled
0 Masked

REG[1138h] Bit 5 **Buffer Read Ready Signal Enable**
1 Enabled
0 Masked

REG[1138h] Bit 4 **Buffer Write Ready Signal Enable**
1 Enabled
0 Masked

REG[1138h] Bit 3 **DMA Interrupt Signal Enable**
1 Enabled
0 Masked

REG[1138h] Bit 2 **Block Gap Event Signal Enable**
1 Enabled
0 Masked

REG[1138h] Bit 1 **Transfer Complete Signal Enable**
1 Enabled
0 Masked

REG[1138h] Bit 0 **Command Complete Signal Enable**
1 Enabled
0 Masked

Error Interrupt Signal Enable Register 0 **REG[113Ah]**

Bit	7	6	5	4	3	2	1	0
	Current Limit Error Signal Enable	Data End Bit Error Signal Enable	Data CRC Error Signal Enable	Data Timeout Error Signal Enable	Command Index Error Signal Enable	Command End Bit Error Signal Enable	Command CRC Error Signal Enable	Command Timeout Error Signal Enable
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset state	0	0	0	0	0	0	0	0

Error Interrupt Signal Enable Register 1 **REG[113Bh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Auto CMD12 Error

Bit	7	6	5	4	3	2	1	0
								Signal Enable
Type	RO	RO	RO	RO	RO	RO	RO	RW
Reset state	0	0	0	0	0	0	0	0

This register is used to select which interrupt status is notified to the Host System as the interrupt. These status bits all share the same 1 bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

REG[113Bh] Bits 7-1 **Reserved bits**

REG[113Bh] Bit 0 **Auto CMDI2 Error Signal Enable**
1 Enabled
0 Masked

REG[113Ah] Bit 7 **Current Limit Error Signal Enable**
1 Enabled
0 Masked

REG[113Ah] Bit 6 **Data End Bit Error Signal Enable**
1 Enabled
0 Masked

REG[113Ah] Bit 5 **Data CRC Error Signal Enable**
1 Enabled
0 Masked

REG[113Ah] Bit 4 **Data Timeout Error Signal Enable**
1 Enabled
0 Masked

REG[113Ah] Bit 3 **Command Index Error Signal Enable**
1 Enabled
0 Masked

REG[113Ah] Bit 2 **Command End Bit Error Signal Enable**
1 Enabled
0 Masked

REG[113Ah] Bit 1 **Command CRC Error Signal Enable**
1 Enabled
0 Masked

REG[113Ah] Bit 0 **Command Timeout Error Signal Enable**
1 Enabled
0 Masked

Auto CMD12 Error Status Register 0 **REG[113Ch]**

Bit	7	6	5	4	3	2	1	0
	Command Not Issued by Auto CMD12 Error	Reserved	Reserved	Auto CMD12 Index Error	Auto CMD12 End Bit Error	Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Auto CMD12 not executed
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Auto CMD12 Error Status Register 1 **REG[113Dh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

When Auto CMDI2 Error Status is set, the Host Driver shall check this register to identify what kind of errorAuto CMD12 indicated. This register is valid only when the Auto CMDI2 Error is set.

REG[113Dh] Bits 7-0 **Reserved bits**

REG[113Ch] Bit 7 **Command Not Issued By Auto CMDI2 Error**
Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register.
1 Not Issued
0 No error

REG[113Ch] Bits 6-5 **Reserved bits**

REG[113Ch] Bit 4 **Auto CMDI2 Index Error**
Occurs if the Command Index error occurs in response to a command.
1 Error
0 No error

REG[113Ch] Bit 3 **Auto CMDI2 End Bit Error**
Occurs when detecting that the end bit of command response is 0.
1 End Bit Error Generated
0 No error

REG[113Ch] Bit 2 **Auto CMDI2 CRC Error**
Occurs when detecting a CRC error in the command response.
1 CRC Error Generated
0 No error

REG[113Ch] Bit 1 **Auto CMDI2 Timeout Error**
Occurs if no response is returned within 64 SDCLK cycles from the end bit of command. If this bit is set to 1, the other error status bits (D04-D02) are meaningless.
1 Time out
0 No error

REG[113Ch] Bit 0 **Auto CMDI2 Not Executed**
If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless.
1 Not executed
0 Executed

The relation between Auto CMDI2 CRC Error and Auto CMDI2 Timeout Error is shown in Table 2-50.

