

# **SSD1961**

## *Advance Information*

### **675KB Embedded Display SRAM LCD Display Controller**

This document contains information on a new product. Specifications and information herein are subject to change without notice.

### Appendix: IC Revision history of SSD1961 Specification

Version	Change Items	Effective Date
0.10 27-Mar-08	1 <sup>st</sup> Release	02-Apr-08
0.20 17-Jul-08	<ol style="list-style-type: none"> <li>1. Added Table 6-5.</li> <li>2. Updated Table 12-1.</li> <li>3. VSSIO and VSSD are tied together and renamed as VSS. Modified Figure 5-1, Table 5-1 and Table 6-4.</li> <li>4. Updated package information in Section 13.4.</li> <li>5. Added Section 13.1.</li> </ol>	28-Jul-08
0.30 09-Sep-08	<ol style="list-style-type: none"> <li>1. Added Section 11 and 11.1.</li> <li>2. Updated Figure 13-1, Figure 13-2 and Figure 13-3.</li> </ol>	10-Sep-08
0.40 21-Nov-08	<ol style="list-style-type: none"> <li>1. Added command descriptions in section 9</li> <li>2. Added application circuit in section 13.4</li> <li>3. Added SSD1961G40R part number</li> <li>4. Added Tape &amp; Reel drawing in section 15</li> <li>5. Revised the VSS pin # (D5 and E5) in Table 6-4</li> <li>6. Revised the Tape &amp; Reel drawing in section 15</li> <li>7. Updated maximum input clock frequency in Table 13-2</li> <li>8. Updated 6-bit serial RGB in Table 6-5</li> <li>9. Updated 9-bit interface in Table 7-1</li> </ol>	24-Nov-08
0.50 08-Dec-08	<ol style="list-style-type: none"> <li>1. Changed the set_pll_mnk to set_pll_mn in section 7.2</li> <li>2. Change register name in section 8</li> <li>3. Removed ABC</li> <li>4. Revised description for REG 0x00, 0x01, 0x0C, 0x0D, 0x0E, 0x10, 0x11, 0x21, 0x26, 0x28, 0x2A, 0x2B, 0x2C, 0x2E, 0x33, 0x34, 0x35, 0x36, 0x37, 0x3A, 0x3C, 0x3E, 0x44, 0x45, 0xA1, 0xB0, 0xB1, 0xB4, 0xB5, 0xB6, 0xB7, 0xB8, 0xB9, 0xBE, 0xBF, 0xD0, 0xD1, 0xD4, 0xE5.</li> <li>5. Added max VIH in Table 12-1</li> <li>6. Added Table 9-1</li> <li>7. Added Table 11-1</li> <li>8. Revised Figure 9-19</li> <li>9. Revised Figure 14-2</li> <li>10. Revised Figure 13-5</li> <li>11. Corrected typo for Table 7-2</li> <li>12. Revised test condition for 12 and 13</li> </ol>	10-Dec-08
1.0 07-May-09	<ol style="list-style-type: none"> <li>1. Changed status to Advance Information</li> <li>2. Update min/max rating of VDDD and VDDPLL in Table 11-1</li> <li>3. Added serial RGB interface timing in Section 13.4</li> <li>4. Revised Section 13.2</li> <li>5. Added 12 bit in Table 7-1</li> <li>6. Removed TTL interface</li> <li>7. Changed the serial TFT interface from 6 bit to 8 bit</li> <li>8. Revised section 7.1.5</li> <li>9. Change the title of section 7.2</li> <li>10. Revised command description in section 8.</li> <li>11. Removed the command 0x0C and 0x3A</li> <li>12. Added figures in section 13.4</li> <li>13. Revised MCU Interface Timing in section 13.2</li> <li>14. Revised figures in section 13.3</li> <li>15. Revise Table 6-1</li> </ol>	18-May-09
1.1 31-May-10	<ol style="list-style-type: none"> <li>1. Updated Table 7-1</li> <li>2. Revised section 9.72</li> <li>3. Add table 13-7</li> </ol>	15-Jul-10
1.2 11-Nov-10	<ol style="list-style-type: none"> <li>1. Update Section 7.2 reset timing</li> <li>2. Correct Section 13.4 the serial RGB timing</li> <li>3. Correct Table 6.1-6.5 Pin Mapping -&gt; Pin description</li> </ol>	07-Dec-10

1.3 21-Jul-11	1. Update Section 9.45 SET_PWM_CONF Update Section 9.30 SET_TEAR_SCANLINE	25-Jul-11
------------------	--	-----------

# CONTENTS

<b>1</b>	<b>GENERAL DESCRIPTION .....</b>	<b>9</b>
<b>2</b>	<b>FEATURES.....</b>	<b>9</b>
<b>3</b>	<b>ORDERING INFORMATION .....</b>	<b>9</b>
<b>4</b>	<b>BLOCK DIAGRAM .....</b>	<b>10</b>
<b>5</b>	<b>PIN ARRANGEMENT.....</b>	<b>11</b>
5.1	64 BALLS TFBGA .....	11
<b>6</b>	<b>PIN DESCRIPTIONS .....</b>	<b>12</b>
<b>7</b>	<b>FUNCTIONAL BLOCK DESCRIPTIONS.....</b>	<b>14</b>
7.1	MCU INTERFACE.....	14
7.1.1	6800 Mode .....	14
7.1.2	8080 Mode .....	14
7.1.3	Register Pin Mapping .....	14
7.1.4	Pixel Data Format .....	14
7.1.5	Tearing Effect Signal (TE).....	15
7.2	SYSTEM CLOCK GENERATION .....	15
7.3	FRAME BUFFER.....	16
7.4	SYSTEM CLOCK AND RESET MANAGER .....	16
7.5	LCD CONTROLLER .....	17
7.5.1	Display Format .....	17
7.5.2	General Purpose Input/Output (GPIO) .....	17
<b>8</b>	<b>COMMAND TABLE .....</b>	<b>18</b>
<b>9</b>	<b>COMMAND DESCRIPTIONS.....</b>	<b>21</b>
9.1	NOP .....	21
9.2	SOFT_RESET .....	21
9.3	GET_POWER_MODE.....	21
9.4	GET_ADDRESS_MODE .....	22
9.5	GET_DISPLAY_MODE .....	22
9.6	GET_TEAR_EFFECT_STATUS .....	23
9.7	ENTER_SLEEP_MODE .....	24
9.8	EXIT_SLEEP_MODE .....	24
9.9	ENTER_PARTIAL_MODE .....	24
9.10	ENTER_NORMAL_MODE .....	24
9.11	EXIT_INVERT_MODE .....	25
9.12	ENTER_INVERT_MODE .....	25
9.13	SET_GAMMA_CURVE .....	26
9.14	SET_DISPLAY_OFF .....	26
9.15	SET_DISPLAY_ON.....	26
9.16	SET_COLUMN_ADDRESS .....	27
9.17	SET_PAGE_ADDRESS.....	27
9.18	WRITE_MEMORY_START .....	28
9.19	READ_MEMORY_START .....	29
9.20	SET_PARTIAL_AREA.....	29
9.21	SET_SCROLL_AREA.....	31
9.22	SET_TEAR_OFF .....	33
9.23	SET_TEAR_ON.....	33
9.24	SET_ADDRESS_MODE.....	33
9.25	SET_SCROLL_START .....	36

9.26	EXIT_IDLE_MODE .....	37
9.27	ENTER_IDLE_MODE .....	37
9.28	WRITE_MEMORY_CONTINUE.....	38
9.29	READ_MEMORY_CONTINUE .....	39
9.30	SET_TEAR_SCANLINE.....	40
9.31	GET_SCANLINE .....	40
9.32	READ_DDB .....	41
9.33	SET_LCD_MODE.....	41
9.34	GET_LCD_MODE .....	43
9.35	SET_HORI_PERIOD .....	44
9.36	GET_HORI_PERIOD.....	44
9.37	SET_VERT_PERIOD.....	45
9.38	GET_VERT_PERIOD .....	46
9.39	SET_GPIO_CONF.....	47
9.40	GET_GPIO_CONF .....	48
9.41	SET_GPIO_VALUE .....	49
9.42	GET_GPIO_STATUS.....	49
9.43	SET_POST_PROC.....	50
9.44	GET_POST_PROC .....	50
9.45	SET_PWM_CONF.....	51
9.46	GET_PWM_CONF .....	52
9.47	SET_LCD_GEN0.....	54
9.48	GET_LCD_GEN0 .....	55
9.49	SET_LCD_GEN1 .....	56
9.50	GET_LCD_GEN1 .....	57
9.51	SET_LCD_GEN2.....	58
9.52	GET_LCD_GEN2 .....	59
9.53	SET_LCD_GEN3.....	60
9.54	GET_LCD_GEN3 .....	61
9.55	SET_GPIO0_ROP .....	62
9.56	GET_GPIO0_ROP.....	62
9.57	SET_GPIO1_ROP.....	63
9.58	GET_GPIO1_ROP.....	64
9.59	SET_GPIO2_ROP.....	64
9.60	GET_GPIO2_ROP.....	65
9.61	SET_GPIO3_ROP .....	66
9.62	GET_GPIO3_ROP.....	66
9.63	SET_DBC_CONF.....	67
9.64	GET_DBC_CONF .....	68
9.65	SET_DBC_TH.....	69
9.66	GET_DBC_TH .....	70
9.67	SET_PLL .....	70
9.68	SET_PLL_MN.....	71
9.69	GET_PLL_MN .....	72
9.70	GET_PLL_STATUS .....	72
9.71	SET_DEEP_SLEEP .....	72
9.72	SET_LSHIFT_FREQ.....	73
9.73	GET_LSHIFT_FREQ .....	73
9.74	SET_PIXEL_DATA_INTERFACE .....	75
9.75	GET_PIXEL_DATA_INTERFACE.....	75
<b>10</b>	<b>MAXIMUM RATINGS.....</b>	<b>76</b>
<b>11</b>	<b>RECOMMENDED OPERATING CONDITIONS .....</b>	<b>76</b>
11.1	POWER-UP SEQUENCE.....	76
<b>12</b>	<b>DC CHARACTERISTICS.....</b>	<b>77</b>
<b>13</b>	<b>AC CHARACTERISTICS.....</b>	<b>77</b>

13.1	CLOCK TIMING .....	77
13.2	MCU INTERFACE TIMING .....	78
13.2.1	<i>Parallel 6800-series Interface Timing</i> .....	78
13.2.2	<i>Parallel 8080-series Interface Timing</i> .....	80
13.3	PARALLEL LCD INTERFACE TIMING.....	82
13.4	SERIAL RGB INTERFACE TIMING .....	83
<b>14</b>	<b>APPLICATION EXAMPLE.....</b>	<b>86</b>
<b>15</b>	<b>PACKAGE INFORMATION.....</b>	<b>88</b>
15.1	PACKAGE MECHANICAL DRAWING FOR TFBGA.....	88
15.2	TAPE & REEL DRAWING FOR TFBGA .....	89

## TABLES

TABLE 3-1: ORDERING INFORMATION .....	9
TABLE 5-1: TFBGA PIN ASSIGNMENT TABLE.....	11
TABLE 6-1: MCU INTERFACE PIN DESCRIPTION .....	12
TABLE 6-2: LCD INTERFACE PIN DESCRIPTION.....	12
TABLE 6-3: CONTROL SIGNAL PIN DESCRIPTION.....	13
TABLE 6-4: POWER PIN DESCRIPTION.....	13
TABLE 6-5: LCD INTERFACE PIN MAPPING.....	13
TABLE 7-1: PIXEL DATA FORMAT .....	14
TABLE 7-2: FRAME BUFFER SETTINGS REGARDING TO SET_ADDRESS_MODE COMMAND 0x36.....	16
TABLE 9-1: ENTER IDLE MODE MEMORY CONTENT VS DISPLAY COLOR .....	38
TABLE 10-1: MAXIMUM RATINGS (VOLTAGE REFERENCED TO $V_{SS}$ ).....	76
TABLE 11-1: RECOMMENDED OPERATING CONDITION.....	76
TABLE 12-1: DC CHARACTERISTICS.....	77
TABLE 13-1: CLOCK INPUT REQUIREMENTS FOR CLK (PLL-BYPASS) .....	77
TABLE 13-2: CLOCK INPUT REQUIREMENTS FOR CLK (PLL-ENABLED).....	77
TABLE 13-3: CLOCK INPUT REQUIREMENTS FOR CRYSTAL OSCILLATOR XTAL (PLL-ENABLED) .....	77
TABLE 13-4: PARALLEL 6800-SERIES INTERFACE TIMING CHARACTERISTICS (USE CS# AS CLOCK).....	78
TABLE 13-5: PARALLEL 6800-SERIES INTERFACE TIMING CHARACTERISTICS (USE E AS CLOCK) .....	79
TABLE 13-6: PARALLEL 8080-SERIES INTERFACE TIMING CHARACTERISTICS.....	80
TABLE 13-7: QUICK REFERENCE TABLE FOR LCD PARAMETER SETTING .....	85

## FIGURES

FIGURE 4-1: SSD1961 BLOCK DIAGRAM .....	10
FIGURE 5-1: PINOUT DIAGRAM – 64 BALLS TFBGA (TOP VIEW) .....	11
FIGURE 7-1: RELATIONSHIP BETWEEN TEARING EFFECT SIGNAL AND MCU MEMORY WRITING.....	15
FIGURE 7-2: CLOCK CONTROL DIAGRAM .....	16
FIGURE 7-3: STATE DIAGRAM OF SSD1961 .....	17
FIGURE 9-1: EXIT INVERT MODE EXAMPLE.....	25
FIGURE 9-2: ENTER INVERT MODE EXAMPLE.....	26
FIGURE 9-3: SET COLUMN ADDRESS EXAMPLE .....	27
FIGURE 9-4: SET PAGE ADDRESS EXAMPLE .....	28
FIGURE 9-5: SET PARTIAL AREA WITH SET_ADDRESS_MODE (0x36) A[4] = 0 WHEN END ROW > START ROW .....	30
FIGURE 9-6: SET PARTIAL AREA WITH SET_ADDRESS_MODE (0x36) A[4] = 1 WHEN END ROW > START ROW .....	30
FIGURE 9-7: SET PARTIAL AREA WITH SET_ADDRESS_MODE (0x36) A[4] = 0 WHEN START ROW > END ROW .....	30
FIGURE 9-8: SET PARTIAL AREA WITH SET_ADDRESS_MODE (0x36) A[4] = 1 WHEN START ROW > END ROW .....	31
FIGURE 9-9: SET SCROLL AREA WITH SET_ADDRESS_MODE (0x36) A[4] = 0.....	32
FIGURE 9-10: SET SCROLL AREA WITH SET_ADDRESS_MODE (0x36) A[4] = 1 .....	32
FIGURE 9-11: A[7] PAGE ADDRESS ORDER .....	34
FIGURE 9-12: A[6] COLUMN ADDRESS ORDER.....	34
FIGURE 9-13: A[5] PAGE / COLUMN ADDRESS ORDER .....	34
FIGURE 9-14: A[3] RGB ORDER.....	35
FIGURE 9-15: A[1] FLIP HORIZONTAL .....	35
FIGURE 9-16: A[0] FLIP VERTICAL .....	36
FIGURE 9-17: SET SCROLL START WITH SET_ADDRESS_MODE (0x36) A[4] = 0.....	36
FIGURE 9-18: SET SCROLL START WITH SET_ADDRESS_MODE (0x36) A[4] = 1.....	37
FIGURE 9-19: PWM SIGNAL .....	51
FIGURE 11-1: POWER-UP SEQUENCE .....	76
FIGURE 13-1: PARALLEL 6800-SERIES INTERFACE TIMING DIAGRAM (USE CS# AS CLOCK).....	78
FIGURE 13-2: PARALLEL 6800-SERIES INTERFACE TIMING DIAGRAM (USE E AS CLOCK) .....	79
FIGURE 13-3: PARALLEL 8080-SERIES INTERFACE TIMING DIAGRAM (WRITE CYCLE) .....	80
FIGURE 13-4: PARALLEL 8080-SERIES INTERFACE TIMING DIAGRAM (READ CYCLE).....	81
FIGURE 13-5: GENERIC TFT PANEL TIMING.....	82
FIGURE 13-6: SERIAL RGB INTERFACE TIMING (WITHOUT DUMMY MODE).....	83
FIGURE 13-7: SERIAL RGB INTERFACE TIMING (WITH DUMMY MODE) .....	84
FIGURE 14-1: APPLICATION CIRCUIT FOR SSD1961 (WITH DIRECT CLOCK INPUT) .....	86
FIGURE 14-2: APPLICATION CIRCUIT FOR SSD1961 (WITH CRYSTAL OSCILLATOR INPUT).....	87



# 1 GENERAL DESCRIPTION

SSD1961 is a display controller of 5,529,600 bit frame buffer to support up to 640 x 480 x 18bit graphics content. It also equips parallel MCU interfaces in different bus width to receive graphics data and command from MCU. Its display interface supports common RAM-less LCD driver of color depth up to 18 bit-per-pixel.

# 2 FEATURES

- Display feature
  - 675kbyte (5,529,600bit) built-in frame buffer. Support up to 640 x 480 at 18bpp display
  - Support TFT 18 bit generic RGB interface panel
  - Support 8-bit serial RGB interface
  - Hardware rotation of 0, 90, 180, 270 degree
  - Hardware display mirroring
  - Hardware windowing
  - Programmable brightness, contrast and saturation control
  - Dynamic Backlight Control (DBC) via PWM signal
- MCU connectivity
  - 8/9/16/18-bit MCU interface
  - Tearing effect signal
- I/O Connectivity
  - 4 GPIO pins
- Built-in clock generator
- Deep sleep mode for power saving
- 64 pin BGA package
- Core supply power ( $V_{DDPLL}$  and  $V_{DDD}$ ): 1.2V±0.1V
- I/O supply power ( $V_{DDIO}$ ): 1.65V to 3.6V
- LCD interface supply power ( $V_{DDLCD}$ ): 1.65V to 3.6V

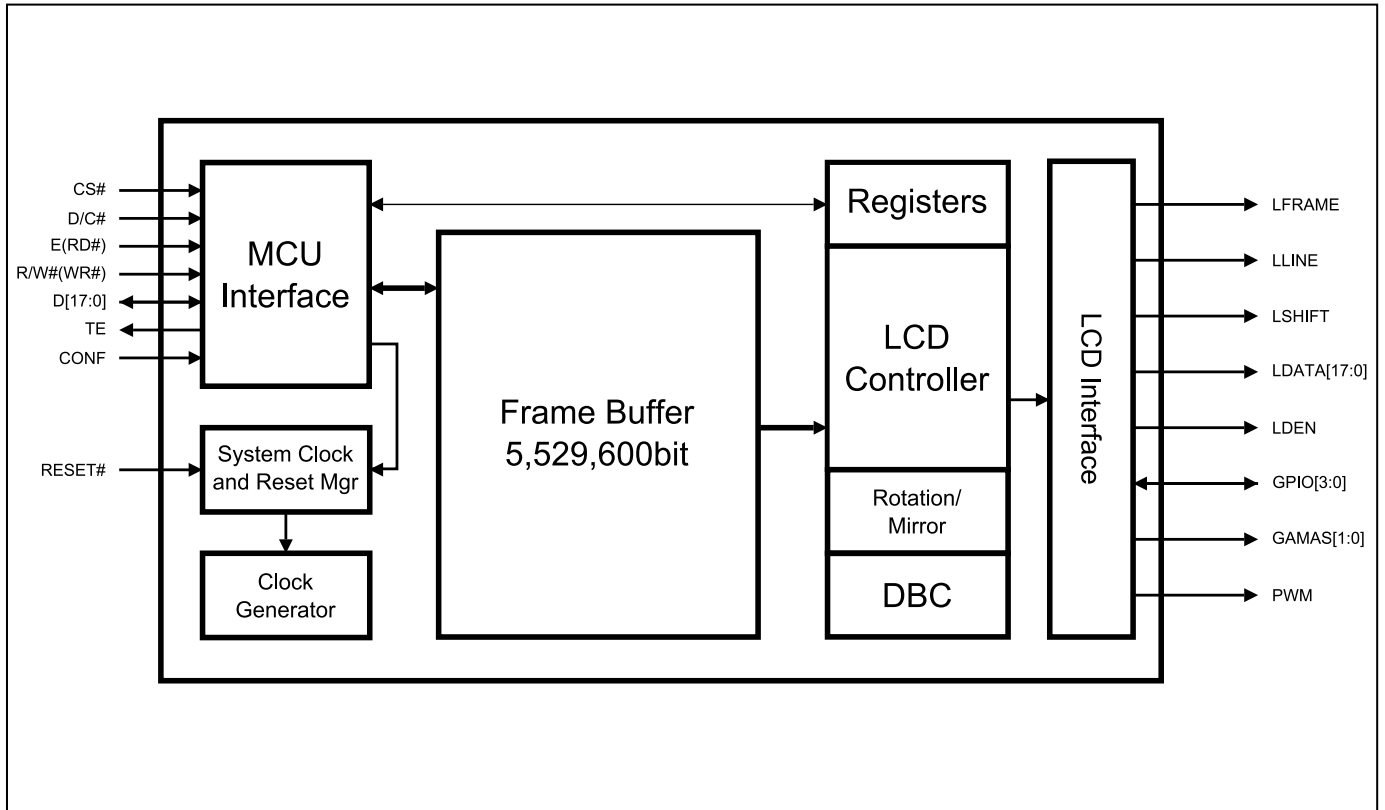
# 3 ORDERING INFORMATION

**Table 3-1: Ordering Information**

<b>Ordering Part Number</b>	<b>Package Form</b>
SSD1961G40	TFBGA-64 (Tray)
SSD1961G40R	TFBGA-64 (Tape & Reel)

## 4 BLOCK DIAGRAM

Figure 4-1: SSD1961 Block Diagram



## 5 PIN ARRANGEMENT

### 5.1 64 Balls TFBGA

Figure 5-1: Pinout Diagram – 64 balls TFBGA (Top view)

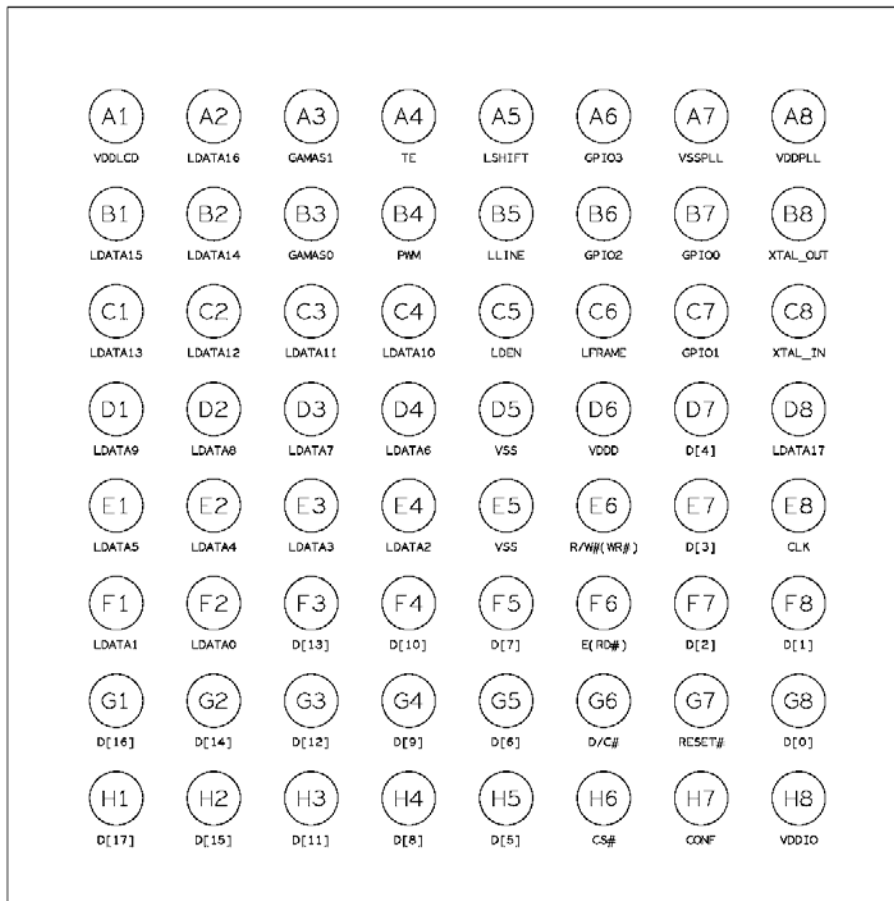


Table 5-1: TFBGA Pin Assignment Table

Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name
A1	VDDLCD	C1	LDATA13	E1	LDATA5	G1	D[16]
A2	LDATA16	C2	LDATA12	E2	LDATA4	G2	D[14]
A3	GAMAS1	C3	LDATA11	E3	LDATA3	G3	D[12]
A4	TE	C4	LDATA10	E4	LDATA2	G4	D[9]
A5	LSHIFT	C5	LDEN	E5	VSS	G5	D[6]
A6	GPIO3	C6	LFRAME	E6	R/W#(WR#)	G6	D/C#
A7	VSSPLL	C7	GPIO1	E7	D[3]	G7	RESET#
A8	VDDPLL	C8	XTAL_IN	E8	CLK	G8	D[0]
B1	LDATA15	D1	LDATA9	F1	LDATA1	H1	D[17]
B2	LDATA14	D2	LDATA8	F2	LDATA0	H2	D[15]
B3	GAMAS0	D3	LDATA7	F3	D[13]	H3	D[11]
B4	PWM	D4	LDATA6	F4	D[10]	H4	D[8]
B5	LLINE	D5	VSS	F5	D[7]	H5	D[5]
B6	GPIO2	D6	VDDD	F6	E(RD#)	H6	CS#
B7	GPIO0	D7	D[4]	F7	D[2]	H7	CONF
B8	XTAL_OUT	D8	LDATA17	F8	D[1]	H8	VDDIO

## 6 PIN DESCRIPTIONS

### Key:

I = Input  
 O = Output  
 IO = Bi-directional (input/output)  
 P = Power pin  
 Hi-Z = High impedance

**Table 6-1: MCU Interface Pin Description**

Pin Name	Type	Reference Voltage Level	Pin #	Description
CLK	I	VDDIO	E8	TTL clock input. This pin should be tied to VSS if TTL clock input is not used
XTAL_IN	I	-	C8	Crystal oscillator input. This pin should be tied to VSS if not used
XTAL_OUT	O	-	B8	Crystal oscillator output. This pin should be floating if not used
CS#	I	VDDIO	H6	Chip select
D/C#	I	VDDIO	G6	Data/Command select
E(RD#)	I	VDDIO	F6	6800 mode: E (enable signal) 8080 mode: RD# (read strobe signal)
R/W#(WR#)	I	VDDIO	E6	6800 mode: R/W# 0: Write cycle 1: Read cycle 8080 mode: WR# (write strobe signal)
D[17:0]	IO	VDDIO	D7, E7, F3-F5, F7-F8, G1-G5, G8, H1-H5	Data bus. Pins not used should be floating
TE	O	VDDLCD	A4	Tear effect

**Table 6-2: LCD Interface Pin Description**

Pin Name	Type	Reference Voltage Level	Pin #	Description
LFRAME	O	VDDLCD	C6	Vertical sync (Frame pulse)
LLINE	O	VDDLCD	B5	Horizontal sync (Line pulse)
LSHIFT	O	VDDLCD	A5	Pixel clock (Pixel shift signal)
LDEN	O	VDDLCD	C5	Data valid
LDATA[17:0]	O	VDDLCD	A2, B1-B2, C1-C4, D1-D4, D8, E1-E4, F1-F2	RGB data
GPIO[3:0]	IO	VDDLCD	A6, B6-B7, C7	These pins be configured for display miscellaneous signals or as general purpose I/O. Default as input
GAMAS [1:0]	O	VDDLCD	A3, B3	Gamma selection for panel
PWM	O	VDDLCD	B4	PWM output for backlight driver

**Table 6-3: Control Signal Pin Description**

Pin Name	Type	Reference Voltage Level	Pin #	Description
RESET#	I	VDDIO	G7	Master synchronize reset
CONF	I	VDDIO	H7	MCU interface configuration 0: 6800 Interface 1: 8080 Interface

**Table 6-4: Power Pin Description**

Pin Name	Type	Pin #	Description
VDDD	P	D6	Power supply for internal digital circuit
VDDLCD	P	A1	Power supply for LCD interface related pads
VDDPLL	P	A8	Power supply for internal analog circuit and analog I/O pads
VDDIO	P	H8	Power supply for digital I/O pads
VSS	P	D5, E5	Ground for internal digital circuit
VSSPLL	P	A7	Ground for internal analog circuit and analog I/O pads

**Table 6-5: LCD Interface Pin Mapping**

Pin Names	18-bit TFT	8-bit Serial
LFRAME	FRAME	
LLINE	LINE	
LSHIFT	SHIFT	
LDEN	DEN	
LDATA17	R5	Drive 0
LDATA16	R4	Drive 0
LDATA15	R3	Drive 0
LDATA14	R2	Drive 0
LDATA13	R1	Drive 0
LDATA12	R0	Drive 0
LDATA11	G5	Drive 0
LDATA10	G4	Drive 0
LDATA9	G3	Drive 0
LDATA8	G2	Drive 0
LDATA7	G1	D7
LDATA6	G0	D6
LDATA5	B5	D5
LDATA4	B4	D4
LDATA3	B3	D3
LDATA2	B2	D2
LDATA1	B1	D1
LDATA0	B0	D0

**Note**

<sup>(1)</sup> These pin mappings use signal names commonly used for each panel type, however signal names may differ between panel manufacturers.

