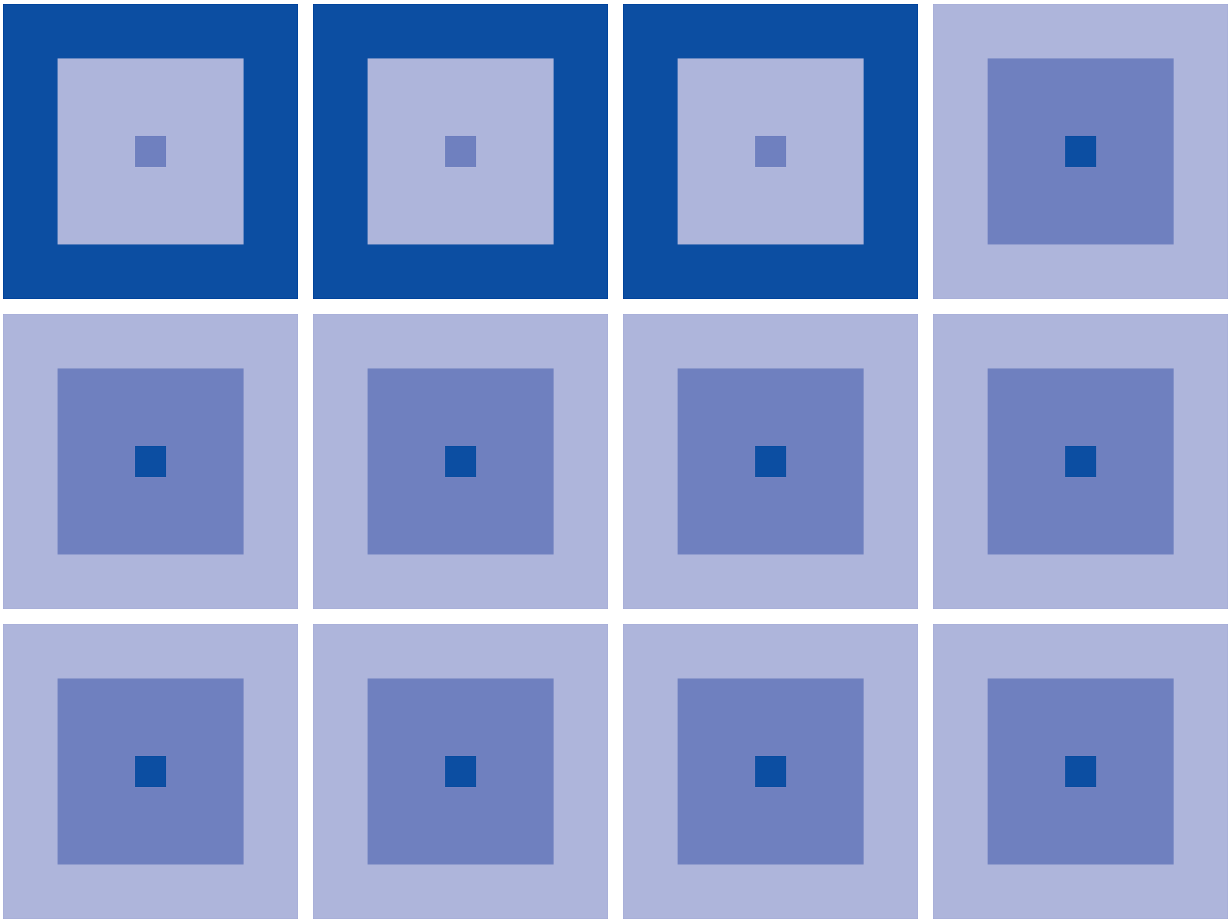
MF1430-01

EPSON

Embedded DRAM Graphics Controller

S1D13806 Series

Technical Manual



ENERGY SAVING
EPSON

SEIKO EPSON CORPORATION

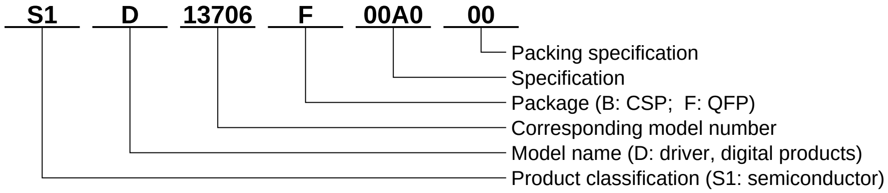
NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of International Trade and Industry or other approval from another government agency.

New configuration of product number

Starting April 1, 2001, the configuration of product number descriptions will be changed as listed below. To order from April 1, 2001 please use these product numbers. For further information, please contact Epson sales representative.

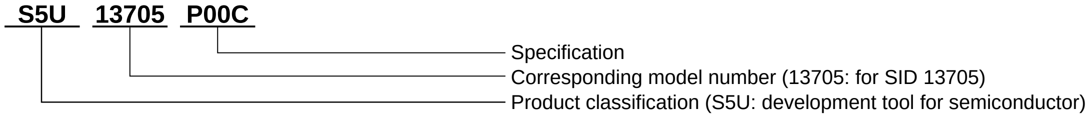
Devices



S1 D 13706 F 00A0 00

- Packing specification
- Specification
- Package (B: CSP; F: QFP)
- Corresponding model number
- Model name (D: driver, digital products)
- Product classification (S1: semiconductor)

Evaluation Board



S5U 13705 P00C

- Specification
- Corresponding model number (13705: for SID 13705)
- Product classification (S5U: development tool for semiconductor)

S1D13806 Series Technical Manual

S1D13806 Series

Embedded DRAM Graphics Controller

Hardware Functional Specification

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Scope

This User Guide provides technical information for the S1D13806 Embedded Memory Display Controller. Included in this document are timing diagrams, AC and DC characteristics, register descriptions, and power management descriptions. This document is intended for two audiences: Video Subsystem Designers and Software Developers.

1.2 Overview Description

The S1D13806 is a highly integrated color LCD/CRT/TV graphics controller with embedded memory supporting a wide range of CPUs and display devices. The S1D13806 architecture is designed to meet the low cost, low power requirements of the embedded markets, such as Mobile Communications, Hand-Held PC's, and Office Automation.

The S1D13806 supports multiple CPUs, all LCD panel types, CRT, TV, and additionally provides a number of differentiating features. EPSON Independent Simultaneous Display technology allows the user to configure two different images on two different displays, while the SwivelView™, Hardware Cursor, Ink Layer, and BitBLT features offer substantial performance benefits. Products requiring digital camera input can take advantage of the directly supported WINNOV Videum® Cam digital interface. While focusing on devices targeted by the Microsoft Windows CE Operating System, the S1D13806's impartiality to CPU type or operating system makes it an ideal display solution for a wide variety of applications.

2 FEATURES

Table 2-1 S1D13806 Features

S1D13806 Features	
Embedded Memory 1280K byte embedded synchronous DRAM. Up to 50MHz data rate (100M Bytes/second). Display buffer address space is directly and contiguously available through the 21-bit address bus.	**Display Modes** 4/8/16 bit-per-pixel (bpp) color depths. Up to 64K colors on TFT,CRT and TV. Up to 64K colors in 16 bpp mode on color passive LCD panels using dithering (4096 colors in 4/8 bpp). Up to 64 shades of gray on monochrome passive panels using Frame Rate Modulation (FRM) and Dithering. 4/8 bit-per-pixel color depths are mapped using three 256x4 Look-Up Tables (LUT). Separate LUTs for LCD and CRT/TV. 16 bit-per-pixel modes are mapped directly bypassing the LUT.
CPU Interfaces Epson E0C33. Hitachi SH-4. Hitachi SH-3. MIPS/ISA. Motorola MC68000. Motorola MC68030. Motorola PowerPC MPC82x. MPU bus interface with programmable READY. NEC MIPS VR41xx. PC Card (PCMCIA). Philips MIPS PR31500/PR31700. StrongARM (PC Card). Toshiba MIPS TX39xx. One-stage write buffer for minimum wait-state CPU writes. Registers are memory-mapped – M/R# pin selects between display buffer and register address space.	**Display Features** SwivelView™: 90°, 180°, 270° hardware rotation of display image. EPSON Independent Simultaneous Display (EISD): displays independent images on different displays (CRT or TV and passive or TFT panel). Virtual Display Support: displays images larger than the physical display size through the use of panning and scrolling. 2D BitBLT Engine. Hardware Cursor/Ink Layer: separate 64x64x2 hardware cursor or 2-bit ink layer for both LCD and CRT/TV. Double Buffering/Multi-pages: for smooth animation and instantaneous screen update.
Display Support 4/8-bit monochrome or 4/8/16-bit color LCD interface for single-panel, single-drive displays. 8-bit monochrome or 8/16-bit color LCD interface for dual-panel, dual-drive displays. Direct support for 9-bit, 12-bit, 18-bit, 2x9-bit, 2x12-bit TFT/D-TFD. Direct support for CRT. Direct support for S-Video/Composite TV output (NTSC or PAL format).	**Miscellaneous** Power save mode is initiated by software. Built-in MediaPlug Interface for Winnov Camera. Highly Flexible Clocking. Eight configuration pins (CONF[7:0]) are used to configure the chip at power-on. 13 General Purpose Input/Output pins. Operating voltage from 3.0 volts to 3.6 volts. 144-pin QFP20 package.

3 Typical System Implementation Diagrams

For pin mapping of each system implementation, see Table 4-9, “CPU Interface Pin Mapping,” on page 18.

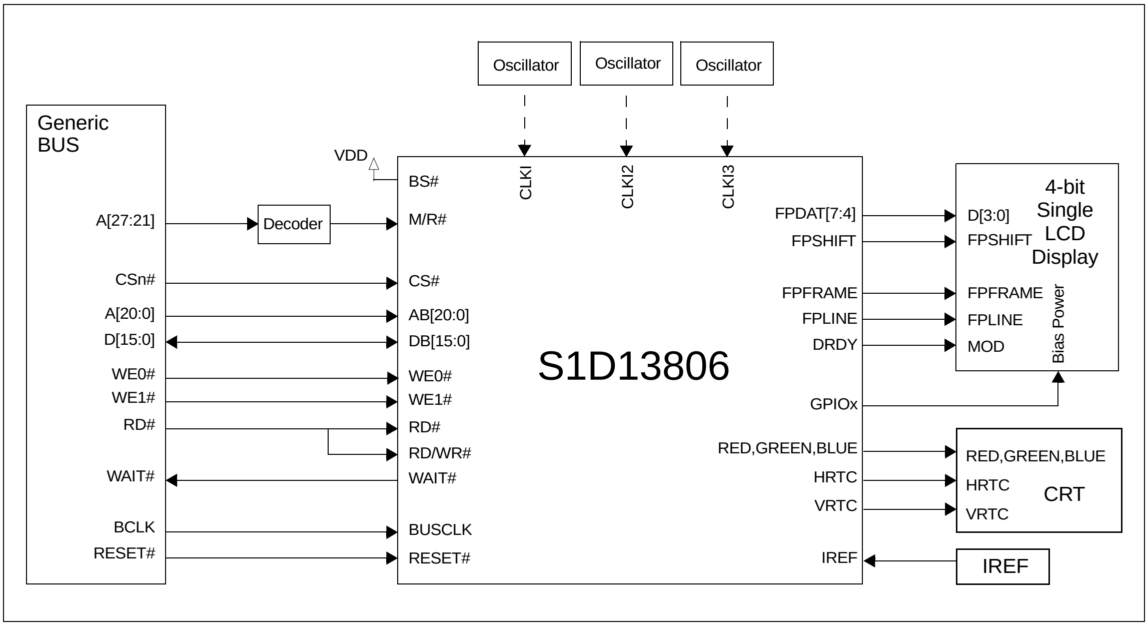


Figure 3-1 Typical System Diagram (Generic Bus)

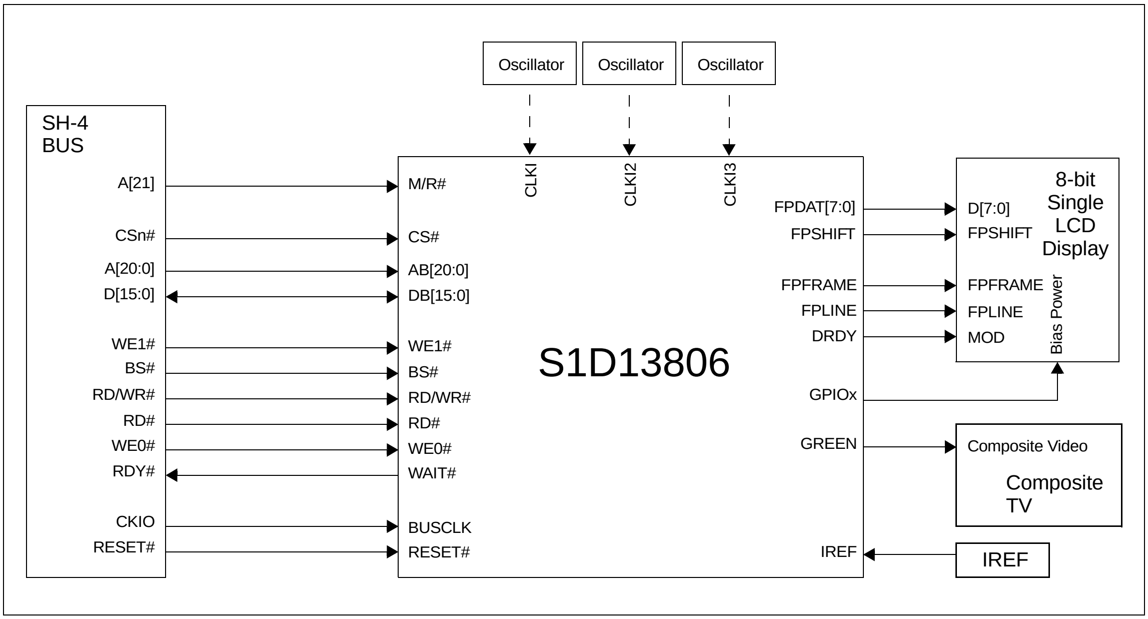


Figure 3-2 Typical System Diagram (Hitachi SH-4 Bus)

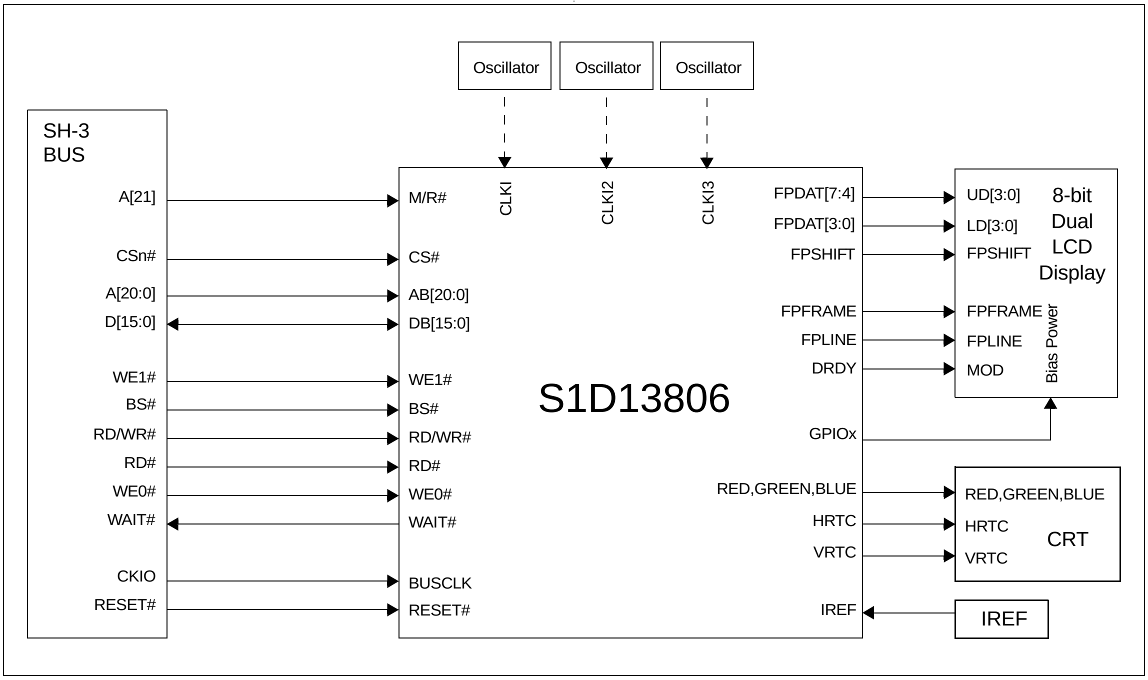


Figure 3-3 Typical System Diagram (Hitachi SH-3 Bus)

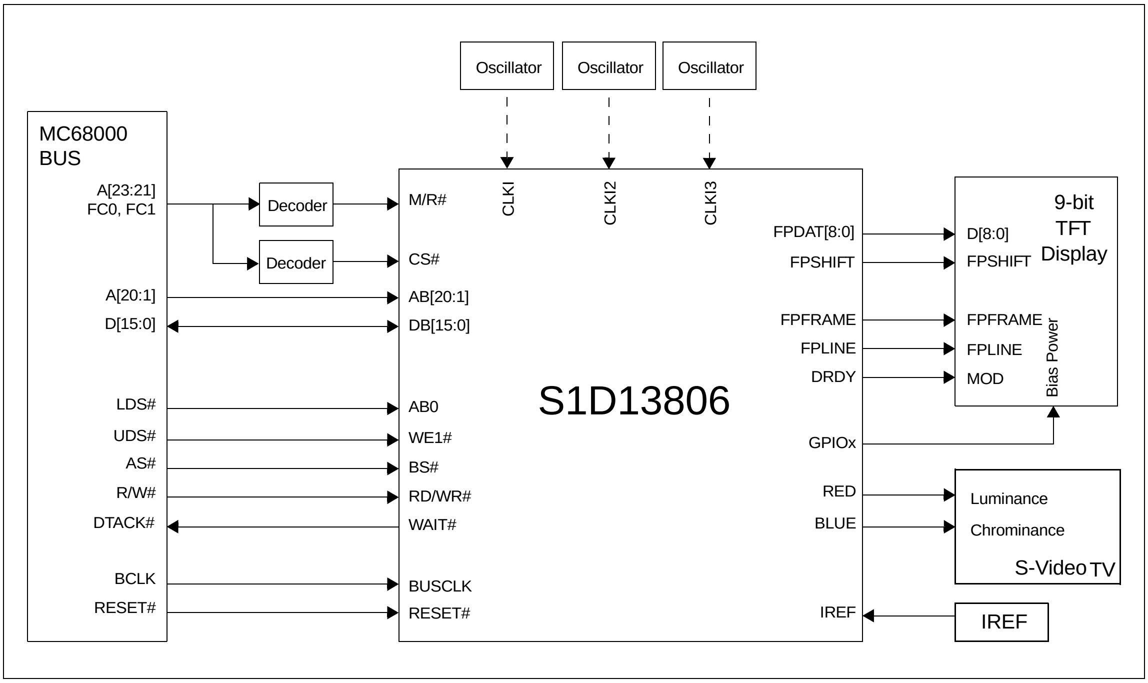


Figure 3-4 Typical System Diagram (MC68K Bus 1, Motorola 16-Bit 68000)

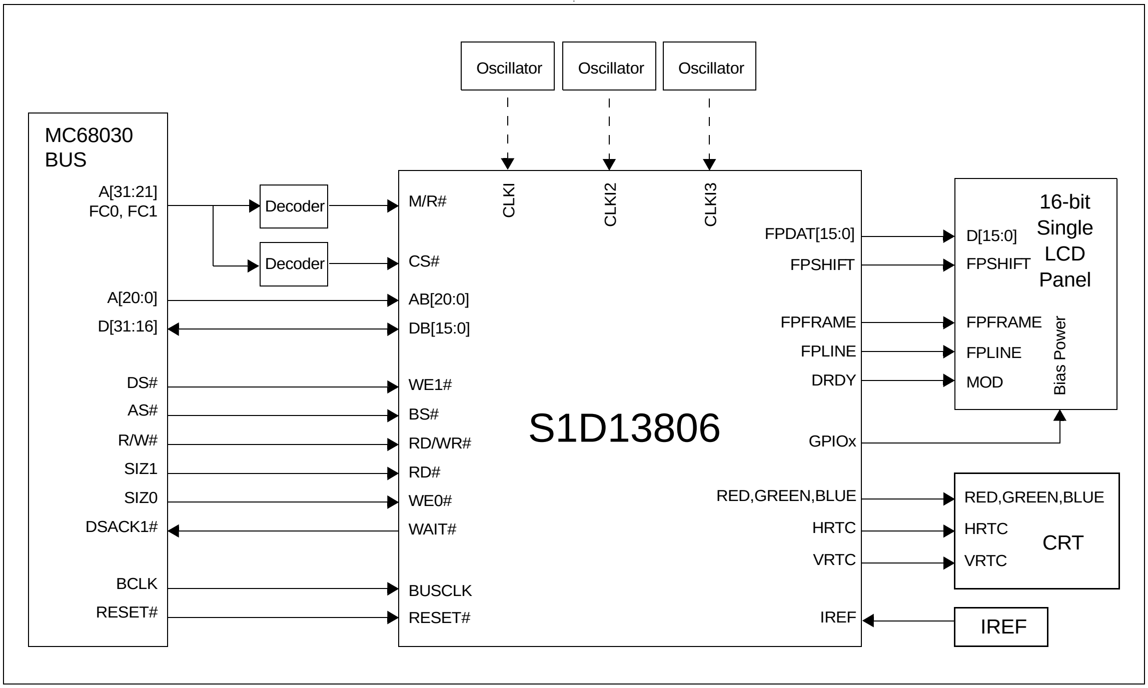


Figure 3-5 Typical System Diagram (MC68K Bus 2, Motorola 32-Bit 68030)

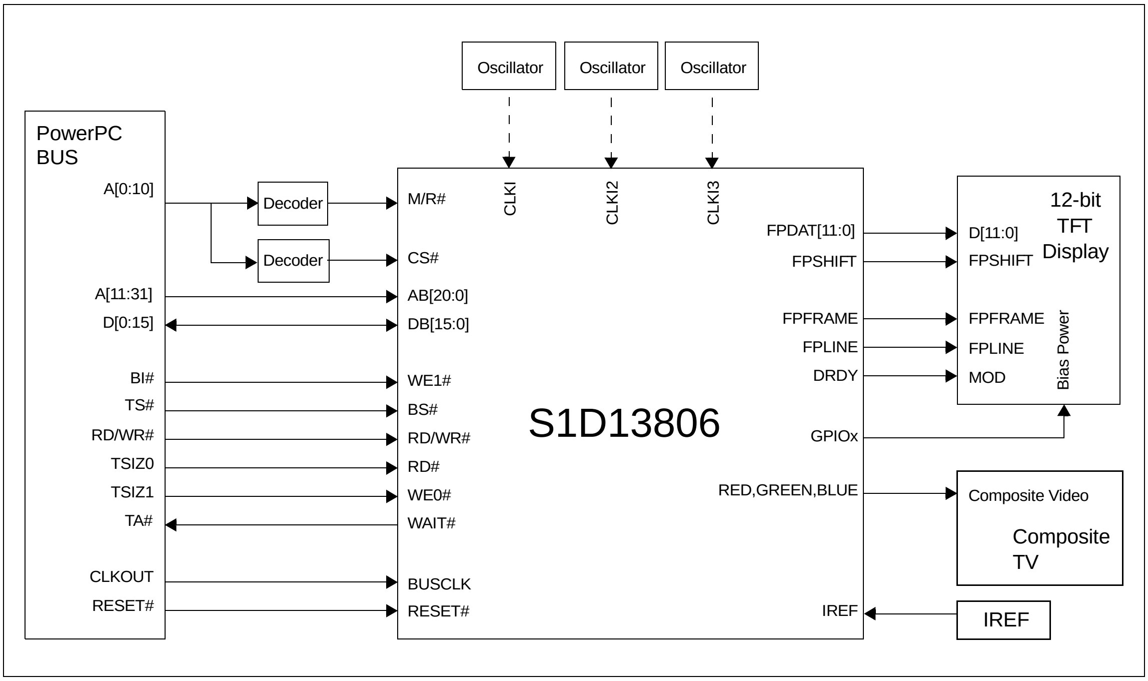


Figure 3-6 Typical System Diagram (Motorola Power PC Bus)

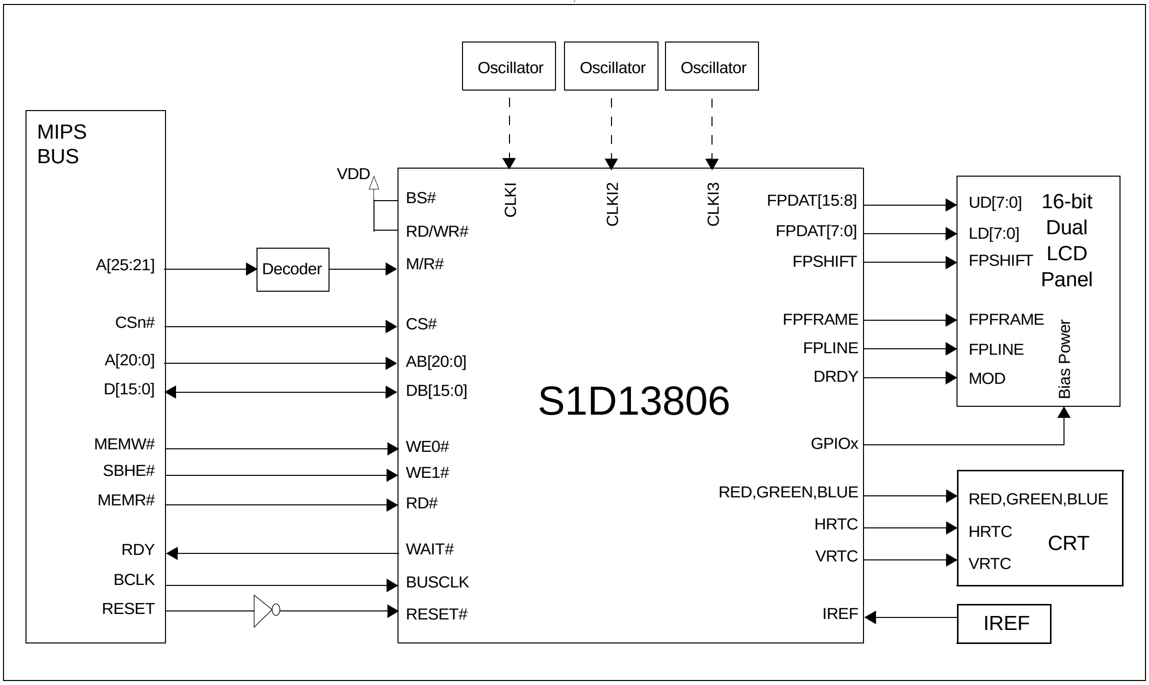


Figure 3-7 Typical System Diagram (NEC MIPS VR41xx Bus)

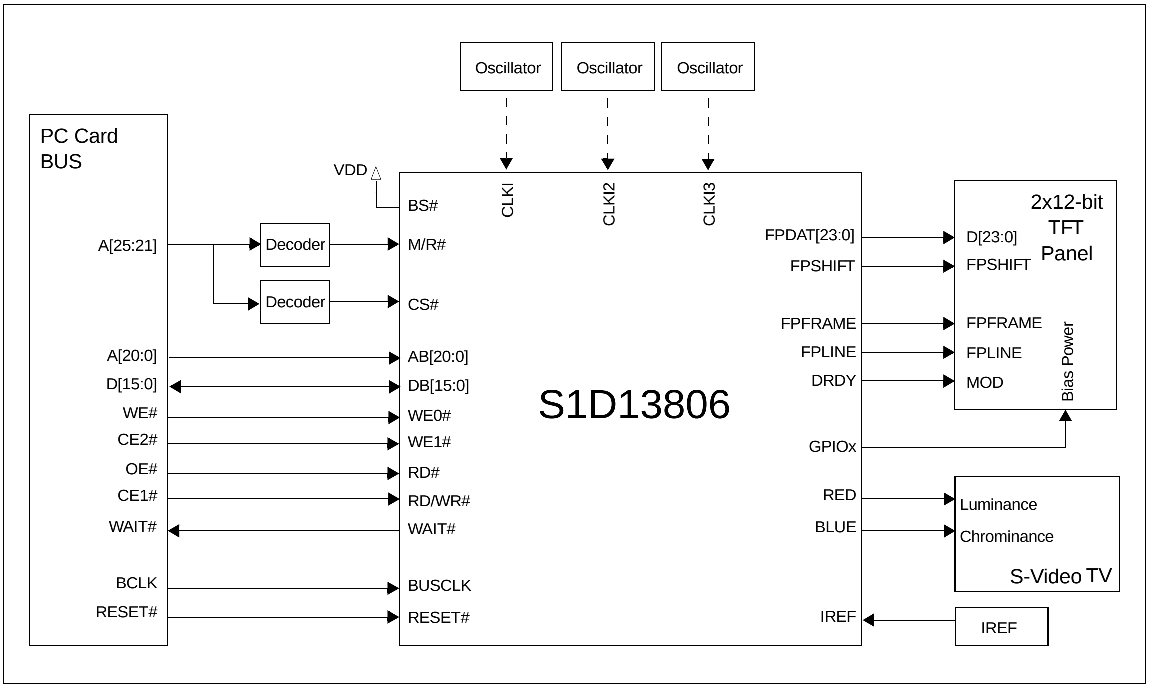


Figure 3-8 Typical System Diagram (PC Card Bus)

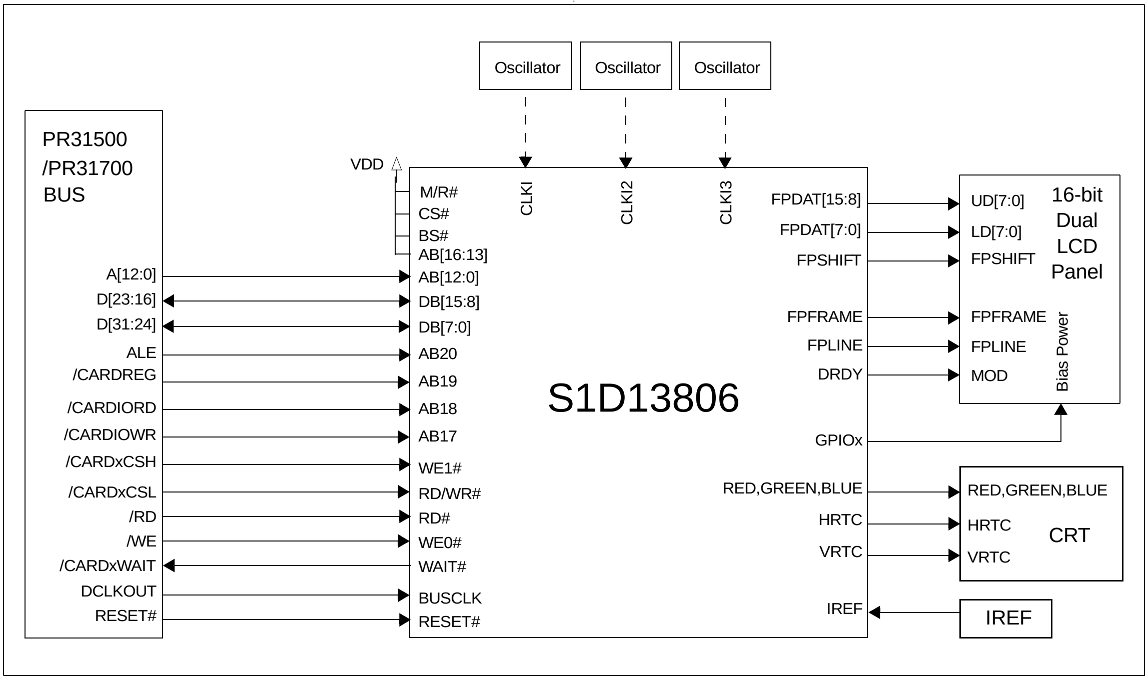


Figure 3-9 Typical System Diagram (Philips MIPS PR31500/PR31700 Bus)

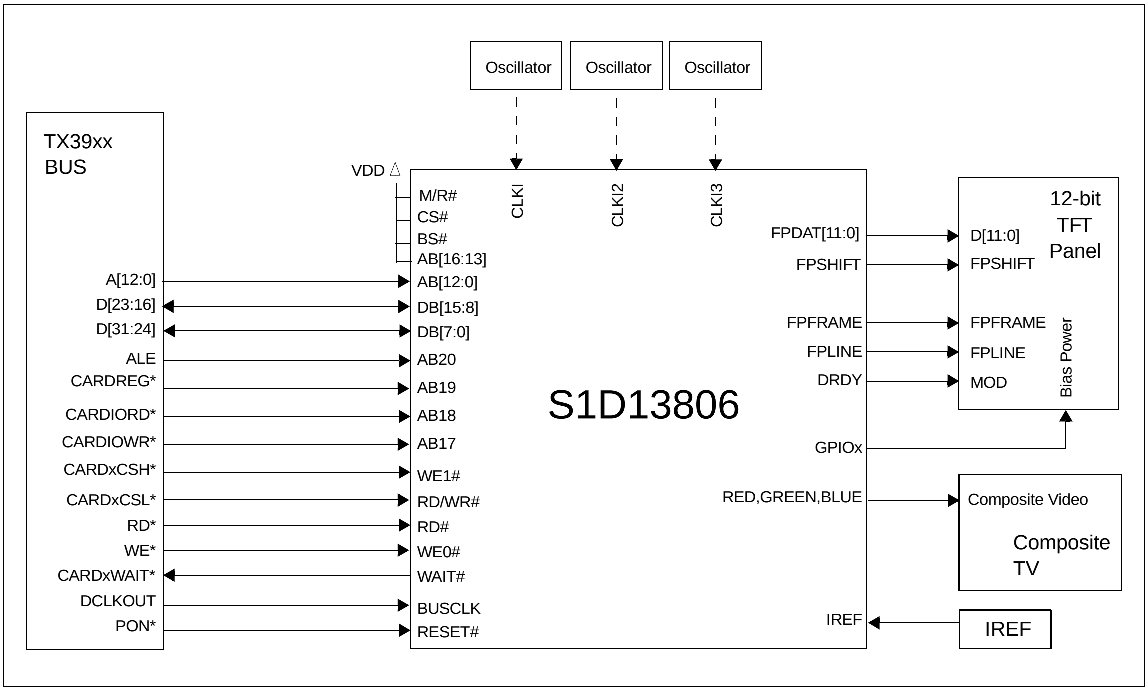


Figure 3-10 Typical System Diagram (Toshiba MIPS TX39xx Bus)

4 PINS

4.1 Pinout Diagram

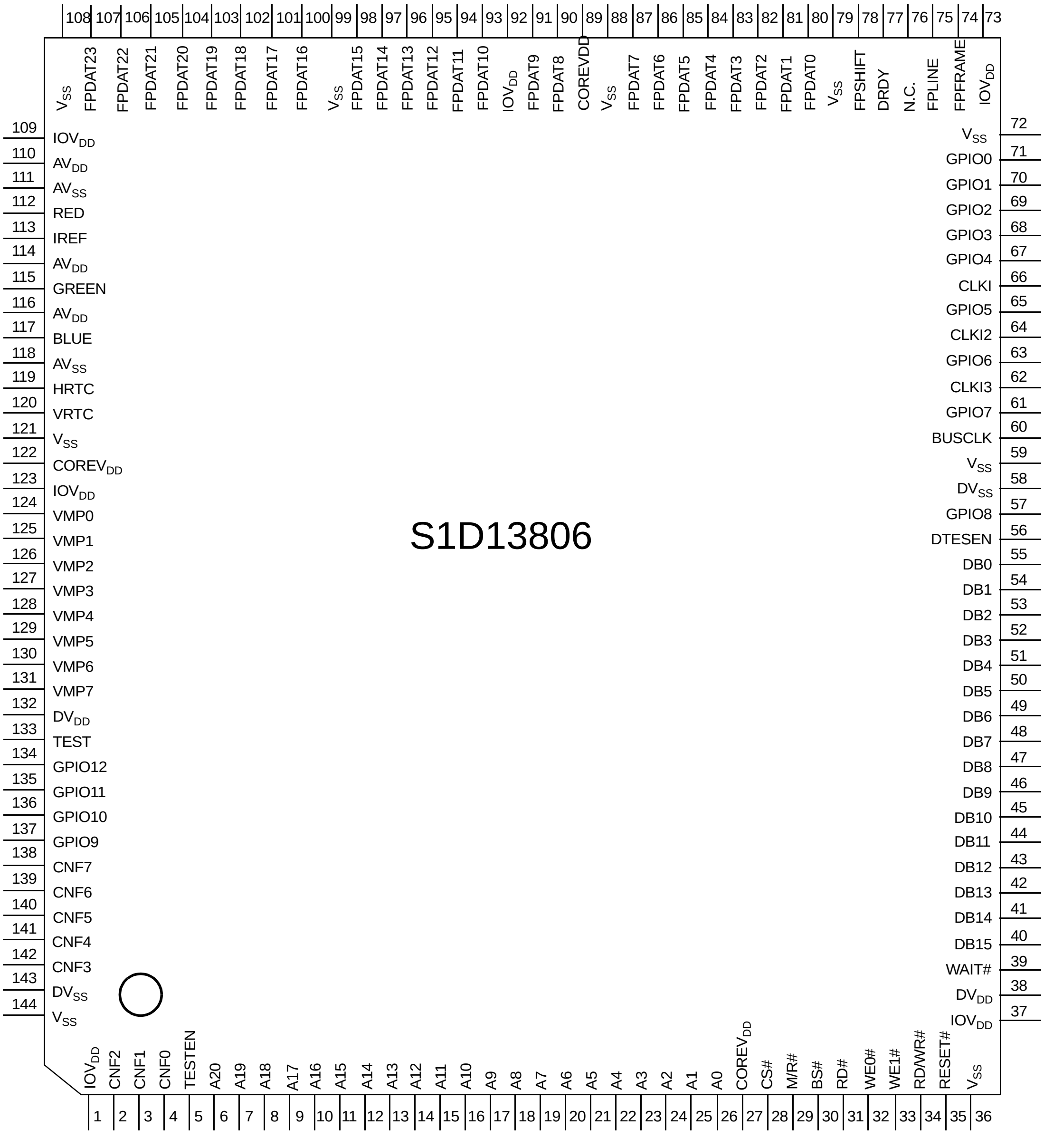


Figure 4-1 Pinout Diagram 144-pin QFP20 surface mount package

4.2 Pin Description

Key:

Hi-Z	=	High impedance
I	=	Input
O	=	Output
IO	=	Bi-Directional (Input/Output)
A	=	Analog
P	=	Power pin
C	=	CMOS level input
CD	=	CMOS level input with pull down resistor (typical value of 50KΩ at 3.3V)
CS	=	CMOS level Schmitt input
COx	=	CMOS output driver, x denotes driver type (1 = 2/-2mA, 2 = 6/-6mA @ 3.3V)
TSx	=	Tri-state CMOS output driver, x denotes driver type (1 = 2/-2mA, 2 = 6/-6mA @ 3.3V)
TSxU	=	Tri-state CMOS output driver with pull up resistor (typical value of 100KΩ at 3.3V), x denotes driver type (1 = 2/-2mA, 2 = 6/-6mA @ 3.3V)
TSxD	=	Tri-state CMOS output driver with pull down resistor (typical value of 100KΩ at 3.3V), x denotes driver type (1 = 2/-2mA, 2 = 6/-6mA @ 3.3V)

4.2.1 Host Interface

Table 4-1 Host Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
AB0	I	26	CS	Hi-Z	• For SH-3/SH-4 Bus, this pin inputs system address bit 0 (A0). • For MC68K Bus 1, this pin inputs the lower data strobe (LDS#). • For MC68K Bus 2, this pin inputs system address bit 0 (A0). • For Generic Bus, this pin inputs system address bit 0 (A0). • For MIPS/ISA Bus, this pin inputs system address bit 0 (SA0). • For Philips PR31500/31700 Bus, this pin inputs system address bit 0 (A0). • For Toshiba TX39xx Bus, this pin inputs system address bit 0 (A0). • For PowerPC Bus, this pin inputs system address bit 31 (A31). • For PC Card (PCMCIA) Bus, this pin inputs system address bit 0 (A0). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
AB[12:1]	I	14-25	C	Hi-Z	• For PowerPC Bus, these pins input the system address bits 19 through 30 (A[19:30]). • For all other busses, these pins input the system address bits 12 through 1 (A[12:1]). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
AB[16:13]	I	10-13	C	Hi-Z	• For Philips PR31500/31700 Bus, these pins are connected to V_{DD}. • For Toshiba TX39xx Bus, these pins are connected to V_{DD}. • For PowerPC Bus, these pins input the system address bits 15 through 18 (A[15:18]). • For all other busses, these pins input the system address bits 16 through 13 (A[16:13]). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
AB17	I	9	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin inputs the IO write command (/CARDIOWR). • For Toshiba TX39xx Bus, this pin inputs the IO write command (CARDIOWR*). • For PowerPC Bus, this pin inputs the system address bit 14 (A14). • For all other busses, this pin inputs the system address bit 17 (A17). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
AB18	I	8	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin inputs the IO read command (/CARDIORD). • For Toshiba TX39xx Bus, this pin inputs the IO read command (CARDIORD*). • For PowerPC Bus, this pin inputs the system address bit 13 (A13). • For all other busses, this pin inputs the system address bit 18 (A18). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
AB19	I	7	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin inputs the card control register access (/CARDREG). • For Toshiba TX39xx Bus, this pin inputs the card control register access (CARDREG*). • For PowerPC Bus, this pin inputs the system address bit 12 (A12). • For all other busses, this pin inputs the system address bit 19 (A19). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
AB20	I	6	C	Hi-Z	• For the MIPS/ISA Bus, this pin inputs system address bit 20. Note that for the ISA Bus, the unlatched LA20 must first be latched before input to AB20. • For Philips PR31500/31700 Bus, this pin inputs the address latch enable (ALE). • For Toshiba TX39xx Bus, this pin inputs the address latch enable (ALE). • For PowerPC Bus, this pin inputs the system address bit 11 (A11). • For all other busses, this pin inputs the system address bit 20 (A20). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.

Table 4-1 Host Interface Pin Descriptions (Continued)

Pin Name	Type	Pin #	Cell	RESET# State	Description
DB[15:0]	IO	40-55	C/TS2	Hi-Z	These pins are the system data bus. • For SH-3/SH-4 Bus, these pins are connected to D[15:0]. • For MC68K Bus 1, these pins are connected to D[15:0]. • For MC68K Bus 2, these pins are connected to D[31:16] for 32-bit devices (e.g. MC68030) or D[15:0] for 16-bit devices (e.g. MC68340). • For Generic Bus, these pins are connected to D[15:0]. • For MIPS/ISA Bus, these pins are connected to SD[15:0]. • For Philips PR31500/31700 Bus, pins DB[15:8] are connected to D[23:16] and pins DB[7:0] are connected to D[31:24]. • For Toshiba TX39xx Bus, pins DB[15:8] are connected to D[23:16] and pins DB[7:0] are connected to D[31:24]. • For PowerPC Bus, these pins are connected to D[0:15]. • For PC Card (PCMCIA) Bus, these pins are connected to D[15:0]. See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
WE1#	IO	33	CS/ TS2	Hi-Z	This is a multi-purpose pin: • For SH-3/SH-4 Bus, this pin inputs the write enable signal for the upper data byte (WE1#). • For MC68K Bus 1, this pin inputs the upper data strobe (UDS#). • For MC68K Bus 2, this pin inputs the data strobe (DS#). • For Generic Bus, this pin inputs the write enable signal for the upper data byte (WE1#). • For MIPS/ISA Bus, this pin inputs the system byte high enable signal (SBHE#). • For Philips PR31500/31700 Bus, this pin inputs the odd byte access enable signal (/CARDxCSH). • For Toshiba TX39xx Bus, this pin inputs the odd byte access enable signal (CARDxCSH*). • For PowerPC Bus, this pin outputs the burst inhibit signal (BI#). • For PC Card (PCMCIA) Bus, this pin inputs the card enable 2 signal (-CE2). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
M/R#	I	29	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin is connected to V_{DD}. • For Toshiba TX39xx Bus, this pin is connected to V_{DD}. • For all other busses, this input pin is used to select between the display buffer and register address spaces of the S1D13806. M/R# is set high to access the display buffer and low to access the registers. See *Register Mapping*. See Table 4-9, “CPU Interface Pin Mapping,” on page 18.
CS#	I	28	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin is connected to V_{DD}. • For Toshiba TX39xx Bus, this pin is connected to V_{DD}. • For all other busses, this is the Chip Select input. See Table 4-9, “CPU Interface Pin Mapping,” on page 18. See the respective AC Timing diagram for detailed functionality.
BUSCLK	I	60	C	Hi-Z	This pin inputs the system bus clock. It is possible to apply a 2x clock and divide it by 2 internally - see CONF5 in *Summary of Configuration Options*. • For SH-3/SH-4 Bus, this pin is connected to CKIO. • For MC68K Bus 1, this pin is connected to CLK. • For MC68K Bus 2, this pin is connected to CLK. • For Generic Bus, this pin is connected to BCLK. • For MIPS/ISA Bus, this pin is connected to CLK. • For Philips PR31500/31700 Bus, this pin is connected to DCLKOUT. • For Toshiba TX39xx Bus, this pin is connected to DCLKOUT. • For PowerPC Bus, this pin is connected to CLKOUT. • For PC Card (PCMCIA) Bus, this pin is connected to the input clock (CLKI, pin 69). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.

Table 4-1 Host Interface Pin Descriptions (Continued)

Pin Name	Type	Pin #	Cell	RESET# State	Description
BS#	I	30	CS	Hi-Z	This is a multi-purpose pin: • For SH-3/SH-4 Bus, this pin inputs the bus start signal (BS#). • For MC68K Bus 1, this pin inputs the address strobe (AS#). • For MC68K Bus 2, this pin inputs the address strobe (AS#). • For Generic Bus, this pin is connected to V_{DD}. • For MIPS/ISA Bus, this pin is connected to V_{DD}. • For Philips PR31500/31700 Bus, this pin is connected to V_{DD}. • For Toshiba TX39xx Bus, this pin is connected to V_{DD}. • For PowerPC Bus, this pin inputs the Transfer Start signal (TS#). • For PC Card (PCMCIA) Bus, this pin is connected to V_{DD}. See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
RD/WR#	I	34	CS	Hi-Z	This is a multi-purpose pin: • For SH-3/SH-4 Bus, this pin inputs the read write signal (RD/WR#). The S1D13806 needs this signal for early decode of the bus cycle. • For MC68K Bus 1, this pin inputs the read write signal (R/W#). • For MC68K Bus 2, this pin inputs the read write signal (R/W#). • For Generic Bus, this pin inputs the read command for the upper data byte (RD1#). • For MIPS/ISA Bus, this pin is connected to V_{DD}. • For Philips PR31500/31700 Bus, this pin inputs the even byte access enable signal (/CARDxCSL). • For Toshiba TX39xx Bus, this pin inputs the even byte access enable signal (CARDxCSL*). • For PowerPC Bus, this pin inputs the read write signal (RD/WR#). • For PC Card (PCMCIA) Bus, this pin inputs the card enable 1 signal (-CE1). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
RD#	I	31	CS	Hi-Z	This is a multi-purpose pin: • For SH-3/SH-4 Bus, this pin inputs the read signal (RD#). • For MC68K Bus 1, this pin is connected to V_{DD}. • For MC68K Bus 2, this pin inputs the bus size bit 1 (SIZ1). • For Generic Bus, this pin inputs the read command for the lower data byte (RD0#). • For MIPS/ISA Bus, this pin inputs the memory read signal (MEMR#). • For Philips PR31500/31700 Bus, this pin inputs the memory read command (/RD). • For Toshiba TX39xx Bus, this pin inputs the memory read command (RD*). • For PowerPC Bus, this pin inputs the transfer size 0 signal (TSIZ0). • For PC Card (PCMCIA) Bus, this pin inputs the output enable signal (-OE). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.

Table 4-1 Host Interface Pin Descriptions (Continued)

Pin Name	Type	Pin #	Cell	RESET# State	Description
WE0#	I	32	CS	Hi-Z	This is a multi-purpose pin: • For SH-3/SH-4 Bus, this pin inputs the write enable signal for the lower data byte (WE0#). • For MC68K Bus 1, this pin must be connected to V_{DD} • For MC68K Bus 2, this pin inputs the bus size bit 0 (SIZ0). • For Generic Bus, this pin inputs the write enable signal for the lower data byte (WE0#). • For MIPS/ISA Bus, this pin inputs the memory write signal (MEMW#). • For Philips PR31500/31700 Bus, this pin inputs the memory write command (/WE). • For Toshiba TX39xx Bus, this pin inputs the memory write command (WE*). • For PowerPC Bus, this pin inputs the Transfer Size 1 signal (TSIZ1). • For PC Card (PCMCIA) Bus, this pin inputs the write enable signal (-WE). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.
RESET#	I	35	CS	0	Active low input that clears all internal registers and forces all outputs to their inactive states. Note that active high RESET signals must be inverted before input to this pin. • For Toshiba TX39xx Bus, this pin is called NOP*.
WAIT#	IO	39	C/TS2	Hi-Z	The active polarity of the WAIT# output is configurable; the state of CONF[3:0] on the rising edge of RESET# defines the active polarity of WAIT# for some busses - see *“Summary of Configuration Options”*. • For SH-3 Bus, this pin outputs the wait request signal (WAIT#). • For SH-4 Bus, this pin outputs the ready signal (RDY#). • For MC68K Bus 1, this pin outputs the data transfer acknowledge signal (DTACK#). • For MC68K Bus 2, this pin outputs the data transfer and size acknowledge bit 1 (DSACK1#). • For Generic Bus, this pin outputs the wait signal (WAIT#). • For MIPS/ISA Bus, this pin outputs the IO channel ready signal (IOCHRDY). • For Philips PR31500/31700 Bus, this pin outputs the wait state signal (/CARDx-WAIT). • For Toshiba TX39xx Bus, this pin outputs the wait state signal (CARDxWAIT*). • For PowerPC Bus, this pin outputs the transfer acknowledge signal (TA#). • For PC Card (PCMCIA) Bus, this pin outputs the wait signal (-WAIT). See Table 4-9, “CPU Interface Pin Mapping,” on page 18 for summary. See the respective AC Timing diagram for detailed functionality.

Note: When WAIT# is always driven, WAIT# is in its inactive state at RESET#. CONF[3:0] determines whether WAIT# is active high or low.

4.2.2 LCD Interface

Table 4-2 LCD Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
FPDAT[23:0]	O	107-100, 91-90, 98-93, 87-80	CO2	0	Panel data bus. Not all pins are used for some panels - see Table 4-9, “CPU Interface Pin Mapping,” on page 18 for details. Unused pins are driven low.
FPFRAME	O	74	CO2	0	Frame pulse
FPLINE	O	75	CO2	0	Line pulse
FPSHIFT	O	78	CO2	0	Shift clock
DRDY	O	77	CO2	0	This is a multi-purpose pin: • For TFT/D-TFD panels this is the display enable output (DRDY). • For passive LCD with Format 1 interface this is the 2nd Shift Clock (FPSHIFT2). • For all other LCD panels this is the LCD backplane bias signal (MOD). See Table 4-10, “LCD Interface Pin Mapping,” on page 19 and REG[030h] for details.

4.2.3 MediaPlug Interface

Table 4-3 MediaPlug Pin Description

Pin Name	Type	Pin #	Cell	RESET# State	Description
VMP[7]	O	131	CO2	0	MediaPlug VMPLCTL pin.
VMP[6]	I	130	CD	Hi-Z	MediaPlug VMPRCTL pin. Internal pull-down resistors (typical value of 50KΩ at 3.3V respectively) pull the reset states to 0. External pull-up resistors can be used to pull the reset states to 1.
VMP[5:2]	IO	129-126	C/TS2U	0 or Hi-Z	MediaPlug VMPD[0:3] pins (note that pin order is reverse of conventional). Internal pull-up resistors (typical value of 100KΩ at 3.3V respectively) pull the reset states to 1. External pull-down resistors can be used to pull the reset states to 0.
VMP[1]	O	125	CO2	0	MediaPlug VMPCLK pin.
VMP[0]	O	124	CO2	0	MediaPlug VMPCLKN pin.

Note: The RESET# states of VMP[5:2] are 0 if VMP is enabled, otherwise Hi-Z.

Note: When the MediaPlug interface is enabled, GPIO12 is configured as the MediaPlug output pin VMPEPWR.

4.2.4 CRT Interface

Table 4-4 CRT Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
HRTC	O	119	CO2	0	Horizontal retrace signal for CRT
VRTC	O	120	CO2	0	Vertical retrace signal for CRT
RED	O	112	A	—	Analog output for CRT color Red / S-Video Luminance
GREEN	O	115	A	—	Analog output for CRT color Green / Composite Video Out
BLUE	O	117	A	—	Analog output for CRT color Blue / S-Video Chrominance
IREF	I	113	A	—	Current reference for DAC. If the DAC is not needed, this pin must be left unconnected and floating.

4.2.5 General Purpose IO

Table 4-5 General Purpose IO Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
GPIO12	IO	134	C/TS2	1 or Hi-Z	Bi-directional GPIO pin. When the MediaPlug interface is enabled, GPIO12 is configured as the MediaPlug output pin VMPEPWR.
GPIO[11:0]	IO	135-137, 57, 61, 63, 65, 67-71	C/TS2	Hi-Z	Bi-directional GPIO pin.

Note: The RESET# state of GPIO12 is 1 if MediaPlug is enabled, otherwise Hi-Z.

4.2.6 Configuration

Table 4-6 Configuration Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
CONF[7:0]	I	138-142, 2-4	C	Hi-Z	Input Configuration pin. State of pins are latched at RESET# to configure S1D13806 — Table 4.3, “Summary of Configuration Options,” on page 17 for details.

4.2.7 Miscellaneous

Table 4-7 Miscellaneous Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
CLKI	I	66	C	Hi-Z	Input clock for the internal pixel clock (PCLK), memory clock (MCLK), and MediaPlug clock.
CLKI2	I	64	C	Hi-Z	Input clock for the internal pixel clock (PCLK) and MediaPlug clock.
CLKI3	I	62	C	Hi-Z	Input clock for memory clock (MCLK) (Possible to use for PCLK and MediaPlug clock.)
TESTEN	I	5	CD	Hi-Z	Test Enable. This pin should be connected to V_{SS} for normal operation.
DTESEN	I	56	C	Hi-Z	Test Enable for embedded SDRAM. This pin should be connected to V_{SS} for normal operation.
TEST	—	133	—	—	Test Pin. **This pin must not be connected.**
IOVDD	P	1, 37, 73, 92, 109, 123	P	—	V_{DD} for IO (IO V_{DD})
COREVDD	P	27, 89, 122	P	—	V_{DD} for core (Core V_{DD})
AVDD	P	110, 114, 116	P	—	V_{DD} for DAC (DAC V_{DD})
DVDD	P	38, 132	P	—	V_{DD} for embedded SDRAM (SDRAM V_{DD})
VSS	P	36, 59, 72, 79, 88, 99, 108, 121, 144	P	—	V_{SS}
AVSS	P	111, 118	P	—	V_{SS} for DAC (DAC V_{SS})
DVSS	P	58, 143	P	—	V_{SS} for embedded SDRAM (SDRAM V_{SS})
N.C.	—	76	—	—	**This pin must not be connected.**

4.3 Summary of Configuration Options

Table 4-8 Summary of Power-On/Reset Options

<table>
<tr><th>Pin Name</th><th colspan="2">state of this pin at rising edge of RESET# is used to configure:(1/0)</th></tr>
<tr><th></th><th>1</th><th>0</th></tr>
<tr><td>CONF[3:0]</td><td colspan="2">Select host bus interface as follows:
<table>
<tr><th>CONF3</th><th>CONF2</th><th>CONF1</th><th>CONF0</th><th>Host Bus</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Generic; Little Endian; Active Low WAIT#</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Generic; Little Endian; Active High WAIT#</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Generic; Big Endian; Active Low WAIT#</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>Generic; Big Endian; Active High WAIT#</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>MIPS/ISA; Little Endian; Active Low WAIT#</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>MIPS/ISA; Little Endian; Active High WAIT#</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>MC68000; Big Endian; Active High WAIT#</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>MC68030; Big Endian; Active High WAIT#</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>PR31500/31700/TX39xx; Little Endian; Active Low WAIT#</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>PC Card; Little Endian; Active Low WAIT#</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>MPC821; Big Endian; Active High WAIT#</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>SH3; Little Endian; Active Low WAIT#</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>SH4; Little Endian; Active High WAIT#</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>SH3; Big Endian; Active Low WAIT#</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>SH4; Big Endian; Active High WAIT#</td></tr>
</table></td></tr>
<tr><td>CONF4</td><td colspan="2">Reserved.</td></tr>
<tr><td>CONF5</td><td>BUSCLK input divided by 2</td><td>BUSCLK input not divided</td></tr>
<tr><td>CONF6</td><td>WAIT# is always driven</td><td>WAIT# is tristated when the chip is not accessed by the host</td></tr>
<tr><td>CONF7</td><td>Configures GPIO12 as MediaPlug output pin VMPEPWR and enables MediaPlug functionality.</td><td>Configure GPIO12 for normal use and disables MediaPlug functionality.</td></tr>
</table>

4.4 Multiple Function Pin Mapping

Table 4-9 CPU Interface Pin Mapping

S1D13806 Pin Names	Generic	Hitachi SH-4/ SH-3	MIPS/ISA	Motorola MC68K Bus 1	Motorola MC68K Bus 2	Motorola PowerPC	PC Card	Philips PR31500 /PR31700	Toshiba TX39xx
AB20	A20	A20	LatchA20	A20	A20	A11	A20	ALE	ALE
AB19	A19	A19	SA19	A19	A19	A12	A19	/CARDREG	CARDREG*
AB18	A18	A18	SA18	A18	A18	A13	A18	/CARDIORD	CARDIORD*
AB17	A17	A17	SA17	A17	A17	A14	A17	/CARDIOWR	CARDIOWR*
AB[16:13]	A[16:13]	A[16:13]	SA[16:13]	A[16:13]	A[16:13]	A[15:18]	A[16:13]	Connected to V_{DD}	
AB[12:1]	A[12:1]	A[12:1]	SA[12:1]	A[12:1]	A[12:1]	A[19:30]	A[12:1]	A[12:1]	A[12:1]
AB0	A0	A0	SA0	LDS#	A0	A31	A0	A0	A0
DB[15:8]	D[15:0]	D[15:8]	SD[15:0]	D[15:8]	D[31:24]	D[0:7]	D[15:0]	D[23:16]	D[23:16]
DB[7:0]	D[7:0]	D[7:0]	SD[7:0]	D[7:0]	D[23:16]	D[8:15]	D[7:0]	D[31:24]	D[31:24]
WE1#	WE1#	WE1#	SBHE#	UDS#	DS#	BI#	-CE2	/CARDxCSH	CARDxCSH*
M/R#	External Decode							Connected to V_{DD}	
CS#	External Decode							Connected to V_{DD}	
BUSCLK	BCLK	CKIO	CLK	CLK	CLK	CLKOUT	CLKI	DCLKOUT	DCLKOUT
BS#	Connected to V_{DD}	BS#	Connected to V_{DD}	AS#	AS#	TS#	Connected to V_{DD}	Connected to V_{DD}	
RD/WR#	RD1#	RD/WR#	Connected to V_{DD}	R/W#	R/W#	RD/WR#	-CE1	/CARDxCSL	CARDxCSL*
RD#	RD0#	RD#	MEMR#	Connected to V_{DD}	SIZ1	TSIZ0	-OE	/RD	RD*
WE0#	WE0#	WE0#	MEMW#	Connected to V_{DD}	SIZ0	TSIZ1	-WE	/WE	WE*
WAIT#	WAIT#	WAIT#	IOCHRDY	DTACK#	DSACK1#	TA#	-WAIT	/CARDxWAIT	CARDxWAIT*
RESET#	RESET#	RESET#	inverted RESET	RESET#	RESET#	RESET#	inverted RESET	RESET#	PON*

Note: All GPIO pins default to input on reset and unless programmed otherwise, should be connected to either V_{SS} or IO V_{DD} if not used.

Table 4-10 LCD Interface Pin Mapping

S1D13806 Pin Names	Monochrome Passive Panel			Color Passive Panel						Color Active (TFT) Panel				
	Single		Dual	Single	Single Format 1	Single Format 2	Single	Dual						
	4-bit	8-bit	8-bit	4-bit	8-bit	8-bit	16-Bit	8-bit	16-bit	9-bit	12-bit	18-bit	2x9-bit	2x12-bit
FPFRAME	FPFRAME													
FPLINE	FPLINE													
FPSHIFT	FPSHIFT													
DRDY	MOD				FPSHIFT 2	MOD				DRDY				
FPDAT0	driven 0	D0	LD0	driven 0	D0	D0	D0	LD0	LD0	R2	R3	R5	R02	R03
FPDAT1	driven 0	D1	LD1	driven 0	D1	D1	D1	LD1	LD1	R1	R2	R4	R01	R02
FPDAT2	driven 0	D2	LD2	driven 0	D2	D2	D2	LD2	LD2	R0	R1	R3	R00	R01
FPDAT3	driven 0	D3	LD3	driven 0	D3	D3	D3	LD3	LD3	G2	G3	G5	G02	G03
FPDAT4	D0	D4	UD0	D0	D4	D4	D8	UD0	UD0	G1	G2	G4	G01	G02
FPDAT5	D1	D5	UD1	D1	D5	D5	D9	UD1	UD1	G0	G1	G3	G00	G01
FPDAT6	D2	D6	UD2	D2	D6	D6	D10	UD2	UD2	B2	B3	B5	B02	B03
FPDAT7	D3	D7	UD3	D3	D7	D7	D11	UD3	UD3	B1	B2	B4	B01	B02
FPDAT8	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D4	driven 0	LD4	B0	B1	B3	B00	B01
FPDAT9	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D5	driven 0	LD5	driven 0	R0	R2	driven 0	R00
FPDAT10	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D6	driven 0	LD6	driven 0	driven 0	R1	R12	R13
FPDAT11	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D7	driven 0	LD7	driven 0	G0	G2	driven 0	G00
FPDAT12	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D12	driven 0	UD4	driven 0	driven 0	G1	G12	G13
FPDAT13	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D13	driven 0	UD5	driven 0	driven 0	G0	G11	G12
FPDAT14	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D14	driven 0	UD6	driven 0	B0	B2	driven 0	B00
FPDAT15	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D15	driven 0	UD7	driven 0	driven 0	B1	B12	B13
FPDAT16	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B0	B11	B12
FPDAT17	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	R0	R11	R12
FPDAT18	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	R10	R11
FPDAT19	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	R10
FPDAT20	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	G10	G11
FPDAT21	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	G10
FPDAT22	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B10	B11
FPDAT23	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B10

4.5 CRT/TV Interface

The following figure shows external circuitry for the CRT/TV interface.

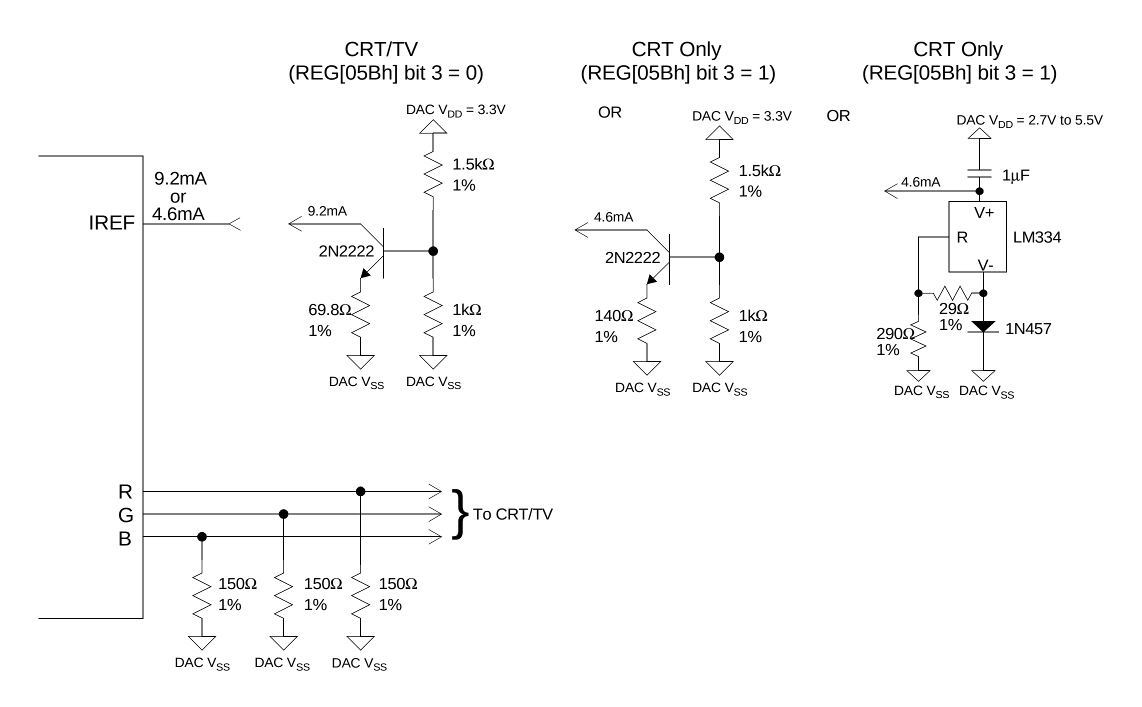


Figure 4-2 External Circuitry for CRT Interface

Note: Example implementation only, individual characteristics of components may affect actual IREF current.

5 D.C. CHARACTERISTICS

Table 5-1 Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
IO V_{DD}	Supply Voltage	V_{SS} - 0.3 to 4.0	V
Core V_{DD}	Supply Voltage	V_{SS} - 0.3 to 4.0	V
DAC V_{DD}	Supply Voltage	V_{SS} - 0.3 to 4.0	V
SDRAM V_{DD}	Supply Voltage	V_{SS} - 0.3 to 4.0	V
V_{IN}	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.5	V
V_{OUT}	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.5	V
T_{STG}	Storage Temperature	-65 to 150	°C
T_{SOL}	Solder Temperature/Time	260 for 10 sec. max at lead	°C

Table 5-2 Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
IO V_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	3.6	V
Core V_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	3.6	V
DAC V_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	3.6	V
SDRAM V_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	3.6	V
V_{IN}	Input Voltage		V_{SS}		V_{DD}	V
T_{OPR}	Operating Temperature		-40	25	85	°C

Table 5-3 Electrical Characteristics for VDD = 3.3V typical

Symbol	Parameter	Condition		Min	Typ	Max	Units
I_{DDS}	Quiescent Current	Quiescent Conditions				170	uA
I_{IZ}	Input Leakage Current	—		-1		1	µA
I_{OZ}	Output Leakage Current	—		-1		1	µA
V_{OH}	High Level Output Voltage	VDD = min I_{OH} = -2mA (Type1), -6mA (Type2)		V_{DD} - 0.3			V
V_{OL}	Low Level Output Voltage	VDD = min I_{OL} = 2mA (Type1), 6mA (Type2)				0.3	V
V_{IH}	High Level Input Voltage	LVTTL level, V_{DD} = max		2.0			V
V_{IL}	Low Level Input Voltage	LVTTL level, V_{DD} = min				0.8	V
V_{T+}	High Level Input Voltage	LVTTL Schmitt		1.1		2.4	V
V_{T-}	Low Level Input Voltage	LVTTL Schmitt		0.6		1.8	V
V_{H1}	Hysteresis Voltage	LVTTL Schmitt		0.1			V
R_{PD}	Pull-Down Resistance	$V_I = V_{DD}$	Type 1	20	50	120	kΩ
			Type 2	40	100	240	kΩ
R_{PU}	Pull-Up Resistance	V_I = 0V	Type 1	20	50	120	kΩ
			Type 2	40	100	240	kΩ
C_I	Input Pin Capacitance					10	pF
C_O	Output Pin Capacitance					10	pF
C_{IO}	Bi-Directional Pin Capacitance					10	pF

6 A.C. CHARACTERISTICS

Conditions: V_{DD} = 3.3V ± 10% (IO and Core)
T_A = -40° C to 85°C
T_{rise} and T_{fall} for all inputs must be $\leq$ 5 ns (10% ~ 90%)
C_L = 50pF (CPU Interface), unless noted
C_L = 100pF (LCD Panel Interface)
C_L = 10pF (Display Memory Interface)
C_L = 10pF (CRT Interface)

6.1 Clock Timing

6.1.1 Input Clocks

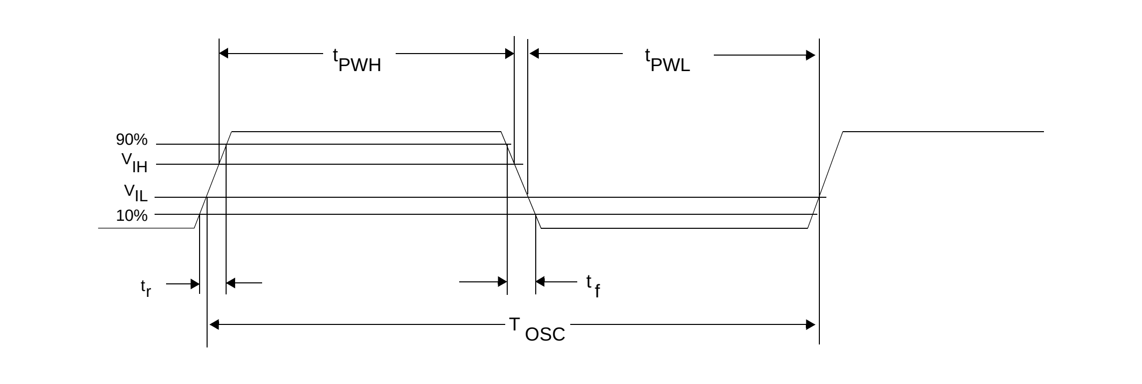


Figure 6-1 Clock Input Requirement

Table 6-1 Clock Input Requirements for CLKI, CLKI2, and CLKI3

Symbol	Parameter	Min	Max	Units
f_{OSC}	Input Clock Frequency		65	MHz
T_{OSC}	Input Clock Period	$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High	6		ns
t_{PWL}	Input Clock Pulse Width Low	6		ns
t_f	Input Clock Fall Time (10% - 90%)		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5	ns

Note: The internal memory clock (MCLK) is restricted to a maximum of 50MHz and a minimum of 5MHz.

Table 6-2 Clock Input Requirements for CLKI, CLKI2, and CLKI3 when divided down for MCLK

Symbol	Parameter	Min	Max	Units
f_{OSC}	Input Clock Frequency		80	MHz
T_{OSC}	Input Clock Period	$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High	5.6		ns
t_{PWL}	Input Clock Pulse Width Low	5.6		ns
t_f	Input Clock Fall Time (10% - 90%)		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5	ns

Table 6-3 Clock Input Requirements for CLKI, CLKI2, and CLKI3 when divided down for LCD PCLK, CRT/TV PCLK, or MediaPlug Clock

Symbol	Parameter	Min	Max	Units
f_{OSC}	Input Clock Frequency		100	MHz
T_{OSC}	Input Clock Period	$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High	4.5		ns
t_{PWL}	Input Clock Pulse Width Low	4.5		ns
t_f	Input Clock Fall Time (10% - 90%)		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5	ns

6.1.2 Internal Clocks

Table 6-4 Internal Clock Requirements

Symbol	Parameter	Min	Max	Units
f_{MCLK}	Memory Clock Frequency	0	50	MHz
$f_{LCD\ PCLK}$	LCD Pixel Clock Frequency	0	Note 1	MHz
$f_{CRT/TV\ PCLK}$	CRT/TV Pixel Clock Frequency	0	Note 2	MHz
$f_{MediaPlug\ Clock}$	MediaPlug Clock Frequency	0	20	MHz

Note: 1. The maximum LCD pixel clock for TFT panels is 65MHz.
The maximum LCD pixel clock for passive panels is 40MHz.

2. The maximum CRT pixel clock is 65MHz.
The TV pixel clock for NTSC output is fixed at 14.318MHz.
The TV pixel clock for PAL output is fixed at 17.734MHz.

Note: For further information on the internal clocks, refer to Section 7, “Clocks” on page 75.