Table 2-50: The relation between Command CRC Error and Command Timeout Error for Auto CMDI2

Auto CMDI2 CRC Error	Auto CMDI2 Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

The timing of changing Auto CMDI2 Error Status can be classified in three scenarios:
(1) When the Host Controller is going to issue Auto CMD12
Set D00 to 1 if Auto CMD12 cannot be issued due to an error in the previous command.
Set D00 to 0 if Auto CMD12 is issued.
(2) At the end bit of an Auto CMD12 response
Check received responses by checking the error bits D0l, D02, D03 and D04.
Set to 1 if error is detected.
Set to 0 if error is not detected.
(3) Before reading the Auto CMD12 Error Status bit D07
Set D07 to 1 if there is a command cannot be issued
Set D07 to 0 if there is no command to issue
Timing of generating the **Auto CMDI2 Error** and writing to the Command register are asynchronous. Then D07 shall be sampled when driver never writing to the Command register. So just before reading the Auto

CMDI2 Error Status register is good timing to set the D07 status bit.
An Auto CMD12 Error Interrupt is generated when one of the error bits D00 to D04 is set to 1. The **Command Not Issued By Auto CMDI2 Error** does not generate an interrupt.

Reserved Register **REG[113Eh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Reserved Register **REG[113Fh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG[113Eh] Bits 7-0, REG[113Fh] Bits 7-0 **Reserved bits**

Capabilities Register 0 **REG[1140h]**

Bit	7	6	5	4	3	2	1	0
	Timeout clock unit	Reserved	Timeout clock frequency 5	Timeout clock frequency 4	Timeout clock frequency 3	Timeout clock frequency 2	Timeout clock frequency 1	Timeout clock frequency 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	1	0	0	0	0	0	0	0

Capabilities Register 1 **REG[1141h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Base clock Freq for SD Clock 5	Base clock Freq for SD Clock 4	Base clock Freq for SD Clock 3	Base clock Freq for SD Clock 2	Base clock Freq for SD Clock 1	Base clock Freq for SD Clock 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Capabilities Register 2 **REG[1142h]**

Bit	7	6	5	4	3	2	1	0
	Suspend / Resume support	DMA support	High speed support	Reserved	Reserved	Reserved	Max block length 1	Max block length 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	1	0	0	0	0	0	1

Capabilities Register 3 **REG[1143h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	3.3V Voltage support
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	1

Capabilities Register 4 **REG[1144h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Bit	7	6	5	4	3	2	1	0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Capabilities Register 5 **REG[1145h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Capabilities Register 6 **REG[1146h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Capabilities Register 7 **REG[1147h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

This register provides the Host Driver with information specific to the Host Controller implementation. The Host Controller may implement these values as fixed or loaded from flash memory during power on initialization. Refer to **Software Reset For All** in the *Software Reset* register for loading from flash memory and completion timing control.

REG[1143h] Bits 7-1 **Reserved bits**

REG[1143h] Bit 0 **Voltage Support 3.3V**
1 3.3V Supported
0 3.3V Not Supported

REG[1142h] Bit 7 **Suspend/Resume Support**
This bit indicates whether the Host Controller supports Suspend / Resume functionality. If this bit is 0, the Suspend and Resume mechanism (Refer to SD Specification [i] 'Suspend and Resume mechanism') are not supported and the Host Driver shall not issue either Suspend or Resume commands.
1 Supported
0 Not supported

REG[1142h] Bit 6 **DMA Support**
This bit indicates whether the Host Controller is capable of using DMA to transfer data between system memory and the Host Controller directly.
1 DMA Supported
0 DMA not supported

REG[1142h] Bit 5 **High Speed Support**
This bit indicates whether the Host Controller and the Host System support High Speed mode and they can supply SD Clock frequency from 25MHz to 50MHz.
1 High Speed Supported
0 High Speed not supported

REG[1142h] Bits 4-2 **Reserved bits**

REG[1142h] Bits 1-0 **Max Block Length bits [1:0]**
This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer shall transfer this block size without wait cycles.

Block sizes can be defined as indicated below.

00/10/11	Reserved
01	1024

REG[1141h] Bits 7-6 **Reserved bits**

REG[1141h] Bits 5-0 **Base Clock Frequency for SD Clock bits [5:0]**
This value indicates the base (maximum) clock frequency for the SD Clock. Unit values are 1MHz. If the real frequency is 16.5MHz, the lager value shall be set 01 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to the **SDCLK Frequency Select** in the *Clock Control* register.) and it shall not exceed upper limit of the SD Clock frequency. The supported clock range is 10MHz to 63MHz. If these bits are all 0, the Host System has to get information via another method.