## 7 FUNCTIONAL BLOCK DESCRIPTIONS

### 7.1 MCU Interface

The MCU interface connects the MCU and SSD1961 graphics controller. The MCU interface can be configured as 6800 mode and 8080 mode by the CONF pin. By pulling the CONF pin to VSS, the MCU interface will be configured as 6800 mode interface. If the CONF pin is connected to VDDIO, the MCU interface will be configured as 8080 mode.

#### 7.1.1 6800 Mode

The 6800 mode MCU interface consist of CS#, D/C#, E, R/W#, D[17:0], and TE signals (Please refer to Table 6-1 for pin multiplexed with 8080 mode). This interface supports both fixed E and clock E scheme to define a read/write cycle. If the E signal is kept high and used as enable signal, the CS# signal acts as a bus clock, the data or command will be latched into the system at the rising edge of CS#. If the user wants to use the E pin as the clock pin, the CS# pin then need to be fixed to logic 0 to select the chip. Then the falling edge of the E signal will latch the data or command. For details, please refer to the timing diagram in chapter 13.2.1.

#### 7.1.2 8080 Mode

The 8080 mode MCU interface consist of CS#, D/C#, RD#, WR#, D[17:0] and TE signals (Please refer to Table 6-1 for pin multiplexed with 6800 mode). This interface use WR# to define a write cycle and RD# for read cycle. If the WR# goes low when the CS# signal is low, the data or command will be latched into the system at the rising edge of WR#. Similarly, the read cycle will start when RD# goes low and end at the rising edge of RD#. The detailed timing will show in the chapter **Error! Reference source not found.**

#### 7.1.3 Register Pin Mapping

When user access the registers via the parallel MCU interface, only D[7:0] will be used regardless the width of the pixel data is. Therefore, D[17:8] will only be used to address the display data only. This provided the possibility that the pixel data format as shown in Table 7-1 can be configured by command 0xF0.

#### 7.1.4 Pixel Data Format

Both 6800 and 8080 support 8 bit, 16 bit, 18 bit data bus. Depending on the width of the data bus, the display data are packed into the data bus in different ways.

Table 7-1: Pixel Data Format

Interface	Cycle	D[17]	D[16]	D[15]	D[14]	D[13]	D[12]	D[11]	D[10]	D[9]	D[8]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]	
18 bits	1 <sup>st</sup>	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0	
16 bits (565 format)	1 <sup>st</sup>			R5	R4	R3	R2	R1	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	
16 bits	1 <sup>st</sup>			R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	
	2 <sup>nd</sup>			B7	B6	B5	B4	B3	B2	B1	B0	R7	R6	R5	R4	R3	R2	R1	R0	
	3 <sup>rd</sup>			G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0	
12 bits	1 <sup>st</sup>							R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	
	2 <sup>nd</sup>							G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0	
9 bits	1 <sup>st</sup>											R5	R4	R3	R2	R1	R0	G5	G4	G3
	2 <sup>nd</sup>											G2	G1	G0	B5	B4	B3	B2	B1	B0
8 bits	1 <sup>st</sup>											R7	R6	R5	R4	R3	R2	R1	R0	
	2 <sup>nd</sup>											G7	G6	G5	G4	G3	G2	G1	G0	
	3 <sup>rd</sup>											B7	B6	B5	B4	B3	B2	B1	B0	

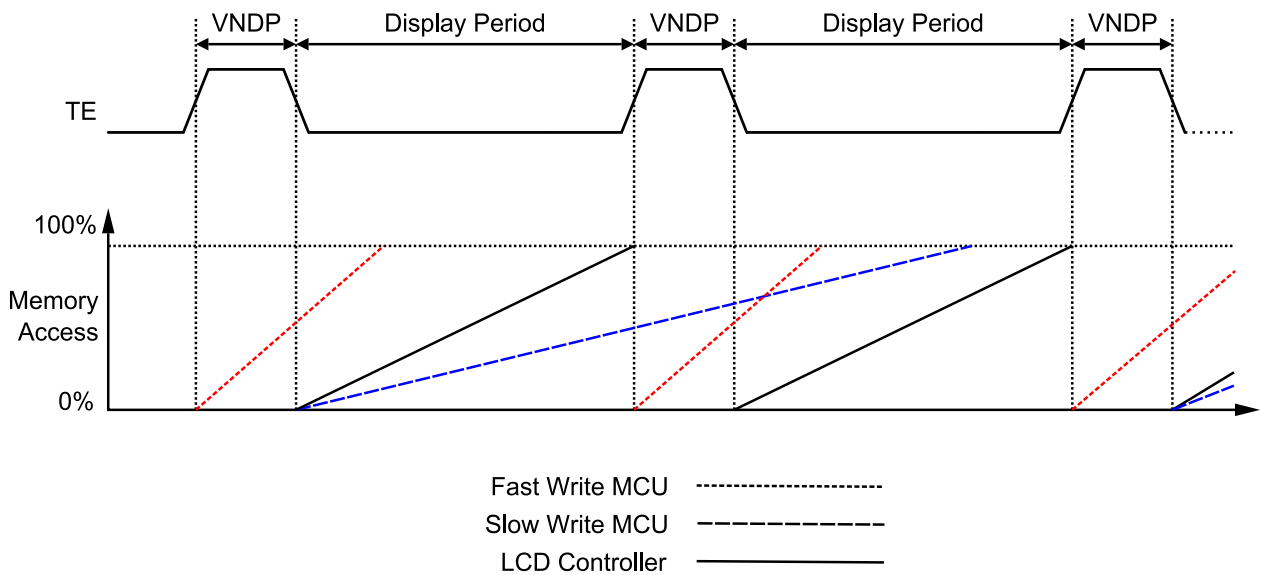
X: Don't Care

### 7.1.5 Tearing Effect Signal (TE)

The Tearing Effect Signal (TE) is a feedback signal from the LCD Controller to MCU. This signal reveals the display status of LCD controller. In the non-display period, the TE signal will go high. Therefore, this signal enables the MCU to send data by observing the non-display period to avoid tearing.

Figure 7-1 shows how the TE signal helps to avoid tearing. If the MCU writing speed is slower than the display speed, the display data should be updated after the LCD controller start to scan the frame buffer. Then the LCD controller will always display the old memory content until the next frame. However, if the MCU is faster than the LCD controller, it should start updating the display content in the vertical non-display period (VNDP) to enable the LCD controller will always get the newly updated data.

**Figure 7-1: Relationship between Tearing Effect Signal and MCU Memory Writing**



In SSD1961, users can configure the TE signal to reflect the vertical non-display period only or reflect both vertical and horizontal non-display period. With the additional horizontal non-display period information, the MCU can control the refresh action in more accurately by counting the horizontal line scanned by the LCD controller. Usually, a fast MCU will not need horizontal non-display period. But a slow MCU will need it to ensure the frame buffer update process always lags behind the LCD controller.

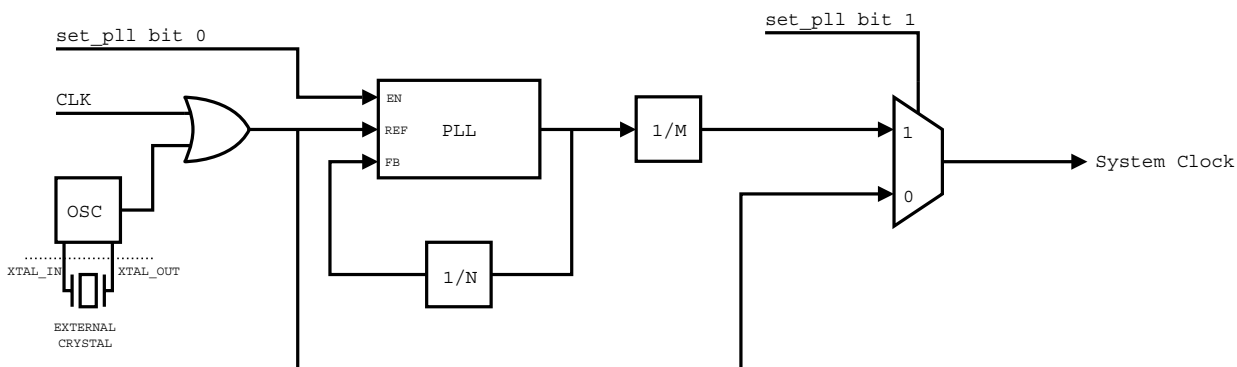
### 7.2 System Clock Generation

The system clock of SSD1961 is generated by the built-in PLL. The reference clock of the PLL can come from either the CLK pin or the external crystal oscillator. Since the CLK pin and the output of the oscillator was connected to PLL with an “OR” gate, the unused clock must be tied to VSS.

Before the PLL output is configured as the system clock by the bit 1 of “set\_pll” command 0xE0, the system will be clocked by the reference clock. This enables the user to send the “set\_pll\_mn” command 0xE2 to the PLL for frequency configuration. When the PLL frequency is configured and the PLL was enabled with the

bit 0 of “set\_pll” command 0xE0, the user should still wait for **100us** for the PLL to lock. Then the PLL is ready and can be configured as system clock with the bit 1 of “set\_pll” command 0xE0.

**Figure 7-2: Clock Control Diagram**



### 7.3 Frame Buffer

There are 5,529,600 bit built-in SRAM inside SSD1961 to use as frame buffer. When the frame buffer is written or read, the “address counter” will automatically increase by one or decrease by one depends on the frame buffer settings.

**Table 7-2: Frame Buffer Settings regarding to set\_address\_mode command 0x36**

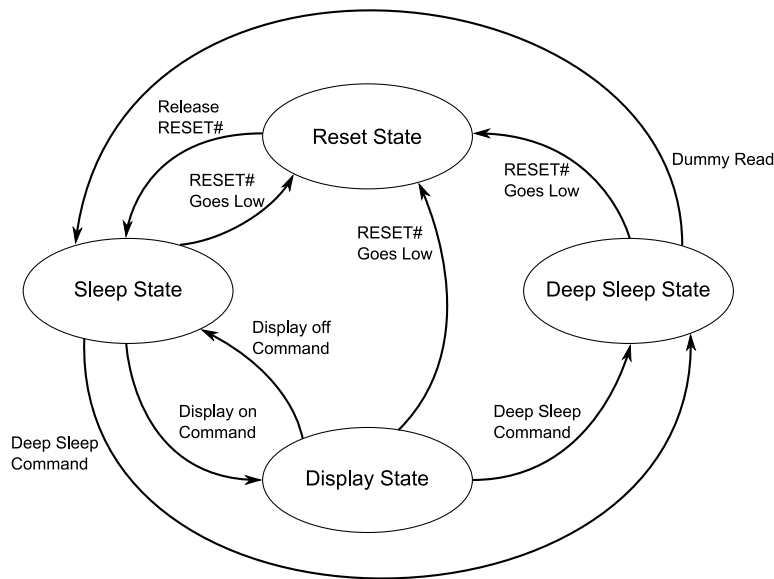
	Option 1 Horizontal: Increment Vertical: Increment B6 = 0; B7 = 0	Option 2 Horizontal: Increment Vertical: Decrement B6 = 0; B7 = 1	Option 3 Horizontal: Decrement Vertical: Increment B6 = 1; B7 = 0	Option 4 Horizontal: Decrement Vertical: Decrement B6 = 1; B7 = 1
Horizontal Frame Buffer Mode B5 = 0	00000h  4B000h	00000h  4B000h	00000h  4B000h	00000h  4B000h
Vertical Frame Buffer Mode B5 = 1	00000h  4B000h	00000h  4B000h	00000h  4B000h	00000h  4B000h

### 7.4 System Clock and Reset Manager

The “System Clock and Reset Manager” distributes the reset signal and clock signal to the entire system. It controls the Clock Generator and contains clock gating circuitry to turn on and off the clock of each functional module. Also, it divides the root clock from Clock Generator to operation clocks for different module. The System Clock and Reset Manager also manage the reset signals to ensure all the module are reset to appropriate status when the system are in reset state, deep sleep state, sleep state and display state. Figure 7-3 shows a state diagram of four operation states of SSD1961.



**Figure 7-3: State Diagram of SSD1961**



**Reset State:**

Clock Generator Stop  
 Unable to Receive Command  
 Unable to Update Frame Buffer  
 Display Off  
 All Settings Reset

**Deep Sleep State:**

Clock Generator Stop  
 Unable to Receive Command  
 Unable to Update Frame Buffer  
 Display Off  
 All Settings Retain

**Sleep State:**

Clock Generator On  
 Able to Receive Command  
 Able to Update Frame Buffer  
 Display Off  
 All Settings Retain

**Display State:**

Clock Generator On  
 Able to Receive Command  
 Able to Update Frame Buffer  
 Display On  
 All Settings Retain

## 7.5 LCD Controller

### 7.5.1 Display Format

The LCD controller reads the frame buffer and generates display signals according to the selected display panel format. SSD1961 supports common RAM-less TFT driver using generic RGB data.

### 7.5.2 General Purpose Input/Output (GPIO)

The GPIO pins can operate in 2 modes, GPIO mode and miscellaneous display signal mode. When the pins are configured as GPIOs, these pins can be controlled directly by MCU. Therefore, user can use these pins to emulate other interface such as SPI or I2C. If these pins are configured as display signals, they will toggle with display periodically according to the signal settings. They can be set to toggle once a frame, once a line or in arbitrary period. Therefore they can be configured as some common signal needed for different panels such as STH or LP.

## 8 COMMAND TABLE

Hex Code	Command	Description
0x00	nop	No operation
0x01	soft_reset	Software Reset
0x0A	get_power_mode	Get the current power mode
0x0B	get_address_mode	Get the frame buffer to the display panel read order
0x0C	Reserved	Reserved
0x0D	get_display_mode	The SSD1961 returns the Display Image Mode.
0x0E	get_tear_effect_status	Get the Tear Effect status
0x0F	Reserved	Reserved
0x10	enter_sleep_mode	Turn off the panel. This command will pull low the GPIO0. If GPIO0 is configured as normal GPIO or LCD miscellaneous signal with command set_gpio_conf, this command will be ignored.
0x11	exit_sleep_mode	Turn on the panel. This command will pull high the GPIO0. If GPIO0 is configured as normal GPIO or LCD miscellaneous signal with command set_gpio_conf, this command will be ignored.
0x12	enter_partial_mode	Part of the display area is used for image display.
0x13	enter_normal_mode	The whole display area is used for image display.
0x20	exit_invert_mode	Displayed image colors are not inverted.
0x21	enter_invert_mode	Displayed image colors are inverted.
0x26	set_gamma_curve	Selects the gamma curve used by the display panel.
0x28	set_display_off	Blanks the display panel
0x29	set_display_on	Show the image on the display panel
0x2A	set_column_address	Set the column address
0x2B	set_page_address	Set the page address
0x2C	write_memory_start	Transfer image information from the host processor interface to the SSD1961 starting at the location provided by set_column_address and set_page_address
0x2E	read_memory_start	Transfer image data from the SSD1961 to the host processor interface starting at the location provided by set_column_address and set_page_address
0x30	set_partial_area	Defines the partial display area on the display panel
0x33	set_scroll_area	Defines the vertical scrolling and fixed area on display area
0x34	set_tear_off	Synchronization information is not sent from the SSD1961 to the host processor
0x35	set_tear_on	Synchronization information is sent from the SSD1961 to the host processor at the start of VFP
0x36	set_address_mode	Set the read order from frame buffer to the display panel
0x37	set_scroll_start	Defines the vertical scrolling starting point
0x38	exit_idle_mode	Full color depth is used for the display panel
0x39	enter_idle_mode	Reduce color depth is used on the display panel.
0x3A	Reserved	Reserved
0x3C	write_memory_continue	Transfer image information from the host processor interface to the SSD1961 from the last written location
0x3E	read_memory_continue	Read image data from the SSD1961 continuing after the last read_memory_continue or read_memory_start

Hex Code	Command	Description
0x44	set_tear_scanline	Synchronization information is sent from the SSD1961 to the host processor when the display panel refresh reaches the provided scanline
0x45	get_scanline	Get the current scan line
0xA1	read_ddb	Read the DDB from the provided location
0xA8	Reserved	Reserved
0xB0	set_lcd_mode_	Set the LCD panel mode and resolution
0xB1	get_lcd_mode	Get the current LCD panel mode, pad strength and resolution
0xB4	set_hori_period	Set front porch
0xB5	get_hori_period	Get current front porch settings
0xB6	set_vert_period	Set the vertical blanking interval between last scan line and next LFRAME pulse
0xB7	get_vert_period	Set the vertical blanking interval between last scan line and next LFRAME pulse
0xB8	set_gpio_conf	Set the GPIO configuration. If the GPIO is not used for LCD, set the direction. Otherwise, they are toggled with LCD signals.
0xB9	get_gpio_conf	Get the current GPIO configuration
0xBA	set_gpio_value	Set GPIO value for GPIO configured as output
0xBB	get_gpio_status	Read current GPIO status. If the individual GPIO was configured as input, the value is the status of the corresponding pin. Otherwise, it is the programmed value.
0xBC	set_post_proc	Set the image post processor
0xBD	get_post_proc	Set the image post processor
0xBE	set_pwm_conf	Set the image post processor
0xBF	get_pwm_conf	Set the image post processor
0xC0	set_lcd_gen0	Set the rise, fall, period and toggling properties of LCD signal generator 0
0xC1	get_lcd_gen0	Get the current settings of LCD signal generator 0
0xC2	set_lcd_gen1	Set the rise, fall, period and toggling properties of LCD signal generator 1
0xC3	get_lcd_gen1	Get the current settings of LCD signal generator 1
0xC4	set_lcd_gen2	Set the rise, fall, period and toggling properties of LCD signal generator 2
0xC5	get_lcd_gen2	Get the current settings of LCD signal generator 2
0xC6	set_lcd_gen3	Set the rise, fall, period and toggling properties of LCD signal generator 3
0xC7	get_lcd_gen3	Get the current settings of LCD signal generator 3
0xC8	set_gpio0_rop	Set the GPIO0 with respect to the LCD signal generators using ROP operation. No effect if the GPIO0 is configured as general GPIO.
0xC9	get_gpio0_rop	Get the GPIO0 properties with respect to the LCD signal generators.
0xCA	set_gpio1_rop	Set the GPIO1 with respect to the LCD signal generators using ROP operation. No effect if the GPIO1 is configured as general GPIO.
0xCB	get_gpio1_rop	Get the GPIO1 properties with respect to the LCD signal generators.
0xCC	set_gpio2_rop	Set the GPIO2 with respect to the LCD signal generators using ROP operation. No effect if the GPIO2 is configured as general GPIO.

<b>Hex Code</b>	<b>Command</b>	<b>Description</b>
0xCD	get_gpio2_rop	Get the GPIO2 properties with respect to the LCD signal generators.
0xCE	set_gpio3_rop	Set the GPIO3 with respect to the LCD signal generators using ROP operation. No effect if the GPIO3 is configured as general GPIO.
0xCF	get_gpio3_rop	Get the GPIO3 properties with respect to the LCD signal generators.
0xD0	set_dbc_conf	Set the dynamic back light configuration
0xD1	get_dbc_conf	Get the current dynamic back light configuration
0xD4	set_dbc_th	Set the threshold for each level of power saving
0xD5	get_dbc_th	Get the threshold for each level of power saving
0xE0	set_pll	Start the PLL. Before the start, the system was operated with the crystal oscillator or clock input
0xE2	set_pll_mn	Set the PLL
0xE3	get_pll_mn	Get the PLL settings
0xE4	get_pll_status	Get the current PLL status
0xE5	set_deep_sleep	Set deep sleep mode
0xE6	set_lshift_freq	Set the LSHIFT (pixel clock) frequency
0xE7	get_lshift_freq	Get current LSHIFT (pixel clock) frequency setting
0xE8	Reserved	Reserved
0xE9	Reserved	Reserved
0xF0	set_pixel_data_interface	Set the pixel data format of the parallel host processor interface
0xF1	get_pixel_data_interface	Get the current pixel data format settings
0xFF	Reserved	Reserved

## 9 COMMAND DESCRIPTIONS

### 9.1 nop

**Command** 0x00  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	0	0	0	0	0	00

#### Description

No operation. .

### 9.2 soft\_reset

**Command** 0x01  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	0	0	0	0	1	01

#### Description

The SSD1961 performs a software reset. All the configuration register will be reset except command 0xE0 to 0xE5.

Note :

The host processor must wait 5ms before sending any new commands to a SSD1961 following this command.

### 9.3 get\_power\_mode

**Command** 0x0A  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	0	1	0	1	0	0A
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	0	0	xx

#### Description

Get the current power mode

A[6] : Idle mode on/off (POR = 0)  
0 Idle mode off  
1 Idle mode on

A[5] : Partial mode on/off (POR = 0)  
0 Partial mode off  
1 Partial mode on

A[4] : Sleep mode on/off (POR = 0)  
0 Sleep mode on  
1 Sleep mode off

A[3] : Display normal mode on/off (POR = 1)  
0 Display normal mode off

- 1 Display normal mode on (partial mode and vertical scroll off)

A[2] : Display on/off (POR = 0)

- 0 Display is off
- 1 Display is on

## 9.4 get\_address\_mode

**Command** 0x0B

**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	0	1	0	1	1	0B
Parameter 1	1	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	0	0	xx

### Description

Get the frame buffer to the display panel read order

A[7] : Page address order (POR = 0)

- 0 Top to bottom
- 1 Bottom to top

A[6] : Column address order (POR = 0)

- 0 Left to right
- 1 Right to left

A[5] : Page / Column order (POR = 0)

- 0 Normal mode
- 1 Reverse mode

A[4] : Line address order (POR = 0)

- 0 LCD refresh top to bottom
- 1 LCD refresh bottom to top

A[3] : RGB / BGR order (POR = 0)

- 0 RGB
- 1 BGR

A[2] : Display data latch data (POR = 0)

- 0 LCD refresh left to right
- 1 LCD refresh right to left

## 9.5 get\_display\_mode

**Command** 0x0D

**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	0	1	1	0	1	0D
Parameter 1	1	A <sub>7</sub>	0	A <sub>5</sub>	0	0	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx

### Description

The SSD1961 returns the Display Image Mode status.

A[7] : Vertical scrolling status on/off (POR = 0)

- 0 Vertical scrolling is off
- 1 Vertical scrolling is on

A[5] : Invert mode on/off (POR = 0)

- 0 Inversion is off
- 1 Inversion is on

A[2:0] : Gamma curve selection (POR = 011)

- 000 Gamma curve 0
- 001 Gamma curve 1
- 010 Gamma curve 2
- 011 Gamma curve 3
- 100 Reserved
- 101 Reserved
- 110 Reserved
- 111 Reserved

## 9.6 get\_tear\_effect\_status

**Command** 0x0E

**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	0	1	1	1	0	0E
Parameter 1	1	A <sub>7</sub>	0	0	0	0	0	0	0	xx

### Description

Get the current Tear Effect mode from the SSD1961

A[7] : Tearing effect line mode (POR = 0)

- 0 Tearing effect off
- 1 Tearing effect on

## 9.7 enter\_sleep\_mode

**Command** 0x10  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	1	0	0	0	0	10

### Description

Turn off the panel. This command causes the SSD1961 to enter sleep mode and pull low the GPIO[0] if set\_gpio\_conf (0xB8) B0 = 0.

If GPIO[0] is configured as normal GPIO or LCD miscellaneous signal with command set\_gpio\_conf (0xB8), this command will not affect the GPIO[0].

Note :

The host processor must wait 5ms before sending any new commands to a SSD1961 following this command.

## 9.8 exit\_sleep\_mode

**Command** 0x11  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	1	0	0	0	1	11

### Description

Turn on the panel. This command causes the SSD1961 to exit sleep mode and will pull high the GPIO[0] if set\_gpio\_conf (0xB8) B0 = 0.

If GPIO[0] is configured as normal GPIO or LCD miscellaneous signal with command set\_gpio\_conf (0xB8), this command will not affect the GPIO[0].

Note : The host processor must wait 5ms after sending this command before sending another command.

\*\*This command will automatic trigger set\_display\_on (0x29)

## 9.9 enter\_partial\_mode

**Command** 0x12  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	1	0	0	1	0	12

### Description

Once enter\_partial\_mode is triggered, the Partial Display Mode window is described by the set\_partial\_area (0x30).

Once enter\_normal\_mode (0x13) is triggered, partial display mode will end.

## 9.10 enter\_normal\_mode

**Command** 0x13  
**Parameters** None



	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	0	1	0	0	1	1	13

**Description**

This command causes the SSD1961 to enter the normal mode. Normal mode is defined as partial display and vertical scroll mode are off. That means the whole display area is used for image display.

**9.11 exit\_invert\_mode**

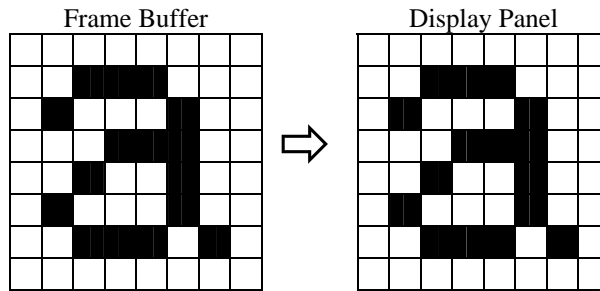
**Command** 0x20  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	0	0	0	0	0	20

**Description**

This command causes the SSD1961 to stop inverting the image data on the display panel. The frame buffer contents remain unchanged.

**Figure 9-1: Exit Invert mode example**



**9.12 enter\_invert\_mode**

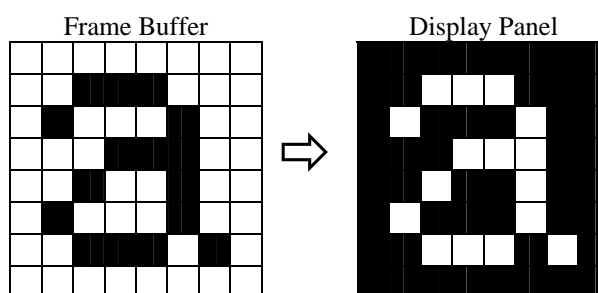
**Command** 0x21  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	0	0	0	0	1	21

**Description**

This command causes the SSD1961 to invert the image data only on the display panel. The frame buffer contents remain unchanged.

**Figure 9-2: Enter Invert mode example**



### 9.13 set\_gamma\_curve

**Command** 0x26  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	0	0	1	1	0	26
Parameter 1	1	0	0	0	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx

#### Description

Selects the gamma curve used by the display panel.

A[3:0]	Gamma curve selection (POR = 1000)	GAMAS[1]	GAMAS[0]
0000	No gamma curve selected (Same as 0001b)	0	0
0001	Gamma curve 0	0	0
0010	Gamma curve 1	0	1
0100	Gamma curve 2	1	0
1000	Gamma curve 3	1	1
Others	Reserved		

### 9.14 set\_display\_off

**Command** 0x28  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	0	1	0	0	0	28

#### Description

Blanks the display panel. The frame buffer contents remain unchanged.

### 9.15 set\_display\_on

**Command** 0x29  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	0	1	0	0	1	29

#### Description

Show the image on the display panel

## 9.16 set\_column\_address

**Command** 0x2A  
**Parameters** 4

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	0	1	0	1	0	2A
Parameter 1	1	SC <sub>15</sub>	SC <sub>14</sub>	SC <sub>13</sub>	SC <sub>12</sub>	SC <sub>11</sub>	SC <sub>10</sub>	SC <sub>9</sub>	SC <sub>8</sub>	xx
Parameter 2	1	SC <sub>7</sub>	SC <sub>6</sub>	SC <sub>5</sub>	SC <sub>4</sub>	SC <sub>3</sub>	SC <sub>2</sub>	SC <sub>1</sub>	SC <sub>0</sub>	xx
Parameter 3	1	EC <sub>15</sub>	EC <sub>14</sub>	EC <sub>13</sub>	EC <sub>12</sub>	EC <sub>11</sub>	EC <sub>10</sub>	EC <sub>9</sub>	EC <sub>8</sub>	xx
Parameter 4	1	EC <sub>7</sub>	EC <sub>6</sub>	EC <sub>5</sub>	EC <sub>4</sub>	EC <sub>3</sub>	EC <sub>2</sub>	EC <sub>1</sub>	EC <sub>0</sub>	xx

### Description

Set the column address of frame buffer accessed by the host processor with the read\_memory\_continue (0x3E) and write\_memory\_continue (0x3C)..