6.2 CPU Interface Timing

6.2.1 Generic Interface Timing

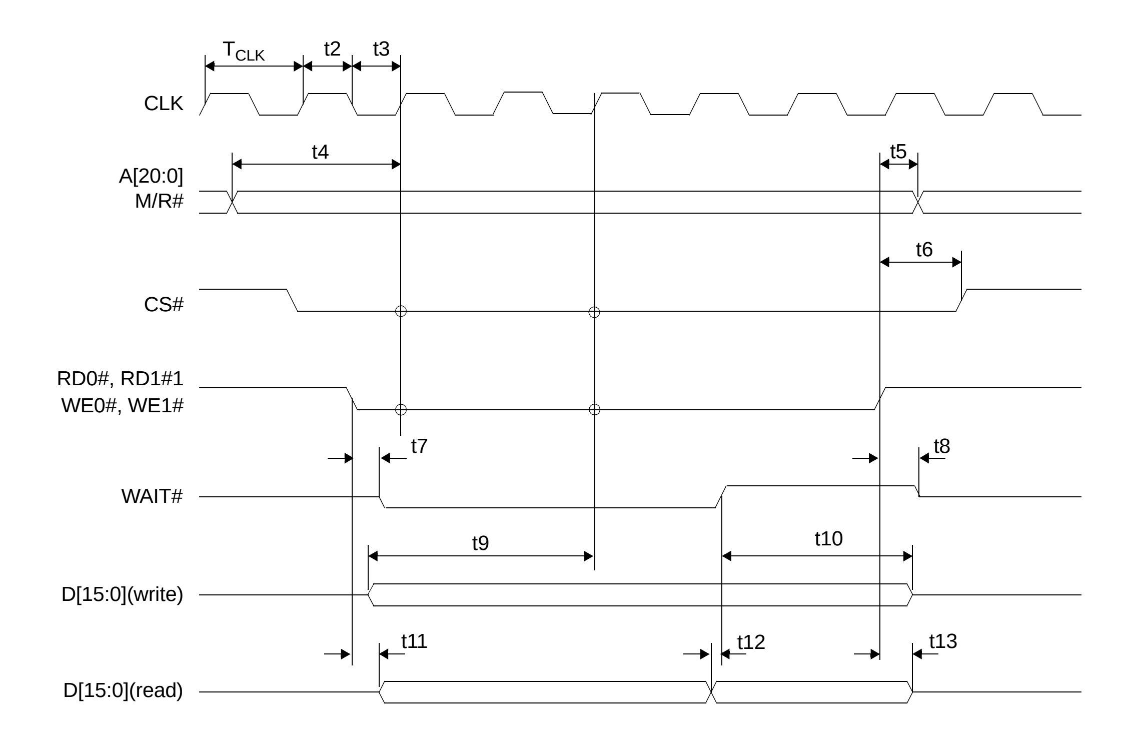


Figure 6-2 Generic Interface Timing

Note: The above timing diagram is not applicable if CONF5 = 1 (BUSCLK divided by 2).

Note: WAIT# is always driven when CONF6 = 1.

Table 6-5 Generic Interface Timing

Symbol	Parameter	3.0V Min	3.0V Max	Units
f_{CLK}	Clock Frequency		50	MHz
T_{CLK}	Clock period	$1/f_{CLK}$		ns
t2	Clock pulse width high	6		ns
t3	Clock pulse width low	6		ns
t4	A[20:0], M/R# setup to first CLK where CS# = 0 and either RD0#, RD1# = 0 or WE0#, WE1# = 0	4		ns
t5	A[20:0], M/R# hold from rising edge of either RD0#, RD1# or WE0#, WE1#	0		ns
t6	CS# hold from rising edge of either RD0#, RD1# or WE0#, WE1#	0		ns
t7	Falling edge of either RD0#, RD1# or WE0#, WE1# to WAIT# driven low	5	15	ns
t8	Rising edge of either RD0#, RD1# or WE0#, WE1# to WAIT# tri-state	4	13	ns
t9	D[15:0] setup to third CLK where CS# = 0 and WE0#, WE1# = 0 (write cycle)	0		ns
t10	D[15:0] hold (write cycle)	0		ns
t11	Falling edge RD0#, RD1# to D[15:0] driven (read cycle)	3		ns
t12	D[15:0] setup to rising edge WAIT# (read cycle)	0		ns
t13	Rising edge of RD0#, RD1# to D[15:0] tri-state (read cycle)	3	10	ns

6.2.2 Hitachi SH-4 Interface Timing

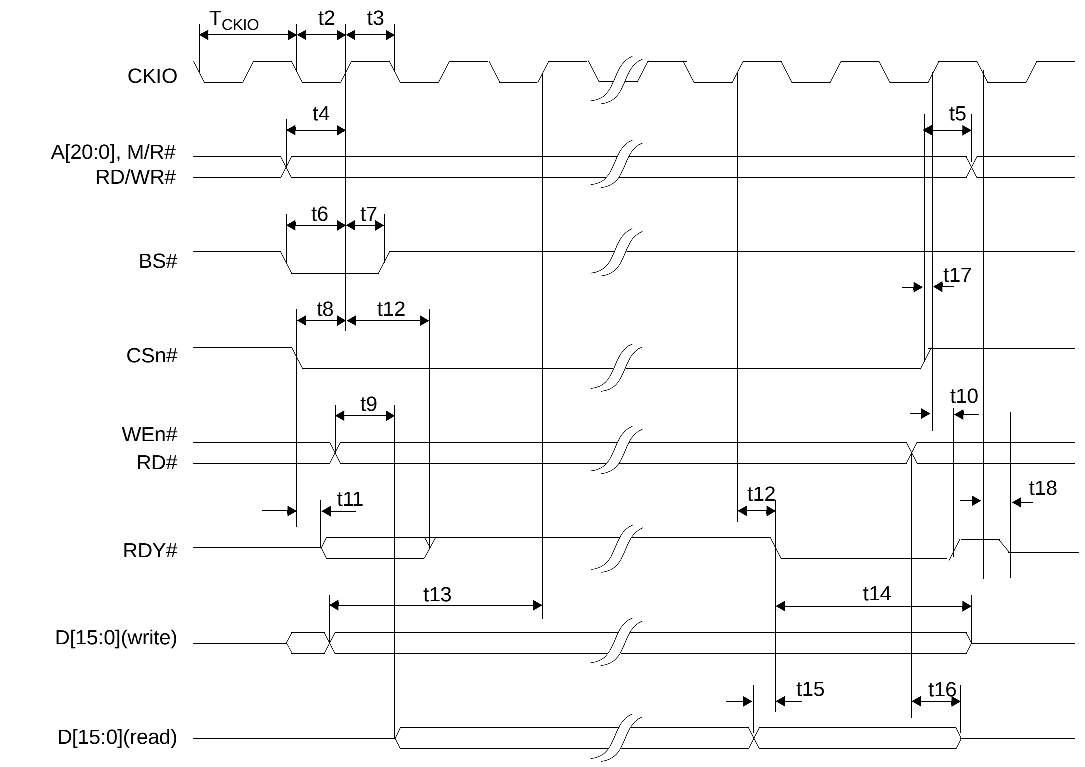


Figure 6-3 Hitachi SH-4 Interface Timing

Note: The above timing diagram is not applicable if CONF5 = 1 (BUSCLK divided by 2).

Note: The SH-4 Wait State Control Register for the area in which the S1D13806 resides must be set to a non-zero value. The SH-4 read-to-write cycle transition must be set to a non-zero value (with reference to BUSCLK).

Note: RDY# is always driven when CONF6 = 1.

Table 6-6 Hitachi SH-4 Interface Timing

		3.0V[1]		
Symbol	Parameter	Min	Max	Units
f_{CKIO}	Clock Frequency		66	MHz
T_{CKIO}	Clock period	$1/f_{CKIO}$		ns
t2	Clock pulse width high	6		ns
t3	Clock pulse width low	6		ns
t4	A[20:0], M/R#, RD/WR# setup to CKIO	4		ns
t5	A[20:0], M/R#, RD/WR# hold from CS#	0		ns
t6	BS# setup	4		ns
t7	BS# hold	3		ns
t8	CSn# setup	3		ns
t9	Falling edge RD# to DB[15:0] driven	3		ns
t10	CKIO to RDY# high	4	22	ns
t11	Falling edge CSn# to RDY# driven	3	12	ns
t12	CKIO to RDY# delay	4	13	ns
t13	DB[15:0] setup to 2nd CKIO after BS# (write cycle)	0		ns
t14	DB[15:0] hold (write cycle)	0		ns
t15	DB[15:0] valid to RDY# falling edge (read cycle)	0		ns
t16	Rising edge RD# to DB[15:0] tri-state (read cycle)	6	29	ns
t17	CSn# high setup to CKIO	3		ns
t18	Falling edge CKIO to RDY# tri-state	3	15	ns

Note: 1. Two software WAIT states are required.

6.2.3 Hitachi SH-3 Interface Timing

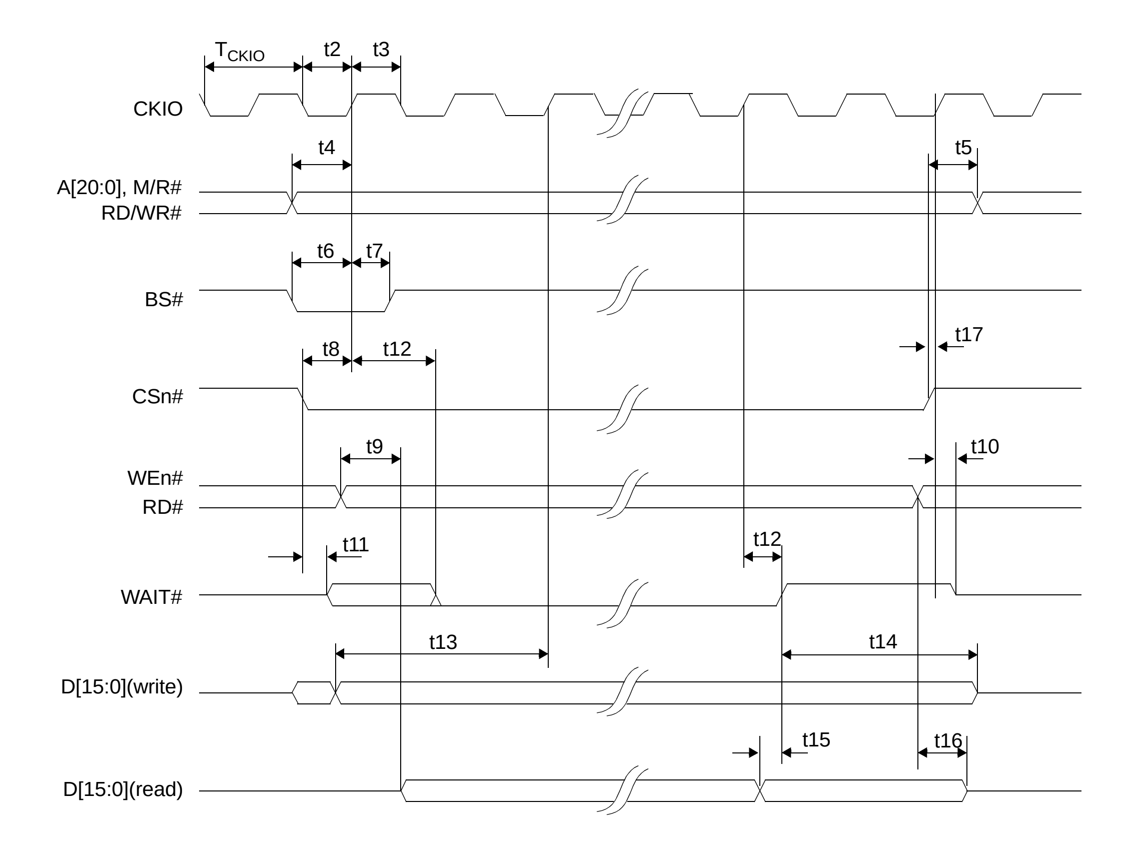


Figure 6-4 Hitachi SH-3 Interface Timing

Note: The above timing diagram is not applicable if CONF5 = 1 (BUSCLK divided by 2).

Note: The SH-3 Wait State Control Register for the area in which the S1D13806 resides must be set to a non-zero value.

Note: WAIT# is always driven when CONF6 = 1.

Table 6-7 Hitachi SH-3 Interface Timing

		3.0V[1]		
Symbol	Parameter	Min	Max	Units
f_{CKIO}	Clock Frequency		66	MHz
T_{CKIO}	Clock period	$1/f_{CKIO}$		ns
t2	Clock pulse width high	6		ns
t3	Clock pulse width low	6		ns
t4	A[20:0], M/R#, RD/WR# setup to CKIO	4		ns
t5	A[20:0], M/R#, RD/WR# hold from CS#	0		ns
t6	BS# setup	4		ns
t7	BS# hold	3		ns
t8	CSn# setup	3		ns
t9	Falling edge RD# to DB[15:0] driven	3		ns
t10	Rising edge CSn# to WAIT# tri-state	4	17	ns
t11	Falling edge CSn# to WAIT# driven	3	16	ns
t12	CKIO to WAIT# delay	4	21	ns
t13	DB[15:0] setup to 2nd CKIO after BS# (write cycle)	0		ns
t14	DB[15:0] hold (write cycle)	0		ns
t15	DB[15:0] valid to WAIT# rising edge (read cycle)	0		ns
t16	Rising edge RD# to DB[15:0] tri-state (read cycle)	6	29	ns
t17	CSn# high setup to CKIO	3		ns

Note: 1. Two software WAIT states are required when f_{CKIO} is greater than 33MHz.

6.2.4 MIPS/ISA Interface Timing (e.g. NEC VR41xx)

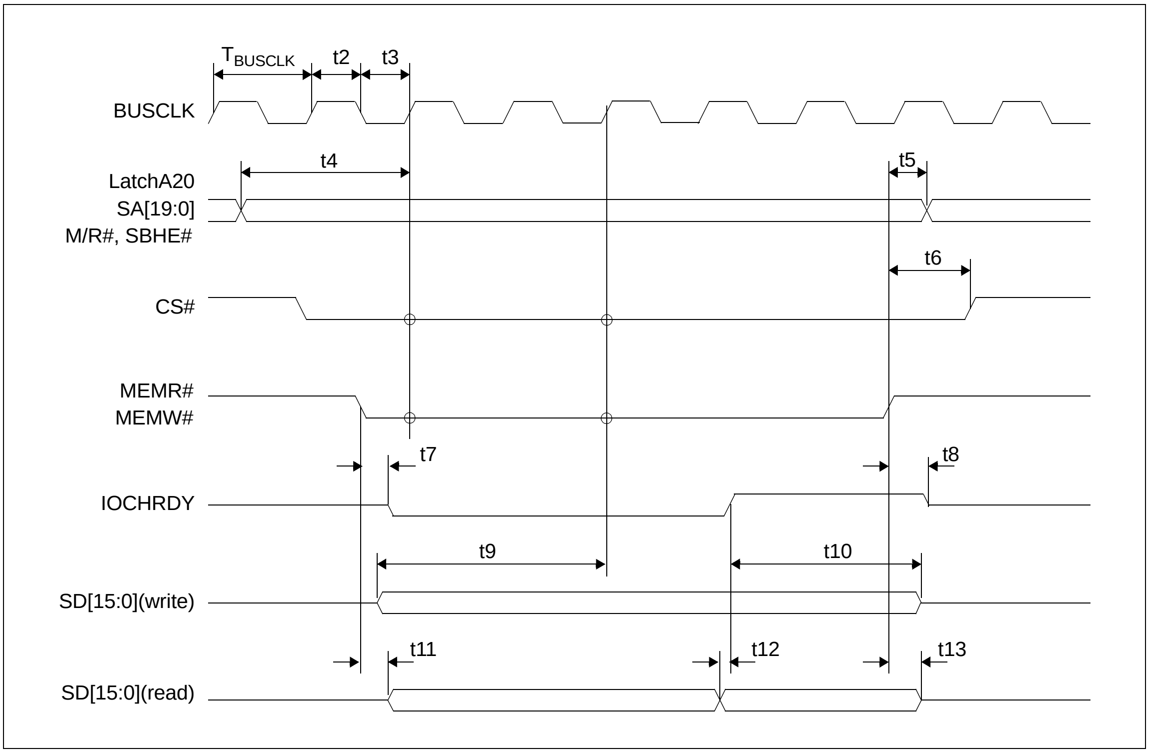


Figure 6-5 MIPS/ISA Interface Timing

Note: The above timing diagram is not applicable if CONF5 = 1 (BUSCLK divided by 2).

Note: IOCHRDY is always driven when CONF6 = 1.

Table 6-8 MIPS/ISA Interface Timing

		3.0V		
Symbol	Parameter	Min	Max	Units
f_{BUSCLK}	Clock Frequency		50	MHz
T_{BUSCLK}	Clock period	$1/f_{BUSCLK}$		ns
t2	Clock pulse width high	6		ns
t3	Clock pulse width low	6		ns
t4	LatchA20, SA[19:0], M/R#, SBHE# setup to first BUSCLK where CS# = 0 and either MEMR# = 0 or MEMW# = 0	4		ns
t5	LatchA20, SA[19:0], M/R#, SBHE# hold from rising edge of either MEMR# or MEMW#	0		ns
t6	CS# hold from rising edge of either MEMR# or MEMW#	0		ns
t7	Falling edge of either MEMR# or MEMW# to IOCHRDY# driven low	3		ns
t8	Rising edge of either MEMR# or MEMW# to IOCHRDY# tri-state	2	11	ns
t9	SD[15:0] setup to third BUSCLK where CS# = 0 MEMW# = 0 (write cycle)	0		ns
t10	SD[15:0] hold (write cycle)	0		ns
t11	Falling edge MEMR# toSD[15:0] driven (read cycle)	4		ns
t12	SD[15:0] setup to rising edge IOCHRDY# (read cycle)	0		ns
t13	Rising edge of MEMR# toSD[15:0] tri-state (read cycle)	6	29	ns

6.2.5 Motorola MC68K Bus 1 Interface Timing (e.g. MC68000)

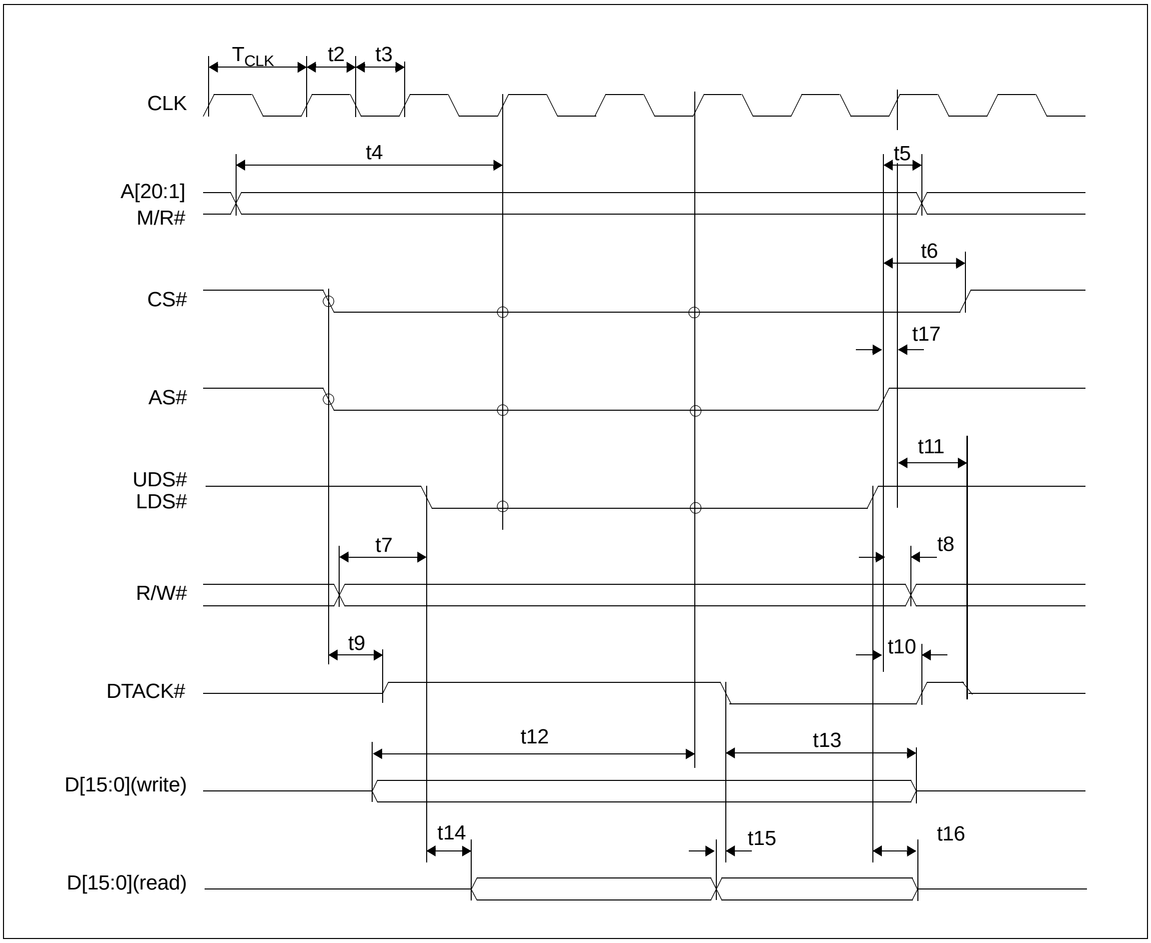


Figure 6-6 Motorola MC68K Bus 1 Interface Timing

Note: The above timing diagram is not applicable if CONF5 = 1 (BUSCLK divided by 2).

Note: DTACK# is always driven when CONF6 = 1.

Table 6-9 Motorola MC68K Bus 1 Interface Timing

Symbol	Parameter	3.0V Min	3.0V Max	Units
f_{CLK}	Clock Frequency		50	MHz
T_{CLK}	Clock period	$1/f_{CLK}$		ns
t2	Clock pulse width high	6		ns
t3	Clock pulse width low	6		ns
t4	A[20:1], M/R# setup to first CLK where CS# = 0 AS# = 0, and either UDS# = 0 or LDS# = 0	5		ns
t5	A[20:1], M/R# hold from AS#	0		ns
t6	CS# hold from AS#	0		ns
t7	R/W# setup to before to either UDS# = 0 or LDS# = 0	10		ns
t8	R/W# hold from AS#	0		ns
t9	AS# = 0 and CS# = 0 to DTACK# driven high	1		ns
t10	AS# high to DTACK# high	4	19	ns
t11	First BCLK where AS# = 1 to DTACK# high impedance		16	ns
t12	D[15:0] valid to third CLK where CS# = 0 AS# = 0, and either UDS# = 0 or LDS# = 0 (write cycle)	0		ns
t13	D[15:0] hold from falling edge of DTACK# (write cycle)	0		ns
t14	Falling edge of UDS# = 0 or LDS# = 0 to DB driven (read cycle)	3		ns
t15	D[15:0] valid to DTACK# falling edge (read cycle)	0		ns
t16	UDS# and LDS# high to D[15:0] invalid/high impedance (read cycle)	6	30	ns
t17	AS# high setup to CLK	4		ns

6.2.6 Motorola MC68K Bus 2 Interface Timing (e.g. MC68030)

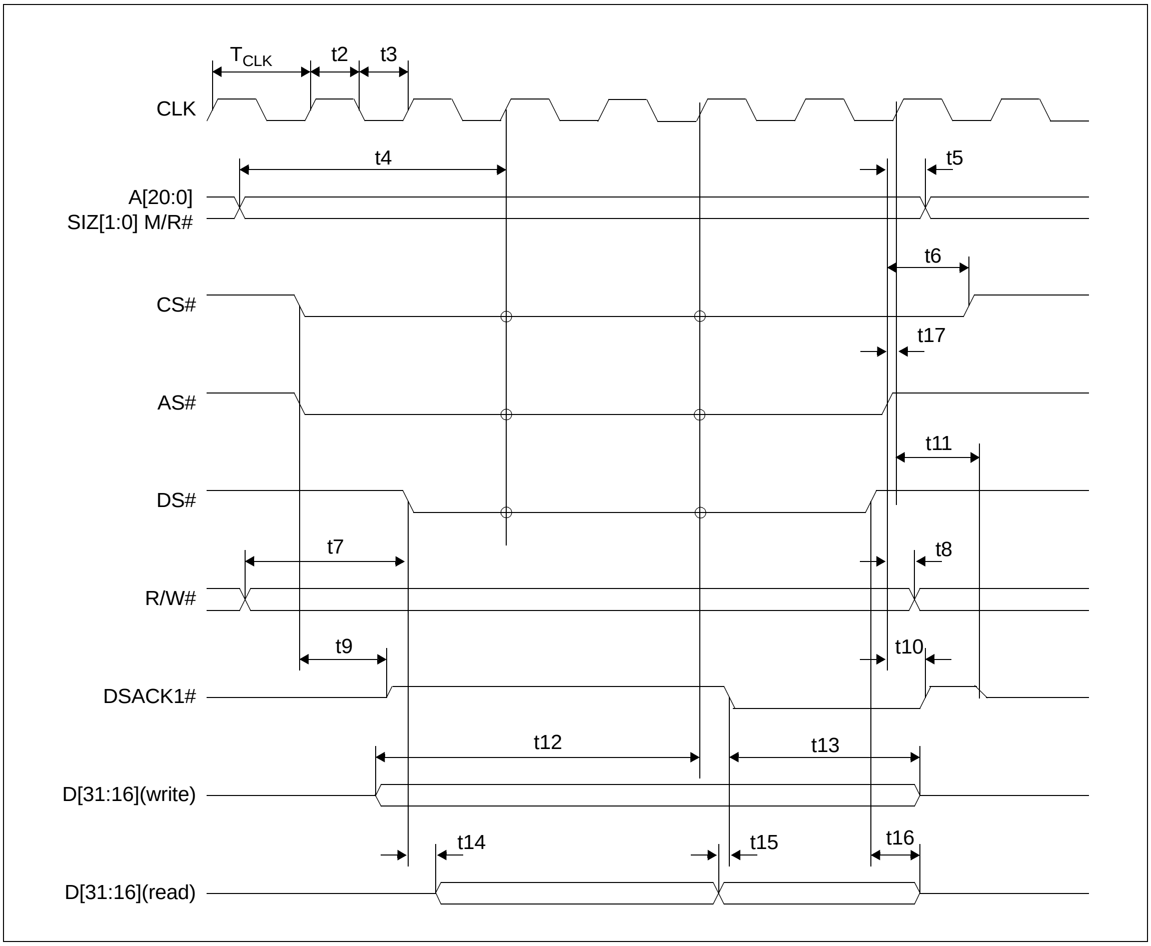


Figure 6-7 Motorola MC68K Bus 2 Interface Timing

Note: The above timing diagram is not applicable if CONF5 = 1 (BUSCLK divided by 2).

Note: DSACK1# is always driven when CONF6 = 1.

Table 6-10 Motorola MC68K Bus 2 Interface Timing

		3.0V		
Symbol	Parameter	Min	Max	Units
f_{CLK}	Clock Frequency		50	MHz
T_{CLK}	Clock period	$1/f_{CLK}$		ns
t2	Clock pulse width high	6		ns
t3	Clock pulse width low	6		ns
t4	A[20:0], SIZ[1:0], M/R# setup to first CLK where CS# = 0 AS# = 0, and DS# = 0	5		ns
t5	A[20:0], SIZ[1:0], M/R# hold from AS#	0		ns
t6	CS# hold from AS#	0		ns
t7	R/W# setup to DS#	10		ns
t8	R/W# hold from AS#	0		ns
t9	AS# = 0 and CS# = 0 to DSACK1# driven high	1		ns
t10	AS# high to DSACK1# high	4	19	ns
t11	First BCLK where AS# = 1 to DSACK1# high impedance	3	16	ns
t12	D[31:16] valid to third CLK where CS# = 0 AS# = 0, and DS# = 0 (write cycle)	0		ns
t13	D[31:16] hold from falling edge of DSACK1# (write cycle)	0		ns
t14	Falling edge of DS# = 0 to DB driven (read cycle)	3		ns
t15	D[31:16] valid to DSACK1# falling edge (read cycle)	0		ns
t16	DS# high to D[31:16] invalid/high impedance (read cycle)	6	30	ns
t17	AS# high setup to CLK	4		ns

6.2.7 Motorola PowerPC Interface Timing (e.g. MPC8xx, MC68040, Coldfire)

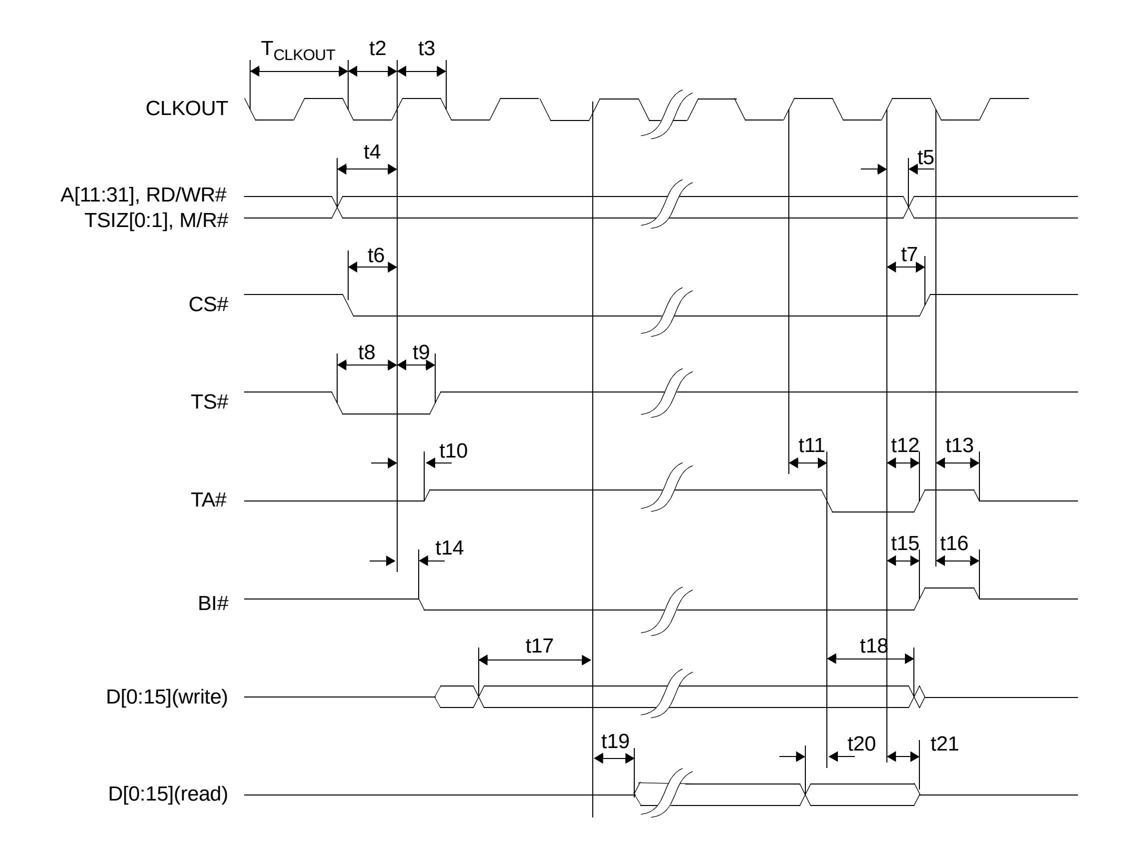


Figure 6-8 Motorola PowerPC Interface Timing

Note: The above timing diagram is not applicable if CONF5 = 1 (BUSCLK divided by 2).

Note: TA# is always driven when CONF6 = 1.

Table 6-11 Motorola PowerPC Interface Timing

		3.0V		
Symbol	Parameter	Min	Max	Units
f_{CLKOUT}	Clock Frequency		45	MHz
T_{CLKOUT}	Clock period	$1/f_{CLKOUT}$		ns
t2	Clock pulse width low	6		ns
t3	Clock pulse width high	6		ns
t4	AB[11:31], RD/WR#, TSIZ[0:1], M/R# setup	0		ns
t5	AB[11:31], RD/WR#, TSIZ[0:1], M/R# hold	0		ns
t6	CS# setup	1		ns
t7	CS# hold	1		ns
t8	TS# setup	1		ns
t9	TS# hold	1		ns
t10	CLKOUT to TA# driven	5		ns
t11	CLKOUT to TA# low	4	14	ns
t12	CLKOUT to TA# high	5	15	ns
t13	negative edge CLKOUT to TA# tri-state	3	12	ns
t14	CLKOUT to BI# driven	5	15	ns
t15	CLKOUT to BI# high	4	14	ns
t16	negative edge CLKOUT to BI# tri-state	3	9	ns
t17	DB[15:0] setup to 2nd CLKOUT after TS# = 0 (write cycle)	0		ns
t18	DB[15:0] hold (write cycle)	0		ns
t19	CLKOUT to DB driven (read cycle)	0		ns
t20	DB[15:0] valid to TA# falling edge (read cycle)	0		ns
t21	CLKOUT to DB[15:0] tri-state (read cycle)	3	11	ns

Note: Output pin loading on DB[15:0], TA#, BI# is 10pF.

6.2.8 PC Card Timing (e.g. StrongARM)

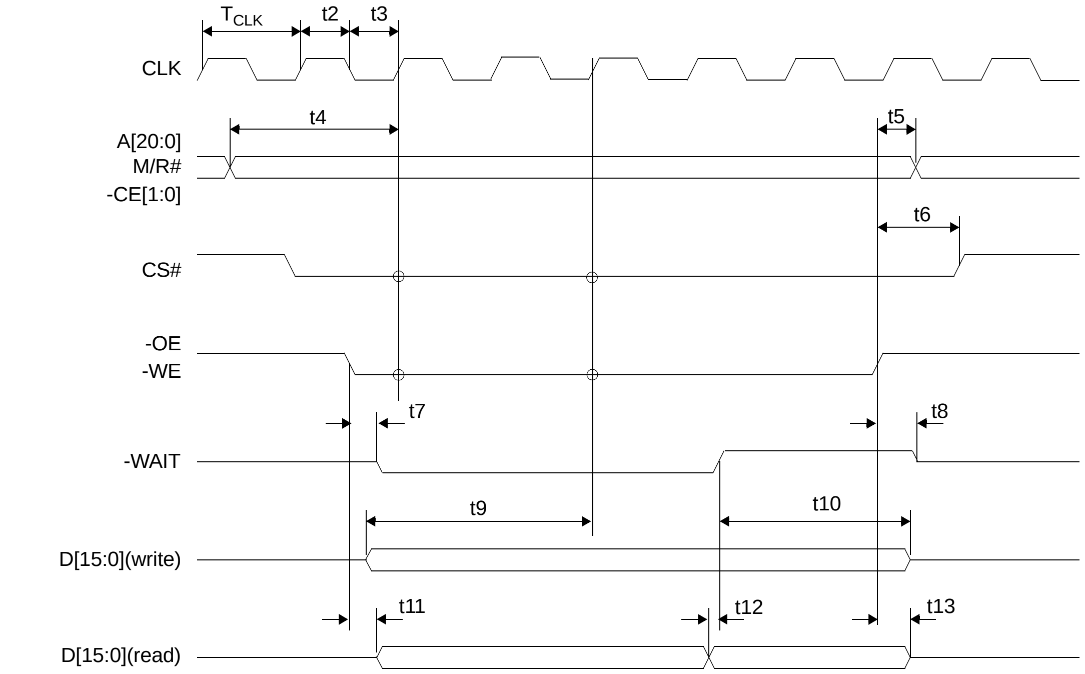


Figure 6-9 PC Card Timing

Note: The above timing diagram is not applicable if CONF5 = 1 (BUSCLK divided by 2).

Table 6-12 PC Card Timing

		3.0V		
Symbol	Parameter	Min	Max	Units
f_{CLK}	Clock frequency		50	MHz
T_{CLK}	Clock period	$1/f_{CLK}$		ns
t2	Clock pulse width high	6		ns
t3	Clock pulse width low	6		ns
t4	A[20:0], M/R# setup to first CLK where -CE = 0 and either -OE = 0 or -WE = 0	4		ns
t5	A[20:0], M/R# hold from rising edge of either -OE or -WE	0		ns
t6	-CE hold from rising edge of either -OE or -WE	0		ns
t7	Falling edge of either -OE or -WE to -WAIT driven low	5	15	ns
t8	Rising edge of either -OE or -WE to -WAIT tri-state	3	13	ns
t9	D[15:0] setup to third CLK where -CE = 0 and -WE = 0 (write cycle)	0		ns
t10	D[15:0] hold (write cycle)	0		ns
t11	Falling edge -OE to D[15:0] driven (read cycle)	9		ns
t12	D[15:0] setup to rising edge -WAIT (read cycle)	0		ns
t13	Rising edge of -OE to D[15:0] tri-state (read cycle)	3	10	ns

6.2.9 Philips Interface Timing (e.g. PR31500/PR31700)

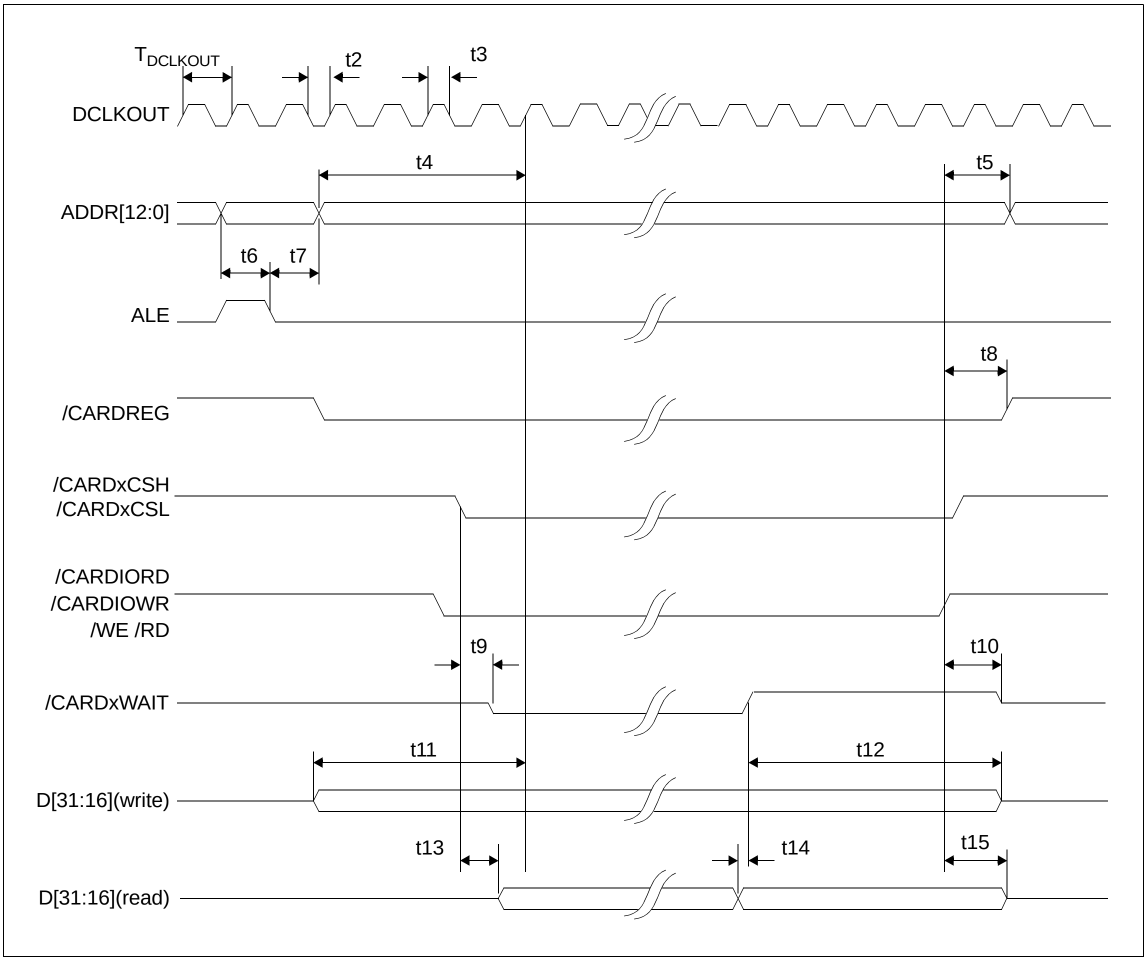


Figure 6-10 Philips Interface Timing

Note: /CARDxWAIT is always driven when CONF6 = 1.

Table 6-13 Philips Interface Timing

Symbol	Parameter	3.0V Min	3.0V Max	Units
$f_{DCLKOUT}$	Clock frequency		75	MHz
$T_{DCLKOUT}$	Clock period	$1/f_{DCLKOUT}$		ns
t2	Clock pulse width low	6		ns
t3	Clock pulse width high	6		ns
t4	ADDR[12:0] setup to first CLK of cycle	10		ns
t5	ADDR[12:0] hold from command invalid	0		ns
t6	ADDR[12:0] setup to falling edge ALE	10		ns
t7	ADDR[12:0] hold from falling edge ALE	5		ns
t8	/CARDREG hold from command invalid	0		ns
t9	Falling edge of chip select to /CARDxWAIT driven	0	15	ns
t10	Command invalid to /CARDxWAIT tri-state	5	25	ns
t11	D[31:16] valid to first CLK of cycle (write cycle)	10		ns
t12	D[31:16] hold from rising edge of /CARDxWAIT	0		
t13	Chip select to D[31:16] driven (read cycle)	1		ns
t14	D[31:16] setup to rising edge /CARDxWAIT (read cycle)	0		ns
t15	Command invalid to D[31:16] tri-state (read cycle)	5	25	ns

Note: If BUSCLK exceeds 37.5MHz, it must be divided by 2 using CONF5 (see Table 4-8, “Summary of Power-On/Reset Options,” on page 17).

6.2.10 Toshiba Interface Timing (e.g. TX39xx)

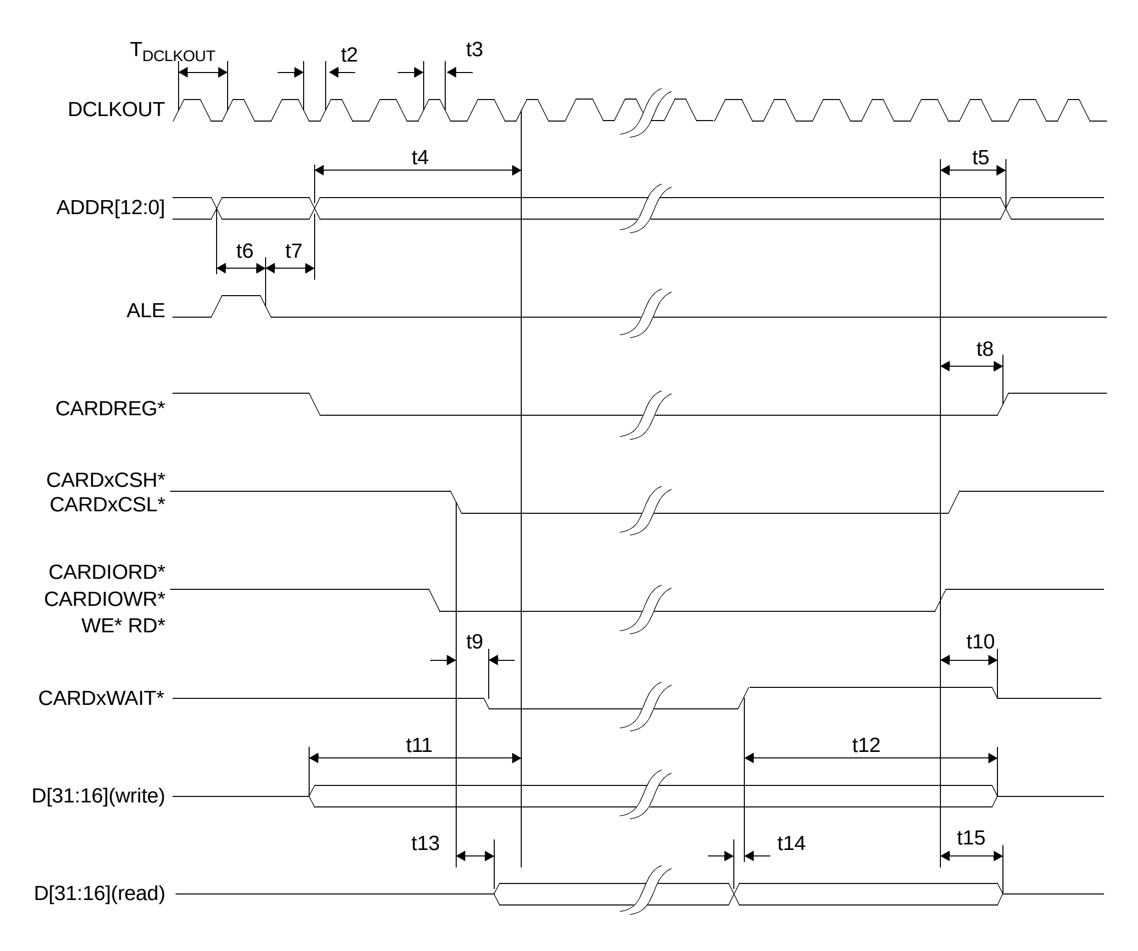


Figure 6-11 Toshiba Interface Timing

Note: CARDxWAIT* is always driven when CONF6 = 1.

Table 6-14 Toshiba Interface Timing

		3.0V		
Symbol	Parameter	Min	Max	Units
$f_{DCLKOUT}$	Clock frequency		75	MHz
$T_{DCLKOUT}$	Clock period	$1/f_{DCLKOUT}$		ns
t2	Clock pulse width low	6		ns
t3	Clock pulse width high	6		ns
t4	ADDR[12:0] setup to first CLK of cycle	10		ns
t5	ADDR[12:0] hold from command invalid	0		ns
t6	ADDR[12:0] setup to falling edge ALE	10		ns
t7	ADDR[12:0] hold from falling edge ALE	5		ns
t8	CARDREG* hold from command invalid	0		ns
t9	Falling edge of chip select to CARDxWAIT* driven	0	15	ns
t10	Command invalid to CARDxWAIT* tri-state	5	25	ns
t11	D[31:16] valid to first CLK of cycle (write cycle)	10		ns
t12	D[31:16] hold from rising edge of CARDxWAIT*	0		
t13	Chip select to D[31:16] driven (read cycle)	1		ns
t14	D[31:16] setup to rising edge CARDxWAIT* (read cycle)	0		ns
t15	Command invalid to D[31:16] tri-state (read cycle)	5	25	ns

Note: If BUSCLK exceeds 37.5MHz, it must be divided by 2 using CONF5 (see Table 4-8, “Summary of Power-On/Reset Options,” on page 17).

6.3 Power Sequencing

6.3.1 LCD Power Sequencing

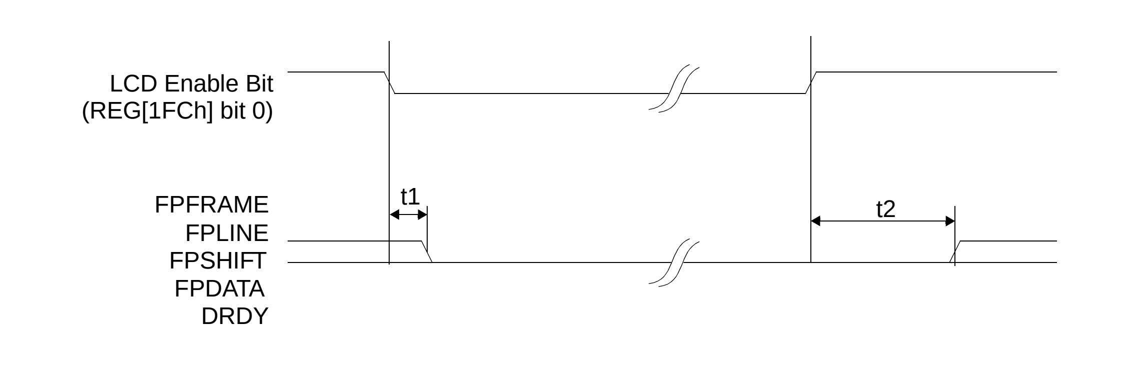


Figure 6-12 LCD Panel Power-off/Power-on Timing

Table 6-15 LCD Panel Power-off/Power-on

Symbol	Parameter	Min	Max	Units
t1	LCD Enable Bit low to FPFRAME, FPLINE, FPSHIFT, FPDATA, DRDY inactive		1	T_{FPLINE}
t2	LCD Enable Bit high to FPFRAME, FPLINE, FPSHIFT, FPDATA, DRDY active	1	2	T_{FPLINE}

Note: Where T_{FPLINE} is the period of FPLINE.

Note: The above timing assumes REG[1F0h] bit 4 is set to 1.

6.3.2 Power Save Status

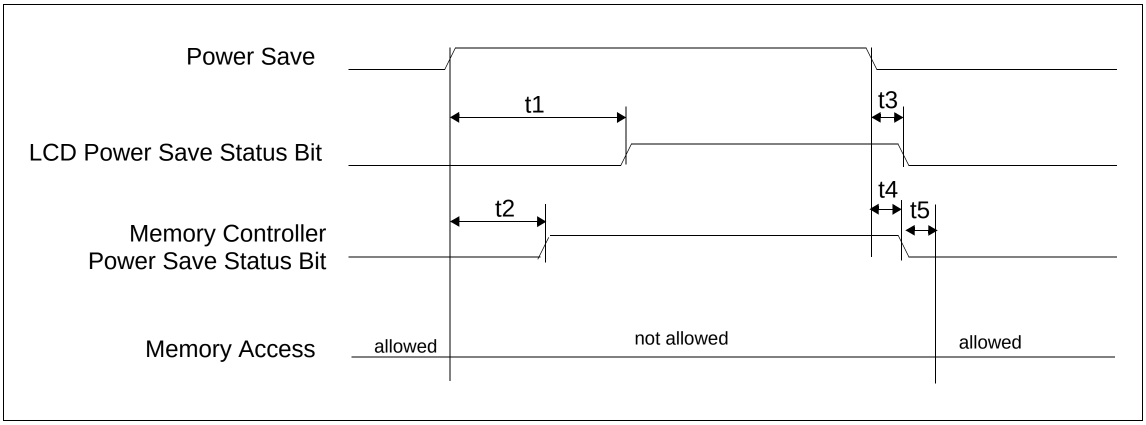


Figure 6-13 Power Save Status Bits and Local Bus Memory Access Relative to Power Save Mode

Note: Memory access should not be performed after Power Save Mode has been initiated.

Note: Power Save is initiated through the Power Save Mode Enable bit (REG[1F0h] bit 0).

Table 6-16 Power Save Status and Local Bus Memory Access Relative to Power Save Mode

Symbol	Parameter	Min	Max	Units
t1	Power Save initiated to rising edge of LCD Power Save Status	1	2	T_{FPLINE}
t2	Power Save initiated to rising edge of Memory Controller Power Save Status		note 1	MCLK
t3	Power Save deactivated to falling edge of LCD Power Save Status		1	$T_{FPFRAME}$
t4	Power Save deactivated to falling edge of Memory Controller Power Save Status		12	MCLK
t5	Falling edge of Memory Controller Power Save Status to the earliest time where memory access is allowed		8	MCLK

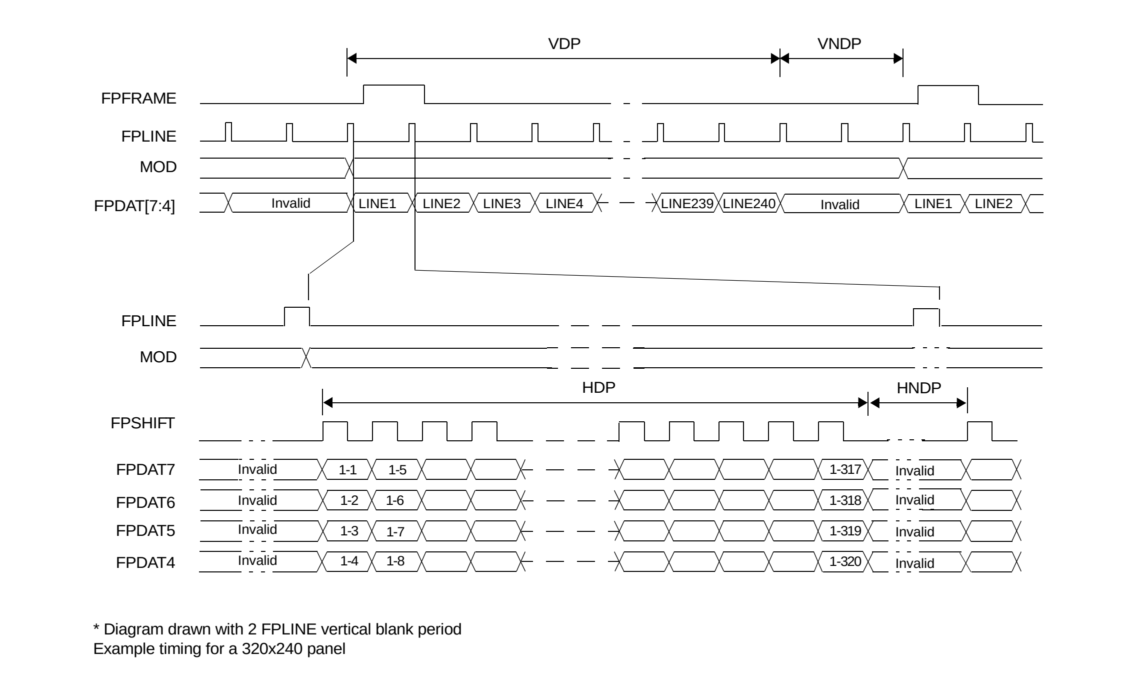
1. $t2_{max}$ = The maximum value for t2 is based on the SDRAM Refresh Rate (REG[021h] bits 2:0) as follows.

Table 6-17 SDRAM Refresh Period Selection

REG[021h] bits 2:0	SDRAM Refresh Period (MCLKs)
000	76
001	140
010	268
011	524

6.4 Display Interface

6.4.1 Single Monochrome 4-Bit Panel Timing



* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 320x240 panel

Figure 6-14 Single Monochrome 4-Bit Panel Timing

VDP =	Vertical Display Period	= (REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1
VNDP =	Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP =	Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP =	Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) × 8Ts

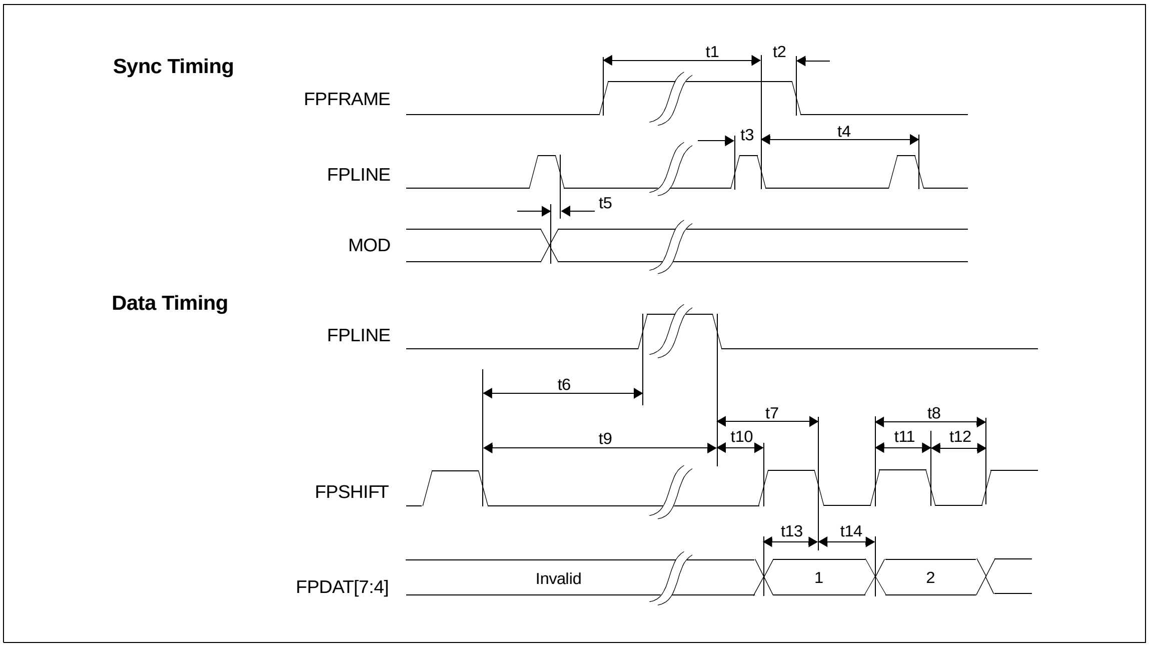


Figure 6-15 Single Monochrome 4-Bit Panel A.C. Timing

Table 6-18 Single Monochrome 4-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	12			Ts
t3	FPLINE pulse width	11			Ts
t4	FPLINE period	note 3			Ts
t5	MOD transition to FPLINE falling edge	3		note 4	Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 5			Ts
t7	FPLINE falling edge to FPSHIFT falling edge	t10 + 2			Ts
t8	FPSHIFT period	4			Ts
t9	FPSHIFT falling edge to FPLINE falling edge	note 6			Ts
t10	FPLINE falling edge to FPSHIFT rising edge	note 7			Ts
t11	FPSHIFT pulse width high	2			Ts
t12	FPSHIFT pulse width low	2			Ts
t13	FPDAT[7:4] setup to FPSHIFT falling edge	2			Ts
t14	FPDAT[7:4] hold to FPSHIFT falling edge	2			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is LCD pixel clock source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t1_{min}$ = $t4_{min}$ - 12
3. $t4_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
4. $t5_{max}$ = [((REG[034h] bits [4:0]) + 1) × 8 + 3]
5. $t6_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 26] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 25] for 16 bpp color depth
6. $t9_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 15] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 14] for 16 bpp color depth
7. $t10_{min}$ = 17 for 4 bpp or 8 bpp color depth
   = 16 for 16 bpp color depth

6.4.2 Single Monochrome 8-Bit Panel Timing

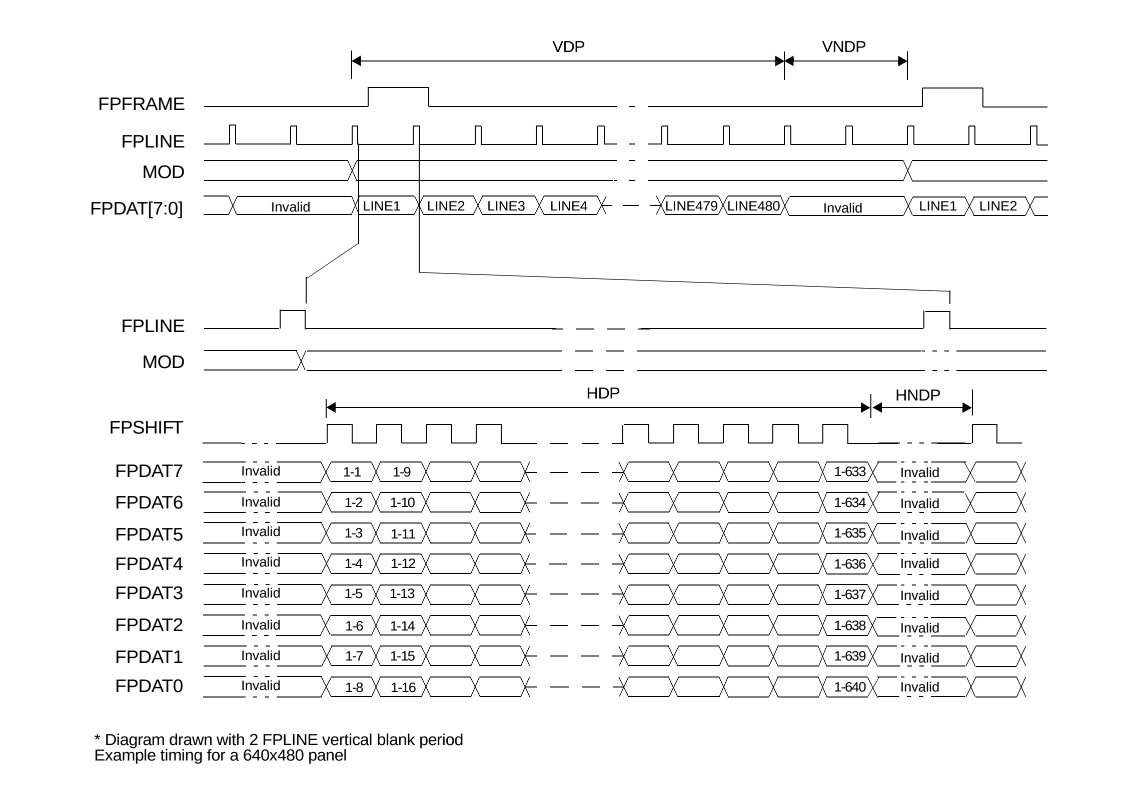


Figure 6-16 Single Monochrome 8-Bit Panel Timing

VDP	= Vertical Display Period	= (REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1
VNDP	= Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP	= Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) × 8Ts

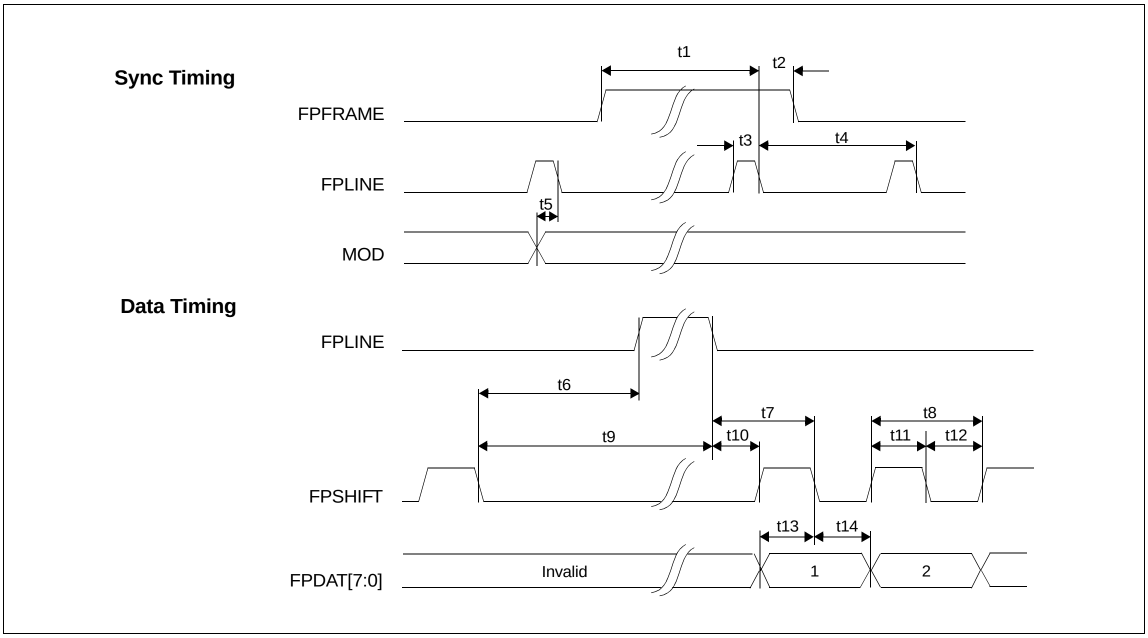


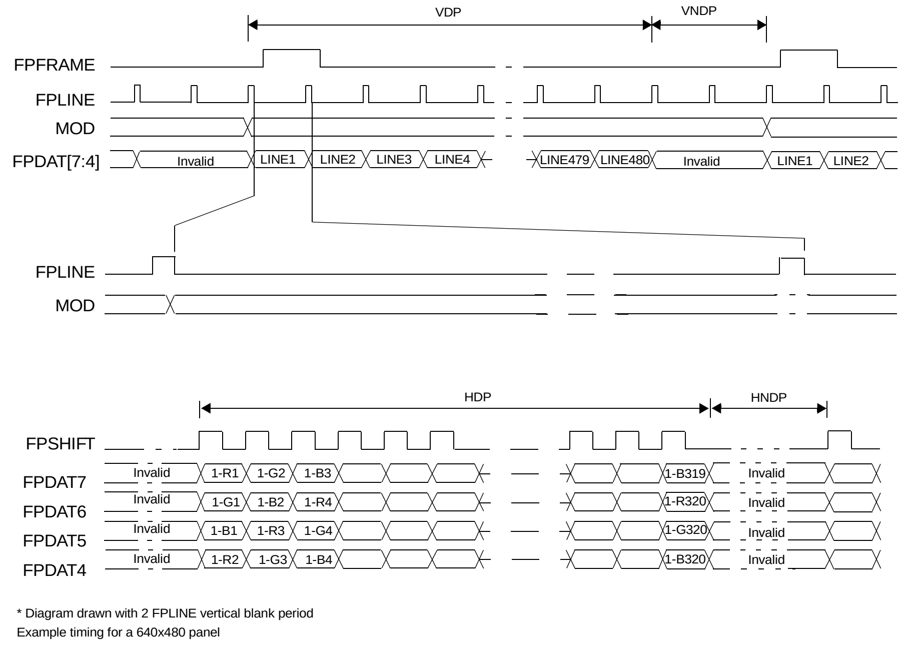
Figure 6-17 Single Monochrome 8-Bit Panel A.C. Timing

Table 6-19 Single Monochrome 8-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	12			Ts
t3	FPLINE pulse width	11			Ts
t4	FPLINE period	note 3			Ts
t5	MOD transition to FPLINE falling edge	3		note 4	Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 5			Ts
t7	FPLINE falling edge to FPSHIFT falling edge	t10 + 4			Ts
t8	FPSHIFT period	8			Ts
t9	FPSHIFT falling edge to FPLINE falling edge	note 6			Ts
t10	FPLINE falling edge to FPSHIFT rising edge	note 7			Ts
t11	FPSHIFT pulse width high	4			Ts
t12	FPSHIFT pulse width low	4			Ts
t13	FPDAT[7:0] setup to FPSHIFT falling edge	4			Ts
t14	FPDAT[7:0] hold to FPSHIFT falling edge	4			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is LCD pixel clock source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t1_{min}$ = $t4_{min}$ - 12
3. $t4_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
4. $t5_{max}$ = [((REG[034h] bits [4:0]) + 1) × 8 + 3]
5. $t6_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 24] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 23] for 16 bpp color depth
6. $t9_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 13] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 12] for 16 bpp color depth
7. $t10_{min}$ = 17 for 4 bpp or 8 bpp color depth
   = 16 for 16 bpp color depth

6.4.3 Single Color 4-Bit Panel Timing



* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640x480 panel

Figure 6-18 Single Color 4-Bit Panel Timing

VDP	= Vertical Display Period	= (REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1
VNDP	= Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP	= Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) × 8Ts

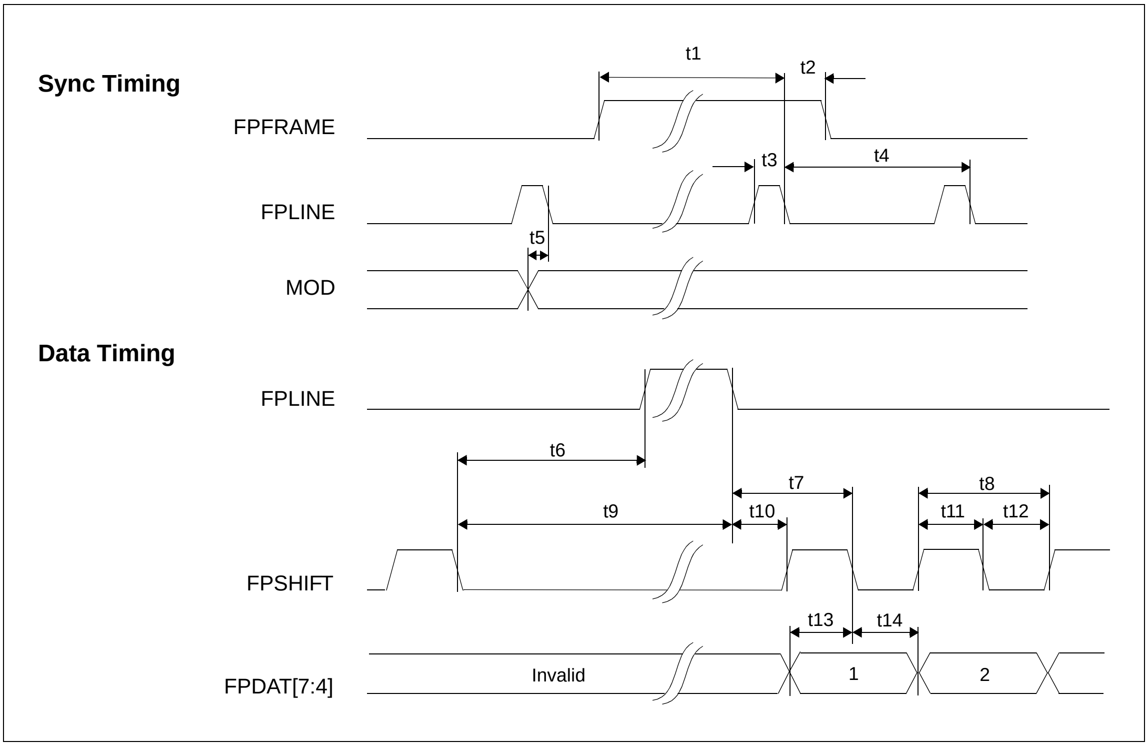


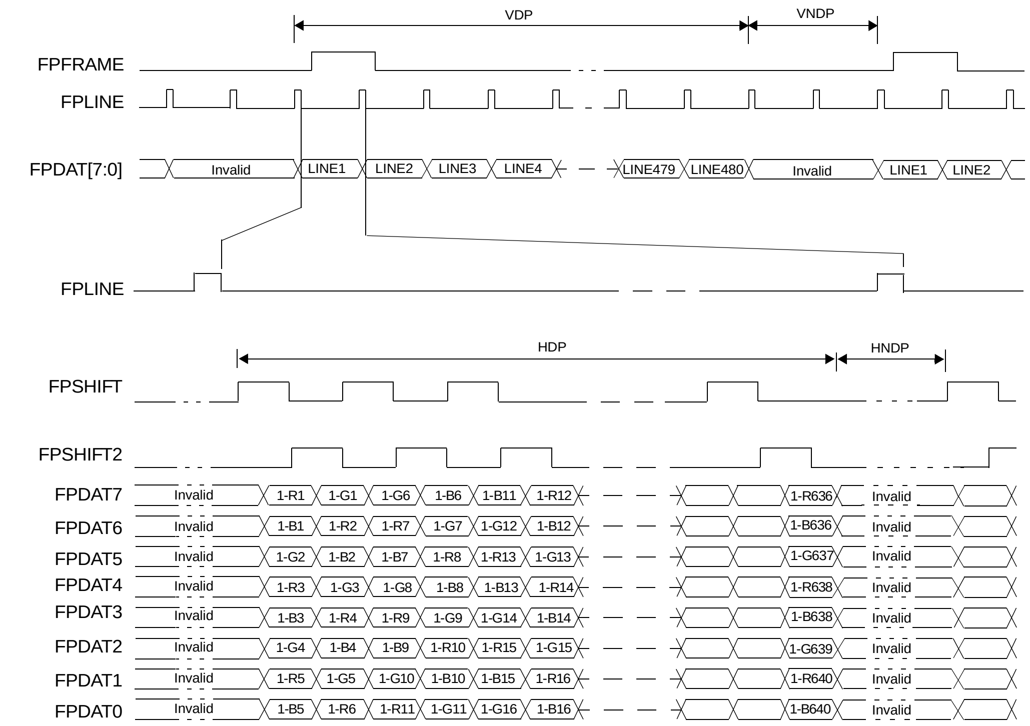
Figure 6-19 Single Color 4-Bit Panel A.C. Timing

Table 6-20 Single Color 4-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	12			Ts
t3	FPLINE pulse width	11			Ts
t4	FPLINE period	note 3			Ts
t5	MOD transition to FPLINE falling edge	3		note 4	Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 5			Ts
t7	FPLINE falling edge to FPSHIFT falling edge	t10 + 0.5			Ts
t8	FPSHIFT period	1			Ts
t9	FPSHIFT falling edge to FPLINE falling edge	note 6			Ts
t10	FPLINE falling edge to FPSHIFT rising edge	note 7			Ts
t11	FPSHIFT pulse width high	0.5			Ts
t12	FPSHIFT pulse width low	0.5			Ts
t13	FPDAT[7:4] setup to FPSHIFT falling edge	0.5			Ts
t14	FPDAT[7:4] hold from FPSHIFT falling edge	0.5			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is LCD pixel clock source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t1_{min}$ = $t4_{min}$ - 12
3. $t4_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
4. $t5_{max}$ = [((REG[034h] bits [4:0]) + 1) × 8 + 3]
5. $t6_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 28.5] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 27.5] for 16 bpp color depth
6. $t9_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 17.5] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 16.5] for 16 bpp color depth
7. $t10_{min}$ = 18 for 4 bpp or 8 bpp color depth
   = 17 for 16 bpp color depth

6.4.4 Single Color 8-Bit Panel Timing (Format 1)



* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640x480 panel

Figure 6-20 Single Color 8-Bit Panel Timing (Format 1)

VDP	= Vertical Display Period	= (REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1
VNDP	= Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP	= Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) × 8Ts