Not 0	Reserved
0	Get information via another method

REG[1140h] Bit 7 **Timeout Clock unit**
This bit shows the unit of base clock frequency used to detect Data Timeout Error.

0	kHz
1	MHz

REG[1140h] Bit 6 **Reserved bit**

REG[1140h] Bits 5-0 **Timeout clock Frequency bits [5:0]**
This bit shows the base clock frequency used to detect **Data Timeout Error**.
The **Timeout Clock Unit** defines the unit of this fields value.
Timeout Clock Unit =0 [KHz] unit: 1KHz to 63KHz
Timeout Clock Unit =1 [MHz] unit: 1MHz to 63MHz

Not 0	1KHz to 63KHz or 1MHz to 63MHz
0	Get information via another method

REG[1144h] Bits 7-0 **Reserved bits**
REG[1145h] Bits 7-0
REG[1146h] Bits 7-0
REG[1147h] Bits 7-0

Maximum Current Capabilities Register 0 **REG[1148h]**

Bit	7	6	5	4	3	2	1	0
	Maximum current for 3.3V bit 7	Maximum current for 3.3V bit 6	Maximum current for 3.3V bit 5	Maximum current for 3.3V bit 4	Maximum current for 3.3V bit 3	Maximum current for 3.3V bit 2	Maximum current for 3.3V bit 1	Maximum current for 3.3V bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Maximum Current Capabilities Register 1 **REG[1149h]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Maximum Current Capabilities Register 2 **REG[114Ah]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Maximum Current Capabilities Register 3 **REG[114Bh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Maximum Current Capabilities Register 4 **REG[114Ch]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Maximum Current Capabilities Register 5 **REG[114Dh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Maximum Current Capabilities Register 6 **REG[114Eh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Maximum Current Capabilities Register 7 **REG[114Fh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG[1149h-114Fh] Bits 7-0 **Reserved bits**

REG[1148h] Bits 7-0 **Maximum Current for 3.3V bits [7:0]**

Slot Interrupt Register 0 **REG[11FCh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Slot 0 Interrupt status
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Slot Interrupt Register 1 **REG[11FDh]**

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

REG[11FDh] Bits 7-0 **Reserved bits**
REG[11FCh] Bits 7-6

REG[11FCh] Bit 0 **Slot Interrupt bit**
The Host Controller supports only 1 slot.
This status bits indicate the logical OR of Interrupt Signal and Wakeup Signal for each slot. A maximum of 8 slots can be defined. If one interrupt signal is associated with multiple slots, the Host Driver can know which interrupt is generated by reading these status bits.
By a power on reset or by setting **Software Reset For All**, the interrupt signal shall be de-asserted and this status shall read 00h.

Host Controller Version Register 0 **REG[11FEh]**

Bit	7	6	5	4	3	2	1	0
	Specification version number bit 7	Specification version number bit 6	Specification version number bit 5	Specification version number bit 4	Specification version number bit 3	Specification version number bit 2	Specification version number bit 1	Specification version number bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	0	0	0	0	0

Host Controller Version Register 1 **REG[11FFh]**

Bit	7	6	5	4	3	2	1	0
	Vendor version number bit 7	Vendor version number bit 6	Vendor version number bit 5	Vendor version number bit 4	Vendor version number bit 3	Vendor version number bit 2	Vendor version number bit 1	Vendor version number bit 0
Type	RO	RO	RO	RO	RO	RO	RO	RO
Reset state	0	0	0	1	0	0	0	0

REG[11FFh] Bits 7-0, **Vendor version bits [7:0]**
The Vendor Version Number field is implementation specific and shall be 2 BCD fields. The value of the Vendor Version Number shall be set to 0x10 to indicate Version 1.0 of the Host Controller design.

REG[11FEh] Bits 7-0, **Specification version bits [7:0]**
The Specification Version Number shall be set to 8' b0 to indicate SD Host Specification, V1.0.

http://www.solomon-systech.com

[i] Part A2, SD Host Controller Standard Specification, Version 1.0, February 2004
[ii] Part 1, Physical Layer Specification, Version 1.01