SC[15:8] : Start column number high byte (POR = 00000000)

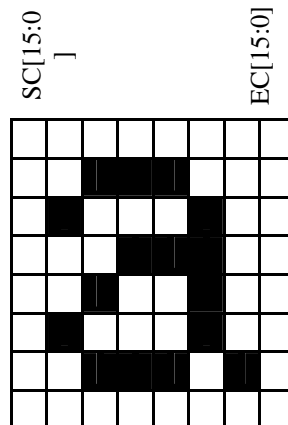
SC[7:0] : Start column number low byte (POR = 00000000)

EC[15:8] : End column number high byte (POR = 00000000)

EC[7:0] : End column number low byte (POR = 00000000)

Note : SC[15:0] must always be equal to or less than EC[15:0]

Figure 9-3: Set Column Address example



## 9.17 set\_page\_address

**Command** 0x2B  
**Parameters** 4

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	0	1	0	0	1	2B
Parameter 1	1	SP <sub>15</sub>	SP <sub>14</sub>	SP <sub>13</sub>	SP <sub>12</sub>	SP <sub>11</sub>	SP <sub>10</sub>	SP <sub>9</sub>	SP <sub>8</sub>	xx
Parameter 2	1	SP <sub>7</sub>	SP <sub>6</sub>	SP <sub>5</sub>	SP <sub>4</sub>	SP <sub>3</sub>	SP <sub>2</sub>	SP <sub>1</sub>	SP <sub>0</sub>	xx
Parameter 3	1	EP <sub>15</sub>	EP <sub>14</sub>	EP <sub>13</sub>	EP <sub>12</sub>	EP <sub>11</sub>	EP <sub>10</sub>	EP <sub>9</sub>	EP <sub>8</sub>	xx
Parameter 4	1	EP <sub>7</sub>	EP <sub>6</sub>	EP <sub>5</sub>	EP <sub>4</sub>	EP <sub>3</sub>	EP <sub>2</sub>	EP <sub>1</sub>	EP <sub>0</sub>	xx

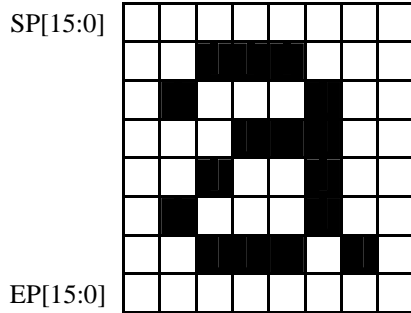
### Description

Set the page address of the frame buffer accessed by the host processor with the read\_memory\_start (0x2C), write\_memory\_start (0x2E), read\_memory\_continue (0x3E) and write\_memory\_continue (0x3C)..

SP[15:8] : Start page (row) number high byte (POR = 00000000)  
 SP[7:0] : Start page (row) number low byte (POR = 00000000)  
 EP[15:8] : End page (row) number high byte (POR = 00000000)  
 EP[7:0] : End page (row) number low byte (POR = 00000000)

Note : SP[15:0] must always be equal to or less than EP[15:0]

**Figure 9-4: Set Page Address example**



### 9.18 write\_memory\_start

**Command** 0x2C  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	0	1	1	0	0	2C

#### Description

Transfer image information from the host processor interface to the SSD1961 starting at the location provided by set\_column\_address (0x2A) and set\_page\_address (0x2B).

If set\_address\_mode (0x36) A[5] = 0:

The column and page address are reset to the Start Column (SC) and Start Page (SP), respectively.

Pixel Data 1 is stored in frame buffer at (SC, SP). The column address is then incremented and pixels are written to the frame buffer until the column address equals the End Column (EC) value. The column address is then reset to SC and the page address is incremented. Pixels are written to the frame buffer until the page address equals the End Page (EP) value and the column address equals the EC value, or the host processor sends another command. If the number of pixels exceeds (EC - SC + 1) \* (EP - SP + 1) the extra pixels are ignored.

If set\_address\_mode (0x36) A[5] = 1:

The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

Pixel Data 1 is stored in frame buffer at (SC, SP). The page address is then incremented and pixels are written to the frame buffer until the page address equals the End Page (EP) value. The page address is then reset to SP and the column address is incremented. Pixels are written to the frame buffer until the column address equals the End column (EC) value and the page address equals the EP value, or the host processor sends another command. If the number of pixels exceeds (EC - SC + 1) \* (EP - SP + 1) the extra pixels are ignored.

## 9.19 read\_memory\_start

**Command** 0x2E  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	0	1	1	1	0	2E

### Description

Transfer image data from the SSD1961 to the host processor interface starting at the location provided by set\_column\_address (0x2A) and set\_page\_address (0x2B).

If set\_address\_mode A[5] = 0:

The column and page address are reset to the Start Column (SC) and Start Page (SP), respectively.

Pixels Data are read from frame buffer at (SC, SP). The column address is then incremented and pixels read from the frame buffer until the column address equals the End Column (EC) value. The column address is then reset to SC and the page address is incremented. Pixels are read from the frame buffer until the page address equals the End Page (EP) value and the column address equals the EC value, or the host processor sends another command.

If set\_address\_mode (0x36) A[5] = 1:

The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

Pixels Data are read from frame buffer at (SC, SP). The page address is then incremented and pixels read from the frame buffer until the page address equals the End Page (EP) value. The page address is then reset to SP and the column address is incremented. Pixels are read from the frame buffer until the column address equals the End Column (EC) value and the page address equals the EP value, or the host processor sends another command.

## 9.20 set\_partial\_area

**Command** 0x30  
**Parameters** 4

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	0	0	0	0	30
Parameter 1	1	SR <sub>15</sub>	SR <sub>14</sub>	SR <sub>13</sub>	SR <sub>12</sub>	SR <sub>11</sub>	SR <sub>10</sub>	SR <sub>9</sub>	SR <sub>8</sub>	xx
Parameter 2	1	SR <sub>7</sub>	SR <sub>6</sub>	SR <sub>5</sub>	SR <sub>4</sub>	SR <sub>3</sub>	SR <sub>2</sub>	SR <sub>1</sub>	SR <sub>0</sub>	xx
Parameter 3	1	ER <sub>15</sub>	ER <sub>14</sub>	ER <sub>13</sub>	ER <sub>12</sub>	ER <sub>11</sub>	ER <sub>10</sub>	ER <sub>9</sub>	ER <sub>8</sub>	xx
Parameter 4	1	ER <sub>7</sub>	ER <sub>6</sub>	ER <sub>5</sub>	ER <sub>4</sub>	ER <sub>3</sub>	ER <sub>2</sub>	ER <sub>1</sub>	ER <sub>0</sub>	xx

### Description

This command defines the Partial Display mode's display area. There are two parameters associated with this command, the first defines the Start Row (SR) and the second the End Row (ER). SR and ER refer to the Frame Buffer Line Pointer.

SR[15:8] : Start display row number high byte (POR = 00000000)

SR[7:0] : Start display row number low byte (POR = 00000000)

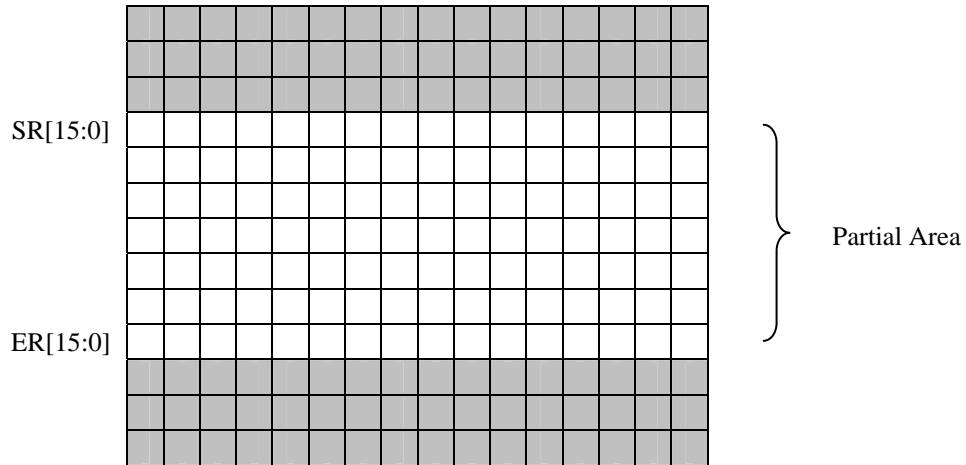
ER[15:8] : End display row number high byte (POR = 00000000)

ER[7:0] : End display row number low byte (POR = 00000000)

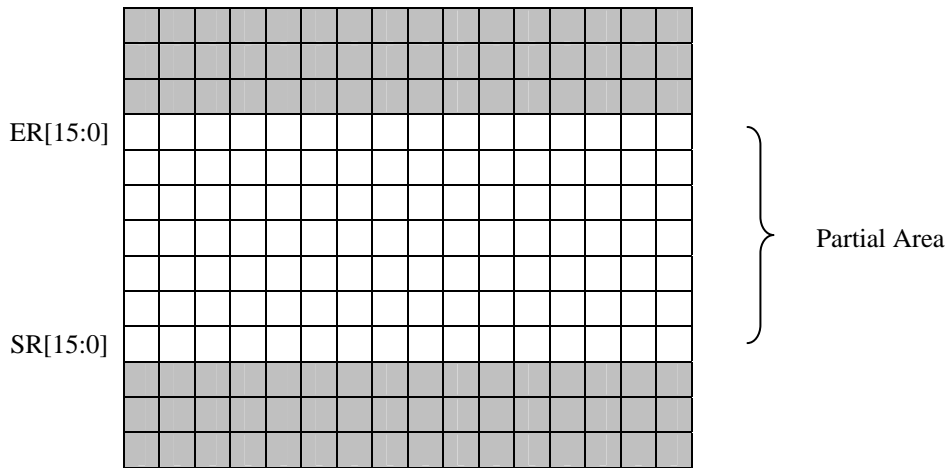
Note : SR[15:0] and ER[15:0] cannot be 0000h nor exceed the last vertical line number.

If End Row > Start Row

**Figure 9-5: Set Partial Area with set\_address\_mode (0x36) A[4] = 0 when End Row > Start Row**

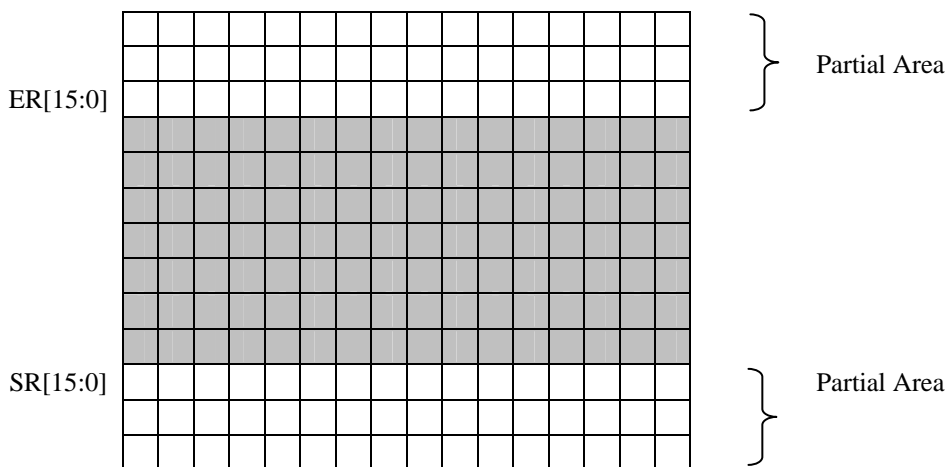


**Figure 9-6: Set Partial Area with set\_address\_mode (0x36) A[4] = 1 when End Row > Start Row**

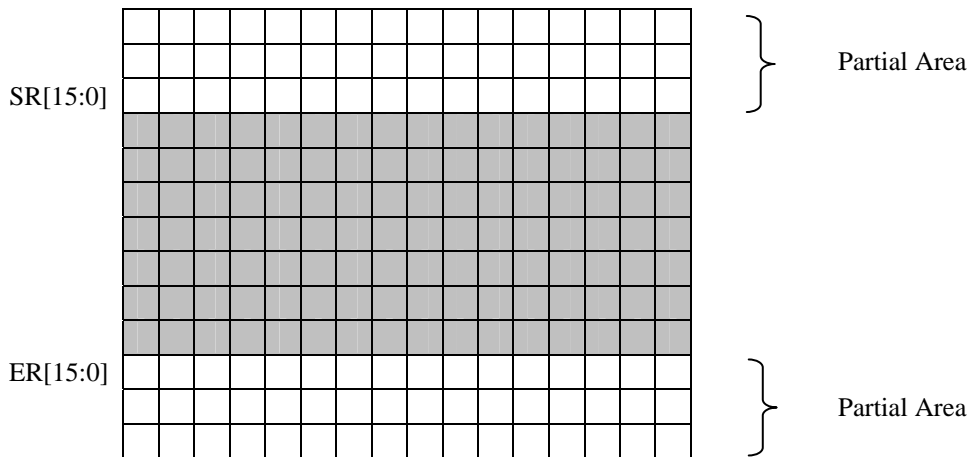


If Start Row > End Row

**Figure 9-7: Set Partial Area with set\_address\_mode (0x36) A[4] = 0 when Start Row > End Row**



**Figure 9-8: Set Partial Area with set\_address\_mode (0x36) A[4] = 1 when Start Row > End Row**



## 9.21 set\_scroll\_area

**Command** 0x33  
**Parameters** 6

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	0	0	1	1	33
Parameter 1	1	TFA <sub>15</sub>	TFA <sub>14</sub>	TFA <sub>13</sub>	TFA <sub>12</sub>	TFA <sub>11</sub>	TFA <sub>10</sub>	TFA <sub>9</sub>	TFA <sub>8</sub>	xx
Parameter 2	1	TFA <sub>7</sub>	TFA <sub>6</sub>	TFA <sub>5</sub>	TFA <sub>4</sub>	TFA <sub>3</sub>	TFA <sub>2</sub>	TFA <sub>1</sub>	TFA <sub>0</sub>	xx
Parameter 3	1	VSA <sub>15</sub>	VSA <sub>14</sub>	VSA <sub>13</sub>	VSA <sub>12</sub>	VSA <sub>11</sub>	VSA <sub>10</sub>	VSA <sub>9</sub>	VSA <sub>8</sub>	xx
Parameter 4	1	VSA <sub>7</sub>	VSA <sub>6</sub>	VSA <sub>5</sub>	VSA <sub>4</sub>	VSA <sub>3</sub>	VSA <sub>2</sub>	VSA <sub>1</sub>	VSA <sub>0</sub>	xx
Parameter 5	1	BFA <sub>15</sub>	BFA <sub>14</sub>	BFA <sub>13</sub>	BFA <sub>12</sub>	BFA <sub>11</sub>	BFA <sub>10</sub>	BFA <sub>9</sub>	BFA <sub>8</sub>	xx
Parameter 6	1	BFA <sub>7</sub>	BFA <sub>6</sub>	BFA <sub>5</sub>	BFA <sub>4</sub>	BFA <sub>3</sub>	BFA <sub>2</sub>	BFA <sub>1</sub>	BFA <sub>0</sub>	xx

### Description

Defines the vertical scrolling and fixed area on display area

TFA[15:8] : High byte of Top Fixed Area number in lines from the top of the frame buffer (POR = 00000000)

TFA[7:0] : Low byte of Top Fixed Area number in lines from the top of the frame buffer (POR = 00000000)

VSA[15:8] : High byte of Vertical scrolling area in number of lines of the frame buffer (POR = 00000000)

VSA[7:0] : Low byte of Vertical scrolling area in number of lines of the frame buffer (POR = 00000000)

BFA[15:8] : High byte of Bottom Fixed Area in number of lines from the bottom of the frame buffer (POR = 00000000)

BFA[7:0] : Low byte of Bottom Fixed Area in number of lines from the bottom of the frame buffer (POR = 00000000)

If set\_address\_mode (0x36) A[4] = 0 :

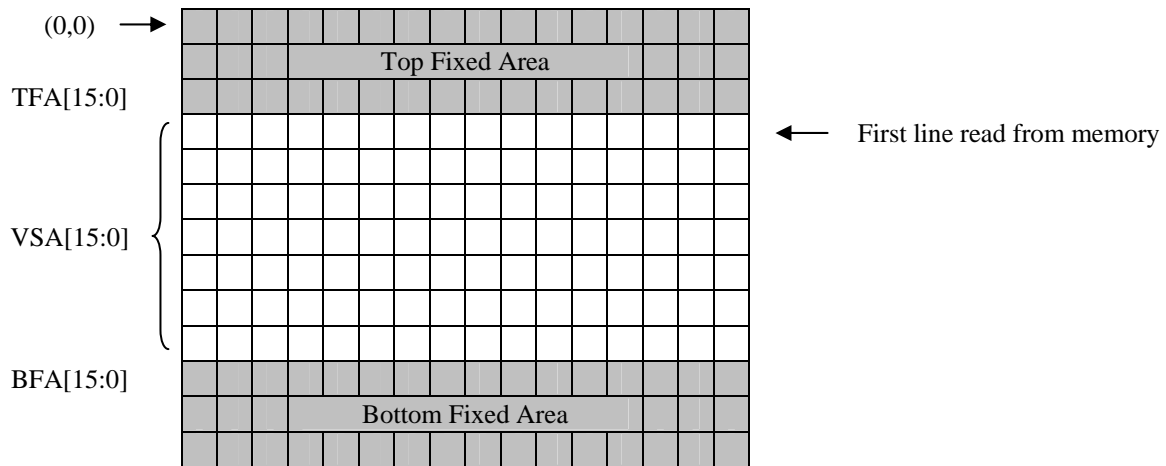
The TFA[15:0] describes the Top Fixed Area in number of lines from the top of the frame buffer. The top of the frame buffer and top of the display panel are aligned.

The VSA[15:0] describes the height of the Vertical Scrolling Area in number of lines of frame buffer from the Vertical Scrolling Start Address. The first line of the Vertical Scrolling Area starts immediately after the bottom most line of the Top Fixed Area. The last line of the Vertical Scrolling Area ends immediately before the top most line of the Bottom Fixed Area.

The BFA[15:0] describes the Bottom Fixed Area in number of lines from the bottom of the frame buffer. The bottom of the frame buffer and bottom of the display panel are aligned.

TFA, VSA and BFA refer to the Frame Buffer Line Pointer.

**Figure 9-9: Set Scroll Area with set\_address\_mode (0x36) A[4] = 0**



If set\_address\_mode (0x36) A[4] = 1 :

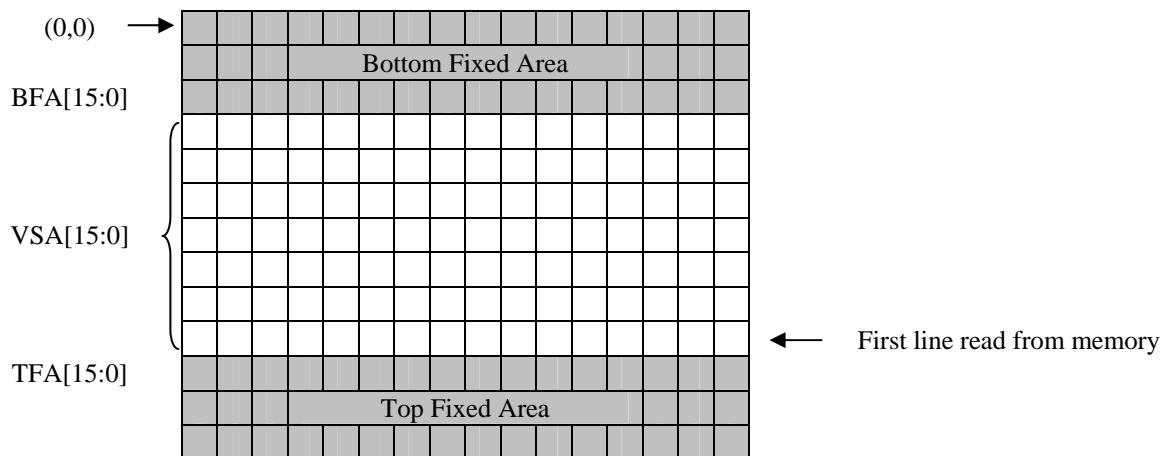
The TFA[15:0], describes the Top Fixed Area in number of lines from the bottom of the frame buffer. The bottom of the frame buffer and bottom of the display panel are aligned.

The VSA[15:0] describes the height of the Vertical Scrolling Area in number of lines of frame buffer from the Vertical Scrolling Start Address. The first line of the Vertical Scrolling Area starts immediately after the top most line of the Top Fixed Area. The last line of the Vertical Scrolling Area ends immediately before the bottom most line of the Bottom Fixed Area.

The BFA[15:0] describes the Bottom Fixed Area in number of lines from the top of the frame buffer. The top of the frame buffer and top of the display panel are aligned.

TFA, VSA and BFA refer to the Frame Buffer Line Pointer.

**Figure 9-10: Set Scroll Area with set\_address\_mode (0x36) A[4] = 1**



Note :

The sum of TFA, VSA and BFA must equal the number of the display panel's horizontal lines (pages), otherwise Scrolling mode is undefined.

In Vertical Scroll Mode, set\_address\_mode (0x36) A[5] should be set to '0' – this only affects the Frame Buffer Write.



## 9.22 set\_tear\_off

**Command** 0x34  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	0	1	0	0	34

### Description

TE signal is not sent from the SSD1961 to the host processor.

## 9.23 set\_tear\_on

**Command** 0x35  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	0	1	0	1	35
Parameter 1	1	0	0	0	0	0	0	0	A <sub>0</sub>	xx

### Description

TE signal is sent from the SSD1961 to the host processor at the start of VFP.

A[0] : Tearing effect line mode (POR = 0)

- 0 The tearing effect output line consists of V-blanking information only.
- 1 The tearing effect output line consists of both V-blanking and H-blanking information by set\_tear\_scanline (0x44)..

The TE signal shall be active low when the display panel is in Sleep mode.

## 9.24 set\_address\_mode

**Command** 0x36  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	0	1	1	0	36
Parameter 1	1	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx

### Description

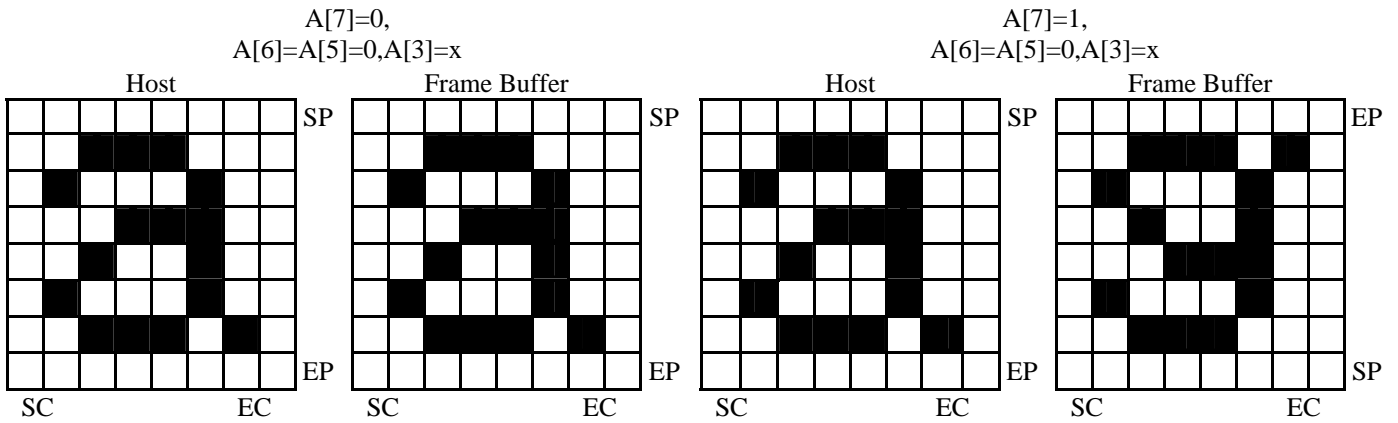
Set the read order from host processor to frame buffer by A[7:5] and A[3] and from frame buffer to the display panel by A[2:0] and A[4].

A[7] : Page address order (POR = 0)

This bit controls the order that pages of data are transferred from the host processor to the SSD1961's frame buffer.

- 0 Top to bottom, pages transferred from SP (Start Page) to EP (End Page).
- 1 Bottom to top, pages transferred from EP (End Page) to SP (Start Page).

**Figure 9-11: A[7] Page Address Order**

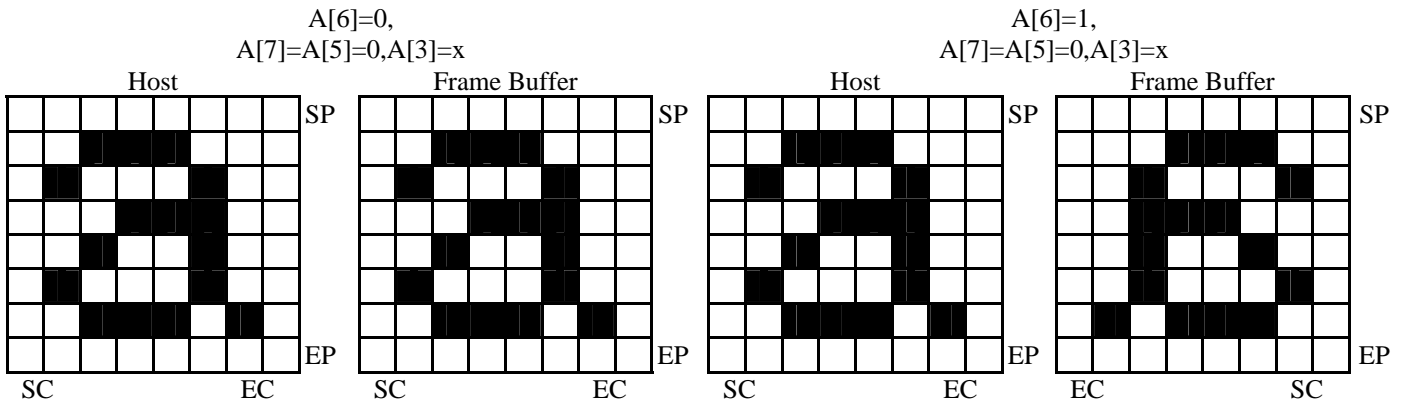


A[6] : Column address order (POR = 0)

This bit controls the order that columns of data are transferred from the host processor to the SSD1961's frame buffer.

- 0 Left to right, columns transferred from SC (Start Column) to EC (End Column).
- 1 Right to left, columns transferred from EC (End Column) to SC (Start Column).

**Figure 9-12: A[6] Column Address Order**

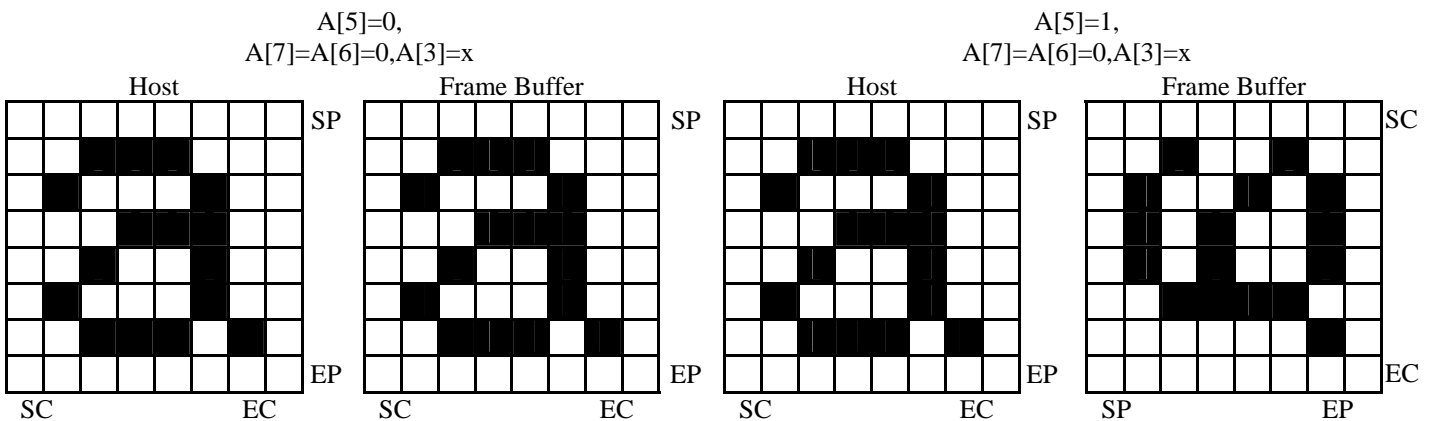


A[5] : Page / Column order (POR = 0)

This bit controls the order that columns of data are transferred from the host processor to the SSD1961's frame buffer.