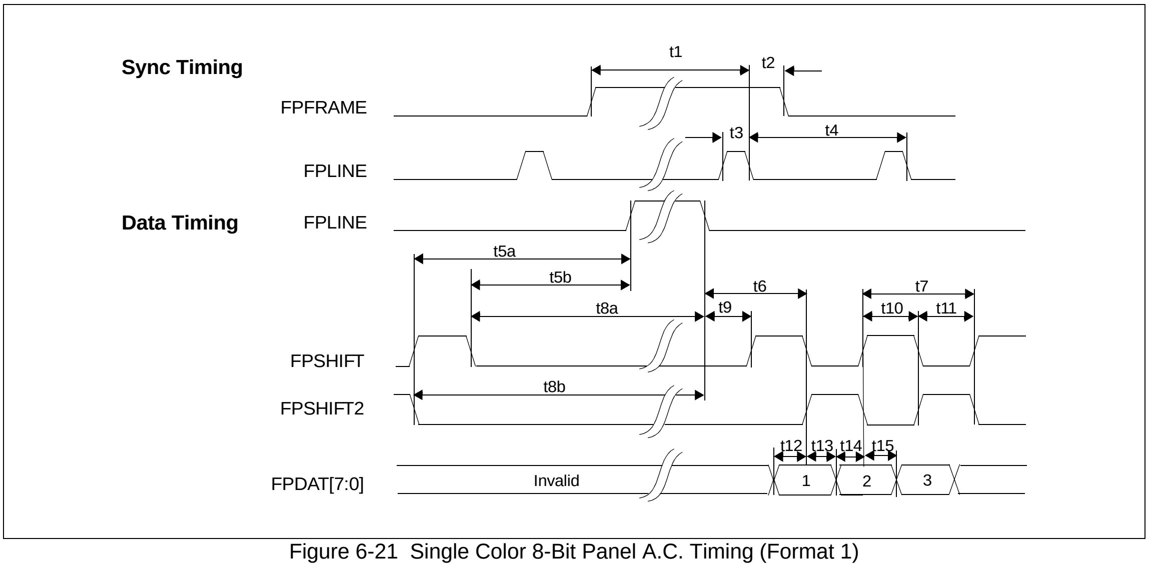


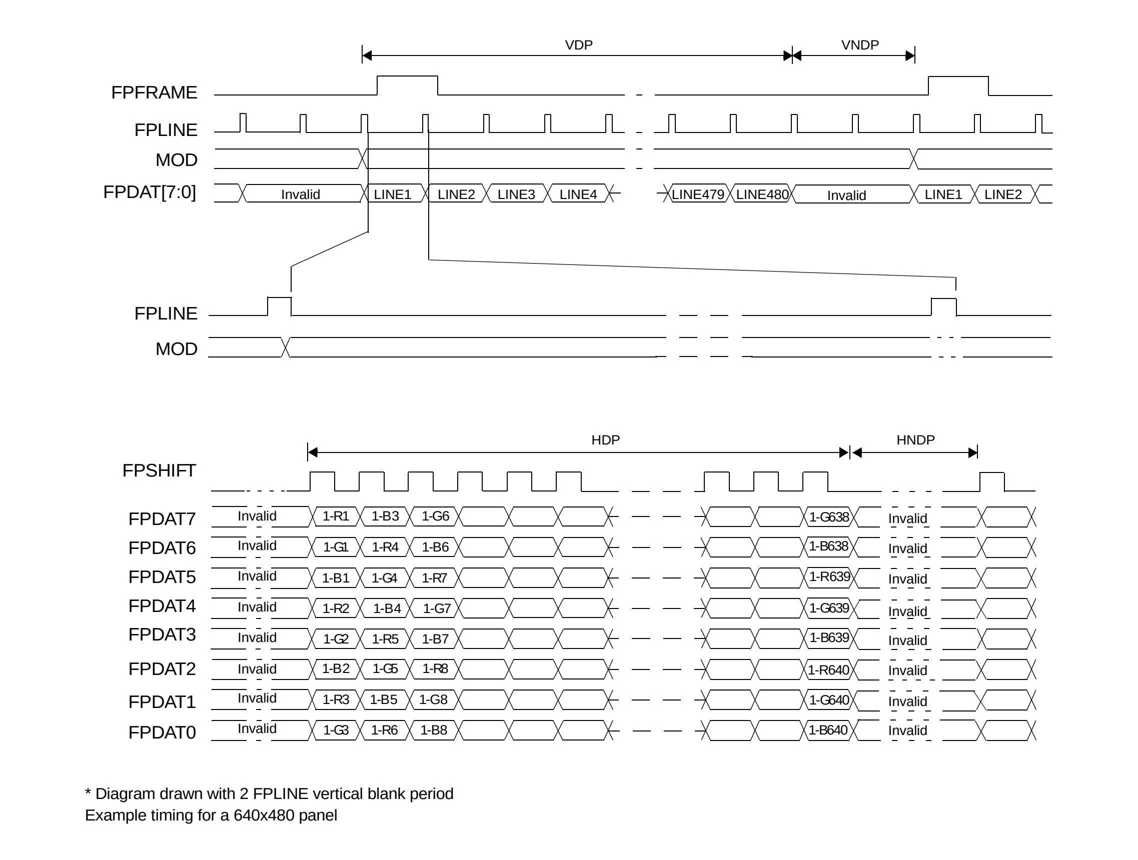
Figure 6-21 Single Color 8-Bit Panel A.C. Timing (Format 1)

Table 6-21 Single Color 8-Bit Panel A.C. Timing (Format 1)

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	12			Ts
t3	FPLINE pulse width	11			Ts
t4	FPLINE period	note 3			Ts
t5a	FPSHIFT2 falling edge to FPLINE rising edge	note 4			Ts
t5b	FPSHIFT falling edge to FPLINE rising edge	note 5			Ts
t6	FPLINE falling edge to FPSHIFT falling, FPSHIFT2 rising edge	t9 + 2			Ts
t7	FPSHIFT, FPSHIFT2 period	4			Ts
t8a	FPSHIFT falling edge to FPLINE falling edge	note 6			Ts
t8b	FPSHIFT2 falling edge to FPLINE falling edge	note 7			Ts
t9	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts
t10	FPSHIFT pulse width high, FPSHIFT2 pulse width low	2			Ts
t11	FPSHIFT pulse width low, FPSHIFT2 pulse width high	2			Ts
t12	FPDAT[7:0] setup to FPSHIFT falling edge	1			Ts
t13	FPDAT[7:0] hold from FPSHIFT falling edge	1			Ts
t14	FPDAT[7:0] setup to FPSHIFT2 falling edge	1			Ts
t15	FPDAT[7:0] hold from FPSHIFT2 falling edge	1			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t1_{min}$ = $t4_{min}$ - 12Ts
3. $t4_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
4. $t5a_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 26] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 25] for 16 bpp color depth
5. $t5b_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 28] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 27] for 16 bpp color depth
6. $t8a_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 17] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 16] for 16 bpp color depth
7. $t8b_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 15] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 14] for 16 bpp color depth
8. $t9_{min}$ = 17 for 4 bpp or 8 bpp color depth
   = 16 for 16 bpp color depth

6.4.5 Single Color 8-Bit Panel Timing (Format 2)


VDP
VNDP
FPFRAME
FPLINE
MOD
FPDAT[7:0]
Invalid
LINE1
LINE2
LINE3
LINE4
LINE479
LINE480
Invalid
LINE1
LINE2
FPLINE
MOD
HDP
HNDP
FPSHIFT
FPDAT7
FPDAT6
FPDAT5
FPDAT4
FPDAT3
FPDAT2
FPDAT1
FPDAT0
Invalid 1-R1 1-B3 1-G6 1-G638 Invalid
Invalid 1-G1 1-R4 1-B6 1-B638 Invalid
Invalid 1-B1 1-G4 1-R7 1-R639 Invalid
Invalid 1-R2 1-B4 1-G7 1-G639 Invalid
Invalid 1-G2 1-R5 1-B7 1-B639 Invalid
Invalid 1-B2 1-G5 1-R8 1-R640 Invalid
Invalid 1-R3 1-B5 1-G8 1-G640 Invalid
Invalid 1-G3 1-R6 1-B8 1-B640 Invalid

* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640x480 panel

Figure 6-22 Single Color 8-Bit Panel Timing (Format 2)

VDP	= Vertical Display Period	= (REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1
VNDP	= Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP	= Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) × 8Ts

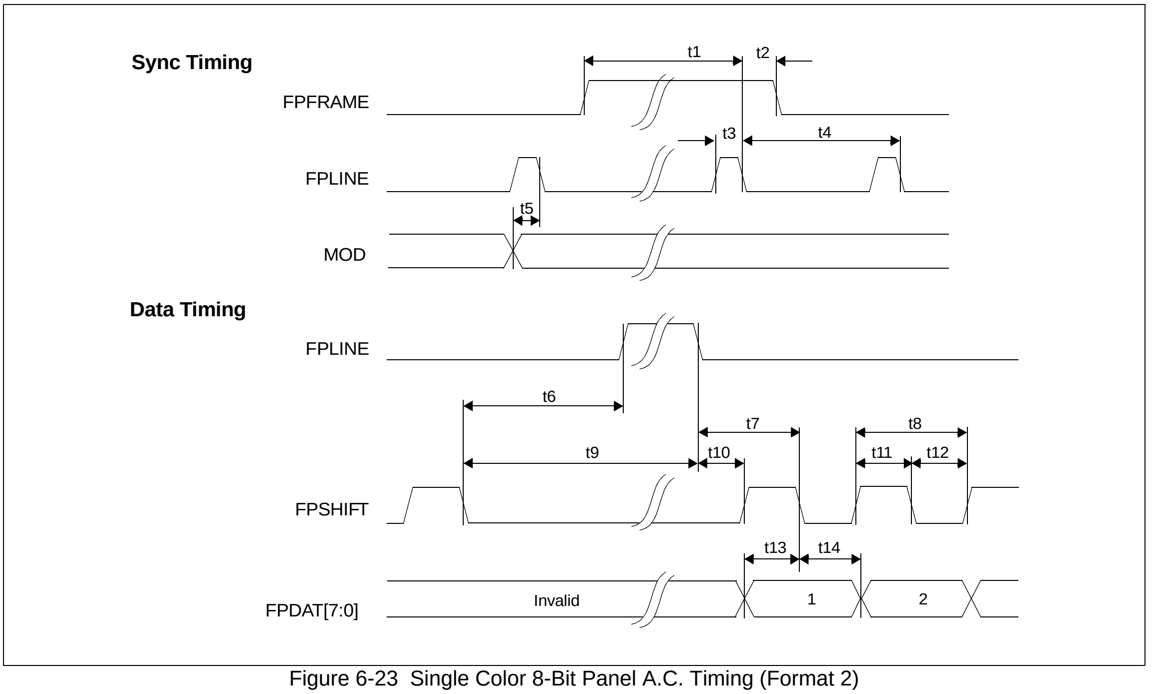


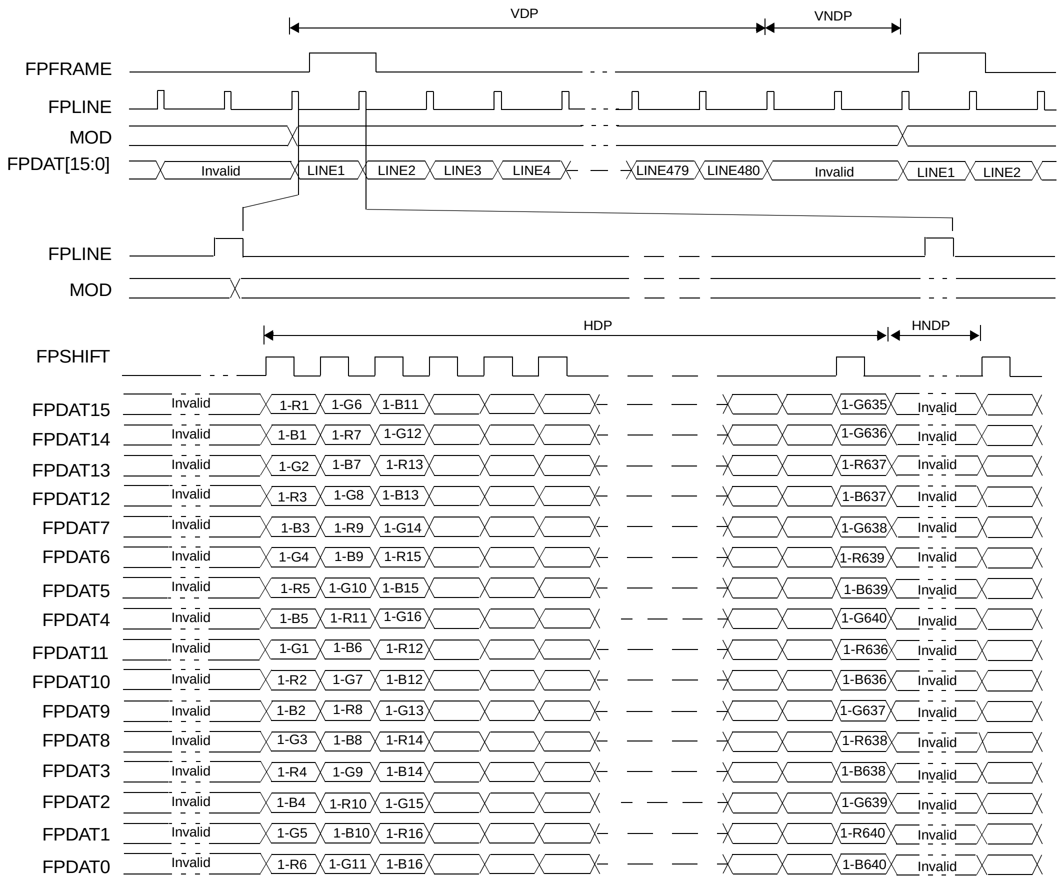
Figure 6-23 Single Color 8-Bit Panel A.C. Timing (Format 2)

Table 6-22 Single Color 8-Bit Panel A.C. Timing (Format 2)

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	12			Ts
t3	FPLINE pulse width	11			Ts
t4	FPLINE period	note 3			Ts
t5	MOD transition to FPLINE falling edge	3		note 4	Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 5			Ts
t7	FPLINE falling edge to FPSHIFT falling edge	t10 + 2			Ts
t8	FPSHIFT period	2			Ts
t9	FPSHIFT falling edge to FPLINE falling edge	note 6			Ts
t10	FPLINE falling edge to FPSHIFT rising edge	note 7			Ts
t11	FPSHIFT pulse width high	1			Ts
t12	FPSHIFT pulse width low	1			Ts
t13	FPDAT[7:0] setup to FPSHIFT falling edge	1			Ts
t14	FPDAT[7:0] hold to FPSHIFT falling edge	1			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t1_{min}$ = $t3_{min}$ - 12
3. $t3_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
4. $t5_{max}$ = [((REG[034h] bits [4:0]) + 1) × 8 + 3]
5. $t6_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 27] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 26] for or 16 bpp color depth
6. $t9_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 16] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 15] for 16 bpp color depth
7. $t10_{min}$ = 17 for 4 bpp or 8 bpp color depth
   = 16 for 16 bpp color depth

6.4.6 Single Color 16-Bit Panel Timing


VDP
VNDP
FPFRAME
FPLINE
MOD
FPDAT[15:0]
Invalid
LINE1
LINE2
LINE3
LINE4
LINE479
LINE480
Invalid
LINE1
LINE2
FPLINE
MOD
HDP
HNDP
FPSHIFT
FPDAT15 Invalid 1-R1 1-G6 1-B11 1-G635 Invalid
FPDAT14 Invalid 1-B1 1-R7 1-G12 1-G636 Invalid
FPDAT13 Invalid 1-G2 1-B7 1-R13 1-R637 Invalid
FPDAT12 Invalid 1-R3 1-G8 1-B13 1-B637 Invalid
FPDAT7 Invalid 1-B3 1-R9 1-G14 1-G638 Invalid
FPDAT6 Invalid 1-G4 1-B9 1-R15 1-R639 Invalid
FPDAT5 Invalid 1-R5 1-G10 1-B15 1-B639 Invalid
FPDAT4 Invalid 1-B5 1-R11 1-G16 1-G640 Invalid
FPDAT11 Invalid 1-G1 1-B6 1-R12 1-R636 Invalid
FPDAT10 Invalid 1-R2 1-G7 1-B12 1-B636 Invalid
FPDAT9 Invalid 1-B2 1-R8 1-G13 1-G637 Invalid
FPDAT8 Invalid 1-G3 1-B8 1-R14 1-R638 Invalid
FPDAT3 Invalid 1-R4 1-G9 1-B14 1-B638 Invalid
FPDAT2 Invalid 1-B4 1-R10 1-G15 1-G639 Invalid
FPDAT1 Invalid 1-G5 1-B10 1-R16 1-R640 Invalid
FPDAT0 Invalid 1-R6 1-G11 1-B16 1-B640 Invalid

* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640x480 panel

Figure 6-24 Single Color 16-Bit Panel Timing

VDP	= Vertical Display Period	= (REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1
VNDP	= Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP	= Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) × 8Ts

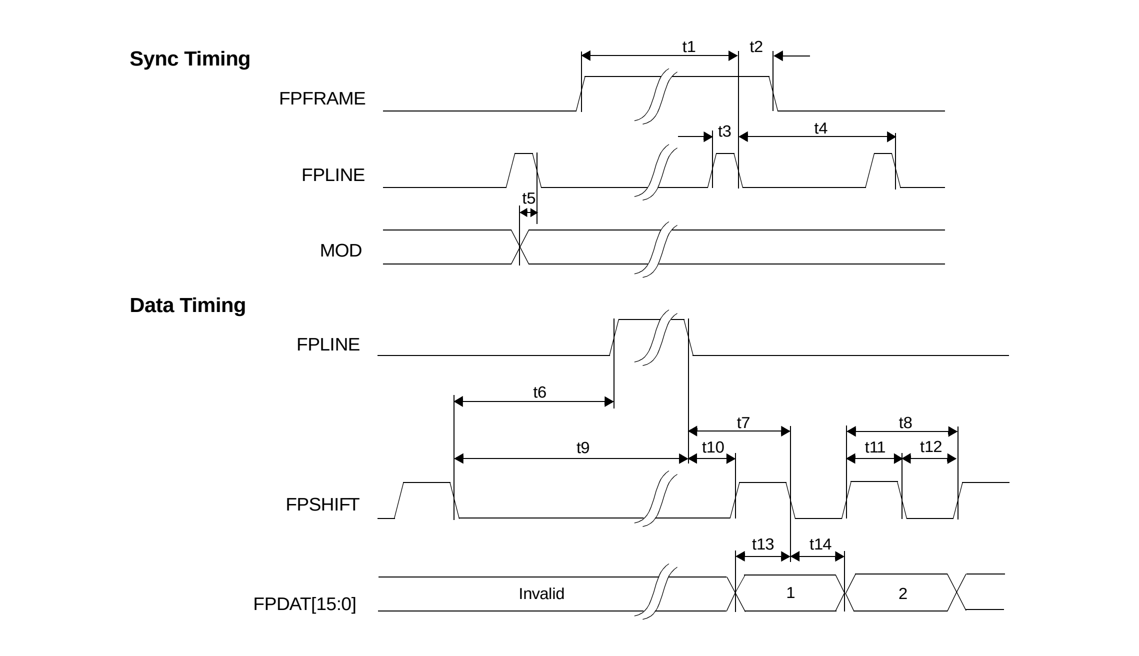


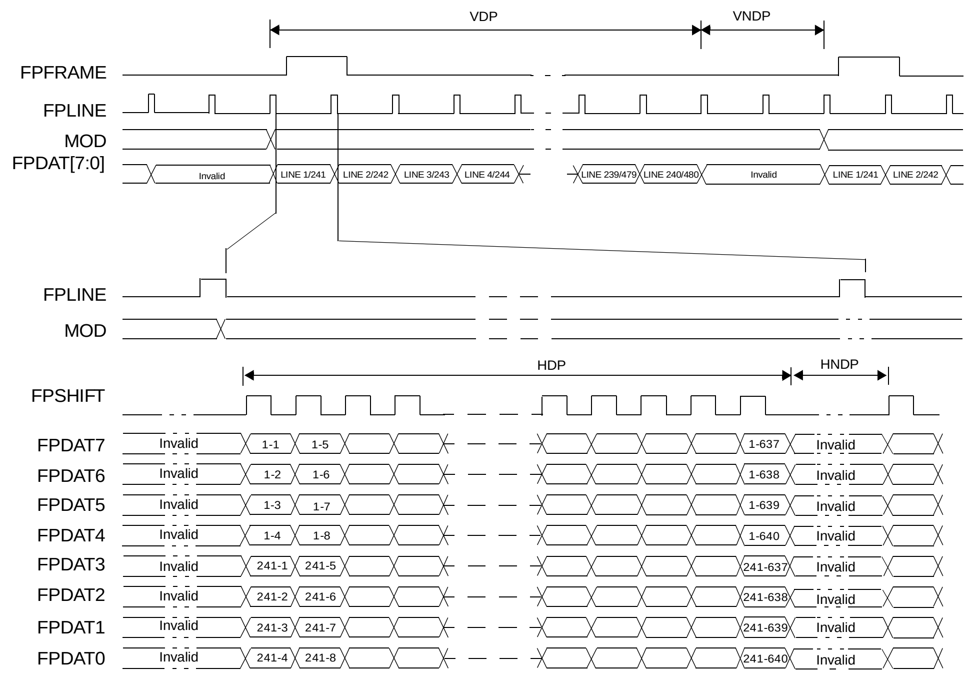
Figure 6-25 Single Color 16-Bit Panel A.C. Timing

Table 6-23 Single Color 16-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	12			Ts
t3	FPLINE pulse width	11			Ts
t4	FPLINE period	note 3			Ts
t5	MOD transition to FPLINE falling edge	3		note 4	Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 5			Ts
t7	FPLINE falling edge to FPSHIFT falling edge	t10 + 3			Ts
t8	FPSHIFT period	5			Ts
t9	FPSHIFT falling edge to FPLINE falling edge	note 6			Ts
t10	FPLINE falling edge to FPSHIFT rising edge	note 7			Ts
t11	FPSHIFT pulse width high	2			Ts
t12	FPSHIFT pulse width low	2			Ts
t13	FPDAT[15:0] setup to FPSHIFT falling edge	2			Ts
t14	FPDAT[15:0] hold to FPSHIFT falling edge	2			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t1_{min}$ = $t3_{min}$ - 12
3. $t3_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
4. $t5_{max}$ = [((REG[034h] bits [4:0]) + 1) × 8 + 3]
5. $t6_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 26] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 25] for 16 bpp color depth
6. $t9_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 15] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 14] for 16 bpp color depth
7. $t10_{min}$ = 17 for 4 bpp or 8 bpp color depth
   = 16 for 16 bpp color depth

6.4.7 Dual Monochrome 8-Bit Panel Timing



* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640x480 panel

Figure 6-26 Dual Monochrome 8-Bit Panel Timing

VDP	= Vertical Display Period	= (REG[039h] bits [1:0], REG[038h] bits [7:1])
VNDP	= Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP	= Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) × 8Ts

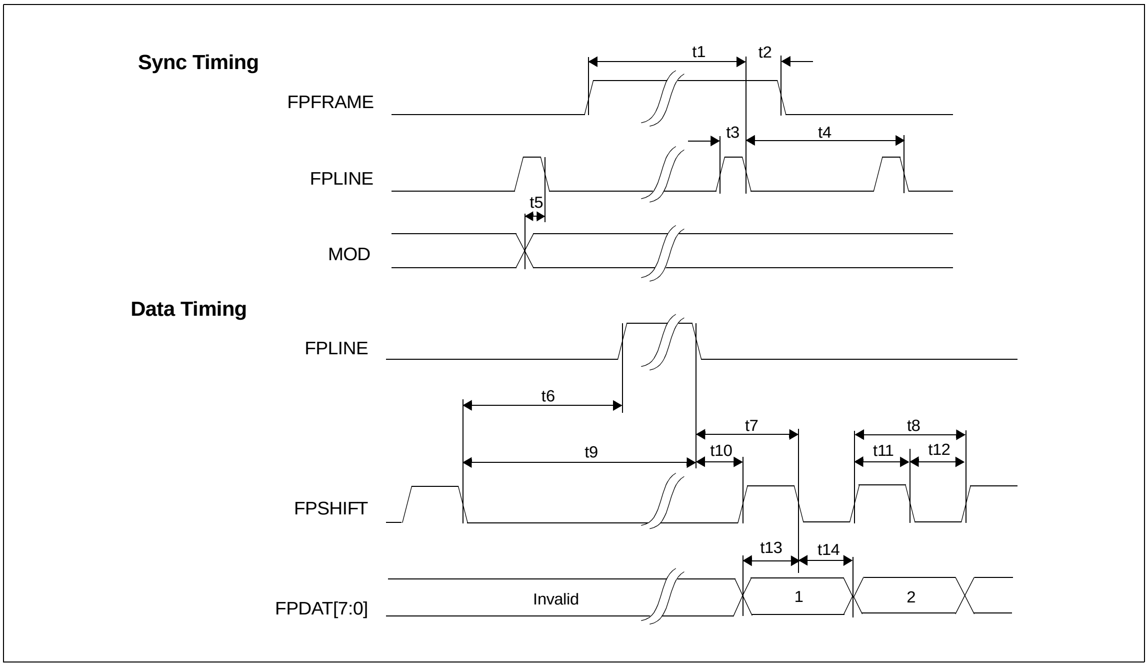


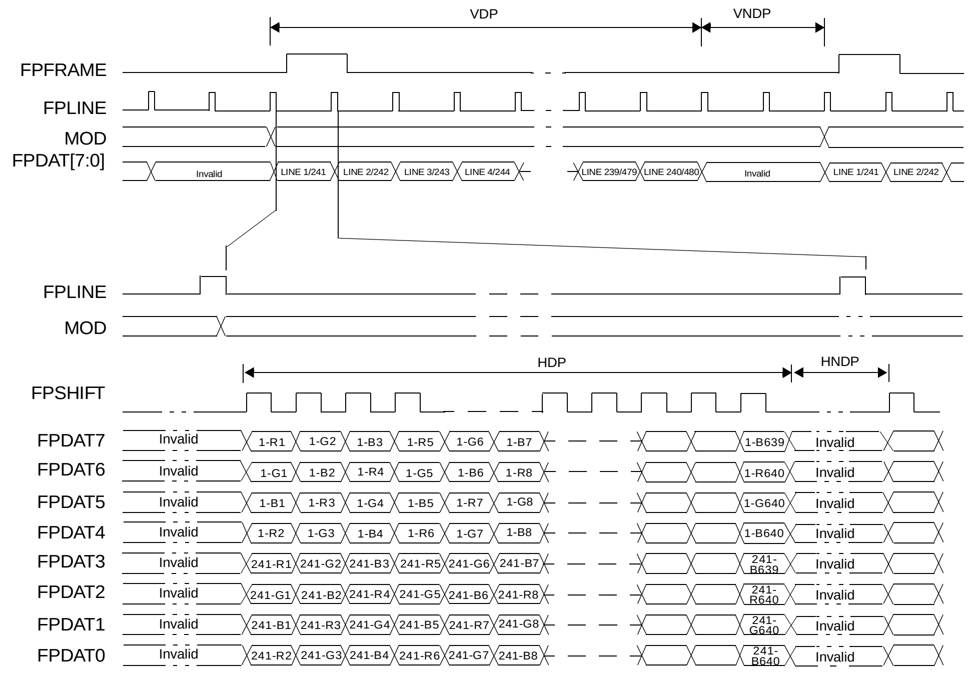
Figure 6-27 Dual Monochrome 8-Bit Panel A.C. Timing

Table 6-24 Dual Monochrome 8-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	12			Ts
t3	FPLINE pulse width	11			Ts
t4	FPLINE period	note 3			Ts
t5	MOD transition to FPLINE falling edge	3		note 4	Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 5			Ts
t7	FPLINE falling edge to FPSHIFT falling edge	t10 + 2			Ts
t8	FPSHIFT period	4			Ts
t9	FPSHIFT falling edge to FPLINE falling edge	note 6			Ts
t10	FPLINE falling edge to FPSHIFT rising edge	note 7			Ts
t12	FPSHIFT pulse width low	2			Ts
t11	FPSHIFT pulse width high	2			Ts
t13	FPDAT[7:0] setup to FPSHIFT falling edge	2			Ts
t14	FPDAT[7:0] hold to FPSHIFT falling edge	2			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t1_{min}$ = $t3_{min}$ - 12
3. $t3_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
4. $t5_{max}$ = [((REG[034h] bits [4:0]) + 1) × 8 + 3]
5. $t6_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 18] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 17] for 16 bpp color depth
6. $t9_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 7] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 6] for 16 bpp color depth
7. $t10_{min}$ = 9 for 4 bpp or 8 bpp color depth
   = 8 for 16 bpp color depth

6.4.8 Dual Color 8-Bit Panel Timing



* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640x480 panel

Figure 6-28 Dual Color 8-Bit Panel Timing

VDP	= Vertical Display Period	= ((REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1) ÷ 2
VNDP	= Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP	= Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) × 8Ts

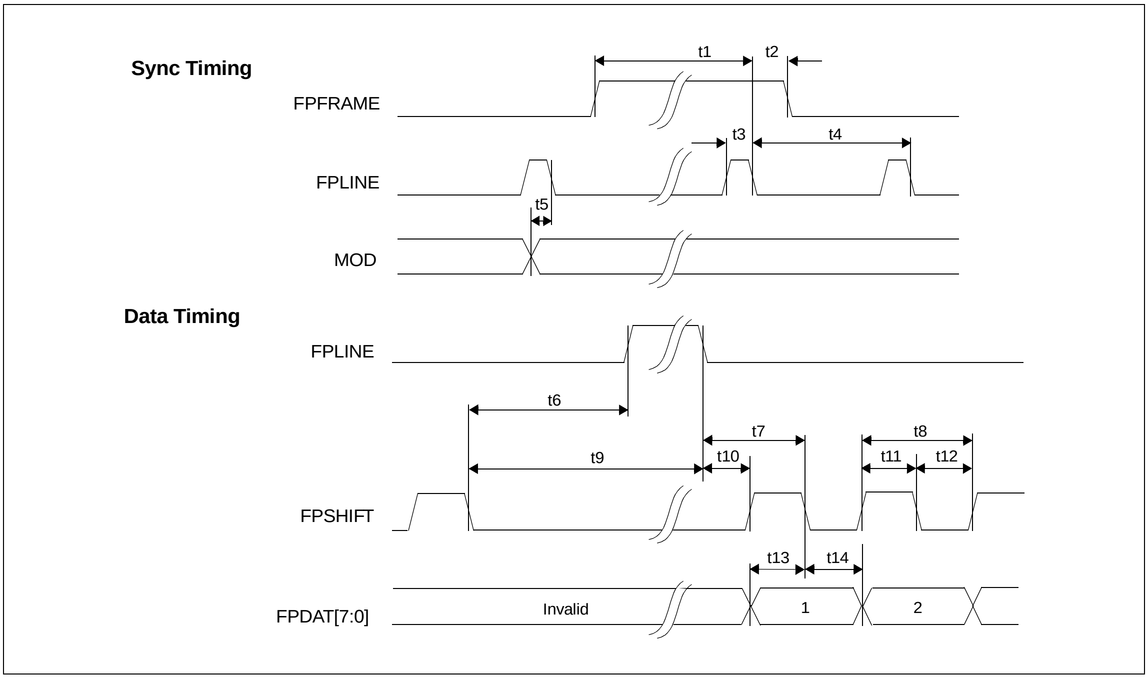


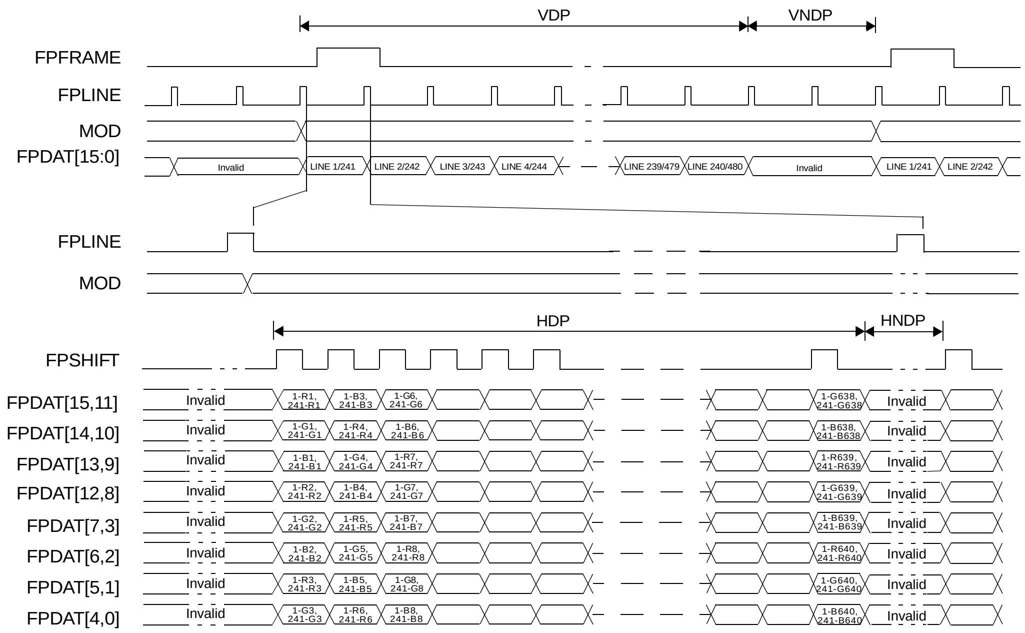
Figure 6-29 Dual Color 8-Bit Panel A.C. Timing

Table 6-25 Dual Color 8-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	12			Ts
t3	FPLINE pulse width	11			Ts
t4	FPLINE period	note 3			Ts
t5	MOD transition to FPLINE falling edge	3		note 4	Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 5			Ts
t7	FPLINE falling edge to FPSHIFT falling edge	t10 + 0.5			Ts
t8	FPSHIFT period	1			Ts
t9	FPSHIFT falling edge to FPLINE falling edge	note 6			Ts
t10	FPLINE falling edge to FPSHIFT rising edge	note 7			Ts
t11	FPSHIFT pulse width high	0.5			Ts
t12	FPSHIFT pulse width low	0.5			Ts
t13	FPDAT[7:0] setup to FPSHIFT falling edge	0.5			Ts
t14	FPDAT[7:0] hold to FPSHIFT falling edge	0.5			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t1_{min}$ = $t3_{min}$ - 12
3. $t3_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
4. $t5_{max}$ = [((REG[034h] bits [4:0]) + 1) × 8 + 3]
5. $t6_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 18.5] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 17.5] for 16 bpp color depth
6. $t9_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 8.5] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 6.5] for 16 bpp color depth
7. $t10_{min}$ = 10 for 4 bpp or 8 bpp color depth
   = 9 for 16 bpp color depth

6.4.9 Dual Color 16-Bit Panel Timing



* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640x480 panel

Figure 6-30 Dual Color 16-Bit Panel Timing

VDP	= Vertical Display Period	= ((REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1) ÷ 2
VNDP	= Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP	= Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) × 8Ts

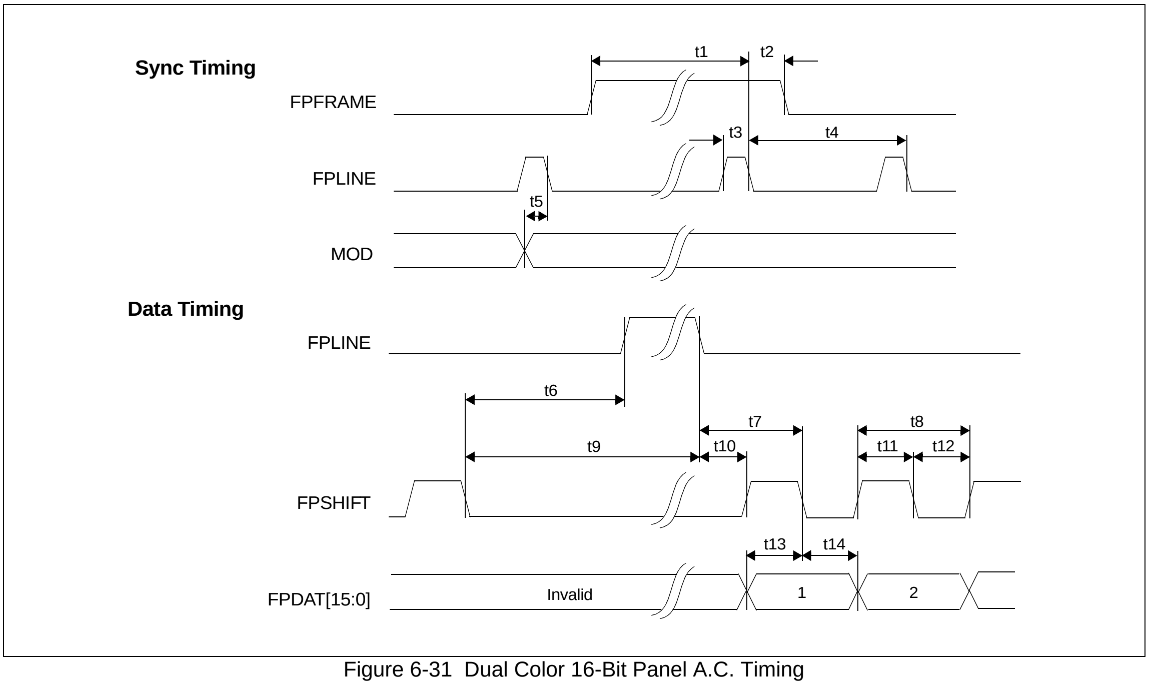


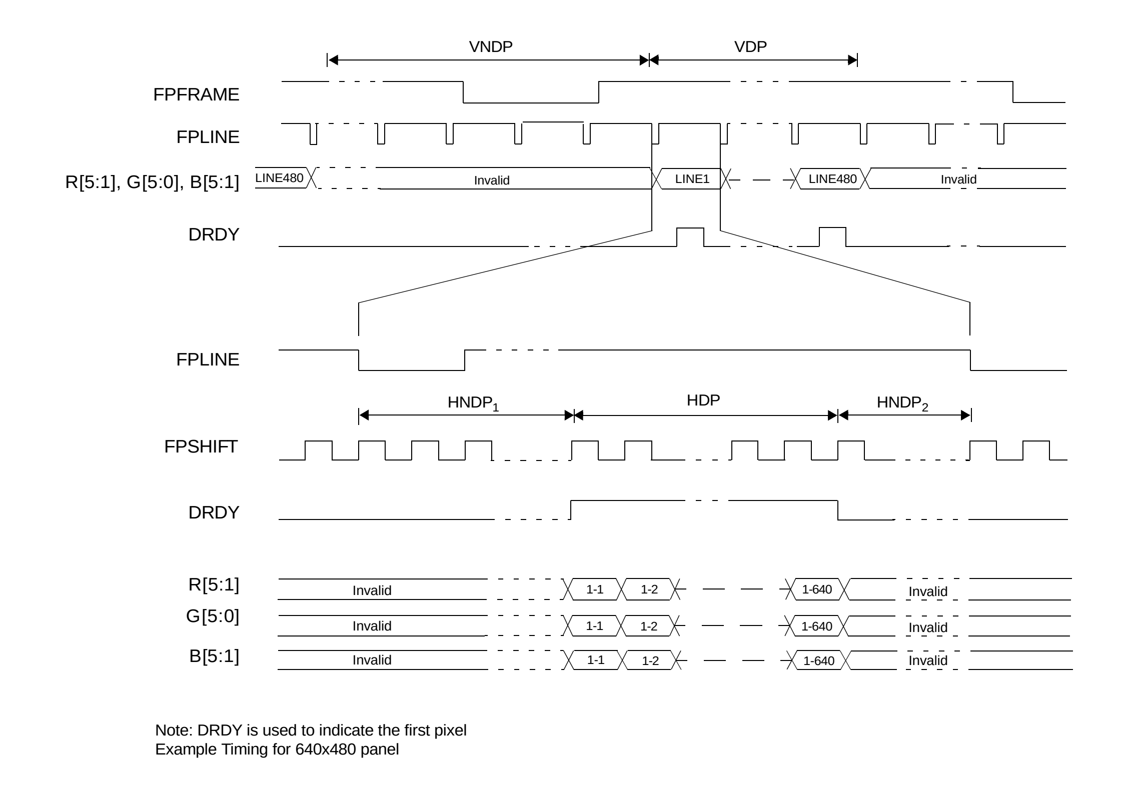
Figure 6-31 Dual Color 16-Bit Panel A.C. Timing

Table 6-26 Dual Color 16-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	12			Ts
t3	FPLINE pulse width	11			Ts
t4	FPLINE period	note 3			Ts
t5	MOD transition to FPLINE falling edge	3		note 4	Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 5			Ts
t7	FPLINE falling edge to FPSHIFT falling edge	t10 + 2			Ts
t8	FPSHIFT period	2			Ts
t9	FPSHIFT falling edge to FPLINE falling edge	note 6			Ts
t10	FPLINE falling edge to FPSHIFT rising edge	note 7			Ts
t11	FPSHIFT pulse width high	1			Ts
t12	FPSHIFT pulse width low	1			Ts
t13	FPDAT[15:0] setup to FPSHIFT falling edge	1			Ts
t14	FPDAT[15:0] hold to FPSHIFT falling edge	1			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t1_{min}$ = $t3_{min}$ - 12
3. $t3_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
4. $t5_{max}$ = [((REG[034h] bits [4:0]) + 1) × 8 + 3]
5. $t6_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 19] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 18] for 16 bpp color depth
6. $t9_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - 8] for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) × 8 - 7] for 16 bpp color depth
7. $t10_{min}$ = 9 for 4 bpp or 8 bpp color depth
   = 8 for 16 bpp color depth

6.4.10 TFT/D-TFD Panel Timing



Note: DRDY is used to indicate the first pixel
Example Timing for 640x480 panel

Figure 6-32 TFT/D-TFD Panel Timing

VDP	= Vertical Display Period	= (REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1
VNDP	= Vertical Non-Display Period	= (REG[03Ah] bits [5:0]) + 1
HDP	= Horizontal Display Period	= ((REG[032h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= HNDP$_1$ + HNDP$_2$ = ((REG[034h] bits [4:0]) + 1) × 8Ts

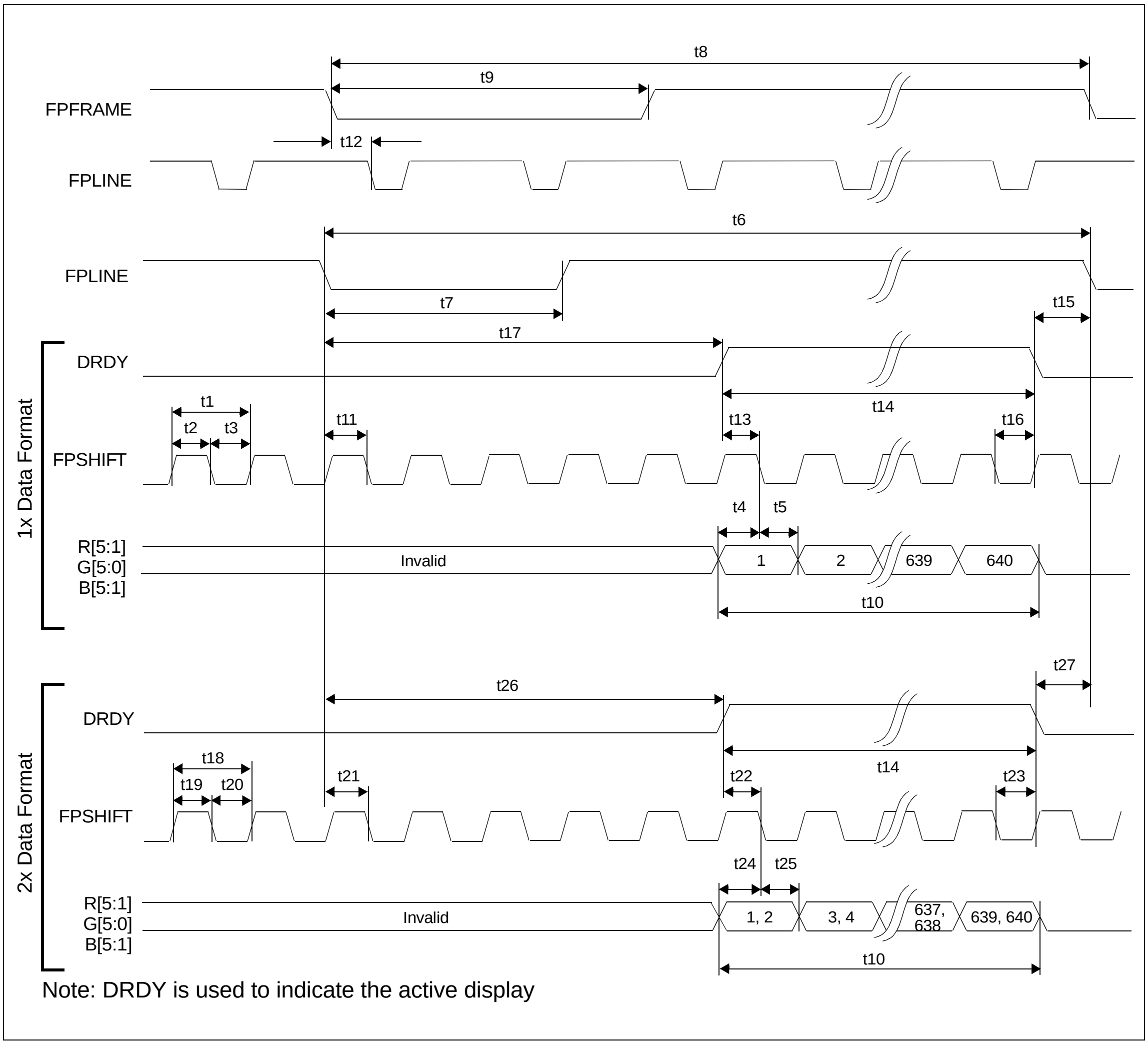


Figure 6-33 TFT/D-TFD A.C. Timing

Table 6-27 TFT/D-TFD A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPSHIFT period	1			Ts (note 1)
t2	FPSHIFT pulse width high	0.45			Ts
t3	FPSHIFT pulse width low	0.45			Ts
t4	data setup to FPSHIFT falling edge	0.45			Ts
t5	data hold from FPSHIFT falling edge	0.45			Ts
t6	FPLINE cycle time	note 2			Ts
t7	FPLINE pulse width low	note 3			Ts
t8	FPFRAME cycle time	note 4			lines
t9	FPFRAME pulse width low	note 5			lines
t10	horizontal display period	note 6			Ts
t11	FPLINE setup to FPSHIFT falling edge	0.45			Ts
t12	FPFRAME falling edge to FPLINE falling edge phase difference	note 7			Ts
t13	DRDY to FPSHIFT falling edge setup time	0.45			Ts
t14	DRDY pulse width	note 8			Ts
t15	DRDY falling edge to FPLINE falling edge (1x)	note 9			Ts
t16	DRDY hold from FPSHIFT falling edge	0.45			Ts
t17	FPLINE Falling edge to DRDY active (1x)	note 10			Ts
t18	FPSHIFT period	2			Ts
t19	FPSHIFT pulse width high	1			Ts
t20	FPSHIFT pulse width low	1			Ts
t21	FPLINE setup to FPSHIFT falling edge	note 11			Ts
t22	DRDY to FPSHIFT falling edge setup time	1			Ts
t23	DRDY hold from FPSHIFT falling edge	1			Ts
t24	data setup to FPSHIFT falling edge	1			Ts
t25	data hold from FPSHIFT falling edge	1			Ts
t26	FPLINE Falling edge to DRDY active (2x)	note 12			Ts
t27	DRDY falling edge to FPLINE falling edge (2x)	note 13			Ts

1. Ts = LCD pixel clock period. LCD pixel clock frequency is source divided by 1, 2, 3 or 4 (see REG[014h]).
2. $t6_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8]
3. $t7_{min}$ = [((REG[036h] bits [3:0]) + 1) × 8]
4. $t8_{min}$ = [((REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1) + ((REG[03Ah] bits [5:0]) + 1)]
5. $t9_{min}$ = [((REG[03Ch] bits [2:0]) + 1)]
6. $t10_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8]
7. $t12_{min}$ = [(REG[035h] bits [4:0]) × 8 + 1]
8. $t14_{min}$ = [((REG[032h] bits [6:0]) + 1) × 8]
9. $t15_{min}$ = [(REG[035h] bits [4:0]) × 8 + 5] for 4 bpp or 8 bpp color depth
   = [(REG[035h] bits [4:0]) × 8 + 6] for 16 bpp color depth
10. $t17_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - (REG[035h] bits [4:0]) × 8 - 5] for 4 bpp or 8 bpp color depth
    = [((REG[034h] bits [4:0]) + 1) × 8 - (REG[035h] bits [4:0]) × 8 - 6] for 16 bpp color depth
11. $t21_{min}$ = 1 for 4 bpp or 8 bpp color depth
    = 0 for 16 bpp color depth
12. $t26_{min}$ = [((REG[034h] bits [4:0]) + 1) × 8 - (REG[035h] bits [4:0]) × 8 - 4] for 4 bpp or 8 bpp color depth
    = [((REG[034h] bits [4:0]) + 1) × 8 - (REG[035h] bits [4:0]) × 8 - 5] for 16 bpp color depth
13. $t27_{min}$ = [(REG[035h] bits [4:0]) × 8 + 4] for 4 bpp or 8 bpp color depth
    = [(REG[035h] bits [4:0]) × 8 + 5] for 16 bpp color depth

6.4.11 CRT Timing

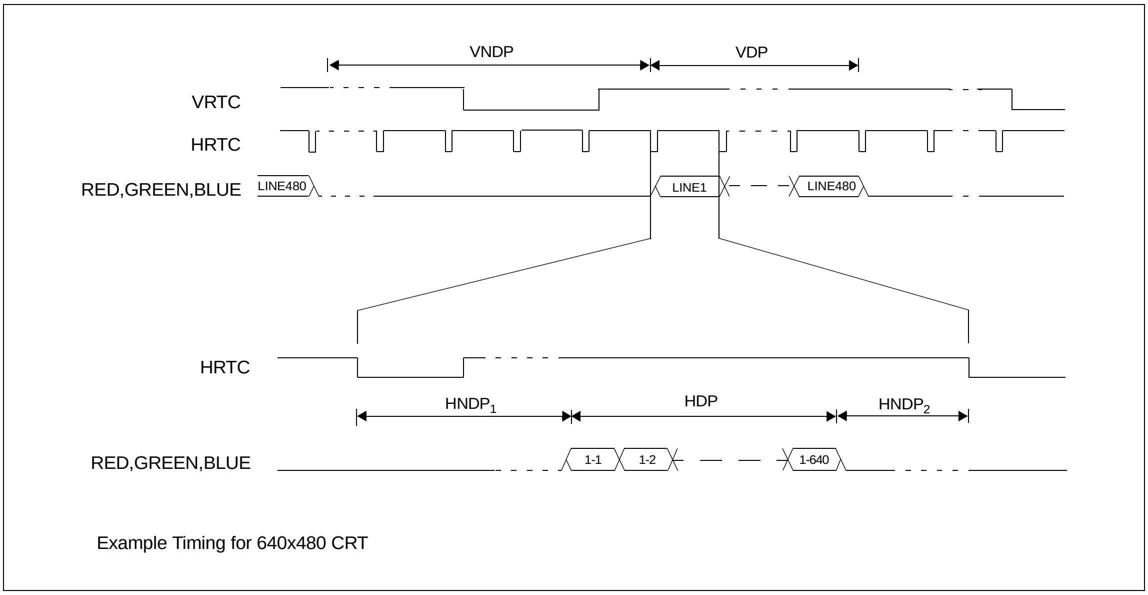


Figure 6-34 CRT Timing

VDP	= Vertical Display Period	= (REG[057h] bits [1:0], REG[056h] bits [7:0]) + 1
VNDP	= Vertical Non-Display Period	= (REG[058h] bits [6:0]) + 1
HDP	= Horizontal Display Period	= ((REG[050h] bits [6:0]) + 1) × 8Ts
HNDP	= Horizontal Non-Display Period	= HNDP$_1$ + HNDP$_2$ = ((REG[052h] bits [5:0]) + 1) × 8Ts
HNDP$_2$	= HRTC Start Position	= (REG[053h] bits [5:0]) × 8 + 4Ts for 4/8 bpp = (REG[053h] bits [5:0]) × 8 + 5Ts for 16 bpp

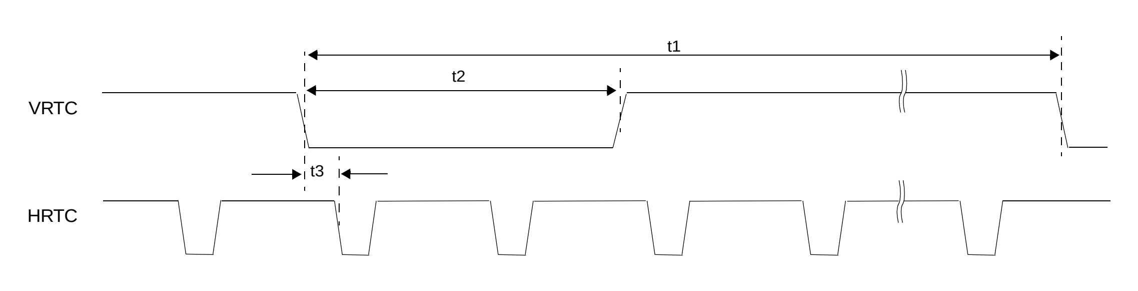


Figure 6-35 CRT A.C. Timing

Table 6-28 CRT A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	VRTC cycle time		note 1		lines
t2	VRTC pulse width low		note 2		lines
t3	VRTC falling edge to FPLINE falling edge phase difference		note 3		Ts

1. t1 = [((REG[057h] bits 1:0, REG[056h] bits 7:0) + 1) + ((REG[058h] bits 6:0) + 1)]
2. t2 = [((REG[05Ah] bits 2:0) + 1)]
3. t3 = [((REG[053h] bits 4:0) + 1) × 8]

6.5 TV Timing

6.5.1 TV Output Timing

The overall NTSC and PAL video timing is shown in Figure 6-36 and Figure 6-37 respectively.

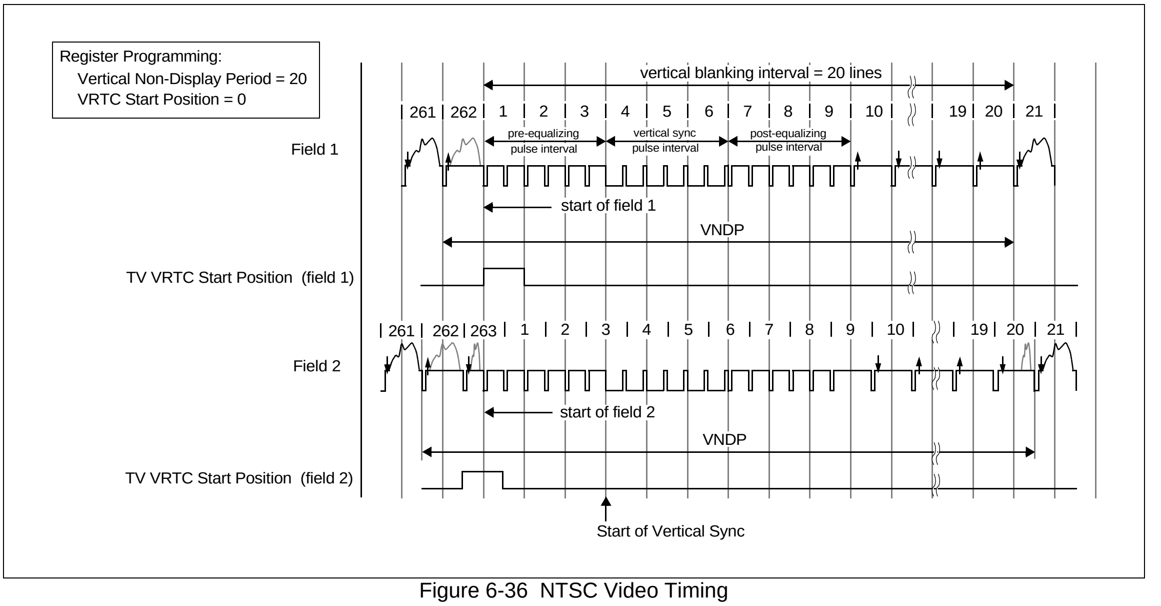

Register Programming:
Vertical Non-Display Period = 20
VRTC Start Position = 0
vertical blanking interval = 20 lines
261
262
1
2
3
4
5
6
7
8
9
10
19
20
21
pre-equalizing pulse interval
vertical sync pulse interval
post-equalizing pulse interval
Field 1
start of field 1
VNDP
TV VRTC Start Position (field 1)
261
262
263
1
2
3
4
5
6
7
8
9
10
19
20
21
Field 2
start of field 2
VNDP
TV VRTC Start Position (field 2)
Start of Vertical Sync

Figure 6-36 NTSC Video Timing

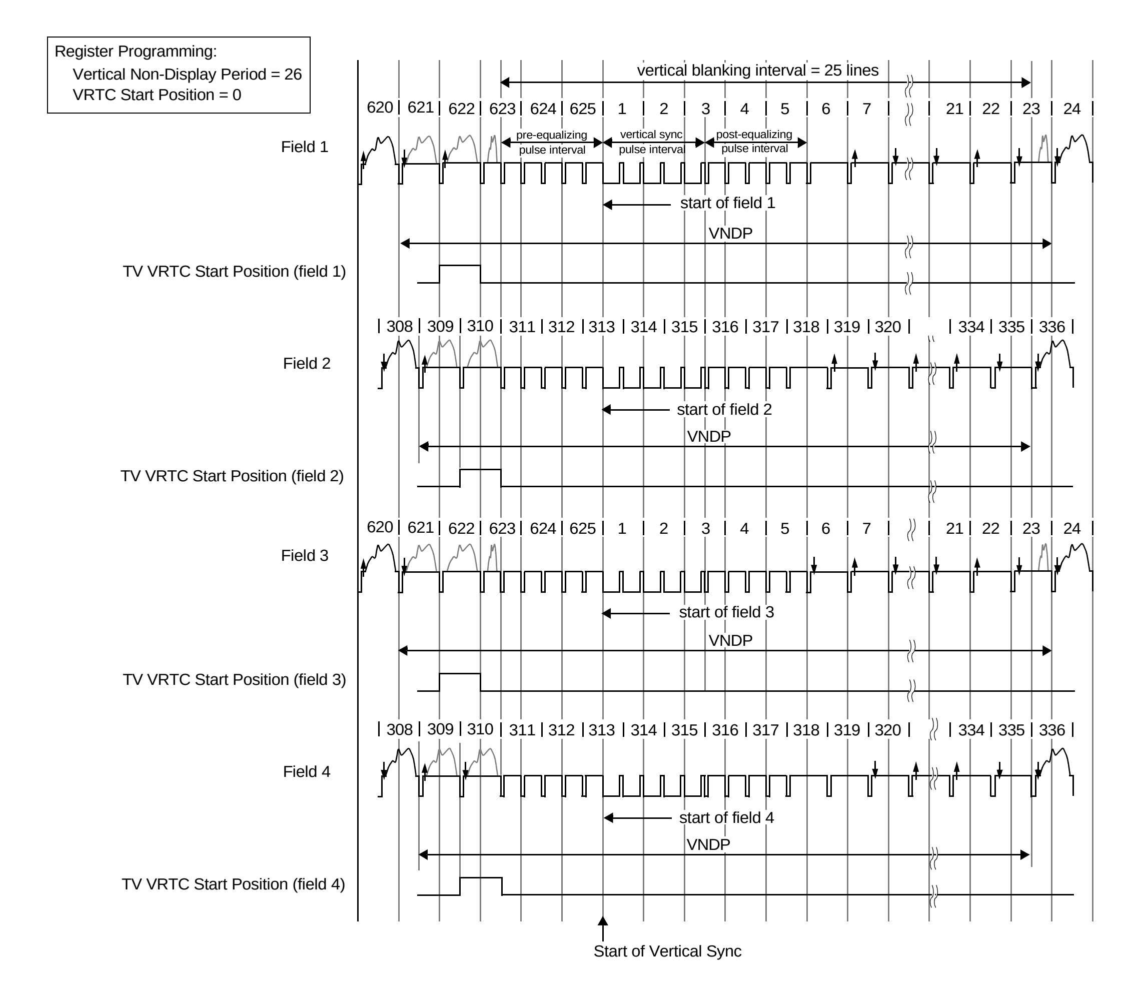


Figure 6-37 PAL Video Timing

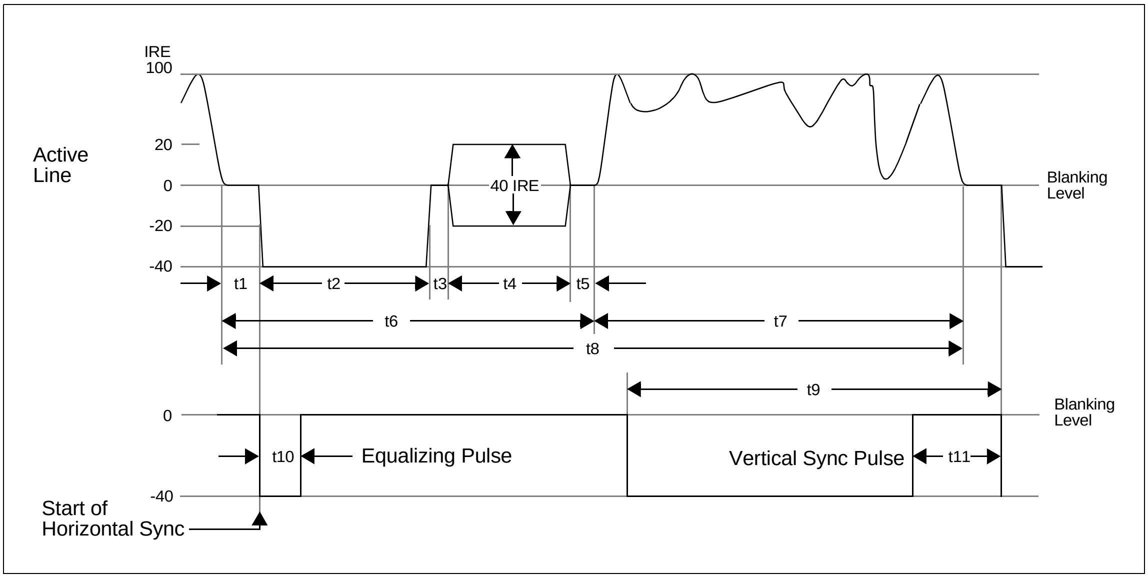


Figure 6-38 Horizontal Timing for NTSC/PAL

Table 6-29 Horizontal Timing for NTSC/PAL

Symbol	Parameter	NTSC	PAL	Units
T_{4SC}	(4x Subcarrier clock) period	69.841	56.387	ns
t1	Front Porch	note 1	note 1	T_{4SC}
t2	Horizontal Sync	67	83	T_{4SC}
t3	Breezeway	9	16	T_{4SC}
t4	Color Burst	39	44	T_{4SC}
t5	Color Back Porch	note 2	note 3	T_{4SC}
t6	Horizontal Blanking	note 4	note 5	T_{4SC}
t7	Active Video	note 6	note 6	T_{4SC}
t8	Line Period	910	1135	T_{4SC}
t9	Half Line Period	455	568 / 567	T_{4SC}
t10	Equalizing Pulse	33	41	T_{4SC}
t11	Vertical Serration	67	83	T_{4SC}

1. t1 = ((REG[053] bits[5:0]) + 1) × 8 - 6 (4bpp, 8bpp modes)
   = ((REG[053] bits[5:0]) + 1) × 8 - 5 (16bpp mode)
2. $t5_{NTSC}$ = (((REG[052] bits[5:0]) × 8) + 6) - (((REG[053] bits[5:0]) + 1) × 8) - 109 (4bpp, 8bpp modes)
   = (((REG[052] bits[5:0]) × 8) + 6) - (((REG[053] bits[5:0]) + 1) × 8) - 110 (16bpp mode)
3. $t5_{PAL}$ = (((REG[052] bits[5:0]) × 8) + 7) - (((REG[053] bits[5:0]) + 1) × 8) - 137 (4bpp, 8bpp modes)
   = (((REG[052] bits[5:0]) × 8) + 7) - (((REG[053] bits[5:0]) + 1) × 8) - 138 (16bpp mode)
4. $t6_{NTSC}$ = ((REG[052] bits[5:0]) × 8) + 6
5. $t6_{PAL}$ = ((REG[052] bits[5:0]) × 8) + 7
6. t7 = ((REG[050] bits[6:0]) + 1) × 8

Important:
REG[050] and REG[052] must be programmed to satisfy the Line Period (t8).
For NTSC, (((REG[050] bits[6:0]) + 1) × 8) + (((REG[052] bits[5:0]) × 8) + 6) = 910.
For PAL, (((REG[050] bits[6:0]) + 1) × 8) + (((REG[052] bits[5:0]) × 8) + 7) = 1135.

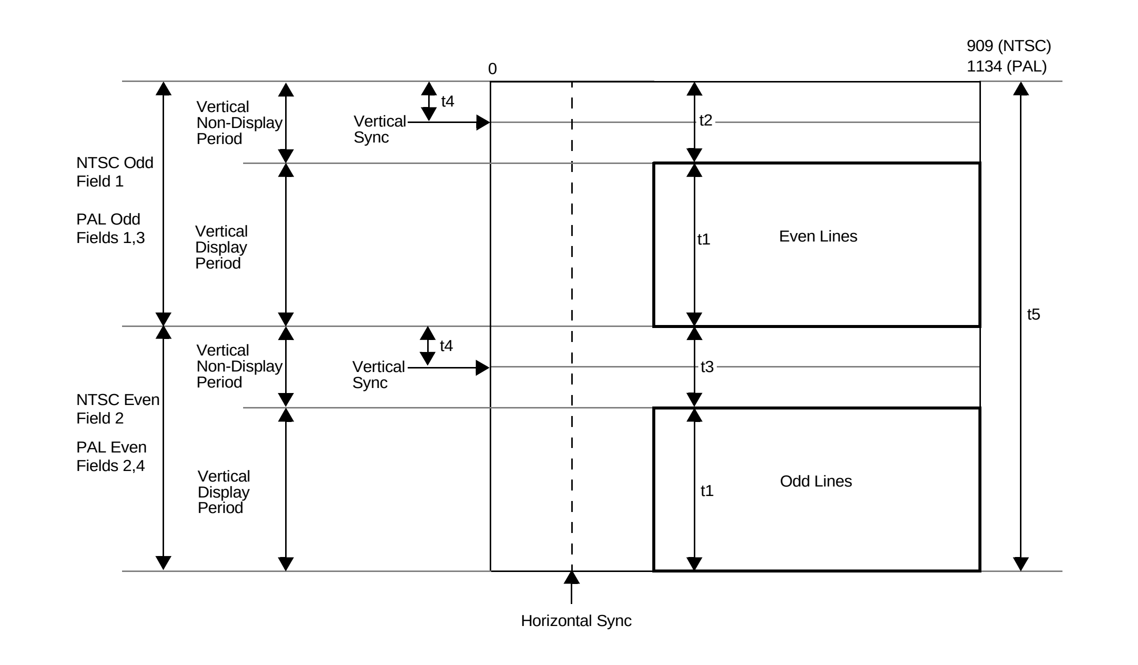


Figure 6-39 Vertical Timing for NTSC/PAL

Table 6-30 Vertical Timing for NTSC/PAL

Symbol	Parameter	NTSC	PAL	Units
T_{LINE}	Line Period	63.55556	63.99964	us
t1	Vertical Field Period	note 1	note 1	T_{LINE}
t2	Vertical Even Blanking	note 2	note 2	T_{LINE}
t3	Vertical Odd Blanking	note 3	note 3	T_{LINE}
t4	Vertical Sync Position	note 4	note 5	T_{LINE}
t5	Frame Period	525	625	T_{LINE}

1. t1 = ({(REG[057h] bits[1:0]), (REG[056] bits[7:0])} + 1) ÷ 2 (rounded up)
2. t2 = ((REG[058h] bits[6:0]) + 1) for NTSC field 1
   = ((REG[058h] bits [6:0]) + 2) for PAL fields 1 and 3
3. t3 = ((REG[058h] bits[6:0]) + 2) for NTSC field 2
   = ((REG[058h] bits [6:0]) + 1) for PAL fields 2 and 4
4. $t4_{NTSC}$ = ((REG[059h] bits[6:0]) + 4) for field 1
   = ((REG[059h] bits[6:0]) + 4.5) for field 2
5. $t4_{PAL}$ = ((REG[059h] bits[6:0]) + 5) for field 1 and field 3
   = ((REG[059h] bits[6:0]) + 4.5) for field 2 and field 4

Important

REG[056], REG[057], and REG[058] must be programmed to satisfy the Frame Period (t5).
For NTSC, ({(REG[057] bits[1:0]), (REG[056] bits[7:0])} + 1) + ((REG[058] bits[6:0]) + 1) × 2 + 1) = 525
For PAL, ({(REG[057] bits[1:0]), (REG[056] bits[7:0])} + 1) + ((REG[058] bits[6:0]) + 1) × 2 + 1) = 625.

6.6 MediaPlug Interface Timing

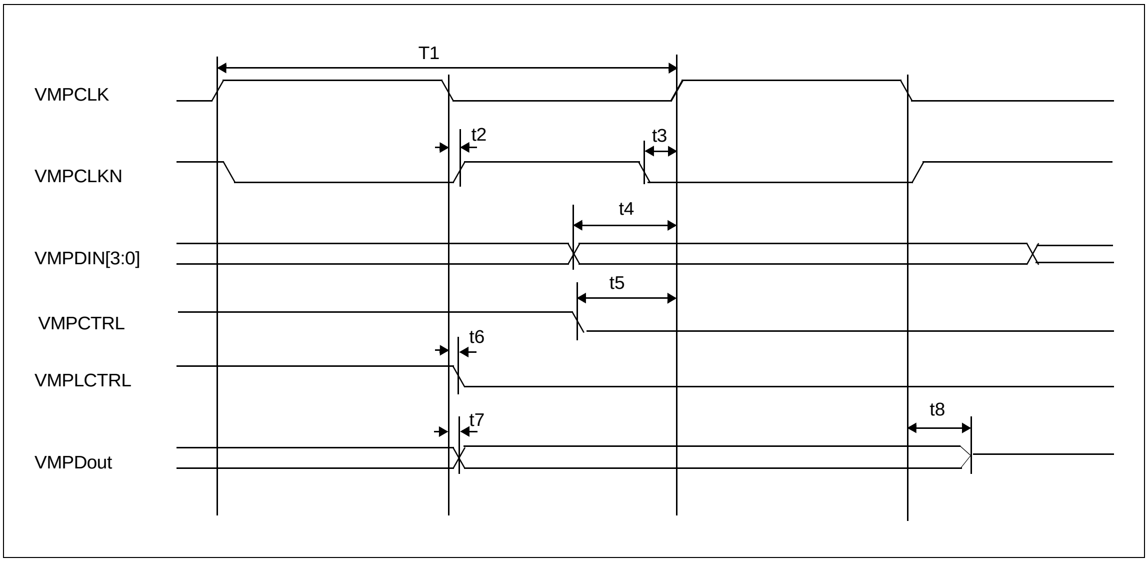


Figure 6-40 MediaPlug A.C. Timing

Note: The above timing diagram assumes no load.

Table 6-31 MediaPlug A.C. Timing

Symbol	Parameter	Min	Max	Units
T1	VMPCLK clock period	50		ns
t2	VMPCLK falling edge to VMPCLKN rising edge skew	0.1	0.6	ns
t3	VMPCLKN falling edge to VMPCLK rising edge skew	.3	1.1	ns
t4	Input data setup	17		ns
t5	VMPCTRL setup	16		ns
t6	Local control signal delay from VMPCLK falling edge	0	1.1	ns
t7	Output data delay from VMPCLK falling edge	0	1.1	ns
t8	Output data tristate delay from VMPCLK falling edge	0.4	1.4	ns

Note: VMPCLK, VMPCLKN are twice the period of the MediaPlug Clock. See Section 7, “Clocks” on page 75.

7 CLOCKS

7.1 Clock Overview

The following diagram provides a logical representation of the S1D13806 internal clocks.

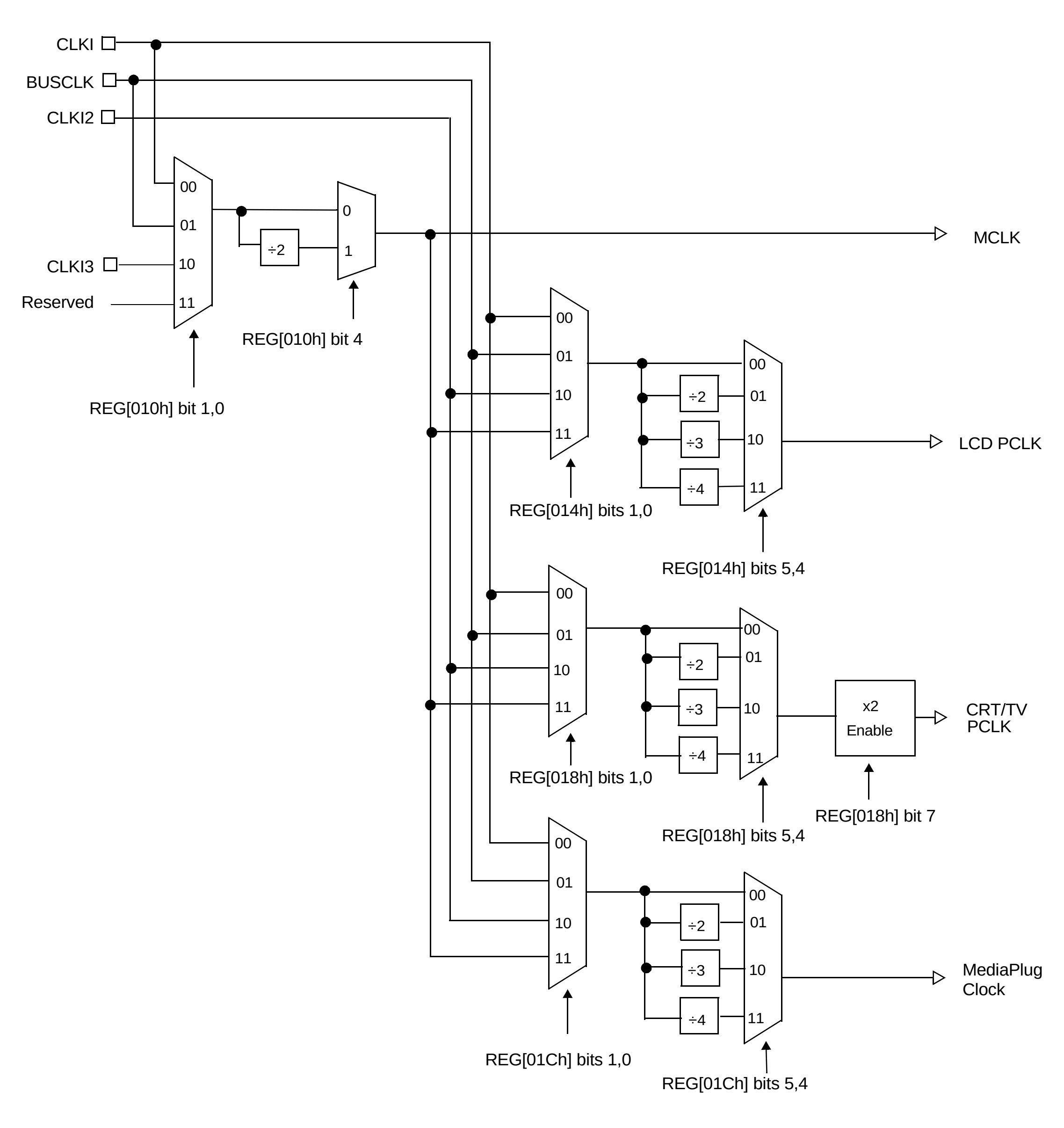


Figure 7-1 Clock Overview Diagram

7.2 Clock Descriptions

7.2.1 MCLK

MCLK should be configured as close to its maximum (50MHz) as possible. The S1D13806 contains sophisticated clock management, therefore, very little power is saved by reducing the MCLK frequency.

The frequency of MCLK is directly proportional to the bandwidth of the video memory. The bandwidth available to the CPU (for screen updates) is that left over after screen refresh takes its share. CPU bandwidth can be seriously reduced when the MCLK frequency is reduced, especially for high-resolution, high-color modes where screen refresh has high bandwidth requirements.

7.2.2 LCD PCLK

LCD PCLK should be chosen to match the optimum frame rate of the panel. See Section 18, “Clocking” on page 169 for details on the relationship between PCLK and frame rate, and for the maximum supportable PCLK frequencies for any given video mode.

Some flexibility is possible in the selection of PCLK. Firstly, panels typically have a range of permissible frame rates. Secondly, it may be possible to choose a higher PCLK frequency and tailor the horizontal non-display period (see REG[052h]) to bring down the frame-rate to its optimal value.

7.2.3 CRT/TV PCLK

TVs and older CRTs usually have very precise frequency requirements, so it may be necessary to dedicate one of the clock inputs to this function. More recent CRTs work within a range of frequencies, so it may be possible to support them with the BUSCLK or MCLK.

7.2.4 MediaPlug Clock

The MediaPlug Clock frequency is internally divided by 2 to provide the output signals VMPCLK and VMPCLKN for the MediaPlug interface. VMPCLK requires a clock in the range of 6-8MHz, therefore the MediaPlug Clock must be in the range of 12-16MHz. For AC timing see Section 6.6, “MediaPlug Interface Timing” on page 74.

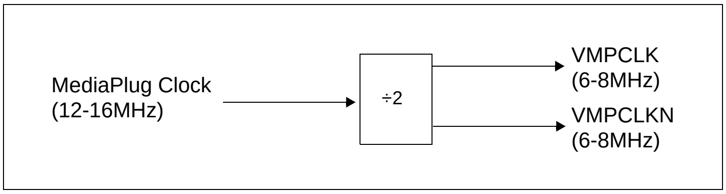


Figure 7-2 MediaPlug Clock Output Signals

7.3 Clock Selection

Table 7-1 Clock Selection

	Clock Source Options	Internal Clocks			
		MCLK	LCD PCLK	CRT/TV PCLK[1]	MediaPlug Clock
External Clocks	CLKI	REG[010h] = 00h	REG[014h] = 00h	REG[018h] = 00h	REG[01Ch] = 00h
	CLKI ÷ 2	REG[010h] = 10h	REG[014h] = 10h	REG[018h] = 10h	REG[01Ch] = 10h
	CLKI ÷ 3	—	REG[014h] = 20h	REG[018h] = 20h	REG[01Ch] = 20h
	CLKI ÷ 4	—	REG[014h] = 30h	REG[018h] = 30h	REG[01Ch] = 30h
	CLKI2	—	REG[014h] = 02h	REG[018h] = 02h	REG[01Ch] = 02h
	CLKI2 ÷ 2	—	REG[014h] = 12h	REG[018h] = 12h	REG[01Ch] = 12h
	CLKI2 ÷ 3	—	REG[014h] = 22h	REG[018h] = 22h	REG[01Ch] = 22h
	CLKI2 ÷ 4	—	REG[014h] = 32h	REG[018h] = 32h	REG[01Ch] = 32h
	CLKI3	REG[010h] = 02h	—	—	—
	CLKI3 ÷ 2	REG[010h] = 12h	—	—	—
	BUSCLK	REG[010h] = 01h	REG[014h] = 01h	REG[018h] = 01h	REG[01Ch] = 01h
	BUSCLK ÷ 2	REG[010h] = 11h	REG[014h] = 11h	REG[018h] = 11h	REG[01Ch] = 11h
	BUSCLK ÷ 3	—	REG[014h] = 21h	REG[018h] = 21h	REG[01Ch] = 21h
	BUSCLK ÷ 4	—	REG[014h] = 31h	REG[018h] = 31h	REG[01Ch] = 31h
	MCLK[2]	—	REG[014h] = 03h	REG[018h] = 03h	REG[01Ch] = 03h
	MCLK ÷ 2[2]	—	REG[014h] = 13h	REG[018h] = 13h	REG[01Ch] = 13h
	MCLK ÷ 3[2]	—	REG[014h] = 23h	REG[018h] = 23h	REG[01Ch] = 23h
	MCLK ÷ 4[2]	—	REG[014h] = 33h	REG[018h] = 33h	REG[01Ch] = 33h

Note: 1. The CRT/TV pixel clock may be further multiplied by 2 when TV with Flicker Filter is enabled using REG[018h] bit 7.

2. MCLK may be a previously divided down version of CLKI, CLKI3, or BUSCLK.

7.4 Clocks vs. Functions

The S1D13806 has five clock signals. Not all clock signals must be active for certain functions to be carried out. The following table shows which clocks are required for each function.

Table 7-2 Clocks vs. Functions

Function	Required Clocks				
	BUSCLK	LCD PCLK	CRT/TV PCLK	MCLK	MediaPlug Clock
Register read/write	Yes	No	No	No	No
LCD LUT read/write	Yes	Yes	—	—	—
CRT/TV LUT read/ write	Yes	—	Yes	—	—
Memory read/write	Yes	—	—	Yes	—
2D Operation	Yes	—	—	Yes	—
MediaPlug Registers read/write	Yes	—	—	—	Yes
Power Save Mode	see Section 19, “Power Save Mode” on page 181				

Note: The S1D13806 contains sophisticated power management that dynamically shuts down clocks when not required.

8 REGISTERS

This section discusses how and where to access the S1D13806 registers. It also provides detailed information about the layout and usage of each register.

8.1 Initializing the S1D13806

Before programming the S1D13806 registers, the following bits must be set.

- Register/Memory Select bit (REG[001h] bit 7)
- SDRAM Initialization bit (REG[020h] bit 7)

8.1.1 Register Memory Select Bit

At reset, the Register/Memory Select bit is set to 1. This means that only REG[000h] (read-only) and REG[001h] are accessible **until a write to REG[001h] sets bit 7 to 0 making all registers and memory accessible**. When debugging a new hardware design, this can sometimes give the appearance that the interface is not working, so it is important to remember to clear this bit before proceeding with debugging.

8.1.2 SDRAM Initialization Bit

To initialize the embedded SDRAM in the S1D13806, this bit must be set to 1 a minimum of 200µs after reset. **This bit must be set to 1 before memory access is performed.**

8.2 Register Mapping

The S1D13806 registers are memory-mapped. When the system decodes the input pins as CS# = 0 and M/R# = 0, the registers may be accessed. The register space is decoded by A[20:0].

When A20 = 1 the BitBLT data register ports are decoded allowing the system to access the display buffer through the 2D BitBLT engine using address lines A[19:0]. When A20 = 0 and A12 = 0 the registers are decoded using A[8:0] as an index. When A20 = 0 and A12 = 1 the MediaPlug register ports are decoded using A[11:0].

The MediaPlug register ports are defined only when the configuration input CONF7 = 1 on reset. When CONF7 = 0 on reset, A12 is always treated as 0 and the MediaPlug register space is not available - see Table 4-8, "Summary of Power-On/Reset Options," on page 17.

Table 8-1, "Register Mapping with CS# = 0 and M/R# = 0" shows the decoding for each register type.

Table 8-1 Register Mapping with CS# = 0 and M/R# = 0

Register Types (Range)	Address A20-A0 Decoding
BitBLT data registers (1M byte)	10 0000 to 1F FFFFh
MediaPlug registers (4K bytes)	1000h to 1FFFh
On-chip registers (512 bytes)	0 to 1FFh

Note: The registers may be aliased within the allocated register space. If aliasing is undesirable, the register space must be fully decoded.

8.3 Register Set

The S1D13806 register set is as follows.

Table 8-2 S1D13806 Registers

Register	Pg	Register	Pg
Basic Registers		REG[046h] LCD Memory Address Offset Register 0	99
REG[000h] Revision Code Register	83	REG[047h] LCD Memory Address Offset Register 1	99
REG[001h] Miscellaneous Register	83	REG[048h] LCD Pixel Panning Register	100
General IO Pins Registers		REG[04Ah] LCD Display FIFO High Threshold Control Register	100
REG[004h] General IO Pins Configuration Register 0	84	REG[04Bh] LCD Display FIFO Low Threshold Control Register	101
REG[005h] General IO Pins Configuration Register 1	84	**CRT/TV Configuration Registers**	
REG[008h] General IO Pins Control Register 0	84	REG[050h] CRT/TV Horizontal Display Width Register	101
REG[009h] General IO Pins Control Register 1	85	REG[052h] CRT/TV Horizontal Non-Display Period Register	101
Configuration Readback Register		REG[053h] CRT/TV HRTC Start Position Register	102
REG[00Ch] Configuration Status Register	85	REG[054h] CRT/TV HRTC Pulse Width Register	102
Clock Configuration Registers		REG[056h] CRT/TV Vertical Display Height Register 0	103
REG[010h] Memory Clock Configuration Register	85	REG[057h] CRT/TV Vertical Display Height Register 1	103
REG[014h] LCD Pixel Clock Configuration Register	86	REG[058h] CRT/TV Vertical Non-Display Period Register	103
REG[018h] CRT/TV Pixel Clock Configuration Register	87	REG[059h] CRT/TV VRTC Start Position Register	104
REG[01Ch] MediaPlug Clock Configuration Register	88	REG[05Ah] CRT VRTC Pulse Width Register	104
REG[01Eh] CPU To memory Wait State Select Register	89	REG[05Bh] TV Output Control Register	105
Memory Configuration Registers		**CRT/TV Display Mode Registers**	
REG[020h] Memory Configuration Register	90	REG[060h] CRT/TV Display Mode Register	106
REG[021h] SDRAM Refresh Rate Register	90	REG[062h] CRT/TV Display Start Address Register 0	107
REG[02Ah] SDRAM Timing Control Register 0	91	REG[063h] CRT/TV Display Start Address Register 1	107
REG[02Bh] SDRAM Timing Control Register 1	91	REG[064h] CRT/TV Display Start Address Register 2	107
Panel Configuration Registers		REG[066h] CRT/TV Memory Address Offset Register 0	107
REG[030h] Panel Type Register	91	REG[067h] CRT/TV Memory Address Offset Register 1	107
REG[031h] MOD Rate Register	92	REG[068h] CRT/TV Pixel Panning Register	108
REG[032h] LCD Horizontal Display Width Register	93	REG[06Ah] CRT/TV FIFO High Threshold Control Register	108
REG[034h] LCD Horizontal Non-Display Period Register	93	REG[06Bh] CRT/TV FIFO Low Threshold Control Register	109
REG[035h] TFT FPLINE Start Position Register	94	**LCD Ink/Cursor Registers**	
REG[036h] TFT FPLINE Pulse Width Register	94	REG[070h] LCD Ink/Cursor Control Register	109
REG[038h] LCD Vertical Display Height Register 0	95	REG[071h] LCD Ink/Cursor Start Address Register	110
REG[039h] LCD Vertical Display Height Register 1	95	REG[072h] LCD Cursor X Position Register 0	110
REG[03Ah] LCD Vertical Non-Display Period Register	95	REG[073h] LCD Cursor X Position Register 1	110
REG[03Bh] TFT FPFRAME Start Position Register	96	REG[074h] LCD Cursor Y Position Register 0	111
REG[03Ch] TFT FPFRAME Pulse Width Register	96	REG[075h] LCD Cursor Y Position Register 1	111
LCD Display Mode Registers		REG[076h] LCD Ink/Cursor Blue Color 0 Register	111
REG[040h] LCD Display Mode Register	97	REG[077h] LCD Ink/Cursor Green Color 0 Register	112
REG[041h] LCD Miscellaneous Register	98	REG[078h] LCD Ink/Cursor Red Color 0 Register	112
REG[042h] LCD Display Start Address Register 0	99	REG[07Ah] LCD Ink/Cursor Blue Color 1 Register	112
REG[043h] LCD Display Start Address Register 1	99	REG[07Bh] LCD Ink/Cursor Green Color 1 Register	112
REG[044h] LCD Display Start Address Register 2	99	REG[07Ch] LCD Ink/Cursor Red Color 1 Register	112

Table 8-2 S1D13806 Registers (Continued)

Register	Pg
REG[07Eh] LCD Ink/Cursor FIFO High Threshold Register	113
CRT/TV Ink/Cursor Registers	
REG[080h] CRT/TV Ink/Cursor Control Register	113
REG[081h] CRT/TV Ink/Cursor Start Address Register	114
REG[082h] CRT/TV Cursor X Position Register 0	115
REG[083h] CRT/TV Cursor X Position Register 1	115
REG[084h] CRT/TV Cursor Y Position Register 0	115
REG[085h] CRT/TV Cursor Y Position Register 1	115
REG[086h] CRT/TV Ink/Cursor Blue Color 0 Register	116
REG[087h] CRT/TV Ink/Cursor Green Color 0 Register	116
REG[088h] CRT/TV Ink/Cursor Red Color 0 Register	116
REG[08Ah] CRT/TV Ink/Cursor Blue Color 1 Register	116
REG[08Bh] CRT/TV Ink/Cursor Green Color 1 Register	117
REG[08Ch] CRT/TV Ink/Cursor Red Color 1 Register	117
REG[08Eh] CRT/TV Ink/Cursor FIFO High Threshold Register	117
BitBLT Configuration Registers	
REG[100h] BitBLT Control Register 0	118
REG[101h] BitBLT Control Register 1	119
REG[102h] BitBLT ROP Code/Color Expansion Register	120
REG[103h] BitBLT Operation Register	121
REG[104h] BitBLT Source Start Address Register 0	122
REG[105h] BitBLT Source Start Address Register 1	122
REG[106h] BitBLT Source Start Address Register 2	122
REG[108h] BitBLT Destination Start Address Register 0	123
REG[109h] BitBLT Destination Start Address Register 1	123
REG[10Ah] BitBLT Destination Start Address Register 2	123
REG[10Ch] BitBLT Memory Address Offset Register 0	123
REG[10Dh] BitBLT Memory Address Offset Register 1	123
REG[110h] BitBLT Width Register 0	124
REG[111h] BitBLT Width Register 1	124
REG[112h] BitBLT Height Register 0	124
REG[113h] BitBLT Height Register 1	124
REG[114h] BitBLT Background Color Register 0	125
REG[115h] BitBLT Background Color Register 1	125
REG[118h] BitBLT Foreground Color Register 0	125
REG[119h] BitBLT Foreground Color Register 1	125
Look-Up Table Registers	
REG[1E0h] Look-Up Table Mode Register	126
REG[1E2h] Look-Up Table Address Register	126
REG[1E4h] Look-Up Table Data Register	127
Power Save Configuration Registers	
REG[1F0h] Power Save Configuration Register	127
REG[1F1h] Power Save Status Register	128
Miscellaneous Register	
REG[1F4h] CPU-To-Memory Access Watchdog Timer Register	129
Common Display Mode Register	
REG[1FCh] Display Mode Register	130
MediaPlug Control Registers	
REG[1000h] MediaPlug LCMD Register	131
REG[1002h] MediaPlug Reserved LCMD Register	133
REG[1004h] MediaPlug CMD Register	133
REG[1006h] MediaPlug Reserved CMD Register	134
MediaPlug Data Registers	
REG[1008h] to REG[1FFEh] MediaPlug Data Register	135
BitBLT Data Registers	
REG[100000h] to REG[1FFFFEh] BitBLT Data Register 0	135

8.4 Register Descriptions

8.4.1 Basic Registers

Revision Code Register REG[000h]							RO
Product Code Bit 5	Product Code Bit 4	Product Code Bit 3	Product Code Bit 2	Product Code Bit 1	Product Code Bit 0	Revision Code Bit 1	Revision Code Bit 0

bits 7-2 Product Code Bits [5:0]
This read-only register indicates the product code of the controller. The product code for S1D13806 is 000111b.

bits 1-0 Revision Code Bits [1:0]
This read-only register indicates the revision code of the controller. The revision code is 00b.

Miscellaneous Register REG[001h]							RW
Register/ Memory Select	n/a	n/a	n/a	n/a	Reserved	Reserved	Reserved

bit 7 Register/Memory Select Bit
At reset, the Register/Memory Select bit is set to 1. This means that only REG[000h] (read-only) and REG[001h] are accessible **until a write to REG[001h] sets bit 7 to 0 making all registers and memory accessible**. When debugging a new hardware design, this can sometimes give the appearance that the interface is not working, so it is important to remember to clear this bit before proceeding with debugging.

bit 2 Reserved.
This bit must be set to 0.

bit 1 Reserved.
This bit must be set to 0.

bit 0 Reserved.
This bit must be set to 0.

8.4.2 General IO Pins Registers

General IO Pins Configuration Register 0 REG[004h]							RW
GPIO7 Pin IO Config.	GPIO6 Pin IO Config.	GPIO5 Pin IO Config.	GPIO4 Pin IO Config.	GPIO3 Pin IO Config.	GPIO2 Pin IO Config.	GPIO1 Pin IO Config.	GPIO0 Pin IO Config.

bit 7-0 GPIO[7:0] Pin IO Configuration Bits
When bit *n* = 1, GPIO[*n*] is configured as an output pin. (where *n* ranges from 0 to 7)
When bit *n* = 0 (default), GPIO[*n*] is configured as an input pin. (where *n* ranges from 0 to 7).

General IO Pins Configuration Register 1 REG[005h]							RW
n/a	n/a	n/a	GPIO12 Pin IO Config.	GPIO11 Pin IO Config.	GPIO10 Pin IO Config.	GPIO9 Pin IO Config.	GPIO8 Pin IO Config.

bit 4-0 GPIO[12:8] Pin IO Configuration Bits
When bit *n* = 1, GPIO[*n+8*] is configured as an output pin. (where *n* ranges from 0 to 4)
When bit *n* = 0 (default), GPIO[*n+8*] is configured as an input pin. (where *n* ranges from 0 to 4)

Note: Note that CONF7 must be properly configured at the rising edge of RESET# to enable GPIO12 as an IO pin, otherwise GPIO12 is used for the Media Plug interface and this register has no effect. The following table shows GPIO12 usage.

Table 8-3 Media Plug/GPIO12 Pin Functionality

Pin	CONF7 on Reset	
	0	1
GPIO12	GPIO12	VMPEPWR

General IO Pins Control Register 0 REG[008h]							RW
GPIO7 Pin IO Status	GPIO6 Pin IO Status	GPIO5 Pin IO Status	GPIO4Pin IO Status	GPIO3 Pin IO Status	GPIO2 Pin IO Status	GPIO1 Pin IO Status	GPIO0 Pin IO Status

bit 7-0 GPIO[7:0] Pin IO Status Bits
When GPIO[*n*] is configured as an output, writing a 1 to bit *n* drives GPIO[*n*] high and writing a 0 to this bit drives GPIO[*n*] low. (*n* ranges from 0 to 7)
When GPIO[*n*] is configured as an input, a read from bit *n* returns the status of GPIO[*n*]. (*n* ranges from 0 to 7)

General IO Pins Control Register 1 REG[009h] RW							
n/a	n/a	n/a	GPIO12 Pin IO Status	GPIO11 Pin IO Status	GPIO10 Pin IO Status	GPIO9 Pin IO Status	GPIO8 Pin IO Status

bit 4-0 GPIO[12:8] Pin IO Status Bits
When GPIO[*n+8*] is configured as an output, writing a 1 to bit *n* drives GPIO[*n+8*] high and writing a 0 to this bit drives GPIO[*n+8*] low. (*n* ranges from 0 to 4)
When GPIO[*n+8*] is configured as an input, a read from bit *n+8* returns the status of GPIO[*n+8*]. (*n* ranges from 0 to 4)

Note: CONF7 must be properly configured at the rising edge of RESET# to enable GPIO12 as an IO pin, otherwise GPIO12 is used for the Media Plug interface and this register has no effect on GPIO12. See Table 8-3, “Media Plug/GPIO12 Pin Functionality” for GPIO12 usage.

8.4.3 Configuration Readback Register

Configuration Status Register REG[00Ch] RO							
CONF[7] Config. Status	CONF[6] Config. Status	CONF[5] Config. Status	CONF[4] Config. Status	CONF[3] Config. Status	CONF[2] Config. Status	CONF[1] Config. Status	CONF[0] Config. Status

bits 7-0 CONF[7:0] Configuration Status Bits
These read-only bits return the status of CONF[7:0] at the rising edge of RESET#.

8.4.4 Clock Configuration Registers

Memory Clock Configuration Register REG[010h] RW							
n/a	n/a	n/a	MCLK Divide Select	n/a	n/a	MCLK Source Select Bit 1	MCLK Source Select Bit 0

Note: For further information on MCLK, see Section 7.2, “Clock Descriptions” on page 76.

bit 4 MCLK Divide Select
When this bit = 1, the internal memory clock (MCLK) frequency is half of the MCLK source frequency.
When this bit = 0, the memory clock frequency is equal to the MCLK source frequency.

Note: The MCLK frequency should always be set to the maximum frequency allowed by the SDRAM. This provides maximum performance and minimizes overall system power
consumption.

bit 1-0 MCLK Source Select Bits [1:0]
These bits determine the source of the internal memory clock (MCLK).

Table 8-4 MCLK Source Select

MCLK Source Select Bits	MCLK Source
00	CLKI
01	BUSCLK
10	CLKI3
11	Reserved

Note: The MCLK Divide Select bit must be set to 1 before changing the MCLK Source Select bits.

LCD Pixel Clock Configuration Register REG[014h]							RW
n/a	n/a	LCD PCLK Divide Select Bit 1	LCD PCLK Divide Select Bit 0	n/a	n/a	LCD PCLK Source Select Bit 1	LCD PCLK Source Select Bit 0

Note: For further information on the LCD PCLK, refer to Section 7.2, “Clock Descriptions” on page 76.

bits 5-4 LCD PCLK Divide Select Bits [1:0]
These bits determine the divide used to generate the LCD pixel clock from the LCD pixel clock source.

Table 8-5 LCD PCLK Divide Selection

LCD PCLK Divide Select Bits	LCD PCLK Source to LPCLK Frequency Ratio
00	1:1
01	2:1
10	3:1
11	4:1

bits 1-0 LCD PCLK Source Select Bits [1:0]
These bits determine the source of the pixel clock for the LCD display.

Table 8-6 LCD PCLK Source Selection

LCD PCLK Source Select Bits	LCD PCLK Source
00	CLKI
01	BUSCLK
10	CLKI2
11	MCLK

Note: MCLK may be a previously divided down version of CLKI, CLKI3, or BUSCLK.

CRT/TV Pixel Clock Configuration Register
REG[018h] RW

Flicker Filter Clock Enable	n/a	CRT/TV PCLK Divide Select Bit 1	CRT/TV PCLK Divide Select Bit 0	n/a	n/a	CRT/TV PCLK Source Select Bit 1	CRT/TV PCLK Source Select Bit 0

Note: For further information on the CRT/TV PCLK, refer to Section 7.2, “Clock Descriptions” on page 76.

bit 7 Flicker Filter Clock Enable
This bit must be set to 1 when TV with flicker filter is enabled. For details on TV with flicker filter, see REG[1FCh] bits 2-0.

bits 5-4 CRT/TV PCLK Divide Select Bits[1:0]
These bits determine the divide used to generate the CRT/TV pixel clock from the CRT/TV pixel clock source.

Table 8-7 CRT/TV PCLK Divide Selection

CRT/TV PCLK Divide Select Bits	CRT/TV PCLK Source to DPCLK Frequency Ratio
00	1:1
01	2:1
10	3:1
11	4:1

bits 1-0 CRT/TV PCLK Source Select Bits [1:0]
These bits determine the source of the pixel clock for the CRT/TV display.

Table 8-8 CRT/TV PCLK Source Selection

CRT/TV PCLK Source Select Bits	CRT/TV PCLK Source
00	CLKI
01	BUSCLK
10	CLKI2
11	MCLK

Note: MCLK may be a previously divided down version of CLKI, CLKI3, or BUSCLK.

MediaPlug Clock Configuration Register REG[01Ch]							RW
n/a	n/a	MediaPlug Clock Divide Select Bit 1	MediaPlug Clock Divide Select Bit 0	n/a	n/a	MediaPlug Clock Source Select Bit 1	MediaPlug Clock Source Select Bit 0

Note: For further information on the MediaPlug Clock, refer to Section 7.2, “Clock Descriptions” on page 76.

bits 5-4 MediaPlug Clock Divide Select Bits [1:0]
These bits determine the divide used to generate the MediaPlug Clock from the MediaPlug Clock source.

Table 8-9 MediaPlug Clock Divide Selection

MediaPlug Clock Divide Select Bits	MediaPlug Clock Source to MediaPlug Clock Frequency Ratio
00	1:1
01	2:1
10	3:1
11	4:1

bits 1-0 MediaPlug Clock Source Select Bits [1:0]
These bits determine the source of the MediaPlug Clock for the MediaPlug Interface.
See Section 6.6, “MediaPlug Interface Timing” on page 74 for AC Timing.

Table 8-10 MediaPlug Clock Source Selection

MediaPlug Clock Source Select Bits	MediaPlug Clock Source
00	CLKI
01	BUSCLK
10	CLKI2
11	MCLK

Note: MCLK may be a previously divided down version of CLKI, CLKI3, or BUSCLK.

CPU To Memory Wait State Select Register REG[01Eh] RW							
n/a	n/a	n/a	n/a	n/a	n/a	CPU to Memory Wait State Select Bit 1	CPU to Memory Wait State Select Bit 0

bits 1-0 CPU to Memory Wait State Select Bits [1:0]
These bits are used to optimize the handshaking between the host interface and the memory controller. The bits should be set according to the relationship between BCLK and MCLK (memory clock)

Note: BCLK can be either BUSCLK or BUSCLK ÷ 2 depending on the setting of CONF5 (see Table 4-8, “Summary of Power-On/Reset Options,” on page 17).

Failure to meet the following conditions may lead to system crash which is recoverable only by RESET.

Table 8-11 Minimum Memory Timing Selection

Wait State Bits [1:0]	Condition
00	no restrictions
01	2 × period (MCLK) - 4ns > period(BCLK)
10	period(MCLK) - 4ns > period(BCLK)
11	Reserved

8.4.5 Memory Configuration Registers

Memory Configuration Register REG[020h]							RW
SDRAM Init	n/a	n/a	n/a	n/a	n/a	n/a	n/a

bit 7 SDRAM Initialization
To initialize the embedded SDRAM, this bit should be set to 1 after Reset. Subsequent toggling of this bit after the first initialization has no effect.

Note: This bit must be set to 1 before memory accesses are performed.

SDRAM Refresh Rate Register REG[021h]							RW
n/a	Reserved	n/a	n/a	n/a	SDRAM Refresh Rate Bit 2	SDRAM Refresh Rate Bit 1	SDRAM Refresh Rate Bit 0

bit 6 Reserved.
This bit must be set to 0.

bits 2-0 SDRAM Refresh Rate Select Bits [2:0]
These bits are set according to the MCLK source frequency (i.e., BUSCLK, CLKI, or CLKI3 as determined by REG[010h] bits 1-0).

Table 8-12 SDRAM Refresh Rate Selection

SDRAM Refresh Rate Bits [2:0]	MCLK Source Frequency (MHz)
000	4.096 <= MClk < 8.192
001	8.192 <= MClk < 16.384
010	16.384 <= MClk < 32.768
011	32.768 <= MClk <= 50.000

SDRAM Timing Control Register 0 REG[02Ah]							RW
SDRAM Timing Control Bit 7	SDRAM Timing Control Bit 6	SDRAM Timing Control Bit 5	SDRAM Timing Control Bit 4	SDRAM Timing Control Bit 3	SDRAM Timing Control Bit 2	SDRAM Timing Control Bit 1	SDRAM Timing Control Bit 0

SDRAM Timing Control Register 1 REG[02Bh]							RW
n/a	n/a	n/a	SDRAM Timing Control Bit 12	SDRAM Timing Control Bit 11	SDRAM Timing Control Bit 10	SDRAM Timing Control Bit 9	SDRAM Timing Control Bit 8

REG[02Ah] bits 7-0 SDRAM Timing Control Bits [12:0]

REG[02Bh] bits 4-0 The SDRAM Timing Control registers must be set according to the frequency of MCLK as follows.

Table 8-13 SDRAM Timings Control Register Settings

MCLK Source Frequency (MHz)	REG[02Ah]	REG[02Bh]
42 ≤ MCLK ≤ 50	00h	01h
33 ≤ MCLK < 42	00h	12h
MCLK < 33	11h	13h

8.4.6 Panel Configuration Registers

Panel Type Register REG[030h]							RW
EL Panel Mode Enable	TFT 2x Data Format Select	Panel Data Width Bit 1	Panel Data Width Bit 0	Panel Data Format Select	Color/Mono. Panel Select	Dual/Single Panel Select	TFT/ Passive LCD Panel Select

bit 7 EL Panel Mode Enable
When this bit = 1, EL Panel support circuit is enabled.
When this bit = 0, there is no hardware effect.
This bit enables the S1D13806 built-in circuit for EL panels which require the Frame Rate Modulation (FRM) to remain static for one frame every 262143 frames (approximately 1 hour at 60Hz refresh). When this bit is enabled, the need for external circuitry to perform the above function is eliminated.

bit 6 TFT 2x Data Format Select
For TFT/D-TFD only.
When this bit = 1, the TFT 2x Data format is selected.
When this bit = 0, the standard TFT Data format is selected.

For details on the TFT 2x Data format, see Section 6.4.10, “TFT/D-TFD Panel Timing” on page 65.

bits 5-4 Panel Data Width Bits [1:0]
These bits select passive LCD/TFT/D-TFD panel data width size.

Table 8-14 Panel Data Width Selection

Panel Data Width Bits [1:0]	Passive LCD Panel Data Width	TFT/D-TFD Panel Data Width	
		1x Data Format	2x Data Format
00	4-bit	9-bit	2 x 9-bit
01	8-bit	12-bit	2 x 12-bit
10	16-bit	18-bit	Reserved
11	Reserved	Reserved	Reserved

bit 3 Panel Data Format Select
When this bit = 1, 8-bit single color passive LCD panel data format 2 is selected. For AC timing see Section 6.4.5, “Single Color 8-Bit Panel Timing (Format 2)” on page 55.
When this bit = 0, 8-bit single color passive LCD panel data format 1 is selected. For AC timing see Section 6.4.4, “Single Color 8-Bit Panel Timing (Format 1)” on page 53.

bit 2 Color/Mono Panel Select
When this bit = 1, color passive LCD panel is selected.
When this bit = 0, monochrome passive LCD panel is selected.

bit 1 Dual/Single Panel Select
When this bit = 1, dual passive LCD panel is selected.
When this bit = 0, single passive LCD panel is selected.

bit 0 TFT/Passive LCD Panel Select
When this bit = 1, TFT/D-TFD panel is selected.
When this bit = 0, passive LCD panel is selected.

MOD Rate Register
REG[031h] RW

n/a	n/a	MOD Rate Bit 5	MOD Rate Bit 4	MOD Rate Bit 3	MOD Rate Bit 2	MOD Rate Bit 1	MOD Rate Bit 0

bits 5-0 MOD Rate Bits [5:0]
For a non-zero value these bits specify the number of FPLINE between toggles of the MOD output signal (DRDY).
When these bits are all 0’s the MOD output signal toggles every FPFRAME. These bits are for passive LCD panels only.

LCD Horizontal Display Width Register REG[032h]							RW
n/a	LCD Horizontal Display Width Bit 6	LCD Horizontal Display Width Bit 5	LCD Horizontal Display Width Bit 4	LCD Horizontal Display Width Bit 3	LCD Horizontal Display Width Bit 2	LCD Horizontal Display Width Bit 1	LCD Horizontal Display Width Bit 0

bits 6-0 LCD Horizontal Display Width Bits [6:0]
These bits specify the LCD panel horizontal display width, in 8 pixel resolution.

Horizontal display width in number of pixels = ((ContentsOfThisRegister)+ 1) × 8

The Horizontal Display Width has certain limitations on the values that may be used for each type of LCD panel. Use of values that do not meet the limitations listed in the following table result in undefined behavior.

Table 8-15 Horizontal Display Width (Pixels)

Panel Type	Horizontal Display Width (Pixels)
Passive Single	must be divisible by 16
Passive Dual	must be divisible by 32
TFT	must be divisible by 8

Note: This register must be programmed such that REG[032h] $\geq$ 3 (32 pixels).

LCD Horizontal Non-Display Period Register REG[034h]							RW
n/a	n/a	n/a	LCD Horizontal Non-Display Period Bit 4	LCD Horizontal Non-Display Period Bit 3	LCD Horizontal Non-Display Period Bit 2	LCD Horizontal Non-Display Period Bit 1	LCD Horizontal Non-Display Period Bit 0

bits 4-0 LCD Horizontal Non-Display Period Bits [4:0]
These bits specify the LCD panel HNDP width in 8 pixel resolution.

HNDP width in number of pixels = ((ContentsOfThisRegister) + 1) × 8

Note: This register must be programmed such that REG[034h] $\geq$ 3 (32 pixels).

Note: For TFT/D-TFD only:
REG[034h] + 1 $\geq$ (REG[035h] + 1) + (REG[036h] bits 3-0 + 1)

TFT FPLINE Start Position Register REG[035h]							RW
n/a	n/a	n/a	TFT FPLINE Start Position Bit 4	TFT FPLINE Start Position Bit 3	TFT FPLINE Start Position Bit 2	TFT FPLINE Start Position Bit 1	TFT FPLINE Start Position Bit 0

bits 4-0 TFT FPLINE Start Position Bits [4:0]
For TFT/D-TFD panels only, these bits specify the delay, in 8 pixel resolution, from the start of the horizontal non-display period to the leading edge of the FPLINE pulse.

For TFT 1x Data Format at 4/8 bpp color depth:
FPLINE start position in number of pixels = [(ContentsOfThisRegister) × 8 + 5]

For TFT 1x Data Format at 16 bpp color depth:
FPLINE start position in number of pixels = [(ContentsOfThisRegister) × 8 + 6]

For TFT 2x Data Format at 4/8 bpp color depth:
FPLINE start position in number of pixels = [(ContentsOfThisRegister) × 8 + 4]

For TFT 2x Data Format at 16 bpp color depth:
FPLINE start position in number of pixels = [(ContentsOfThisRegister) × 8 + 5]

Note: REG[034h] + 1 ≥ (REG[035h] + 1) + (REG[036h] bits 3-0 + 1)

TFT FPLINE Pulse Width Register REG[036h]							RW
LCD FPLINE Polarity Select	n/a	n/a	n/a	TFT FPLINE Pulse Width Bit 3	TFT FPLINE Pulse Width Bit 2	TFT FPLINE Pulse Width Bit 1	TFT FPLINE Pulse Width Bit 0

bit 7 LCD FPLINE Polarity Select
This bit selects the polarity of FPLINE for all LCD panels.
When this bit = 1, the FPLINE pulse is active high for TFT/D-TFD and active low for passive LCD.
When this bit = 0, the FPLINE pulse is active low for TFT/D-TFD and active high for passive LCD.

Table 8-16 LCD FPLINE Polarity Selection

LCD FPLINE Polarity Select	Passive LCD FPLINE Polarity	TFT FPLINE Polarity
0	active high	active low
1	active low	active high

bits 3-0 TFT FPLINE Pulse Width Bits [3:0]
For TFT/D-TFD panels only, these bits specify the pulse width of the FPLINE output signal in 8 pixel resolution.

FPLINE pulse width in number of pixels = ((ContentsOfThisRegister) + 1) × 8

The maximum FPLINE pulse width is 128 pixels.

Note: REG[034h] + 1 ≥ (REG[035h] + 1) + (REG[036h] bits 3-0 + 1)

LCD Vertical Display Height Register 0
REG[038h] RW

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

LCD Vertical Display Height Register 1
REG[039h] RW

n/a	n/a	n/a	n/a	n/a	n/a	LCD Vertical Display Height Bit 9	LCD Vertical Display Height Bit 8

REG[038h] bits 7-0 LCD Vertical Display Height Bits [9:0]

REG[039h] bits 1-0 These bits specify the LCD panel vertical display height, in 1 line resolution.

Vertical display height in number of lines = (ContentsOfThisRegister) + 1

LCD Vertical Non-Display Period Register
REG[03Ah] RW

LCD Vertical Non-Display Period Status (RO)	n/a	LCD Vertical Non-Display Period Bit 5	LCD Vertical Non-Display Period Bit 4	LCD Vertical Non-Display Period Bit 3	LCD Vertical Non-Display Period Bit 2	LCD Vertical Non-Display Period Bit 1	LCD Vertical Non-Display Period Bit 0

bit 7 LCD Vertical Non-Display Period Status
This is a read-only status bit.
When a read from this bit = 1, a LCD panel vertical non-display period is occurring.
When a read from this bit = 0, the LCD panel output is in a vertical display period.

bits 5-0 LCD Vertical Non-Display Period Bits [5:0]
These bits specify the LCD panel vertical non-display period height in 1 line resolution.

Vertical non-display period height in number of lines = (ContentsOfThisRegister) + 1

Note: For TFT/D-TFD only:
(REG[03Ah] bits 5-0 + 1) ≥ (REG[03Bh] + 1) + (REG[03Ch] bits 2-0 + 1)

TFT FPFRAME Start Position Register REG[03Bh] RW							
n/a	n/a	TFT FPFRAME Start Position Bit 5	TFT FPFRAME Start Position Bit 4	TFT FPFRAME Start Position Bit 3	TFT FPFRAME Start Position Bit 2	TFT FPFRAME Start Position Bit 1	TFT FPFRAME Start Position Bit 0

bits 5-0 TFT FPFRAME Start Position Bits [5:0]
For TFT/D-TFD panels only, these bits specify the delay in lines from the start of the vertical non-display period to the leading edge of the FPFRAME pulse.

FPFRAME start position in number of lines = (ContentsOfThisRegister) + 1

Note: (REG[03Ah] bits 5-0 + 1) ≥ (REG[03Bh] + 1) + (REG[03Ch] bits 2-0 + 1)

TFT FPFRAME Pulse Width Register REG[03Ch] RW							
LCD FPFRAME Polarity Select	n/a	n/a	n/a	n/a	TFT FPFRAME Pulse Width Bit 2	TFT FPFRAME Pulse Width Bit 1	TFT FPFRAME Pulse Width Bit 0

bit 7 LCD FPFRAME Polarity Select
This bit selects the polarity of FPFRAME for all LCD panels.
When this bit = 1, the FPFRAME pulse is active high for TFT/D-TFD and active low for passive LCD.
When this bit = 0, the FPFRAME pulse is active low for TFT/D-TFD and active high for passive LCD.

Table 8-17 LCD FPFRAME Polarity Selection

LCD FPFRAME Polarity Select	Passive LCD FPFRAME Polarity	TFT FPFRAME Polarity
0	active high	active low
1	active low	active high

bits 2-0 TFT FPFRAME Pulse Width Bits [2:0]
For TFT/D-TFD panels only, these bits specify the pulse width of the FPFRAME output signal in number of lines.

FPFRAME pulse width in number of lines = (ContentsOfThisRegister) + 1

Note: (REG[03Ah] bits 5-0 + 1) ≥ (REG[03Bh] + 1) + (REG[03Ch] bits 2-0 + 1)

8.4.7 LCD Display Mode Registers

LCD Display Mode Register REG[040h]							RW
LCD Display Blank	n/a	n/a	SwivelView Enable Bit 1	n/a	LCD Bit-per-pixel Select Bit 2	LCD Bit-per-pixel Select Bit 1	LCD Bit-per-pixel Select Bit 0

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

Note: If a dual panel is used, the Dual Panel Buffer (REG[041h] bit 0) must be disabled (set to 1) before blanking the LCD display.

bit 4 SwivelView Enable Bit 1
When this bit = 1, the LCD display image is rotated 180° clockwise. Please refer to Section 15, “SwivelView™” on page 156 for application and limitations.
When this bit = 0, there is no hardware effect.
This bit in conjunction with SwivelView™ Enable Bit 0 achieves the following hardware rotations.

Table 8-18 Setting SwivelView Modes

SwivelView Enable Bits	SwivelView™ Modes			
	Normal	SwivelView 90°	SwivelView 180°	SwivelView 270°
SwivelView Enable Bit 0 (REG[1FCh] bit 6)	0	1	0	1
SwivelView Enable Bit 1 (REG[040h] bit 4)	0	0	1	1

bits 2-0 LCD Bit-per-pixel Select Bits [2:0]
These bits select the color depth (bit-per-pixel) for the displayed data.

Note: 16 bpp color depth bypasses the LUT and supports up to 64K colors (4096 colors if dithering disabled, see REG[041h] bit 1). TFT/D-TFD panels support up to 64K colors.

Table 8-19 LCD Bit-per-pixel Selection

Bit-per-pixel Select Bits [1:0]	Color Depth (bpp)
000-001	Reserved
010	4 bpp
011	8 bpp
100	Reserved
101	16 bpp
110-111	Reserved

LCD Miscellaneous Register REG[041h]							RW
n/a	n/a	n/a	n/a	n/a	n/a	Dithering Disable	Dual Panel Buffer Disable

bit 1 Dithering Disable
When this bit = 1, dithering on the passive LCD panel for 16 bpp mode is disabled allowing a maximum of 4096 colors (2^{12}).
When this bit = 0, dithering on the passive LCD panel for 16 bpp mode is enabled allowing a maximum of 64K colors (2^{16}).
The dithering algorithm provides more shades of each primary color when in 16 bpp mode. This bit has no effect in 4/8 bpp modes where dithering is not supported.

bit 0 Dual Panel Buffer Disable
This bit is used to disable the dual panel buffer.
When this bit = 1, the dual panel buffer is disabled.
When this bit = 0, the dual panel buffer is enabled.
When a single panel is selected, the dual panel buffer is automatically disabled and this bit has no effect.

Note: The dual panel buffer is needed to fully support dual panels. Disabling the dual panel buffer may allow higher resolution/color display modes than would otherwise be possible. However, disabling the dual panel buffer reduces image contrast and overall display quality. For details on Frame Rate Calculation, see Section 18, “Clocking” on page 169.

LCD Display Start Address Register 0
REG[042h] RW

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

LCD Display Start Address Register 1
REG[043h] RW

LCD Display Start Address Bit 15	LCD Display Start Address Bit 14	LCD Display Start Address Bit 13	LCD Display Start Address Bit 12	LCD Display Start Address Bit 11	LCD Display Start Address Bit 10	LCD Display Start Address Bit 9	LCD Display Start Address Bit 8

LCD Display Start Address Register 2
REG[044h] RW

n/a	n/a	n/a	n/a	LCD Display Start Address Bit 19	LCD Display Start Address Bit 18	LCD Display Start Address Bit 17	LCD Display Start Address Bit 16

REG[042h] bits 7-0 LCD Display Start Address Bits [19:0]

REG[043h] bits 7-0 This register forms the 20-bit address of the starting word of the LCD image in the

REG[044h] bits 3-0 display buffer.

This is a word address. An entry of 0 0000h into these registers represents the first word of the display buffer, an entry of 0 0001h represents the second word of the display buffer, and so on.

LCD Memory Address Offset Register 0
REG[046h] RW

LCD Memory Address Offset Bit 7	LCD Memory Address Offset Bit 6	LCD Memory Address Offset Bit 5	LCD Memory Address Offset Bit 4	LCD Memory Address Offset Bit 3	LCD Memory Address Offset Bit 2	LCD Memory Address Offset Bit 1	LCD Memory Address Offset Bit 0

LCD Memory Address Offset Register 1
REG[047h] RW

n/a	n/a	n/a	n/a	n/a	LCD Memory Address Offset Bit 10	LCD Memory Address Offset Bit 9	LCD Memory Address Offset Bit 8

REG[046h] bits 7-0 LCD Memory Address Offset Bits [10:0]

REG[047h] bits 2-0 These bits are the LCD display’s 11-bit address offset from the starting word of line “n” to the starting word of line “n + 1”.

A virtual image can be formed by setting this register to a value greater than the width of the display. The displayed image is a window into the larger virtual image.

LCD Pixel Panning Register REG[048h]							RW
n/a	n/a	n/a	n/a	n/a	n/a	LCD Pixel Panning Bit 1	LCD Pixel Panning Bit 0

bits 1-0 LCD Pixel Panning Bits [1:0]

This register is used to control the horizontal pixel panning of the LCD display. The display can be panned to the left by programming its respective Pixel Panning Bits to a non-zero value. This value represents the number of pixels panned. The maximum pan value is dependent on the display mode as shown in the table below.

Table 8-20 LCD Pixel Panning Selection

Color Depth (bpp)	Screen 2 Pixel Panning Bits Used
4 bpp	Bits [1:0]
8 bpp	Bit 0
16 bpp	—

Note: Smooth horizontal panning can be achieved by a combination of this register and the LCD Display Start Address registers (REG[042h], REG[043h], REG[044h]).

LCD Display FIFO High Threshold Control Register REG[04Ah]							RW
n/a	n/a	LCD Display FIFO High Threshold Bit 5	LCD Display FIFO High Threshold Bit 4	LCD Display FIFO High Threshold Bit 3	LCD Display FIFO High Threshold Bit 2	LCD Display FIFO High Threshold Bit 1	LCD Display FIFO High Threshold Bit 0

bits 5-0 LCD Display FIFO High Threshold Bits [5:0]

These bits are used to optimize the display memory request arbitration. When this register is set to 00h, the threshold is automatically set in hardware. However, programming may be required if screen corruption is present (see Section 18.2, “Example Frame Rates” on page 172).

Note: This register does not need to be used in single display modes and may only be required in some display modes where two displays are active (see Section 16.3, “Bandwidth Limitation” on page 167).

LCD Display FIFO Low Threshold Control Register
REG[04Bh] RW

n/a	n/a	LCD Display FIFO Low Threshold Bit 5	LCD Display FIFO Low Threshold Bit 4	LCD Display FIFO Low Threshold Bit 3	LCD Display FIFO Low Threshold Bit 2	LCD Display FIFO Low Threshold Bit 1	LCD Display FIFO Low Threshold Bit 0

its 5-0 LCD Display FIFO Low Threshold Bits [5:0]
When this register is set to 00h, the threshold is automatically set in hardware. If it becomes necessary to adjust REG[04Ah] from its default value, then the following
formula must be maintained:

REG[04Bh] ≥ REG[04Ah] and REG[04Bh] ≤ 3Ch

8.4.8 CRT/TV Configuration Registers

CRT/TV Horizontal Display Width Register
REG[050h] RW

n/a	CRT/TV Horizontal Display Width Bit 6	CRT/TV Horizontal Display Width Bit 5	CRT/TV Horizontal Display Width Bit 4	CRT/TV Horizontal Display Width Bit 3	CRT/TV Horizontal Display Width Bit 2	CRT/TV Horizontal Display Width Bit 1	CRT/TV Horizontal Display Width Bit 0

bits 6-0 CRT/TV Horizontal Display Width Bits [6:0]
These bits specify the CRT/TV horizontal display width, in 8 pixel resolution.

Horizontal display width in number of pixels = ((ContentsOfThisRegister) + 1) × 8

CRT/TV Horizontal Non-Display Period Register
REG[052h] RW

n/a	n/a	CRT/TV Horizontal Non-Display Period Bit 5	CRT/TV Horizontal Non-Display Period Bit 4	CRT/TV Horizontal Non-Display Period Bit 3	CRT/TV Horizontal Non-Display Period Bit 2	CRT/TV Horizontal Non-Display Period Bit 1	CRT/TV Horizontal Non-Display Period Bit 0

bits 5-0 CRT/TV Horizontal Non-Display Period Bits [5:0]
These bits specify the CRT/TV horizontal non-display period width in 8 pixel resolution.

Horizontal non-display period width in number of pixels =
((ContentsOfThisRegister) + 1) × 8 for CRT mode
(ContentsOfThisRegister) × 8 + 6 for TV mode with NTSC output
(ContentsOfThisRegister) × 8 + 7 for TV mode with PAL output

Note: For CRT, the recommended minimum value which should be programmed into this register is 3 (32 pixels).

Note: REG[052h] + 1 ≥ (REG[053h] + 1) + (REG[054h] bits 3-0 + 1)

CRT/TV HRTC Start Position Register REG[053h] RW							
n/a	n/a	CRT/TV HRTC Start Position Bit 5	CRT/TV HRTC Start Position Bit 4	CRT/TV HRTC Start Position Bit 3	CRT/TV HRTC Start Position Bit 2	CRT/TV HRTC Start Position Bit 1	CRT/TV HRTC Start Position Bit 0

bits 5-0 CRT/TV HRTC Start Position Bits [5:0]
For CRT/TV, these bits specify the delay, in 8 pixel resolution, from the start of the horizontal non-display period to the leading edge of the HRTC pulse.

The following equations can be used to determine the HRTC start position in number of pixels for each display type:

HRTC start position in number of pixels =:
[(ContentsOfThisRegister) × 8 + 4] for CRT with 4/8 bpp color depth
[(ContentsOfThisRegister) × 8 + 5] for CRT in 16 bpp color depth
[((ContentsOfThisRegister) + 1) × 8 - 7] for TV-NTSC in 4/8 bpp color depth
[((ContentsOfThisRegister) + 1) × 8 - 5] for TV-NTSC in 16 bpp color depth
[((ContentsOfThisRegister) + 1) × 8 - 7] for TV-PAL in 4/8 bpp color depth
[((ContentsOfThisRegister) + 1) × 8 - 5] for TV-PAL in 16 bpp color depth

Note: REG[052h] + 1 ≥ (REG[053h] + 1) + (REG[054h] bits 3-0 + 1)

CRT/TV HRTC Pulse Width Register REG[054h] RW							
CRT HRTC Polarity Select	n/a	n/a	n/a	CRT HRTC Pulse Width Bit 3	CRT HRTC Pulse Width Bit 2	CRT HRTC Pulse Width Bit 1	CRT HRTC Pulse Width Bit 0

bit 7 CRT HRTC Polarity Select
This bit selects the polarity of HRTC for CRTs.
When this bit = 1, the HRTC pulse is active high.
When this bit = 0, the HRTC pulse is active low.

Note: For TV, this bit must be set to 0.

bits 3-0 CRT HRTC Pulse Width Bits [3:0]
These bits specify the pulse width of the CRT HRTC output signal in 8 pixel resolution.

HRTC pulse width in number of pixels = ((ContentsOfThisRegister) + 1) × 8

Note: For TV, these bits must be set to 0.

Note: REG[052h] + 1 ≥ (REG[053h] + 1) + (REG[054h] bits 3-0 + 1)

CRT/TV Vertical Display Height Register 0 REG[056h]							RW
CRT/TV Vertical Display Height Bit 7	CRT/TV Vertical Display Height Bit 6	CRT/TV Vertical Display Height Bit 5	CRT/TV Vertical Display Height Bit 4	CRT/TV Vertical Display Height Bit 3	CRT/TV Vertical Display Height Bit 2	CRT/TV Vertical Display Height Bit 1	CRT/TV Vertical Display Height Bit 0

CRT/TV Vertical Display Height Register 1 REG[057h]							RW
n/a	n/a	n/a	n/a	n/a	n/a	CRT/TV Vertical Display Height Bit 9	CRT/TV Vertical Display Height Bit 8

REG[056h] bits 7-0 REG[057h] bits 1-0 — CRT/TV Vertical Display Height Bits [9:0]

These bits specify the CRT/TV vertical display height, in 1 line resolution.

Vertical display height in number of lines = (ContentsOfThisRegister) + 1

CRT/TV Vertical Non-Display Period Register REG[058h]							RW
CRT/TV Vertical Non-Display Period Status (RO)	CRT/TV Vertical Non-Display Period Bit 6	CRT/TV Vertical Non-Display Period Bit 5	CRT/TV Vertical Non-Display Period Bit 4	CRT/TV Vertical Non-Display Period Bit 3	CRT/TV Vertical Non-Display Period Bit 2	CRT/TV Vertical Non-Display Period Bit 1	CRT/TV Vertical Non-Display Period Bit 0

bit 7 — CRT/TV Vertical Non-Display Period Status
This is a read-only status bit.
When a read from this bit = 1, a CRT/TV vertical non-display period is occurring.
When a read from this bit = 0, the CRT/TV output is in a vertical display period.

bits 6-0 — CRT/TV Vertical Non-Display Period Bits [6:0]
These bits specify the CRT/TV vertical non-display period height in 1 line resolution.

Vertical non-display period height in number of lines = (ContentsOfThisRegister) + 1

Note: (REG[058h] bits 6-0 + 1) ≥ (REG[059h] + 1) + (REG[05Ah] bits 2-0 + 1)

CRT/TV VRTC Start Position Register REG[059h]							RW
n/a	CRT/TV VRTC Start Position Bit 6	CRT/TV VRTC Start Position Bit 5	CRT/TV VRTC Start Position Bit 4	CRT/TV VRTC Start Position Bit 3	CRT/TV VRTC Start Position Bit 2	CRT/TV VRTC Start Position Bit 1	CRT/TV VRTC Start Position Bit 0

bits 6-0 CRT/TV VRTC Start Position Bits [6:0]
For CRT/TV, these bits specify the delay in lines from the start of the vertical non-display period to the leading edge of the VRTC pulse.

VRTC start position in number of lines = (ContentsOfThisRegister) + 1

Note: (REG[058h] bits 6-0 + 1) ≥ (REG[059h] + 1) + (REG[05Ah] bits 2-0 + 1)

CRT VRTC Pulse Width Register REG[05Ah]							RW
CRT VRTC Polarity Select	n/a	n/a	n/a	n/a	CRT VRTC Pulse Width Bit 2	CRT VRTC Pulse Width Bit 1	CRT VRTC Pulse Width Bit 0

bit 7 CRT VRTC Polarity Select
This bit selects the polarity of VRTC for CRT.
When this bit = 1, the VRTC pulse is active high.
When this bit = 0, the VRTC pulse is active low.

Note: For TV, this bit must be set to 0.

bits 2-0 CRT VRTC Pulse Width Bits [2:0]
These bits specify the pulse width of the CRT VRTC output signal in number of lines.

VRTC pulse width in number of lines = (ContentsOfThisRegister) + 1

Note: For TV, these bits must be set to 0.

Note: (REG[058h] bits 6-0 + 1) ≥ (REG[059h] + 1) + (REG[05Ah] bits 2-0 + 1)

TV Output Control Register REG[05Bh]							RW
n/a	n/a	TV Chrominance Filter Enable	TV Luminance Filter Enable	DAC Output Level Select	n/a	TV S-Video/ Composite Output Select	TV PAL/ NTSC Output Select

bit 5 TV Chrominance Filter Enable

When this bit = 1, the TV chrominance filter is enabled.
When this bit = 0, the TV chrominance filter is disabled.
The chrominance filter adjusts the color of the TV by limiting the bandwidth of the chrominance signal (reducing cross-luminance distortion). This reduces the “ragged edges” seen at boundaries between sharp color transitions. This filter is most useful for composite video output.

bit 4 TV Luminance Filter Enable

When this bit = 1, the TV luminance filter is enabled.
When this bit = 0, the TV luminance filter is disabled.
The luminance filter adjusts the brightness of the TV by limiting the bandwidth of the luminance signal (reducing cross-chrominance distortion). This reduces the “rainbow-like” colors at boundaries between sharp luminance transitions. This filter is most useful for composite video output.

bit 3 DAC Output Level Select

When this bit is set to 1 it allows IREF to be reduced. This bit should be set as described in the following table.

Table 8-21 DAC Output Level Selection

LCD	CRT	TV	REG[05Bh] bit 3	IREF (mA)
Enabled	Disabled	Disabled	x	x
x	Enabled	Disabled	1	4.6
x	Enabled	Enabled	0	9.2

x = don’t care

Note: Figure 4-2 “External Circuitry for CRT Interface” on page 20 shows an example implementation of the required external CRT/TV IREF circuitry.

bit 1 TV S-Video/Composite Output Select

When this bit = 1, S-Video TV signal output is selected.
When this bit = 0, Composite TV signal output is selected.

bit 0 TV PAL/NTSC Output Select

When this bit = 1, PAL format TV signal output is selected.
When this bit = 0, NTSC format TV signal output is selected.
This bit must be set to 0 when CRT is enabled.

8.4.9 CRT/TV Display Mode Registers

CRT/TV Display Mode Register REG[060h]							RW
CRT/TV Display Blank	n/a	n/a	n/a	n/a	CRT/TV Bit-per-pixel Select Bit 2	CRT/TV Bit-per-pixel Select Bit 1	CRT/TV Bit-per-pixel Select Bit 0

bit 7 CRT/TV Display Blank
When this bit = 1 the CRT/TV display pipeline is disabled and all CRT/TV data outputs are forced to zero (the screen is blanked).
When this bit = 0 the CRT display pipeline is enabled.

bits 2-0 CRT/TV Bit-per-pixel Select Bits [2:0]
These bits select the bit-per-pixel for the displayed data.

Note: Color depth of 16 bpp bypasses the LUT and support up to 64K colors on the CRT/TV.

Table 8-22 CRT/TV Bit-per-pixel Selection

Bit-per-pixel Select Bits 1:0	Color Depth (bpp)
000	Reserved
001	Reserved
010	4 bpp
011	8 bpp
100	Reserved
101	16 bpp
110-111	Reserved

CRT/TV Display Start Address Register 0 REG[062h]							RW
CRT/TV Display Start Address Bit 7	CRT/TV Display Start Address Bit 6	CRT/TV Display Start Address Bit 5	CRT/TV Display Start Address Bit 4	CRT/TV Display Start Address Bit 3	CRT/TV Display Start Address Bit 2	CRT/TV Display Start Address Bit 1	CRT/TV Display Start Address Bit 0

CRT/TV Display Start Address Register 1 REG[063h]							RW
CRT/TV Display Start Address Bit 15	CRT/TV Display Start Address Bit 14	CRT/TV Display Start Address Bit 13	CRT/TV Display Start Address Bit 12	CRT/TV Display Start Address Bit 11	CRT/TV Display Start Address Bit 10	CRT/TV Display Start Address Bit 9	CRT/TV Display Start Address Bit 8

CRT/TV Display Start Address Register 2 REG[064h]							RW
n/a	n/a	n/a	n/a	CRT/TV Display Start Address Bit 19	CRT/TV Display Start Address Bit 18	CRT/TV Display Start Address Bit 17	CRT/TV Display Start Address Bit 16

REG[062h] bits 7-0 CRT/TV Start Address Bits [19:0]

REG[063h] bits 7-0 This register forms the 20-bit address for the starting word of the CRT/TV image in the

REG[064h] bits 3-0 display buffer.

This is a word address. An entry of 00000h into these registers represents the first word of the display buffer, an entry of 00001h represents the second word of the display buffer, and so on.

CRT/TV Memory Address Offset Register 0 REG[066h]							RW
CRT/TV Memory Address Offset Bit 7	CRT/TV Memory Address Offset Bit 6	CRT/TV Memory Address Offset Bit 5	CRT/TV Memory Address Offset Bit 4	CRT/TV Memory Address Offset Bit 3	CRT/TV Memory Address Offset Bit 2	CRT/TV Memory Address Offset Bit 1	CRT/TV Memory Address Offset Bit 0

CRT/TV Memory Address Offset Register 1 REG[067h]							RW
n/a	n/a	n/a	n/a	n/a	CRT/TV Memory Address Offset Bit 10	CRT/TV Memory Address Offset Bit 9	CRT/TV Memory Address Offset Bit 8

REG[066h] bits 7-0 CRT/TV Memory Address Offset Bits [10:0]

REG[067h] bits 2-0 These bits are the CRT/TV display’s 11-bit address offset from the starting word of line “n” to the starting word of line “n + 1”. A virtual image can be formed by setting this register to a value greater than the width of the display. The displayed image is a window into the larger virtual image.

CRT/TV Pixel Panning Register REG[068h] RW							
n/a	n/a	n/a	n/a	n/a	n/a	CRT/TV Pixel Panning Bit 1	CRT/TV Pixel Panning Bit 0

bits 1-0

CRT/TV Pixel Panning Bits [1:0]
This register is used to control the horizontal pixel panning of the CRT/TV display. The display can be panned to the left by programming its respective Pixel Panning Bits to a non-zero value. This value represents the number of pixels panned. The maximum pan value is dependent on the display mode as shown in the table below.

Table 8-23 CRT/TV Pixel Panning Selection

Color Depth (bpp)	Screen 2 Pixel Panning Bits Used
4 bpp	Bits [1:0]
8 bpp	Bit 0
16 bpp	—

Note: Smooth horizontal panning can be achieved by a combination of this register and the CRT/TV Display Start Address registers (REG[062h], REG[063h], REG[064h]).

CRT/TV Display FIFO High Threshold Control Register REG[06Ah] RW							
n/a	n/a	CRT/TV Display FIFO High Threshold Bit 5	CRT/TV Display FIFO High Threshold Bit 4	CRT/TV Display FIFO High Threshold Bit 3	CRT/TV Display FIFO High Threshold Bit 2	CRT/TV Display FIFO High Threshold Bit 1	CRT/TV Display FIFO High Threshold Bit 0

bits 5-0

CRT/TV Display FIFO High Threshold Bits [5:0]
These bits are used to optimize the display memory request arbitration. When this register is set to 00h, the threshold is automatically set in hardware. However, programming may be required if screen corruption is present (see Section 18.2, “Example Frame Rates” on page 172).

Note: This register does not need to be used in single display modes and may only be required in some display modes where two displays are active (see Section 16.3, “Bandwidth Limitation” on page 167).

CRT/TV Display FIFO Low Threshold Control Register REG[06Bh] RW							
n/a	n/a	CRT/TV Display FIFO Low Threshold Bit 5	CRT/TV Display FIFO Low Threshold Bit 4	CRT/TV Display FIFO Low Threshold Bit 3	CRT/TV Display FIFO Low Threshold Bit 2	CRT/TV Display FIFO Low Threshold Bit 1	CRT/TV Display FIFO Low Threshold Bit 0

bits 5-0 CRT/TV Display FIFO Low Threshold Bits [5:0]
When this register is set to 00h, the threshold is automatically set in hardware. If it becomes necessary to adjust REG[06Ah] from its default value, then the following formula must be maintained.

REG[06Bh] ≥ REG[06Ah] and REG[06Bh] ≤ 3Ch

8.4.10 LCD Ink/Cursor Registers

LCD Ink/Cursor Control Register REG[070h] RW							
n/a	n/a	n/a	n/a	n/a	n/a	LCD Ink/ Cursor Mode Bit 1	LCD Ink/ Cursor Mode Bit 0

bits 1-0 LCD Ink/Cursor Control Bits [1:0]
These bits enable the LCD Ink/Cursor circuitry.

Table 8-24 LCD Ink/Cursor Selection

LCD Ink/Cursor Bits [1:0]	Mode
00	Inactive
01	Cursor
10	Ink
11	Reserved

Note: While in Ink mode, the Cursor X and Y Position registers must be set to 00h.

LCD Ink/Cursor Start Address Register REG[071h]							RW
LCD Ink/ Cursor Start Address Bit 7	LCD Ink/ Cursor Start Address Bit 6	LCD Ink/ Cursor Start Address Bit 5	LCD Ink/ Cursor Start Address Bit 4	LCD Ink/ Cursor Start Address Bit 3	LCD Ink/ Cursor Start Address Bit 2	LCD Ink/ Cursor Start Address Bit 1	LCD Ink/ Cursor Start Address Bit 0

bits 7-0

LCD Ink/Cursor Start Address Bits [7:0]
Encoded bits defining the start address for the LCD Ink/Cursor. For Cursor modes, a start address of 0 should be valid for most applications. For Ink or special Cursor modes, the start address should be set at an address location that does not conflict with the display memory of dual panel buffer, which always takes the top M memory locations in bytes.

M = (Panel Height × Panel Width / 16) × c

where
c = 1 for monochrome panel
= 4 for color panel

Table 8-25 LCD Ink/Cursor Start Address Encoding

LCD Ink/Cursor Start Address Bits [7:0]	Start Address
0	Memory Size - 1024
n = 160...1	Memory Size - n × 8192
n = 255...161	invalid

Note: The effect of this register takes place at the next LCD vertical non-display period.

Note: See Section 10, “Display Buffer” on page 137 for display buffer organization.

LCD Cursor X Position Register 0 REG[072h]							RW
LCD Cursor X Position Bit 7	LCD Cursor X Position Bit 6	LCD Cursor X Position Bit 5	LCD Cursor X Position Bit 4	LCD Cursor X Position Bit 3	LCD Cursor X Position Bit 2	LCD Cursor X Position Bit 1	LCD Cursor X Position Bit 0

LCD Cursor X Position Register 1 REG[073h]							RW
LCD Cursor X Sign	n/a	n/a	n/a	n/a	n/a	LCD Cursor X Position Bit 9	LCD Cursor X Position Bit 8

REG[073h] bit 7

LCD Cursor X Sign
When this bit = 1, it defines the LCD Cursor X Position register to be a negative number. The negative number shall not exceed 63 decimal.
When this bit = 0, it defines the LCD Cursor X Position register to be a positive number.

REG[072h] bits 7-0 LCD Cursor X Position Bits [9:0]

REG[073h] bits 1-0 A 10-bit register that defines the horizontal position of the LCD Cursor's top left hand corner in pixel units. This register is only valid when Cursor has been selected in the LCD Ink/Cursor select registers.

Note: The effect of REG[072h] through REG[074h] takes place only after REG[075h] is written and at the next LCD vertical non-display period.The effect of REG[075h] takes place at the next LCD vertical non-display period.

LCD Cursor Y Position Register 0
REG[074h] RW

LCD Cursor Y Position Bit 7	LCD Cursor Y Position Bit 6	LCD Cursor Y Position Bit 5	LCD Cursor Y Position Bit 4	LCD Cursor Y Position Bit 3	LCD Cursor Y Position Bit 2	LCD Cursor Y Position Bit 1	LCD Cursor Y Position Bit 0

LCD Cursor Y Position Register 1
REG[075h] RW

LCD Cursor Y Sign	n/a	n/a	n/a	n/a	n/a	LCD Cursor Y Position Bit 9	LCD Cursor Y Position Bit 8

REG[075h] bit 7 LCD Cursor Y Sign
When this bit = 1, it defines the LCD Cursor Y Position register to be a negative number. The negative number shall not exceed 63 decimal.
When this bit = 0, it defines the LCD Cursor Y Position register to be a positive number.

REG[074h] bits 7-0 LCD Cursor Y Position Bits [9:0]

REG[075h] bits 1-0 A 10-bit register that defines the vertical position of the LCD Cursor's top left hand corner in pixel units. This register is only valid when Cursor has been selected in the LCD Ink/Cursor select registers.

Note: The effect of REG[072h] through REG[074h] takes place only after REG[075h] is written and at the next LCD vertical non-display period.The effect of REG[075h] takes place at the next LCD vertical non-display period.

LCD Ink/Cursor Blue Color 0 Register
REG[076h] RW

n/a	n/a	n/a	LCD Ink/ Cursor Blue Color 0 Bit 4	LCD Ink/ Cursor Blue Color 0 Bit 3	LCD Ink/ Cursor Blue Color 0 Bit 2	LCD Ink/ Cursor Blue Color 0 Bit 1	LCD Ink/ Cursor Blue Color 0 Bit 0

bits 4-0 LCD Ink/Cursor Blue Color 0 Bits[4:0]
These bits define the blue LCD Ink/Cursor color 0.

LCD Ink/Cursor Green Color 0 Register
REG[077h] RW

n/a	n/a	LCD Ink/ Cursor Green Color 0 Bit 5	LCD Ink/ Cursor Green Color 0 Bit 4	LCD Ink/ Cursor Green Color 0 Bit 3	LCD Ink/ Cursor Green Color 0 Bit 2	LCD Ink/ Cursor Green Color 0 Bit 1	LCD Ink/ Cursor Green Color 0 Bit 0

bits 5-0 LCD Ink/Cursor Green Color 0 Bits[5:0]
These bits define the green LCD ink/Cursor color 0.

LCD Ink/Cursor Red Color 0 Register
REG[078h] RW

n/a	n/a	n/a	LCD Ink/ Cursor Red Color 0 Bit 4	LCD Ink/ Cursor Red Color 0 Bit 3	LCD Ink/ Cursor Red Color 0 Bit 2	LCD Ink/ Cursor Red Color 0 Bit 1	LCD Ink/ Cursor Red Color 0 Bit 0

bits 4-0 LCD Ink/Cursor Red Color 0 Bits[4:0]
These bits define the red LCD Ink/Cursor color 0.

LCD Ink/Cursor Blue Color 1 Register
REG[07Ah] RW

n/a	n/a	n/a	LCD Ink/ Cursor Blue Color 1 Bit 4	LCD Ink/ Cursor Blue Color 1 Bit 3	LCD Ink/ Cursor Blue Color 1 Bit 2	LCD Ink/ Cursor Blue Color 1 Bit 1	LCD Ink/ Cursor Blue Color 1 Bit 0

bits 4-0 LCD Ink/Cursor Blue Color 1 Bits[4:0]
These bits define the blue LCD Ink/Cursor color 1.

LCD Ink/Cursor Green Color 1 Register
REG[07Bh] RW

n/a	n/a	LCD Ink/ Cursor Green Color 1 Bit 5	LCD Ink/ Cursor Green Color 1 Bit 4	LCD Ink/ Cursor Green Color 1 Bit 3	LCD Ink/ Cursor Green Color 1 Bit 2	LCD Ink/ Cursor Green Color 1 Bit 1	LCD Ink/ Cursor Green Color 1 Bit 0

bits 5-0 LCD Ink/Cursor Green Color 1 Bits[5:0]
These bits define the green LCD Ink/Cursor color 1.

LCD Ink/Cursor Red Color 1 Register
REG[07Ch] RW

n/a	n/a	n/a	LCD Ink/ Cursor Red Color 1 Bit 4	LCD Ink/ Cursor Red Color 1 Bit 3	LCD Ink/ Cursor Red Color 1 Bit 2	LCD Ink/ Cursor Red Color 1 Bit 1	LCD Ink/ Cursor Red Color 1 Bit 0

bits 4-0 LCD Ink/Cursor Red Color 1 Bits[4:0]
These bits define the red LCD Ink/Cursor color 1.

LCD Ink/Cursor FIFO High Threshold Register REG[07Eh]							RW
n/a	n/a	n/a	n/a	LCD Ink/ Cursor FIFO High Threshold Bit 3	LCD Ink/ Cursor FIFO High Threshold Bit 2	LCD Ink/ Cursor FIFO High Threshold Bit 1	LCD Ink/ Cursor FIFO High Threshold Bit 0

bits 3-0 LCD Ink/Cursor FIFO High Threshold Bits [3:0]
These bits are used to optimize the display memory request arbitration for the Hardware Cursor/Ink Layer. When this register is set to 00h, the threshold is automatically set in hardware.

8.4.11 CRT/TV Ink/Cursor Registers

CRT/TV Ink/Cursor Control Register REG[080h]							RW
n/a	n/a	n/a	n/a	n/a	n/a	CRT/TV Ink/ Cursor Mode Bit 1	CRT/TV Ink/ Cursor Mode Bit 0

bits 1-0 CRT/TV Ink/Cursor Control Bits [1:0]
These bits enable the CRT/TV Ink/Cursor circuitry.

Table 8-26 CRT/TV Ink/Cursor Selection

CRT/TV Ink/Cursor Bits [1:0]	Mode
00	Inactive
01	Cursor
10	Ink
11	Reserved

Note: While in Ink mode, the Cursor X and Y Position registers must be set to 00h.

CRT/TV Ink/Cursor Start Address Register REG[081h] RW							
CRT/TV Ink/ Cursor Start Address Bit 7	CRT/TV Ink/ Cursor Start Address Bit 6	CRT/TV Ink/ Cursor Start Address Bit 5	CRT/TV Ink/ Cursor Start Address Bit 4	CRT/TV Ink/ Cursor Start Address Bit 3	CRT/TV Ink/ Cursor Start Address Bit 2	CRT/TV Ink/ Cursor Start Address Bit 1	CRT/TV Ink/ Cursor Start Address Bit 0

bits 7-0

CRT/TV Ink/Cursor Start Address Bits [7:0]
Encoded bits defining the start address for the CRT/TV Ink/Cursor. For Cursor modes, a start address of 0 should be valid for most applications. For Ink or special Cursor modes, the start address should be set at an address location that does not conflict with the display memory of dual panel buffer, which always takes the top memory locations (M) in bytes.