- 0 Normal mode
- 1 Reverse mode

**Figure 9-13: A[5] Page / Column Address Order**



A[4] : Line address order (POR = 0)

This bit controls the display panel's horizontal line refresh order. The image shown on the display panel is unaffected, regardless of the bit setting.

- 0 LCD refresh from top line to bottom line.
- 1 LCD refresh from bottom line to top line.

A[3] : RGB / BGR order (POR = 0)

This bit controls the RGB data order transferred from the SSD1961's frame buffer to the display panel.

- 0 RGB
- 1 BGR

**Figure 9-14: A[3] RGB Order**



A[2] : Display data latch data (POR = 0)

This bit controls the display panel's vertical line data latch order. The image shown on the display panel is unaffected, regardless of the bit setting.

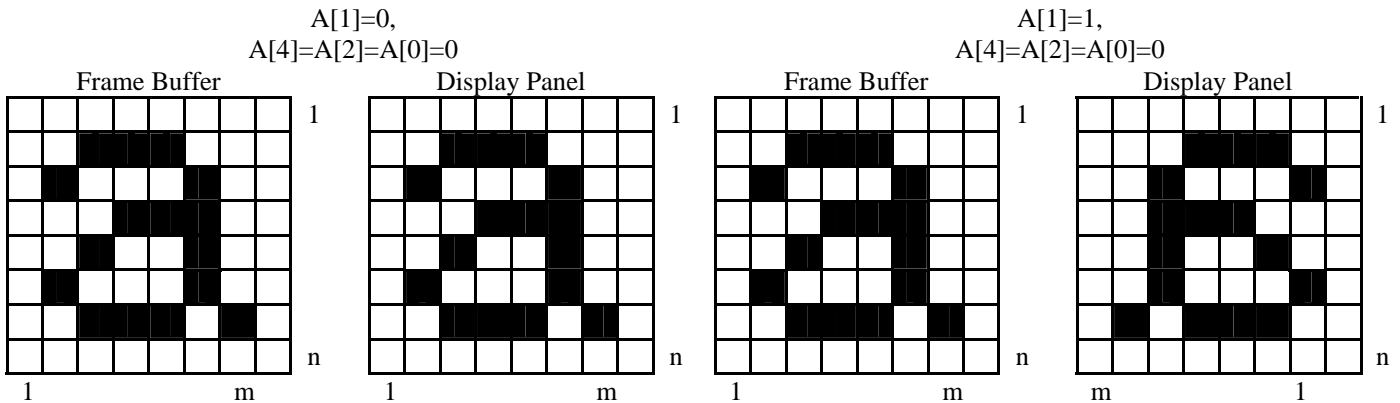
- 0 LCD refresh from left side to right side
- 1 LCD refresh from right side to left side

A[1] : Flip Horizontal (POR = 0)

This bit flips the image shown on the display panel left to right. No change is made to the frame buffer.

- 0 Normal
- 1 Flipped

**Figure 9-15: A[1] Flip Horizontal**

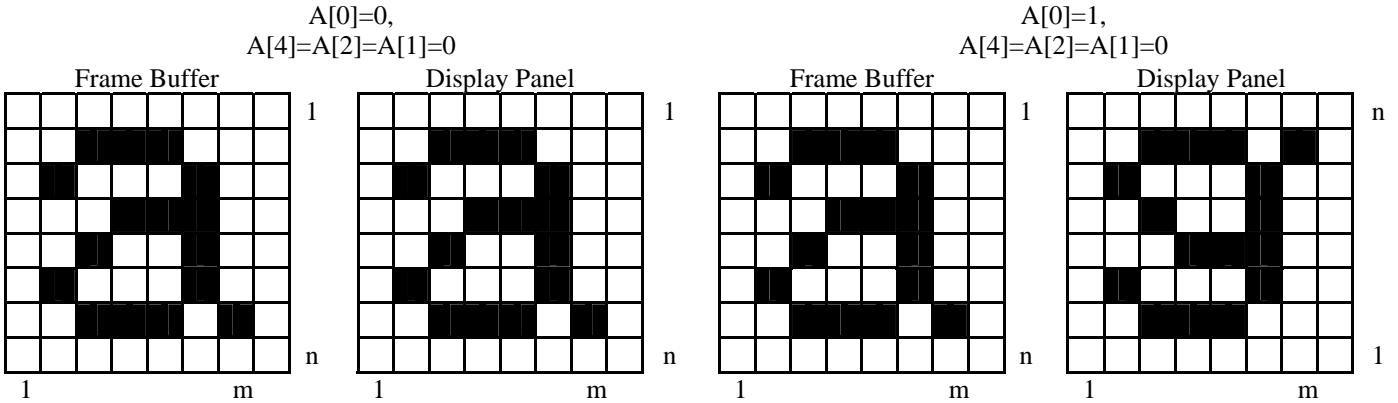


A[0] : Flip Vertical (POR = 0)

This bit flips the image shown on the display panel top to bottom. No change is made to the frame buffer.

- 0 Normal
- 1 Flipped

**Figure 9-16: A[0] Flip Vertical**



**9.25 set\_scroll\_start**

**Command** 0x37  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	0	1	1	1	37
Parameter 1	1	VSP <sub>15</sub>	VSP <sub>14</sub>	VSP <sub>13</sub>	VSP <sub>12</sub>	VSP <sub>11</sub>	VSP <sub>10</sub>	VSP <sub>9</sub>	VSP <sub>8</sub>	xx
Parameter 2	1	VSP <sub>7</sub>	VSP <sub>6</sub>	VSP <sub>5</sub>	VSP <sub>4</sub>	VSP <sub>3</sub>	VSP <sub>2</sub>	VSP <sub>1</sub>	VSP <sub>0</sub>	xx

**Description**

This command sets the start of the vertical scrolling area in the frame buffer. The vertical scrolling area is fully defined when this command is used with the set\_scroll\_area (0x33).

VSP[15:8] : High byte of the line number in frame buffer that is written to the display as the first line of the vertical scrolling area (POR = 00000000)

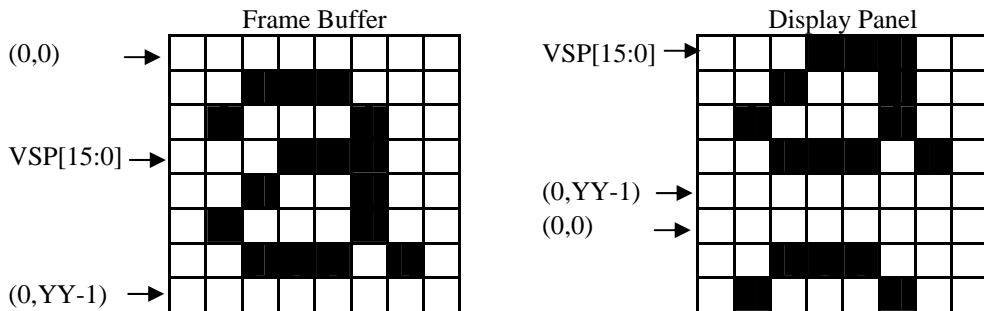
VSP[7:0] : Low byte of the line number in frame buffer that is written to the display as the first line of the vertical scrolling area (POR = 00000000)

If set\_address\_mode (0x36) A[4] = 0:

Example:

When Top Fixed Area = Bottom Fixed Area = 0, Vertical Scrolling Area = YY and VSP = 3.

**Figure 9-17: Set Scroll Start with set\_address\_mode (0x36) A[4] = 0**

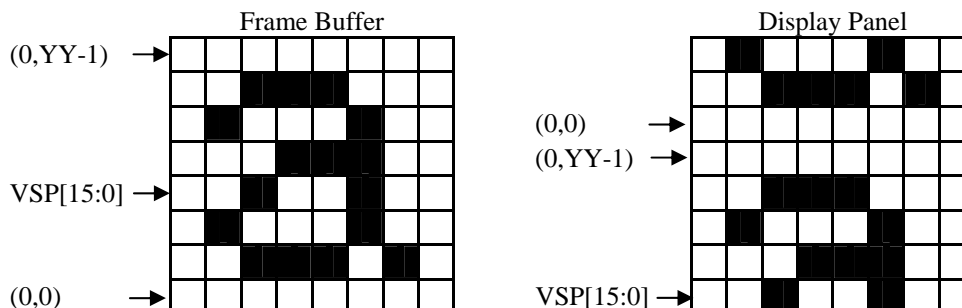


If set\_address\_mode (0x36) A[4] = 1:

Example:

When Top Fixed Area = Bottom Fixed Area = 0, Vertical Scrolling Area = YY and VSP = 3.

**Figure 9-18: Set Scroll Start with set\_address\_mode (0x36) A[4] = 1**



Note :

If set\_address\_mode, (0x36) A[4] = 0, TFA[15:0] - 1 < VSP[15:0] < # of lines in frame buffer - BFA[15:0]

If set\_address\_mode, (0x36) A[4] = 1, BFA[15:0] - 1 < VSP[15:0] < # of lines in frame buffer - TFA[15:0]

## 9.26 exit\_idle\_mode

**Command** 0x38

**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	1	0	0	0	38

### Description

This command causes the SSD1961 to exit Idle Mode.

Full color depth is used for the display panel.

## 9.27 enter\_idle\_mode

**Command** 0x39

**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	1	0	0	1	39

### Description

This command causes the SSD1961 to enter Idle Mode.

In Idle Mode, color depth is reduced. Colors are shown on the display panel using the MSB of each of the R, G and B color components in the frame buffer.

**Table 9-1: Enter Idle Mode memory content vs display color**

Color	R <sub>7</sub> R <sub>6</sub> R <sub>5</sub> R <sub>4</sub> R <sub>3</sub> R <sub>2</sub> R <sub>1</sub> R <sub>0</sub>	G <sub>7</sub> G <sub>6</sub> G <sub>5</sub> G <sub>4</sub> G <sub>3</sub> G <sub>2</sub> G <sub>1</sub> G <sub>0</sub>	B <sub>7</sub> B <sub>6</sub> B <sub>5</sub> B <sub>4</sub> B <sub>3</sub> B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>
Black	0XXXXXXXX	0XXXXXXXX	0XXXXXXXX
Blue	0XXXXXXXX	0XXXXXXXX	1XXXXXXXX
Red	1XXXXXXXX	0XXXXXXXX	0XXXXXXXX
Magenta	1XXXXXXXX	0XXXXXXXX	1XXXXXXXX
Green	0XXXXXXXX	1XXXXXXXX	0XXXXXXXX
Cyan	0XXXXXXXX	1XXXXXXXX	1XXXXXXXX
Yellow	1XXXXXXXX	1XXXXXXXX	0XXXXXXXX
White	1XXXXXXXX	1XXXXXXXX	1XXXXXXXX

## 9.28 write\_memory\_continue

**Command** 0x3C  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	1	1	0	0	3C

### Description

Transfer image information from the host processor interface to the SSD1961 from the last write\_memory\_continue (0x3C) or write\_memory\_start (0x2C).

If set\_address\_mode (0x36) A[5] = 0:

Data is written continuing from the pixel location after the write range of the previous write\_memory\_start (0x2C) or write\_memory\_continue (0x3C). The column address is then incremented and pixels are written to the frame buffer until the column address equals the End Column (EC) value. The column address is then reset to SC and the page address is incremented. Pixels are written to the frame buffer until the page address equals the End Page (EP) value and the column register equals the EC value, or the host processor sends another command. If the number of pixels exceeds (EC – SC + 1) \* (EP – SP + 1) the extra pixels are ignored.

If set\_address\_mode (0x36) A[5] = 1:

Data is written continuing from the pixel location after the write range of the previous write\_memory\_start (0x2C) or write\_memory\_continue (0x3C). The page address is then incremented and pixels are written to the frame buffer until the page register equals the End Page (EP) value. The page address is then reset to SP and the column address is incremented. Pixels are written to the frame buffer until the column register equals the End column (EC) value and the page address equals the EP value, or the host processor sends another command. If the number of pixels exceeds (EC – SC + 1) \* (EP – SP + 1) the extra pixels are ignored.

## 9.29 read\_memory\_continue

**Command** 0x3E  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	0	1	1	1	1	1	0	3E

### Description

Read image data from the SSD1961 to host processor continuing after the last read\_memory\_continue (0x3E) or read\_memory\_start (0x2E).

If set\_address\_mode (0x36) A[5] = 0:

Pixels are read continuing from the pixel location after the read range of the previous read\_memory\_start (0x2E) or read\_memory\_continue (0x3E). The column address is then incremented and pixels are read from the frame buffer until the column address equals the End Column (EC) value. The column address is then reset to SC and the page address is incremented. Pixels are read from the frame buffer until the page address equals the End Page (EP) value and the column address equals the EC value, or the host processor sends another command.

If set\_address\_mode (0x36) A[5] = 1:

Pixels are read continuing from the pixel location after the read range of the previous read\_memory\_start (0x2E) or read\_memory\_continue (0x3E). The page address is then incremented and pixels are read from the frame buffer until the page address equals the End Page (EP) value. The page address is then reset to SP and the column address is incremented. Pixels are read from the frame buffer until the column address equals the End Column (EC) value and the page address equals the EP value, or the host processor sends another command.

### 9.30 set\_tear\_scanline

**Command** 0x44  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	1	0	0	0	1	0	0	44
Parameter 1	1	N <sub>15</sub>	N <sub>14</sub>	N <sub>13</sub>	N <sub>12</sub>	N <sub>11</sub>	N <sub>10</sub>	N <sub>9</sub>	N <sub>8</sub>	xx
Parameter 2	1	N <sub>7</sub>	N <sub>6</sub>	N <sub>5</sub>	N <sub>4</sub>	N <sub>3</sub>	N <sub>2</sub>	N <sub>1</sub>	N <sub>0</sub>	xx

#### Description

TE signal is sent from the SSD1961 to the host processor when the display panel refresh reaches the provided scanline, N.

N[15:8] : High byte of the scanline (POR = 00000000)

N[7:0] : Low byte of the scanline (POR = 00000000)

Note :

Valid setting for TE signal: 0x0000, 0x0002 to 0xFFFF.

The number of Tear Scanline = N[15:0] +1, except N = 0.

Set Tear Scanline with N = 0 is equivalent to set\_tear\_on (0x35) A[0] = 0.

When Tear Scanline, N >= Vertical panel size, TE signal will always pull high.

Program set\_tear\_scanline will automatic change the operating mode of set\_tear\_on (0x35) A[0] = 1.

This command takes affect on the frame following the current frame. Therefore, if the Tear Effect (TE) signal is already ON, the TE output shall continue to operate as programmed by the previous set\_tear\_on (0x35) or set\_tear\_scanline (0x44) until the end of the frame.

### 9.31 get\_scanline

**Command** 0x45  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	0	1	0	0	0	1	0	1	45
Parameter 1	1	N <sub>15</sub>	N <sub>14</sub>	N <sub>13</sub>	N <sub>12</sub>	N <sub>11</sub>	N <sub>10</sub>	N <sub>9</sub>	N <sub>8</sub>	xx
Parameter 2	1	N <sub>7</sub>	N <sub>6</sub>	N <sub>5</sub>	N <sub>4</sub>	N <sub>3</sub>	N <sub>2</sub>	N <sub>1</sub>	N <sub>0</sub>	xx

#### Description

Get the current scan line, N.

N[15:8] : High byte of the current scanline (POR = 00000000)

N[7:0] : Low byte of the current scanline (POR = 00000000)



### 9.32 read\_ddb

**Command** 0xA1  
**Parameters** 5

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	0	0	0	0	1	A1
Parameter 1	1	SSL <sub>15</sub>	SSL <sub>14</sub>	SSL <sub>13</sub>	SSL <sub>12</sub>	SSL <sub>11</sub>	SSL <sub>10</sub>	SSL <sub>9</sub>	SSL <sub>8</sub>	xx
Parameter 2	1	SSL <sub>7</sub>	SSL <sub>6</sub>	SSL <sub>5</sub>	SSL <sub>4</sub>	SSL <sub>3</sub>	SSL <sub>2</sub>	SSL <sub>1</sub>	SSL <sub>0</sub>	xx
Parameter 3	1	PROD <sub>7</sub>	PROD <sub>6</sub>	PROD <sub>5</sub>	PROD <sub>4</sub>	PROD <sub>3</sub>	PROD <sub>2</sub>	PROD <sub>1</sub>	PROD <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	REV <sub>2</sub>	REV <sub>1</sub>	REV <sub>0</sub>	xx
Parameter 5	1	1	1	1	1	1	1	1	1	FF

#### Description

Read the DDB (Device Descriptor Block) information of SSD1961.

SSL[15:8] : Supplier ID of Solomon Systech Limited high byte, always 01h (POR = 00000001)

SSL[7:0] : Supplier ID of Solomon Systech Limited low byte, always 57h (POR = 01010111)

PROD[7:0] : Product ID, always 61h (POR = 01100001)

REV[2:0] : Revision code, always 01h (POR = 001)

Exit code, always FFh (POR = 11111111)

### 9.33 set\_lcd\_mode

**Command** 0xB0  
**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	0	0	0	0	B0
Parameter 1	1	0	0	0	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	0	B <sub>6</sub>	B <sub>5</sub>	0	0	0	0	0	xx
Parameter 3	1	0	0	0	0	0	HDP <sub>10</sub>	HDP <sub>9</sub>	HDP <sub>8</sub>	xx
Parameter 4	1	HDP <sub>7</sub>	HDP <sub>6</sub>	HDP <sub>5</sub>	HDP <sub>4</sub>	HDP <sub>3</sub>	HDP <sub>2</sub>	HDP <sub>1</sub>	HDP <sub>0</sub>	xx
Parameter 5	1	0	0	0	0	0	VDP <sub>10</sub>	VDP <sub>9</sub>	VDP <sub>8</sub>	xx
Parameter 6	1	VDP <sub>7</sub>	VDP <sub>6</sub>	VDP <sub>5</sub>	VDP <sub>4</sub>	VDP <sub>3</sub>	VDP <sub>2</sub>	VDP <sub>1</sub>	VDP <sub>0</sub>	xx
Parameter 7	1	0	0	G <sub>5</sub>	G <sub>4</sub>	G <sub>3</sub>	G <sub>2</sub>	G <sub>1</sub>	G <sub>0</sub>	xx

#### Description

Set the LCD panel mode and resolution

A[4] : TFT color depth enhancement enable (POR = 0)

0 Disable FRC or dithering

1 Enable FRC or dithering for color depth enhancement

A[3] : TFT FRC enable (POR = 0)

0 TFT dithering enable

1 TFT FRC enable

A[4]	A[3]	TFT FRC	TFT dithering
0	X	Disable	Disable
1	0	Disable	Enable
1	1	Enable	Disable

A[2] : LSHIFT polarity (POR = 0)

Set the dot clock pulse polarity.

0	Data latch in falling edge
1	Data latch in rising edge

A[1] : LLINE polarity (POR = 0)

Set the horizontal sync pulse polarity.

0	Active low
1	Active high

A[0] : LFRAME polarity (POR = 0)

Set the vertical sync pulse polarity.

0	Active low
1	Active high

B[6:5] : TFT type (POR = 01)

00, 01	TFT mode
10	Serial RGB mode
11	Serial RGB+dummy mode

HDP[10:8] : High byte of the horizontal panel size (POR = 010)

HDP[7:0] : Low byte of the horizontal panel size (POR = 01111111)

Horizontal panel size = (HDP + 1) pixels

VDP[10:8] : High byte of the vertical panel size (POR = 001)

VDP[7:0] : Low byte of the vertical panel size (POR = 11011111)

Vertical panel size = (VDP + 1) lines

G[5:3] : Even line RGB sequence for serial TFT interface (POR = 000)

000	RGB
001	RBG
010	GRB
011	GBR
100	BRG
101	BGR
11x	Reserved

G[2:0] : Odd line RGB sequence for serial TFT interface (POR = 000)

000	RGB
001	RBG
010	GRB
011	GBR
100	BRG
101	BGR
11x	Reserved

## 9.34 get\_lcd\_mode

**Command** 0xB1  
**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	0	0	0	1	B1
Parameter 1	1	0	0	0	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	0	B <sub>6</sub>	B <sub>5</sub>	0	0	0	0	0	xx
Parameter 3	1	0	0	0	0	0	HDP <sub>10</sub>	HDP <sub>9</sub>	HDP <sub>8</sub>	xx
Parameter 4	1	HDP <sub>7</sub>	HDP <sub>6</sub>	HDP <sub>5</sub>	HDP <sub>4</sub>	HDP <sub>3</sub>	HDP <sub>2</sub>	HDP <sub>1</sub>	HDP <sub>0</sub>	xx
Parameter 5	1	0	0	0	0	0	VDP <sub>10</sub>	VDP <sub>9</sub>	VDP <sub>8</sub>	xx
Parameter 6	1	VDP <sub>7</sub>	VDP <sub>6</sub>	VDP <sub>5</sub>	VDP <sub>4</sub>	VDP <sub>3</sub>	VDP <sub>2</sub>	VDP <sub>1</sub>	VDP <sub>0</sub>	xx
Parameter 7	1	0	0	G <sub>5</sub>	G <sub>4</sub>	G <sub>3</sub>	G <sub>2</sub>	G <sub>1</sub>	G <sub>0</sub>	xx

### Description

Get the current LCD panel mode and resolution

A[4] : TFT color depth enhancement enable(POR = 0)

- 0 Disable FRC or dithering
- 1 Enable FRC or dithering for color depth enhancement

A[3] : TFT FRC enable (POR = 0)

- 0 TFT dithering enable
- 1 TFT FRC enable

A[2] : LSHIFT polarity (POR = 0)

The dot clock pulse polarity.

- 0 Data latch in falling edge
- 1 Data latch in rising edge

A[1] : LLINE polarity (POR = 0)

The horizontal sync pulse polarity.

- 0 Active low
- 1 Active high

A[0] : LFRAME polarity (POR = 0)

The vertical sync pulse polarity.

- 0 Active low
- 1 Active high

B[6:5] : TFT type(POR = 01)

- 00, 01 TFT mode
- 10 Serial RGB mode
- 11 Serial RGB+dummy mode

HDP[10:8] : High byte of the horizontal panel size (POR = 010)

HDP[7:0] : Low byte of the horizontal panel size (POR = 01111111)

VDP[10:8] : High byte of the vertical panel size (POR = 001)

VDP[7:0] : Low byte of the vertical panel size (POR = 11011111)

G[5:3] : Even line RGB sequence (POR = 000)

- 000 RGB
- 001 RBG
- 010 GRB
- 011 GBR

100 BRG  
 101 BGR  
 11x Reserved

G[2:0] : Odd line RGB sequence (POR = 000)

000 RGB  
 001 RBG  
 010 GRB  
 011 GBR  
 100 BRG  
 101 BGR  
 11x Reserved

### 9.35 set\_hori\_period

**Command** 0xB4  
**Parameters** 8

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	0	1	0	0	B4
Parameter 1	1	0	0	0	0	0	HT <sub>10</sub>	HT <sub>9</sub>	HT <sub>8</sub>	xx
Parameter 2	1	HT <sub>7</sub>	HT <sub>6</sub>	HT <sub>5</sub>	HT <sub>4</sub>	HT <sub>3</sub>	HT <sub>2</sub>	HT <sub>1</sub>	HT <sub>0</sub>	xx
Parameter 3	1	0	0	0	0	0	HPS <sub>10</sub>	HPS <sub>9</sub>	HPS <sub>8</sub>	xx
Parameter 4	1	HPS <sub>7</sub>	HPS <sub>6</sub>	HPS <sub>5</sub>	HPS <sub>4</sub>	HPS <sub>3</sub>	HPS <sub>2</sub>	HPS <sub>1</sub>	HPS <sub>0</sub>	xx
Parameter 5	1	0	HPW <sub>6</sub>	HPW <sub>5</sub>	HPW <sub>4</sub>	HPW <sub>3</sub>	HPW <sub>2</sub>	HPW <sub>1</sub>	HPW <sub>0</sub>	xx
Parameter 6	1	0	0	0	0	0	LPS <sub>10</sub>	LPS <sub>9</sub>	LPS <sub>8</sub>	xx
Parameter 7	1	LPS <sub>7</sub>	LPS <sub>6</sub>	LPS <sub>5</sub>	LPS <sub>4</sub>	LPS <sub>3</sub>	LPS <sub>2</sub>	LPS <sub>1</sub>	LPS <sub>0</sub>	xx
Parameter 8	1	0	0	0	0	0	0	LPSPP <sub>1</sub>	LPSPP <sub>0</sub>	xx

#### Description

Set front porch and back porch

HT[10:8] : High byte of horizontal total period (display + non-display) in pixel clock (POR = 010)

HT[7:0] : Low byte of the horizontal total period (display + non-display) in pixel clock (POR = 10101111)  
 Horizontal total period = (HT + 1) pixels

HPS[10:8] : High byte of the non-display period between the start of the horizontal sync (LLINE) signal and the first display data. (POR = 000)

HPS[7:0] : Low byte of the non-display period between the start of the horizontal sync (LLINE) signal and the first display data. (POR = 00100000)  
 For TFT : Horizontal Sync Pulse Start Position = HPS pixels  
 For Serial TFT : Horizontal Sync Pulse Start Position = HPS pixels + LPSPP subpixels

HPW[6:0] : Set the horizontal sync pulse width (LLINE) in pixel clock. (POR = 0000111)  
 Horizontal Sync Pulse Width = (HPW + 1) pixels

LPS[10:8] : Set the horizontal sync pulse (LLINE) start location in pixel clock. (POR = 000)

LPS[7:0] : Set the horizontal sync pulse width (LLINE) in start. (POR = 00000000)  
 Horizontal Display Period Start Position = LPS pixels

LPSPP[1:0] : Set the horizontal sync pulse subpixel start position for serial TFT interface (POR = 00)

### 9.36 get\_hori\_period

**Command** 0xB5  
**Parameters** 8

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	0	1	0	1	B5
Parameter 1	1	0	0	0	0	0	HT <sub>10</sub>	HT <sub>9</sub>	HT <sub>8</sub>	xx
Parameter 2	1	HT <sub>7</sub>	HT <sub>6</sub>	HT <sub>5</sub>	HT <sub>4</sub>	HT <sub>3</sub>	HT <sub>2</sub>	HT <sub>1</sub>	HT <sub>0</sub>	xx
Parameter 3	1	0	0	0	0	0	HPS <sub>10</sub>	HPS <sub>9</sub>	HPS <sub>8</sub>	xx
Parameter 4	1	HPS <sub>7</sub>	HPS <sub>6</sub>	HPS <sub>5</sub>	HPS <sub>4</sub>	HPS <sub>3</sub>	HPS <sub>2</sub>	HPS <sub>1</sub>	HPS <sub>0</sub>	xx
Parameter 5	1	0	HPW <sub>6</sub>	HPW <sub>5</sub>	HPW <sub>4</sub>	HPW <sub>3</sub>	HPW <sub>2</sub>	HPW <sub>1</sub>	HPW <sub>0</sub>	xx
Parameter 6	1	0	0	0	0	0	LPS <sub>10</sub>	LPS <sub>9</sub>	LPS <sub>8</sub>	xx
Parameter 7	1	LPS <sub>7</sub>	LPS <sub>6</sub>	LPS <sub>5</sub>	LPS <sub>4</sub>	LPS <sub>3</sub>	LPS <sub>2</sub>	LPS <sub>1</sub>	LPS <sub>0</sub>	xx
Parameter 8	1	0	0	0	0	0	LPSPP <sub>1</sub>	LPSPP <sub>0</sub>		xx

### Description

Get current front porch and back porch settings

HT[10:8] : High byte of the horizontal total period (display + non-display) in pixel clock (POR = 010)

HT[7:0] : Low byte of the horizontal total period (display + non-display) in pixel clock (POR = 10101111)

HPS[10:8] : High byte of the non-display period between the start of the horizontal sync (LLINE) signal and the first display data. (POR = 000)

HPS[7:0] : Low byte of the non-display period between the start of the horizontal sync (LLINE) signal and the first display data. (POR = 00100000)

HPW[6:0] : The horizontal sync pulse width (LLINE) in pixel clock. (POR = 0000111)

LPS[10:8] : High byte of the horizontal sync pulse (LLINE) start location in pixel clock. (POR = 000)

LPS[7:0] : Low byte of the horizontal sync pulse width (LLINE) in start. (POR = 00000000)

LPSPP[1:0] : The horizontal sync pulse subpixel start position (POR = 00)