M = (Panel Height × Panel Width / 16) × c

where
c = 1 for monochrome panel
= 4 for color panel

Table 8-27 CRT/TV Ink/Cursor Start Address Encoding

CRT/TV Ink/Cursor Start Address Bits [7:0]	Start Address
0	Memory Size - 1024
n = 160...1	Memory Size - n × 8192
n = 255...161	Invalid

Note: The effect of this register takes place at the next CRT/TV vertical non-display period.

Note: See Section 10, “Display Buffer” on page 137 for display buffer organization.

CRT/TV Cursor X Position Register 0 REG[082h]							RW
CRT/TV Cursor X Position Bit 7	CRT/TV Cursor X Position Bit 6	CRT/TV Cursor X Position Bit 5	CRT/TV Cursor X Position Bit 4	CRT/TV Cursor X Position Bit 3	CRT/TV Cursor X Position Bit 2	CRT/TV Cursor X Position Bit 1	CRT/TV Cursor X Position Bit 0

CRT/TV Cursor X Position Register 1 REG[083h]							RW
CRT/TV Cursor X Sign	n/a	n/a	n/a	n/a	n/a	CRT/TV Cursor X Position Bit 9	CRT/TV Cursor X Position Bit 8

REG[083h] bit 7 CRT/TV Cursor X Sign
When this bit = 1, it defines the CRT/TV Cursor X Position register to be a negative number. The negative number should not exceed 63 decimal.
When this bit = 0, it defines the CRT/TV Cursor X Position register to be a positive number.

REG[082h] bits 7-0 CRT/TV Cursor X Position Bits [9:0]

REG[083h] bits 1-0 A 10-bit register that defines the horizontal position of the CRT/TV Cursor's top left hand corner in pixel units. This register is only valid when Cursor has been selected in the CRT/TV Ink/Cursor select registers.

Note: The effect of REG[082h] through REG[084h] takes place only after REG[085h]is written to and at the next CRT/TV vertical non-display period.The effect of REG[085h] takes place at the next CRT/TV vertical non-display period.

CRT/TV Cursor Y Position Register 0 REG[084h]							RW
CRT/TV Cursor Y Position Bit 7	CRT/TV Cursor Y Position Bit 6	CRT/TV Cursor Y Position Bit 5	CRT/TV Cursor Y Position Bit 4	CRT/TV Cursor Y Position Bit 3	CRT/TV Cursor Y Position Bit 2	CRT/TV Cursor Y Position Bit 1	CRT/TV Cursor Y Position Bit 0

CRT/TV Cursor Y Position Register 1 REG[085h]							RW
CRT/TV Cursor Y Sign	n/a	n/a	n/a	n/a	n/a	CRT/TV Cursor Y Position Bit 9	CRT/TV Cursor Y Position Bit 8

REG[084h] bit 7 CRT/TV Cursor YSign
When this bit = 1, it defines the CRT/TV Cursor Y Position register to be a negative number. The negative number shall not exceed 63 decimal.
When this bit = 0, it defines the CRT/TV Cursor Y Position register to be a positive number.

REG[084h] bits 7-0 CRT/TV Cursor Y Position Bits [9:0]
REG[085h] bits 1-0 A 10-bit register that defines the vertical position of the CRT/TV Cursor's top left hand corner in pixel units. This register is only valid when Cursor has been selected in the CRT/TV Ink/Cursor select registers.

Note: The effect of REG[082h] through REG[084h] takes place only after REG[085h]is written to and at the next CRT/TV vertical non-display period.The effect of REG[085h] takes place at the next CRT/TV vertical non-display period.

CRT/TV Ink/Cursor Blue Color 0 Register
REG[086h] RW

n/a	n/a	n/a	CRT/TV Ink/ Cursor Blue Color 0 Bit 4	CRT/TV Ink/ Cursor Blue Color 0 Bit 3	CRT/TV Ink/ Cursor Blue Color 0 Bit 2	CRT/TV Ink/ Cursor Blue Color 0 Bit 1	CRT/TV Ink/ Cursor Blue Color 0 Bit 0

bits 4-0 CRT/TV Ink/Cursor Blue Color 0 Bits[4:0]
These bits define the blue CRT/TV Ink/Cursor color 0.

CRT/TV Ink/Cursor Green Color 0 Register
REG[087h] RW

n/a	n/a	CRT/TV Ink/ Cursor Green Color 0 Bit 5	CRT/TV Ink/ Cursor Green Color 0 Bit 4	CRT/TV Ink/ Cursor Green Color 0 Bit 3	CRT/TV Ink/ Cursor Green Color 0 Bit 2	CRT/TV Ink/ Cursor Green Color 0 Bit 1	CRT/TV Ink/ Cursor Green Color 0 Bit 0

bits 5-0 CRT/TV Ink/Cursor Green Color 0 Bits[5:0]
These bits define the green CRT/TV Ink/Cursor color 0.

CRT/TV Ink/Cursor Red Color 0 Register
REG[088h] RW

n/a	n/a	n/a	CRT/TV Ink/ Cursor Red Color 0 Bit 4	CRT/TV Ink/ Cursor Red Color 0 Bit 3	CRT/TV Ink/ Cursor Red Color 0 Bit 2	CRT/TV Ink/ Cursor Red Color 0 Bit 1	CRT/TV Ink/ Cursor Red Color 0 Bit 0

bits 4-0 CRT/TV Ink/Cursor Red Color 0 Bits[4:0]
These bits define the red CRT/TV Ink/Cursor color 0.

CRT/TV Ink/Cursor Blue Color 1 Register
REG[08Ah] RW

n/a	n/a	n/a	CRT/TV Ink/ Cursor Blue Color 1 Bit 4	CRT/TV Ink/ Cursor Blue Color 1 Bit 3	CRT/TV Ink/ Cursor Blue Color 1 Bit 2	CRT/TV Ink/ Cursor Blue Color 1 Bit 1	CRT/TV Ink/ Cursor Blue Color 1 Bit 0

bits 4-0 CRT/TV Ink/Cursor Blue Color 1 Bits[4:0]
These bits define the blue CRT/TV Ink/Cursor color 1.

CRT/TV Ink/Cursor Green Color 1 Register
REG[08Bh] RW

n/a	n/a	CRT/TV Ink/ Cursor Green Color 1 Bit 5	CRT/TV Ink/ Cursor Green Color 1 Bit 4	CRT/TV Ink/ Cursor Green Color 1 Bit 3	CRT/TV Ink/ Cursor Green Color 1 Bit 2	CRT/TV Ink/ Cursor Green Color 1 Bit 1	CRT/TV Ink/ Cursor Green Color 1 Bit 0

bits 5-0 CRT/TV Ink/Cursor Green Color 1 Bits[5:0]
These bits define the green CRT/TV Ink/Cursor color 1.

CRT/TV Ink/Cursor Red Color 1 Register
REG[08Ch] RW

n/a	n/a	n/a	CRT/TV Ink/ Cursor Red Color 1 Bit 4	CRT/TV Ink/ Cursor Red Color 1 Bit 3	CRT/TV Ink/ Cursor Red Color 1 Bit 2	CRT/TV Ink/ Cursor Red Color 1 Bit 1	CRT/TV Ink/ Cursor Red Color 1 Bit 0

bits 4-0 CRT/TV Ink/Cursor Red Color 1 Bits[4:0]
These bits define the red CRT/TV Ink/Cursor color 1.

CRT/TV Ink/Cursor FIFO High Threshold Register
REG[08Eh] RW

n/a	n/a	n/a	n/a	CRT/TV Ink/ Cursor FIFO High Threshold Bit 3	CRT/TV Ink/ Cursor FIFO High Threshold Bit 2	CRT/TV Ink/ Cursor FIFO High Threshold Bit 1	CRT/TV Ink/ Cursor FIFO High Threshold Bit 0

bits 3-0 CRT/TV Ink/Cursor FIFO High Threshold Bits [5:0]
These bits are used to optimize the display memory request arbitration for the Hardware Cursor/Ink Layer. When this register is set to 00h, the threshold is automatically set in hardware.

8.4.12 BitBLT Configuration Registers

BitBLT Control Register 0 REG[100h]							RW
BitBLT Active Status	BitBLT FIFO Not Empty Status (RO)	BitBLT FIFO Half Full Status (RO)	BitBLT FIFO Full Status(RO)	n/a	n/a	BitBLT Destination Linear Select	BitBLT Source Linear Select

bit 7 BitBLT Active Status
This register bit has two data paths, one for write, the other for read.

Write Data Path
When software writes a one to this bit, it initiates the 2D operation.

Read Data Path
The read back of this register indicates the status of the 2D engine.
When a read from this bit = 1, the 2D engine is busy.
When a read from this bit = 0, the 2D engine is idle and is ready for the next operation.

Table 8-28 BitBLT Active Status

BitBLT Active Status		State
Write	Read	
0	0	Idle
0	1	Reserved
1	0	Initiating operation
1	1	Operation in progress

bit 6 BitBLT FIFO Not-Empty Status
This is a read-only status bit.
When this bit = 0, the BitBLT FIFO is empty.
When this bit = 1, the BitBLT FiFO has at least one data.
To reduce system memory read latency, software can monitor this bit prior to a BitBLT read burst operation.

The following table shows the number of words available in BitBLT FIFO under different status conditions.

Table 8-29 BitBLT FIFO Words Available

BitBLT FIFO Full Status (REG[100h] Bit 4)	BitBLT FIFO Half Full Status (REG[100h] Bit 5)	BitBLT FIFO Not Empty Status (REG[100h] Bit 6)	Number of Words available in BitBLT FIFO
0	0	0	0
0	0	1	1 to 6
0	1	1	7 to 14
1	1	1	15 to 16

bit 5 BitBLT FIFO Half Full Status
This is a read-only status bit.
When this bit = 1, the BitBLT FIFO is half full or greater than half full.
When this bit = 0, the BitBLT FIFO is less than half full.

bit 4 BitBLT FIFO Full Status
This is a read-only status bit.
When this bit = 1, the BitBLT FIFO is full.
When this bit = 0, the BitBLT FIFO is not full.

bit 1 BitBLT Destination Linear Select
When this bit = 1, the Destination BitBLT is stored as a contiguous linear block of memory.
When this bit = 0, the Destination BitBLT is stored as a rectangular region of memory.
The BitBLT Memory Address Offset (REG[10Ch], REG[10Dh]) determines the address offset from the start of one line to the next line.

bit 0 BitBLT Source Linear Select
When this bit = 1, the Source BitBLT is stored as a contiguous linear block of memory.
When this bit = 0, the Source BitBLT is stored as a rectangular region of memory.
The BitBLT Memory Address Offset (REG[10Ch], REG[10Dh]) determines the address offset from the start of one line to the next line.

BitBLT Control Register 1
REG[101h] RW

n/a	n/a	n/a	Reserved	n/a	n/a	n/a	BitBLT Color Format Select

bit 4 Reserved.
This bit must be set to 0.

bit 0 BitBLT Color Format Select
This bit selects the color format that the 2D operation is applied to.
When this bit = 0, 8 bpp (256 color) format is selected.
When this bit = 1, 16 bpp (64K color) format is selected.

BitBLT ROP Code/Color Expansion Register REG[102h] RW							
n/a	n/a	n/a	n/a	BitBLT ROP Code Bit 3	BitBLT ROP Code Bit 2	BitBLT ROP Code Bit 1	BitBLT ROP Code Bit 0

bits 3-0 BitBLT Raster Operation Code/Color Expansion Bits [3:0]
ROP Code for Write BitBLT and Move BitBLT. Bits 2-0 also specify the start bit position for Color Expansion.

Table 8-30 BitBLT ROP Code/Color Expansion Function Selection

BitBLT ROP Code Bits [3:0]	Boolean Function for Write BitBLT and Move BitBLT	Boolean Function for Pattern Fill	Start Bit Position for Color Expansion
0000	0 (Blackness)	0 (Blackness)	bit 0
0001	~S . ~D or ~(S + D)	~P . ~D or ~(P + D)	bit 1
0010	~S . D	~P . D	bit 2
0011	~S	~P	bit 3
0100	S . ~D	P . ~D	bit 4
0101	~D	~D	bit 5
0110	S ^ D	P ^ D	bit 6
0111	~S + ~D or ~(S . D)	~P + ~D or ~(P . D)	bit 7
1000	S . D	P . D	bit 0
1001	~(S ^ D)	~(P ^ D)	bit 1
1010	D	D	bit 2
1011	~S + D	~P + D	bit 3
1100	S	P	bit 4
1101	S + ~D	P + ~D	bit 5
1110	S + D	P + D	bit 6
1111	1 (Whiteness)	1 (Whiteness)	bit 7

Note: S = Source, D = Destination, P = Pattern.

BitBLT Operation Register REG[103h]							RW
n/a	n/a	n/a	n/a	BitBLT Operation Bit 3	BitBLT Operation Bit 2	BitBLT Operation Bit 1	BitBLT Operation Bit 0

bits 3-0 BitBLT Operation Bits [3:0]
Specifies the 2D Operation to be carried out based on the following table.

Table 8-31 BitBLT Operation Selection

BitBLT Operation Bits [3:0]	BitBLT Operation
0000	Write BitBLT with ROP.
0001	Read BitBLT.
0010	Move BitBLT in positive direction with ROP.
0011	Move BitBLT in negative direction with ROP.
0100	Transparent Write BitBLT.
0101	Transparent Move BitBLT in positive direction.
0110	Pattern Fill with ROP.
0111	Pattern Fill with transparency.
1000	Color Expansion.
1001	Color Expansion with transparency.
1010	Move BitBLT with Color Expansion.
1011	Move BitBLT with Color Expansion and transparency.
1100	Solid Fill.
Other combinations	Reserved

Note: The BitBLT operations Pattern Fill with ROP and Pattern Fill with transparency require a BitBLT width > 2 for 8 bpp color depths and a BitBLT width > 1 for 16 bpp color depths. The BitBLT width is set in REG[110h], REG[111h].

BitBLT Source Start Address Register 0
REG[104h] RW

BitBLT Source Start Address Bit 7	BitBLT Source Start Address Bit 6	BitBLT Source Start Address Bit 5	BitBLT Source Start Address Bit 4	BitBLT Source Start Address Bit 3	BitBLT Source Start Address Bit 2	BitBLT Source Start Address Bit 1	BitBLT Source Start Address Bit 0

BitBLT Source Start Address Register 1
REG[105h] RW

BitBLT Source Start Address Bit 15	BitBLT Source Start Address Bit 14	BitBLT Source Start Address Bit 13	BitBLT Source Start Address Bit 12	BitBLT Source Start Address Bit 11	BitBLT Source Start Address Bit 10	BitBLT Source Start Address Bit 9	BitBLT Source Start Address Bit 8

BitBLT Source Start Address Register 2
REG[106h] RW

n/a	n/a	n/a	BitBLT Source Start Address Bit 20	BitBLT Source Start Address Bit 19	BitBLT Source Start Address Bit 18	BitBLT Source Start Address Bit 17	BitBLT Source Start Address Bit 16

REG[104h] bits 7-0 BitBLT Source Start Address Bits [20:0]

REG[105h] bits 7-0 A 21-bit register that specifies the source start address for the BitBLT operation.

REG[106h] bits 4-0 If data is sourced from the CPU, then bit 0 is used for byte alignment within a 16-bit word and the other address bits are ignored. In pattern fill operation, the BitBLT Source Start Address is defined by the following equation.

Value programmed to the Source Start Address Register =Pattern Base Address + Pattern Line Offset + Pixel Offset.

The following table shows how Source Start Address Register is defined for 8 and 16 bpp color depths.

Table 8-32 BitBLT Source Start Address Selection

Color Format	Pattern Base Address[20:0]	Pattern Line Offset[2:0]	Pixel Offset[3:0]
8 bpp	BitBLT Source Start Address[20:6]	BitBLT Source Start Address[5:3]	BitBLT Source Start Address[2:0]
16 bpp	BitBLT Source Start Address[20:7]	BitBLT Source Start Address[6:4]	BitBLT Source Start Address[3:0]

Note: For further information on the BitBLT Source Start Address register, see the *S1D13806 Programming Notes and Examples*, document number X28B-G-003-xx.

BitBLT Destination Start Address Register 0
REG[108h] RW

BitBLT Destination Start Address Bit 7	BitBLT Destination Start Address Bit 6	BitBLT Destination Start Address Bit 5	BitBLT Destination Start Address Bit 4	BitBLT Destination Start Address Bit 3	BitBLT Destination Start Address Bit 2	BitBLT Destination Start Address Bit 1	BitBLT Destination Start Address Bit 0

BitBLT Destination Start Address Register 1
REG[109h] RW

BitBLT Destination Start Address Bit 15	BitBLT Destination Start Address Bit 14	BitBLT Destination Start Address Bit 13	BitBLT Destination Start Address Bit 12	BitBLT Destination Start Address Bit 11	BitBLT Destination Start Address Bit 10	BitBLT Destination Start Address Bit 9	BitBLT Destination Start Address Bit 8

BitBLT Destination Start Address Register 2
REG[10Ah] RW

n/a	n/a	n/a	BitBLT Destination Start Address Bit 20	BitBLT Destination Start Address Bit 19	BitBLT Destination Start Address Bit 18	BitBLT Destination Start Address Bit 17	BitBLT Destination Start Address Bit 16

REG[108h] bits 7-0 BitBLT Destination Start Address Bits [20:0]

REG[109h] bits 7-0 A 21-bit register that specifies the destination start address for the BitBLT operation.

REG[10Ah] bits 4-0

BitBLT Memory Address Offset Register 0
REG[10Ch] RW

BitBLT Memory Address Offset Bit 7	BitBLT Memory Address Offset Bit 6	BitBLT Memory Address Offset Bit 5	BitBLT Memory Address Offset Bit 4	BitBLT Memory Address Offset Bit 3	BitBLT Memory Address Offset Bit 2	BitBLT Memory Address Offset Bit 1	BitBLT Memory Address Offset Bit 0

BitBLT Memory Address Offset Register 1
REG[10Dh] RW

n/a	n/a	n/a	n/a	n/a	BitBLT Memory Address Offset Bit 10	BitBLT Memory Address Offset Bit 9	BitBLT Memory Address Offset Bit 8

REG[10Ch] bits 7-0 BitBLT Memory Address Offset Bits [10:0]

REG[10Dh] bits 2-0 These bits are the display’s 11-bit address offset from the starting word of line *n* to the starting word of line *n + 1*. They are used only for address calculation when the BitBLT is configured as a rectangular region of memory. They are not used for the displays.

BitBLT Width Register 0 REG[110h]							RW
BitBLT Width Bit 7	BitBLT Width Bit 6	BitBLT Width Bit 5	BitBLT Width Bit 4	BitBLT Width Bit 3	BitBLT Width Bit 2	BitBLT Width Bit 1	BitBLT Width Bit 0

BitBLT Width Register 1 REG[111h]							RW
n/a	n/a	n/a	n/a	n/a	n/a	BitBLT Width Bit 9	BitBLT Width Bit 8

REG[110h] bits 7-0 BitBLT Width Bits [9:0]

REG[111h] bits 1-0 A 10-bit register that specifies the BitBLT width in pixels - 1.

BitBLT width in pixels = (ContentsOfThisRegister) + 1

Note: The BitBLT operations Pattern Fill with ROP and Pattern Fill with transparency require a BitBLT width > 2 for 8 bpp color depths and a BitBLT width > 1 for 16 bpp color depths.

BitBLT Height Register 0 REG[112h]							RW
BitBLT Height Bit 7	BitBLT Height Bit 6	BitBLT Height Bit 5	BitBLT Height Bit 4	BitBLT Height Bit 3	BitBLT Height Bit 2	BitBLT Height Bit 1	BitBLT Height Bit 0

BitBLT Height Register 1 REG[113h]							RW
n/a	n/a	n/a	n/a	n/a	n/a	BitBLT Height Bit 9	BitBLT Height Bit 8

REG[112h] bits 7-0 BitBLT Height Bits [9:0]

REG[113h] bits 1-0 A 10-bit register that specifies the BitBLT height in lines - 1.

BitBLT height in lines = (ContentsOfThisRegister) + 1

BitBLT Background Color Register 0 REG[114h]							RW
BitBLT Background Color Bit 7	BitBLT Background Color Bit 6	BitBLT Background Color Bit 5	BitBLT Background Color Bit 4	BitBLT Background Color Bit 3	BitBLT Background Color Bit 2	BitBLT Background Color Bit 1	BitBLT Background Color Bit 0

BitBLT Background Color Register 1 REG[115h]							RW
BitBLT Background Color Bit 15	BitBLT Background Color Bit 14	BitBLT Background Color Bit 13	BitBLT Background Color Bit 12	BitBLT Background Color Bit 11	BitBLT Background Color Bit 10	BitBLT Background Color Bit 9	BitBLT Background Color Bit 8

REG[114h] bits 7-0 BitBLT Background Color Bits [15:0]

REG[115h] bits 15-8 A 16-bit register that specifies the BitBLT background color for Color Expansion or key color for Transparent BitBLTBitBLT. For 16 bpp color depths (REG[101h] bit 0 = 1), all 16 bits are used. For 8 bpp color depths (REG[101h] bit 0 = 0), only bits 7-0 are used.

BitBLT Foreground Color Register 0 REG[118h]							RW
BitBLT Foreground Color Bit 7	BitBLT Foreground Color Bit 6	BitBLT Foreground Color Bit 5	BitBLT Foreground Color Bit 4	BitBLT Foreground Color Bit 3	BitBLT Foreground Color Bit 2	BitBLT Foreground Color Bit 1	BitBLT Foreground Color Bit 0

BitBLT Foreground Color Register 1 REG[119h]							RW
BitBLT Foreground Color Bit 15	BitBLT Foreground Color Bit 14	BitBLT Foreground Color Bit 13	BitBLT Foreground Color Bit 12	BitBLT Foreground Color Bit 11	BitBLT Foreground Color Bit 10	BitBLT Foreground Color Bit 9	BitBLT Foreground Color Bit 8

REG[118h] bits 7-0 BitBLT Foreground Color Bits [15:0]

REG[119h] bits 7-0 A 16-bit register that specifies the BitBLT foreground color for Color Expansion or Solid Fill. For 16 bpp color depths (REG[101h] bit 0 = 1), all 16 bits are used. For 8 bpp color depths (REG[101h] bit 0 = 0), only bits 7-0 are used.

8.4.13 Look-Up Table Registers

Note: Accessing the LCD Look-Up Table (LUT) requires an active LCD PCLK and accessing the CRT/TV LUT requires an active CRT/TV PCLK. For further information on the clocks, see Section 7, “Clocks” on page 75.

Look-Up Table Mode Register REG[1E0h]							RW
n/a	n/a	n/a	n/a	n/a	n/a	LUT Mode Bit 1	LUT Mode Bit 0

bits 1-0 Look-Up Table Mode Bits [1:0]
These bits determine which of the on-chip Look-Up Tables (LUT) (LCD and CRT/TV) are accessible by REG[1E2h] and REG[1E4h].

Table 8-33 LUT Mode Selection

LUT Mode Bits [1:0]	Read	Write
00	LCD LUT	LCD and CRT/TV LUT’s
01	LCD LUT	LCD LUT
10	CRT/TV LUT	CRT/TV LUT
11	Reserved	Reserved

Look-Up Table Address Register REG[1E2h]							RW
LUT Address Bit 7	LUT Address Bit 6	LUT Address Bit 5	LUT Address Bit 4	LUT Address Bit 3	LUT Address Bit 2	LUT Address Bit 1	LUT Address Bit 0

bits 7-0 LUT Address Bits [7:0]
These 8 bits control a pointer into the Look-Up Tables (LUT). The S1D13806 has three 256-position, 4-bit wide LUTs, one for each of red, green, and blue – refer to Section 12, “Look-Up Table Architecture” on page 141 for details.

This register selects which LUT entry is read/write accessible through the LUT Data Register (REG[1E4h]). Writing the LUT Address Register automatically sets the pointer to the Red LUT. Accesses to the LUT Data Register automatically increment the pointer.

For example, writing a value 03h into the LUT Address Register sets the pointer to R[3]. A subsequent access to the LUT Data Register accesses R[3] and moves the pointer onto G[3]. Subsequent accesses to the LUT Data Register move the pointer onto B[3], R[4], G[4], B[4], R[5], etc.

Note: The RGB data is inserted into the LUT after the Blue data is written, i.e. all three colors must be written before the LUT is updated.

Look-Up Table Data Register REG[1E4h] RW							
LUT Data Bit 3	LUT Data Bit 2	LUT Data Bit 1	LUT Data Bit 0	n/a	n/a	n/a	n/a

bits 7-4 LUT Data Bits [3:0]
This register is used to read/write the RGB Look-Up Tables. This register accesses the entry at the pointer controlled by the Look-Up Table Address register (REG[1E2h]). Accesses to the Look-Up Table Data register automatically increment the pointer.

Note: The RGB data is inserted into the LUT after the Blue data is written, i.e. all three colors must be written before the LUT is updated.

8.4.14 Power Save Configuration Registers

For further information on Power Save Mode, refer to Section 19, “Power Save Mode” on page 181.

Power Save Configuration Register REG[1F0h] RW							
n/a	n/a	n/a	Reserved	n/a	n/a	n/a	Power Save Mode Enable

bit 4 Reserved.
This bit must be set to 1.

bit 0 Power Save Mode Enable
When this bit = 1, power save mode is enabled.
When this bit = 0, power save mode is disabled.

Note: For details on Power Save Mode, see Section 19, “Power Save Mode” on page 181.

Power Save Status Register REG[1F1h]							RO
n/a	n/a	n/a	n/a	n/a	n/a	LCD Power Save Status	Memory Controller Power Save Status

bit 1 LCD Power Save Status
This bit indicates the power save state of the LCD panel.
When this bit = 1, the panel is powered down.
When this bit = 0, the panel is powered up, or in transition of powering up or down.

Note: When this bit reads a 1, the system may safely shut down the LCD pixel clock source.

Note: When the LCD panel is not enabled (REG[1FCh] bit 0 = 0), this bit returns a 1.

bit 0 Memory Controller Power Save Status
This bit indicates the power save state of the memory controller.
When this bit = 1, the memory controller is powered down and the SDRAM is in self refresh mode.
When this bit = 0, the memory controller is powered up and is in normal mode.

Note: When this bit reads a 1, the system may safely shut down the memory clock source.

8.4.15 Miscellaneous Registers

CPU-to-Memory Access Watchdog Timer Register REG[1F4h]							RW
n/a	n/a	Mem. Access Watchdog Timer bit 5	Mem. Access Watchdog Timer bit 4	Mem. Access Watchdog Timer bit 3	Mem. Access Watchdog Timer bit 2	Mem. Access Watchdog Timer bit 1	Mem. Access Watchdog Timer bit 0

bits 5-0

CPU-to-Memory Access Watchdog Timer Bits [5:0]
A non-zero value in this register enables the watchdog timer for CPU-to-memory access. When enabled, any CPU-to-memory access cycle is completed successfully within a time determined by the following equation.

Maximum CPU-to-memory access cycle time = $(8n + 7) \times T_{bclk} + 13 \times T_{mclk}$

where:

n = A non-zero value in this register
T_{bclk} = Bus clock period, or Bus clock period x 2 (if CONF5 = 1, see Table 4-8 on page 17)
T_{mclk} = Memory clock period

This function is required by some busses which time-out if the cycle duration exceeds a certain time period. This function is **not intended to arbitrarily shorten the CPU-to-memory access cycle time** in order gain higher CPU bandwidth. Doing so may significantly reduce the available display refresh bandwidth which may cause display corruption. This register does not affect CPU-to-register access or BitBLT access.

8.4.16 Common Display Mode Register

Display Mode Register REG[1FCh]							RW
n/a	SwivelView Enable Bit 0	n/a	n/a	n/a	Display Mode Select Bit 2	Display Mode Select Bit 1	Display Mode Select Bit 0

bit 6 SwivelView Enable Bit 0
When this bit = 1, the LCD display image is rotated 90° clockwise.
When this bit = 0, there is no hardware effect.
This bit in conjunction with SwivelView™ Enable Bit 1 achieves the following hardware rotations.

Table 8-34 Setting SwivelView Modes

SwivelView Enable Bits	SwivelView™ Modes			
	Normal	SwivelView 90°	SwivelView 180°	SwivelView 270°
SwivelView Enable Bit 0 (REG[1FCh] bit 6)	0	1	0	1
SwivelView Enable Bit 1 (REG[040h] bit 4)	0	0	1	1

Note: Please refer to Section 15, “SwivelView™” on page 156 for application and limitations.

bits 2-0 Display Mode Select Bits [2:0]
These bits select the display model according to the following table. The LCD display mode is enabled/disabled using bit 0.

Table 8-35 Display Mode Selection

Display Mode Select Bits [2:0]	Display Mode Enabled
000	no display
001	LCD only
010	CRT only
011	EISD (CRT and LCD)
100	TV with flicker filter off
101	EISD (TV with flicker filter off and LCD)
110	TV with flicker filter on
111	EISD (TV with flicker filter on and LCD)

Note: REG[018h] bit 7 must be set to 1 when the flicker filter is enabled.

Note: The **Flicker Filter** reduces the “flickering” effect seen on interlaced displays by averaging adjacent lines on the TV display. This “flickering” is caused by sharp vertical image transitions that occur over one line (1 vertical pixel). For example, one pixel high lines, edges of window boxes, etc. Flickering occurs because these high resolution lines are effectively displayed at half the refresh frequency due to interlacing.

8.5 MediaPlug Registers Descriptions

The S1D13806 has built-in support for Winnov's MediaPlug connection designed for video cameras. The following registers are used to control the connection and accept data from the camera. The MediaPlug registers decode A11-A0 and require A20 = 0 and A12 = 1. The MediaPlug registers are 16-bit wide. Byte access to the MediaPlug registers is not allowed. For further information, see Section 17, "MediaPlug Interface" on page 168.

Note: The MediaPlug control registers must not be accessed while Power Save Mode is enabled (REG[1F0h] bit 0 = 1).

8.5.1 MediaPlug Control Registers

MediaPlug LCMD Register
REG[1000h] RW

LCMD Bit 7	LCMD Bit 6	LCMD Bit 5	LCMD Bit 4	LCMD Bit 3	LCMD Bit 2	LCMD Bit 1	LCMD Bit 0
LCMD Bit 15	LCMD Bit 14	LCMD Bit 13	LCMD Bit 12	LCMD Bit 11	LCMD Bit 10	LCMD Bit 9	LCMD Bit 8

REG[1000h] bits 15-0 MediaPlug LCMD Bits [15:0]

A 16-bit register for setting and detecting various modes of operation of the MediaPlug Local Slave. This register is handled differently for reads and writes. The following table shows the MediaPlug description of the LCMD Register. See bit descriptions for details.

Table 8-36 MediaPlug LCMD Read/Write Descriptions

Data	D15	D14	D13	D12	D11	D10	D9	D8
Write	TO[2:0]		Xxxxxx					
Read	TO[2:0]		00b		Rev[3:0]			

Data	D7	D6	D5	D4	D3	D2	D1	D0
Write	Xxxx				IC	MC	P	W
Read	Rstat[2:0]			0b	IC	MC	P	W

bits 15-14 Timeout Option
These bits select the timeout delay in MediaPlug clock cycles.

Table 8-37 Timeout Option Delay

Timeout Option Bits[15:14]	Timeout (MediaPlug clock cycles)
00	1023 (default)
01	64
10	128
11	64

bits 13-12 A read from these bits always returns 00b.
A write to these bits has no hardware effect.

bits 11-8 MediaPlug IC Revision
The revision for this MediaPlug IC is “0011b”.
A write to these bits has no hardware effect.

bit 7 Cable Detected Status
The cable detected status as determined by the MPD(1) pin.
When this bit = 0, a MediaPlug cable is connected.
When this bit = 1, a MediaPlug cable is not detected.
A write to this bit has no hardware effect.

bit 6 A read from this bit always returns 0b.
A write to this bit has no hardware effect.

bit 5 Remote Powered Status
The remote powered status as determined by the RCTRL pin.
When this bit = 0, the remote is not powered.
When this bit = 1, the remote is powered and connected.
A write to this bit has no hardware effect.

Table 8-38 Cable Detect and Remote Powered Status

Cable Detected Status [bit 7]	Remote Powered Status [bit 5]	Status
0	0	cable connected but remote not powered
0	1	cable connected and remote powered
1	x	cable not connected

bit 4 A read from this bit always returns 0b.
A write to this bit has no hardware effect.

bit 3 MediaPlug Clock Enable
When this bit = 0, the MediaPlug clock is disabled (default).
When this bit = 1, the MediaPlug clock is enabled.

bit 2 MediaPlug Clock
When this bit = 0, the MediaPlug cable clock (VMPCLK) is disabled (default).
When this bit = 1, the MediaPlug cable clock (VMPCLK) is enabled.

bit 1 Power Enable to Remote
When this bit = 0, power to remote is off (default).
When this bit = 1, power to remote is on.

bit 0 Watchdog Disable
When this bit = 0, the MediaPlug watchdog is enabled (default).
When this bit = 1, the MediaPlug watchdog is disabled.

MediaPlug Reserved LCMD Register
REG[1002h] RW

LCMD Bit 23	LCMD Bit 22	LCMD Bit 21	LCMD Bit 20	LCMD Bit 19	LCMD Bit 18	LCMD Bit 17	LCMD Bit 16
LCMD Bit 31	LCMD Bit 30	LCMD Bit 29	LCMD Bit 28	LCMD Bit 27	LCMD Bit 26	LCMD Bit 25	LCMD Bit 24

REG[1002h] bits 15-0 MediaPlug Reserved LCMD Bits [15:0]

This register is not implemented and is reserved for future expansion of the LCMD register. A write to this register has no hardware effect. A read from this register always return 0000h.

MediaPlug CMD Register
REG[1004h] RW

CMD Bit 7	CMD Bit 6	CMD Bit 5	CMD Bit 4	CMD Bit 3	CMD Bit 2	CMD Bit 1	CMD Bit 0
CMD Bit 15	CMD Bit 14	CMD Bit 13	CMD Bit 12	CMD Bit 11	CMD Bit 10	CMD Bit 9	CMD Bit 8

REG[1002h] bits 15-0 MediaPlug CMD Bits [15:0]

A 16-bit register for setting the MediaPlug commands. This register is handled differently for reads and writes. The following table shows the MediaPlug description of the CMD Register. See bit descriptions for details.

Table 8-39 MediaPlug CMD Read/Write Descriptions

Data	D15	D14	D13	D12	D11	D10	D9	D8
Write	I[12:5]							
Read	D	T	I[10:5]					

Data	D7	D6	D5	D4	D3	D2	D1	D0
Write	I[4:0]					C[2:0]		
Read	I[4:0]					C[2:0]		

bit 15 Dirty Bit
This bit is set by the hardware when the command register is written.
It is cleared by hardware by the following conditions:
1. Remote-Reset (After this command has been acknowledged by remote.
2. End_Stream (After this command has been acknowledged by remote.
3. Write to DATA register if the CCC field is Write_Reg.
4. Read to DATA register if the CCC field is Read_Reg.
It is also set when the Remote Machine loses power or the cable is disconnected.

bit 14 Timeout Bit
It is set when Watchdog is enabled and MediaPlug read or write cycle takes longer than 64, 128, 1024 cycles of MediaPlug clock depending on LCMD register settings.
It is also set when the remote is not powered.
It is cleared at the beginning of every command write by the host.

bits 13-3 Index Field
This field is the address presented by the remote to the remote function. MediaPlug transmits the entire 16-bits of the first word of the command Register as written, but I12 (D15) and I11 (D14) are hidden from readback by the dirty bit and Watchdog error bit.

bit 2-0 Command Field
Selects the command as follows:

Table 8-40 MediaPlug Commands

Command Field [bits 2:0]	Command
000	Remote-Reset: Hardware reset of remote.
001	Stream-End: Indicates end of data streaming operation.
010	Write-Register: Write remote register INDEX[5:0] with DATA.
011	Read-Register: Read remote register INDEX[5:0] to DATA.
100	Write_Stream: Begin streaming data to the remote.
101	NOP: The command is sent across the MediaPlug. There is no other effect.
110	NOP: The command is sent across the MediaPlug. There is no other effect.
111	Read-Stream: Begin streaming data from the remote.

MediaPlug Reserved CMD Register
REG[1006h] RW

CMD Bit 23	CMD Bit 22	CMD Bit 21	CMD Bit 20	CMD Bit 19	CMD Bit 18	CMD Bit 17	CMD Bit 16
CMD Bit 31	CMD Bit 30	CMD Bit 29	CMD Bit 28	CMD Bit 27	CMD Bit 26	CMD Bit 25	CMD Bit 24

REG[1006h] bits 15-0 MediaPlug Reserved CMD Bits [15:0]
This register is not implemented and is reserved for future expansion of the CMD register. A write to this register has no hardware effect. A read from this register always return 0000h.

8.5.2 MediaPlug Data Registers

MediaPlug Data Register REG[1008h] to REG[1FFEh], even address RW							
Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
Data Bit 15	Data Bit 14	Data Bit 13	Data Bit 12	Data Bit 11	Data Bit 10	Data Bit 9	Data Bit 8

Data Register bits 15-0 MediaPlug Data Bits [15:0]
A 16-bit register used for read/write and streaming read/write of MediaPlug data. This register is loosely decoded from 1008h to 1FFEh so that the port may be accessed using DWORD block transfer instructions.

8.6 BitBLT Data Registers Descriptions

The BitBLT data registers decode A19-A0 and require A20 = 1. The BitBLT data registers are 16-bit wide. Byte access to the BitBLT data registers is not allowed.

BitBLT Data Register 0 A20-A0 = 100000h-1FFFFEh, even address RW							
Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
Data Bit 15	Data Bit 14	Data Bit 13	Data Bit 12	Data Bit 11	Data Bit 10	Data Bit 9	Data Bit 8

Data Register bits 15-0 BitBLT Data Bits [15:0]
A 16-bit register that specifies the BitBLT data. This register is loosely decoded from 100000h to 1FFFFEh.

9 2D BitBLT Engine

9.1 Overview

The S1D13806 is designed with a built-in 2D BitBLT engine which increases the performance of Bit Block Transfers (BitBLT). It supports 8 and 16 bit-per-pixel color depths.

The BitBLT engine supports rectangular and linear addressing modes for source and destination in a positive direction for all BitBLT operations except the move BitBLT which also supports in a negative direction.

The BitBLT operations support byte alignment of all types. The BitBLT engine has a dedicated BitBLT IO access space allowing it to support multi-tasking applications. This allows the BitBLT engine to support simultaneous BitBLT and CPU read/write operations.

9.2 BitBLT Operations

The S1D13806 2D BitBLT engine supports the following BitBLTs. For detailed information on using the individual BitBLT operations, refer to the S1D13806 Programming Notes and Examples, document number X28B-G-003-xx.

- Write BitBLT.
- Move BitBLT.
- Solid Fill BitBLT.
- Pattern Fill BitBLT.
- Transparent Write BitBLT.
- Transparent Move BitBLT.
- Read BitBLT.
- Color Expansion BitBLT.
- Move BitBLT with Color Expansion.

Note: For details on the BitBLT registers, see Section 8.4.12, “BitBLT Configuration Registers” on page 118.

10 Display Buffer

The system addresses the display buffer using CS#, M/R#, and the input pins AB[20:0]. When CS# = 0 and M/R# = 1, the display buffer is addressed by bits AB[20:0]. See the table below:

Table 10-1 S1D13806 Addressing

CS#	M/R#	Access
0	0	Register access - see Section 8.2, “Register Mapping” on page 80. • REG[000h] is addressed when AB[12:0] = 0 • REG[001h] is addressed when AB[12:0] = 1 • REG[n] is addressed when AB[12:0] = n
0	1	Memory access: the 1.25M byte display buffer is addressed by AB[20:0]
1	X	S1D13806 not selected

The display buffer address space is always 2M bytes. However, the physical display buffer is 1280k bytes. The space above the 1280k boundary is unavailable (see Figure 10-1 “Display Buffer Addressing”).

The display buffer can contain an image buffer, one or more Ink Layer/Hardware Cursor buffers, and a dual panel buffer.

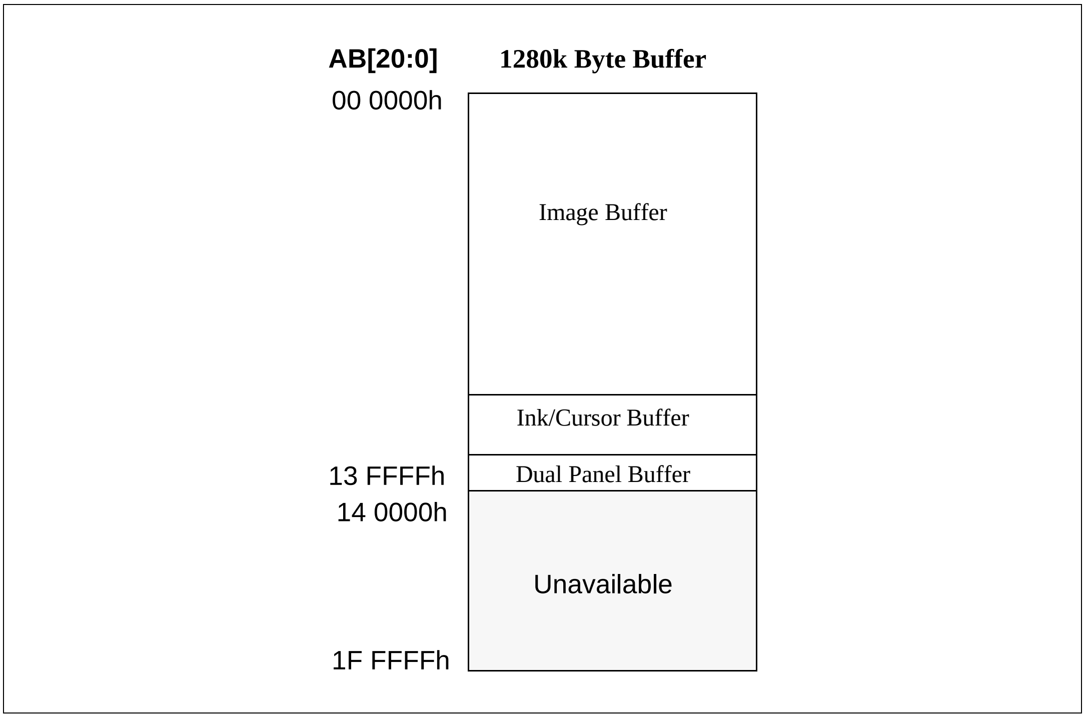


Figure 10-1 Display Buffer Addressing

10.1 Image Buffer

The image buffer contains the formatted display mode data – see Section 11.1, “Display Mode Data Format” on page 139.

The displayed image(s) may occupy only a portion of this space with the remaining area used for multiple images – possibly for animation or general storage. Section 11, “Display Configuration” on page 139 for the relationship between the image buffer and the displayed image.

10.2 Ink Layer/Hardware Cursor Buffers

The Ink Layer/Hardware Cursor buffers contain formatted image data for the Ink Layer and Hardware Cursor. There may be several Ink Layer/Hardware Cursor images stored in the display buffer but only one may be active at any given time.

For further information, see Section 14, “Ink Layer/Hardware Cursor Architecture” on page 152.

10.3 Dual Panel Buffer

In dual panel mode with the dual panel buffer enabled, the top of the display buffer is allocated to the dual panel buffer. The size of the dual panel buffer is a function of the panel resolution and the type of panel (color or monochrome).

Dual Panel Buffer Size (in bytes) = (panel width x panel height) × factor ÷ 16

where factor: = 4 for color panel
= 1 for monochrome panel

Note: Calculating the size of the dual panel buffer is required to avoid overwriting the Hardware Cursor/Ink Layer buffer.

Example 1

**For a 800 × 600 color panel the dual panel buffer size is 120,000 bytes. With a 1280k byte display buffer, the dual panel buffer resides from
12 2b40h to 13 FFFFh.**

11 DISPLAY CONFIGURATION

11.1 Display Mode Data Format

The following diagram show the display mode data formats for a little endian system.

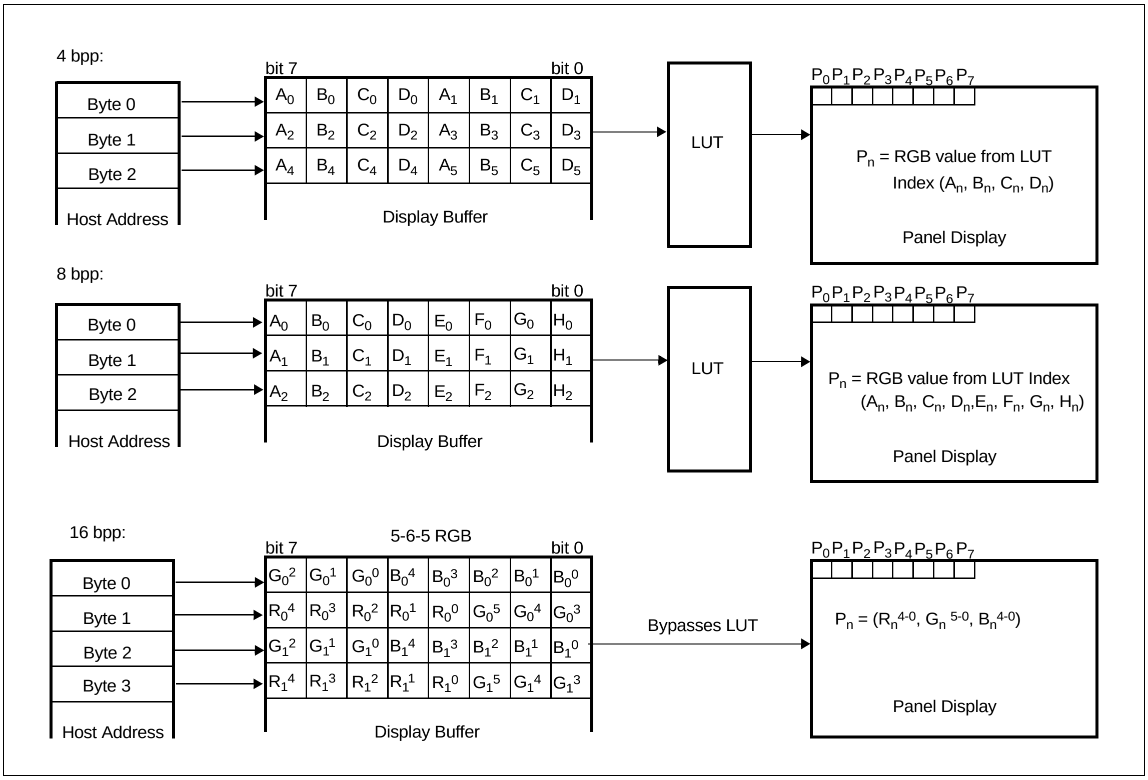

4 bpp:
Byte 0
Byte 1
Byte 2
Host Address
bit 7
bit 0
A_0 B_0 C_0 D_0 A_1 B_1 C_1 D_1
A_2 B_2 C_2 D_2 A_3 B_3 C_3 D_3
A_4 B_4 C_4 D_4 A_5 B_5 C_5 D_5
Display Buffer
LUT
$P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7$
P_n = RGB value from LUT Index (A_n, B_n, C_n, D_n)
Panel Display
8 bpp:
Byte 0
Byte 1
Byte 2
Host Address
bit 7
bit 0
A_0 B_0 C_0 D_0 E_0 F_0 G_0 H_0
A_1 B_1 C_1 D_1 E_1 F_1 G_1 H_1
A_2 B_2 C_2 D_2 E_2 F_2 G_2 H_2
Display Buffer
LUT
$P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7$
P_n = RGB value from LUT Index (A_n, B_n, C_n, D_n,E_n, F_n, G_n, H_n)
Panel Display
16 bpp:
5-6-5 RGB
Byte 0
Byte 1
Byte 2
Byte 3
Host Address
bit 7
bit 0
G_0^2 G_0^1 G_0^0 B_0^4 B_0^3 B_0^2 B_0^1 B_0^0
R_0^4 R_0^3 R_0^2 R_0^1 R_0^0 G_0^5 G_0^4 G_0^3
G_1^2 G_1^1 G_1^0 B_1^4 B_1^3 B_1^2 B_1^1 B_1^0
R_1^4 R_1^3 R_1^2 R_1^1 R_1^0 G_1^5 G_1^4 G_1^3
Display Buffer
Bypasses LUT
$P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7$
$P_n = (R_n^{4-0}, G_n^{5-0}, B_n^{4-0})$
Panel Display

Figure 11-1 4/8/16 Bit-per-pixel Format Memory Organization

Note: 1. The Host-to-Display mapping shown here is for a little endian system.

2. For the 16 bit-per-pixel format, R_n, G_n, B_n represent the red, green, and blue color components.

11.2 Image Manipulation

The figure below shows how the screen image is stored in the image buffer and positioned on the LCD display. The screen image on the CRT/TV is manipulated similarly. When EISD is enabled (see Section 16, “EPSON Independent Simultaneous Display (EISD)” on page 166), the images on the LCD and on the CRT/TV are independent of each other.

- For LCD: (REG[047h], REG[046h]) define the width of the virtual image.
  For CRT/TV: (REG[067h], REG[066h]) define the width of the virtual image.
- For LCD: (REG[044h], REG[043h], REG[042h]) define the starting word of the displayed image.
  For CRT/TV: (REG[064h], REG[063h], REG[062h]) define the starting word of the displayed image.
- For LCD: REG[048h] defines the starting pixel within the starting word.
  For CRT/TV: REG[068h] defines the starting pixel within the starting word.
- For LCD: REG[032h] defines the width of the LCD display.
  For CRT/TV: REG[050h] defines the width of the CRT/TV display.
- For LCD: (REG[039h], REG[038h]) define the height of the LCD display.
  For CRT/TV: (REG[057h], REG[056h]) define the height of the CRT/TV display.

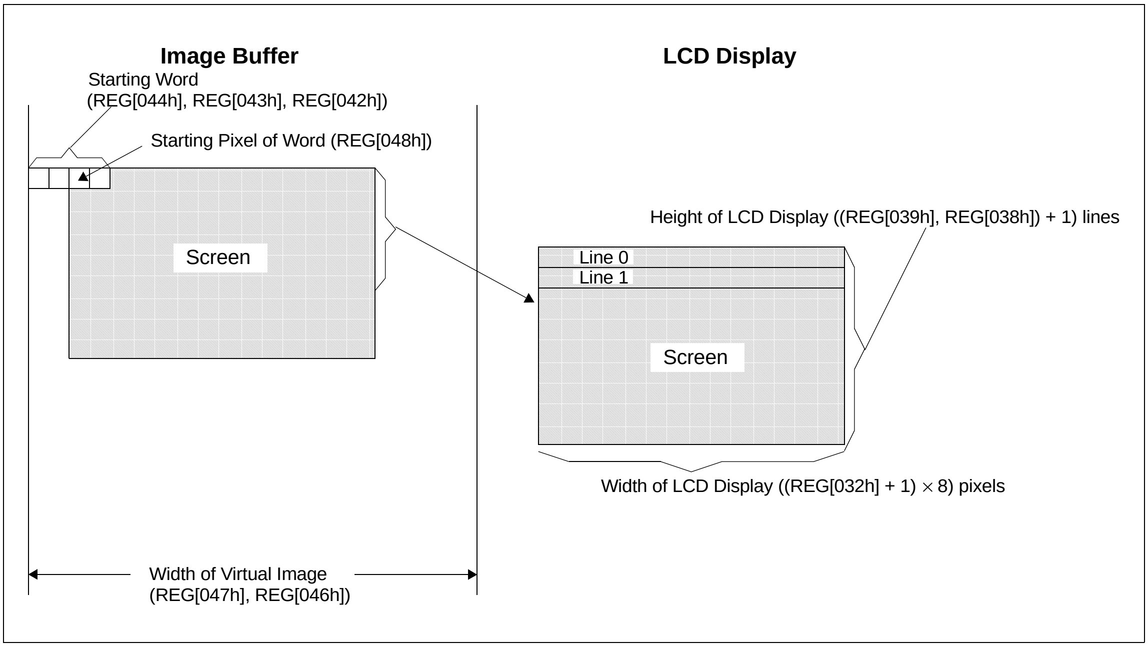


Image Buffer
LCD Display
Starting Word
(REG[044h], REG[043h], REG[042h])
Starting Pixel of Word (REG[048h])
Screen
Height of LCD Display ((REG[039h], REG[038h]) + 1) lines
Line 0
Line 1
Screen
Width of LCD Display ((REG[032h] + 1) × 8) pixels
Width of Virtual Image
(REG[047h], REG[046h])

Figure 11-2 Image Manipulation

12 Look-Up Table Architecture

The following depictions are intended to show the display data output path only.

12.1 Monochrome Modes

The green LUT is used for all monochrome modes.

4 Bit-Per-Pixel Monochrome Mode

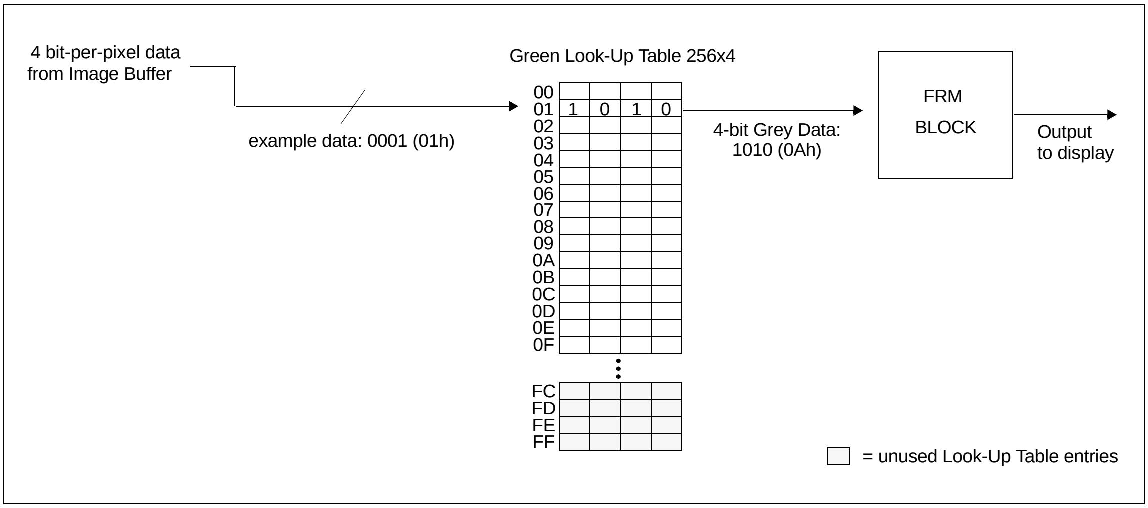


Figure 12-1 4 Bit-Per-Pixel Monochrome Mode Data Output Path

16 Bit-Per-Pixel Monochrome Mode

A color depth of 16 bpp is required to achieve 64 gray shades in monochrome mode. In this mode the LUT is bypassed and the green component of the pixel is mapped to the FRM.

12.2 Color Modes

4 Bit-Per-Pixel Color Mode

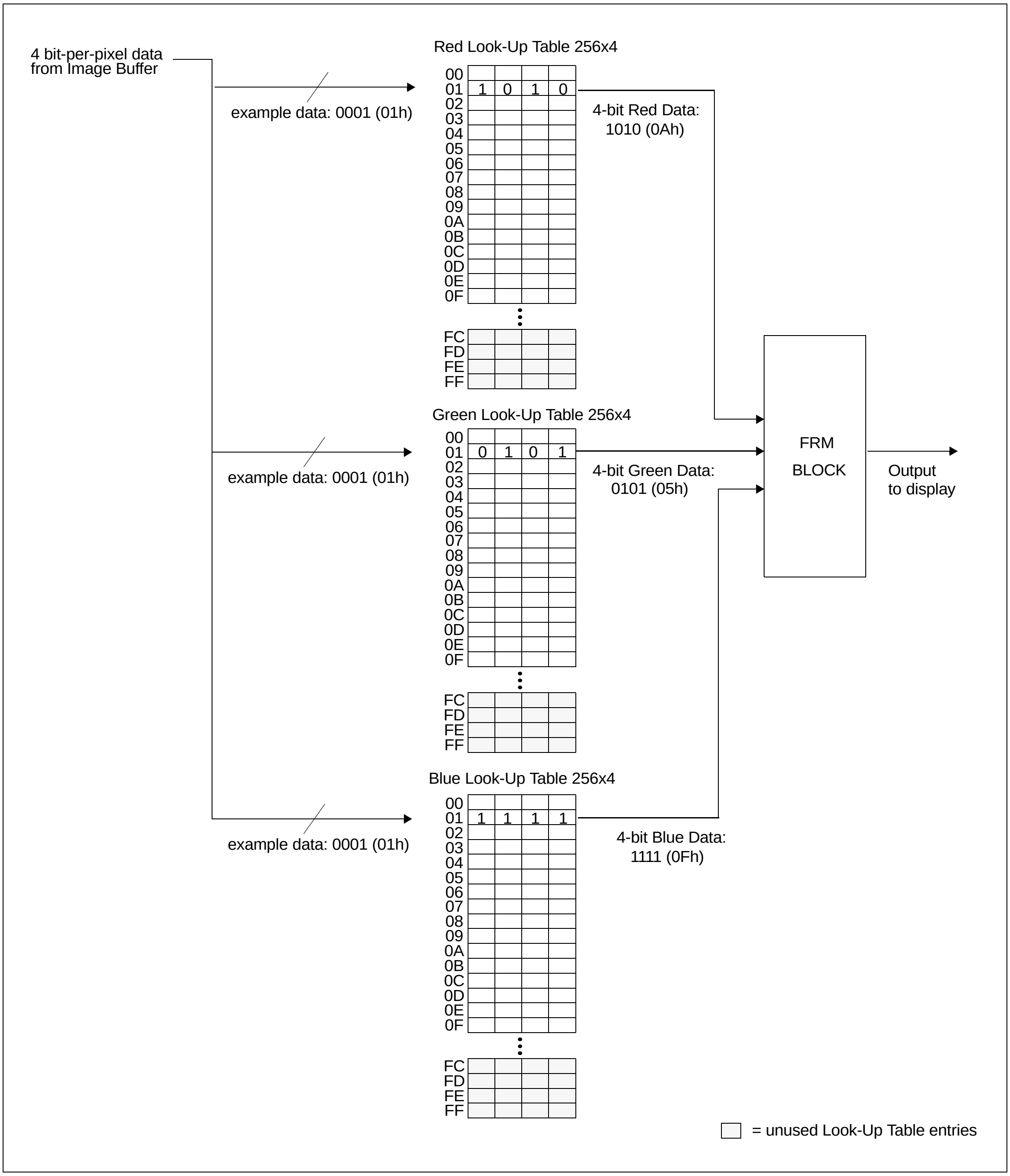

4 bit-per-pixel data from Image Buffer
Red Look-Up Table 256x4
example data: 0001 (01h)
4-bit Red Data: 1010 (0Ah)
Green Look-Up Table 256x4
example data: 0001 (01h)
4-bit Green Data: 0101 (05h)
Blue Look-Up Table 256x4
example data: 0001 (01h)
4-bit Blue Data: 1111 (0Fh)
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
FC FD FE FF
1 0 1 0
0 1 0 1
1 1 1 1
FRM BLOCK
Output to display
= unused Look-Up Table entries

Figure 12-2 4 Bit-Per-Pixel Color Mode Data Output Path

8 Bit-Per-Pixel Color Mode

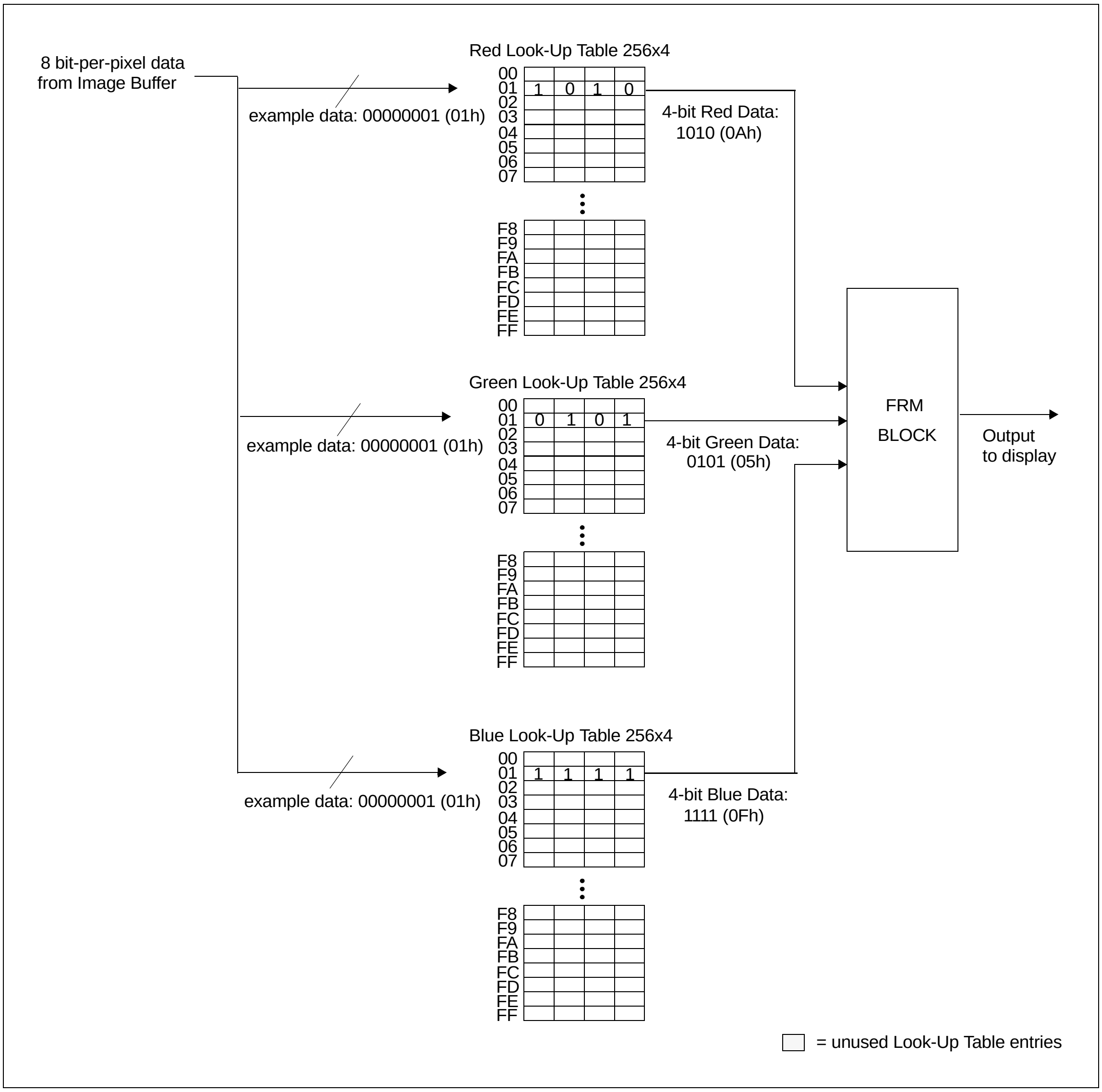


Figure 12-3 8 Bit-Per-Pixel Color Mode Data Output Path

16 Bit-Per-Pixel Color Modes

The LUT is bypassed and the color data is directly mapped for this color mode – Section 11, “Display Configuration” on page 139.

13 TV CONSIDERATIONS

13.1 NTSC/PAL Operation

NTSC or PAL video is supported in either composite or S-video format. Filters may be enabled to reduce the distortion associated with displaying high resolution computer images on an interlaced TV display. The image can be vertically and horizontally positioned on the TV. Additionally, a dedicated Hardware Cursor (independent from the LCD display) is supported.

13.2 Clock Source

The required clock frequencies for NTSC/PAL are given in the following table.

Table 13-1 Required Clock Frequencies for NTSC/PAL

TV Format	Required Clock Frequency
NTSC	14.318180 MHz (3.579545 MHz subcarrier)
PAL	17.734475 MHz (4.43361875 MHz subcarrier)

13.3 Filters

When displaying computer images on a TV, several image distortions are likely to arise:

- cross-luminance distortion.
- cross-chrominance distortion.
- flickering.

These distortions are caused by the high-resolution nature of computer images which typically contain sharp color transitions, and sharp luminance transitions (e.g., high contrast one pixel wide lines and fonts, window edges, etc.). Three filters are available to reduce these distortions.

13.3.1 Chrominance Filter (REG[05Bh] bit 5)

The chrominance filter adjusts the color of the TV by limiting the bandwidth of the chrominance signal (reducing cross-luminance distortion). This reduces the “ragged edges” seen at boundaries between sharp color transitions. This filter is controlled using REG[05Bh] bit 5 and is most useful for composite video output.

13.3.2 Luminance Filter (REG[05Bh] bit 4)

The luminance filter adjusts the brightness of the TV by limiting the bandwidth of the luminance signal (reducing cross-chrominance distortion). This reduces the “rainbow-like” colors at boundaries between sharp luminance transitions. This filter is controlled using REG[05Bh] bit 4 and is most useful for composite video output.

13.3.3 Anti-flicker Filter (REG[1FCh] bits [2:1])

The “flickering” effect seen on interlaced displays is caused by sharp vertical image transitions that occur over one line (1 vertical pixel). For example, one pixel high lines, edges of window boxes, etc. Flickering occurs because these high resolution lines are effectively displayed at half the refresh frequency due to interlacing. The anti-flicker filter averages adjacent lines on the TV display to reduce flickering. This filter is controlled using the Display Mode register (REG[1FCh] bits [2:1]).

13.4 TV Output Levels

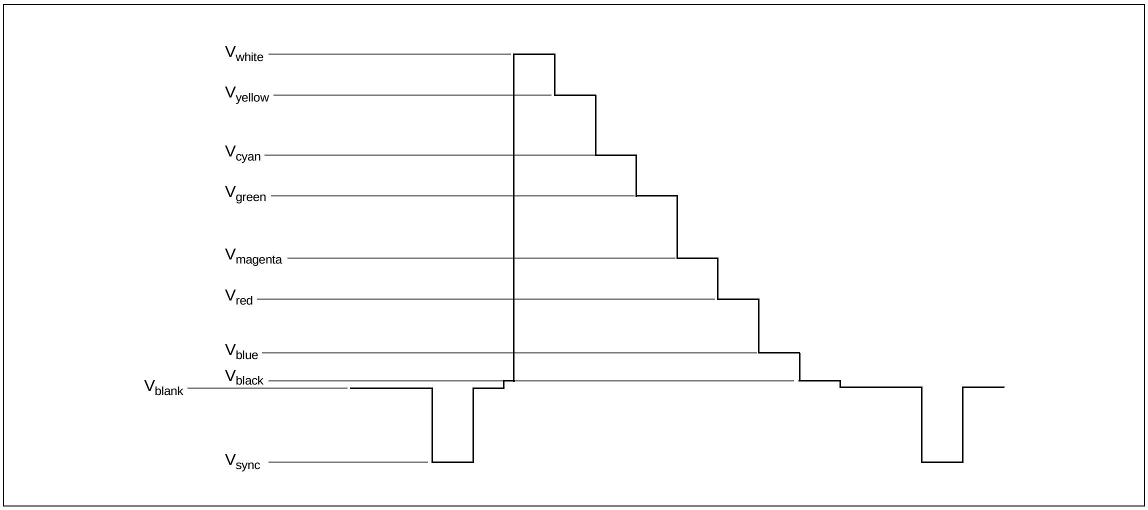


Figure 13-1 NTSC/PAL SVideo-Y (Luminance) Output Levels

Table 13-2 NTSC/PAL SVideo-Y (Luminance) Output Levels

Symbol	Parameter	RGB	NTSC / PAL (mv)	NTSC / PAL (IRE)
V_{white}	White	1F 3F 1F	996	99.5
V_{yellow}	Yellow	1F 3F 00	923	89
V_{cyan}	Cyan	00 3F 1F	798	72
V_{green}	Green	00 3F 00	725	62
$V_{magenta}$	Magenta	1F 00 1F	608	45
V_{red}	Red	1F 00 00	536	35
V_{blue}	Blue	00 00 1F	410	17
V_{black}	Black	00 00 00	338	7.3
$V_{blanking}$	Blanking	N.A.	284	0
V_{sync}	Sync Tip	N.A.	0	-40

Note: RGB values assume a 16 bpp color depth with 5-6-5 pixel packing.

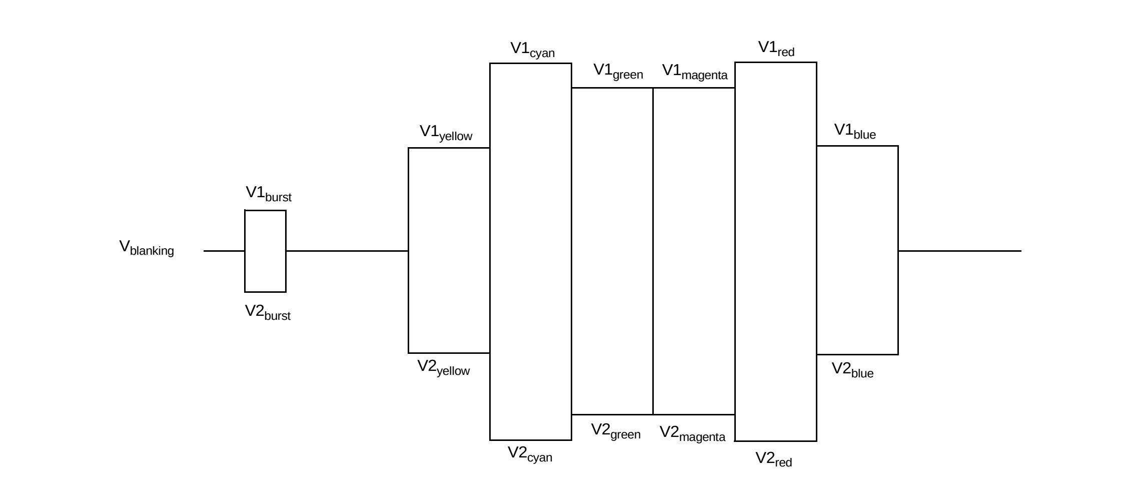


Figure 13-2 NTSC/PAL SVideo-C (Chrominance) Output Levels

Table 13-3 NTSC/PAL SVideo-C (Chrominance) Output Levels

Symbol	Parameter	RGB	NTSC / PAL (mv)	NTSC / PAL (IRE)
$V1_{burst}$	Burst positive peak	N.A.	552 / 541	20 / 18.5
$V1_{yellow}$	Yellow positive peak	1F 3F 00	700	40.8
$V1_{cyan}$	Cyan positive peak	00 3F 1F	815	57
$V1_{green}$	Green positive peak	00 3F 00	751	48
$V1_{magenta}$	Magenta positive peak	1F 00 1F	751	48
$V1_{red}$	Red positive peak	1F 00 00	815	57
$V1_{blue}$	Blue positive peak	00 00 1F	700	40.8
$V_{blanking}$	Blanking	N.A.	410	0
$V2_{burst}$	Burst negative peak	N.A.	268 / 279	-20 / -18.5
$V2_{yellow}$	Yellow negative peak	1F 3F 00	121	-40.8
$V2_{cyan}$	Cyan negative peak	00 3F 1F	5	-57
$V2_{green}$	Green negative peak	00 3F 00	70	-48
$V2_{magenta}$	Magenta negative peak	1F 00 1F	70	-48
$V2_{red}$	Red negative peak	1F 00 00	5	-57
$V2_{blue}$	Blue negative peak	00 00 1F	121	-40.8

Note: RGB values assume a 16 bpp color depth with 5-6-5 pixel packing.

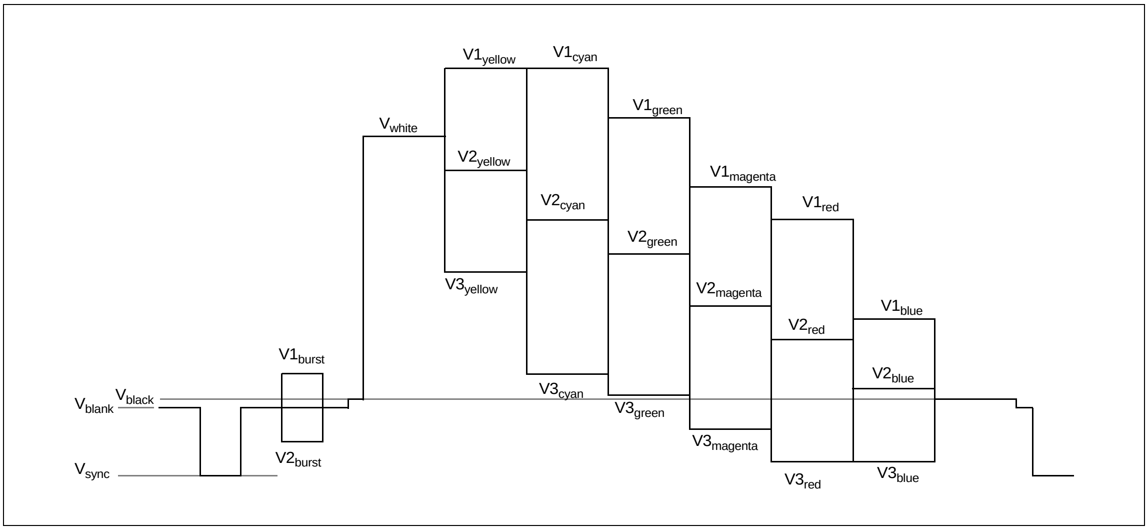


Figure 13-3 NTSC/PAL Composite Output Levels

Table 13-4 NTSC/PAL Composite Output Levels

Symbol	Parameter	RGB	NTSC / PAL (mv)	NTSC / PAL (IRE)
$V1_{yellow}$	Yellow chrominance positive peak	1F 3F 00	1211	130
$V1_{cyan}$	Cyan chrominance positive peak	00 3F 1F	1202	128
$V1_{green}$	Green chrominance positive peak	00 3F 00	1065	109
$V1_{magenta}$	Magenta chrominance positive peak	1F 00 1F	948	93
$V1_{red}$	Red chrominance positive peak	1F 00 00	939	92
$V1_{blue}$	Blue chrominance positive peak	00 00 1F	699	58
V_{white}	White luminance level	1F 3F 1F	995	99
$V2_{yellow}$	Yellow luminance level	1F 3F 00	923	89
$V2_{cyan}$	Cyan luminance level	00 3F 1F	797	72
$V2_{green}$	Green luminance level	00 3F 00	725	62
$V2_{magenta}$	Magenta luminance level	1F 00 1F	608	45
$V2_{red}$	Red luminance level	1F 00 00	535	35
$V2_{blue}$	Blue luminance level	00 00 1F	411	18
V_{black}	Black luminance level	00 00 00	338	7.3
$V3_{yellow}$	Yellow chrominance negative peak	1F 3F 00	634	49
$V3_{cyan}$	Cyan chrominance negative peak	00 3F 1F	392	15
$V3_{green}$	Green chrominance negative peak	00 3F 00	384	14
$V3_{magenta}$	Magenta chrominance negative peak	1F 00 1F	267	-2.6
$V3_{red}$	Red chrominance negative peak	1F 00 00	130	-22
$V3_{blue}$	Blue chrominance negative peak	00 00 1F	122	-23
V_{blank}	Blank Level	N.A.	284	0
$V1_{burst}$	Burst positive peak	N.A.	426 / 415	20 / 18
$V2_{burst}$	Burst negative peak	N.A.	142 / 153	-20 / -19
V_{sync}	Sync Tip	N.A.	0	-40

Note: RGB values assume a 16 bpp color depth with 5-6-5 pixel packing.