## 9.37 set\_vert\_period

**Command** 0xB6

**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	0	1	1	0	B6
Parameter 1	1	0	0	0	0	0	VT <sub>10</sub>	VT <sub>9</sub>	VT <sub>8</sub>	xx
Parameter 2	1	VT <sub>7</sub>	VT <sub>6</sub>	VT <sub>5</sub>	VT <sub>4</sub>	VT <sub>3</sub>	VT <sub>2</sub>	VT <sub>1</sub>	VT <sub>0</sub>	xx
Parameter 3	1	0	0	0	0	0	VPS <sub>10</sub>	VPS <sub>9</sub>	VPS <sub>8</sub>	xx
Parameter 4	1	VPS <sub>7</sub>	VPS <sub>6</sub>	VPS <sub>5</sub>	VPS <sub>4</sub>	VPS <sub>3</sub>	VPS <sub>2</sub>	VPS <sub>1</sub>	VPS <sub>0</sub>	xx
Parameter 5	1	0	VPW <sub>6</sub>	VPW <sub>5</sub>	VPW <sub>4</sub>	VPW <sub>3</sub>	VPW <sub>2</sub>	VPW <sub>1</sub>	VPW <sub>0</sub>	xx
Parameter 6	1	0	0	0	0	0	FPS <sub>10</sub>	FPS <sub>9</sub>	FPS <sub>8</sub>	xx
Parameter 7	1	FPS <sub>7</sub>	FPS <sub>6</sub>	FPS <sub>5</sub>	FPS <sub>4</sub>	FPS <sub>3</sub>	FPS <sub>2</sub>	FPS <sub>1</sub>	FPS <sub>0</sub>	xx

### Description

Set the vertical blanking interval between last scan line and next LFRAME pulse

VT[10:8] : High byte of the vertical total (display + non-display) period in lines (POR = 001)

VT[7:0] : Low byte of the vertical total (display + non-display) period in lines (POR = 11101111)  
Vertical Total = (VT + 1) lines

- VPS[10:8] : High byte the non-display period in lines between the start of the frame and the first display data in line. (POR = 000)
- VPS[7:0] : The non-display period in lines between the start of the frame and the first display data in line. (POR = 00000100)  
Vertical Sync Pulse Start Position = VPS lines
- VPW[6:0] : Set the vertical sync pulse width (LFRAME) in lines. (POR = 000001)  
Vertical Sync Pulse Width = (VPW + 1) lines
- FPS[10:8] : High byte of the vertical sync pulse (LFRAME) start location in lines. (POR = 000)
- FPS[7:0] : Low byte of the vertical sync pulse (LFRAME) start location in lines. (POR = 00000000)  
Vertical Display Period Start Position = FPS lines

### 9.38 get\_vert\_period

**Command**        0xB7  
**Parameters**     7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	0	1	1	1	B7
Parameter 1	1	0	0	0	0	0	VT <sub>10</sub>	VT <sub>9</sub>	VT <sub>8</sub>	xx
Parameter 2	1	VT <sub>7</sub>	VT <sub>6</sub>	VT <sub>5</sub>	VT <sub>4</sub>	VT <sub>3</sub>	VT <sub>2</sub>	VT <sub>1</sub>	VT <sub>0</sub>	xx
Parameter 3	1	0	0	0	0	0	VPS <sub>10</sub>	VPS <sub>9</sub>	VPS <sub>8</sub>	xx
Parameter 4	1	VPS <sub>7</sub>	VPS <sub>6</sub>	VPS <sub>5</sub>	VPS <sub>4</sub>	VPS <sub>3</sub>	VPS <sub>2</sub>	VPS <sub>1</sub>	VPS <sub>0</sub>	xx
Parameter 5	1	0	VPW <sub>6</sub>	VPW <sub>5</sub>	VPW <sub>4</sub>	VPW <sub>3</sub>	VPW <sub>2</sub>	VPW <sub>1</sub>	VPW <sub>0</sub>	xx
Parameter 6	1	0	0	0	0	0	FPS <sub>10</sub>	FPS <sub>9</sub>	FPS <sub>8</sub>	xx
Parameter 7	1	FPS <sub>7</sub>	FPS <sub>6</sub>	FPS <sub>5</sub>	FPS <sub>4</sub>	FPS <sub>3</sub>	FPS <sub>2</sub>	FPS <sub>1</sub>	FPS <sub>0</sub>	xx

#### Description

Get the vertical blanking interval between last scan line and next LFRAME pulse

- VT[10:8] : High byte of the vertical total (display + non-display) period in lines (POR = 001)
- VT[7:0] : Low byte of the vertical total (display + non-display) period in lines (POR = 01111111)
- VPS[10:8] : High byte of the non-display period in lines between the start of the frame and the first display data in line. (POR = 000)
- VPS[7:0] : Low byte of the non-display period in lines between the start of the frame and the first display data in line. (POR = 00000100)
- VPW[6:0] : The vertical sync pulse width (LFRAME) in lines. (POR = 000001)
- FPS[10:8] : High byte of the vertical sync pulse (LFRAME) start location in lines. (POR = 000)
- FPS[7:0] : Low byte of the vertical sync pulse (LFRAME) start location in lines. (POR = 00000000)

### 9.39 set\_gpio\_conf

**Command** 0xB8  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	1	0	0	0	B8
Parameter 1	1	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	0	0	0	0	0	0	0	B <sub>0</sub>	xx

#### Description

Set the GPIOs configuration. If the GPIOs are not used for LCD, set the direction. Otherwise, they are toggled with LCD signals by 0xC0 – 0xCF.

A[7] : GPIO3 configuration (POR = 0)

- 0 GPIO3 is controlled by host
- 1 GPIO3 is controlled by LCDC

A[6] : GPIO2 configuration (POR = 0)

- 0 GPIO2 is controlled by host
- 1 GPIO2 is controlled by LCDC

A[5] : GPIO1 configuration (POR = 0)

- 0 GPIO1 is controlled by host
- 1 GPIO1 is controlled by LCDC

A[4] : GPIO0 configuration (POR = 0)

- 0 GPIO0 is controlled by host
- 1 GPIO0 is controlled by LCDC

A[3] : GPIO3 direction (POR = 0)

- 0 GPIO3 is input
- 1 GPIO3 is output

A[2] : GPIO2 direction (POR = 0)

- 0 GPIO2 is input
- 1 GPIO2 is output

A[1] : GPIO1 direction (POR = 0)

- 0 GPIO1 is input
- 1 GPIO1 is output

A[0] : GPIO0 direction (POR = 0)

- 0 GPIO0 is input
- 1 GPIO0 is output

B[0] : GPIO0 direction (POR = 0)

- 0 GPIO0 is used to control the panel power with Enter Sleep Mode 0x10 or Exit Sleep Mode 0x11.
- 1 GPIO0 is used as normal GPIO

## 9.40 get\_gpio\_conf

**Command** 0xB9

**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	1	0	0	1	B9
Parameter 1	1	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	0	0	0	0	0	0	0	B <sub>0</sub>	xx

### Description

Get the current GPIOs configuration

A[7] : GPIO3 configuration (POR = 0)

0 GPIO3 is controlled by host

1 GPIO3 is controlled by LCDC

A[6] : GPIO2 configuration (POR = 0)

0 GPIO2 is controlled by host

1 GPIO2 is controlled by LCDC

A[5] : GPIO1 configuration (POR = 0)

0 GPIO1 is controlled by host

1 GPIO1 is controlled by LCDC

A[4] : GPIO0 configuration (POR = 0)

0 GPIO0 is controlled by host

1 GPIO0 is controlled by LCDC

A[3] : GPIO3 direction (POR = 0)

0 GPIO3 is input

1 GPIO3 is output

A[2] : GPIO2 direction (POR = 0)

0 GPIO2 is input

1 GPIO2 is output

A[1] : GPIO1 direction (POR = 0)

0 GPIO1 is input

1 GPIO1 is output

A[0] : GPIO0 direction (POR = 0)

0 GPIO0 is input

1 GPIO0 is output

B[0] : GPIO0 direction (POR = 0)

0 GPIO0 is used to control the panel power with enter\_sleep\_mode (0x10) or exit\_sleep\_mode (0x11)

1 GPIO0 is used as normal GPIO



## 9.41 set\_gpio\_value

**Command** 0xBA

**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	1	0	1	0	BA
Parameter 1	1	0	0	0	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx

### Description

Set GPIO value for GPIO configured as output

A[3] : GPIO3 value (POR = 0)

0 GPIO3 outputs 0

1 GPIO3 outputs 1

A[2] : GPIO2 value (POR = 0)

0 GPIO2 outputs 0

1 GPIO2 outputs 1

A[1] : GPIO1 value (POR = 0)

0 GPIO1 outputs 0

1 GPIO1 outputs 1

A[0] : GPIO0 value (POR = 0)

0 GPIO0 outputs 0

1 GPIO0 outputs 1

## 9.42 get\_gpio\_status

**Command** 0xBB

**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	1	0	1	1	BB
Parameter 1	1	0	0	0	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx

### Description

Read current GPIO status. If the individual GPIO was configured as input, the value is the status of the corresponding pin. Otherwise, it is the programmed value.

A[3] : GPIO3 value (POR : depends on pad value)

0 GPIO3 is pulled low

1 GPIO3 is pulled high

A[2] : GPIO2 value (POR : depends on pad value)

0 GPIO2 is pulled low

1 GPIO2 is pulled high

A[1] : GPIO1 value (POR : depends on pad value)

0 GPIO1 is pulled low

1 GPIO1 is pulled high

A[0] : GPIO0 value (POR : depends on pad value)

0 GPIO0 is pulled low

1 GPIO0 is pulled high

### 9.43 set\_post\_proc

**Command** 0xBC  
**Parameters** 4

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	1	1	0	0	BC
Parameter 1	1	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx
Parameter 3	1	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	0	0	D <sub>0</sub>	xx

#### Description

Set the image post processor

A[7:0] : Set the contrast value (POR = 01000000)

B[7:0] : Set the brightness value (POR = 10000000)

C[7:0] : Set the saturation value (POR = 01000000)

D[0] : Post Processor Enable (POR = 0)  
0 Disable the postprocessor  
1 Enable the postprocessor

### 9.44 get\_post\_proc

**Command** 0xBD  
**Parameters** 4

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	1	1	0	1	BD
Parameter 1	1	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx
Parameter 3	1	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	0	0	D <sub>0</sub>	xx

#### Description

Get the image post processor

A[7:0] : Get the contrast value (POR = 01000000)

B[7:0] : Get the brightness value (POR = 10000000)

C[7:0] : Get the saturation value (POR = 01000000)

D[0] : Post Processor Enable (POR = 0)  
0 Disable the postprocessor  
1 Enable the postprocessor

## 9.45 set\_pwm\_conf

**Command** 0xBE

**Parameters** 6

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	1	1	1	0	BE
Parameter 1	1	PWMF <sub>7</sub>	PWMF <sub>6</sub>	PWMF <sub>5</sub>	PWMF <sub>4</sub>	PWMF <sub>3</sub>	PWMF <sub>2</sub>	PWMF <sub>1</sub>	PWMF <sub>0</sub>	xx
Parameter 2	1	PWM <sub>7</sub>	PWM <sub>6</sub>	PWM <sub>5</sub>	PWM <sub>4</sub>	PWM <sub>3</sub>	PWM <sub>2</sub>	PWM <sub>1</sub>	PWM <sub>0</sub>	xx
Parameter 3	1	0	0	0	0	C <sub>3</sub>	0	0	C <sub>0</sub>	xx
Parameter 4	1	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	xx
Parameter 5	1	E <sub>7</sub>	E <sub>6</sub>	E <sub>5</sub>	E <sub>4</sub>	E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	xx
Parameter 6	1	0	0	0	0	F <sub>3</sub>	F <sub>2</sub>	F <sub>1</sub>	F <sub>0</sub>	xx

### Description

Set the PWM configuration

PWMF[7:0] : Set the PWM frequency in system clock (POR = 00000000)

$$\text{PWM signal frequency} = \text{PLL clock} / (256 * (\text{PWMF}[7:0] + 1)) / 256$$

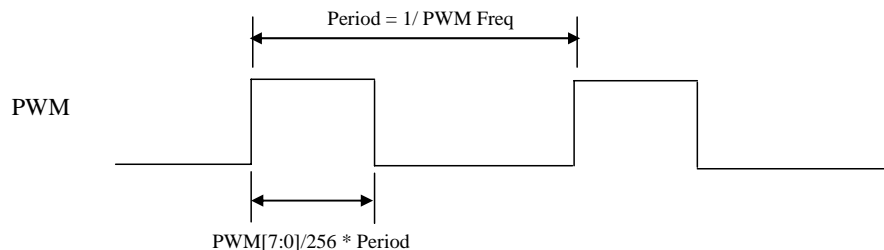
PWM[7:0] : Set the PWM duty cycle (POR = 00000000)

$$\text{PWM duty cycle} = \text{PWM}[7:0] / 256 \text{ for DBC disable (0xD0) A0 = 0}$$

If DBC enable (0xD0) A0 = 1, these parameter will be ignored

Note : PWM always 0 if PWM[7:0] = 00h

**Figure 9-19: PWM signal**



C[3] : PWM configuration (POR = 0)

0 PWM controlled by host

1 PWM controlled by DBC

C[0] : PWM enable (POR = 0)

0 PWM disable

1 PWM enable

D[7:0] : DBC manual brightness (POR = 00000000)

Set the manual brightness level. When Manual Brightness Mode (0xD0) A[6] is enabled, the final DBC duty cycle output will be multiplied by this value / 255.

$$\text{PWM duty cycle} = \text{DBC output} * \text{D}[7:0] / 255$$

00 Dimmest

FF brightest

E[7:0] : DBC minimum brightness (POR = 00000000)

Set the minimum brightness level. When Manual Brightness Mode (0xD0) A[6] is enabled, DBC duty cycle output will be limited by this value. This will prevent from backlight being too dark or off.

00 Dimmest

FF        Brightest

F[3:0] : Brightness prescaler (POR = 0000)

Set the brightness prescaler to control how gradually the manual brightness is changed between different levels. There is a filter will undergo a number of iterations before the manual brightness saturated. This parameter is valid when Transition Effect enable (0xD0) A5 = 1

The iteration ration = system frequency / Divcode / 32768

F[3:0]	Divcode
0000	off
0001	1
0010	2
0011	3
0100	4
0101	6
0110	8
0111	12
1000	16
1001	24
1010	32
1011	48
1100	64
1101	96
1110	128
1111	192

## 9.46 get\_pwm\_conf

**Command**        0xBF

**Parameters**     7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	0	1	1	1	1	1	1	BF
Parameter 1	1	PWMF <sub>7</sub>	PWMF <sub>6</sub>	PWMF <sub>5</sub>	PWMF <sub>4</sub>	PWMF <sub>3</sub>	PWMF <sub>2</sub>	PWMF <sub>1</sub>	PWMF <sub>0</sub>	xx
Parameter 2	1	PWM <sub>7</sub>	PWM <sub>6</sub>	PWM <sub>5</sub>	PWM <sub>4</sub>	PWM <sub>3</sub>	PWM <sub>2</sub>	PWM <sub>1</sub>	PWM <sub>0</sub>	xx
Parameter 3	1	0	0	0	0	C <sub>3</sub>	0	0	C <sub>0</sub>	xx
Parameter 4	1	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	xx
Parameter 5	1	E <sub>7</sub>	E <sub>6</sub>	E <sub>5</sub>	E <sub>4</sub>	E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	xx
Parameter 6	1	0	0	0	0	F <sub>3</sub>	F <sub>2</sub>	F <sub>1</sub>	F <sub>0</sub>	xx
Parameter 7	1	G <sub>7</sub>	G <sub>6</sub>	G <sub>5</sub>	G <sub>4</sub>	G <sub>3</sub>	G <sub>2</sub>	G <sub>1</sub>	G <sub>0</sub>	xx

### Description

Get the PWM configuration

PWMF[7:0] : Get the PWM frequency in system clock (POR = 00000000)

PWM[7:0] : Get the PWM duty cycle (POR = 00000000)

C[3] : PWM configuration (POR = 0)

0        PWM controlled by host  
1        PWM controlled by DBC

C[0] : PWM enable (POR = 0)

0        PWM disable  
1        PWM enable

D[7:0] : DBC manual brightness (POR = 00000000)

Get the brightness level

00        Dimmest  
FF        brightest

E[7:0] : DBC minimum brightness (POR = 00000000)

Get the minimum brightness level.

00 Dimmest

FF Brightest

F[3:0] : Brightness prescaler (POR = 0000)

Get the brightness prescaler

G[7:0] : Dynamic backlight duty cycle : Get the current PWM duty cycle controlled by PWM (POR = 00000000)

## 9.47 set\_lcd\_gen0

**Command** 0xC0  
**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	0	0	0	0	C0
Parameter 1	1	A <sub>7</sub>	0	0	0	0	0	0	0	xx
Parameter 2	1	0	0	0	0	0	GF0 <sub>10</sub>	GF0 <sub>9</sub>	GF0 <sub>8</sub>	xx
Parameter 3	1	GF0 <sub>7</sub>	GF0 <sub>6</sub>	GF0 <sub>5</sub>	GF0 <sub>4</sub>	GF0 <sub>3</sub>	GF0 <sub>2</sub>	GF0 <sub>1</sub>	GF0 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	GR0 <sub>10</sub>	GR0 <sub>9</sub>	GR0 <sub>8</sub>	xx
Parameter 5	1	GR0 <sub>7</sub>	GR0 <sub>6</sub>	GR0 <sub>5</sub>	GR0 <sub>4</sub>	GR0 <sub>3</sub>	GR0 <sub>2</sub>	GR0 <sub>1</sub>	GR0 <sub>0</sub>	xx
Parameter 6	1	F <sub>7</sub>	F <sub>6</sub>	F <sub>5</sub>	F <sub>4</sub>	F <sub>3</sub>	GP0 <sub>10</sub>	GP0 <sub>9</sub>	GP0 <sub>8</sub>	xx
Parameter 7	1	GP0 <sub>7</sub>	GP0 <sub>6</sub>	GP0 <sub>5</sub>	GP0 <sub>4</sub>	GP0 <sub>3</sub>	GP0 <sub>2</sub>	GP0 <sub>1</sub>	GP0 <sub>0</sub>	xx

### Description

Set the rise, fall, period and toggling properties of LCD signal generator 0

A[7] : Reset LCD generator 0 at every frame start

- 0 The generator 0 will not reset in the starting point of a frame
- 1 The generator 0 will reset in the starting point of a frame

GF0[10:8] : The highest 3 bits of the generator 0 falling position (POR = 000)

GF0[7:0] : The lower byte of the generator 0 falling position (POR = 00000001)

GR0[10:8] : The highest 3 bits of the generator 0 rising position (POR = 000)

GR0[7:0] : The lower byte of the generator 0 rising position (POR = 00000000)

F[7] : Force the generator 0 output to 0 in non-display period

- 0 generator 0 is normal
- 1 generator 0 output is forced to 0 in non-display period

F[6:5] : Force the generator 0 output to 0 in odd or even lines

- 00 generator 0 is normal in both odd and even lines
- 01 generator 0 output is force to 0 in odd lines
- 10 generator 0 output is force to 0 in even lines
- 11 generator 0 is normal in both odd and even line

F[4:3] : Generator 0 toggle mode

- 00 Disable
- 01 Toggle by pixel clock (LSHIFT)
- 10 Toggle by Line (LLINE)
- 11 Toggle by Frame (LFRAME)

GP0[10:8] : The highest 3 bits of the generator 0 period (POR = 100)

GP0[7:0] : The lower byte of the generator 0 period (POR = 00000000)

## 9.48 get\_lcd\_gen0

**Command** 0xC1  
**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	0	0	0	1	C1
Parameter 1	1	A <sub>7</sub>	0	0	0	0	0	0	0	xx
Parameter 2	1	0	0	0	0	0	GF0 <sub>10</sub>	GF0 <sub>9</sub>	GF0 <sub>8</sub>	xx
Parameter 3	1	GF0 <sub>7</sub>	GF0 <sub>6</sub>	GF0 <sub>5</sub>	GF0 <sub>4</sub>	GF0 <sub>3</sub>	GF0 <sub>2</sub>	GF0 <sub>1</sub>	GF0 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	GR0 <sub>10</sub>	GR0 <sub>9</sub>	GR0 <sub>8</sub>	xx
Parameter 5	1	GR0 <sub>7</sub>	GR0 <sub>6</sub>	GR0 <sub>5</sub>	GR0 <sub>4</sub>	GR0 <sub>3</sub>	GR0 <sub>2</sub>	GR0 <sub>1</sub>	GR0 <sub>0</sub>	xx
Parameter 6	1	F <sub>7</sub>	F <sub>6</sub>	F <sub>5</sub>	F <sub>4</sub>	F <sub>3</sub>	GP0 <sub>10</sub>	GP0 <sub>9</sub>	GP0 <sub>8</sub>	xx
Parameter 7	1	GP0 <sub>7</sub>	GP0 <sub>6</sub>	GP0 <sub>5</sub>	GP0 <sub>4</sub>	GP0 <sub>3</sub>	GP0 <sub>2</sub>	GP0 <sub>1</sub>	GP0 <sub>0</sub>	xx

### Description

Get the rise, fall, period and toggling properties of LCD signal generator 0

A[7] : Reset LCD generator 0 at every frame start

- 0 The generator 0 will not reset in the starting point of a frame
- 1 The generator 0 will reset in the starting point of a frame

GF0[10:8] : The highest 3 bits of the generator 0 falling position (POR = 000)

GF0[7:0] : The lower byte of the generator 0 falling position (POR = 00000001)

GR0[10:8] : The highest 3 bits of the generator 0 rising position (POR = 000)

GR0[7:0] : The lower byte of the generator 0 rising position (POR = 00000000)

F[7] : Force the generator 0 output to 0 in non-display period

- 0 generator 0 is normal
- 1 generator 0 output is forced to 0 in non-display period

F[6:5] : Force the generator 0 output to 0 in odd or even lines

- 00 generator 0 is normal in both odd and even lines
- 01 generator 0 output is force to 0 in odd lines
- 10 generator 0 output is force to 0 in even lines
- 11 generator 0 is normal in both odd and even line

F[4:3] : Generator 0 toggle mode

- 00 Disable
- 01 Toggle by pixel clock (LSHIFT)
- 10 Toggle by Line (LLINE)
- 11 Toggle by Frame (LFRAME)

GP0[10:8] : The highest 3 bits of the generator 0 period (POR = 100)

GP0[7:0] : The lower byte of the generator 0 period (POR = 00000000)

## 9.49 set\_lcd\_gen1

**Command**      0xC2  
**Parameters**    7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	0	0	1	0	C2
Parameter 1	1	A <sub>7</sub>	0	0	0	0	0	0	0	xx
Parameter 2	1	0	0	0	0	0	GF1 <sub>10</sub>	GF1 <sub>9</sub>	GF1 <sub>8</sub>	xx
Parameter 3	1	GF1 <sub>7</sub>	GF1 <sub>6</sub>	GF1 <sub>5</sub>	GF1 <sub>4</sub>	GF1 <sub>3</sub>	GF1 <sub>2</sub>	GF1 <sub>1</sub>	GF1 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	GR1 <sub>10</sub>	GR1 <sub>9</sub>	GR1 <sub>8</sub>	xx
Parameter 5	1	GR1 <sub>7</sub>	GR1 <sub>6</sub>	GR1 <sub>5</sub>	GR1 <sub>4</sub>	GR1 <sub>3</sub>	GR1 <sub>2</sub>	GR1 <sub>1</sub>	GR1 <sub>0</sub>	xx
Parameter 6	1	F <sub>7</sub>	F <sub>6</sub>	F <sub>5</sub>	F <sub>4</sub>	F <sub>3</sub>	GP1 <sub>10</sub>	GP1 <sub>9</sub>	GP1 <sub>8</sub>	xx
Parameter 7	1	GP1 <sub>7</sub>	GP1 <sub>6</sub>	GP1 <sub>5</sub>	GP1 <sub>4</sub>	GP1 <sub>3</sub>	GP1 <sub>2</sub>	GP1 <sub>1</sub>	GP1 <sub>0</sub>	xx

### Description

Set the rise, fall, period and toggling properties of LCD signal generator 1

A[7] : Reset LCD generator 1 at every frame start

- 0      The generator 1 will not reset in the starting point of a frame
- 1      The generator 1 will reset in the starting point of a frame

GF1[10:8] : The highest 3 bits of the generator 1 falling position (POR = 000)

GF1[7:0] : The lower byte of the generator 1 falling position (POR = 00000001)

GR1[10:8] : The highest 3 bits of the generator 1 rising position (POR = 000)

GR1[7:0] : The lower byte of the generator 1 rising position (POR = 00000000)

F[7] : Force the generator 1 output to 0 in non-display period

- 0      generator 1 is normal
- 1      generator 1 output is forced to 0 in non-display period

F[6:5] : Force the generator 1 output to 0 in odd or even lines

- 00      generator 1 is normal in both odd and even lines
- 01      generator 1 output is force to 0 in odd lines
- 10      generator 1 output is force to 0 in even lines
- 11      generator 1 is normal in both odd and even line

F[4:3] : Generator 1 toggle mode

- 00      Disable
- 01      Toggle by pixel clock (LSHIFT)
- 10      Toggle by Line (LLINE)
- 11      Toggle by Frame (LFRAME)

GP1[10:8] : The highest 3 bits of the generator 1 period (POR = 100)

GP1[7:0] : The lower byte of the generator 1 period (POR = 00000000)



## 9.50 get\_lcd\_gen1

**Command** 0xC3

**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	0	0	1	1	C3
Parameter 1	1	A <sub>7</sub>	0	0	0	0	0	0	0	xx
Parameter 2	1	0	0	0	0	0	GF1 <sub>10</sub>	GF1 <sub>9</sub>	GF1 <sub>8</sub>	xx
Parameter 3	1	GF1 <sub>7</sub>	GF1 <sub>6</sub>	GF1 <sub>5</sub>	GF1 <sub>4</sub>	GF1 <sub>3</sub>	GF1 <sub>2</sub>	GF1 <sub>1</sub>	GF1 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	GR1 <sub>10</sub>	GR1 <sub>9</sub>	GR1 <sub>8</sub>	xx
Parameter 5	1	GR1 <sub>7</sub>	GR1 <sub>6</sub>	GR1 <sub>5</sub>	GR1 <sub>4</sub>	GR1 <sub>3</sub>	GR1 <sub>2</sub>	GR1 <sub>1</sub>	GR1 <sub>0</sub>	xx
Parameter 6	1	F <sub>7</sub>	F <sub>6</sub>	F <sub>5</sub>	F <sub>4</sub>	F <sub>3</sub>	GP1 <sub>10</sub>	GP1 <sub>9</sub>	GP1 <sub>8</sub>	xx
Parameter 7	1	GP1 <sub>7</sub>	GP1 <sub>6</sub>	GP1 <sub>5</sub>	GP1 <sub>4</sub>	GP1 <sub>3</sub>	GP1 <sub>2</sub>	GP1 <sub>1</sub>	GP1 <sub>0</sub>	xx

### Description

Get the rise, fall, period and toggling properties of LCD signal generator 1

A[7] : Reset LCD generator 1 at every frame start

0 The generator 1 will not reset in the starting point of a frame

1 The generator 1 will reset in the starting point of a frame

GF1[10:8] : The highest 3 bits of the generator 1 falling position (POR = 000)

GF1[7:0] : The lower byte of the generator 1 falling position (POR = 00000001)

GR1[10:8] : The highest 3 bits of the generator 1 rising position (POR = 000)

GR1[7:0] : The lower byte of the generator 1 rising position (POR = 00000000)

F[7] : Force the generator 1 output to 0 in non-display period

0 generator 1 is normal

1 generator 1 output is forced to 0 in non-display period

F[6:5] : Force the generator 1 output to 0 in odd or even lines

00 generator 1 is normal in both odd and even lines

01 generator 1 output is force to 0 in odd lines

10 generator 1 output is force to 0 in even lines

11 generator 1 is normal in both odd and even line

F[4:3] : Generator 1 toggle mode

00 Disable

01 Toggle by pixel clock (LSHIFT)

10 Toggle by Line (LLINE)

11 Toggle by Frame (LFRAME)

GP1[10:8] : The highest 3 bits of the generator 1 period (POR = 100)

GP1[7:0] : The lower byte of the generator 1 period (POR = 00000000)

## 9.51 set\_lcd\_gen2

**Command** 0xC4  
**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	0	1	0	0	C4
Parameter 1	1	A <sub>7</sub>	0	0	0	0	0	0	0	xx
Parameter 2	1	0	0	0	0	0	GF2 <sub>10</sub>	GF2 <sub>9</sub>	GF2 <sub>8</sub>	xx
Parameter 3	1	GF2 <sub>7</sub>	GF2 <sub>6</sub>	GF2 <sub>5</sub>	GF2 <sub>4</sub>	GF2 <sub>3</sub>	GF2 <sub>2</sub>	GF2 <sub>1</sub>	GF2 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	GR2 <sub>10</sub>	GR2 <sub>9</sub>	GR2 <sub>8</sub>	xx
Parameter 5	1	GR2 <sub>7</sub>	GR2 <sub>6</sub>	GR2 <sub>5</sub>	GR2 <sub>4</sub>	GR2 <sub>3</sub>	GR2 <sub>2</sub>	GR2 <sub>1</sub>	GR2 <sub>0</sub>	xx
Parameter 6	1	F <sub>7</sub>	F <sub>6</sub>	F <sub>5</sub>	F <sub>4</sub>	F <sub>3</sub>	GP2 <sub>10</sub>	GP2 <sub>9</sub>	GP2 <sub>8</sub>	xx
Parameter 7	1	GP2 <sub>7</sub>	GP2 <sub>6</sub>	GP2 <sub>5</sub>	GP2 <sub>4</sub>	GP2 <sub>3</sub>	GP2 <sub>2</sub>	GP2 <sub>1</sub>	GP2 <sub>0</sub>	xx

### Description

Set the rise, fall, period and toggling properties of LCD signal generator 2

A[7] : Reset LCD generator 2 at every frame start

- 0 The generator 2 will not reset in the starting point of a frame
- 1 The generator 2 will reset in the starting point of a frame

GF2[10:8] : The highest 3 bits of the generator 2 falling position (POR = 000)

GF2[7:0] : The lower byte of the generator 2 falling position (POR = 00000001)

GR2[10:8] : The highest 3 bits of the generator 2 rising position (POR = 000)

GR2[7:0] : The lower byte of the generator 2 rising position (POR = 00000000)

F[7] : Force the generator 2 output to 0 in non-display period

- 0 generator 2 is normal
- 1 generator 2 output is forced to 0 in non-display period

F[6:5] : Force the generator 2 output to 0 in odd or even lines

- 00 generator 2 is normal in both odd and even lines
- 01 generator 2 output is force to 0 in odd lines
- 10 generator 2 output is force to 0 in even lines
- 11 generator 2 is normal in both odd and even line

F[4:3] : Generator 2 toggle mode

- 00 Disable
- 01 Toggle by pixel clock (LSHIFT)
- 10 Toggle by Line (LLINE)
- 11 Toggle by Frame (LFRAME)

GP2[10:8] : The highest 3 bits of the generator 2 period (POR = 100)

GP2[7:0] : The lower byte of the generator 2 period (POR = 00000000)

## 9.52 get\_lcd\_gen2

**Command** 0xC5  
**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	0	1	0	1	C5
Parameter 1	1	A <sub>7</sub>	0	0	0	0	0	0	0	xx
Parameter 2	1	0	0	0	0	0	GF2 <sub>10</sub>	GF2 <sub>9</sub>	GF2 <sub>8</sub>	xx
Parameter 3	1	GF2 <sub>7</sub>	GF2 <sub>6</sub>	GF2 <sub>5</sub>	GF2 <sub>4</sub>	GF2 <sub>3</sub>	GF2 <sub>2</sub>	GF2 <sub>1</sub>	GF2 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	GR2 <sub>10</sub>	GR2 <sub>9</sub>	GR2 <sub>8</sub>	xx
Parameter 5	1	GR2 <sub>7</sub>	GR2 <sub>6</sub>	GR2 <sub>5</sub>	GR2 <sub>4</sub>	GR2 <sub>3</sub>	GR2 <sub>2</sub>	GR2 <sub>1</sub>	GR2 <sub>0</sub>	xx
Parameter 6	1	F <sub>7</sub>	F <sub>6</sub>	F <sub>5</sub>	F <sub>4</sub>	F <sub>3</sub>	GP2 <sub>10</sub>	GP2 <sub>9</sub>	GP2 <sub>8</sub>	xx
Parameter 7	1	GP2 <sub>7</sub>	GP2 <sub>6</sub>	GP2 <sub>5</sub>	GP2 <sub>4</sub>	GP2 <sub>3</sub>	GP2 <sub>2</sub>	GP2 <sub>1</sub>	GP2 <sub>0</sub>	xx

### Description

Get the rise, fall, period and toggling properties of LCD signal generator 2

A[7] : Reset LCD generator 2 at every frame start

- 0 The generator 2 will not reset in the starting point of a frame
- 1 The generator 2 will reset in the starting point of a frame

GF2[10:8] : The highest 3 bits of the generator 2 falling position (POR = 000)

GF2[7:0] : The lower byte of the generator 2 falling position (POR = 00000001)

GR2[10:8] : The highest 3 bits of the generator 2 rising position (POR = 000)

GR2[7:0] : The lower byte of the generator 2 rising position (POR = 00000000)

F[7] : Force the generator 2 output to 0 in non-display period

- 0 generator 2 is normal
- 1 generator 2 output is forced to 0 in non-display period

F[6:5] : Force the generator 2 output to 0 in odd or even lines

- 00 generator 2 is normal in both odd and even lines
- 01 generator 2 output is force to 0 in odd lines
- 10 generator 2 output is force to 0 in even lines
- 11 generator 2 is normal in both odd and even line

F[4:3] : Generator 2 toggle mode

- 00 Disable
- 01 Toggle by pixel clock (LSHIFT)
- 10 Toggle by Line (LLINE)
- 11 Toggle by Frame (LFRAME)

GP2[10:8] : The highest 3 bits of the generator 2 period (POR = 100)

GP2[7:0] : The lower byte of the generator 2 period (POR = 00000000)

## 9.53 set\_lcd\_gen3

**Command** 0xC6  
**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	0	1	1	0	C6
Parameter 1	1	A <sub>7</sub>	0	0	0	0	0	0	0	xx
Parameter 2	1	0	0	0	0	0	GF3 <sub>10</sub>	GF3 <sub>9</sub>	GF3 <sub>8</sub>	xx
Parameter 3	1	GF3 <sub>7</sub>	GF3 <sub>6</sub>	GF3 <sub>5</sub>	GF3 <sub>4</sub>	GF3 <sub>3</sub>	GF3 <sub>2</sub>	GF3 <sub>1</sub>	GF3 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	GR3 <sub>10</sub>	GR3 <sub>9</sub>	GR3 <sub>8</sub>	xx
Parameter 5	1	GR3 <sub>7</sub>	GR3 <sub>6</sub>	GR3 <sub>5</sub>	GR3 <sub>4</sub>	GR3 <sub>3</sub>	GR3 <sub>2</sub>	GR3 <sub>1</sub>	GR3 <sub>0</sub>	xx
Parameter 6	1	F <sub>7</sub>	F <sub>6</sub>	F <sub>5</sub>	F <sub>4</sub>	F <sub>3</sub>	GP3 <sub>10</sub>	GP3 <sub>9</sub>	GP3 <sub>8</sub>	xx
Parameter 7	1	GP3 <sub>7</sub>	GP3 <sub>6</sub>	GP3 <sub>5</sub>	GP3 <sub>4</sub>	GP3 <sub>3</sub>	GP3 <sub>2</sub>	GP3 <sub>1</sub>	GP3 <sub>0</sub>	xx

### Description

Set the rise, fall, period and toggling properties of LCD signal generator 3

A[7] : Reset LCD generator 3 at every frame start

- 0 The generator 3 will not reset in the starting point of a frame
- 1 The generator 3 will reset in the starting point of a frame

GF3[10:8] : The highest 3 bits of the generator 3 falling position (POR = 000)

GF3[7:0] : The lower byte of the generator 3 falling position (POR = 00000001)

GR3[10:8] : The highest 3 bits of the generator 3 rising position (POR = 000)

GR3[7:0] : The lower byte of the generator 3 rising position (POR = 00000000)

F[7] : Force the generator 3 output to 0 in non-display period

- 0 generator 3 is normal
- 1 generator 3 output is forced to 0 in non-display period

F[6:5] : Force the generator 3 output to 0 in odd or even lines

- 00 generator 3 is normal in both odd and even lines
- 01 generator 3 output is force to 0 in odd lines
- 10 generator 3 output is force to 0 in even lines
- 11 generator 3 is normal in both odd and even line

F[4:3] : Generator 3 toggle mode

- 00 Disable
- 01 Toggle by pixel clock (LSHIFT)
- 10 Toggle by Line (LLINE)
- 11 Toggle by Frame (LFRAME)

GP3[10:8] : The highest 3 bits of the generator 3 period (POR = 100)

GP3[7:0] : The lower byte of the generator 3 period (POR = 00000000)

## 9.54 get\_lcd\_gen3

**Command** 0xC7  
**Parameters** 7

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	0	1	1	1	C7
Parameter 1	1	A <sub>7</sub>	0	0	0	0	0	0	0	xx
Parameter 2	1	0	0	0	0	0	GF3 <sub>10</sub>	GF3 <sub>9</sub>	GF3 <sub>8</sub>	xx
Parameter 3	1	GF3 <sub>7</sub>	GF3 <sub>6</sub>	GF3 <sub>5</sub>	GF3 <sub>4</sub>	GF3 <sub>3</sub>	GF3 <sub>2</sub>	GF3 <sub>1</sub>	GF3 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	GR3 <sub>10</sub>	GR3 <sub>9</sub>	GR3 <sub>8</sub>	xx
Parameter 5	1	GR3 <sub>7</sub>	GR3 <sub>6</sub>	GR3 <sub>5</sub>	GR3 <sub>4</sub>	GR3 <sub>3</sub>	GR3 <sub>2</sub>	GR3 <sub>1</sub>	GR3 <sub>0</sub>	xx
Parameter 6	1	F <sub>7</sub>	F <sub>6</sub>	F <sub>5</sub>	F <sub>4</sub>	F <sub>3</sub>	GP3 <sub>10</sub>	GP3 <sub>9</sub>	GP3 <sub>8</sub>	xx
Parameter 7	1	GP3 <sub>7</sub>	GP3 <sub>6</sub>	GP3 <sub>5</sub>	GP3 <sub>4</sub>	GP3 <sub>3</sub>	GP3 <sub>2</sub>	GP3 <sub>1</sub>	GP3 <sub>0</sub>	xx

### Description

Get the rise, fall, period and toggling properties of LCD signal generator 3

A[7] : Reset LCD generator 3 at every frame start

- 0 The generator 3 will not reset in the starting point of a frame
- 1 The generator 3 will reset in the starting point of a frame

GF3[10:8] : The highest 3 bits of the generator 3 falling position (POR = 000)

GF3[7:0] : The lower byte of the generator 3 falling position (POR = 00000001)

GR3[10:8] : The highest 3 bits of the generator 3 rising position (POR = 000)

GR3[7:0] : The lower byte of the generator 3 rising position (POR = 00000000)

F[7] : Force the generator 3 output to 0 in non-display period

- 0 generator 3 is normal
- 1 generator 3 output is forced to 0 in non-display period

F[6:5] : Force the generator 3 output to 0 in odd or even lines

- 00 generator 3 is normal in both odd and even lines
- 01 generator 3 output is force to 0 in odd lines
- 10 generator 3 output is force to 0 in even lines
- 11 generator 3 is normal in both odd and even line

F[4:3] : Generator 3 toggle mode

- 00 Disable
- 01 Toggle by pixel clock (LSHIFT)
- 10 Toggle by Line (LLINE)
- 11 Toggle by Frame (LFRAME)

GP3[10:8] : The highest 3 bits of the generator 3 period (POR = 100)

GP3[7:0] : The lower byte of the generator 3 period (POR = 00000000)

## 9.55 set\_gpio0\_rop

**Command** 0xC8

**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	1	0	0	0	C8
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx

### Description

Set the GPIO0 with respect to the LCD signal generators using ROP operation. No effect if the GPIO0 is configured as general GPIO.

A[6:5] : Source 1 for GPIO0 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

A[3:2] : Source 2 for GPIO0 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

A[1:0] : Source 3 for GPIO0 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

B[7:0] : ROP operation to mux the source 1, 2 and 3 for GPIO0 (POR = 00000000)

Please refer to the Application note for the ROP operation

## 9.56 get\_gpio0\_rop

**Command** 0xC9

**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	1	0	0	1	C9
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx

### Description

Get the GPIO0 properties with respect to the LCD signal generators.

A[6:5] : Source 1 for GPIO0 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

A[3:2] : Source 2 for GPIO0 when controlled by LCD (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

A[1:0] : Source 3 for GPIO0 when controlled by LCD (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

B[7:0] : ROP operation to mux the source 1, 2 and 3 for GPIO0 (POR = 00000000)  
 Please refer to the Application note for the ROP operation

### 9.57 set\_gpio1\_rop

**Command** 0xCA  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	1	0	1	0	CA
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx

#### Description

Set the GPIO1 with respect to the LCD signal generators using ROP operation. No effect if the GPIO1 is configured as general GPIO.

A[6:5] : Source 1 for GPIO1 when controlled by LCD (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

A[3:2] : Source 2 for GPIO1 when controlled by LCD (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

A[1:0] : Source 3 for GPIO1 when controlled by LCD (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

B[7:0] : ROP operation to mux the source 1, 2 and 3 for GPIO1 (POR = 00000000)  
 Please refer to the Application note for the ROP operation

## 9.58 get\_gpio1\_rop

**Command** 0xCB  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	1	0	1	1	CB
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx

### Description

Get the GPIO1 properties with respect to the LCD signal generators.

A[6:5] : Source 1 for GPIO1 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

A[3:2] : Source 2 for GPIO1 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

A[1:0] : Source 3 for GPIO1 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

B[7:0] : ROP operation to mux the source 1, 2 and 3 for GPIO1 (POR = 00000000)

Please refer to the Application note for the ROP operation

## 9.59 set\_gpio2\_rop

**Command** 0xCC  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	1	1	0	0	CC
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx

### Description

Set the GPIO2 with respect to the LCD signal generators using ROP operation. No effect if the GPIO2 is configured as general GPIO.

A[6:5] : Source 1 for GPIO2 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3



A[3:2] : Source 2 for GPIO2 when controlled by LCDC (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

A[1:0] : Source 3 for GPIO2 when controlled by LCDC (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

B[7:0] : ROP operation to mux the source 1, 2 and 3 for GPIO2 (POR = 00000000)  
 Please refer to the Application note for the ROP operation

## 9.60 get\_gpio2\_rop

**Command** 0xCD  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	1	1	0	1	CD
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx

### Description

Get the GPIO2 properties with respect to the LCD signal generators.

A[6:5] : Source 1 for GPIO2 when controlled by LCDC (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

A[3:2] : Source 2 for GPIO2 when controlled by LCDC (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

A[1:0] : Source 3 for GPIO2 when controlled by LCDC (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

B[7:0] : ROP operation to mux the source 1, 2 and 3 for GPIO2 (POR = 00000000)  
 Please refer to the Application note for the ROP operation

## 9.61 set\_gpio3\_rop

**Command** 0xCE  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	1	1	1	0	CE
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx

### Description

Set the GPIO3 with respect to the LCD signal generators using ROP operation. No effect if the GPIO3 is configured as general GPIO.

A[6:5] : Source 1 for GPIO3 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

A[3:2] : Source 2 for GPIO3 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

A[1:0] : Source 3 for GPIO3 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

B[7:0] : ROP operation to mux the source 1, 2 and 3 for GPIO3 (POR = 00000000)  
Please refer to the Application note for the ROP operation

## 9.62 get\_gpio3\_rop

**Command** 0xCF  
**Parameters** 2

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	0	1	1	1	1	CF
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx
Parameter 2	1	B <sub>7</sub>	B <sub>6</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	xx

### Description

Get the GPIO3 properties with respect to the LCD signal generators.

A[6:5] : Source 1 for GPIO3 when controlled by LCDC (POR = 00)

00 Generator 0  
01 Generator 1  
10 Generator 2  
11 Generator 3

A[3:2] : Source 2 for GPIO3 when controlled by LCDC (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

A[1:0] : Source 3 for GPIO3 when controlled by LCDC (POR = 00)  
 00 Generator 0  
 01 Generator 1  
 10 Generator 2  
 11 Generator 3

B[7:0] : ROP operation to mux the source 1, 2 and 3 for GPIO3 (POR = 00000000)  
 Please refer to the Application note for the ROP operation

### 9.63 set\_dbc\_conf

**Command** 0xD0  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	1	0	0	0	0	D0
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	0	A <sub>0</sub>	xx

#### Description

Set the Dynamic Backlight Control configuration.

A[6] : DBC Manual Brightness enable (POR = 1)  
 0 Enable  
 1 Disable

A[5] : Transition effect (POR = 0)  
 0 Transition effect disable  
 1 Transition effect enable

Transition effect is used to remove visible backlight flickering. If rapid brightness change is required, it is recommended to enable this bit.

A[3:2] : Energy saving selection for DBC (POR = 00)  
 00 DBC is disable  
 01 Conservative mode  
 10 Normal mode  
 11 Aggressive mode

A[0] : Master enable of DBC (POR = 0)  
 0 DBC disable  
 1 DBC enable

The hardware pin, PWM is the output signal from SSD1961 to the system backlight driver. So it should configure PWM module before enable DBC.

```

WRITE COMMAND "0xBE"
WRITE DATA "0x0E" (set PWM frequency)
WRITE DATA "0xFF" (dummy value if DBC is used)
WRITE DATA "0x09" (enable PWM controlled by DBC)
WRITE DATA "0xFF"
WRITE DATA "0x00"
WRITE DATA "0x00"

WRITE COMMAND "0xD4" (Define the threshold value)
WRITE DATA .....

WRITE COMMAND "0xD0"
WRITE DATA "0x0D" (Enable DBC with Aggressive mode)

```

### 9.64 get\_dbc\_conf

**Command** 0xD1  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	1	0	0	0	1	D1
Parameter 1	1	0	A <sub>6</sub>	A <sub>5</sub>	0	A <sub>3</sub>	A <sub>2</sub>	1	A <sub>0</sub>	xx

#### Description

Get the current dynamic back light configuration.

A[6] : DBC Manual Brightness enable (POR = 1)

- 0 Enable
- 1 Disable

A[5] : Transition effect (POR = 0)

- 0 Transition effect disable
- 1 Transition effect enable

A[3:2] : Energy saving selection for DBC (POR = 00)

- 00 DBC is disable
- 01 Conservative mode
- 10 Normal mode
- 11 Aggressive mode

A[0] : Master enable DBC (POR = 0)

- 0 DBC disable
- 1 DBC enable

## 9.65 set\_dbc\_th

**Command** 0xD4  
**Parameters** 9

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	1	0	1	0	0	D4
Parameter 1	1	0	0	0	0	0	0	0	DBC_TH1 <sub>16</sub>	xx
Parameter 2	1	DBC_TH1 <sub>15</sub>	DBC_TH1 <sub>14</sub>	DBC_TH1 <sub>13</sub>	DBC_TH1 <sub>12</sub>	DBC_TH1 <sub>11</sub>	DBC_TH1 <sub>10</sub>	DBC_TH1 <sub>9</sub>	DBC_TH1 <sub>8</sub>	xx
Parameter 3	1	DBC_TH1 <sub>7</sub>	DBC_TH1 <sub>6</sub>	DBC_TH1 <sub>5</sub>	DBC_TH1 <sub>4</sub>	DBC_TH1 <sub>3</sub>	DBC_TH1 <sub>2</sub>	DBC_TH1 <sub>1</sub>	DBC_TH1 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	0	0	DBC_TH2 <sub>16</sub>	xx
Parameter 5	1	DBC_TH2 <sub>15</sub>	DBC_TH2 <sub>14</sub>	DBC_TH2 <sub>13</sub>	DBC_TH2 <sub>12</sub>	DBC_TH2 <sub>11</sub>	DBC_TH2 <sub>10</sub>	DBC_TH2 <sub>9</sub>	DBC_TH2 <sub>8</sub>	xx
Parameter 6	1	DBC_TH2 <sub>7</sub>	DBC_TH2 <sub>6</sub>	DBC_TH2 <sub>5</sub>	DBC_TH2 <sub>4</sub>	DBC_TH2 <sub>3</sub>	DBC_TH2 <sub>2</sub>	DBC_TH2 <sub>1</sub>	DBC_TH2 <sub>0</sub>	xx
Parameter 7	1	0	0	0	0	0	0	0	DBC_TH3 <sub>16</sub>	xx
Parameter 8	1	DBC_TH3 <sub>15</sub>	DBC_TH3 <sub>14</sub>	DBC_TH3 <sub>13</sub>	DBC_TH3 <sub>12</sub>	DBC_TH3 <sub>11</sub>	DBC_TH3 <sub>10</sub>	DBC_TH3 <sub>9</sub>	DBC_TH3 <sub>8</sub>	xx
Parameter 9	1	DBC_TH3 <sub>7</sub>	DBC_TH3 <sub>6</sub>	DBC_TH3 <sub>5</sub>	DBC_TH3 <sub>4</sub>	DBC_TH3 <sub>3</sub>	DBC_TH3 <sub>2</sub>	DBC_TH3 <sub>1</sub>	DBC_TH3 <sub>0</sub>	xx

### Description

Set the threshold for each level of power saving.

DBC\_TH1[16] : High byte of the threshold setting for the Conservative mode of DBC. (POR = 0)  
 DBC\_TH1[15:8] : 2nd byte of the threshold setting for the Conservative mode of DBC. (POR = 00000000)  
 DBC\_TH1[7:0] : Low byte of the threshold setting for the Conservative mode of DBC. (POR = 00000000)

TH1 = display width \* display height \* 3 \* 0.1 / 16

DBC\_TH2[16] : High byte of the threshold setting for the Normal mode of DBC. (POR = 0)  
 DBC\_TH2[15:8] : 2nd byte of the threshold setting for the Normal mode of DBC. (POR = 00000000)  
 DBC\_TH2[7:0] : Low byte of the threshold setting for the Normal mode of DBC. (POR = 00000000)

TH2 = display width \* display height \* 3 \* 0.25 / 16

DBC\_TH3[16] : High byte of the threshold setting for the Aggressive mode of DBC. (POR = 0)  
 DBC\_TH3[15:8] : 2nd byte of the threshold setting for the Aggressive mode of DBC. (POR = 00000000)  
 DBC\_TH3[7:0] : Low byte of the threshold setting for the Aggressive mode of DBC. (POR = 00000000)

TH3 = display width \* display height \* 3 \* 0.6 / 16

## 9.66 get\_dbc\_th

**Command** 0xD5  
**Parameters** 9

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	0	1	0	1	0	1	D5
Parameter 1	1	0	0	0	0	0	0	0	DBC_TH1 <sub>16</sub>	xx
Parameter 2	1	DBC_TH1 <sub>15</sub>	DBC_TH1 <sub>14</sub>	DBC_TH1 <sub>13</sub>	DBC_TH1 <sub>12</sub>	DBC_TH1 <sub>11</sub>	DBC_TH1 <sub>10</sub>	DBC_TH1 <sub>9</sub>	DBC_TH1 <sub>8</sub>	xx
Parameter 3	1	DBC_TH1 <sub>7</sub>	DBC_TH1 <sub>6</sub>	DBC_TH1 <sub>5</sub>	DBC_TH1 <sub>4</sub>	DBC_TH1 <sub>3</sub>	DBC_TH1 <sub>2</sub>	DBC_TH1 <sub>1</sub>	DBC_TH1 <sub>0</sub>	xx
Parameter 4	1	0	0	0	0	0	0	0	DBC_TH2 <sub>16</sub>	xx
Parameter 5	1	DBC_TH2 <sub>15</sub>	DBC_TH2 <sub>14</sub>	DBC_TH2 <sub>13</sub>	DBC_TH2 <sub>12</sub>	DBC_TH2 <sub>11</sub>	DBC_TH2 <sub>10</sub>	DBC_TH2 <sub>9</sub>	DBC_TH2 <sub>8</sub>	xx
Parameter 6	1	DBC_TH2 <sub>7</sub>	DBC_TH2 <sub>6</sub>	DBC_TH2 <sub>5</sub>	DBC_TH2 <sub>4</sub>	DBC_TH2 <sub>3</sub>	DBC_TH2 <sub>2</sub>	DBC_TH2 <sub>1</sub>	DBC_TH2 <sub>0</sub>	xx
Parameter 7	1	0	0	0	0	0	0	0	DBC_TH3 <sub>16</sub>	xx
Parameter 8	1	DBC_TH3 <sub>15</sub>	DBC_TH3 <sub>14</sub>	DBC_TH3 <sub>13</sub>	DBC_TH3 <sub>12</sub>	DBC_TH3 <sub>11</sub>	DBC_TH3 <sub>10</sub>	DBC_TH3 <sub>9</sub>	DBC_TH3 <sub>8</sub>	xx
Parameter 9	1	DBC_TH3 <sub>7</sub>	DBC_TH3 <sub>6</sub>	DBC_TH3 <sub>5</sub>	DBC_TH3 <sub>4</sub>	DBC_TH3 <sub>3</sub>	DBC_TH3 <sub>2</sub>	DBC_TH3 <sub>1</sub>	DBC_TH3 <sub>0</sub>	xx

### Description

Get the threshold for each level of power saving.

DBC\_TH1[16] : High byte of the threshold setting for the Conservative mode of DBC. (POR = 0)  
 DBC\_TH1[15:8] : 2nd byte of the threshold setting for the Conservative mode of DBC. (POR = 00000000)  
 DBC\_TH1[7:0] : Low byte of the threshold setting for the Conservative mode of DBC. (POR = 00000000)

DBC\_TH2[16] : High byte of the threshold setting for the Normal mode of DBC. (POR = 0)  
 DBC\_TH2[15:8] : 2nd byte of the threshold setting for the Normal mode of DBC. (POR = 00000000)  
 DBC\_TH2[7:0] : Low byte of the threshold setting for the Normal mode of DBC. (POR = 00000000)

DBC\_TH3[16] : High byte of the threshold setting for the Aggressive mode of DBC. (POR = 0)  
 DBC\_TH3[15:8] : 2nd byte of the threshold setting for the Aggressive mode of DBC. (POR = 00000000)  
 DBC\_TH3[7:0] : Low byte of the threshold setting for the Aggressive mode of DBC. (POR = 00000000)

## 9.67 set\_pll

**Command** 0xE0  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	1	0	0	0	0	0	E0
Parameter 1	1	0	0	0	0	0	0	A <sub>1</sub>	A <sub>0</sub>	xx

### Description

Start the PLL. Before the start, the system was operated with the crystal oscillator or clock input.

A[1] : Lock PLL (POR = 0)  
 After PLL enabled for 100us, can start to lock PLL  
 0 Use reference clock as system clock  
 1 Use PLL output as system clock

A[0] : Enable PLL (POR = 0)  
 0 Disable PLL  
 1 Enable PLL

Before enabling PLL, the PLL setting (“0xE2”) have to be configured first. After PLL enabled for 100us, can start to lock PLL. SSD1961 needed to switch to PLL output as system clock after PLL is locked. The following is the program sequence.

```
WRITE COMMAND “0xE0”
WRITE DATA “0x01”
Wait 100us to let the PLL stable
WRITE COMMAND “0xE0”
WRITE DATA “0x03”
WRITE COMMAND “0x01”
```

\* Note : SSD1961 is operating under reference clock before PLL is locked, registers cannot be set faster than half of the reference clock frequency. For instance, SSD1961 with a 10MHz reference clock is not allowed to be programmed higher than 5M words/s.

### 9.68 set\_pll\_mn

**Command**        0xE2  
**Parameters**     3

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	1	0	0	0	1	0	E2
Parameter 1	1	M <sub>7</sub>	M <sub>6</sub>	M <sub>5</sub>	M <sub>4</sub>	M <sub>3</sub>	M <sub>2</sub>	M <sub>1</sub>	M <sub>0</sub>	xx
Parameter 2	1	0	0	1	N <sub>4</sub>	N <sub>3</sub>	N <sub>2</sub>	N <sub>1</sub>	N <sub>0</sub>	xx
Parameter 3	1	0	0	0	0	0	C <sub>2</sub>	0	0	xx

#### Description

Set the MN of PLL

M[7:0] : Multiplier (M) of PLL. (POR = 00101101)

N[4:0] : Divider (N) of PLL. (POR = 00011)

C[2] : Effectuate MN value (POR = 0)

- 0        Ignore the multiplier (M) and divider (N) values
- 1        Effectuate the multiplier and divider value

VCO = Reference input clock x (M + 1)

PLL frequency = VCO / (N + 1)

**\* Note : 250MHz < VCO < 800MHz**

For a 10MHz reference clock to obtain 100MHz PLL frequency, user cannot program M = 19 and N = 1. The setting in this situation is setting M=29 and N=2, where  $10 \times 30 / 3 = 100\text{MHz}$ .

```
WRITE COMMAND “0xE2”
WRITE DATA “0x1D” (M=29)
WRITE DATA “0x02” (N=2)
WRITE DATA “0x54” (Dummy Byte)
```

## 9.69 get\_pll\_mn

**Command** 0xE3  
**Parameters** 3

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	1	0	0	0	1	1	E3
Parameter 1	1	M <sub>7</sub>	M <sub>6</sub>	M <sub>5</sub>	M <sub>4</sub>	M <sub>3</sub>	M <sub>2</sub>	M <sub>1</sub>	M <sub>0</sub>	xx
Parameter 2	1	0	0	1	N <sub>4</sub>	N <sub>3</sub>	N <sub>2</sub>	N <sub>1</sub>	N <sub>0</sub>	xx
Parameter 3	1	0	0	0	0	0	C <sub>2</sub>	0	0	xx

### Description

Get the MN setting of PLL

M[7:0] : Multiplier (M) of PLL. (POR = 00101101)

N[4:0] : Divider (N) of PLL. (POR = 00011)

C[2] : Effectuate MN value (POR = 0)

- 0 Ignore the multiplier (M) and divider (N) values
- 1 Effectuate the multiplier and divider value

## 9.70 get\_pll\_status

**Command** 0xE4  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	1	0	0	1	0	0	E4
Parameter 1	1	0	0	0	0	0	A <sub>2</sub>	0	0	xx

### Description

Get the PLL status

A[2] : PLL Lock

- 0 Not locked
- 1 Locked

## 9.71 set\_deep\_sleep

**Command** 0xE5  
**Parameters** None

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	1	0	0	1	0	1	E5

### Description

Set deep sleep mode. PLL would be stopped.