13.4.1 TV Image Display and Positioning

This section describes how to setup and position an image to be displayed on a TV. Figure 13-4 “NTSC/PAL Image Positioning” shows an image positioned in the TV display with the related programmable parameters. The TV display area is shaded.

The size of the display image determines the register values for the Horizontal Display Period, Horizontal Non-Display Period, Vertical Display Period, and Vertical Non-Display Period. The maximum and minimum values for these registers are given in Table 13-5, “Minimum and Maximum Values for NTSC/PAL TV,” on page 150. The line period and frame period determined by these registers must also satisfy the following equations.

NTSC:

(((REG[050] bits[6:0]) + 1) × 8) + (((REG[052] bits[5:0]) × 8) + 6) = 910

({(REG[057] bits[1:0]), (REG[056] bits[7:0])} + 1) + ((REG[058] bits[6:0]) + 1) × 2 + 1) = 525

PAL:

(((REG[050] bits[6:0]) + 1) × 8) + (((REG[052] bits[5:0]) × 8) + 7) = 1135

({(REG[057] bits[1:0]), (REG[056] bits[7:0])} + 1) + ((REG[058] bits[6:0]) + 1) × 2 + 1) = 625

The HRTC Start Position and VRTC Start Position registers position the image horizontally and vertically. The maximum and minimum register values for these registers are given in Table 13-5, “Minimum and Maximum Values for NTSC/PAL TV”. Increasing the HRTC Start Position moves the image left, while increasing the VRTC Start Position moves the image up.

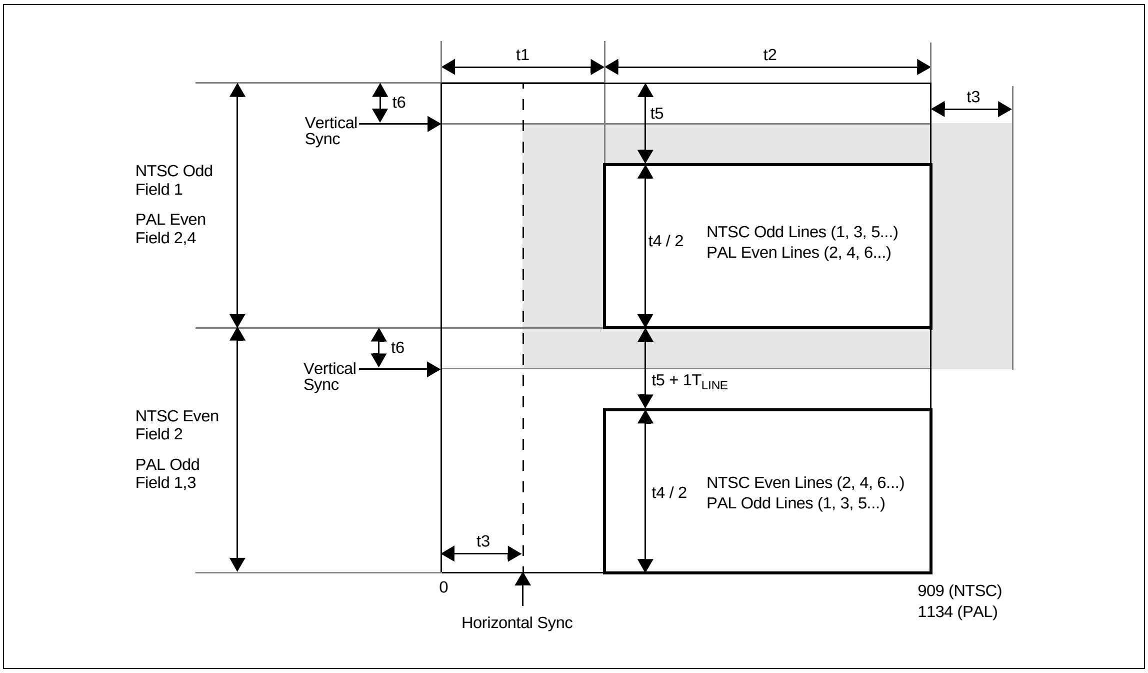


Figure 13-4 NTSC/PAL Image Positioning

The maximum Horizontal and Vertical Display Widths shown in Table 13-5, “Minimum and Maximum Values for NTSC/PAL TV” include display areas that are normally hidden by the edges of the TV. The visible display dimensions are shown in Figure 13-5 “Typical Display Dimensions and Visible Display Dimensions for NTSC and PAL” as a guideline. The actual visible display area for a particular television may differ slightly from those dimensions given. Table 13-6, “Register Values for Example NTSC/PAL Images” lists some register values for some example images.

Table 13-5 Minimum and Maximum Values for NTSC/PAL TV

Symbol	Parameter	Register(s)	NTSC		PAL		Units
			min	max	min	max	
t1	TV Horizontal Non-Display Period	52	158	510	215	511	T_{4SC}
t2	TV Horizontal Display Width	50	400	752	624	920	T_{4SC}
t3	TV HRTC Start Position	53	25	t2 - 158	25	t2 - 215	T_{4SC}
t4	TV Vertical Display Height	57, 56	270	484	370	572	T_{LINE}
t5	TV Vertical Non-Display Period	58	20 (21)	127 (128)	26 (27)	127 (128)	T_{LINE}
t6	TV Vertical Start Position	59	0	t5 - 20	0	t5 - 26	T_{LINE}

Note: The TV Vertical Non-Display Period (t5) varies by 1 line depending on the field that it follows.

Note: For NTSC panels the minimum and maximum values will vary for each application.

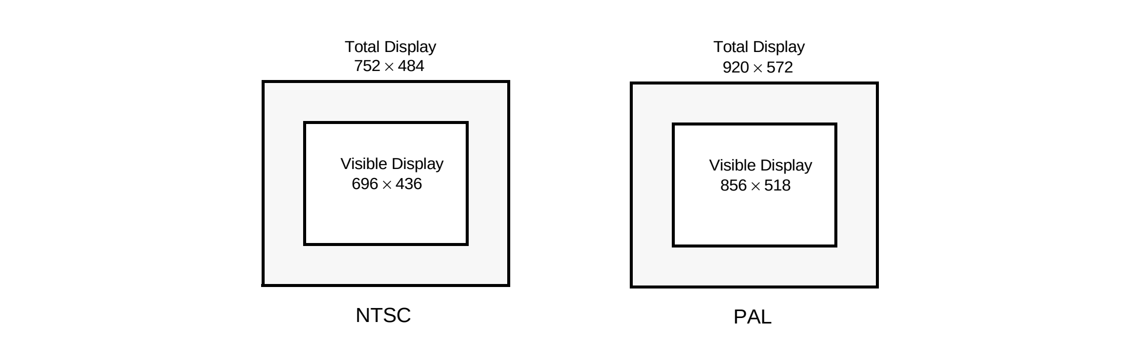


Figure 13-5 Typical Display Dimensions and Visible Display Dimensions for NTSC and PAL

Note: For most implementations, the visible display does not equal the total display. The total display dimensions and the visible display dimensions must be determined for each specific implementation.

Table 13-6 Register Values for Example NTSC/PAL Images

Parameter	Register	NTSC			PAL			
		752 × 484	696 × 436	640 × 480	920 × 572	856 × 518	800 × 572	640 × 480
TV Horizontal Display Width	50	5Dh	56h	4Fh	72h	6Ah	63h	4Fh
TV Horizontal Non-Display Period	52	13h	1Ah	21h	1Ah	22h	29h	3Dh
TV HRTC Start Position	53	02h	04h	09h	02h	05h	09h	15h
TV Vertical Display Height	57	01h	01h	01h	02h	02h	02h	01h
	56	E3h	B3h	DFh	3Bh	05h	3Bh	DFh
TV Vertical Non-Display Period	58	13h	2Bh	15h	19h	34h	19h	47h
TV Vertical Start Position	59	00h	0Ch	00h	00h	0Dh	00h	17h

13.4.2 TV Cursor Operation

See Section 14, “Ink Layer/Hardware Cursor Architecture” on page 152.

14 Ink Layer/Hardware Cursor Architecture

14.1 Ink Layer/Hardware Cursor Buffers

The Ink Layer/Hardware Cursor buffers contain formatted image data for the Ink Layer or Hardware Cursor. There may be several Ink Layer/Hardware Cursor images stored in the display buffer but only one may be active at any given time. The active Ink Layer/Hardware Cursor buffer is selected by the Ink/Cursor Start Address register (REG[071h] for LCD, REG[081h] for CRT/TV). This register defines the start address for the active Ink/Cursor buffer. The Ink/Cursor buffer must be positioned where it does not conflict with the image buffer and dual panel buffer. The start address for the Ink/Cursor buffer is programmed as shown in the following table.

Table 14-1 Ink/Cursor Start Address Encoding

Ink/Cursor Start Address Bits [7:0]	Start Address (Bytes)	Comments
0	1280K - 1024	This default value is suitable for a Hardware Cursor when there is no dual panel buffer.
n = 160...1	1280K - (n × 8192)	These positions can be used to: • position an Ink Layer buffer at the top of the display buffer; • position an Ink Layer buffer between the image and dual panel buffers; • position a Hardware Cursor buffer between the image and dual panel buffers; • select from a multiple of Hardware Cursor buffers.
n = 255...161	Invalid	

The Ink/Cursor image is stored contiguously. The address offset from the starting word of line *n* to the starting word of line *n+1* is calculated as follows:

LCD Ink Address Offset (words) = REG[032h] + 1
CRT/TV Ink Address Offset (words) = REG[050h] + 1
LCD or CRT/TV Cursor Address Offset (words) = 8

14.2 Ink/Cursor Data Format

The Ink/Cursor image is always 2 bit-per-pixel. The following diagram shows the Ink/Cursor data format for a little endian system.

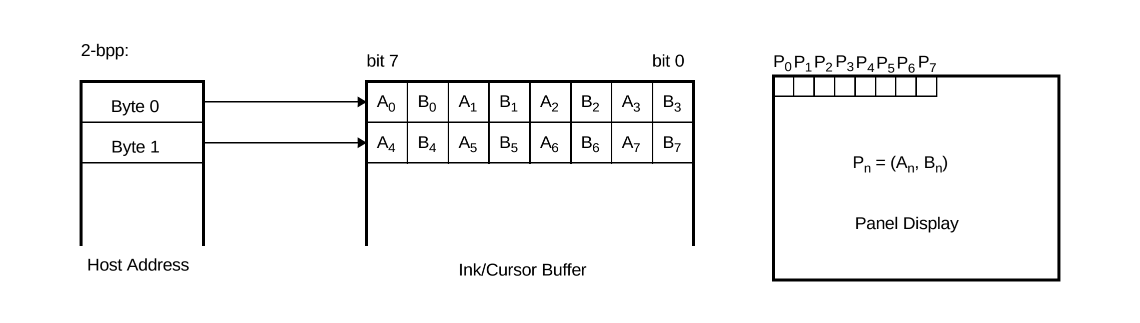

2-bpp:
Byte 0
Byte 1
Host Address
bit 7
bit 0
A_0 B_0 A_1 B_1 A_2 B_2 A_3 B_3
A_4 B_4 A_5 B_5 A_6 B_6 A_7 B_7
Ink/Cursor Buffer
$P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7$
$P_n = (A_n, B_n)$
Panel Display

Figure 14-1 Ink/Cursor Data Format

The image data for pixel n, (A_n,B_n), selects the color for pixel n as follows.

Table 14-2 Ink/Cursor Color Select

(A_n,B_n)	Color	Comments
00	Color 0	Ink/Cursor Color 0 Register, (REG[078h], REG[077h], REG[076h] for LCD, REG[088h], REG[087h], REG[086h] for CRT/TV)
01	Color 1	Ink/Cursor Color 1 Register, (REG[07Ah], REG[07Bh],REG[07Ah] for LCD, REG[08Ah], REG[08Bh], REG[08Ah] for CRT/TV)
10	Background	Ink/Cursor is transparent – show background
11	Inverted Background	Ink/Cursor is transparent – show inverted background

14.3 Ink/Cursor Image Manipulation

14.3.1 Ink Image

The Ink image should always start at the top left pixel (i.e. Cursor X Position and Cursor Y Position registers should always be set to zero). The width and height of the ink image are automatically calculated to completely cover the display.

14.3.2 Cursor Image

The Cursor image size is always 64 x 64 pixels. The Cursor X Position and Cursor Y Position registers specify the position of the top left pixel. The following diagram shows how to position an unclipped cursor.

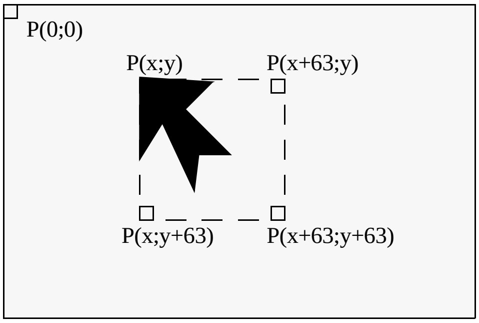


Figure 14-2 Unclipped Cursor Positioning

where

For LCD:
x = (REG[073h] bits [1:0], REG[072h]) **and** REG[073h] bit 7 = 0
y = (REG[075h] bits [1:0], REG[074h]) **and** REG[075h] bit 7 = 0

For CRT/TV:
x = (REG[083h] bits [1:0], REG[082h]) **and** REG[083h] bit 7 = 0
y = (REG[085h] bits [1:0], REG[084h]) **and** REG[085h] bit 7 = 0

The following diagram shows how to position a cursor that is clipped at the top and left sides of the display.

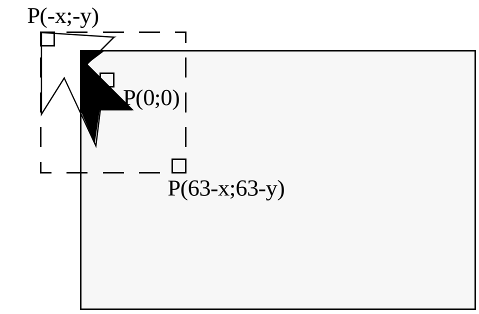


Figure 14-3 Clipped Cursor Positioning

where

For LCD:
x = (REG[073h] bits [1:0], REG[072h]) <= 63 **and** REG[073h] bit 7 = 1
y = (REG[075h] bits [1:0], REG[074h]) <= 63 **and** REG[075h] bit 7 = 1

For CRT/TV:
x = (REG[083h] bits [1:0], REG[082h]) <= 63 **and** REG[083h] bit 7 = 1
y = (REG[085h] bits [1:0], REG[084h]) <= 63 **and** REG[085h] bit 7 = 1

15 SWIVELVIEW™

15.1 Concept

Most computer displays are refreshed in landscape – from left to right and top to bottom. Computer images are stored in the same manner. SwivelView is designed to rotate the displayed image on an LCD by 90°, 180°, or 270° in a clockwise direction. The rotation is done in hardware and is transparent to the user for all display buffer reads and writes. By processing the rotation in hardware, SwivelView offers a performance advantage over software rotation of the displayed image.

15.2 90° SwivelView

90° SwivelView uses a 1024 × 1024 pixel virtual window. The following figures show how the display buffer memory map changes in 90° SwivelView. The display is refreshed in the following sense: C–A–D–B. The application image is written to the S1D13806 in the following sense: A–B–C–D. The S1D13806 rotates and stores the application image in the following sense: C–A–D–B, the same sense as display refresh.

The user can read/write to the display buffer naturally, without the need to rotate the image first in software. The registers that control the panning and scrolling of the panel window are designed for a landscape window. However, it is still possible to pan and scroll the portrait window in 90° SwivelView, but the user must program these registers somewhat differently (See Section 15.2.1, “Register Programming” on page 157).

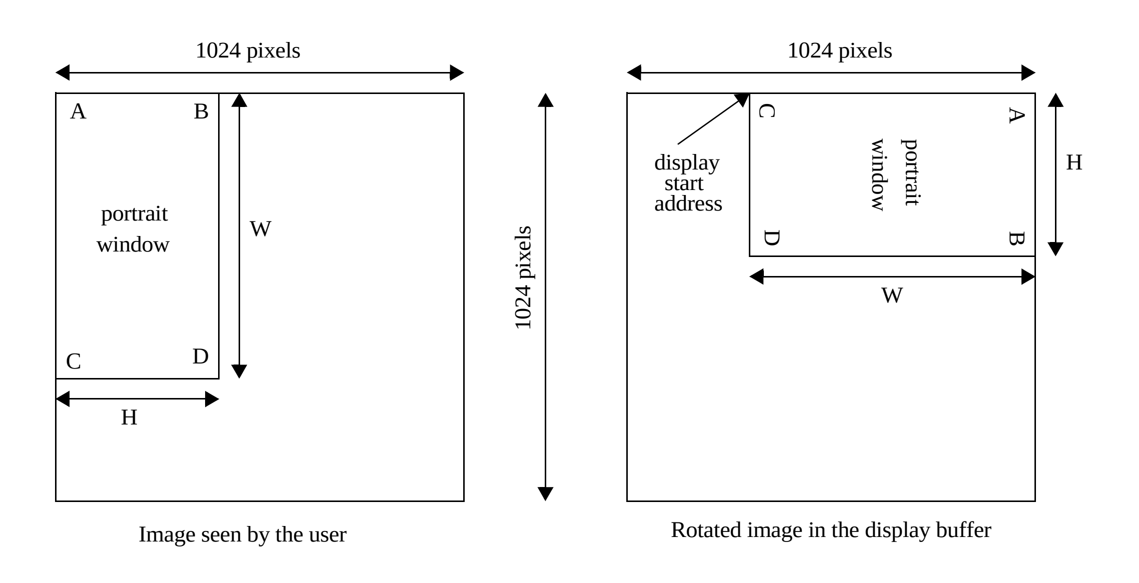


Figure 15-1 Relationship Between Screen Image and 90° Rotated Image in the Display Buffer

Note: W is the width of the LCD panel in number of pixels, (or the height of the portrait window in number of lines).
H is the height of the panel in number of lines, (or the width of the portrait window in number of pixels).

Note: The image must be written with a 1024 pixel offset between adjacent lines (1024 bytes for 8 bpp color depth or 2048 bytes for 16 bpp color depth) and the display start address must be calculated (see below).

15.2.1 Register Programming

Enabling 90° Rotation on CPU Read/Write to Display Buffer

Set SwivelView Enable bits 1:0 to 01b. All CPU accesses to the display buffer are translated to provide 90° clockwise rotation of the display image.

Memory Address Offset

The LCD Memory Address Offset register (REG[046h], REG[047h]) must be set for a 1024 pixel offset:

LCD Memory Address Offset (words)
= 1024 for 16 bpp color depth
= 512 for 8 bpp color depth

Display Start Address

As seen in Figure 15-1 “Relationship Between Screen Image and 90° Rotated Image in the Display Buffer” on page 157, the Display Start Address is determined by the location of the image corner “C”, and it is generally non-zero. The LCD Display Start Address register (REG[042h], REG[043h], REG[044h]) must be set accordingly.

LCD Display Start Address (words)

= (1024 - W)	for 16 bpp color depth
= (1024 - W) / 2	for 8 bpp color depth

where W is the width of the panel in number of pixels.

Horizontal Panning

Horizontal panning is achieved by changing the LCD Display Start Address register:

- Increase/decrease LCD Display Start Address register by 1024 (16 bpp color depth) or 512 (8 bpp color depth) pans the display window to the right/left by 1 pixel.

The amount the display window can be panned to the right is limited to 1024 pixels and limited by the amount of physical memory installed.

Vertical Scrolling

Vertical scrolling is achieved by changing the LCD Display Start Address register and/or the LCD Pixel Panning register:

- Increment/decrement LCD Display Start Address register in 8 bpp color depth scrolls the display window up/down by 2 lines.
- Increment/decrement LCD Display Start Address register in 16 bpp color depth scrolls the display window up/down by 1 line.
- Increment/decrement LCD Pixel Panning register in 8 bpp color depth scrolls the display window up/down by 1 line.

15.2.2 Physical Memory Requirement

Because the user must now deal with a 1024 × 1024 virtual display, the amount of image buffer required for a particular display mode has increased. The minimum amount of image buffer required is:

Minimum Required Image Buffer (bytes)

$$= (1024 \times H) \times 2 \quad \text{for 16 bpp color depth}$$

$$= (1024 \times H) \qquad \text{for 8 bpp color depth}$$

where H is the height of the panel in number of lines.

This minimum amount is required to display a 90° SwivelView image without panning; scrolling, however, is permissible. The degree an image can be panned depends on the amount of physical memory installed and how much of that is used by the dual panel buffer, Ink Layer, or Hardware Cursor. An image cannot be panned outside the 1024 × 1024 virtual display. Often it cannot be panned within the entire virtual display because part of the virtual display memory may be taken up by the dual panel buffer, Ink Layer, Hardware Cursor, or even the CRT/TV display buffer.

The dual panel buffer is used for dual panel mode and has the following memory requirements.

Dual Panel Buffer (bytes)
= (W × H) / 4 for color mode
= (W × H) / 16 for monochrome mode

where W is the width of the panel in number of pixels, and H is the height of the panel in number of lines.

The dual panel buffer is always located at the end of the physical memory.

The Hardware Cursor or Ink Layer also takes up memory. If this memory is > 1KB, it must be located at an 8KB boundary, otherwise it may be located at the last 1KB area. The Hardware Cursor or Ink Layer must not overlap the image buffer or the dual panel buffer.

Even though the virtual display is 1024 × 1024 pixels, the actual panel window is always smaller. Thus it is possible for the display buffer to be smaller than the virtual display but large enough to fit both the required image buffer and the dual panel buffer. This situation limits the maximum “accessible” horizontal virtual size as follows.

Maximum Accessible Horizontal Virtual Size (pixels)
= (Physical Memory – Dual Panel Buffer) / 2048 for 16 bpp color depth
= (Physical Memory – Dual Panel Buffer) / 1024 for 8 bpp color depth

For example, a 800 × 600 TFT panel running a color depth of 16 bpp requires 1200K byte of image buffer and 0K byte of dual panel buffer memory. The virtual display size is 2048 × 2048 = 2M byte. This display can still be supported by the 1280K embedded DRAM even though it is smaller than the 2M byte virtual display because the size of the embedded DRAM is larger than the 1200K byte minimum required image buffer. The maximum accessible horizontal virtual size is = (1280K byte - 0K byte) / 2048 = 640. The programmer therefore has room to pan the portrait window to the right by 640 - 600 = 40 pixels. The programmer also should not read/write to the memory beyond the maximum accessible horizontal virtual size because that memory is either reserved for the dual panel buffer or not associated with any real memory at all.

The following table summarizes the SwivelView memory requirements for different panel sizes and display modes. Note that the S1D13806 memory size is 1280K byte. The calculation of the minimum required image buffer size is based on the image buffer and the dual panel buffer only. As noted in the table, the memory requirements of the Hardware Cursor/Ink Layer are not taken into account. The Hardware Cursor requires 1K byte of memory and the 2-bit Ink Layer requires (W × H) / 4 bytes of memory. Both the Hardware cursor and Ink Layer must reside at 8K byte boundaries, but only one is supported at a time. The following table shows only one possible Hardware Cursor/ Ink Layer location – at the highest possible 8K byte boundary below the dual panel buffer which is always at the top.

Table 15-1 Memory Size Required for SwivelView

Panel Size	Panel Type		Color Depth	Image Buffer Size	Dual Panel Buffer Size	Ink/Cursor Buffer Size	Ink/Cursor Location
320 × 240	Single	Color	8 bpp	240KB	0KB	18.75KB/1KB	1256KB / 1279KB
			16 bpp	480KB			
		Mono	8 bpp	240KB			
			16 bpp	480KB			
640 × 480	Single	Color	8 bpp	480KB	0KB	75KB/1KB	1200KB / 1279KB
			16 bpp	960KB			
		Mono	8 bpp	480KB			
			16 bpp	960KB			
	Dual	Color	8 bpp	480KB	75KB		1128KB / 1200KB
			16 bpp	960KB			
		Mono	8 bpp	480KB	18.75KB		1184KB / 1256KB
			16 bpp	960KB			
800 × 600	Single	Color	8 bpp	600KB	0KB	117.19KB/1KB	1160KB / 1279KB
			16 bpp	1.2MB			— / 1279KB
		Mono	8 bpp	600KB			1160KB / 1279KB
			16 bpp	1.2MB			— / 1279KB
	Dual	Color	8 bpp	600KB	117.19KB		1040KB/1160KB
			16 bpp[1]	1.2MB[1]			
		Mono	8 bpp	600KB	29.30KB		— /1248KB
			16 bpp	1.2MB			

Note: 1. 800 × 600 color 16bpp dual panel is not supported.

Note: Where KB = 1024 bytes, and MB = 1024K bytes.

15.2.3 Limitations

The following limitations apply to 90° SwivelView:

- Only 8/16 bpp color depths are supported – 4 bpp color depth is not supported.
- Hardware cursor and ink images are not rotated – software rotation must be used. SwivelView Enable bit 0 must be set to 0 when the user is accessing the Hardware Cursor or the Ink Layer buffer.
- CRT/TV mode is not supported. SwivelView Enable bit 0 must be set to 0 when the user is accessing the CRT/TV display buffer.
- For 90° SwivelView modes, BitBLT (Bit Block Transfer) operations are still supported. However, the BitBLT data must first be rotated by software. For further information, refer to the *S1D13806 Programmers Notes And Examples*, document number X28B-G-003-xx.

15.3 180° SwivelView

180° SwivelView is where the image is simply **displayed** 180° clockwise rotated. For 180° SwivelView a virtual window is not required and all color depths (4/8/16 bpp) are supported.

15.3.1 Register Programming

Reverse Display Buffer Fetching Address Direction

Set SwivelView Enable bits 1:0 to 10b. During screen refresh, the direction of the address for display buffer fetching is reversed. This setting does not affect CPU to display buffer access in any way.

Display Start Address

The Display Start Address must be programmed to be at the bottom-right corner of the image, since the display is now refreshed in the reverse direction. The LCD Display Start Address register (REG[042h], REG[043h], REG[044h]) must be set accordingly.

LCD Display Start Address (words)

$= (\text{MA\_Offset} \times H) - (\text{MA\_Offset} - W) - 1$ for 16 bpp color depth

$= (\text{MA\_Offset} \times H) - (\text{MA\_Offset} - W/2) - 1$ for 8 bpp color depth

$= (\text{MA\_Offset} \times H) - (\text{MA\_Offset} - W/4) - 1$ for 4 bpp color depth

where H is the height of the panel in number of lines, W is the width of the panel in number of pixels, and MA_Offset is the LCD Memory Address Offset.

Horizontal Panning

Horizontal panning works in the same way as when SwivelView is not enabled, except that the effect of the LCD Pixel Panning register is reversed:

- Increment/decrement LCD Display Start Address register pans the display window to the right/left.
- Increment/decrement LCD Pixel Panning register pans the display window to the left/right.

Vertical Panning

Vertical panning works in the same way as when SwivelView is not enabled:

- Increase/decrease LCD Display Start Address register by one memory address offset scrolls the display window down/up by 1 line.

15.3.2 Limitations

The following limitations apply to 180° SwivelView:

- Hardware Cursor and Ink Layer images are not rotated – software rotation must be used.
- CRT/TV mode is not supported.
- For 180° SwivelView modes, BitBLT (Bit Block Transfer) operations are supported normally. For further information, refer to the "*S1D13806 Programmers Notes And Examples*," document number X28B-G-003-xx.

15.4 270° SwivelView

270° SwivelView is where the image is displayed 270° clockwise rotated. A 1024 × 1024 pixel virtual window is required as in 90° SwivelView. See Figure 15-1 "Relationship Between Screen Image and 90° Rotated Image in the Display Buffer" on page 157.

15.4.1 Register Programming

Enabling 270° Rotation on CPU Read/Write to Display Buffer

Set SwivelView Enable bits 1:0 to 11b.

The LCD Memory Address Offset register (REG[046h], REG[047h]) must be set for a 1024 pixel offset.

LCD Memory Address Offset (words)

= 1024	for 16 bpp color depth
= 512	for 8 bpp color depth

Display Start Address

The Display Start Address must be programmed to be at the bottom-right corner of the image, since the display is now refreshed in the reverse direction. The LCD Display Start Address register (REG[042h], REG[043h], REG[044h]) must be set accordingly.

LCD Display Start Address (words)
$$= ((\text{LCD Memory Address Offset}) \times H) - 1$$

where H is the height of the panel in number of lines.

Horizontal Panning

Horizontal panning is achieved by changing the LCD Display Start Address register. It works in the same way as in 90° SwivelView mode:

- Increase/decrease LCD Display Start Address register by 1024 (16 bpp color depth) or 512 (8 bpp color depth) pans the display window to the right/left by 1 pixel.

The amount the display window can be panned to the right is limited to 1024 pixels and limited by the amount of physical memory installed.

Vertical Scrolling

Vertical scrolling is achieved by changing the LCD Display Start Address register and/or the LCD Pixel Panning register. It works in the same way as in 90° SwivelView mode, except that the effect of the LCD Pixel Panning register is reversed:

- Increment/decrement LCD Display Start Address register in 8 bpp color depth scrolls the display window up/down by 2 lines.
- Increment/decrement LCD Display Start Address register in 16 bpp color depth scrolls the display window up/down by 1 line.
- Increment/decrement LCD Pixel Panning register in 8 bpp color depth scrolls the display window down/up by 1 line.

15.4.2 Physical Memory Requirement

270° SwivelView mode has the same physical memory requirement as in 90° SwivelView mode.

15.4.3 Limitations

The following limitations apply to 270° SwivelView:

- Only 8/16 bpp color depths are supported – 4 bpp color depth is not supported.
- Hardware Cursor and Ink Layer images are not rotated – software rotation must be used. SwivelView Enable bit 0 must be set to 0 when the user is accessing the Hardware Cursor or the Ink Layer memory.
- CRT/TV mode is not supported. SwivelView Enable bit 0 must be set to 0 when the user is accessing the CRT/TV display buffer.
- For 270° SwivelView modes, BitBLT (Bit Block Transfer) operations are still supported. However, the BitBLT data must first be rotated by software. For further information, refer to the "*S1D13806 Programmers Notes And Examples*," document number X28B-G-003-xx.

16 EPSON INDEPENDENT SIMULTANEOUS DISPLAY (EISD)

EPSON Independent Simultaneous Display (EISD) allows the S1D13806 to display independent images on two different displays (LCD panel and CRT or TV).

16.1 Registers

The LCD panel timings and mode setup are programmed through the Panel Configuration Registers (REG[03Xh]) and the LCD Display Mode Registers (REG[04Xh]). The CRT/TV timings and mode setup are programmed through the CRT/TV Configuration Registers (REG[05Xh]) and the CRT/TV Display Mode Registers (REG[06Xh]). The Ink Layer or Hardware Cursor can also be independently controlled on the two displays. The LCD Ink/Cursor Registers (REG[07Xh]) control the Ink/Cursor on the LCD display; the CRT/TV Ink/Cursor Registers (REG[08Xh]) control the Ink/Cursor on the CRT or TV. Each display uses its own Look-Up Table (LUT), although there is only one set of LUT Registers (REG[1E0h], REG[1E2h], REG[1E4h]). Use the LUT Mode Register (REG[1E0h]) to select access to the LCD and/or CRT/TV LUTs.

The pixel clock source for the two displays may be independent. Use the Clock Configuration Registers (REG[014h], REG[018h]) to select the LCD pixel clock source and the CRT/TV pixel clock source, respectively. Typically, CLKI2 is used for the CRT/TV display, while CLKI is used for the LCD display. Memory clock may come from CLKI or BUSCLK.

16.2 Display Mapping

To display different images on the LCD and CRT/TV, the two images should reside in non-overlapping areas of the display buffer, and the display start addresses point to the corresponding areas. The display buffer is mapped to the CPU address AB[20:0] linearly.

The LCD and CRT/TV may display identical images by setting the display start addresses for the LCD and the CRT/TV to the same address. In this case only one image is needed in the display buffer.

16.3 Bandwidth Limitation

When EISD is enabled, the LCD and CRT/TV displays must share the total bandwidth available to the S1D13806. The result is that display modes with a high resolution or color depth may not be supported. In some cases, Ink Layers may not be possible on one or both of the displays. EISD increases the total demand for display refresh bandwidth and reduces CPU bandwidth, resulting in lower CPU performance.

In a few cases when EISD is enabled, the default LCD and CRT/TV Display FIFO High Threshold Control register values are not optimally set, causing display problems with one or both of the displays. This condition may be corrected by adjusting the values of the LCD and CRT/TV Display FIFO High Threshold Control registers (REG[04Ah] for LCD and REG[06Ah] for CRT/TV). See Section 18.2, “Example Frame Rates” on page 172 for required FIFO settings.

Changing this register to a non-zero value sets the high threshold FIFO level to this value. This register may not exceed 59 decimal. The high threshold FIFO level controls how often display fetch requests are issued by the FIFO. In general, a higher high threshold FIFO level increases the bandwidth to that display pipe, and a lower level reduces it.

When the FIFO High Threshold Control register is set to 00h (default), the following settings are used:

- 11h for 4 bpp color depth
- 21h for 8 bpp color depth
- 23h for 16 bpp color depth

Most display problems may be corrected by increasing the associated high threshold FIFO level for that display. However, because the total available bandwidth is fixed, this change may create display problem for the other display. In this case, reducing the high threshold FIFO level for the other display instead may work. Sometimes, a combination of these two methods is required. Correcting EISD display problems by adjusting the FIFO High Threshold Control registers is mostly a trial-and-error process.

Note: While the user is free to experiment with these registers, recommended FIFO level settings for some of the more common EISD modes requiring non-default FIFO level settings are listed in Section 18.2, “Example Frame Rates” on page 172.

17 MediaPlug Interface

Winnov's MediaPlug Slave interface has been incorporated into the S1D13806. The MediaPlug Slave follows the "*Specification For Winnov MediaPlug Slave, Local module*," Document Rev 0.3 with the following exceptions.

17.1 Revision Code

The MediaPlug Slave Revision Code can be determined by reading bits 11:8 of the LCMD register. The revision code for this implementation is 0011b.

17.2 How to enable the MediaPlug Slave

The MediaPlug Slave interface is enabled/disabled at the rising edge of RESET# by the state of CONF7. When CONF7 is set to 1, the MediaPlug functionality is enabled and GPIO12 is configured as the MediaPlug power control output pin (VMPEPWR) - see Table 4-8, "Summary of Power-On/ Reset Options," on page 17.

17.3 MediaPlug Interface Pin Mapping

The S1D13806 provides 8 pins for use by the MediaPlug interface (VMP[7:0]). GPIO12 is also used as the MediaPlug power control output pin (VMPEPWR) when the MediaPlug interface is enabled. The following table lists the MediaPlug pin mapping when the interface is enabled.

Table 17-1 MediaPlug Interface Pin Mapping

S1D13806 Pin Names	IO Type	MediaPlug I/F
VMP0	O	VMPCLKN
VMP1	O	VMPCLK
VMP2	IO	VMPD3
VMP3	IO	VMPD2
VMP4	IO	VMPD1
VMP5	IO	VMPD0
VMP6	I	VMPRCTL
VMP7	O	VMPLCRL
GPIO12	O	VMPEPWR

Note: VMPEPWR is controlled by bit 1 of the MediaPlug LCMD register.

18 CLOCKING

18.1 Frame Rate Calculation

18.1.1 LCD Frame Rate Calculation

The maximum LCD frame rate is calculated using the following formula.

$$max.\ LCD\ Frame\ Rate = \frac{LPCLK_{max}}{(LHDP + LHNDP) \times (LVDP + LVNDP)} \times \frac{1}{n}$$

Where:

LPCLKmax = maximum LCD pixel clock frequency

LVDP = LCD Vertical Display Height
= REG[039h] bits [1:0], REG[038h] bits [7:0] + 1

LVNDP = LCD Vertical Non-Display Period
= REG[03Ah] bits [5:0] + 1

LHDP = LCD Horizontal Display Width
= ((REG[032h] bits [6:0]) + 1) × 8Ts

LHNDP = LCD Horizontal Non-Display Period
= ((REG[034h] bits [4:0]) + 1) × 8Ts

Ts = minimum LCD pixel clock (LPCLK) period

n = 1 for single panel
= 2 for dual panel

18.1.2 CRT Frame Rate Calculation

The maximum CRT frame rate is calculated using the following formula.

$$max.\ CRT\ Frame\ Rate = \frac{CPCLK_{max}}{(CHDP + CHNDP) \times (CVDP + CVNDP)}$$

Where:

CPCLKmax = maximum CRT pixel clock frequency

CVDP = CRT Vertical Display Height
= REG[057h] bits [1:0], REG[056h] bits [7:0] + 1

CVNDP = CRT Vertical Non-Display Period
= REG[058h] bits [6:0] + 1

CHDP = CRT Horizontal Display Width
= ((REG[050h] bits [6:0]) + 1) × 8Ts

CHNDP = CRT Horizontal Non-Display Period
= ((REG[052h] bits [5:0]) + 1) × 8Ts

Ts = minimum CRT pixel clock (CPCLK) period

18.1.3 TV Frame Rate Calculation

The maximum TV frame rate is calculated using the following formula.

$$max.\ TV\ Frame\ Rate = \frac{TPCLK_{max}}{(THDP + THNDP) \times (TVDP + TVNDP + 0.5)}$$

Where:

TPCLKmax = maximum TV pixel clock frequency

TVDP = TV Vertical Display Height
= REG[057h] bits [1:0], REG[056h] bits [7:0] + 1

TVNDP = TV Vertical Non-Display Period
= REG[058h] bits [6:0] + 1

THDP = TV Horizontal Display Width
= ((REG[050h] bits [6:0]) + 1) × 8Ts

THNDP = TV Horizontal Non-Display Period
= for NTSC output use ((REG[052h] bits [5:0]) × 8Ts) + 6
= for PAL output use ((REG[052h] bits [5:0]) × 8Ts) + 7

Ts = minimum TV pixel clock (TPCLK) period

18.2 Example Frame Rates

For all example frame rates the following conditions apply:

- Dual panel buffer is enabled for dual panel.
- TV Flicker Filter is enabled for TV.
- MCLK is 50MHz.

18.2.1 Frame Rates for 640 × 480 with EISD Disabled

Table 18-1 Frame Rates for 640 × 480 with EISD Disabled

LCD Type		Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	max PCLK (MHz)	min HNDP (pixels)	min VNDP (lines)	max Frame Rate (Hz)	CRT/ TV	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	PCLK (MHz)	HNDP (pixels)	VNDP (lines)	Frame Rate (MHz)
Passive Single / TFT		No	640	480	4	40	56	1	119.5	—	—	—	—	—	—	—	—	—
		No	640	480	8	40	64	1	118.1	—	—	—	—	—	—	—	—	—
		No	640	480	16	40	64	1	118.1	—	—	—	—	—	—	—	—	—
Mono Passive Dual		No	640	480	4	40	64	1	235.8	—	—	—	—	—	—	—	—	—
		No	640	480	8	40	72	1	233.1	—	—	—	—	—	—	—	—	—
		No	640	480	16	31	72	1	181.0	—	—	—	—	—	—	—	—	—
Color Passive Dual		No	640	480	4	40	64	1	235.8	—	—	—	—	—	—	—	—	—
		No	640	480	8	40	72	1	233.1	—	—	—	—	—	—	—	—	—
		No	640	480	16	30	64	1	176.8	—	—	—	—	—	—	—	—	—
Passive Single / TFT		Yes	640	480	4	40	56	1	119.5	—	—	—	—	—	—	—	—	—
		Yes	640	480	8	40	64	1	118.1	—	—	—	—	—	—	—	—	—
		Yes	640	480	16	40	64	1	118.1	—	—	—	—	—	—	—	—	—
Mono Passive Dual		Yes	640	480	4	40	64	1	235.8	—	—	—	—	—	—	—	—	—
		Yes	640	480	8	40	72	1	233.1	—	—	—	—	—	—	—	—	—
		Yes	640	480	16	30	64	1	176.8	—	—	—	—	—	—	—	—	—
Color Passive Dual		Yes	640	480	4	40	64	1	235.8	—	—	—	—	—	—	—	—	—
		Yes	640	480	8	36	72	1	209.8	—	—	—	—	—	—	—	—	—
		Yes	640	480	16	26	56	1	155.0	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	CRT	No	640	480	4	36	192	29	85.0
—	—	—	—	—	—	—	—	—	—	CRT	No	640	480	8	36	192	29	85.0
—	—	—	—	—	—	—	—	—	—	CRT	No	640	480	16	36	192	29	85.0
—	—	—	—	—	—	—	—	—	—	NTSC TV	No	640	480	4	14.32	270	22	62.7
—	—	—	—	—	—	—	—	—	—	NTSC TV	No	640	480	8	14.32	270	22	62.7
—	—	—	—	—	—	—	—	—	—	NTSC TV	No	640	480	16	14.32	270	22	62.7
—	—	—	—	—	—	—	—	—	—	PAL TV	No	640	480	4	17.73	495	72	56.6
—	—	—	—	—	—	—	—	—	—	PAL TV	No	640	480	8	17.73	495	72	56.6
—	—	—	—	—	—	—	—	—	—	PAL TV	No	640	480	16	17.73	495	72	56.6

▨ Example Frame Rates with Ink Layer Enabled

Table 18-1 Frame Rates for 640 × 480 with EISD Disabled (Continued)

LCD Type		Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	max PCLK (MHz)	min HNDP (pixels)	min VNDP (lines)	max Frame Rate (Hz)	CRT/ TV	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	PCLK (MHz)	HNDP (pixels)	VNDP (lines)	Frame Rate (MHz)
—	—	—	—	—	—	—	—	—	—	CRT	Yes	640	480	4	36	192	29	85.0
—	—	—	—	—	—	—	—	—	—	CRT	Yes	640	480	8	36	192	29	85.0
—	—	—	—	—	—	—	—	—	—	CRT	Yes	640	480	16	36	200	20	85.7
—	—	—	—	—	—	—	—	—	—	NTSC TV	Yes	640	480	4	14.32	270	22	62.7
—	—	—	—	—	—	—	—	—	—	NTSC TV	Yes	640	480	8	14.32	270	22	62.7
—	—	—	—	—	—	—	—	—	—	NTSC TV	Yes	640	480	16	14.32	270	22	62.7
—	—	—	—	—	—	—	—	—	—	PAL TV	Yes	640	480	4	17.73	495	72	56.6
—	—	—	—	—	—	—	—	—	—	PAL TV	Yes	640	480	8	17.73	495	72	56.6
—	—	—	—	—	—	—	—	—	—	PAL TV	Yes	640	480	16	17.73	495	72	56.6

▨	Example Frame Rates with Ink Layer Enabled

18.2.2 Frame Rates for 800 × 600 with EISD Disabled

Table 18-2 Frame Rates for 800 × 600 with EISD Disabled

LCD Type		Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	max PCLK (MHz)	min HNDP (pixels)	min VNDP (lines)	max Frame Rate (Hz)	CRT/ TV	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	PCLK (MHz)	HNDP (pixels)	VNDP (lines)	Frame Rate (MHz)
TFT		No	800	600	8	65	232	35	99.2									
		No	800	600	16	44.4	160	35	72.8									
Color Passive Dual		No	800	600	4	40	64	1	153.8	—	—	—	—	—	—	—	—	—
		No	800	600	8	40	72	1	152.4	—	—	—	—	—	—	—	—	—
		No	800	600	16	30	64	1	115.3	—	—	—	—	—	—	—	—	—
TFT		Yes	800	600	8	65	232	35	99.2									
TFT[1]		Yes	800	600	16	40	144	35	66.7									
Color Passive Dual		Yes	800	600	4	38	64	1	146.1	—	—	—	—	—	—	—	—	—
		Yes	800	600	8	36	72	1	137.2	—	—	—	—	—	—	—	—	—
		Yes	800	600	16	26	56	1	100.9	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	CRT	No	800	600	4	40	256	28	60.3
—	—	—	—	—	—	—	—	—	—	CRT	No	800	600	8	40	256	28	60.3
—	—	—	—	—	—	—	—	—	—	CRT	No	800	600	16	40	224	25	62.5
										CRT	No	800	600	4	49.5	256	28	74.6
										CRT	No	800	600	8	49.5	256	28	74.6
										CRT	No	800	600	4	56.25	256	28	84.8
										CRT	No	800	600	8	56.25	256	28	84.8
—	—	—	—	—	—	—	—	—	—	CRT	Yes	800	600	4	40	256	28	60.3
—	—	—	—	—	—	—	—	—	—	CRT	Yes	800	600	8	40	256	28	60.3
—	—	—	—	—	—	—	—	—	—	CRT[2]	Yes	800	600	16	40	224	25	62.5
										CRT	Yes	800	600	4	49.5	256	28	74.6
										CRT	Yes	800	600	8	49.5	256	28	74.6
										CRT	Yes	800	600	4	56.25	256	28	84.8
										CRT	Yes	800	600	8	56.25	256	28	84.8

▨ Example Frame Rates with Ink Layer Enabled

The FIFO values for these display modes must be set as follows:

1. REG[07Eh] = 0Ch.
2. REG[08Eh] = 0Ah.

18.2.3 Frame Rates for 1024 × 768 with EISD Disabled

Table 18-3 Frame Rates for 1024 × 768 with EISD Disabled

LCD Type		Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	max PCLK (MHz)	min HNDP (pixels)	min VNDP (lines)	max Frame Rate (Hz)	CRT/ TV	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	PCLK (MHz)	HNDP (pixels)	VNDP (lines)	Frame Rate (MHz)
TFT		No	1024	768	8	65	160	37	68.2									
TFT[1]		Yes	1024	768	8	65	160	37	68.2									
—	—	—	—	—	—	—	—	—	—	CRT	No	1024	768	4	65	320	41	59.8
—	—	—	—	—	—	—	—	—	—	CRT	No	1024	768	8	65	320	41	59.8
—	—	—	—	—	—	—	—	—	—	CRT	Yes	1024	768	4	65	320	41	59.8
—	—	—	—	—	—	—	—	—	—	CRT	Yes	1024	768	8	65	320	41	59.8

▨ Example Frame Rates with Ink Layer Enabled

The FIFO values for these display modes must be set as follows:

1. REG[07Eh] = 0Ch.

18.2.4 Frame Rates for LCD and CRT (640 × 480) with EISD Enabled

Table 18-4 Frame Rates for LCD and CRT (640 × 480) with EISD Enabled

LCD Type	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	max PCLK (MHz)	min HNDP (pixels)	min VNDP (lines)	max Frame Rate (Hz)	CRT/ TV	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	PCLK (MHz)	HNDP (pixels)	VNDP (lines)	Frame Rate (MHz)
Passive Single	No	320	240	16	17	64	1	183.7	CRT	No	640	480	16	25.18	160	44	60.1
	No	640	240	16	17	64	1	100.2	CRT	No	640	480	16	25.18	160	44	60.1
	No	640	480	4	40	112	1	110.6	CRT	No	640	480	4	25.18	160	44	60.1
	No	640	480	8	40	144	1	106.1	CRT	No	640	480	8	25.18	160	44	60.1
Color Passive Dual	No	640	480	16	12	56	1	71.5	CRT	No	640	480	16	25.18	160	44	60.1
TFT	No	800	600	8	41	144	26	69.4	CRT	No	640	480	8	25.18	160	44	60.1
TFT	No	1024	768	8	41	112	37	44.8	CRT	No	640	480	8	25.18	160	44	60.1
Color Passive Dual	No	800	600	4	38	120	1	137.2	CRT	No	640	480	4	25.18	160	44	60.1
	No	800	600	8	27.6	104	1	101.4	CRT	No	640	480	8	25.18	160	44	60.1
Passive Single	Yes	640	480	4	40	112	1	110.6	CRT	No	640	480	4	25.18	160	44	60.1
	Yes	640	480	8	31.6	112	1	87.4	CRT	No	640	480	8	25.18	160	44	60.1
TFT	Yes	800	600	8	31.6	112	26	55.4	CRT	No	640	480	8	25.18	160	44	60.1
Mono Passive Dual	Yes	640	480	8	27.6	104	1	153.9	CRT	No	640	480	8	25.18	160	44	60.1
Color Passive Dual	Yes	640	480	8	23.2	88	1	132.2	CRT	No	640	480	8	25.18	160	44	60.1
Color Passive Dual[1]	Yes	640	480	16	11.6	48	1	70.0	CRT	No	640	480	16	25.18	160	44	60.1
Color Passive Dual	Yes	800	600	8	23.2	88	1	86.8	CRT	No	640	480	8	25.18	160	44	60.1
Passive Single[2]	No	640	240	16	13.7	56	1	81.7	CRT	Yes	640	480	16	25.18	160	44	60.1
Passive Single	No	640	480	8	31.7	112	1	87.6	CRT	Yes	640	480	8	25.18	160	44	60.1
Mono Passive Dual[2]	No	640	480	8	27.1	104	1	151.1	CRT	Yes	640	480	8	25.18	160	44	60.1
	No	640	480	16	11	48	1	66.4	CRT	Yes	640	480	16	25.18	160	44	60.1
Color Passive Dual	No	640	480	8	19	88	1	108.3	CRT	Yes	640	480	8	25.18	160	44	60.1
Color Passive Dual	No	800	600	4	33.2	104	1	122.0	CRT	Yes	640	480	4	25.18	160	44	60.1
Color Passive Dual[2]	No	800	600	8	22.6	88	1	84.6	CRT	Yes	640	480	8	25.18	160	44	60.1
Passive Single[3]	Yes	640	240	16	11.8	48	1	71.2	CRT	Yes	640	480	16	25.18	160	44	60.1
Passive Single	Yes	640	480	8	25.7	96	1	72.6	CRT	Yes	640	480	8	25.18	160	44	60.1
Mono Passive Dual[2]	Yes	640	480	8	22.6	88	1	128.8	CRT	Yes	640	480	8	25.18	160	44	60.1
Color Passive Dual	Yes	640	480	8	15	72	1	87.4	CRT	Yes	640	480	8	25.18	160	44	60.1
Color Passive Dual[3]	Yes	640	480	16	9.57	40	1	58.4	CRT	Yes	640	480	16	25.18	160	44	60.1
Color Passive Dual[2]	Yes	800	600	8	19.4	72	1	73.9	CRT	Yes	640	480	8	25.18	160	44	60.1

▨ Example Frame Rates with Ink Layer Enabled

The FIFO values for these display modes must be set as follows:

1. REG[06Ah] = 3Ch. REG[06Bh] = 3Ch.
2. REG[08Eh] = 0Ch.
3. REG[06Ah] = 3Ch. REG[06Bh] = 3Ch. REG[07Eh] = 0Ch. REG[08Eh] = 0Ch.

18.2.5 Frame Rates for LCD and CRT (800 × 600) with EISD Enabled

Table 18-5 Frame Rates for LCD and CRT (800 × 600) with EISD Enabled

LCD Type	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	max PCLK (MHz)	min HNDP (pixels)	min VNDP (lines)	max Frame Rate (Hz)	CRT/ TV	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	PCLK (MHz)	HNDP (pixels)	VNDP (lines)	Frame Rate (MHz)
Passive Single	No	640	240	8	34	120	1	185.6	CRT	No	800	600	8	40	256	28	60.3
	No	640	480	8	34	120	1	93.0	CRT	No	800	600	8	40	256	28	60.3
Color Passive Dual	No	640	480	8	22.5	88	1	128.2	CRT	No	800	600	8	40	256	28	60.3
TFT[1]	No	800	600	8	34	120	28	58.8	CRT	No	800	600	8	40	256	28	60.3
Color Passive Dual	No	800	600	4	38.6	112	1	140.6	CRT	No	800	600	4	40	256	28	60.3
	No	800	600	8	22.5	88	1	84.2	CRT	No	800	600	8	40	256	28	60.3
Passive Single	Yes	640	240	8	26.2	96	1	147.7	CRT	No	800	600	8	40	256	28	60.3
	Yes	640	480	8	26.2	96	1	74.0	CRT	No	800	600	8	40	256	28	60.3
Mono Passive Dual	Yes	640	480	8	22.5	88	1	128.2	CRT	No	800	600	8	40	256	28	60.3
Color Passive Dual[2]	Yes	640	480	8	19	72	1	110.7	CRT	No	800	600	8	40	256	28	60.3
Passive Single[3]	No	640	240	8	24.4	88	1	139.1	CRT	Yes	800	600	8	40	256	28	60.3
Passive Single	No	640	480	8	24.4	88	1	69.7	CRT	Yes	800	600	8	40	256	28	60.3
Mono Passive Dual[4]	No	640	480	8	20.6	80	1	118.7	CRT	Yes	800	600	8	40	256	28	60.3
Color Passive Dual[5]	No	640	480	8	17.2	64	1	101.4	CRT	Yes	800	600	8	40	256	28	60.3
	No	800	600	8	17.2	64	1	66.1	CRT	Yes	800	600	8	40	256	28	60.3
Passive Single[3]	Yes	640	240	8	19.8	72	1	115.4	CRT	Yes	800	600	8	40	256	28	60.3
Passive Single	Yes	640	480	8	19.8	72	1	57.8	CRT	Yes	800	600	8	40	256	28	60.3
Mono Passive Dual[6]	Yes	640	480	8	17.2	64	1	101.4	CRT	Yes	800	600	8	40	256	28	60.3
Color Passive Dual[6]	Yes	640	480	8	14.7	56	1	87.6	CRT	Yes	800	600	8	40	256	28	60.3

▨ Example Frame Rates with Ink Layer Enabled

The FIFO values for these display modes must be set as follows:

1. REG[04Ah] = 30h. REG[06Ah] = 30h. REG[04Bh] = 3Ch. REG[06Bh] = 3Ch.
2. REG[04Ah] = 1Ah. REG[06Bh] = 25h.
3. REG[06Ah] = 23h. REG[08Eh] = 0Ch.
4. REG[08Eh] = 0Ch.

18.2.6 Frame Rates for LCD and CRT (1024 × 768) with EISD Enabled

Table 18-6 Frame Rates for LCD and CRT (1024 × 768) with EISD Enabled

LCD Type	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	max PCLK (MHz)	min HNDP (pixels)	min VNDP (lines)	max Frame Rate (Hz)	CRT/ TV	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	PCLK (MHz)	HNDP (pixels)	VNDP (lines)	Frame Rate (MHz)
Passive Single[1]	No	320	240	8	31	144	1	277.2	CRT	No	1024	768	8	65	320	41	59.8
Passive Single[2]	No	640	240	8	21.1	80	1	121.6	CRT	No	1024	768	8	65	320	41	59.8
Passive Single	No	640	480	8	21.1	80	1	60.9	CRT	No	1024	768	8	65	320	41	59.8
Color Passive Dual[2]	No	640	480	8	13.8	56	1	82.3	CRT	No	1024	768	8	65	320	41	59.8
Passive Single[3]	Yes	320	240	8	16.2	56	1	178.8	CRT	No	1024	768	8	65	320	41	59.8
	Yes	640	240	8	16.2	56	1	96.6	CRT	No	1024	768	8	65	320	41	59.8
Passive Single	Yes	640	480	8	20	72	1	58.4	CRT	No	1024	768	8	65	320	41	59.8

▨ Example Frame Rates with Ink Layer Enabled

The FIFO values for these display modes must be set as follows:

1. REG[04Ah] = 25h. REG[04Bh] = 3Ch. REG[06Ah] = 30h. REG[06Bh] = 3Ch.
2. REG[04Ah] = 1Ah. REG[06Ah] = 30h. REG[06Bh] = 3Ch.
3. REG[07Eh] = 0Ch.

18.2.7 Frame Rates for LCD and NTSC TV with EISD Enabled

Table 18-7 Frame Rates for LCD and NTSC TV with EISD Enabled

LCD Type	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	max PCLK (MHz)	min HNDP (pixels)	min VNDP (lines)	max Frame Rate (Hz)	CRT/ TV	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	PCLK (MHz)	HNDP (pixels)	VNDP (lines)	Frame Rate (MHz)
Passive Single / TFT	No	320	240	16	10.7	56	1	118.1	NTSC TV	No	640	480	16	14.32	270	22	62.7
	No	640	240	16	10.7	56	1	63.8	NTSC TV	No	640	480	16	14.32	270	22	62.7
	No	640	480	4	40	152	1	105.0	NTSC TV	No	640	480	4	14.32	270	22	62.7
	No	640	480	8	27.6	136	1	73.9	NTSC TV	No	640	480	8	14.32	270	22	62.7
Mono Passive Dual	No	640	480	8	24	128	1	129.6	NTSC TV	No	640	480	8	14.32	270	22	62.7
Color Passive Dual	No	640	480	8	21.1	112	1	116.4	NTSC TV	No	640	480	8	14.32	270	22	62.7
TFT	No	800	600	8	27.6	136	35	46.4	NTSC TV	No	640	480	8	14.32	270	22	62.7
Color Passive Dual	No	800	600	8	21.1	112	1	76.9	NTSC TV	No	640	480	8	14.32	270	22	62.7
Passive Single	Yes	640	480	4	39	144	1	103.4	NTSC TV	No	640	480	4	14.32	270	22	62.7
Mono Passive Dual	Yes	640	480	8	20.5	112	1	113.1	NTSC TV	No	640	480	8	14.32	270	22	62.7
Color Passive Dual	Yes	640	480	4	28.1	112	1	155.0	NTSC TV	No	640	480	4	14.32	270	22	62.7
	Yes	640	480	8	18.2	96	1	102.6	NTSC TV	No	640	480	8	14.32	270	22	62.7
Passive Single	No	640	240	8	20.4	104	1	113.8	NTSC TV	Yes	640	480	8	14.32	270	22	62.7
	No	640	480	4	33.7	128	1	91.2	NTSC TV	Yes	640	480	4	14.32	270	22	62.7
Mono Passive Dual	No	640	480	8	18.4	96	1	103.7	NTSC TV	Yes	640	480	8	14.32	270	22	62.7
Color Passive Dual	No	640	480	4	23.7	96	1	133.6	NTSC TV	Yes	640	480	4	14.32	270	22	62.7
	No	640	480	8	16	88	1	91.2	NTSC TV	Yes	640	480	8	14.32	270	22	62.7
Passive Single	Yes	640	240	4	27.4	104	1	152.8	NTSC TV	Yes	640	480	4	14.32	270	22	62.7
Passive Single[1]	Yes	640	240	8	17.5	88	1	99.7	NTSC TV	Yes	640	480	8	14.32	270	22	62.7
Mono Passive Dual	Yes	640	480	4	23.7	96	1	133.6	NTSC TV	Yes	640	480	4	14.32	270	22	62.7
Mono Passive Dual[1]	Yes	640	480	8	15.6	80	1	89.9	NTSC TV	Yes	640	480	8	14.32	270	22	62.7
Color Passive Dual	Yes	640	480	4	20.4	88	1	116.3	NTSC TV	Yes	640	480	4	14.32	270	22	62.7
	Yes	640	480	8	14.2	80	1	81.8	NTSC TV	Yes	640	480	4	14.32	270	22	62.7

▨ Example Frame Rates with Ink Layer Enabled

The FIFO values for these display modes must be set as follows:

1. REG[07Eh] = 0Ch.

18.2.8 Frame Rates for LCD and PAL TV with EISD Enabled

Table 18-8 Frame Rates for LCD and PAL TV with EISD Enabled

LCD Type	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	max PCLK (MHz)	min HNDP (pixels)	min VNDP (lines)	max Frame Rate (Hz)	CRT/ TV	Ink	Horiz Res (pixels)	Vert Res (lines)	bpp	PCLK (MHz)	HNDP (pixels)	VNDP (lines)	Frame Rate (MHz)
Passive Single	No	320	240	16	7.5	40	1	86.4	PAL TV	No	640	480	16	17.73	495	72	56.6
	No	640	240	16	9	40	1	55.0	PAL TV	No	640	480	16	17.73	495	72	56.6
	No	640	480	4	40	152	1	105.0	PAL TV	No	640	480	4	17.73	495	72	56.6
	No	640	480	8	25.8	128	1	69.8	PAL TV	No	640	480	8	17.73	495	72	56.6
Mono Passive Dual	No	640	480	8	22.2	120	1	121.2	PAL TV	No	640	480	8	17.73	495	72	56.6
Color Passive Dual	No	640	480	8	19	104	1	106.0	PAL TV	No	640	480	8	17.73	495	72	56.6
TFT[1]	No	800	600	8	43	136	35	72.3	PAL TV	No	640	480	8	17.73	495	72	56.6
Color Passive Dual	No	800	600	8	19	104	1	69.8	PAL TV	No	640	480	8	17.73	495	72	56.6
Passive Single	Yes	640	480	4	37.2	144	1	98.6	PAL TV	No	640	480	4	17.73	495	72	56.6
Mono Passive Dual[2]	Yes	640	480	8	19	104	1	106.0	PAL TV	No	640	480	8	17.73	495	72	56.6
Color Passive Dual	Yes	640	480	4	26.2	104	1	146.1	PAL TV	No	640	480	4	17.73	495	72	56.6
	Yes	640	480	8	16.3	88	1	92.9	PAL TV	No	640	480	8	17.73	495	72	56.6
Passive Single	No	640	240	8	18.1	96	1	102.0	PAL TV	Yes	640	480	8	17.73	495	72	56.6
	No	640	480	4	31.7	120	1	86.7	PAL TV	Yes	640	480	4	17.73	495	72	56.6
Mono Passive Dual	No	640	480	4	26	112	1	143.4	PAL TV	Yes	640	480	4	17.73	495	72	56.6
	No	640	480	8	16.1	88	1	91.8	PAL TV	Yes	640	480	8	17.73	495	72	56.6
Color Passive Dual	No	640	480	4	22.7	96	1	128.0	PAL TV	Yes	640	480	4	17.73	495	72	56.6
	No	640	480	8	13.9	72	1	81.0	PAL TV	Yes	640	480	8	17.73	495	72	56.6
Passive Single	Yes	640	240	4	21	96	1	118.4	PAL TV	Yes	640	480	4	17.73	495	72	56.6
Passive Single[2]	Yes	640	240	8	15.4	80	1	88.8	PAL TV	Yes	640	480	8	17.73	495	72	56.6
Mono Passive Dual	Yes	640	480	4	22.7	96	1	128.0	PAL TV	Yes	640	480	4	17.73	495	72	56.6
Mono Passive Dual[2]	Yes	640	480	8	13.9	72	1	81.0	PAL TV	Yes	640	480	8	17.73	495	72	56.6
Color Passive Dual	Yes	640	480	4	19.6	80	1	113.0	PAL TV	Yes	640	480	4	17.73	495	72	56.6
	Yes	640	480	8	12.3	64	1	72.5	PAL TV	Yes	640	480	4	17.73	495	72	56.6

▨ Example Frame Rates with Ink Layer Enabled

The FIFO values for these display modes must be set as follows:

1. REG[04Ah] = 3Ch. REG[04Bh] = 3Ch.
2. REG[07Eh] = 7Ch.

19 Power Save Mode

The S1D13806 includes a software initiated power save mode designed for very low-power applications. In addition, the S1D13806 dynamically disables internal clock networks when not required. Similarly, the LCD and/or CRT/TV pipelines are shut down when not required for the selected display mode.

For power save mode AC Timing, see Section 6.3.2, “Power Save Status” on page 46.

19.1 Overview

Power save mode is initiated by setting REG[1F0h] bit 0 to 1. When power save mode is enabled the following conditions apply.

- LCD display is disabled.
- CRT/TV display is disabled.
- Memory access is not allowed.
- Memory is in self-refresh mode.
- Register access is allowed.

19.2 Power Save Status Bits

LCD Power Save Status bit

The LCD Power Save Status bit (REG[1F1h] bit 0) indicates the state of the LCD panel. When this bit returns a 1, the panel is powered down. When this bit returns a 0, the panel is powered up, or in transition of powering up or down.

The system may disable the LCD pixel clock source when this bit returns a 1. The LCD Power Save Status bit is set to 1 after chip reset.

Memory Controller Power Save Status bit

The Memory Controller Power Save Status bit (REG[1F1h] bit 1) indicates the state of the SDRAM interface. When this bit returns a 1, the SDRAM interface is powered down and the SDRAM is in self-refresh mode. This condition occurs shortly after power save mode is invoked. When this bit returns a 0, the SDRAM interface is active.

The system may disable the memory clock source when this bit returns a 1. The Memory Controller Power Save Status bit is set to 0 after chip reset.

19.3 Power Save Mode Summary

Table 19-1 Power Save Mode Summary

Function	LCD Disabled	CRT/TV Disabled	Power Save Mode Enabled
LCD Display Active?	no	—	No
CRT/TV Display Active?	—	no	No
Register Access Possible?	Yes	Yes	Yes
Memory Access Possible?	Yes	Yes	No
LCD LUT Access Possible?	Yes[1]	—	Yes
CRT/TV LUT Access Possible?	—	Yes[2]	Yes
LCD interface	Forced Low	—	Forced Low
CRT/TV interface	—	Disabled	Disabled
SDRAM interface	Active	Active	Self-Refresh
Host Interface	Active	Active	Active

Note: 1. LCD pixel clock required.

Note: 2. CRT/TV pixel clock required.

20 MECHANICAL DATA

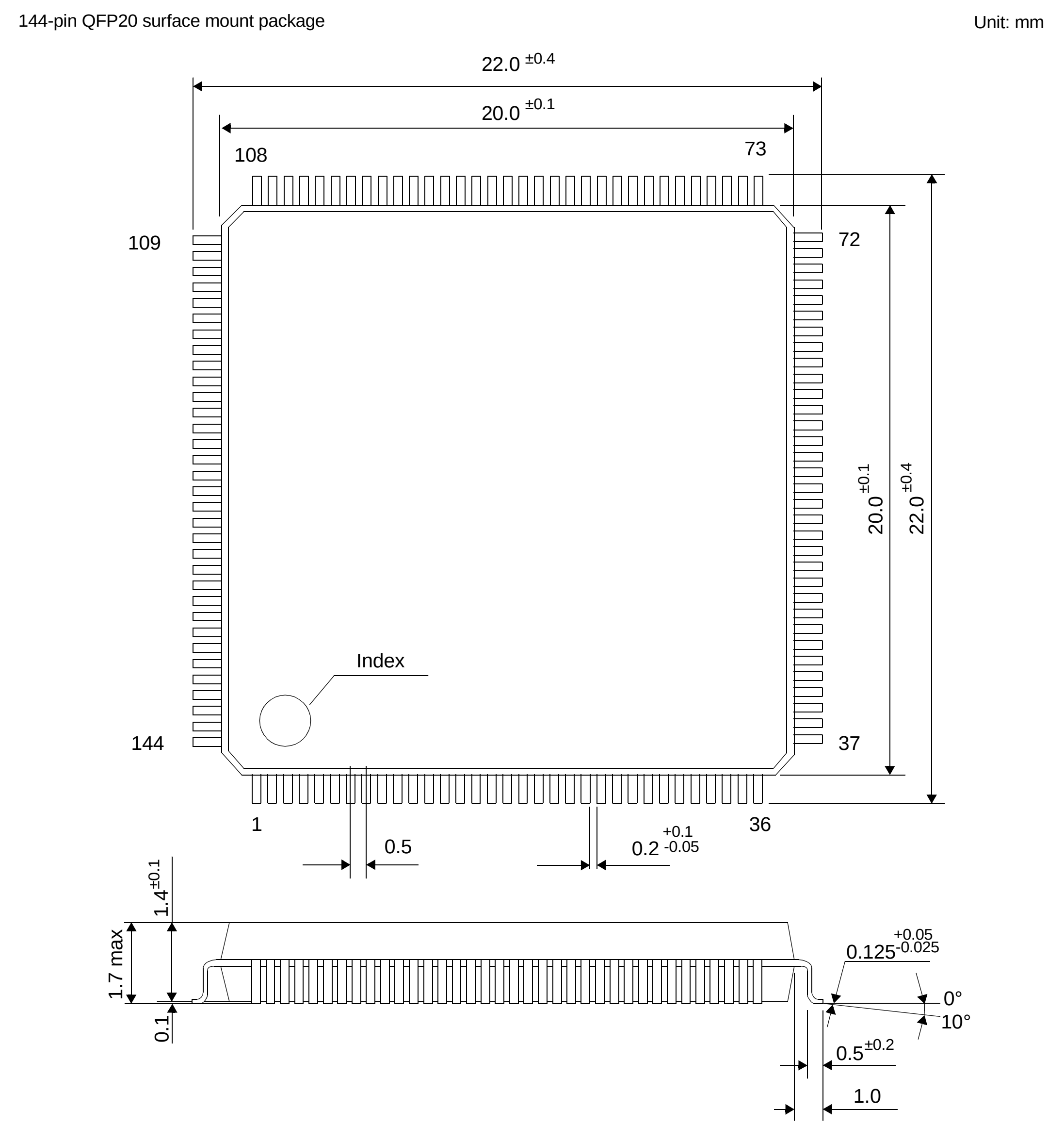


Figure 20-1 Mechanical Drawing 144-pin QFP20

THIS PAGE IS BLANK.

S1D13806 Series

Embedded DRAM Graphics Controller

Programming Notes and Examples

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

This guide provides information on programming the S1D13806 Embedded Memory Display Controller. Included are algorithms which demonstrate how to program the S1D13806. This guide discusses Power-on Initialization, Panning and Scrolling, LUT initialization, LCD Power Sequencing, SwivelView™, etc.

This document will be updated as appropriate. Please check the latest revision of this document and source before beginning any development.

2 INITIALIZATION

This section describes how to initialize the S1D13806.

S1D13806 initialization can be broken into three steps.

- Enable the S1D13806 controller (if necessary identify the specific controller).
- Set all the registers to their initial values**.**
- Program the Look-Up Table (LUT) with color values. This section does not deal with programming the LUT, for details see Section 4, “Look-Up Table (LUT)” .

The simplest way to generate initialization tables for the S1D13806 is to use the utility program 1386CFG.EXE which to generates a header file that can be used by Windows CE. Otherwise modify the init13806.c file directly.

The following table represents the sequence and values written to the S1D13806 registers to control a configuration with these specifications:

- 640 × 480 color format 1 dual passive LCD @ 78Hz.
- 16-bit data interface.
- 8 bit-per-pixel (bpp) color depth - 256 colors.
- 40MHz input clock CLKI.
- CLKI used for BUSCLK (1:1); PCLK (2:1); MCLK (1:1).
- Embedded SDRAM.

Table 2-1 S1D13806 Initialization Sequence

Register	Value	Notes	See Also
[001h]	0000 0000	**Enable the Memory/Register Select Bit.**	
[1FCh]	0000 0000	**Disable the display outputs.**	
[004h] [005h] [008h] [009h]	0000 0000 0000 0000 0000 0000 0000 0000	**Setup GPIO** as inputs; force low if outputs. The OEM may wish GPIO for other purposes which our example does not accommodate for.	
[010h] [014h] [018h] [01Ch]	0000 0000 0001 0000 0000 0010 0000 0010	**Program the Clock Source selects**. In this case we have a single input clock source attached to the CLKI pin. This example uses this as BUSCLK, as MCLK and divide by 2 for PCLK. The CRT clock and MediaPlug clocks are set to CLKI2 reducing power consumption (there is no CLKI2 in this example). If either the CRT or MediaPlug is to be used an input clock must be enabled before accessing the control registers or LUT.	
[01Eh]	0000 0001	**Program CPU Wait States**.	see REG[01Eh] for details
[020h] [021h] [02Ah] [02Bh]	1000 0000 0000 0100 0000 0000 0000 0001	**Program the Frame Buffer Memory Configuration Registers.**	see REG[020h] - REG[02Bh] for details

Table 2-1 S1D13806 Initialization Sequence (Continued)

Register	Value	Notes	See Also
[030h]	0010 0110	**Program the LCD Panel type and Panel Timing Registers.** Panel width = 16-bit; Color Format = don’t care; Color Panel selected; Dual Panel selected; Passive LCD selected. MOD rate = don’t care; Display width = 640 pixels = 4Fh. Horizontal and Vertical Non-display time has been adjusted to provide 78Hz frame rate. TFT FPLINE registers = don’t care for passive panels. Display height = 480 therefore register = 1DFh TFT FPFRAME = don’t care for passive panels.	
[031h]	0000 0000		
[032h]	0100 1111		
[034h]	0001 1111		
[035h]	0000 0000		
[036h]	0000 0000		
[038h]	1101 1111		
[039h]	0000 0001		
[03Ah]	0010 1100		
[03Bh]	0000 0000		
[03Ch]	0000 0000		
[040h]	0000 0003	**Program the Display Output Format and Start Locations for the LCD output. This includes programming the FIFOs.** Select 8 bpp in REG[040h] Ensure that the Dual Panel Buffer is enabled REG [41h] bit 0 = 0 LCD Start Address should typically be from location 0 in the frame buffer. Pixel Pan register is 0 for normal operation. Memory offset register is set to ‘the panel width for normal operation, therefore 640 ÷ 2 for words = 320 words= 140h words Set FIFO values to 0 for “automatic” calculation.	
[041h]	0000 0000		
[042h]	0000 0000		
[043h]	0000 0000		
[044h]	0000 0000		
[046h]	0100 0000		
[047h]	0000 0001		
[048h]	0000 0000		
[04Ah]	0000 0000		
[04Bh]	0000 0000		
[050h]	0000 0000	**Program the CRT/TV Timing control registers.** All values are = don’t care for this example.	
[052h]	0000 0000		
[053h]	0000 0000		
[054h]	0000 0000		
[056h]	0000 0000		
[057h]	0000 0000		
[058h]	0000 0000		
[059h]	0000 0000		
[05Ah]	0000 0000		
[05Bh]	0000 0000		
[060h]	0000 0000	**Program the CRT/TV Display Output Format and Configuration Registers including the FIFOs.** For this example, these values are = don’t care.	
[062h]	0000 0000		
[063h]	0000 0000		
[064h]	0000 0000		
[066h]	0000 0000		
[067h]	0000 0000		
[068h]	0000 0000		
[06Ah]	0000 0000		
[06Bh]	0000 0000		

Table 2-1 S1D13806 Initialization Sequence (Continued)

Register	Value	Notes	See Also
[070h] [071h] [072h] [073h] [074h] [075h] [076h] [077h] [078h] [07Ah] [07Bh] [07Ch] [07Eh]	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	**Program the LCD Ink Layer/Cursor Control, Position, Color and FIFO registers.** For this example, since no Ink Layer or Cursor is used, these registers are = don’t care.	
[080h] [081h] [082h] [083h] [084h] [085h] [086h] [087h] [088h] [08Ah] [08Bh] [08Ch] [08Eh]	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	**Program the CRT/TV Ink Layer/Cursor Control, Position, Color and FIFO registers.** For this example, since no Ink Layer or Cursor is used, these registers are = don’t care.	

Table 2-1 S1D13806 Initialization Sequence (Continued)

Register	Value	Notes	See Also
[100h]	0000 0000	**Program the 2D acceleration (BitBLT) registers to a known state.**	
[101h]	0000 0000		
[102h]	0000 0000		
[103h]	0000 0000		
[104h]	0000 0000		
[105h]	0000 0000		
[106h]	0000 0000		
[108h]	0000 0000		
[109h]	0000 0000		
[10Ah]	0000 0000		
[10Ch]	0000 0000		
[10Dh]	0000 0000		
[110h]	0000 0000		
[111h]	0000 0000		
[112h]	0000 0000		
[113h]	0000 0000		
[114h]	0000 0000		
[115h]	0000 0000		
[116h]	0000 0000		
[118h]	0000 0000		
[119h]	0000 0000		
[1E0]h [1E2h] [1E4h]	0000 0001 0000 0000 0000 0000	**Program the Look-Up Table to a known state.** Selects LUT access to the LCD LUT only. Programming the Look-Up Table is dealt with in a separate section of this document. The init13806.c file shows the example.	see Section Section 4, “Look-Up Table (LUT)” on page 9.
[1F0h]]	0001 0000	**Turn off Power Save Mode.** Sets reserved bit to 1.	
[1F4h]	0000 0000	**Disable Watchdog Timer.**	‘
[1FCh]	0000 0001	**Enable the Display.** For this example, enable the LCD panel only. **Note that the LCD Power Sequencing procedures outlined in** Section 7.1, “Enabling the LCD Panel” **should be used when enabling the LCD panel.**	see REG[1FCh]

3 MEMORY MODELS

The S1D13806 is capable of several color depths. The memory model for each color depth is packed pixel. The S1D13806 supports 4, 8, and 16 bit-per-pixel (bpp) memory models.

3.1 Display Buffer Location

The S1D13806 supports a display buffer of 1.25M byte embedded SDRAM. The display buffer is memory mapped and is accessible directly by software. The memory block location assigned to the S1D13806 display buffer varies with each individual hardware platform.

For further information on the display buffer, see the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx.

3.2 Memory Organization for 4 Bpp (16 Colors/16 Gray Shades)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Pixel 0 Bits 3-0				Pixel 1 Bits 3-0			

Figure 3-1 Pixel Storage for 4 Bpp in One Byte of Display Buffer

In this memory format each byte of display buffer contains two adjacent pixels. Setting or resetting any pixel will require reading the entire byte, masking out the upper or lower nibble (4 bits) and setting the appropriate bits to 1.

Four bit pixels provide 16 gray shades/color possibilities. For monochrome panels the gray shades are generated by indexing into the first 16 elements of the green component of the Look-Up Table (LUT). For color panels the 16 colors are derived by indexing into the first 16 positions of the LUT.

3.3 Memory Organization for 8 Bpp (256 Colors/16 Gray Shades)

<table>
<tr><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td colspan="8">Pixel 0
Bits 7-0</td></tr>
</table>

Figure 3-2 Pixel Storage for 8 Bpp in One Byte of Display Buffer

At a color depth of eight bpp each byte of display buffer represents one pixel on the display. At this color depth the read-modify-write cycles of 4 bpp are eliminated making the update of each pixel faster.

Each byte indexes into one of the 256 positions of the LUT. The S1D13806 LUT supports four bits per primary color. This translates into 4096 possible colors when color mode is selected. Therefore the displayed mode has 256 colors available out of a possible 4096.

When a monochrome panel is selected, the green component of the LUT is used to determine the gray shade intensity. The green indices, with only four bits, can resolve 16 gray shades.

Note: When a monochrome panel (REG[030h] bit 2 = 0) is selected, a four bpp color depth also provides 16 gray shades and uses less display buffer.

3.4 Memory Organization for 16 Bpp (65536 Colors/64 Gray Shades)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Red Component Bits 4-0					Green Component Bits 5-3		
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Green Component Bits 2-0			Blue Component Bits 4-0				

Figure 3-3 Pixel Storage for 16 Bpp in Two Bytes of Display Buffer

At a color depth of 16 bpp the S1D13806 is capable of displaying 65536 colors. The 65536 color pixel is divided into three parts: five bits for red, six bits for green, and five bits for blue. In this mode the LUT is bypassed and output goes directly into the Frame Rate Modulator.

When dithering is enabled (REG[041h) bit 1) the full color range is available on all display types. If dithering is disabled the full color range is only available on TFT/D-TFD or CRT displays. Passive LCD displays are limited to using the four most significant bits from each of the red, green and blue portions of each color resulting in 4096 ($2^4 \times 2^4 \times 2^4$) possible colors.

Should monochrome mode be chosen at this color depth, the output sends the six bits of the green LUT component to the modulator for a total of 64 possible gray shades. If dithering is disabled, the maximum number of gray shades is 16.

4 Look-Up Table (LUT)

This section discusses programming the S1D13806 Look-Up Table (LUT). Included is a summary of the LUT registers, recommendations for color/gray shade LUT values, and additional programming considerations. For a discussion of the LUT architecture, refer to the “*S1D13806 Hardware Functional Specification,*” document number X28B-A-001-xx.

The S1D13806 is designed with a separate LUT for both the LCD and CRT/TV. Each LUT consists of 256 indexed red/green/blue entries. Each LUT entry is four bits wide. The color depth determines how many indices are used to output the image to the display. 4 bpp uses the first 16 indices, 8 bpp uses all 256 indices, and 16 bpp color depths bypass the LUT entirely.

In color modes, the pixel values stored in the display buffer index directly to an RGB value stored in the LUT. In monochrome modes, the pixel value indexes into the green component of the LUT and the amount of green at that index controls the intensity. Monochrome mode look-ups are done based on the Color/Mono Panel Select bit (REG[030h] bit 2). The CRT interface receives the RGB values from the LUT even if simultaneous display is used with a monochrome panel. Therefore, it is important to program the R, G, and B components of the CRT LUT either with a unique set of values, or with R, G, and B values all equivalent.

4.1 Registers

REG[1E0h] Look-Up Table Mode Register							
n/a	n/a	n/a	n/a	n/a	n/a	LUT Mode Bit 1	LUT Mode Bit 0

The S1D13806 is designed with a separate LUT for both the LCD and CRT/TV. The LUT Mode register selects which of the LUTs will be accessed by the CPU when reads/writes are made to REG[1E2h] and REG[1E4h]. LUT mode selection allows the LUTs to be individually written or have identical data written to both LUTs. Individual writes to these registers are useful for Epson Independent Simultaneous Display (EISD) modes where independent images are displayed on the LCD and the CRT/TV. For further information on Epson Independent Simultaneous Display, see the “*S1D13806 Hardware Functional Specification,*” document number X28B-A-001-xx.

For normal operation, this register should be set to 00h which will read the LCD LUT and write both the LCD and CRT/TV LUTs with identical data. For selection of other LUT modes, see REG[1E0h] in the “*S1D13806 Hardware Functional Specification,*” document number X28B-A-001-xx.

REG[1E2h] Look-Up Table Address Register							
LUT Address Bit 7	LUT Address Bit 6	LUT Address Bit 5	LUT Address Bit 4	LUT Address Bit 3	LUT Address Bit 2	LUT Address Bit 1	LUT Address Bit 0

The LUT address register selects which of the 256 LUT entries will be accessed. Writing to this register will select the red bank. After three successive reads or writes to the data register (REG[1E4h]) this register is automatically incremented by one.

REG[1E4h] Look-Up Table Data Register							
LUT Data Bit 3	LUT Data Bit 2	LUT Data Bit 1	LUT Data Bit 0	n/a	n/a	n/a	n/a

This register is where the 4-bit red/green/blue data is written to/read from. With each successive read or write the **internal bank select** is incremented. Three successive reads from this register will result in reading the red, then the green, and finally the blue values associated with the index set in the LUT address register.

After the third read the LUT address register is incremented and the internal bank select points to the red bank again.

4.2 Look-Up Table Organization

- The Look-Up Table treats the value of a pixel as an index into an array of colors or gray shades. For example, a pixel value of zero would point to the first LUT entry, whereas a pixel value of seven would point to the eighth LUT entry.
- The value contained in each LUT entry represents the intensity of the given color or gray shade. This intensity can range in value between 0 and 0Fh.
- The S1D13806 Look-Up Table is linear. This means increasing the LUT entry number results in a lighter color or gray shade. For example, a LUT entry of 0Fh in the red bank results in bright red output while a LUT entry of 05h results in dull red.

Table 4-1 Look-Up Table Configurations

Display Mode	4-Bit Wide Look-Up Table			Effective Gray Shades/ Colors on an Passive Panel With Dithering Disabled	Effective Gray Shades/ Colors on a Passive Panel With Dithering Enabled
	RED	GREEN	BLUE		
4 bpp gray	▨	16	▨	16 gray shades	16 gray shades
8 bpp gray	▨	16	▨	16 gray shades	16 gray shades
16 bpp gray	▨	▨	▨	16 gray shades	64 gray shades
4 bpp color	16	16	16	16 colors	16 colors
8 bpp color	256	256	256	256 colors	256 colors
16 bpp color	▨	▨	▨	4096 colors	65536 colors

▨ = Indicates the Look-Up Table is not used for that display mode

4.2.1 Color Modes

In color display modes, the number of LUT entries used is automatically selected depending on the color depth.

4 bpp color

When the S1D13806 is configured for 4 bpp color mode the first 16 entries in the LUT are used. Each byte in the display buffer contains two adjacent pixels. The upper and lower nibbles of the byte are used as indices into the LUT.

The following table shows LUT values that will simulate those of a VGA operating in 16 color mode.

Table 4-2 Suggested LUT Values to Simulate VGA Default 16 Color Palette

Index	Red	Green	Blue
00	00	00	00
01	00	00	0A
02	00	0A	00
03	00	0A	0A
04	0A	00	00
05	0A	00	0A
06	0A	0A	00
07	0A	0A	0A
08	00	00	00
09	00	00	0F
0A	00	0F	00
0B	00	0F	0F
0C	0F	00	00
0D	0F	00	0F
0E	0F	0F	00
0F	0F	0F	0F
10	00	00	00
...	00	00	00
FF	00	00	00

▨ = Indicates unused entries in the LUT

8 bpp color

When the S1D13806 is configured for 8 bpp color mode all 256 entries in the LUT are used. Each byte in the display buffer corresponds to one pixel and is used as an index value into the LUT.

The S1D13806 LUT has four bits (16 intensities) of intensity control per primary color while a standard VGA RAMDAC has six bits (64 intensities). This four to one difference must be considered when attempting to match colors between a VGA RAMDAC and the S1D13806 LUT. (i.e. VGA levels 0 - 3 map to LUT level 0, VGA levels 4 - 7 map to LUT level 1...). Additionally, the significant bits of the color tables are located at different offsets within their respective bytes. After calculating the equivalent intensity value the result must be shifted into the correct bit positions.

The following table shows LUT values that will approximate the VGA default color palette.

Table 4-3 Suggested LUT Values to Simulate VGA Default 256 Color Palette

Index	R	G	B	Index	R	G	B	Index	R	G	B	Index	R	G	B
00	00	00	00	40	F0	70	70	80	30	30	70	C0	00	40	00
01	00	00	A0	41	F0	90	70	81	40	30	70	C1	00	40	10
02	00	A0	00	42	F0	B0	70	82	50	30	70	C2	00	40	20
03	00	A0	A0	43	F0	D0	70	83	60	30	70	C3	00	40	30
04	A0	00	00	44	F0	F0	70	84	70	30	70	C4	00	40	40
05	A0	00	A0	45	D0	F0	70	85	70	30	60	C5	00	30	40
06	A0	50	00	46	B0	F0	70	86	70	30	50	C6	00	20	40
07	A0	A0	A0	47	90	F0	70	87	70	30	40	C7	00	10	40
08	50	50	50	48	70	F0	70	88	70	30	30	C8	20	20	40
09	50	50	F0	49	70	F0	90	89	70	40	30	C9	20	20	40
0A	50	F0	50	4A	70	F0	B0	8A	70	50	30	CA	30	20	40
0B	50	F0	F0	4B	70	F0	D0	8B	70	60	30	CB	30	20	40
0C	F0	50	50	4C	70	F0	F0	8C	70	70	30	CC	40	20	40
0D	F0	50	F0	4D	70	D0	F0	8D	60	70	30	CD	40	20	30
0E	F0	F0	50	4E	70	B0	F0	8E	50	70	30	CE	40	20	30
0F	F0	F0	F0	4F	70	90	F0	8F	40	70	30	CF	40	20	20
10	00	00	00	50	B0	B0	F0	90	30	70	30	D0	40	20	20
11	10	10	10	51	C0	B0	F0	91	30	70	40	D1	40	20	20
12	20	20	20	52	D0	B0	F0	92	30	70	50	D2	40	30	20
13	20	20	20	53	E0	B0	F0	93	30	70	60	D3	40	30	20
14	30	30	30	54	F0	B0	F0	94	30	70	70	D4	40	40	20
15	40	40	40	55	F0	B0	E0	95	30	60	70	D5	30	40	20
16	50	50	50	56	F0	B0	D0	96	30	50	70	D6	30	40	20
17	60	60	60	57	F0	B0	C0	97	30	40	70	D7	20	40	20
18	70	70	70	58	F0	B0	B0	98	50	50	70	D8	20	40	20
19	80	80	80	59	F0	C0	B0	99	50	50	70	D9	20	40	20
1A	90	90	90	5A	F0	D0	B0	9A	60	50	70	DA	20	40	30
1B	A0	A0	A0	5B	F0	E0	B0	9B	60	50	70	DB	20	40	30
1C	B0	B0	B0	5C	F0	F0	B0	9C	70	50	70	DC	20	40	40
1D	C0	C0	C0	5D	E0	F0	B0	9D	70	50	60	DD	20	30	40
1E	E0	E0	E0	5E	D0	F0	B0	9E	70	50	60	DE	20	30	40
1F	F0	F0	F0	5F	C0	F0	B0	9F	70	50	50	DF	20	20	40
20	00	00	F0	60	B0	F0	B0	A0	70	50	50	E0	20	20	40
21	40	00	F0	61	B0	F0	C0	A1	70	50	50	E1	30	20	40

Table 4-3 Suggested LUT Values to Simulate VGA Default 256 Color Palette (Continued)

Index	R	G	B	Index	R	G	B	Index	R	G	B	Index	R	G	B
22	70	00	F0	62	B0	F0	D0	A2	70	60	50	E2	30	20	40
23	B0	00	F0	63	B0	F0	E0	A3	70	60	50	E3	30	20	40
24	F0	00	F0	64	B0	F0	F0	A4	70	70	50	E4	40	20	40
25	F0	00	B0	65	B0	E0	F0	A5	60	70	50	E5	40	20	30
26	F0	00	70	66	B0	D0	F0	A6	60	70	50	E6	40	20	30
27	F0	00	40	67	B0	C0	F0	A7	50	70	50	E7	40	20	30
28	F0	00	00	68	00	00	70	A8	50	70	50	E8	40	20	20
29	F0	40	00	69	10	00	70	A9	50	70	50	E9	40	30	20
2A	F0	70	00	6A	30	00	70	AA	50	70	60	EA	40	30	20
2B	F0	B0	00	6B	50	00	70	AB	50	70	60	EB	40	30	20
2C	F0	F0	00	6C	70	00	70	AC	50	70	70	EC	40	40	20
2D	B0	F0	00	6D	70	00	50	AD	50	60	70	ED	30	40	20
2E	70	F0	00	6E	70	00	30	AE	50	60	70	EE	30	40	20
2F	40	F0	00	6F	70	00	10	AF	50	50	70	EF	30	40	20
30	00	F0	00	70	70	00	00	B0	00	00	40	F0	20	40	20
31	00	F0	40	71	70	10	00	B1	10	00	40	F1	20	40	30
32	00	F0	70	72	70	30	00	B2	20	00	40	F2	20	40	30
33	00	F0	B0	73	70	50	00	B3	30	00	40	F3	20	40	30
34	00	F0	F0	74	70	70	00	B4	40	00	40	F4	20	40	40
35	00	B0	F0	75	50	70	00	B5	40	00	30	F5	20	30	40
36	00	70	F0	76	30	70	00	B6	40	00	20	F6	20	30	40
37	00	40	F0	77	10	70	00	B7	40	00	10	F7	20	30	40
38	70	70	F0	78	00	70	00	B8	40	00	00	F8	00	00	00
39	90	70	F0	79	00	70	10	B9	40	10	00	F9	00	00	00
3A	B0	70	F0	7A	00	70	30	BA	40	20	00	FA	00	00	00
3B	D0	70	F0	7B	00	70	50	BB	40	30	00	FB	00	00	00
3C	F0	70	F0	7C	00	70	70	BC	40	40	00	FC	00	00	00
3D	F0	70	D0	7D	00	50	70	BD	30	40	00	FD	00	00	00
3E	F0	70	B0	7E	00	30	70	BE	20	40	00	FE	00	00	00
3F	F0	70	90	7F	00	10	70	BF	10	40	00	FF	00	00	00

16 bpp color

The Look-Up Table is bypassed at this color depth, hence programming the LUT is not required.

4.2.2 Gray Shade Modes

This discussion of gray shade (monochrome) modes only applies to the panel interface. Monochrome mode is selected when REG[030h] bit 2 returns a 0. In this mode the value output to the panel is derived solely from the green component of the LUT. The CRT/TV image is formed from all three LUT components (RGB).

Note: In order to match the colors on a CRT/TV with the colors on a monochrome panel when displaying identical images on the panel and CRT/TV, the red and blue components of the LUT must be set to the same intensity as the green component.

4 bpp gray shade

The 4 bpp gray shade mode uses the green component of the first 16 LUT entries. The remaining indices of the LUT are unused.

Table 4-4 Suggested LUT Values for 4 Bpp Gray Shade

Index	Red	Green	Blue
00	00	00	00
01	10	10	10
02	20	20	20
03	30	30	30
04	40	40	40
05	50	50	50
06	60	60	60
07	70	70	70
08	80	80	80
09	90	90	90
0A	A0	A0	A0
0B	B0	B0	B0
0C	C0	C0	C0
0D	D0	D0	D0
0E	E0	E0	E
0F	F0	F0	F0
10	00	00	00
...	00	00	00
FF	00	00	00

░	Required to match CRT to panel
▒	Unused entries

8 bpp gray shade

The 8 bpp gray shade mode uses the green component of the first 16 LUT entries. The green portion of the LUT provides 16 possible intensities. There is no increase in gray shades when selecting 8 bpp mode over 4 bpp mode; however, Swivelview and the BitBLT engine can be used in 8 bpp mode but not in 4 bpp mode.

16 bpp gray shade

The Look-Up Table is bypassed at this color depth, hence programming the LUT is not required.

As with 8 bpp there are limitations to the colors which can be displayed. In this mode the six bits of green are used to set the absolute intensity of the image. This results in 64 gray shades when dithering is enabled and 16 gray shades when dithering is disabled.

5 VIRTUAL DISPLAYS

This section discusses the concept of a virtual display and covers navigation within a virtual display using panning and scrolling.

5.1 Virtual Display

Virtual display is where the image to be viewed is larger than the physical display. This can be in the horizontal, vertical or both dimensions. To view the image, the display is used as a window (or viewport) into the display buffer. At any given time only a portion of the image is visible. Panning and scrolling are used to view the full image. For further information on panning and scrolling, see Section 5.2, “Panning and Scrolling” on page 19.

The Memory Address Offset registers determine the number of horizontal pixels in the virtual image. The offset registers can be set for a maximum of 2^{11} or 2048 words. At a color depth of 4 bpp, 2048 words cover 8,192 pixels. At a color depth of 16 bpp, 2048 words cover 2048 pixels.

The maximum number of lines of the virtual image is the size of the display buffer divided by the number of bytes per horizontal line. The number of bytes per line equals the number of words in the offset register multiplied by two. At the maximum horizontal size, the greatest number of lines that can be displayed using 1.25M bytes of display memory is 320. Reducing the horizontal size makes more display buffer available, thus increasing the available virtual vertical size.

In addition to the calculated limit, the virtual vertical size is limited by the size and location of the Dual Panel Buffer and the Ink Layer/Hardware Cursor (if present).

The maximum horizontal/vertical sizes are seldom used. Figure 5-1 “Viewport Inside a Virtual Display,” shows a more typical use of a virtual display. With a display panel of 320 × 240 pixels, an image of 640 × 480 pixels can be viewed by navigating a 320 × 240 pixel viewport around the image using panning and scrolling.

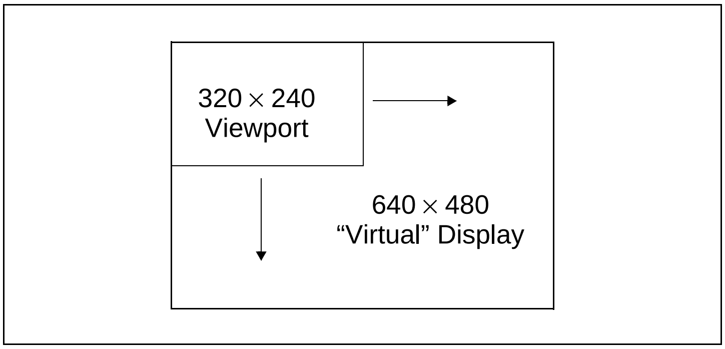


Figure 5-1 Viewport Inside a Virtual Display

5.1.1 Registers

REG[046h] LCD Memory Address Offset Register 0							
LCD Memory Address Offset Bit 7	LCD Memory Address Offset Bit 6	LCD Memory Address Offset Bit 5	LCD Memory Address Offset Bit 4	LCD Memory Address Offset Bit 3	LCD Memory Address Offset Bit 2	LCD Memory Address Offset Bit 1	LCD Memory Address Offset Bit 0

REG[047h] LCD Memory Address Offset Register 1							
n/a	n/a	n/a	n/a	n/a	LCD Memory Address Offset Bit 10	LCD Memory Address Offset Bit 9	LCD Memory Address Offset Bit 8

These registers form the 11-bit memory address offset for the LCD display. This offset equals the number of words from the beginning of one line of the LCD display to the beginning of the next line.

To maintain a constant virtual width as color depth changes, the memory address offset must also change. At a color depth of 4 bpp each word contains 4 pixels, at 16 bpp each word contains one pixel. The formula to determine the value for the memory address registers is:

$$\text{Offset} = \text{PixelsPerVirtualLine} \div \text{PixelsPerWord}$$

This value may not necessarily represent the number of words shown on the LCD display. This is the virtual width of the display image and may be greater than or equal to the physical display width. If PixelsPerVirtualLine equals the physical display width as set in the LCD Horizontal Display Width register (REG[032h]), then the virtual display and physical display are the same size.

REG[066h] CRT/TV Memory Address Offset Register 0							
CRT/TV Memory Address Offset Bit 7	CRT/TV Memory Address Offset Bit 6	CRT/TV Memory Address Offset Bit 5	CRT/TV Memory Address Offset Bit 4	CRT/TV Memory Address Offset Bit 3	CRT/TV Memory Address Offset Bit 2	CRT/TV Memory Address Offset Bit 1	CRT/TV Memory Address Offset Bit 0

REG[067h] CRT/TV Memory Address Offset Register 1							
n/a	n/a	n/a	n/a	n/a	CRT/TV Memory Address Offset Bit 10	CRT/TV Memory Address Offset Bit 9	CRT/TV Memory Address Offset Bit 8

These registers form the 11-bit memory address offset for the CRT/TV display. This offset equals the number of words form the beginning of one line of the CRT/TV display to the beginning of the next line.

To maintain a constant virtual width as color depth changes, the memory address offset must also change. At a color depth of 4 bpp each word contains 4 pixels, at 16 bpp each word contains one pixel. The formula to determine the value for the memory address registers is:

Offset = PixelsPerVirtualLine ÷ PixelsPerWord

This value may not necessarily represent the number of words shown on the CRT/TV display. This is the virtual width of the display image and may be greater than or equal to the physical display width. If PixelsPerVirtualLine equals the physical display width as set in the CRT/TV Horizontal Display Width register (REG[050h]), then the virtual display and physical display are the same size.

5.1.2 Examples

Example 1
Determine the offset value required for a line of 800 pixels at a color depth of 8 bpp.

At a color depth of 8 bpp each byte contains one pixel, therefore each word contains two pixels.

PixelsPerWord = 16 ÷ bpp
= 16 ÷ 8
= 2

To calculate the offset value for this example, the following formula is used.

Offset = PixelsPerVirtualLine ÷ PixelsPerWord
= 800 ÷ 2
= 400
= 190h words

For the LCD, REG[047h] is set to 01h and REG[046h] is set to 90h.
For the CRT/TV, REG[067h] is set to 01h and REG[066h] is set to 90h.

Example 2

Program the Memory Address Offset Registers to support a 16 color (4 bpp) 800 × 600 virtual display on a 640 × 480 LCD panel.

To create a virtual display the offset registers must be programmed to the horizontal size of the larger "virtual" image. After determining the amount of memory used by each line (see example 1), calculate whether there is enough memory to support the desired number of lines.

1. Initialize the S1D13806 registers for a 640 × 480 panel. (See Section 2, "Initialization" on page 2).
2. Calculate the number of pixels per word.

   PixelsPerWord = 16 ÷ bpp
   = 16 ÷ 4
   = 4

3. Determine the offset register value.

   Offset = PixelsPerVirtualLine ÷ PixelsPerWord
   = 800 ÷ 4
   = 200 words
   = 0C8h words

   For the LCD, REG[047h] is set to 00h and REG[046h] is set to C8h.
   For the CRT/TV, REG[067h] is set to 00h and REG[066h] is set to C8h.

4. To confirm whether there is enough memory for the required virtual height, the following formula is used.

   MemoryRequired = WordsPerVirtualLine × 2 × NumberOfLines
   = 200 × 2 × 600
   = 240,000bytes

   The S1D13806 contains 1.25M bytes of embedded SDRAM (or 1,310,720bytes). As long as the calculated value is less than this, it is safe to continue with these values.

5.2 Panning and Scrolling

The terms panning and scrolling refer to the actions used to move a viewport about a virtual display. Although the entire image is stored in the display buffer, only a portion is visible at any given time.

Panning describes the horizontal (side to side) motion of the viewport. When panning to the right the image in the viewport appears to slide to the left. When panning to the left the image to appears to slide to the right. Scrolling describes the vertical (up and down) motion of the viewport. Scrolling down causes the image to appear to slide up and scrolling up causes the image to appear to slide down.

Both panning and scrolling are performed by modifying the start address registers. The start address refers to the word offset in the display buffer where the beginning of the image is displayed from. At color depths other than 16 bpp, another register is required for smooth movement. The pixel pan registers (REG[048h] for LCD, REG[068h] for CRT/TV) allow panning in smaller increments than changing the start address alone.

Internally, the S1D13806 latches different signals at different times. Due to this internal sequence, the start address and pixel pan registers should be accessed in a specific order during panning and scrolling operations, in order to provide the smoothest scrolling. Setting the registers in the wrong sequence, or at the wrong time, results in a “tearing” or jitter effect on the display.

The start address is latched at the beginning of each frame, so the start address can be set within the vertical non-display period (VNDP). The pixel pan register values are latched at the beginning of each display line and must be set during the vertical non-display period. The correct sequence for programing these registers is:

1. Wait for the beginning of the vertical non-display period - For the LCD, REG[03Ah] bit 7 will return a 1 during VNDP; for the CRT/TV, REG[058h] bit 7 will return a 1 during VNDP. Wait for the transition of the appropriate bit to go from 0 to 1. This ensures the register updates are carried out at the beginning of VNDP.

2. Update the start address registers - For the LCD, REG[042h], REG[043h], REG[044h]; for the CRT/TV, REG[062h], REG[063h], REG[064h].

3. Update the pixel panning register - For the LCD, REG[048h] bits 1-0; for the CRT/TV REG[068h] bits 1-0.

5.2.1 Registers

REG[042h] LCD Display Start Address Register 0							
LCD Display Start Address Bit 7	LCD Display Start Address Bit 6	LCD Display Start Address Bit 5	LCD Display Start Address Bit 4	LCD Display Start Address Bit 3	LCD Display Start Address Bit 2	LCD Display Start Address Bit 1	LCD Display Start Address Bit 0

REG[043h] LCD Display Start Address Register 1							
LCD Display Start Address Bit 15	LCD Display Start Address Bit 14	LCD Display Start Address Bit 13	LCD Display Start Address Bit 12	LCD Display Start Address Bit 11	LCD Display Start Address Bit 10	LCD Display Start Address Bit 9	LCD Display Start Address Bit 8

REG[044h] LCD Display Start Address Register 2							
n/a	n/a	n/a	n/a	LCD Display Start Address Bit 19	LCD Display Start Address Bit 18	LCD Display Start Address Bit 17	LCD Display Start Address Bit 16

REG[062h] CRT/TV Display Start Address Register 0							
CRT/TV Display Start Address Bit 7	CRT/TV Display Start Address Bit 6	CRT/TV Display Start Address Bit 5	CRT/TV Display Start Address Bit 4	CRT/TV Display Start Address Bit 3	CRT/TV Display Start Address Bit 2	CRT/TV Display Start Address Bit 1	CRT/TV Display Start Address Bit 0

REG[063h] CRT/TV Display Start Address Register 1							
CRT/TV Display Start Address Bit 15	CRT/TV Display Start Address Bit 14	CRT/TV Display Start Address Bit 13	CRT/TV Display Start Address Bit 12	CRT/TV Display Start Address Bit 11	CRT/TV Display Start Address Bit 10	CRT/TV Display Start Address Bit 9	CRT/TV Display Start Address Bit 8

REG[064h] CRT/TV Display Start Address Register 2							
n/a	n/a	n/a	n/a	CRT/TV Display Start Address Bit 19	CRT/TV Display Start Address Bit 18	CRT/TV Display Start Address Bit 17	CRT/TV Display Start Address Bit 16

The Display Start Address registers form the word address to the display buffer where the LCD or CRT/TV starts displaying from. An address of 0 points to the beginning of the display buffer. Changing the start address registers by one pans from 1 to 4 pixels depending on the current color depth. The following table lists the maximum number of pixels affected by a change of one to these registers.

Table 5-1 Number of Pixels Panned When Start Address Changed By 1

Color Depth (bpp)	Pixels per Word	Number of Pixels Panned
4	4	4
8	2	2
16	1	1

REG[048h] LCD Pixel Panning Register							
n/a	n/a	n/a	n/a	Reserved	Reserved	LCD Pixel Panning Bit 1	LCD Pixel Panning Bit 0

REG[068h] CRT/TV Pixel Panning Register							
n/a	n/a	n/a	n/a	Reserved	Reserved	CRT/TV Pixel Panning Bit 1	CRT/TV Pixel Panning Bit 0

The pixel panning register offers finer control over panning than is available using the start address registers. Using the pixel panning register, it is possible to pan the displayed image one pixel at a time. The number of bits required to pan a single pixel at a time, change with the color depth. The following table shows the bits of the pixel pan register which are used for each color depth.

Table 5-2 Active Pixel Pan Bits

Color Depth (bpp)	Pixel Pan bits used
4	bits [1:0]
8	bit 0
16	none

Note: The pixel panning registers are not required for color depths of 16 bpp.

The pixel panning registers must be updated in conjunction with the start address registers. The pixel panning registers can be thought of as the least significant bit(s) of the start address registers.

When panning to the right on an LCD set for a color depth of 4 bpp, the registers would be updated as follows.

1. Pan right by 1 pixel - increment the pixel panning register by 1: REG[048h] = 01b.
2. Pan right by 1 pixel - increment the pixel panning register by 1: REG[048h] = 10b.
3. Pan right by 1 pixel - increment the pixel panning register by 1: REG[048h] = 11b.
4. Pan right by 1 pixel - reset the pixel panning register to 0: REG[048h] = 00b.
   - increment the start address register by 1: (REG[042h], REG[043h], REG[044h]) + 1.

Note: The above example assumes the pixel panning register is initially set at 0.

When panning to the left on an LCD set for a color depth of 4 bpp, the registers would be updated as follows.

1. Pan left by 1 pixel - decrement the pixel panning register by 1: REG[048h] = 11b.
   - decrement the start address register by 1: (REG[042h], REG[043h], REG[044h]) - 1.
2. Pan left by 1 pixel - decrement the pixel panning register by 1: REG[048h] = 10b.
3. Pan left by 1 pixel - decrement the pixel panning register by 1: REG[048h] = 01b.
4. Pan left by 1 pixel - decrement the pixel panning register by 1: REG[048h] = 00b.

Note: The above example assumes the pixel panning register is initially set at 0.

5.2.2 Examples

The following examples assume the display system has been configured to view a 800 × 600 pixel image in a 640 × 480 viewport. Refer to Section 2, “Initialization” on page 2 and Section 5.1, “Virtual Display” on page 15 for assistance with these settings.

Example 3
Panning - Right and Left

To pan to the right, increment the value in the pixel panning register (REG[048h] for LCD, REG[068h] for CRT/TV). When the pixel pan value reaches the maximum value for the current color depth (i.e. 11b for 4 bpp, 1b for 8 bpp) then set the pixel pan value to zero and increment the start address value. To pan to the left (assuming the pixel panning register is zero), decrement the value in the pixel panning register and decrement the start address register. When the pixel pan value reaches zero then decrement both the pixel panning register and start address register again. If the pixel panning register contains a value other than zero, decrement the value in the pixel panning register only and when the pixel pan value reaches zero, decrement both the pixel panning register and start address register.

Note: Panning operations are easier to follow if a variable (e.g. PanValue) is used to track both the pixel panning and start address registers. The least significant bits of PanValue will represent the pixel panning register value and the more significant bits are the start address register value.

The following example pans to the right by one pixel when the color depth is 4 bpp.

1. Increment PanValue.

   PanValue = PanValue + 1

2. Mask off the values from PanValue for the pixel panning and start address register portions. In this case, 4 bpp, the lower two bits are the pixel panning value and the upper bits are the start address.

   PixelPan = PanValue AND 3

   StartAddress = PanValue SHR 2 (remove PixelPan bits)

3. Write the pixel panning and start address register values using the procedure outlined in Section 5.2.1, “Registers” on page 20.

Example 4
Scrolling - Up and Down

To scroll down, increase the value in the Display Start Address Registers (REG[042h], REG[043h], REG[044h] for LCD, REG[062h], REG[063h], REG[064h] for CRT/TV) by the number of words in one *virtual* scan line. To scroll up, decrease the value in the Display Start Address Registers by the number of words in one *virtual* scan line.
The following example scrolls down one line for a 16 color (4 bpp) 800 × 600 virtual image using a 640 × 480 single panel LCD.

1. Determine the number of words in each line of the virtual image. For a color depth of 4 bpp each byte contains two pixels so each word contains 4 pixels.

   OffsetWords = PixelsPerVirtualLine ÷ PixelsPerWord
   = 800 ÷ 4
   = 200
   = C8h

2. Increment the display start address by the number of words per virtual line.

   StartAddress = StartAddress + OffsetWords
   = StartAddress + C8h

3. Separate the display start address value into three bytes. For the LCD, write the LSB to REG[042h] and the MSB to REG[044h]. For the CRT/TV, write the LSB to REG[062h] and the MSB to REG[064h].

   For the LCD, REG[044h] is set to 00h, REG[043h] is set to 00h, and REG[042h] is set to C8h.

   For the CRT/TV, REG[064h] is set to 00h, REG[063h] is set to 00h, and REG[062h] is set to C8h.

Note: The above example assumes the display start address was initially 0 (the beginning of the display buffer).

6 Power Save Mode

The S1D13806 has been designed for very low-power applications. During normal operation, the internal clocks are dynamically disabled when not required. The S1D13806 design also includes a Power Save Mode to further save power. When Power Save Mode is initiated, LCD power sequencing is required to ensure the LCD bias power supply is disabled properly. For further information on LCD power sequencing, see Section 7, “LCD Power Sequencing” on page 27.

For Power Save Mode AC Timing, see the “*S1D13806 Hardware Functional Specification,*” document number X28B-A-001-xx.

6.1 Overview

The S1D13806 supports a software initiated Power Save Mode. Enabling/disabling Power Save Mode is controlled using the Power Save Mode Enable bit (REG[1F0h] bit 0).

While Power Save Mode is enabled the following conditions apply.

- Display(s) are inactive.
- Registers are accessible.
- Memory is in-accessible.
- LUT is accessible.
- MediaPlug registers are not accessible.

6.2 Registers

6.2.1 Enabling Power Save Mode

REG[1F0h] Power Save Configuration Register							
n/a	n/a	n/a	Reserved	n/a	n/a	n/a	Power Save Mode Enable

The Power Save Mode Enable bit initiates Power Save Mode when set to 1. Setting the bit back to 0 returns the S1D13806 back to normal mode.

Note: Bit 4 is a reserved bit and must be programmed to 1.

Note: Enabling/disabling Power Save Mode requires proper LCD Power Sequencing. See Section 7, “LCD Power Sequencing” on page 27.

6.2.2 Power Save Status Bits

REG[1F1h] Power Save Status Register							
n/a	n/a	n/a	n/a	n/a	n/a	LCD Power Save Status	Memory Controller Power Save Status

The LCD Power Save Status bit is a read-only status bit which indicates the power save state of the LCD panel. When this bit returns a 1, the panel is powered-off. When this bit returns a 0, the LCD panel is powered up or in transition of powering up or down. This bit will return a 1 after a chip reset.

Note: The LCD pixel clock source may be disabled when this bit returns a 1.

REG[1F1h] Power Save Status Register							
n/a	n/a	n/a	n/a	n/a	n/a	LCD Power Save Status	Memory Controller Power Save Status

The Memory Controller Power Save Status bit is a read-only status bit which indicates the power save state of the S1D13806 SDRAM interface. When this bit returns a 1, the SDRAM interface is powered down (the SDRAM is in self-refresh mode). When this bit returns a 0, the SDRAM interface is active. This bit will return a 0 after a chip reset.

Note: The memory clock source may be disabled when this bit returns a 1.

6.3 Enabling Power Save Mode

Power Save Mode must be enabled using the following steps.

1. Disable the LCD power using GPIO11.

Note: The S5U13806B00C uses GPIO11 to control the LCD bias power supplies. Your system design may vary.

2. Wait for the LCD bias power supply to discharge as well as the delay time specified in the LCD panel specification.
3. Enable Power Save Mode - set REG[1F0h] bit 0 to 1.
4. At this time, the LCD pixel clock source may be disabled (Optional). Note the LUT must not be accessed if the pixel clock is not active.

6.4 Disabling Power Save Mode

Power Save Mode must be disabled using the following steps.

1. Disable Power Save Mode - set REG[1F0h] bit 0 to 0.
2. Enable the LCD signals - Set Display Mode Select bit 0 (REG[1FCh] bit 0) to 1.
3. Wait the required delay time as specified in the LCD panel specification.
4. Enable GPIO11 to activate the LCD bias power.

Note: The S5U13806B00C uses GPIO11 to control the LCD bias power supplies. Your system design may vary.

7 LCD POWER SEQUENCING

The S1D13806 requires LCD power sequencing (the process of powering-on and powering-off the LCD panel). LCD power sequencing allows the LCD bias voltage to discharge prior to shutting down the LCD signals, preventing long term damage to the panel and avoiding unsightly “lines” at power-on/power-off.

Proper LCD power sequencing for power-off requires a delay from the time the LCD power is disabled to the time the LCD signals are shut down. Power-on requires the LCD signals to be active prior to applying power to the LCD. This time interval depends on the LCD bias power supply design. For example, the LCD bias power supply on the S5U13806 Evaluation board requires 0.5 seconds to fully discharge. Other power supply designs may vary.

This section assumes the LCD bias power is controlled through GPIO11. The S1D13806 GPIO pins are multi-use pins and may not be available in all system designs. For further information on the availability of GPIO pins, see the *“S1D13806 Hardware Functional Specification,”* document number X28B-A-001-xx.

Note: REG[1F0h] bit 4 must be set to 1 for proper LCD power sequencing.

7.1 Enabling the LCD Panel

The function enables the LCD panel using the following steps.

1. Enable the LCD signals - Set Display Mode Select bit 0 (REG[1FCh] bit 0) to 1.
2. Wait the required delay time as specified in the LCD panel specification (must be set using 1386CFG, document number).
3. Enable GPIO11 to activate the LCD bias power.

7.2 Disabling the LCD Panel

The function disables the LCD panel using the following steps.

1. Disable the LCD power using GPIO11.
2. Wait for the LCD bias power supply to discharge (based on the delay time as specified in the LCD panel specification).
3. Disable the LCD signals - Set Display Mode Select bit 0 (REG[1FCh] bit 0) to 0.
4. At this time, the LCD pixel clock source may be disabled (Optional). Note the LUT must not be accessed if the pixel clock is not active.

8 Hardware Cursor/Ink Layer

8.1 Introduction

The S1D13806 supports either a Hardware Cursor or an Ink Layer for the LCD, and either a Hardware Cursor or an Ink Layer for the CRT/TV. The LCD and CRT/TV are supported independently, so it is possible to select combinations such as a Hardware Cursor on the LCD and an Ink Layer on the CRT/TV.

A Hardware Cursor improves video throughput in graphical operating systems by off-loading much of the work typically assigned to software. For example, consider the actions which must be performed when the user moves the mouse. On a system without hardware support, the operating system must restore the area under the current cursor position, save the area under the new location, and finally draw the cursor shape. Contrast that with the hardware assisted system where the operating system must simply update the cursor X and cursor Y position registers.

An Ink Layer is designed to support stylus or pen input. Without an ink layer, the operating system must save the area of the display buffer (possibly all) where pen input is to occur. After the system recognizes the characters entered, the display would have to be restored and the characters redrawn in a system font. When an Ink Layer is present, the stylus path is drawn in the Ink Layer where it overlays the displayed image. After character recognition finishes the display is updated with the new characters and the ink layer is simply cleared. Saving and restoring the display data is not required providing faster throughput.

The S1D13806 Hardware Cursor/Ink Layer supports a 2 bpp (four color) overlay image. Two of the available colors are transparent and invert. The remaining two colors are user definable.

The Hardware Cursor uses many of the same registers as the Ink Layer. Additionally, the cursor has positional registers for movement. The cursor resolution is 64 × 64 at a color depth of 2 bpp. The Ink Layer resolution is the width of the display by the height of the display at a color depth of 2 bpp. Both the Hardware Cursor and the Ink Layer use the same pixel values to select colors. The Hardware Cursor requires 1024bytes of display buffer and the Ink Layer requires (display width × display height ÷ 4) bytes of display buffer.

8.2 Registers

REG[070h] LCD Ink/Cursor Control Register							
n/a	n/a	n/a	n/a	n/a	n/a	LCD Ink/Cursor Mode Bit 1	LCD Ink/Cursor Mode Bit 0

REG[080h] CRT/TV Ink/Cursor Control Register							
n/a	n/a	n/a	n/a	n/a	n/a	CRT/TV Ink/Cursor Mode Bit 1	CRT/TV Ink/Cursor Mode Bit 0

The Ink/Cursor mode bits determine which of the Hardware Cursor or Ink Layer is active as shown in following table.

Table 8-1 Ink/Cursor Mode

Ink/Cursor Control		Operating Mode
bit 1	bit 0	
0	0	Inactive
0	1	Cursor
1	0	Ink
1	1	Reserved

Note: When cursor mode is selected the cursor image is always 64 × 64 pixels. Selecting an ink layer will result in an area which completely covers the display.

REG[071h] LCD Ink/Cursor Start Address Register							
LCD Ink/Cursor Start Address Bit 7	LCD Ink/Cursor Start Address Bit 6	LCD Ink/Cursor Start Address Bit 5	LCD Ink/Cursor Start Address Bit 4	LCD Ink/Cursor Start Address Bit 3	LCD Ink/Cursor Start Address Bit 2	LCD Ink/Cursor Start Address Bit 1	LCD Ink/Cursor Start Address Bit 0

REG[081h] CRT/TV Ink/Cursor Start Address Register							
CRT/TV Ink/ Cursor Start Address Bit 7	CRT/TV Ink/ Cursor Start Address Bit 6	CRT/TV Ink/ Cursor Start Address Bit 5	CRT/TV Ink/ Cursor Start Address Bit 4	CRT/TV Ink/ Cursor Start Address Bit 3	CRT/TV Ink/ Cursor Start Address Bit 2	CRT/TV Ink/ Cursor Start Address Bit 1	CRT/TV Ink/ Cursor Start Address Bit 0

REG[071h] and REG[081h] determine the display buffer location of the Hardware Cursor/Ink Layer for the LCD and CRT/TV respectively. The Ink/Cursor Start Address register does not contain an actual address, but a value based on the following table.

Table 8-2 Cursor/Ink Start Address Encoding

Ink/Cursor Start Address Bits [7:0]	Start Address (Bytes)
0	1280K - 1024
01h - A0h	1280K - (n × 8192)
A1h - FFh	Invalid

REG[072h] LCD Cursor X Position Register 0							
LCD Cursor X Position Bit 7	LCD Cursor X Position Bit 6	LCD Cursor X Position Bit 5	LCD Cursor X Position Bit 4	LCD Cursor X Position Bit 3	LCD Cursor X Position Bit 2	LCD Cursor X Position Bit 1	LCD Cursor X Position Bit 0

REG[073h] LCD Cursor X Position Register 1							
LCD Cursor X Sign	n/a	n/a	n/a	n/a	n/a	LCD Cursor X Position Bit 9	LCD Cursor X Position Bit 8

REG[082h] CRT/TV Cursor X Position Register 0							
CRT/TV Cursor X Position Bit 7	CRT/TV Cursor X Position Bit 6	CRT/TV Cursor X Position Bit 5	CRT/TV Cursor X Position Bit 4	CRT/TV Cursor X Position Bit 3	CRT/TV Cursor X Position Bit 2	CRT/TV Cursor X Position Bit 1	CRT/TV Cursor X Position Bit 0

REG[083h] CRT/TV Cursor X Position Register 1							
CRT/TV Cursor X Sign	n/a	n/a	n/a	n/a	n/a	CRT/TV Cursor X Position Bit 9	CRT/TV Cursor X Position Bit 8

REG[072h], REG[073h] and REG[082h], REG[083h] control the horizontal position of the Hardware Cursor for the LCD and CRT/TV respectively. The value in these registers specify the location of the left edge of the cursor. When ink mode is selected these registers must be set to zero.

The Cursor X Position supports values of the range -63 to 1023. Negative values allow for the Cursor to be clipped (partially off the screen). The following procedure sets the Cursor X Position.

1. Write the absolute (non-negative) value of the position in bits 9-0.
2. If the position is negative, write a 1 in the Cursor X Sign bit; otherwise write a 0 to the sign bit.

Note: The cursor position is not updated until the Cursor Y Position Register 1 is written (REG[075h] or REG[085h]). When updating the cursor position, always update both the X and Y registers; X first and Y second.

REG[074h] LCD Cursor Y Position Register 0							
LCD Cursor Y Position Bit 7	LCD Cursor Y Position Bit 6	LCD Cursor Y Position Bit 5	LCD Cursor Y Position Bit 4	LCD Cursor Y Position Bit 3	LCD Cursor Y Position Bit 2	LCD Cursor Y Position Bit 1	LCD Cursor Y Position Bit 0

REG[075h] LCD Cursor Y Position Register 1							
LCD Cursor Y Sign	n/a	n/a	n/a	n/a	n/a	LCD Cursor Y Position Bit 9	LCD Cursor Y Position Bit 8

REG[084h] CRT/TV Cursor Y Position Register 0							
CRT/TV Cursor Y Position Bit 7	CRT/TV Cursor Y Position Bit 6	CRT/TV Cursor Y Position Bit 5	CRT/TV Cursor Y Position Bit 4	CRT/TV Cursor Y Position Bit 3	CRT/TV Cursor Y Position Bit 2	CRT/TV Cursor Y Position Bit 1	CRT/TV Cursor Y Position Bit 0

REG[085h] CRT/TV Cursor Y Position Register 1							
CRT/TV Cursor Y Sign	n/a	n/a	n/a	n/a	n/a	CRT/TV Cursor Y Position Bit 9	CRT/TV Cursor Y Position Bit 8

REG[074h], REG[075h] and REG[084h], REG[085h] control the vertical position of the Hardware Cursor for the LCD and CRT/TV respectively. The value in these registers specify the location of the top edge of the cursor. When ink mode is selected these registers must be set to zero.

The Cursor Y Position supports values of the range -63 to 1023. Negative values allow for the Cursor to be clipped (partially off the screen). The following procedure sets the Cursor X Position.

1. Write the absolute (non-negative) value of the position in bits 9-0.
2. If the position is negative, write a 1 in the Cursor Y Sign bit; otherwise write a 0 to the sign bit.

Note: The cursor position is not updated until the Cursor Y Position Register 1 is written (REG[075h] or REG[085h]). When updating the cursor position, always update both the X and Y registers; X first and Y second.

REG[076h] LCD Ink/Cursor Blue Color 0 Register							
n/a	n/a	n/a	LCD Ink/Cursor Blue Color 0 Bit 4	LCD Ink/Cursor Blue Color 0 Bit 3	LCD Ink/Cursor Blue Color 0 Bit 2	LCD Ink/Cursor Blue Color 0 Bit 1	LCD Ink/Cursor Blue Color 0 Bit 0

REG[077h] LCD Ink/Cursor Green Color 0 Register							
n/a	n/a	LCD Ink/Cursor Green Color 0 Bit 5	LCD Ink/Cursor Green Color 0 Bit 4	LCD Ink/Cursor Green Color 0 Bit 3	LCD Ink/Cursor Green Color 0 Bit 2	LCD Ink/Cursor Green Color 0 Bit 1	LCD Ink/Cursor Green Color 0 Bit 0

REG[078h] LCD Ink/Cursor Red Color 0 Register							
n/a	n/a	n/a	LCD Ink/Cursor Red Color 0 Bit 4	LCD Ink/Cursor Red Color 0 Bit 3	LCD Ink/Cursor Red Color 0 Bit 2	LCD Ink/Cursor Red Color 0 Bit 1	LCD Ink/Cursor Red Color 0 Bit 0

These registers form the 16 bpp (5-6-5) RGB values of user-defined color 0 for the LCD Ink Layer/ Hardware Cursor.

REG[07Ah] LCD Ink/Cursor Blue Color 1 Register							
n/a	n/a	n/a	LCD Ink/Cursor Blue Color 1 Bit 4	LCD Ink/Cursor Blue Color 1 Bit 3	LCD Ink/Cursor Blue Color 1 Bit 2	LCD Ink/Cursor Blue Color 1 Bit 1	LCD Ink/Cursor Blue Color 1 Bit 0

REG[07Bh] LCD Ink/Cursor Green Color 1 Register							
n/a	n/a	LCD Ink/Cursor Green Color 1 Bit 5	LCD Ink/Cursor Green Color 1 Bit 4	LCD Ink/Cursor Green Color 1 Bit 3	LCD Ink/Cursor Green Color 1 Bit 2	LCD Ink/Cursor Green Color 1 Bit 1	LCD Ink/Cursor Green Color 1 Bit 0

REG[07Ch] LCD Ink/Cursor Red Color 1 Register							
n/a	n/a	n/a	LCD Ink/Cursor Red Color 1 Bit 4	LCD Ink/Cursor Red Color 1 Bit 3	LCD Ink/Cursor Red Color 1 Bit 2	LCD Ink/Cursor Red Color 1 Bit 1	LCD Ink/Cursor Red Color 1 Bit 0

These registers form the 16 bpp (5-6-5) RGB values of user-defined color 1 for the LCD Ink Layer/ Hardware Cursor.

REG[086h] CRT/TV Ink/Cursor Blue Color 0 Register							
n/a	n/a	n/a	CRT/TV Ink/Cursor Blue Color 0 Bit 4	CRT/TV Ink/Cursor Blue Color 0 Bit 3	CRT/TV Ink/Cursor Blue Color 0 Bit 2	CRT/TV Ink/Cursor Blue Color 0 Bit 1	CRT/TV Ink/Cursor Blue Color 0 Bit 0

REG[087h] CRT/TV Ink/Cursor Green Color 0 Register							
n/a	n/a	CRT/TV Ink/Cursor Green Color 0 Bit 5	CRT/TV Ink/Cursor Green Color 0 Bit 4	CRT/TV Ink/Cursor Green Color 0 Bit 3	CRT/TV Ink/Cursor Green Color 0 Bit 2	CRT/TV Ink/Cursor Green Color 0 Bit 1	CRT/TV Ink/Cursor Green Color 0 Bit 0

REG[088h] CRT/TV Ink/Cursor Red Color 0 Register							
n/a	n/a	n/a	CRT/TV Ink/Cursor Red Color 0 Bit 4	CRT/TV Ink/Cursor Red Color 0 Bit 3	CRT/TV Ink/Cursor Red Color 0 Bit 2	CRT/TV Ink/Cursor Red Color 0 Bit 1	CRT/TV Ink/Cursor Red Color 0 Bit 0

These registers form the 16 bpp (5-6-5) RGB values of user-defined color 0 for the CRT/TV Ink Layer/Hardware Cursor.

REG[08Ah] CRT/TV Ink/Cursor Blue Color 1 Register							
n/a	n/a	n/a	CRT/TV Ink/Cursor Blue Color 1 Bit 4	CRT/TV Ink/Cursor Blue Color 1 Bit 3	CRT/TV Ink/Cursor Blue Color 1 Bit 2	CRT/TV Ink/Cursor Blue Color 1 Bit 1	CRT/TV Ink/Cursor Blue Color 1 Bit 0

REG[08Bh] CRT/TV Ink/Cursor Green Color 1 Register							
n/a	n/a	CRT/TV Ink/Cursor Green Color 1 Bit 5	CRT/TV Ink/Cursor Green Color 1 Bit 4	CRT/TV Ink/Cursor Green Color 1 Bit 3	CRT/TV Ink/Cursor Green Color 1 Bit 2	CRT/TV Ink/Cursor Green Color 1 Bit 1	CRT/TV Ink/Cursor Green Color 1 Bit 0

REG[08Ch] CRT/TV Ink/Cursor Red Color 1 Register							
n/a	n/a	n/a	CRT/TV Ink/Cursor Red Color 1 Bit 4	CRT/TV Ink/Cursor Red Color 1 Bit 3	CRT/TV Ink/Cursor Red Color 1 Bit 2	CRT/TV Ink/Cursor Red Color 1 Bit 1	CRT/TV Ink/Cursor Red Color 1 Bit 0

These registers form the 16 bpp (5-6-5) RGB values of user-defined color 1 for the CRT/TV Ink Layer/Hardware Cursor.

REG[07Eh] LCD Ink/Cursor FIFO High Threshold Register							
n/a	n/a	n/a	n/a	LCD Ink/Cursor FIFO High Threshold Bit 3	LCD Ink/Cursor FIFO High Threshold Bit 2	LCD Ink/Cursor FIFO High Threshold Bit 1	LCD Ink/Cursor FIFO High Threshold Bit 0

REG[08Eh] CRT/TV Ink/Cursor FIFO High Threshold Register							
n/a	n/a	n/a	n/a	CRT/TV Ink/Cursor FIFO High Threshold Bit 3	CRT/TV Ink/Cursor FIFO High Threshold Bit 2	CRT/TV Ink/Cursor FIFO High Threshold Bit 1	CRT/TV Ink/Cursor FIFO High Threshold Bit 0

These registers control the Ink Layer/Hardware Cursor FIFO depth in order to sustain uninterrupted display fetches.

REG[07Eh] determines the FIFO high threshold for the LCD Hardware Cursor/Ink Layer. REG[08Eh] determines the FIFO high threshold for the CRT/TV Hardware Cursor/Ink Layer. When this register is set to 00h, the threshold is automatically set in hardware. For further information, see the “*13806 Hardware Functional Specification,*” document number X28B-A-001-xx.

8.3 Initialization

This section describes the process of initializing the S1D13806 for a Hardware Cursor or Ink Layer.

8.3.1 Memory Considerations

Both the Hardware Cursor and Ink Layer are positioned in the display buffer by the LCD Ink/Cursor Start Address register (REG[071h]) and CRT/TV Ink/Cursor Start Address register (REG[081h]). The Hardware Cursor and Ink Layer should be allocated the highest possible available memory address. If a Dual Panel Buffer is required, or if another Hardware Cursor or Ink Layer is required, additional memory must be allocated and programmed in the appropriate Ink/Cursor Start Address register.

The size of the Dual Panel Buffer is determined by the following.

Dual Panel Buffer Size (in bytes) = (Panel Width × Panel Height) × factor ÷ 16

where:
factor = 4 for color panel
= 1 for monochrome panel

Note: The dual panel buffer always starts at (1280K - Dual Panel Buffer Size).

The size of a hardware cursor is always 1024bytes.
The size of the ink layer in bytes is (display width × display height ÷ 4).

8.3.2 *Examples*

Example 5
Initializing the Hardware Cursor

The following example places an LCD Hardware Cursor at the end of a 1.25M byte display buffer. SwivelView™ modes require software rotation of the Ink Layer. This can only occur when a Dual Panel Buffer is not required. Color 0 is set to black, and color 1 is set to white.

Note: The Hardware Cursor always requires 1024 (400h) bytes.

Table 8-3 LCD Hardware Cursor Initialization Sequence

Register	Value	Notes
[070h]	0000 0001	Enable LCD hardware cursor
[071h]	0000 0000	Set cursor start address to Memory Size - 1024
[072h] [073h]	0000 0000 0000 0000	Set LCD Cursor X Position to 0
[074h] [075h]	0000 0000 0000 0000	Set LCD Cursor Y Position to 0

[076h]
[077h]
[078h]
	0000 0000
	0000 0000
	0000 0000
		Set Color 0 to black

[07Ah]
[07Bh]
[07Ch]
	0001 1111
	0011 1111
	0001 1111
		Set Color 1 to white

[07Eh]	0000 0000	Set FIFO High Threshold to default

Example 6
Initializing the Ink Layer

The following example places an Ink Layer at the end of a 1.25M byte display buffer. SwivelView™ modes require software rotation of the Ink Layer. Color 0 is set to black, and color 1 is set to white.

For a system with a 640 × 480 LCD display, the ink layer size is calculated as follows.

$$\begin{aligned}\text{InkLayerSize} &= (\text{PanelWidth} \times \text{PanelHeight}) \div 4\\ &= (640 \times 480) \div 4\\ &= 76{,}800\text{bytes}\end{aligned}$$

The Ink Layer must be allocated in 8K byte blocks. The value of the LCD Ink/Cursor Start Address register is determined from the following table and calculation.

Table 8-4 Ink Layer Start Address Encoding

Ink/Cursor Start Address Bits [7:0]	Start Address (Bytes)
0	1280K - 1024
01h - A0h	1280K - (n × 8192)
A1h - FFh	Invalid

$$\begin{aligned}n &= \text{InkLayerSize} \div \text{RequiredBlockSize}\\ &= 76{,}800 \div 8192\\ &= 9.375\end{aligned}$$

Fractional values cannot be programmed, therefore round up to an address of 10 (0Ah). This reserves 10 × 8192 = 81,920bytes for the Ink Layer from the end of display buffer.

Note: Always round up the Ink/Cursor Start Address when calculating, otherwise insufficient memory will be allocated for the Ink Layer.

Table 8-5 LCD Ink Layer Initialization Sequence

Register	Value	Notes
[070h]	0000 0010	Enable LCD ink layer
[071h]	0000 1010	Set cursor start address to 0Ah (Memory Size - (8192 × 10))

[076h]
[077h]
[078h]
	0000 0000
	0000 0000
	0000 0000
		Set Color 0 to black

[07Ah]
[07Bh]
[07Ch]
	0001 1111
	0011 1111
	0001 1111
		Set Color 1 to white

[07Eh]	0000 0000	Set FIFO High Threshold to default

8.4 *Writing Cursor/Ink Layer Images*

This section describes how to write images to the Hardware Cursor and Ink Layer. The Hardware Cursor is a 64 × 64 image at a color depth of 2 bpp. The Ink Layer is the same size as the virtual display (width × height) at a color depth of 2 bpp. The Ink Layer may be described as a non-moveable cursor with the same resolution as the display device.

8.4.1 *Hardware Cursor/Ink Layer Data Format*

The Hardware Cursor/Ink Layer image is fixed at a color depth of 2 bpp. The following diagram shows the Hardware Cursor/Ink Layer data format for a little endian system.

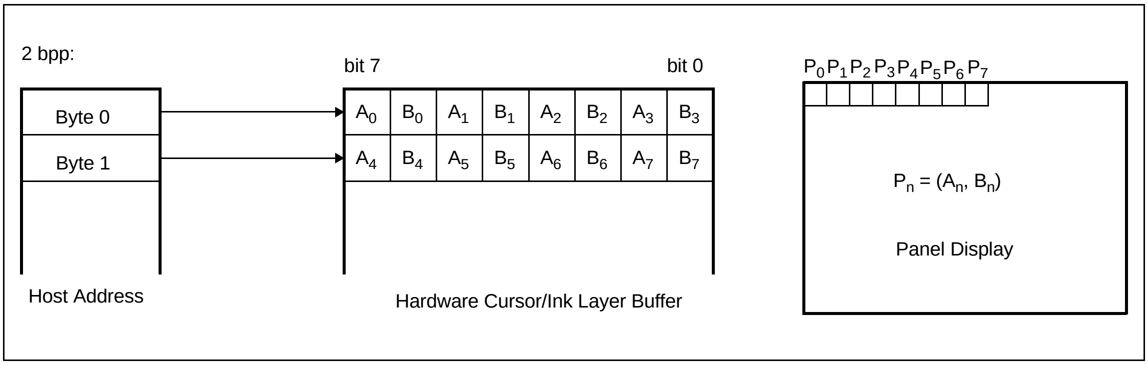

2 bpp:
bit 7
bit 0
Byte 0
Byte 1
A_0 B_0 A_1 B_1 A_2 B_2 A_3 B_3
A_4 B_4 A_5 B_5 A_6 B_6 A_7 B_7
P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7
$P_n = (A_n, B_n)$
Panel Display
Host Address
Hardware Cursor/Ink Layer Buffer

Figure 8-1 Hardware Cursor/Ink Layer Data Format

The image data for pixel n, (A_n,B_n), selects the color for pixel n as follows:

Table 8-6 Ink/Cursor Color Select

(A_n,B_n)	Color	Comments
00	Color 0	Ink/Cursor Color 0 Register: For LCD, REG[076h], REG[077h], REG[078h]. For CRT/TV, REG[086h], REG[087h], REG[088h].
01	Color 1	Ink/Cursor Color 1 Register: For LCD, REG[07Ah], REG[07Bh],REG[07Ch]. For CRT/TV, REG[08Ah], REG[08Bh], REG[08Ch].
10	Background	Ink/Cursor is transparent – show background
11	Inverted Background	Ink/Cursor is transparent – show inverted background

8.4.2 Cursor Image

The following procedures demonstrate how to write an image to the Hardware Cursor buffer.

Landscape Mode (no rotation)

1. For the LCD cursor, calculate the start address based on the value in REG[071h].
   For the CRT/TV cursor, calculate the start address based on the value in REG[081h]. Refer to the REG[071h] and REG[081h] register descriptions for more information.
2. Write the cursor image to the display buffer. The image must be exactly 1024bytes.

SwivelView Modes

1. Save the current state of REG[1FCh] bit 6.
2. Set REG[1FCh] bit 6 to 0.
3. For the LCD cursor, calculate the start address based on the value in REG[071h].
   For the CRT/TV cursor, calculate the start address based on the value in REG[081h]. Refer to the REG[071h] and REG[081h] register descriptions for more information.
4. Perform a software rotate of the cursor image.
5. Write the rotated cursor image to the display buffer. The image must be exactly 1024bytes.
6. Restore the original state of REG[1FCh] bit 6.

Note: It is possible to use the same cursor image for both LCD and CRT/TV displays. Program the LCD and CRT/TV Ink/Cursor Start Address registers (REG[071h] and REG[081h]) to the same location. This saves some display buffer which would otherwise be used by a second cursor image. Note this saves 8192bytes of display buffer, not 1024bytes,
because the start address moves in steps of 8192bytes.

8.4.3 *Ink Layer Image*

The following procedures demonstrate how to write an image to the Ink Layer buffer.

Landscape Mode (no rotation)

1. For the LCD, calculate the start address based on the value in REG[071h].
   For the CRT/TV, calculate the start address based on the value in REG[081h].
   Refer to the REG[071h] and REG[081h] register descriptions for more information.
2. Write the Ink Layer image to the display buffer. The image must be exactly (display width × display height ÷ 4) bytes.

SwivelView Modes

1. Save the current state of REG[1FCh] bit 6.
2. Set REG[1FCh] bit 6 to 0.
3. For the LCD, calculate the start address based on the value in REG[071h].
   For the CRT/TV, calculate the start address based on the value in REG[081h].
   Refer to the REG[071h] and REG[081h] register descriptions for more information.
4. Perform a software rotate of the Ink Layer image.
5. Write the rotated Ink Layer image to the display buffer. The image must be exactly (display width × display height ÷ 4) bytes.
6. Restore the original state of REG[1FCh] bit 6.

Note: It is possible to use the same Ink Layer image for both LCD and CRT/TV displays. Program the LCD and CRT/TV Ink/Cursor Start Address registers (REG[071h] and REG[081h]) to the same location. This save some display buffer which would otherwise be used by a second Ink Layer.

8.5 Cursor Movement

The following section discusses cursor movement in landscape, SwivelView 90°, SwivelView 180°, and SwivelView 270° modes.

It is possible to move the top left corner of the cursor to a negative position (-63, -63). This allows the cursor to be clipped (only a portion is visible on-screen).

Cursor positions don’t take effect until the most significant byte of the Y position register is written. Therefore, the following register write order is recommended.

1. Set X Position Register 0
2. Set X Position Register 1
3. Set Y Position Register 0
4. Set Y Position Register 1.

8.5.1 Move Cursor in Landscape Mode (no rotation)

In the following example, (x, y) represents the desired cursor position.

1. Calculate abs(x), the absolute (non-negative) value of x.
2. Write the least significant byte of abs(x) to X Position Register 0.
3. **If x is negative**, take the value of the most significant byte of abs(x) and logically OR with 80h. Write the result to X Position Register 1.
   If x is positive, write the most significant byte of abs(x) to X Position Register 1.
4. Calculate abs(y), the absolute (non-negative) value of y.
5. Write the least significant byte of abs(y) to Y Position Register 0.
6. **If y is negative**, take the value of the most significant byte of abs(y) and logically OR with 80h. Write the result to Y Position Register 1.
   If y is positive, take the value of the most significant byte of abs(y) and write to Y Position Register 1.

8.5.2 Move Cursor in SwivelView 90° Rotation

In the following example, (x, y) represent the desired cursor position.

1. Calculate abs(x), the absolute (non-negative) value of x.
2. Write the least significant byte of abs(x) to Y Position Register 0.
3. **If x is negative**, take the value of the most significant byte of abs(x) and logically OR with 80h. Write the result to Y Position Register 1.
   If x is positive, write the most significant byte of abs(x) to Y Position Register 1.
4. Calculate a value for y2,
   where y2 = display width - y - 64.
5. Calculate abs(y2), the absolute (non-negative) value of y2.
6. Write the least significant byte of abs(y2) to X Position Register 0.
7. **If y2 is negative**, take the value of the most significant byte of abs(y2) and logically OR with 80h. Write the result to X Position Register 1.
   If y2 is positive, write the most significant byte of abs(y2) to X Position Register 1.

8.5.3 Move Cursor in SwivelView 180° Rotation

In the following example, (x, y) represent the desired cursor position.

1. Calculate the value of x2,
   where x2 = display width - x - 64
2. Calculate abs(x2), the absolute (non-negative) value of x2.
3. Write the least significant byte of abs(x2) to X Position Register0.
4. **If x2 is negative**, take the value of the most significant byte of abs(x2) and logically OR with 80h. Write the result to X Position Register 1.
   If x2 is positive, write the most significant byte of abs(x2) to X Position Register 1.
5. Calculate the value of y2,
   where y2 = display height - y - 64
6. Calculate abs(y2), the absolute (non-negative) value of y2.
7. Write the least significant byte of abs(y2) to Y Position Register 0.
8. **If y2 is negative**, take the value of the most significant byte of abs(y2) and logically OR with 80h. Write the result to Y Position Register 1.
   If y2 is positive, write the most significant byte of abs(y2) to Y Position Register 1.

8.5.4 Move Cursor in SwivelView 270° Rotation

In the following example, (x, y) represent the desired cursor position.

1. Calculate the value of x2,
   where x2 = display width - x - 64
2. Calculate abs(x2), the absolute (non-negative) value of x2.
3. Write the least significant byte of abs(x2) to Y Position Register 0.
4. **If x2 is negative**, take the value of the most significant byte of abs(x2) and logically OR with 80h. Write the result to Y Position Register 1.
   If x2 is positive, write the most significant byte of abs(x2) to Y Position Register 1.
5. Calculate abs(y), the absolute (non-negative) value of y.
6. Write the least significant byte of abs(y) to X Position Register 0.
7. **If y is negative**, take the value of the most significant byte of abs(y) and logically OR with 80h. Write the result to X Position Register 1.
   If y is positive, write the most significant byte of abs(y) to X Position Register 1.

9 SwivelView™

Most computer displays operate in landscape mode. In landscape mode the display is wider than it is high. For example, a standard display size of 640 × 480 is 640 pixels wide and 480 pixels wide.

SwivelView rotates the display image clockwise in ninety degree increments, possibly resulting in a display that is higher than it is wide. Rotating the image on a 640 × 480 display by 90 or 270 degrees yields a display that is now 480 pixels wide and 640 pixels high.

SwivelView also works with panels that are designed with a “portrait” orientation. In this case, when SwivelView 0° is selected, the panel will be in a “portrait” orientation. A selection of SwivelView 90° or SwivelView 270° rotates to a landscape orientation.

9.1 S1D13806 SwivelView

The S1D13806 provides hardware support for SwivelView in 8 and 16 bpp color depths on LCD panels. SwivelView is not supported on CRT or TV displays.

Certain conditions must be considered when SwivelView is enabled.

- The virtual display offset (scan line) must be set to 1024 pixels (1024bytes in 8 bpp, 2048bytes in 16 bpp) when SwivelView Enable Bit 0 is set to 1.
- The display start address is calculated differently when SwivelView is enabled.
- Calculations that would result in panning in landscape mode, may result in scrolling when SwivelView is enabled and vice-versa.

9.2 Registers

REG[01FCh] Display Mode Register							
n/a	SwivelView Enable Bit 0	n/a	n/a	n/a	Display Mode Select Bit 2	Display Mode Select Bit 1	Display Mode Select Bit 0

REG[040h] LCD Display Mode Register							
LCD Display Blank	n/a	n/a	SwivelView Enable Bit 1	n/a	Bit-Per-Pixel Select Bit 2	Bit-Per-Pixel Select Bit 1	Bit-Per-Pixel Select Bit 0

The SwivelView modes are enabled using a combination of 2 enable bits - SwivelView Enable Bit 0 (REG[1FCh]) and SwivelView Enable Bit 1 (REG[040h]). The combinations of these bits provide the following rotations.