It needs to issue 2 dummy read to exit Deep Sleep mode.



## 9.72 set\_lshift\_freq

**Command** 0xE6  
**Parameters** 3

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	1	0	0	1	1	0	E6
Parameter 1	1	0	0	0	0	LCDC_FPR <sub>19</sub>	LCDC_FPR <sub>18</sub>	LCDC_FPR <sub>17</sub>	LCDC_FPR <sub>16</sub>	xx
Parameter 2	1	LCDC_FPR <sub>15</sub>	LCDC_FPR <sub>14</sub>	LCDC_FPR <sub>13</sub>	LCDC_FPR <sub>12</sub>	LCDC_FPR <sub>11</sub>	LCDC_FPR <sub>10</sub>	LCDC_FPR <sub>9</sub>	LCDC_FPR <sub>8</sub>	xx
Parameter 3	1	LCDC_FPR <sub>7</sub>	LCDC_FPR <sub>6</sub>	LCDC_FPR <sub>5</sub>	LCDC_FPR <sub>4</sub>	LCDC_FPR <sub>3</sub>	LCDC_FPR <sub>2</sub>	LCDC_FPR <sub>1</sub>	LCDC_FPR <sub>0</sub>	xx

### Description

Set the LSHIFT (pixel clock) frequency

LCDC\_FPR[19:16] : The highest 4 bits for the pixel clock frequency settings. (POR = 0111)

LCDC\_FPR[15:8] : The higher byte for the pixel clock frequency settings. (POR = 11111111)

LCDC\_FPR[7:0] : The low byte for the pixel clock frequency settings. (POR = 11111111)

Configure the pixel clock to PLL freq x ((LCDC\_FPR + 1) / 2<sup>20</sup>)

To obtain PCLK = 5.3MHz with PLL Frequency = 100MHz,

$$5.3\text{MHz} = 100\text{MHz} * (\text{LCDC\_FPR} + 1) / 2^{20}$$

$$\text{LCDC\_FPR} = 55574$$

WRITE COMMAND "0xE6"

WRITE DATA "0x00" (LCDC\_FPR = 55574)

WRITE DATA "0xD9"

WRITE DATA "0x16"

For serial LCD interface:

Configure the pixel clock to PLL freq x ((LCDC\_FPR + 1) / 2<sup>20</sup>) \*4

To obtain PCLK = 5.3MHz with PLL Frequency = 100MHz,

$$5.3\text{MHz} = 100\text{MHz} * ((\text{LCDC\_FPR} + 1) / 2^{20}) * 4$$

$$\text{LCDC\_FPR} = 13892$$

WRITE COMMAND "0xE6"

WRITE DATA "0x00" (LCDC\_FPR = 13892)

WRITE DATA "0x36"

WRITE DATA "0x44"

## 9.73 get\_lshift\_freq

**Command** 0xE7  
**Parameters** 3

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	1	0	0	1	1	1	E7
Parameter 1	1	0	0	0	0	LCDC_FPR <sub>19</sub>	LCDC_FPR <sub>18</sub>	LCDC_FPR <sub>17</sub>	LCDC_FPR <sub>16</sub>	xx
Parameter 2	1	LCDC_FPR <sub>15</sub>	LCDC_FPR <sub>14</sub>	LCDC_FPR <sub>13</sub>	LCDC_FPR <sub>12</sub>	LCDC_FPR <sub>11</sub>	LCDC_FPR <sub>10</sub>	LCDC_FPR <sub>9</sub>	LCDC_FPR <sub>8</sub>	xx
Parameter 3	1	LCDC_FPR <sub>7</sub>	LCDC_FPR <sub>6</sub>	LCDC_FPR <sub>5</sub>	LCDC_FPR <sub>4</sub>	LCDC_FPR <sub>3</sub>	LCDC_FPR <sub>2</sub>	LCDC_FPR <sub>1</sub>	LCDC_FPR <sub>0</sub>	xx

### Description

Get the current LSHIFT (pixel clock) frequency setting

LCDC\_FPR[19:16] : The highest 4 bits for the pixel clock frequency settings. (POR = 0111)  
LCDC\_FPR[15:8] : The higher byte for the pixel clock frequency settings. (POR = 11111111)  
LCDC\_FPR[7:0] : The low byte for the pixel clock frequency settings. (POR = 11111111)

## 9.74 set\_pixel\_data\_interface

**Command** 0xF0  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	1	1	0	0	0	0	F0
Parameter 1	1	0	0	0	0	0	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx

### Description

Set the pixel data format to 8-bit / 9-bit / 12-bit / 16-bit / 16-bit(565) / 18-bit in the parallel host processor interface. This command is used for display data only, the command format is always 8 bit.

A[2:0] : Pixel Data Interface Format (POR = 101)

000 8-bit  
001 12-bit  
010 16-bit packed  
011 16-bit (565 format)  
100 18-bit  
110 9-bit  
Others Reserved

\* Note : The un-used data bus will be driven to ground by SSD1961, so don't connect the un-used data bus to MCU.

## 9.75 get\_pixel\_data\_interface

**Command** 0xF1  
**Parameters** 1

	D/C	D7	D6	D5	D4	D3	D2	D1	D0	Hex
Command	0	1	1	1	1	0	0	0	1	F1
Parameter 1	1	0	0	0	0	0	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	xx

### Description

Get the current pixel data format settings in the parallel host processor interface.

A[2:0] : Pixel Data Interface Format (POR = 101)

000 8-bit  
001 12-bit  
010 16-bit packed  
011 16-bit (565 format)  
100 18-bit  
110 9-bit  
Others Reserved

## 10 MAXIMUM RATINGS

**Table 10-1: Maximum Ratings (Voltage Referenced to V<sub>SS</sub>)**

Symbol	Parameter	Value	Unit
V <sub>DDD</sub>	Digital Core power supply	-0.5 to 1.8	V
V <sub>DDPLL</sub>	PLL power supply	-0.5 to 1.8	V
V <sub>DDLCD</sub>	LCD Interface power supply	-0.5 to 4.6	V
V <sub>DDIO</sub>	I/O power supply	-0.5 to 4.6	V
V <sub>IN</sub>	Input Voltage	-0.5 to 4.6	V
V <sub>OUT</sub>	Output Voltage	-0.5 to 4.6	V
T <sub>A</sub>	Solder Temperature / Time	225 for 40 sec max at solder ball	°C
T <sub>STG</sub>	Storage temperature	-45 to 125	°C

Maximum ratings are those values beyond which damages to the device may occur. Functional operation should be restricted to the limits in the Electrical Characteristics tables or Pin Description section

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V<sub>IN</sub> and V<sub>OUT</sub> be constrained to the range V<sub>SS</sub> < (V<sub>IN</sub> or V<sub>OUT</sub>) < V<sub>DDIO</sub>. Reliability of operation is enhanced if unused input is connected to an appropriate logic voltage level (e.g., either V<sub>SS</sub> or V<sub>DDIO</sub>). Unused outputs must be left open. This device may be light sensitive. Caution should be taken to avoid exposure of this device to any light source during normal operation. This device is not radiation protected.

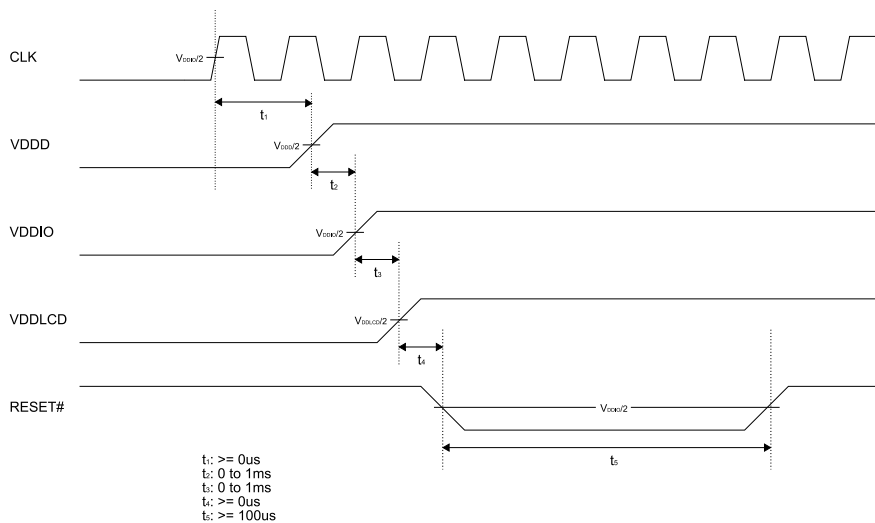
## 11 RECOMMENDED OPERATING CONDITIONS

**Table 11-1: Recommended Operating Condition**

Symbol	Parameter	Min	Typ	Max	Unit
V <sub>DDD</sub>	Digital Core power supply	1.10	1.2	1.30	V
V <sub>DDPLL</sub>	PLL power supply	1.10	1.2	1.30	V
V <sub>DDLCD</sub>	LCD Interface power supply	1.65	3.3	3.6	V
V <sub>DDIO</sub>	I/O power supply	1.65	3.3	3.6	V
T <sub>A</sub>	Operating temperature	-30	25	85	°C

### 11.1 Power-up sequence

**Figure 11-1: Power-up Sequence**



**Note**

Clock reference is only applicable when CLK is used.

## 12 DC CHARACTERISTICS

### Conditions:

Voltage referenced to  $V_{SS}$   
 $V_{DDD}, V_{DDPLL} = 1.2V$   
 $V_{DDIO}, V_{DDLCD} = 3.3V$   
 $T_A = 25^{\circ}C$

Table 12-1: DC Characteristics

Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
PSTY	Quiescent Power			300	500	uW
IIZ	Input leakage current		-1		1	uA
IOZ	Output leakage current		-1		1	uA
VOH	Output high voltage		$0.8V_{DDIO}$			V
VOL	Output low voltage				$0.2V_{DDIO}$	V
VIH	Input high voltage		$0.8V_{DDIO}$		$V_{DDIO} + 0.5$	V
VIL	Input low voltage				$0.2V_{DDIO}$	V

## 13 AC CHARACTERISTICS

### Conditions:

Voltage referenced to  $V_{SS}$   
 $V_{DDD}, V_{DDPLL} = 1.2V$   
 $V_{DDIO}, V_{DDLCD} = 3.3V$   
 $T_A = 25^{\circ}C$   
 $C_L = 50pF$  (Bus/CPU Interface)  
 $C_L = 0pF$  (LCD Panel Interface)

### 13.1 Clock Timing

Table 13-1: Clock Input Requirements for CLK (PLL-bypass)

Symbol	Parameter	Min	Max	Units
$F_{CLK}$	Input Clock Frequency (CLK)		110	MHz
$T_{CLK}$	Input Clock period (CLK)	$1/f_{CLK}$		ns

Table 13-2: Clock Input Requirements for CLK (PLL-enabled)

Symbol	Parameter	Min	Max	Units
$F_{CLK}$	Input Clock Frequency (CLK)	2.5	50	MHz
$T_{CLK}$	Input Clock period (CLK)	$1/f_{CLK}$		ns

Table 13-3: Clock Input Requirements for crystal oscillator XTAL (PLL-enabled)

Symbol	Parameter	Min	Max	Units
$F_{XTAL}$	Input Clock Frequency	2.5	10	MHz
$T_{XTAL}$	Input Clock period	$1/f_{XTAL}$		ns

## 13.2 MCU Interface Timing

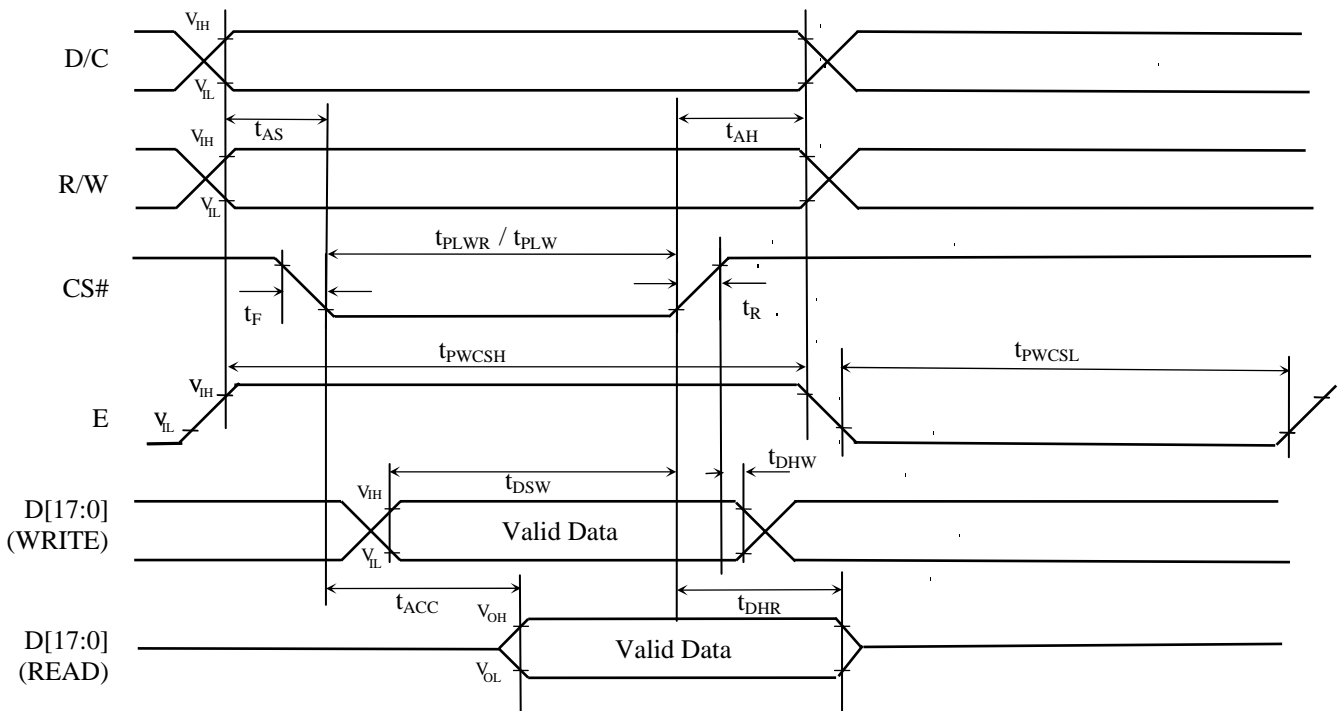
### 13.2.1 Parallel 6800-series Interface Timing

Table 13-4: Parallel 6800-series Interface Timing Characteristics (Use CS# as clock)

Symbol	Parameter	Min	Typ	Max	Unit																																																
$f_{MCLK}$	System Clock Frequency*	1	-	110	MHz																																																
$t_{MCLK}$	System Clock Period*	$1/f_{MCLK}$	-	-	ns																																																
$t_{PWCSH}$	Control Pulse High Width Write	13	$1.5 * t_{MCLK}$	-	ns																																																
	Control Pulse High Width Read	30	$3.5 * t_{MCLK}$	-	ns																																																
$t_{PWCSL}$	Control Pulse Low Width Write (next write cycle)	13	$1.5 * t_{MCLK}$	-	ns																																																
	Control Pulse Low Width Write (next read cycle)	80	$9 * t_{MCLK}$	-	ns																																																
	Control Pulse Low Width Read	80	$9 * t_{MCLK}$	-	ns																																																
$t_{AS}$	Address Setup Time	2	-	-	ns																																																
$t_{AH}$	Address Hold Time	2	-	-	ns																																																
$t_{DSW}$	Data Setup Time	4	-	-	ns </tr <tr> <td><math>t_{DHW}</math></td> <td>Data Hold Time</td> <td>1</td> <td>-</td> <td>-</td> <td>ns</td> </tr> <tr> <td><math>t_{PLW}</math></td> <td>Write Low Time</td> <td>14</td> <td>-</td> <td>-</td> <td>ns</td> </tr> <tr> <td><math>t_{PHW}</math></td> <td>Write High Time</td> <td>14</td> <td>-</td> <td>-</td> <td>ns</td> </tr> <tr> <td><math>t_{PLWR}</math></td> <td>Read Low Time</td> <td>38</td> <td>-</td> <td>-</td> <td>ns</td> </tr> <tr> <td><math>t_{ACC}</math></td> <td>Data Access Time</td> <td>32</td> <td>-</td> <td>-</td> <td>ns</td> </tr> <tr> <td><math>t_{DHR}</math></td> <td>Output Hold time</td> <td>1</td> <td>-</td> <td>-</td> <td>ns</td> </tr> <tr> <td><math>t_R</math></td> <td>Rise Time</td> <td>-</td> <td>-</td> <td>0.5</td> <td>ns</td> </tr> <tr> <td><math>t_F</math></td> <td>Fall Time</td> <td>-</td> <td>-</td> <td>0.5</td> <td>ns</td> </tr>	$t_{DHW}$	Data Hold Time	1	-	-	ns	$t_{PLW}$	Write Low Time	14	-	-	ns	$t_{PHW}$	Write High Time	14	-	-	ns	$t_{PLWR}$	Read Low Time	38	-	-	ns	$t_{ACC}$	Data Access Time	32	-	-	ns	$t_{DHR}$	Output Hold time	1	-	-	ns	$t_R$	Rise Time	-	-	0.5	ns	$t_F$	Fall Time	-	-	0.5	ns
$t_{DHW}$	Data Hold Time	1	-	-	ns																																																
$t_{PLW}$	Write Low Time	14	-	-	ns																																																
$t_{PHW}$	Write High Time	14	-	-	ns																																																
$t_{PLWR}$	Read Low Time	38	-	-	ns																																																
$t_{ACC}$	Data Access Time	32	-	-	ns																																																
$t_{DHR}$	Output Hold time	1	-	-	ns																																																
$t_R$	Rise Time	-	-	0.5	ns																																																
$t_F$	Fall Time	-	-	0.5	ns																																																

\* System Clock denotes external input clock (PLL-bypass) or internal generated clock (PLL-enabled)

Figure 13-1: Parallel 6800-series Interface Timing Diagram (Use CS# as Clock)

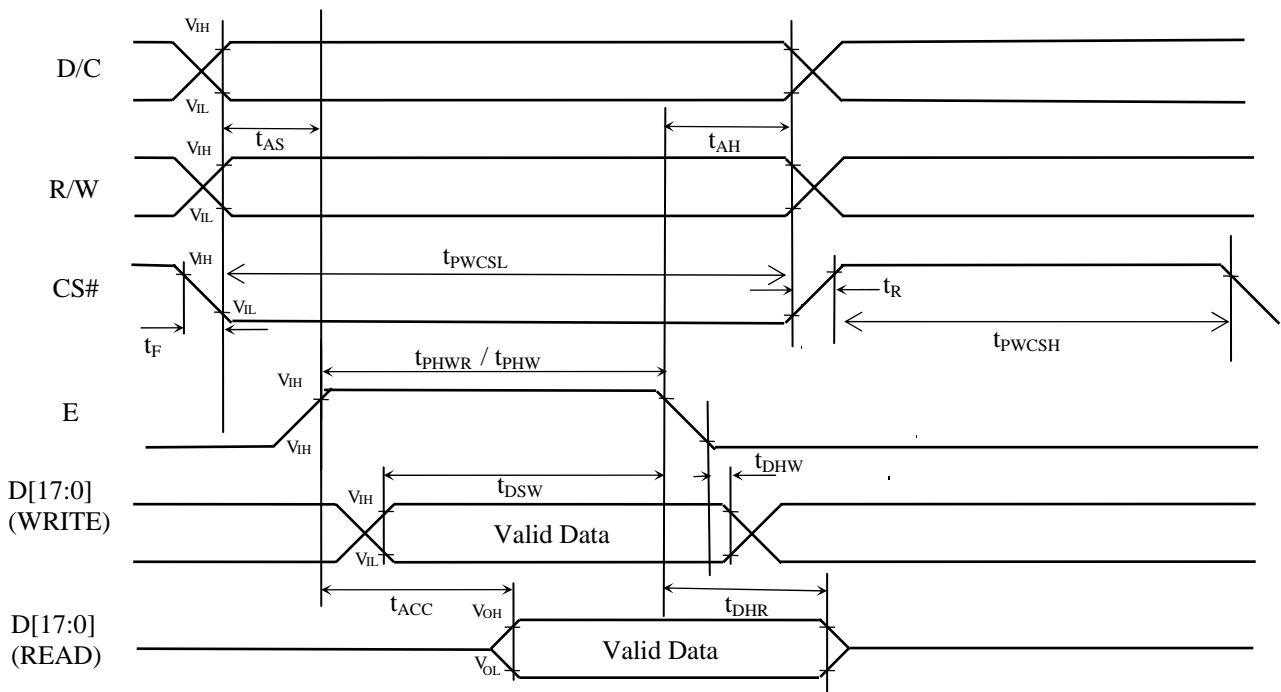


**Table 13-5: Parallel 6800-series Interface Timing Characteristics (Use E as clock)**

Symbol	Parameter	Min	Typ	Max	Unit
$f_{MCLK}$	System Clock Frequency*	1	-	110	MHz
$t_{MCLK}$	System Clock Period*	$1/f_{MCLK}$	-	-	ns
$t_{PWCSH}$	Control Pulse Low Width	13	$1.5^* t_{MCLK}$	-	ns
	Write (next write cycle)	80	$9^* t_{MCLK}$	-	
	Write (next read cycle)	80	$9^* t_{MCLK}$	-	
$t_{PWCSL}$	Control Pulse High Width	13	$1.5^* t_{MCLK}$	-	ns
	Write	30	$3.5^* t_{MCLK}$	-	
$t_{AS}$	Address Setup Time	2	-	-	ns
$t_{AH}$	Address Hold Time	2	-	-	ns
$t_{DSW}$	Data Setup Time	4	-	-	ns
$t_{DHW}$	Data Hold Time	1	-	-	ns
$t_{PLW}$	Write Low Time	14	-	-	ns
$t_{PHW}$	Write High Time	14	-	-	ns
$t_{PLWR}$	Read Low Time	38	-	-	ns
$t_{ACC}$	Data Access Time	32	-	-	ns
$t_{DHR}$	Output Hold time	1	-	-	ns
$t_R$	Rise Time	-	-	0.5	ns
$t_F$	Fall Time	-	-	0.5	ns

\* System Clock denotes external input clock (PLL-bypass) or internal generated clock (PLL-enabled)

**Figure 13-2: Parallel 6800-series Interface Timing Diagram (Use E as Clock)**



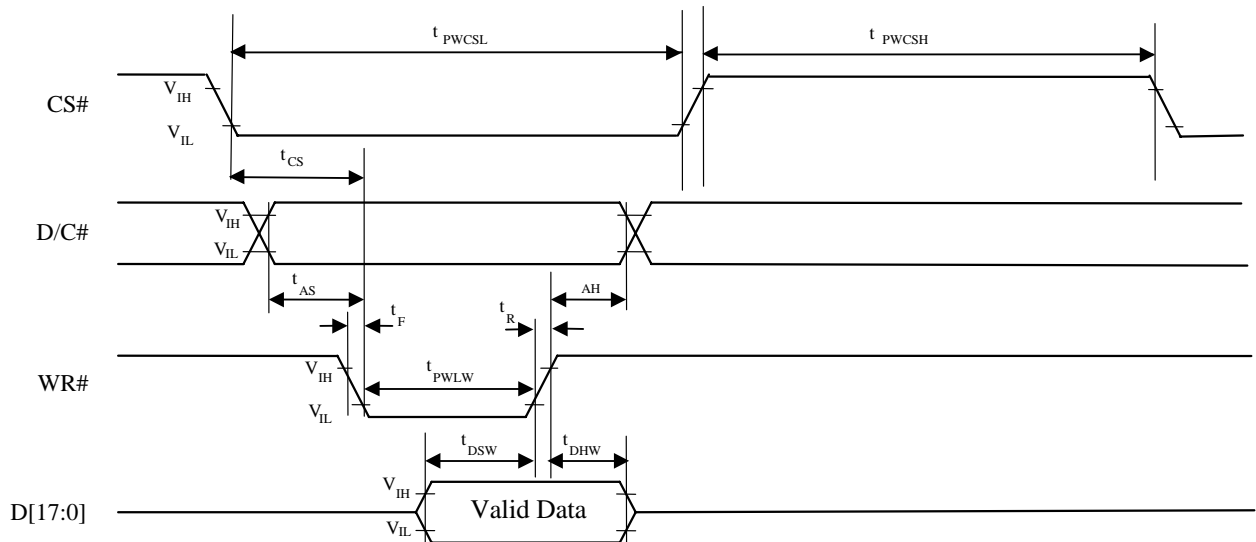
### 13.2.2 Parallel 8080-series Interface Timing

Table 13-6: Parallel 8080-series Interface Timing Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
$f_{MCLK}$	System Clock Frequency*	1	-	110	MHz
$t_{MCLK}$	System Clock Period*	$1/f_{MCLK}$	-	-	ns
$t_{PWCSL}$	Control Pulse High Width Write Read	13 30	$1.5 * t_{MCLK}$ $3.5 * t_{MCLK}$	-	ns
$t_{PWCSH}$	Control Pulse Low Width Write (next write cycle) Write (next read cycle) Read	13 80 80	$1.5 * t_{MCLK}$ $9 * t_{MCLK}$ $9 * t_{MCLK}$	-	ns
$t_{AS}$	Address Setup Time	1	-	-	ns
$t_{AH}$	Address Hold Time	2	-	-	ns
$t_{DSW}$	Write Data Setup Time	4	-	-	ns
$t_{DHW}$	Write Data Hold Time	1	-	-	ns
$t_{PWLW}$	Write Low Time	12	-	-	ns
$t_{DHR}$	Read Data Hold Time	1	-	-	ns
$t_{ACC}$	Access Time	32	-	-	ns
$t_{PWLR}$	Read Low Time	36	-	-	ns
$t_R$	Rise Time	-	-	0.5	ns
$t_F$	Fall Time	-	-	0.5	ns
$t_{CS}$	Chip select setup time	2	-	-	ns
$t_{CSH}$	Chip select hold time to read signal	3	-	-	ns

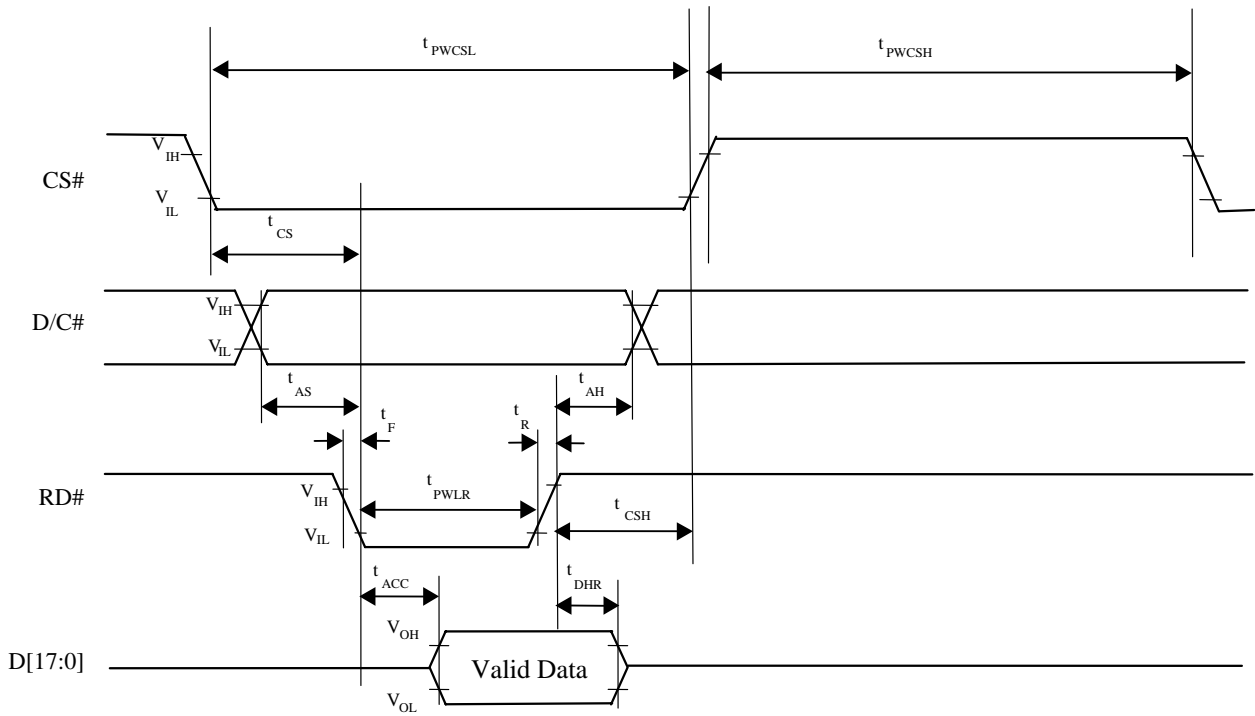
\* System Clock denotes external input clock (PLL-bypass) or internal generated clock (PLL-enabled)

Figure 13-3: Parallel 8080-series Interface Timing Diagram (Write Cycle)



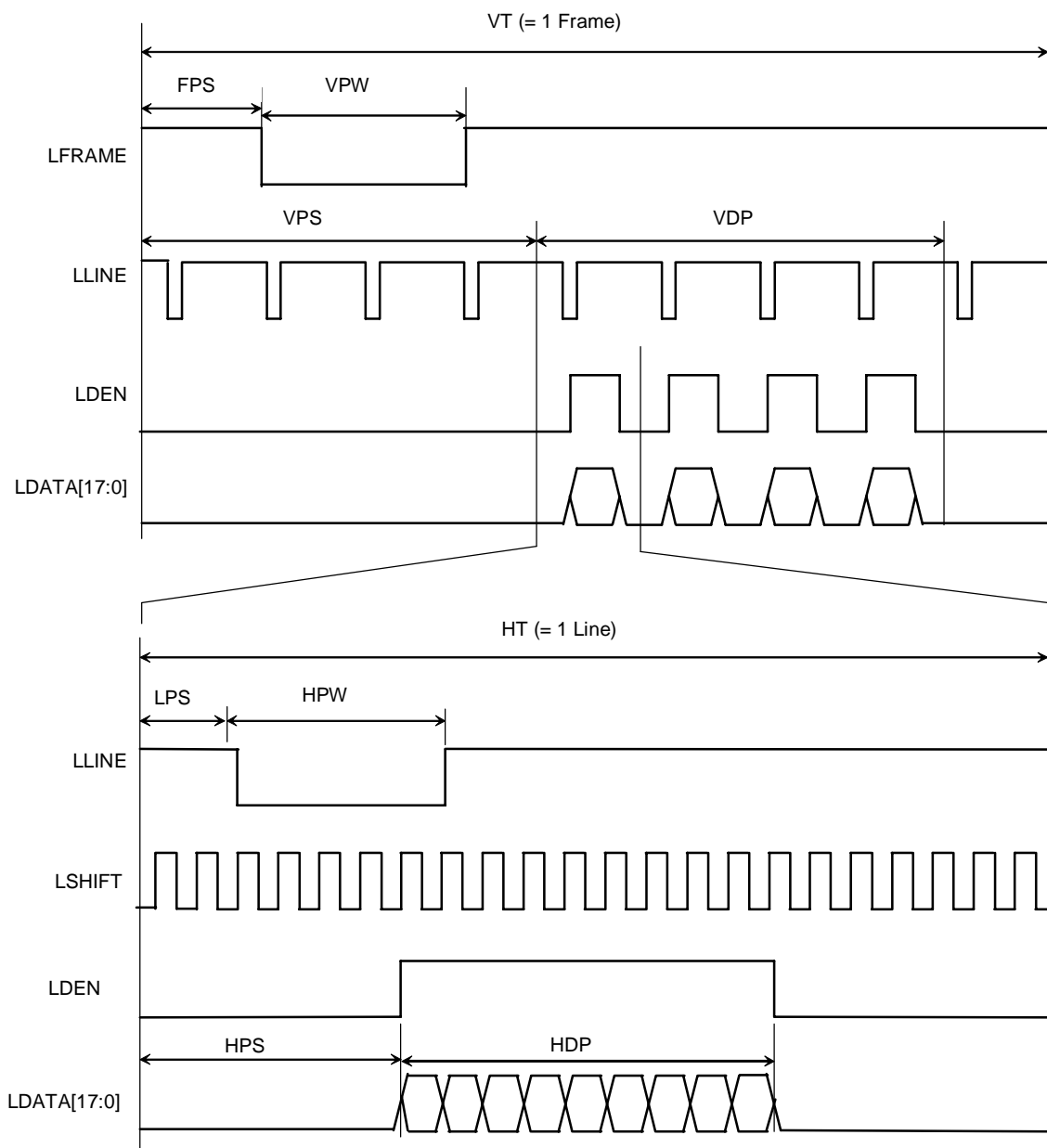


**Figure 13-4: Parallel 8080-series Interface Timing Diagram (Read Cycle)**



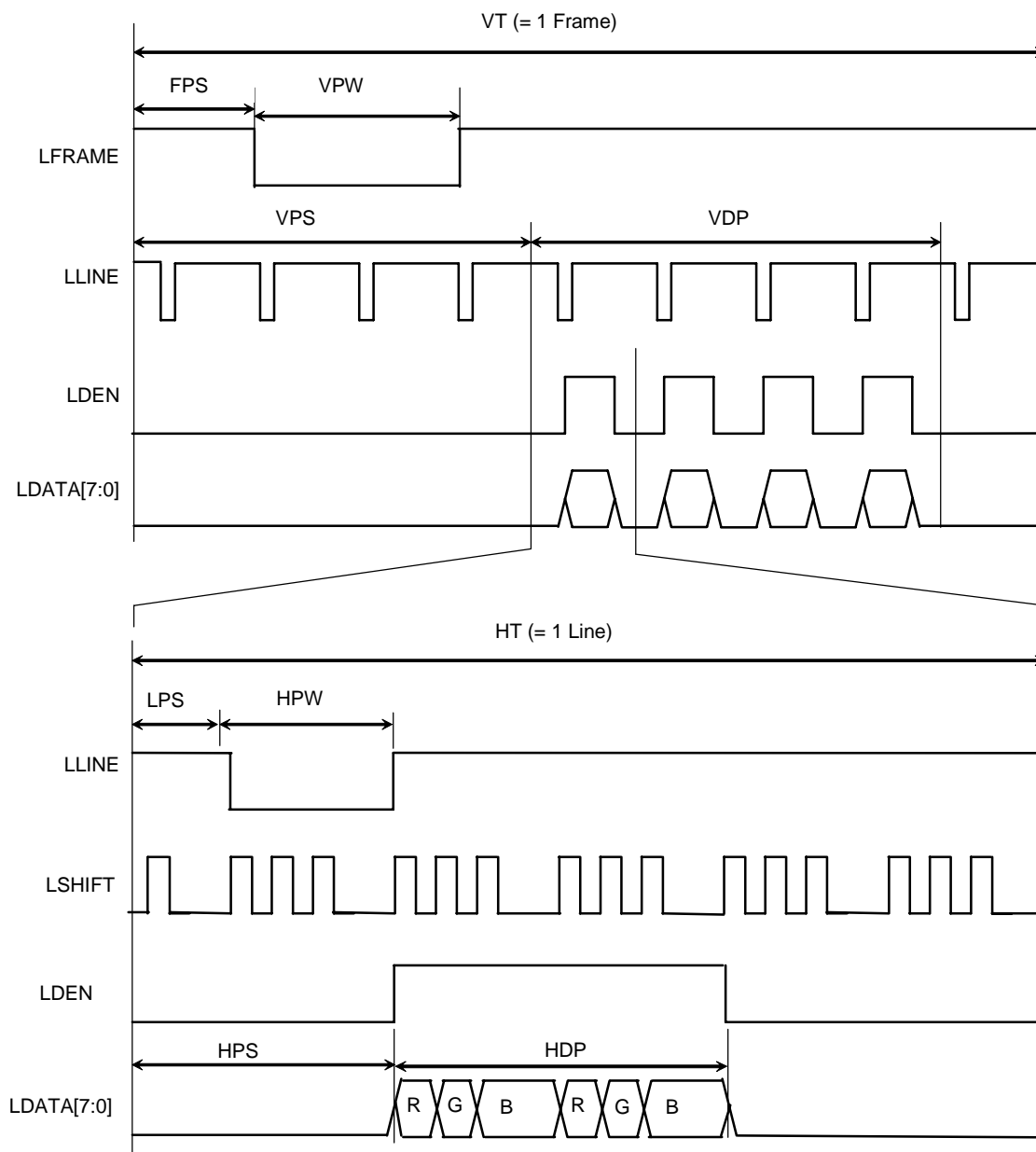
### 13.3 Parallel LCD Interface Timing

Figure 13-5: Generic TFT Panel Timing

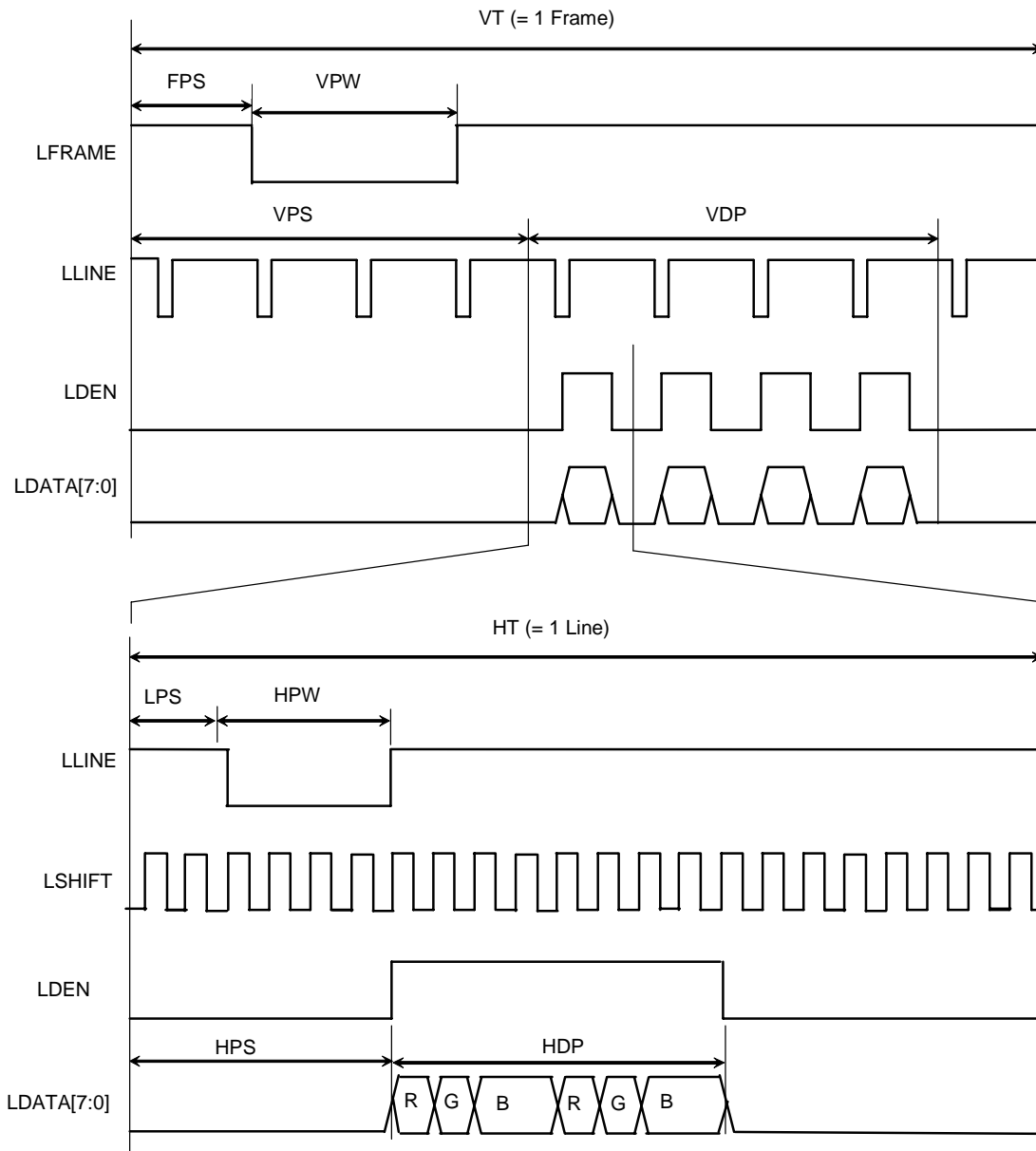


### 13.4 Serial RGB Interface Timing

Figure 13-6: Serial RGB Interface Timing (without dummy mode)



**Figure 13-7: Serial RGB Interface Timing (with dummy mode)**



**Table 13-7: Quick reference table for LCD parameter setting**

LCD Parameter setting	Corresponding Command	Register	Maximum Setting	
			Parallel mode	Serial mode
VDP	0xB0	VDP[10:0]	2048 Vertical panel size = (VDP + 1) pixels	
HDP		HDP[10:0]	2048 Horizontal panel size = (HDP + 1) pixels	
LPS	0xB4	LPS[10:0]	2047	8188 (2047 x 4)
HPW		HPW[6:0]	128 (HPW + 1) pixels	512 (128 x4)
HPS		HPS[10:0]	2047	8188 (2047 x 4)
HT		HT[10:0]	2048 Horizontal Total = (HT + 1) lines	
FPS	0xB6	FPS[10:0]	2047	
VPW		VPW[6:0]	128 Vertical Sync Pulse Width = (VPW+1) lines	
VPS		VPS[10:0]	2047	
VT		VT[10:0]	2048 Vertical Total = (VT + 1) lines	

# 14 APPLICATION EXAMPLE

Figure 14-1: Application circuit for SSD1961 (With Direct clock input)

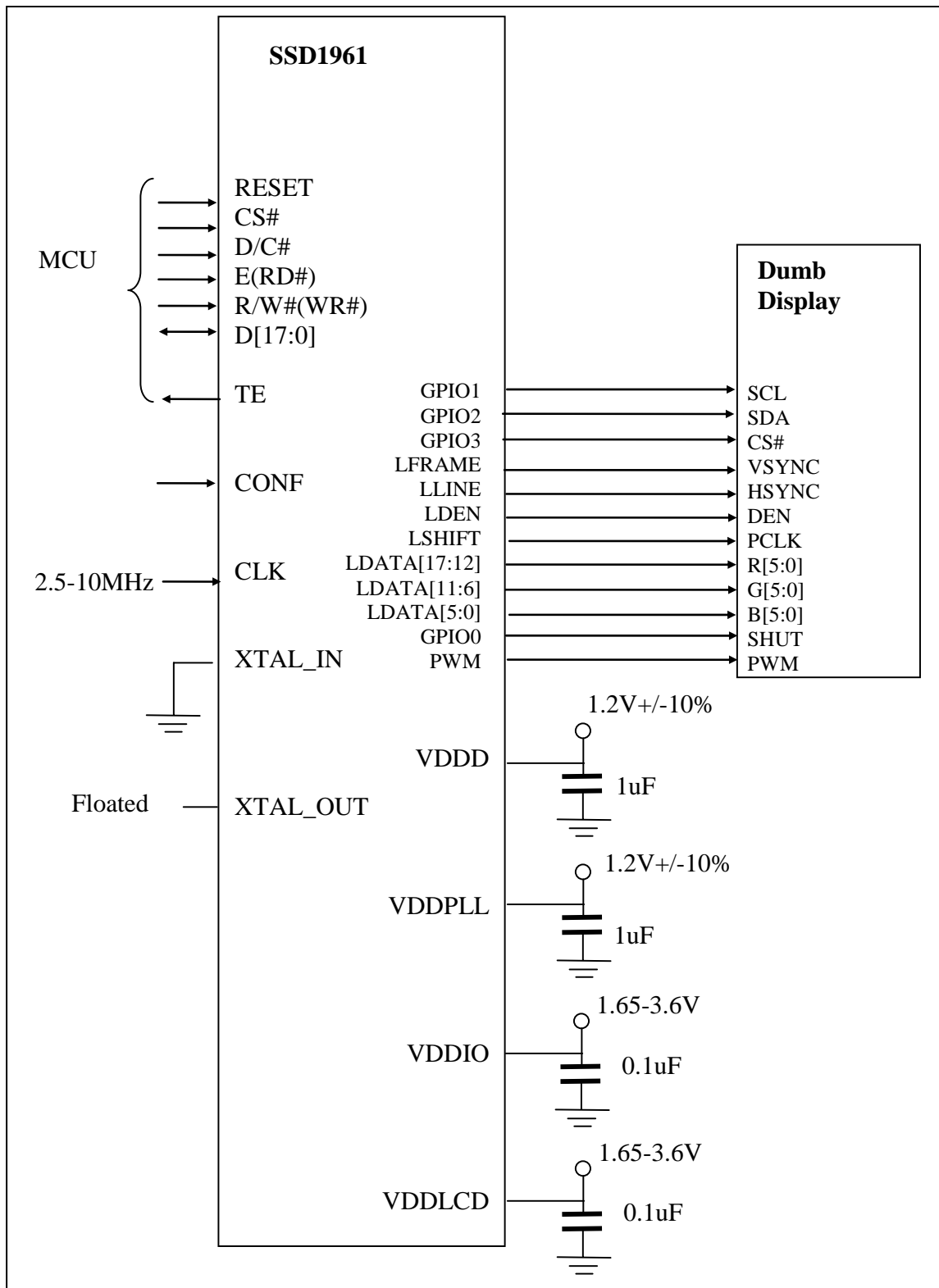
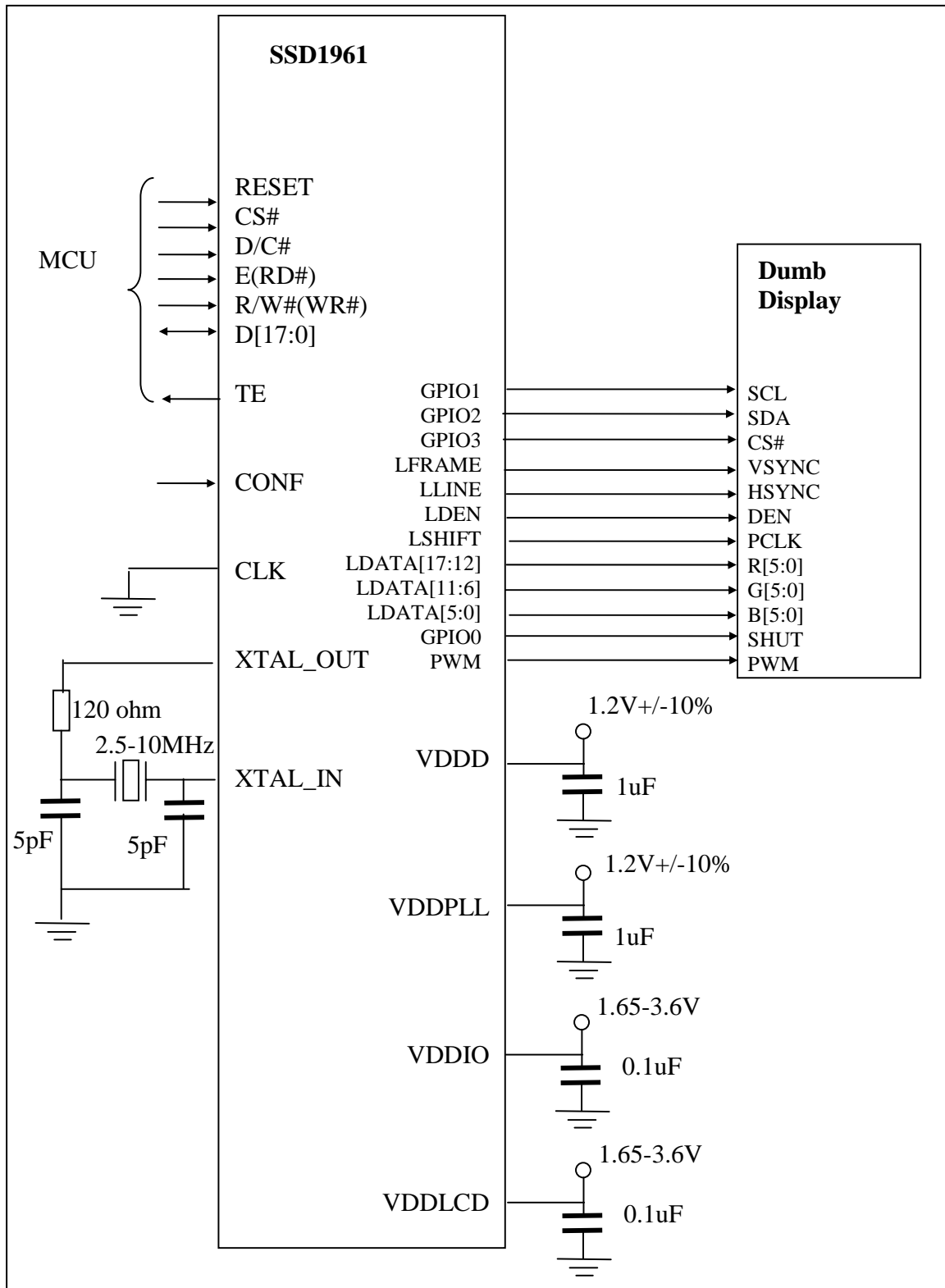
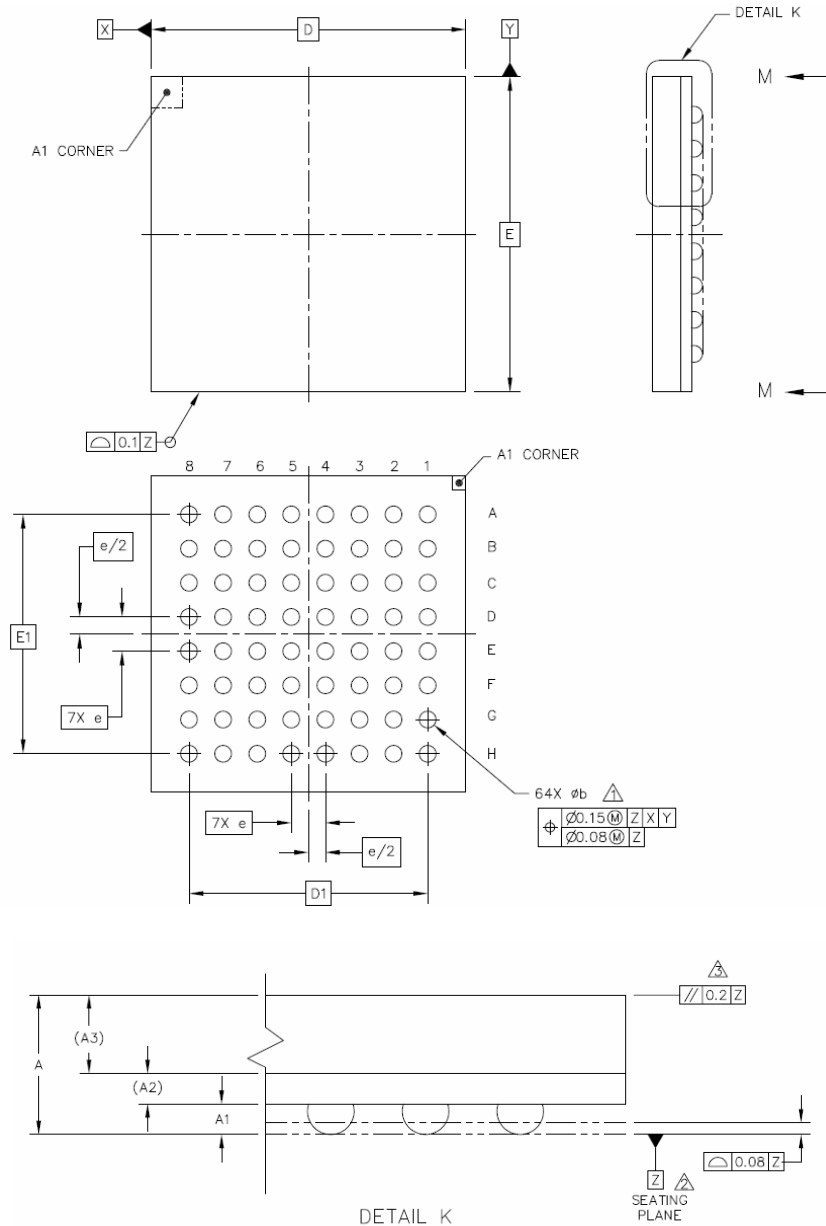


Figure 14-2: Application circuit for SSD1961 (With crystal oscillator input)



# 15 PACKAGE INFORMATION

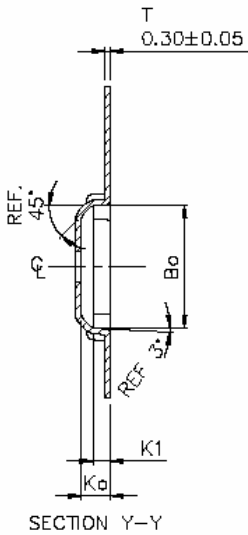
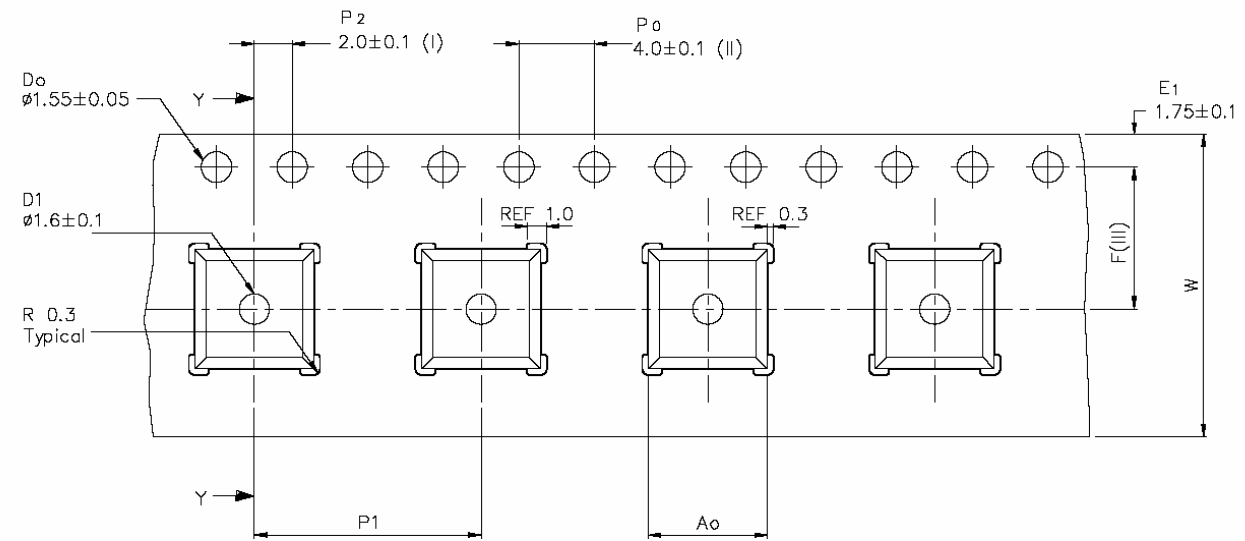
## 15.1 Package Mechanical Drawing for TFBGA



Dimension in mm			
Symbol	Min	Typical	Max
A	--	---	1.1
A1	0.16	---	0.26
A2	---	0.21	---
A3	---	0.54	---
b	0.27	---	0.37
D	---	6.00 BSC	---
E	---	6.00 BSC	---
e	---	0.65	---
D1	---	4.55 BSC	---
E1	---	4.55 BSC	---



## 15.2 Tape & Reel Drawing for TFBGA



- (I) Measured from centreline of sprocket hole to centreline of pocket.
- (II) Cumulative tolerance of 10 sprocket holes is  $\pm 0.20$ .
- (III) Measured from centreline of sprocket hole to centreline of pocket.
- (IV) Other material available.

$A_0$	$6.30 + / - 0.1$
$B_0$	$6.30 + / - 0.1$
$K_0$	$1.50 + / - 0.1$
$K_1$	$0.90 + / - 0.1$
$F$	$7.50 + / - 0.1$
$P_1$	$12.00 + / - 0.1$
$W$	$16.00 + / - 0.3$

ALL DIMENSIONS IN MILLIMETRES UNLESS OTHERWISE STATED.

Solomon Systech reserves the right to make changes without notice to any products herein. Solomon Systech makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Solomon Systech assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any, and all, liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typical" must be validated for each customer application by the customer's technical experts. Solomon Systech does not convey any license under its patent rights nor the rights of others. Solomon Systech products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Solomon Systech product could create a situation where personal injury or death may occur. Should Buyer purchase or use Solomon Systech products for any such unintended or unauthorized application, Buyer shall indemnify and hold Solomon Systech and its offices, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Solomon Systech was negligent regarding the design or manufacture of the part.



The product(s) listed in this datasheet comply with Directive 2002/95/EC of the European Parliament and of the council of 27 January 2004 on the restriction of the use of certain hazardous substances in electrical and electronic equipment and People's Republic of China Electronic Industry Standard SJ/T 11363-2006 "Requirements for concentration limits for certain hazardous substances in electronic information products (电子信息产品中有毒有害物质的限量要求)". Hazardous Substances test report is available upon request.

<http://www.solomon-systech.com>