Table 9-1 SwivelView Enable Bits

SwivelView Enable Bit 1	SwivelView Enable Bit 0	Display Rotated (degrees)
0	0	0
0	1	90
1	0	180
1	1	270

REG[046h] LCD Memory Address Offset Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[047h] LCD Memory Address Offset Register 1							
n/a	n/a	n/a	n/a	n/a	Bit 10	Bit 9	Bit 8

The LCD Memory Address Offset Registers must be adjusted according to the desired SwivelView rotation and color depth. Set the LCD Memory Address Offset Registers based on the values provided for each color depth in the following table. Panel Width (PW) is the horizontal panel size in pixels (i.e. for a 640 × 480 panel, PW is 640 regardless of the display rotation).

Table 9-2 LCD Memory Address Offset Values

SwivelView Enable		Display Rotated	Memory Address Offset Value	
Bit 1	Bit 0		16 bpp	8 bpp
0	0	0 degrees	PW	PW ÷ 2
0	1	90 degrees	1024	512
1	0	180 degrees	PW	PW ÷ 2
1	1	270 degrees	1024	512

REG[042h] LCD Display Start Address Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[043h] LCD Display Start Address Register 1							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

REG[044h] LCD Display Start Address Register 2							
n/a	n/a	n/a	n/a	Bit 19	Bit 18	Bit 17	Bit 16

The LCD Display Start Address Registers must be adjusted according to the desired SwivelView rotation and color depth. Set the LCD Display Start Address Registers based on the values provided for each color depth in the following table. Panel Width (PW) is the horizontal panel size in pixels. Panel Height (PH) is the vertical panel size in lines (i.e. for a 640 × 480 panel, PW is 640 and PH is 480 regardless of display rotation). Stride is the based on the previously calculated memory offset in bytes (Stride = MemoryOffset × 2).

Table 9-3 LCD Display Start Address Values

SwivelView Enable		Display Rotated	LCD Display Start Address Value	
Bit 1	Bit 0		16 Bpp mode	8 Bpp mode
0	0	0 degrees	0	0
0	1	90 degrees	(1024 - PW)	(1024 - PW) ÷ 2
1	0	180 degrees	(Stride × PH - (Stride - 2 × PW)) ÷ 2 - 1	(Stride × PH - (Stride - PW)) ÷ 2 - 1
1	1	270 degrees	(Stride × PH) ÷ 2 - 1	(Stride × PH) ÷ 2 - 1

9.3 Limitations

The following limitations apply when SwivelView bit 0 is set to 1 (rotation by 90° or 270°):

- Only 8/16 bpp modes can be rotated 90 or 270 degrees.
- Hardware Cursor and Ink Layer images are not rotated - software rotation must be used. SwivelView must be turned off when the programmer is accessing the Hardware Cursor or the Ink Layer. The BitBLT engine ignores the SwivelView bits also.
- Pixel panning works vertically.
- It is not possible to rotate an already displayed image. The image must be redrawn.

Note: Drawing into the Hardware Cursor/Ink Layer with SwivelView enabled requires disabling SwivelView, drawing in the Hardware Cursor/Ink Layer buffer, then re-enabling SwivelView.

9.4 Examples

Source code demonstrating various SwivelView rotations is provided in the file 13806swivel.c available on the internet at www.eea.epson.com.

Example 7
Rotate Image 90° for a 640 × 480 display at a color depth of 8 bpp.

Before enabling SwivelView, the display buffer should be cleared. This makes the transition smoother as existing display images cannot be rotated by hardware - a repaint is necessary.

1. Set the line offset to 1024 pixels. The Memory Offset register is the offset in words.

   Write 02h to REG[047h] and write 00h to REG[046h].

2. Set the LCD Display Start Address. The Display Start Address registers form a pointer to a word, therefore the value to set the start.

   Write C0h (192 or (1024 - 480) ÷ 2) to REG[042h], REG[043h] and REG[044h]. That is write C0h to REG[042h], 00h to REG[043h] and 00h to REG[044h].

3. Enable SwivelView Bit 0 and clear SwivelView Bit 1. Set REG[1FCh] to 1 and REG[040h] to 0.
4. The display is now configured for SwivelView 90° mode. Offset zero into the display buffer corresponds to the upper left corner of the display. The only difference seen by the programmer is the display offset is now 1024 pixels regardless of the physical dimensions of the display.
5. Draw the desired image.

Example 8
Rotate Image 180 degrees for a 640 × 480 display at a color depth of 16 bpp.

Assuming the existing image is unrotated, the display buffer does not have to be cleared. Existing display images are simply be rotated by hardware. In this case a repaint is not necessary.

1. The Memory Offset register **does not** need to be modified.
2. Set the LCD Display Start Address. The Display Start Address registers form a pointer to a word, therefore the value to set the start. Calculate the value based on the following formula.

$$\begin{aligned}\text{StartAddress} &= (\text{ScanBytes} \times \text{PanelHeight} - (\text{ScanBytes} - 2 \times \text{PanelWidth})) \div 2 - 1\\ &= (1280 \times 480 - (1280 - 2 \times 640)) \div 2 - 1\\ &= (1280 \times 480) \div 2 - 1\\ &= 307199\\ &= \text{4AFFFh}\end{aligned}$$

   Program the LCD Display Start Address Registers. REG[044h] is set to 04h, REG[043h] is set to AFh, and REG[042h] is set to FFh.

3. Set SwivelView Bit 1 by setting bit 4 of REG[040h]
4. The display will now show the previous image rotated by 180 degrees. Offset zero into display memory will correspond to the lower right corner of the display.
5. Draw any new desired image. The drawing software can be completely unaware of the display being rotated.

9.5 Simultaneous Display Considerations

Although only the LCD panel image can be rotated, it is possible to simultaneously display **an independent** image on the CRT or TV display. In this case, the programmer should be aware of the following:

- As the LCD display buffer must start at offset 0 when a rotated display is required, the CRT display buffer must be located after the LCD display buffer.
- When modifying the CRT display buffer, SwivelView Enable Bit 0 must be cleared and then restored when finished. The following demonstrates this principle.

1. Save SwivelView Bit 0
2. Clear SwivelView Bit 0
3. Draw the CRT/TV image
4. Restore the saved SwivelView Bit 0.

10 2D BitBLT Engine

The term BitBLT is an acronym for Bit Block Transfer. During a BitBLT operation data is transferred from one memory location (source) to another memory location (destination). With current graphical user interfaces (GUIs) this term generally refers to the transfer of bitmap images to or from video memory (display buffer).

The resulting bitmap image may be derived from up to three items or operands:

- the source data.
- an optional pattern.
- the current destination data.

The operands are combined using logical AND, OR, XOR and NOT operations. The combining process is called a Raster Operation (ROP). The S1D13806 2D Accelerator supports all possible 16 ROPs between source data and destination data. The destination is always the display buffer and the source is either data in the display buffer, a pattern in the display buffer, or data provided by the host CPU.

The 2D BitBLT Engine in the S1D13806 is designed to increase the speed of the most common GUI operations by off-loading work from the CPU, thus reducing traffic on the system bus and improving the efficiency of the display buffer interface. The 2D BitBLT Engine is designed to work at color depths of 8 bpp and 16 bpp.

10.1 Registers

The BitBLT control registers on the S1D13806 are located at registers 100h through 119h. The following is a description of all BitBLT registers.

REG[100h] BitBLT Control Register 0							
BitBLT Active Status	BitBLT FIFO Not Empty Status (RO)	BitBLT FIFO Half Full Status (RO)	BitBLT FIFO Full Status (RO)	n/a	n/a	BitBLT Destination Linear Select	BitBLT Source Linear Select

The BitBLT Active Status bit has two data paths, one for write and one for read.

Write Data Path
When this bit is set to 1, the BitBLT as selected in the BitBLT Operation Register (REG[103h]) is started.

Read Data Path
When this bit is read, it returns the status of the BitBLT engine. When a read from this bit returns 0, the BitBLT engine is idle and is ready for the next operation. When a read from this bit returns a 1, the BitBLT engine is busy.

REG[100h] BitBLT Control Register 0							
BitBLT Active Status	BitBLT FIFO Not Empty Status (RO)	BitBLT FIFO Half Full Status (RO)	BitBLT FIFO Full Status (RO)	n/a	n/a	BitBLT Destination Linear Select	BitBLT Source Linear Select

The BitBLT FIFO Not Empty Status bit is a read-only status bit. When this bit returns a 0, the BitBLT FIFO is empty. When this bit returns a 1, the BitBLT FIFO contains at least one data (or one word).

REG[100h] BitBLT Control Register 0							
BitBLT Active Status	BitBLT FIFO Not Empty Status (RO)	BitBLT FIFO Half Full Status (RO)	BitBLT FIFO Full Status (RO)	n/a	n/a	BitBLT Destination Linear Select	BitBLT Source Linear Select

The BitBLT FIFO Half Full Status bit is a read-only status bit. When this bit returns a 0, the BitBLT FIFO is less than half full (contains 7 or less data). When this bit returns a 1, the BitBLT FIFO is half full or greater than half full (contains 8 or more data).

REG[100h] BitBLT Control Register 0							
BitBLT Active Status	BitBLT FIFO Not Empty Status (RO)	BitBLT FIFO Half Full Status (RO)	BitBLT FIFO Full Status (RO)	n/a	n/a	BitBLT Destination Linear Select	BitBLT Source Linear Select

The BitBLT FIFO Full Status bit is a read-only status bit. When this bit returns a 0, the BitBLT FIFO is not full (contains less than 16 data). When this bit returns a 1, the BitBLT FIFO is full (contains 16 data).

REG[100h] BitBLT Control Register 0							
BitBLT Active Status	BitBLT FIFO Not Empty Status (RO)	BitBLT FIFO Half Full Status (RO)	BitBLT FIFO Full Status (RO)	n/a	n/a	BitBLT Destination Linear Select	BitBLT Source Linear Select

The BitBLT Destination Linear Select bit specifies the storage method of the destination BitBLT. If this bit = 0, the destination BitBLT is stored as a rectangular region of memory. If this bit = 1, the destination BitBLT is stored as a contiguous linear block of memory.

REG[100h] BitBLT Control Register 0							
BitBLT Active Status	BitBLT FIFO Not Empty Status (RO)	BitBLT FIFO Half Full Status (RO)	BitBLT FIFO Full Status (RO)	n/a	n/a	BitBLT Destination Linear Select	BitBLT Source Linear Select

The BitBLT Source Linear Select bit specifies the storage method of the source BitBLT. If this bit = 0, the source BitBLT is stored as a rectangular region of memory. If this bit = 1, the source BitBLT is stored as a contiguous linear block of memory.

REG[101h] BitBLT Control Register 1							
n/a	n/a	n/a	Reserved	n/a	n/a	n/a	BitBLT Color Format Select

This bit is reserved and must be set to 0.

REG[101h] BitBLT Control Register 1							
n/a	n/a	n/a	Reserved	n/a	n/a	n/a	BitBLT Color Format Select

The BitBLT Color Format Select bit selects the color format that the BitBLT operation is applied to. When this bit = 0, 8 bpp (256 color) format is selected. When this bit = 1, 16 bpp (64K color) format is selected.

REG[102h] BitBLT ROP Code/Color Expansion Register							
n/a	n/a	n/a	n/a	BitBLT ROP Code Bit 3	BitBLT ROP Code Bit 2	BitBLT ROP Code Bit 1	BitBLT ROP Code Bit 0

The BitBLT ROP Code/Color Expansion Register selects the Raster Operation (ROP) used for the Write BitBLT, Move BitBLT, and Pattern fill. It is also used to specify the start bit position for BitBLTs with color expansion. The following table summarizes the functionality of this register.

Table 10-1 BitBLT ROP Code/Color Expansion Function Selection

BitBLT ROP Code Bits [3:0]	Boolean Function for Write BitBLT and Move BitBLT	Boolean Function for Pattern Fill	Start Bit Position for Color Expansion
0000	0 (Blackness)	0 (Blackness)	bit 0
0001	~S . ~D or ~(S + D)	~P . ~D or ~(P + D)	bit 1
0010	~S . D	~P . D	bit 2
0011	~S	~P	bit 3
0100	S . ~D	P . ~D	bit 4
0101	~D	~D	bit 5
0110	S ^ D	P ^ D	bit 6
0111	~S + ~D or ~(S . D)	~P + ~D or ~(P . D)	bit 7
1000	S . D	P . D	bit 0
1001	~(S ^ D)	~(P ^ D)	bit 1
1010	D	D	bit 2
1011	~S + D	~P + D	bit 3
1100	S	P	bit 4
1101	S + ~D	P + ~D	bit 5
1110	S + D	P + D	bit 6
1111	1 (Whiteness)	1 (Whiteness)	bit 7

S = Source, D = Destination, P = Pattern

Operators: ~ = NOT, . = Logical AND, + = Logical OR, ^ = Logical XOR

REG[103h] BitBLT Operation Register							
n/a	n/a	n/a	n/a	BitBLT Operation Bit 3	BitBLT Operation Bit 2	BitBLT Operation Bit 1	BitBLT Operation Bit 0

The BitBLT Operation Register selects the BitBLT operation to be carried out based on the following table:

Table 10-2 BitBLT Operation Selection

BitBLT Operation Bits [3:0]	BitBLT Operation
0000	Write BitBLT with ROP
0001	Read BitBLT
0010	Move BitBLT in positive direction with ROP
0011	Move BitBLT in negative direction with ROP
0100	Transparent Write BitBLT
0101	Transparent Move BitBLT in positive direction
0110	Pattern Fill with ROP
0111	Pattern Fill with transparency
1000	Color Expansion
1001	Color Expansion with transparency
1010	Move BitBLT with Color Expansion
1011	Move BitBLT with Color Expansion and transparency
1100	Solid Fill
Other combinations	Reserved

REG[104h] BitBLT Source Start Address Register 0							
BitBLT Source Start Address Bit 7	BitBLT Source Start Address Bit 6	BitBLT Source Start Address Bit 5	BitBLT Source Start Address Bit 4	BitBLT Source Start Address Bit 3	BitBLT Source Start Address Bit 2	BitBLT Source Start Address Bit 1	BitBLT Source Start Address Bit 0

REG[105h] BitBLT Source Start Address Register 1							
BitBLT Source Start Address Bit 15	BitBLT Source Start Address Bit 14	BitBLT Source Start Address Bit 13	BitBLT Source Start Address Bit 12	BitBLT Source Start Address Bit 11	BitBLT Source Start Address Bit 10	BitBLT Source Start Address Bit 9	BitBLT Source Start Address Bit 8

REG[106h] BitBLT Source Start Address Register 2							
n/a	n/a	n/a	BitBLT Source Start Address Bit 20	BitBLT Source Start Address Bit 19	BitBLT Source Start Bit 18	BitBLT Source Start Address Bit 17	BitBLT Source Start Address Bit 16

The BitBLT Source Start Address Registers form a 21-bit register that specifies the source start address for the BitBLT operation selected by the BitBLT Operation Register (REG[103h]).

If data is sourced from the CPU, then bit 0 is used for byte alignment within a 16-bit word and the other address bits are ignored. In pattern fill operation, the BitBLT Source Start Address is defined by the following equation:

Source Start Address Register = Pattern Base Address + Pattern Line Offset + Pixel Offset.

The following table shows how Source Start Address Register is defined for 8 and 16 bpp color depths:

Table 10-3 BitBLT Source Start Address Selection

Color Format	Pattern Base Address[20:0]	Pattern Line Offset[2:0]	Pixel Offset[3:0]
8 bpp	BitBLT Source Start Address[20:6], 6’b0	BitBLT Source Start Address[5:3]	1’b0, BitBLT Source Start Address[2:0]
16 bpp	BitBLT Source Start Address[20:7], 7’b0	BitBLT Source Start Address[6:4]	BitBLT Source Start Address[3:0]

REG[108h] BitBLT Destination Start Address Register 0							
BitBLT Destination Start Address Bit 7	BitBLT Destination Start Address Bit 6	BitBLT Destination Start Address Bit 5	BitBLT Destination Start Address Bit 4	BitBLT Destination Start Address Bit 3	BitBLT Destination Start Address Bit 2	BitBLT Destination Start Address Bit 1	BitBLT Destination Start Address Bit 0

REG[109h] BitBLT Destination Start Address Register 1							
BitBLT Destination Start Address Bit 15	BitBLT Destination Start Address Bit 14	BitBLT Destination Start Address Bit 13	BitBLT Destination Start Address Bit 12	BitBLT Destination Start Address Bit 11	BitBLT Destination Start Address Bit 10	BitBLT Destination Start Address Bit 9	BitBLT Destination Start Address Bit 8

REG[10Ah] BitBLT Destination Start Address Register 2							
n/a	n/a	n/a	BitBLT Destination Start Address Bit 20	BitBLT Destination Start Address Bit 19	BitBLT Destination Start Address Bit 18	BitBLT Destination Start Address Bit 17	BitBLT Destination Start Address Bit 16

The BitBLT Destination Start Address Registers form a 21-bit register that specifies the destination start address for the BitBLT operation selected by the BitBLT Operation Register (REG[103h]). The destination address represents the upper left corner of the BitBLT rectangle (lower right corner of the BitBLT rectangle for Move BitBLT in Negative Direction).

REG[10Ch] BitBLT Memory Address Offset Register 0							
BitBLT Memory Address Offset Bit 7	BitBLT Memory Address Offset Bit 6	BitBLT Memory Address Offset Bit 5	BitBLT Memory Address Offset Bit 4	BitBLT Memory Address Offset Bit 3	BitBLT Memory Address Offset Bit 2	BitBLT Memory Address Offset Bit 1	BitBLT Memory Address Offset Bit 0

REG[10Dh] BitBLT Memory Address Offset Register 1							
n/a	n/a	n/a	n/a	n/a	BitBLT Memory Address Offset Bit 10	BitBLT Memory Address Offset Bit 9	BitBLT Memory Address Offset Bit 8

The BitBLT Memory Address Offset Registers form the BitBLTs 11-bit address offset from the starting word of line “n” to the starting word of line “n + 1”. They are used for address calculation only when the BitBLT is configured as a rectangular region of memory using the BitBLT Destination/Source Linear Select bits (REG[100h] bits 1-0). They are not used for the displays.

REG[110h] BitBLT Width Register 0							
BitBLT Width Bit 7	BitBLT Width Bit 6	BitBLT Width Bit 5	BitBLT Width Bit 4	BitBLT Width Bit 3	BitBLT Width Bit 2	BitBLT Width Bit 1	BitBLT Width Bit 0

REG[111h] BitBLT Width Register 1							
n/a	n/a	n/a	n/a	n/a	n/a	BitBLT Width Bit 9	BitBLT Width Bit 8

The BitBLT Width Registers form a 10-bit register that specifies the BitBLT width in pixels less 1.

Note: The BitBLT operations Pattern Fill with ROP and Pattern Fill with transparency require a BitBLT Width > 1 for 16 bpp color depths and > 2 for 8 bpp.

REG[112h] BitBLT Height Register 0							
BitBLT Height Bit 7	BitBLT Height Bit 6	BitBLT Height Bit 5	BitBLT Height Bit 4	BitBLT Height Bit 3	BitBLT Height Bit 2	BitBLT Height Bit 1	BitBLT Height Bit 0

REG[113h] BitBLT Height Register 1							
n/a	n/a	n/a	n/a	n/a	n/a	BitBLT Height Bit 9	BitBLT Height Bit 8

The BitBLT Height Registers form a 10-bit register that specifies the BitBLT height in pixels less 1.

REG[114h] BitBLT Background Color Register 0							
BitBLT Background Color Bit 7	BitBLT Background Color Bit 6	BitBLT Background Color Bit 5	BitBLT Background Color Bit 4	BitBLT Background Color Bit 3	BitBLT Background Color Bit 2	BitBLT Background Color Bit 1	BitBLT Background Color Bit 0

REG[115h] BitBLT Background Color Register 1							
BitBLT Background Color Bit 15	BitBLT Background Color Bit 14	BitBLT Background Color Bit 13	BitBLT Background Color Bit 12	BitBLT Background Color Bit 11	BitBLT Background Color Bit 10	BitBLT Background Color Bit 9	BitBLT Background Color Bit 8

The BitBLT Background Color Registers form a 16-bit register that specifies the BitBLT background color for Color Expansion or the key color for transparent BitBLTs. For 16 bpp color depth (REG[101h] bit 0 = 1), all 16 bits are used. For 8 bpp color depth (REG[101h] bit 0 = 0), only bits 7-0 are used.

REG[118h] BitBLT Foreground Color Register 0							
BitBLT Foreground Color Bit 7	BitBLT Foreground Color Bit 6	BitBLT Foreground Color Bit 5	BitBLT Foreground Color Bit 4	BitBLT Foreground Color Bit 3	BitBLT Foreground Color Bit 2	BitBLT Foreground Color Bit 1	BitBLT Foreground Color Bit 0

REG[119h] BitBLT Foreground Color Register 1							
BitBLT Foreground Color Bit 15	BitBLT Foreground Color Bit 14	BitBLT Foreground Color Bit 13	BitBLT Foreground Color Bit 12	BitBLT Foreground Color Bit 11	BitBLT Foreground Color Bit 10	BitBLT Foreground Color Bit 9	BitBLT Foreground Color Bit 8

The BitBLT Foreground Color Registers form a 16-bit register that specifies the BitBLT foreground color for Color Expansion or the Solid Fill BitBLT. For 16 bpp color depth (REG[101h] bit 0 = 1), all 16 bits are used. For 8 bpp color depth (REG[101h] bit 0 = 0), only bits 7-0 are used.

10.2 BitBLT Descriptions

The S1D13806 supports 13 fundamental BitBLT operations:

- Write BitBLT with ROP.
- Read BitBLT.
- Move BitBLT in positive direction with ROP.
- Move BitBLT in negative direction with ROP.
- Transparent Write BitBLT.
- Transparent Move BitBLT in positive direction.
- Pattern Fill with ROP.
- Pattern Fill with Transparency.
- Color Expansion.
- Color Expansion with Transparency.
- Move BitBLT with Color Expansion.
- Move BitBLT with Color Expansion and Transparency.
- Solid Fill.

Most of the 13 operations are self completing. This means once they begin they complete on their own, not requiring data transfers with the CPU. The remaining five BitBLT operations (Write BitBLT with ROP, Transparent Write BitBLT, Color Expansion, Color Expansion with Transparency, Read BitBLT) require data to be written/read to/from the display buffer. This data must be transferred one word (16-bits) at a time. This does not imply only 16-bit CPU instructions are acceptable. If a system is able to separate one DWORD write into two WORD writes and the CPU writes the low word before the high word, then 32-bit CPU instructions are acceptable. Otherwise, 16-bit CPU instructions are required.

The data is not directly written/read to/from the display buffer. It is written/read to/from the BitBLT FIFO through the 1M BitBLT aperture specified at the address of REG[100000h]. The 16 word FIFO can be written to only when not full and can be read from only when not empty. Failing to monitor the FIFO status can result in a BitBLT FIFO overflow or underflow.

While the FIFO is being written to by the CPU, it is also being emptied by the S1D13806. If the S1D13806 empties the FIFO faster than the CPU can fill it, it may not be possible to cause an overflow/underflow. In these cases, performance can be improved by not monitoring the FIFO status. However, this is very much platform dependent and must be determined for each system.

Note: When TV with flicker filter is enabled or simultaneous display is active, **always test the FIFO status before reading from/writing to the FIFO**.

10.2.1 Write BitBLT with ROP

The Write BitBLT increases the speed of transferring data from system memory to the display buffer.

The Write BitBLT with ROP fills a specified area of the display buffer with data supplied by the CPU. This BitBLT is typically used to copy a bitmap image from system memory to the display buffer. The Write BitBLT supports all 16 ROPs, although the most frequent ROP is ROP 0Ch (Copy Source into Destination). It also supports both Destination Linear and Destination Rectangular modes.

The Write BitBLT requires the CPU to provide data. The BitBLT engine expects to receive a certain number of WORDS. For 16 bpp color depths, the number of WORDS is the same as the number of pixels due to the fact that each pixel is one WORD wide. The number of WORD writes the BitBLT engine expects is calculated using the following formula.

$$\begin{aligned}\text{nWORDS} &= \text{nPixels}\\ &= \text{BitBLTWidth} \times \text{BitBLTHeight}\end{aligned}$$

For 8 bpp color depths, the formula must take into consideration that the BitBLT engine accepts only WORD accesses and each pixel is one BYTE. The BitBLT engine needs to know whether the first pixel of a line is stored in the low byte or high byte. This is determined by bit 0 of the Source Start Address Register 0 (REG[104h]). If the Source Phase is 1 (bit 0 of the Source Start Address Register 0 is set), the first pixel of each line is in the high byte of the WORD and the contents of the low byte are ignored. If the Source Phase is 0, the first pixel is in the low byte and the second pixel is in the high byte. Depending on the Source Phase and the BitBLT Width, the last WORD may contain only one pixel. In this case it is always in the low byte. The number of WORD writes the BitBLT engine expects for 8 bpp color depths is shown in the following formula.

$$\text{nWORDS} = ((\text{BitBLTWidth} + 1 + \text{SourcePhase}) \div 2) \times \text{BitBLTHeight}$$

Note: The BitBLT engine counts WORD writes in the BitBLT address space. This does not imply only 16-bit CPU instructions are acceptable. If a system is able to separate one DWORD write into two WORD writes and the CPU writes the low word before the high word, then 32-bit CPU instructions are acceptable. Otherwise, 16-bit CPU instructions are required.

Example 9

Write a 100 × 20 rectangle at the screen coordinates x = 25, y = 38 using a 640 × 480 display at a color depth of 8 bpp.

1. Calculate the destination address (upper left corner of the screen BitBLT rectangle) using the following formula.

   $$\begin{aligned}\text{DestinationAddress} &= (\text{y} \times \text{ScreenStride}) + (\text{x} \times \text{BytesPerPixel})\\ &= (38 \times 640) + (25 \times 1)\\ &= 24345\\ &= \text{5F19h}\end{aligned}$$

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp
   ScreenStride = DisplayWidthInPixels × BytesPerPixel = 640 for 8 bpp

   Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 00h, REG[109h] is set to 5Fh, and REG[108h] is set to 19h.

2. Program the BitBLT Width Registers to 100 - 1. REG[111h] is set to 00h and REG[110h] is set to 63h (99 decimal).
3. Program the BitBLT Height Registers to 20 - 1. REG[113h] is set to 00h and REG[112h] is set to 13h (19 decimal).
4. Program the Source Phase in the BitBLT Source Start Address Register. In this example the data is WORD aligned, so the source phase is 0. REG[104h] is set to 00h.
5. Program the BitBLT Operation Register to select the Write BitBLT with ROP. REG[103h] is set to 00h.
6. Program the BitBLT ROP Code Register to select Destination = Source. REG[102h] is set to 0Ch.
7. Program the BitBLT Color Format Select bit for 8 bpp operations. REG[101h] is set to 00h.
8. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS:

   $$\begin{aligned}\text{BLTMemoryOffset} &= \text{DisplayWidthInPixels} \div \text{BytesPerPixel}\\ &= 640 \div 2\\ &= \text{140h}\end{aligned}$$

   REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

9. Calculate the number of WORDS the BitBLT engine expects to receive.

   $$\begin{aligned}\text{nWORDS} &= ((\text{BLTWidth} + 1 + \text{SourcePhase}) \div 2) \times \text{BLTHeight}\\ &= (100 + 1) \div 2 \times 20\\ &= 1000\\ &= \text{3E8h}\end{aligned}$$

10. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

    Start the BitBLT operation **and wait for the BitBLT engine to start**. REG[100h] is set to 80h, then wait until REG[100h] bit 7 returns a 1.

11. Prior to writing all nWORDS to the BitBLT FIFO, confirm the BitBLT FIFO is not full (REG[100h] bit 4 returns a 0). If the BitBLT FIFO Not Empty Status returns a 0 (the FIFO is empty), write up to 16 WORDS. If the BitBLT FIFO Not Empty Status returns a 1 and the BitBLT FIFO Half Full Status returns a 0 then you can write up to 8 WORDS. If the BitBLT FIFO Full Status returns a 1, do not write to the BitBLT FIFO until it returns a 0.

    The following table summarizes how many words can be written to the BitBLT FIFO.

Table 10-4 Possible BitBLT FIFO Writes

BitBLT Control Register 0 (REG[100h])			Word Writes Available
FIFO Not Empty Status	FIFO Half Full Status	FIFO Full Status	
0	0	0	16
1	0	0	8
1	1	0	up to 8
1	1	1	0 (do not write)

12. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is started.

10.2.2 Color Expand BitBLT

This Color Expand BitBLT is similar to the Write BitBLT. It differs in that a bit set to 1 in the source data becomes a pixel of foreground color. A source bit set to 0 is converted to a pixel of background color. This function increases the speed of writing text while in graphical modes.

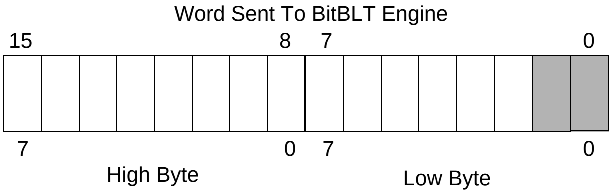
This BitBLT operation includes several options which enhance its text handling capabilities. As with the Write BitBLT, all data sent to the BitBLT engine must be word (16-bit) writes. **The BitBLT engine expands the low byte, then the high byte starting at bit 7 of each byte.** The start byte of the first WORD to be expanded and the start bit position within this byte must be specified. The start byte position is selected by setting source address bit 0 to 0 to start expanding the low byte or 1 to start expanding the high byte.

Partially “masked” color expand BitBLT can be used when drawing a portion of a pattern (i.e. a portion of a character) on the screen. The following examples illustrate how one WORD is expanded using the Color Expand BitBLT.

1. To expand bits 0-1 of the word:

   Source Address = 0
   Start Bit Position = 1
   BitBLT Width = 2

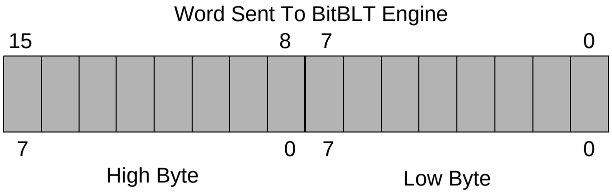
   The following bits are expanded.



2. To expand bits 0-15 of the word (entire word)

   Source Address = 0
   Start Bit Position = 7 (bit seven of the low byte)
   BitBLT Width = 16

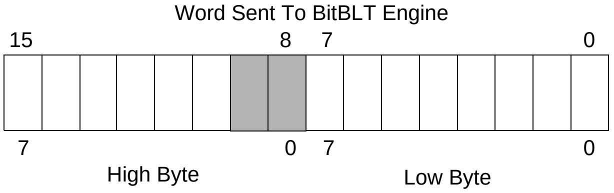
   The following bits are expanded.



3. To expand bits 8-9 of the word

   Source Address = 1
   Start Bit Position = 1
   BitBLT Width = 2

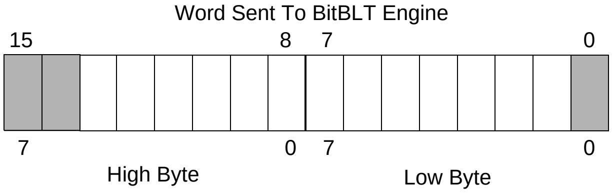
   The following bits are expanded.



4. To expand bits 0,15-14 of the word

   Source Address = 0
   Start Bit Position = 0
   BitBLT Width = 3

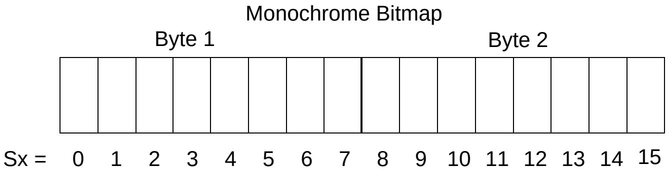
   The following bits are expanded.



All subsequent WORDS in one BitBLT line are then serially expanded starting at bit 7 of the low byte until the end of the BitBLT line. All unused bits in the last WORD are discarded. It is extremely important that the exact number of WORDS is provided to the BitBLT engine. The number of WORDS is calculated from the following formula. This formula is valid for all color depths (8/16 bpp).

$$\text{nWords} = ((\text{Sx MOD } 16 + \text{BitBLTWidth} + 15) \div 16) \times \text{BitBLTHeight}$$

where:
Sx is the X coordinate of the starting pixel in a word aligned monochrome bitmap.



Example 10
Color expand a rectangle of 12 × 18 starting at the coordinates Sx = 125, Sy = 17 using a 640 × 480 display at a color depth of 8 bpp.

This example assumes a monochrome, WORD aligned bitmap of dimensions 300 × 600 with the origin at an address A. The color expanded rectangle will be displayed at the screen coordinates X = 20, Y = 30. The foreground color corresponds to the LUT entry at index 134, the background color to index 124.

1. First we need to calculate the address of the WORD within the monochrome bitmap containing the pixel x = 125,y = 17.

$$\begin{aligned}\text{SourceAddress} &= \text{BitmapOrigin} + (y \times \text{SourceStride}) + (x \div 8)\\ &= A + (Sy \times \text{SourceStride}) + (Sx \div 8)\\ &= A + (17 \times 38) + (125 \div 8)\\ &= A + 646 + 15\\ &= A + 661\end{aligned}$$

   where:

$$\begin{aligned}\text{SourceStride} &= (\text{BitmapWidth} + 15) \div 16\\ &= (300 + 15) \div 16\\ &= 19 \text{ WORDS per line}\\ &= 38\text{BYTES per line}\end{aligned}$$

2. Calculate the destination address (upper left corner of the screen BitBLT rectangle) using the following formula.

$$\begin{aligned}\text{DestinationAddress} &= (Y \times \text{ScreenStride}) + (X \times \text{BytesPerPixel})\\ &= (30 \times 640) + (20 \times 1)\\ &= 19220\\ &= \text{4B14h}\end{aligned}$$

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp
   ScreenStride = DisplayWidthInPixels × BytesPerPixel = 640 for 8 bpp

   Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 00h, REG[109h] is set to 4Bh, and REG[108h] is set to 14h.

3. Program the BitBLT Width Registers to 12 - 1. REG[111h] is set to 00h, REG[110h] is set to 0Bh (11 decimal).

4. Program the BitBLT Height Registers to 18 - 1. REG[113h] is set to 00h, REG[112h] is set to 11h (17 decimal).

5. Program the Source Phase in the BitBLT Source Start Address Register. In this example the source address equals A + 661 (odd), so REG[104h] is set to 1.

   Since only bit 0 flags the source phase, more efficient code would simply write the low byte of the SourceAddress into REG[104h] directly -- not needing to test for an odd/even address. Note that in 16 bpp color depths the Source address is guaranteed to be even.

6. Program the BitBLT Operation Register to select the Color Expand BitBLT. REG[103h] is set to 08h.
7. Program the Color Expansion Register. The formula for this example is as follows.

   Reg[102h] = 7 - (Sx MOD 8)
   = 7 - (125 MOD 8)
   = 7 - 5
   = 2

   REG[102h] is set to 02h.
8. Program the Background Color Registers to the background color. REG[115h] is set to 00h and REG[114h] is set to 7Ch (124 decimal).

   Note that for 16 bpp color depths, REG[115h] and REG[114h] are both required and programmed directly with the value of the background color.
9. Program the Foreground Color Registers to the foreground color. REG[119h] is set to 00h and REG[118h] is set to 86h (134 decimal).

   Note that for 16 bpp color depths REG[119h] and Reg[118h] are both required and programmed directly with the value of the foreground color.
10. Program the BitBLT Color Format Register for 8 bpp operation. REG[101h] is set to 00h.
11. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

    BltMemoryOffset = ScreenStride ÷ 2
    = 640 ÷ 2
    = 140h

    REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.
12. Calculate the number of WORDS the BitBLT engine expects to receive.

    First, the number of WORDS in one BitBLT line must be calculated as follows.

    nWordsOneLine = ((125 MOD 16) + 12 + 15) ÷ 16
    = (13 + 12 + 15) ÷ 16
    = 40 ÷ 16
    = 2

    Therefore, the total WORDS the BitBLT engine expects to receive is calculated as follows.

    nWords = nWordsOneLine × 18
    = 2 × 18
    = 36

13. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

    Start the BitBLT operation and **wait for the BitBLT Engine to start**. REG[100h] is set to 80h, then wait until REG[100h] bit 7 returns a 1.

14. Prior to writing all nWORDS to the BitBLT FIFO, confirm the BitBLT FIFO is not full (REG[100h] bit 4 returns a 0). One WORD expands into 16 pixels which fills all 16 FIFO words in 16 bpp or 8 FIFO words in 8 bpp.

    The following table summarizes how many words can be written to the BitBLT FIFO.

Table 10-5 Possible BitBLT FIFO Writes

BitBLT Control Register 0 (REG[100h])			8 bpp Word Writes Available	16 bpp Word Writes Available
FIFO Not Empty Status	FIFO Half Full Status	FIFO Full Status		
0	0	0	2	1
1	0	0	1	0 (do not write)
1	1	0	0 (do not write)	
1	1	1		

15. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.2.3 Color Expand BitBLT With Transparency

This BitBLT operation is virtually identical to the Color Expand BitBLT, except the background color is completely ignored. All bits set to 1 in the source monochrome bitmap are color expanded to the foreground color. All bits set to 0 that would be expanded to the background color in the Color Expand BitBLT are not expanded at all.

Program REG[103h] to 09h instead of 08h. Programming the background color is not required.

10.2.4 Solid Fill BitBLT

The Solid Fill BitBLT fills a rectangular area of the display buffer with a solid color. This operation is used to paint large screen areas or to set areas of the display buffer to a given value.

Example 11

Fill a red 9 × 321 rectangle at the screen coordinates x = 100, y = 10 using a 640 × 480 display at a color depth of 16 bpp.

1. Calculate the destination address (upper left corner of the destination rectangle) using the following formula.

   $$\begin{aligned}\text{DestinationAddress} &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel})\\ &= (10 \times (640 \times 2)) + (100 \times 2)\\ &= 13000\\ &= \text{32C8h}\end{aligned}$$

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp
   ScreenStride = DisplayWidthInPixels × BytesPerPixel = 1280 for 16 bpp.

   Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 00h, REG[109h] is set to 32h, and REG[108h] is set to C8h.

2. Program the BitBLT Width Registers to 9 - 1. REG[111h] is set to 00h and REG[110h] is set to 08h.
3. Program the BitBLT Height Registers to 321 - 1. REG[113h] is set to 01h and REG[112h] is set to 40h (320 decimal).
4. Program the BitBLT Foreground Color Registers. REG[119h] is set to F8h and REG[118h] is set to 00h (Full intensity red in 16 bpp is F800h).
5. Program the BitBLT Operation Register to select Solid Fill. REG[103h] is set to 0Ch.
6. Program the BitBLT Color Format Register for 16 bpp operations. REG[101h] is set to 01h.
7. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

   $$\begin{aligned}\text{BltMemoryOffset} &= \text{ScreenStride} \div 2\\ &= 640\\ &= \text{280h}\end{aligned}$$

   REG[10Dh] is set to 02h and REG[10Ch] is set to 80h.

8. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the BitBLT operation. REG[100h] is set to 80h.

9. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.2.5 Move BitBLT in a Positive Direction with ROP

The Move BitBLT moves an area of the display buffer to a different area of the display buffer. This operation has two intended purposes:

- Copying unattended display buffer to display buffer.
- Saving a visible bitmap to off-screen display buffer.

The Move BitBLT may move data from one rectangular area to another, or it may be specified as linear. This allows the temporary saving of a portion of the visible display buffer to an area off-screen. The linear configuration may be applied to the source or destination. Defining the Move BitBLT as linear allows each line of the Move BitBLT area to be placed directly after the previous line, rather than requiring a complete row of address space for each line.

Note: When the destination area overlaps the original source area and the destination address is greater then the source address, use the Move BitBLT in Negative Direction with ROP.

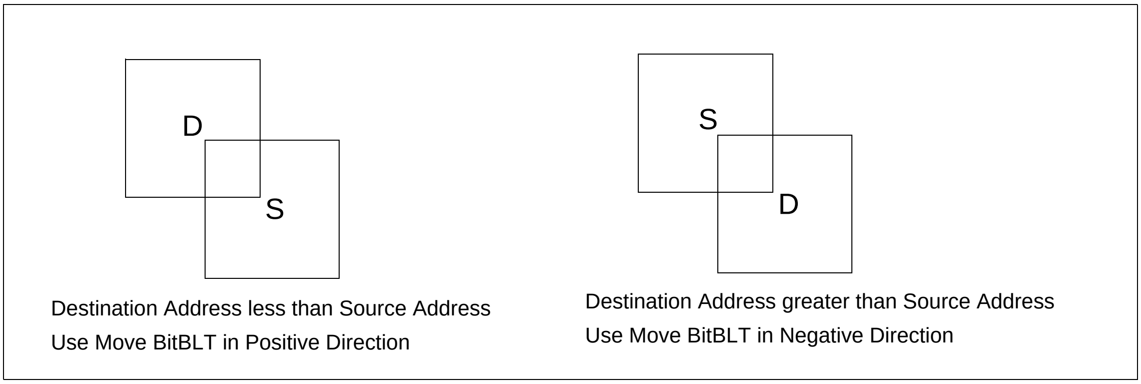

D
S
Destination Address less than Source Address
Use Move BitBLT in Positive Direction
S
D
Destination Address greater than Source Address
Use Move BitBLT in Negative Direction

Figure 10-1 Move BitBLT Usage

Example 12
Copy a 9 × 321 rectangle at the screen coordinates x = 100, y = 10 to screen coordinates x = 200, y = 20 using a 640 × 480 display at a color depth of 16 bpp.

1. Calculate the source and destination addresses (upper left corners of the source and destination rectangles), using the following formula.

SourceAddress	= (y × ScreenStride) + (x × BytesPerPixel)
	= (10 × (640 × 2)) + (100 × 2)
	= 13000
	= 32C8h

DestinationAddress	= (y × ScreenStride) + (x × BytesPerPixel)
	= (20 × (640 × 2)) + (200 × 2)
	= 26000
	= 6590h

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp
   ScreenStride = DisplayWidthInPixels × BytesPerPixel = 1280 for 16 bpp

   Program the BitBLT Source Start Address Registers. REG[106h] is set to 00h, REG[105h] is set to 32h, and REG[104h] is set to C8h.

   Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 00h, REG[109h] is set to 65h, and REG[108h] is set to 90h.

2. Program the BitBLT Width Registers to 9 - 1. REG[111h] is set to 00h and REG[110h] is set to 08h.
3. Program the BitBLT Height Registers to 321 - 1. REG[113h] is set to 01h and REG[112h] is set to 40h (320 decimal).
4. Program the BitBLT Operation Register to select the Move BitBLT in Positive Direction with ROP. REG[103h] is set to 02h.
5. Program the BitBLT ROP Code Register to select Destination = Source. REG[102h] is set to 0Ch.
6. Program the BitBLT Color Format Select bit for 16 bpp operations. REG[101h] is set to 01h.
7. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

BltMemoryOffset	= ScreenStride ÷ 2
	= 640
	= 280h

   REG[10Dh] is set to 02h and REG[10Ch] is set to 80h.

8. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the BitBLT operation. REG[100h] is set to 80h.

9. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.2.6 Move BitBLT in Negative Direction with ROP

The Move BitBLT in Negative Direction with ROP is very similar to the Move BitBLT in Positive direction and must be used when the source and destination BitBLT areas overlap and the destination address is greater then the source address.

Note: For the Move BitBLT in Negative Direction it is necessary to calculate the addresses of the last pixel as opposed to the first pixel. This means calculating the addresses of the lower right corners as opposed to the upper left corners.

Example 13

Copy a 9 × 321 rectangle at the screen coordinates x = 100, y = 10 to screen coordinates X = 105, Y = 20 using a 640 × 480 display at a color depth of 16 bpp.

In the following example, the coordinates of the source and destination rectangles intentionally overlap.

1. Calculate the source and destination addresses (**lower right** corners of the source and destination rectangles) using the following formula.

SourceAddress
= ((y + Height - 1) × ScreenStride) + ((x + Width - 1) × BytesPerPixel)
= ((10 + 321 - 1) × (640 × 2)) + ((100 + 9 - 1) × 2)
= 422616
= 672D8h

DestinationAddress
= ((Y + Height - 1) × ScreenStride) + ((X + Width - 1) × BytesPerPixel)
= ((20 + 321 - 1) × (640 × 2)) + ((105 + 9 - 1) × 2)
= 435426
= 6A4E2h

where:
BytesPerPixel = 1 for 8 bpp
BytesPerPixel = 2 for 16 bpp
ScreenStride = DisplayWidthInPixels × BytesPerPixel = 1280 for 16 bpp

Program the BitBLT Source Start Address Registers. REG[106h] is set to 06h, REG[105h] is set to 72h, and REG[104h] is set to D8h.

Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 06h, REG[109h] is set to A4h, and REG[108h] is set to E2h.

2. Program the BitBLT Width Registers to 9 - 1. REG[111h] is set to 00h and REG[110h] is set to 08h.
3. Program the BitBLT Height Registers to 321 - 1. REG[113h] is set to 01h and REG[112h] is set to 40h (320 decimal).
4. Program the BitBLT Operation Register to select the Move BitBLT in Negative Direction with ROP. REG[103] is set to 03h.
5. Program the BitBLT ROP Code Register to select Destination = Source. REG[102h] is set to 0Ch.
6. Program the BitBLT Color Format Select bit for 16 bpp operations. REG[101h] is set to 01h.
7. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

   BltMemoryOffset = ScreenStride ÷ 2
   = 640
   = 280h

   REG[10Dh] is set to 02h and REG[10Ch] is set to 80h.
8. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the BitBLT operation. REG[100h] is set to 80h.
9. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.2.7 Transparent Write BitBLT

The Transparent Write BitBLT increases the speed of transferring data from system memory to the display buffer. Once the Transparent Write BitBLT begins, the BitBLT engine remains active until all pixels have been written.

The Transparent Write BitBLT updates a specified area of the display buffer with data supplied by the CPU. This BitBLT is typically used to copy a bitmap image from system memory to the display buffer with one color marked as transparent. Any pixel of the transparent color is not transferred. This allows fast display of non-rectangular images. For example, consider a source bitmap having a red circle on a blue background. By selecting the blue color as the transparent color and using the Transparent Write BitBLT on the whole rectangle, the effect is a BitBLT of the red circle only. The Transparent Write BitBLT supports both Destination Linear and Destination Rectangular modes.

The Transparent Write BitBLT requires the CPU to provide data. The BitBLT engine expects to receive a certain number of WORDS. For 16 bpp color depths, the number of WORDS is the same as the number of pixels due to the fact that each pixel is one WORD wide. The number of WORD writes the BitBLT engine expects is calculated using the following formula.

$$\begin{aligned} \text{nWORDS} &= \text{nPixels} \\ &= \text{BitBLT Width} \times \text{BitBLT Height} \end{aligned}$$

For 8 bpp color depths, the formula must take into consideration that the BitBLT engine accepts only WORD accesses and each pixel is one BYTE. The BitBLT engine needs to know whether the first pixel of a line is stored in the low byte or high byte. This is determined by bit 0 of the Source Start Address Register 0 (REG[104h]). If the Source Phase is 1 (bit 0 of the Source Start Address Register 0 is set), the first pixel of each line is in the high byte of the WORD and the contents of the low byte are ignored. If the Source Phase is 0, the first pixel is in the low byte and the second pixel is in the high byte. Depending on the Source Phase and the BitBLT Width, the last WORD in each line may contain only one pixel. It is always in the low byte if more than one WORD per line is required. The number of WORD reads the BitBLT engine expects for 8 bpp color depths is shown in the following formula.

$$\text{nWORDS} = ((\text{BitBLTWidth} + 1 + \text{SourcePhase}) \div 2) \times \text{BitBLTHeight}$$

Note: The BitBLT engine counts WORD writes in the BitBLT address space. This does not imply only 16-bit CPU instructions are acceptable. If a system is able to separate one DWORD write into two WORD writes, then 32-bit CPU instructions are acceptable. Otherwise, 16-bit CPU instructions are required.

Example 14

Write 100 × 20 pixels at the screen coordinates x = 25, y = 38 using a 640 × 480 display at a color depth of 8 bpp. Transparent color is high intensity blue (assumes LUT Index 124).

1. Calculate the destination address (upper left corner of the screen BitBLT rectangle), using the formula:

   DestinationAddress = (y × ScreenStride) + (x × BytesPerPixel)
   = (38 × 640) + (25 × 1)
   = 24345
   = 5F19h

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp
   ScreenStride = DisplayWidthInPixels × BytesPerPixel = 640 for 8 bpp

   Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 00h, REG[109h] is set to 5Fh, and REG[108h] is set to 19h.

2. Program the BitBLT Width Registers to 100 - 1. REG[111h] is set to 00h and REG[110h] is set to 63h (99 decimal).
3. Program the BitBLT Height Registers to 20 - 1. REG[113h] is set to 00h and REG[112h] is set to 13h (19 decimal).
4. Program the Source Phase in the BitBLT Source Start Address Register. In this example, the data is WORD aligned, so the source phase is 0. REG[104h] is set to 00h.
5. Program the BitBLT Operation Register to select Transparent Write BitBLT. REG[103h] is set to 04h.
6. Program the BitBLT Background Color Registers to select transparent color. REG[114h] is set to 7Ch (124 decimal).

   Note that for 16 bpp color depths, REG[115h] and REG[114h] are both required and programmed directly with the value of the transparent background color.

7. Program the BitBLT Color Format Select bit for 8 bpp operations. REG[101h] is set to 00h.
8. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

   BltMemoryOffset = ScreenStride ÷ 2
   = 640 ÷ 2
   = 320
   = 140h

   REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

9. Calculate the number of WORDS the BitBLT engine expects to receive.

   nWORDS = ((BLTWidth + 1 + SourcePhase) ÷ 2) × BLTHeight
   = (100 + 1 + 0) ÷ 2 × 20
   = 1000
   = 3E8h

10. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

    Start the BitBLT operation **and wait for the BitBLT engine to start**. REG[100h] is set to 80h, then wait until REG[100h] bit 7 returns a 1.

11. Prior to writing all nWORDS to the BitBLT FIFO, confirm the BitBLT FIFO is not full (REG[100h] bit 4 returns a 0). If the BitBLT FIFO Not Empty Status returns a 0 (the FIFO is empty), write up to 16 WORDS. If the BitBLT FIFO Not Empty Status returns a 0 and the BitBLT FIFO Half Full Status returns a 0 then you can write up to 8 WORDS. If the BitBLT FIFO Full Status returns a 1, do not write to the BitBLT FIFO until it returns a 0.

    The following table summarizes how many words can be written to the BitBLT FIFO.

Table 10-6 Possible BitBLT FIFO Writes

BitBLT Control Register 0 (REG[100h])			Word Writes Available
FIFO Not Empty Status	FIFO Half Full Status	FIFO Full Status	
0	0	0	16
1	0	0	8
1	1	0	less than 8
1	1	1	0 (do not write)

12. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.2.8 Transparent Move BitBLT in Positive Direction

The Transparent Move BitBLT in Positive Direction moves an area of the display buffer to a different area of the display buffer. It allows for selection of a transparent color which is not copied during the BitBLT. This allows fast display of non-rectangular images. For example, consider a source bitmap having a red circle on a blue background. By selecting the blue color as the transparent color and using the Transparent Move BitBLT on the whole rectangle, the effect is a BitBLT of the red circle only.

The Transparent Move BitBLT may move data from one rectangular area to another, or it may be specified as linear. The linear configuration may be applied to the source or destination. Defining the Move BitBLT as linear allows each line of the Move BitBLT area to be placed directly after the previous line, rather than requiring a complete row of address space for each line.

Note: The Transparent Move BitBLT is supported **only** in a positive direction.

Example 15

Copy a 9 × 321 rectangle at the screen coordinates x = 100, y = 10 to screen coordinates X = 200, Y = 20 using a 640 × 480 display at a color depth of 16 bpp. Transparent color is blue.

1. Calculate the source and destination addresses (upper left corners of the source and destination rectangles), using the formula:

   SourceAddress = (y × ScreenStride) + (x × BytesPerPixel)
   = (10 × (640 × 2)) + (100 × 2)
   = 13000
   = 32C8h

   DestinationAddress = (Y × ScreenStride) + (X × BytesPerPixel)
   = (20 × (640 × 2)) + (200 × 2)
   = 26000
   = 6590h

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp
   ScreenStride = DisplayWidthInPixels × BytesPerPixel = 1280 for 16 bpp

   Program the BitBLT Source Start Address Registers. REG[106h] is set to 00h, REG[105h] is set to 32h, and REG[104h] is set to C8h.

   Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 00h, REG[109h] is set to 65h, and REG[108h] is set to 90h.

2. Program the BitBLT Width Registers to 9 - 1. REG[111h] is set to 00h and REG[110h] is set to 08h.

3. Program the BitBLT Height Registers to 321 - 1. REG[113h] is set to 01h and REG[112h] is set to 40h (320 decimal).

4. Program the BitBLT Operation Register to select the Transparent Move BitBLT in Positive Direction. REG[103h] is set to 05h.
5. Program the BitBLT Background Color Registers to select blue as the transparent color. REG[115h] is set to 00h and REG[114h] is set to 1Fh (Full intensity blue in 16 bpp is 001Fh).
6. Program the BitBLT Color Format Register to select 16 bpp operations. REG[101h] is set to 01h.
7. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

   $$\begin{aligned} \text{BltMemoryOffset} &= \text{ScreenStride} \div 2 \\ &= 640 \\ &= 280\text{h} \end{aligned}$$

   REG[10Dh] is set to 02h and REG[10Ch] is set to 80h.
8. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the BitBLT operation. REG[100h] is set to 80h.
9. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.2.9 Pattern Fill BitBLT with ROP

The Pattern Fill BitBLT with ROP fills a specified rectangular area of the display buffer with a pattern. The fill pattern is an array of pixels stored in off-screen display buffer. The fill pattern is limited to an eight by eight pixel array and must be loaded to off-screen memory prior to the BitBLT starting. The pattern can be logically combined with the destination using all 16 ROP codes, but typically the copy pattern ROP is used (ROP code 0Ch).

The pattern itself must be stored in a consecutive array of pixels. As a pattern is defined to be 8 × 8 pixels, this results in 64 consecutive bytes for 8 bpp color depths and 128bytes for 16 bpp color depths. For 8 bpp color depths the pattern must begin on a 64byte boundary, for 16 bpp color depths the pattern must begin on a 128byte boundary.

To fill an area using the pattern BitBLT, the BitBLT engine requires the location of the pattern, the destination rectangle position and size, and the ROP code. The BitBLT engine also needs to know which pixel from the pattern is the first pixel in the destination rectangle (the pattern start phase). This allows seamless redrawing of any part of the screen using the pattern fill.

Example 16

Fill a 100 × 250 rectangle at the screen coordinates x = 10, y = 20 with the pattern in off-screen memory at offset 10 0000h using a 640 × 480 display at a color depth of 8 bpp. The first pixel (upper left corner) of the rectangle is the pattern pixel at x = 3, y = 4.

1. Calculate the destination address (upper left corner of the destination rectangle), using the formula:

   DestinationAddress = (y × ScreenStride) + (x × BytesPerPixel)
   = (20 × 640) + (10 × 1)
   = 12810
   = 320Ah

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp
   ScreenStride = DisplayWidthInPixels × BytesPerPixels = 640 for 8 bpp

   Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 00h, REG[109h] is set to 32h, and REG[108h] is set to 0Ah.

2. Calculate the source address. This is the address of the pixel in the pattern that is the origin of the destination fill area. The pattern begins at offset 1M, but the first pattern pixel is at x = 3, y = 4. Therefore, an offset within the pattern itself must be calculated.

   SourceAddress
   = PatternOffset + StartPatternY × 8 × BytesPerPixel + StartPatternX × BytesPerPixel
   = 1M + (4 × 8 × 1) + (3 × 1)
   = 1M + 35
   = 1048611
   = 100023h

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp

   Program the BitBLT Source Start Address Registers. REG[106h] is set to 10h, REG[105h] is set to 00h, and REG[104h] is set 23h.

3. Program the BitBLT Width Registers to 100 - 1. REG[111h] is set to 00h, REG[110h] is set to 63h (99 decimal).

4. Program the BitBLT Height Registers to 250-1. REG[113h] is set to 00h, and REG[112h] is set to F9h (249 decimal).

5. Program the BitBLT Operation Register to select the Pattern Fill with ROP. REG[103h] is set to 06h.

6. Program the BitBLT ROP Code Register to select Destination = Source. REG[102h] is set to 0Ch.

7. Program the BitBLT Color Format Select bit for 8 bpp operations. REG[101h] is set to 00h.

8. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

   BltMemoryOffset = ScreenStride ÷ 2
   = 640 ÷ 2
   = 320
   = 140h

   REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

9. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the BitBLT operation. REG[100h] is set to 80h.

10. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.2.10 Pattern Fill BitBLT with Transparency

The Pattern Fill BitBLT with Transparency fills a specified rectangular area of the display buffer with a pattern. When a transparent color is selected, pattern pixels of the transparent color will not get copied, allowing creation of hatched patterns. The fill pattern is an eight by eight array of pixels stored in off-screen display buffer. The fill pattern must be loaded to off-screen display buffer prior to the BitBLT starting.

The pattern itself must be stored in a consecutive array of pixels. As a pattern is defined to be eight pixels square, this results in 64 consecutive bytes for 8 bpp color depths and 128bytes for 16 bpp color depths. For 8 bpp color depths the pattern must begin on a 64byte boundary, for 16 bpp color depths the pattern must begin on a 128byte boundary.

To fill an area using the Pattern Fill BitBLT with Transparency, the BitBLT engine requires the location of the pattern, the destination rectangle position and size, and the transparency color. The BitBLT engine also needs to know which pixel from the pattern is the first pixel in the destination rectangle (the pattern start phase). This allows seamless redrawing of any part of the screen using the pattern fill.

Example 17

Fill a 100 × 250 rectangle at the screen coordinates x = 10, y = 20 with the pattern in off-screen memory at offset 10000h using a 640 × 480 display at a color depth of 8 bpp. The first pixel (upper left corner) of the rectangle is the pattern pixel at x = 3, y = 4. Transparent color is blue (assumes LUT index 1).

1. Calculate the destination address (upper left corner of destination rectangle), using the formula:

   $$\begin{aligned}\text{DestinationAddress} &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel}) \\ &= (20 \times 640) + (10 \times 1) \\ &= 12810 \\ &= 320\text{Ah}\end{aligned}$$

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp
   ScreenStride = DisplayWidthInPixels × BytesPerPixels = 640 for 8 bpp

   Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 00h, REG[109h] is set to 32h, and REG[108h] is set to 0Ah.

2. Calculate the source address. This is the address of the pixel in the pattern that is the origin of the destination fill area. The pattern begins at offset 1M, but the first pattern pixel is at x = 3, y = 4. Therefore, an offset within the pattern itself must be calculated.

   SourceAddress

   $$\begin{aligned}&= \text{PatternOffset} + \text{StartPatternY} \times 8 \times \text{BytesPerPixel} + \text{StartPatternX} \times \text{BytesPerPixel} \\ &= 1\text{M} + (4 \times 8 \times 1) + (3 \times 1) \\ &= 1\text{M} + 35 \\ &= 1048611 \\ &= 100023\text{h}\end{aligned}$$

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp

   Program the BitBLT Source Start Address Registers. REG[106h] is set to 10h, REG[105h] is set to 00h, and REG[104h] is set 23h.

3. Program the BitBLT Width Registers to 100 - 1. REG[111h] is set to 00h and REG[110h] is set to 63h (99 decimal).

4. Program the BitBLT Height Registers to 250-1. REG[113h] is set to 00h, and REG[112h] is set to F9h (249 decimal).

5. Program the BitBLT Operation Register to select the Pattern Fill BitBLT with Transparency. REG[103h] is set to 07h.

6. Program the BitBLT Background Color Registers to select transparent color. This example uses blue (LUT index 1) as the transparent color. REG[114h] is set to 01h.

   Note that for 16 bpp color depths, REG[115h] and REG[114h] are both required and programmed directly with the value of the transparent background color. For example, for full intensity green to be the transparent color in 16 bpp, REG[115h] is set to 07h and REG[114h] is set to E0h.

7. Program the BitBLT Color Format Select bit for 8 bpp operations. REG[101h] is set to 00h.

8. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

   $$\begin{aligned}\text{BltMemoryOffset} &= \text{ScreenStride} \div 2\\ &= 640 \div 2\\ &= 320\\ &= \text{140h}\end{aligned}$$

   REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

9. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the BitBLT operation. REG[100h] is set to 80h.

10. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.2.11 Move BitBLT with Color Expansion

The Move BitBLT with Color Expansion takes a monochrome bitmap as the source and color expands it into the destination. Color expansion moves all bits in the monochrome source to pixels in the destination. All bits in the source set to one are expanded into destination pixels of the selected foreground color. All bits in the source set to zero are expanded into pixels of the selected background color.

The Move BitBLT with Color Expansion is used to accelerate text drawing on the screen. A monochrome bitmap of a font in off-screen memory occupies very little space and takes advantage of the hardware acceleration. Since the foreground and background colors are programmable, text of any color can be created.

The Move BitBLT with Color Expansion may move data from one rectangular area to another, or it may be specified as linear. The linear configuration may be applied to the source or destination. Defining the Move BitBLT as linear allows each line of the Move BitBLT area to be placed directly after the previous line, rather than requiring a complete row of address space for each line.

Note: The BitBLT ROP Code/Color Expansion Register must be programmed to value 07h. Therefore, the first word in a line color expansion starts with the most significant bit of the low or high byte.

Example 18
Color expand a 9 × 16 rectangle using the pattern in off-screen memory at 100000h and move it to the screen coordinates x = 200, y = 20. Assume a 640 × 480 display at a color depth of 16 bpp, Foreground color of black, and background color of white.

1. Calculate the destination and source addresses (upper left corner of the destination and source rectangles), using the formula.

   DestinationAddress = (y × ScreenStride) + (x × BytesPerPixel)
   = (20 × (640 × 2)) + (200 × 2)
   = 26000
   = 6590h

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 16 bpp
   ScreenStride = DisplayWidthInPixels × BytesPerPixels = 1280 for 16 bpp

   SourceAddress = 1M
   = 100000h

   Program the BitBLT Destination Start Address Registers. REG[10Ah] is set to 00h, REG[109h] is set to 65h, and REG[108h] is set to 90h.

   Program the BitBLT Source Start Address Registers. REG[106h] is set to 10h, REG[105h] is set to 00h, and REG[104h] is set to 00h.

2. Program the BitBLT Width Registers to 9 - 1. REG[111h] is set to 00h and REG[110h] is set to 08h.
3. Program the BitBLT Height Registers to 16 - 1. REG[113h] is set to 00h and REG[112h] is set to 0Fh.
4. Program the BitBLT ROP Code/Color Expansion Register. REG[102h] is set to 07h.
5. Program the BitBLT Operation Register to select the Move BitBLT with Color Expansion. REG[103h] is set to 0Bh.
6. Program the BitBLT Foreground Color Register to select black (in 16 bpp black = 0000h). REG[119h] is set to 00h and REG[118h] is set to 00h.
7. Program the BitBLT Background Color Register to select white (in 16 bpp white = FFFFh). REG[115h] is set to FFh and REG[114h] is set to FFh.
8. Program the BitBLT Color Format Select bit for 16 bpp operations. REG[101h] is set to 01h.
9. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

   BltMemoryOffset = ScreenStride ÷ 2
   = 640
   = 280h

   REG[10Dh] is set to 02h and REG[10Ch] is set to 80h.
10. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

    Start the BitBLT operation. REG[100h] is set to 80h.
11. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.2.12 Transparent Move BitBLT with Color Expansion

The Transparent Move BitBLT with Color Expansion is virtually identical to the Move BitBLT with Color Expansion. The background color is ignored and bits in the monochrome source bitmap set to 0 are not color expanded.

10.2.13 Read BitBLT

This Read BitBLT increases the speed of transferring data from the display buffer to system memory. This BitBLT complements the Write BitBLT and is typically used to save a part of the display buffer to the system memory. Once the Read BitBLT begins, the BitBLT engine remains active until all the pixels have been read.

The BitBLT engine requires the address to copy from and the size of the area to copy (width × height). The BitBLT engine expects to read a certain number of words. For 16 bpp color depths, the number of words is the same as the number of pixels due to the fact that each pixel is one WORD wide. The number of WORD reads the BitBLT engine expects is calculated using the following formula.

$$\begin{aligned} \text{nWORDS} &= \text{nPixels} \\ &= \text{BitBLT Width} \times \text{BitBLT Height} \end{aligned}$$

For 8 bpp color depths, the formula must take into consideration that the BitBLT engine accepts only WORD accesses and each pixel is one BYTE. The BitBLT engine needs to know whether the first pixel of each line is stored in the low byte or high byte. This is determined by bit 0 of the Destination Start Address Register 0 (REG[108h]). If the Destination Phase is 1 (bit 0 of the Destination Start Address Register 0 is set), the first pixel of each line is placed in the high byte of the WORD and the contents of the low byte is undefined. If the Destination Phase is 0, the first pixel is placed in the low byte and the second pixel is placed in the high byte. Depending on the Destination Phase and the BitBLT Width, the last WORD in each line may contain only one pixel. It is always in the low byte if more than one WORD per line is required. The number of WORD reads the BitBLT engine expects for 8 bpp color depths is shown in the following formula.

$$\text{nWORDS} = ((\text{BLTWidth} + 1 + \text{DestinationPhase}) \div 2) \times \text{BLTHeight}$$

Example 19

Read 100 × 20 pixels at the screen coordinates x = 25, y = 38 and save to system memory. Assume a display of 640 × 480 at a color depth of 8 bpp.

1. Calculate the source address (upper left corner of the screen BitBLT rectangle), using the formula.

$$\begin{aligned} \text{SourceAddress} &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel}) \\ &= (38 \times 640) + (25 \times 1) \\ &= 24345 \\ &= \text{5F19h} \end{aligned}$$

where:
BytesPerPixel = 1 for 8 bpp
BytesPerPixel = 2 for 16 bpp
ScreenStride = DisplayWidthInPixels × BytesPerPixels = 640 for 8 bpp

Program the BitBLT Source Start Address Registers. REG[106h] is set to 00h, REG[105h] is set to 5Fh, and REG[104h] is set to 19h.

2. Program the BitBLT Width Registers to 100 - 1. REG[111h] is set to 00h and REG[110h] is set to 63h (99 decimal).

3. Program the BitBLT Height Registers to 20 - 1. REG[113h] is set to 00h and REG[112h] is set to 13h (19 decimal).

4. Program the Destination Phase in the BitBLT Destination Start Address Register. In this example, the data is WORD aligned, so the destination phase is 0. REG[108h] is set to 0.

5. Program the BitBLT Operation to select the Read BitBLT. REG[103h] is set to 01h.

6. Program the BitBLT Color Format Select bit for 8 bpp operations. REG[101h] is set to 00h.

7. Program the BitBLT Memory Offset Registers to the ScreenStride in WORDS.

   $$\begin{aligned}\text{BltMemoryOffset} &= \text{ScreenStride} \div 2\\ &= 640 \div 2\\ &= 320\\ &= 140\text{h}\end{aligned}$$

   REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

8. Calculate the number of WORDS the BitBLT engine expects to receive.

   $$\begin{aligned}\text{nWORDS} &= ((\text{BLTWidth} + 1 + \text{DestinationPhase}) \div 2) \times \text{BLTHeight}\\ &= (100 + 1 + 0) \div 2 \times 20\\ &= 1000\\ &= 3\text{E8h}\end{aligned}$$

9. Program the BitBLT Destination/Source Linear Select bits for a rectangular BitBLT (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the BitBLT operation **and wait for the BitBLT engine to start**. REG[100h] is set to 80h, then wait until REG[100h] bit 7 returns a 1.

10. Prior to reading all nWORDS from the BitBLT FIFO, confirm the BitBLT FIFO is not empty (REG[100h] bit 4 returns a 1). If the BitBLT FIFO Not Empty Status returns a 1 and the BitBLT FIFO Half Full Status returns a 0 then you can read up to 8 WORDS. If the BitBLT FIFO Full Status returns a 1, read up to 16 WORDS. If the BitBLT FIFO Not Empty Status returns a 0 (the FIFO is empty), do not read from the BitBLT FIFO until it returns a 1.

    The following table summarizes how many words can be read from the BitBLT FIFO.

Table 10-7 Possible BitBLT FIFO Reads

BitBLT Control Register 0 (REG[100h])			Word Reads Available
FIFO Not Empty Status	FIFO Half Full Status	FIFO Full Status	
0	0	0	0 (do not read)
1	0	0	up to 8
1	1	0	8
1	1	1	16

11. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.3 S1D13806 BitBLT Synchronization

A BitBLT operation can only be started if the BitBLT engine is not busy servicing another BitBLT. Before a new BitBLT operation is started, software must confirm the BitBLT Active Status bit (REG[100h] bit 7) returns a 0. Software can either test this bit **after** each BitBLT operation, or **before** each BitBLT operation.

Testing the BitBLT Status After

Testing the BitBLT Active Status after starting a new BitBLT is simpler and less prone to errors. To test after each BitBLT operation, perform the following.

1. Program and start the BitBLT engine.
2. Wait for the current BitBLT operation to finish -- Poll the BitBLT Active Status bit (REG[100h] bit) until it returns a 0.
3. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.
4. Continue the program.

Testing the BitBLT Status Before

Testing the BitBLT Active Status before starting a new BitBLT results in better performance, as both CPU and BitBLT engine can be running at the same time. This is most useful for BitBLTs that are self completing (once started they don't require any CPU assistance). While the BitBLT engine is busy, the CPU can do other tasks. To test before each BitBLT operation, perform the following.

1. Wait for the current BitBLT operation to finish -- Poll the BitBLT Active Status bit (REG[100h] bit 7) until it returns a 0.
2. Once the BitBLT operation is finished, read one word from offset 0 in the BitBLT memory area to shutdown the BitBLT engine.
3. Program and start the new BitBLT operation.
4. Continue the program (CPU and BitBLT engine work independently).

However, this approach can pose problems when mixing CPU and BitBLT access to the display buffer. For example, if the CPU writes a pixel while the BitBLT engine is running and the CPU writes a pixel before the BitBLT finishes, the pixel may be overwritten by the BitBLT. To avoid this scenario, always assure no BitBLT is in progress before accessing the display buffer with the CPU, or don't use the CPU to access display buffer at all.

10.4 S1D13806 BitBLT Known Limitations

The S1D13806 BitBLT engine has the following limitations.

- BitBLT Width must be greater than 0.
- BitBLT Height must be greater than 0.
- The BitBLT engine is not SwivelView aware. If BitBLTs are used when SwivelView is enabled, the coordinates and verticies are swapped. It may be possible to recalculate these coordinates and vertices allowing use of some of the BitBLT functions. However the coordinate transformations required may nullify the benefits of the BitBLT.
- The Pattern Fill with ROP (0Ch or 03h) and Transparent Pattern Fill are designed such that the BitBLT Width must be > 1 for 16 bpp color depths and > 2 for 8 bpp.
- One word must be read from the BitBLT area between each BitBLT operation.

11 CRT/TV CONSIDERATIONS

The S1D13806 is capable of driving an LCD panel, CRT display, or a TV monitor. However, only an LCD panel and CRT or an LCD panel and TV can be driven simultaneously. It is not possible to drive both a CRT and TV at the same time.

The horizontal and vertical timing requirements of LCD panels allows for a wide timing variance. In comparison, a CRT display has very strict timing requirements with even a very small timing variance degrading the displayed image. TV monitors require timings based on the NTSC or PAL specifications.

The utility 1386CFG.EXE can be used to generate a header file containing the register values required for CRT/TV or LCD panel timings. For further information on 1386CFG.EXE, see the "*1386CFG Users Manual*," document number X28B-B-001-xx.

11.1 CRT Considerations

CRT timings are based on the VESA Monitor Timing Specifications. The VESA specification details all the parameters of the display and non-display times, as well as the input clock required to meet the times. **Failing to use correct timings can result in an unsynchronized image on a particular monitor, which can permanently damage the monitor.** Virtually all VGA monitors sync if VESA timings are used.

For more information on VESA timings, contact the Video Electronics Standards Association on the internet at www.vesa.org.

11.1.1 Generating CRT timings with 1386CFG

1386CFG.EXE will generate correct VESA timings for 640 × 480 and 800 × 600 if provided the correct VESA input clock. The following timings can be generated:

- 640 × 480 @ 60Hz (Input Clock = 25.175MHz)
- 640 × 480 @ 72Hz (Input Clock = 31.500MHz)
- 640 × 480 @ 75Hz (Input Clock = 31.500MHz)
- 640 × 480 @ 85Hz (Input Clock = 36.000MHz)
- 800 × 600 @ 56Hz (Input Clock = 36.000MHz)
- 800 × 600 @ 60Hz (Input Clock = 40.000MHz)

11.1.2 DAC Output Level Selection

When the CRT is active, the DAC Output Level Select bit (REG[05Bh] bit 3) can be used to double values output to the DAC. This would normally result in very bright colors on the display, but if IREF is reduced at the same time the display will remain at its intended brightness and power consumption is reduced.

11.1.3 Examples

Example 20
Enable the CRT display. Assume the CRT timing registers are already programmed.

1. Confirm the TV PAL/NTSC Output Select bit is clear. REG[05Bh] bit 0 is set to 0.
2. Confirm the CRT and TV displays are disabled. REG[1FCh] bits 2-1 are set to 0.
3. Enable the CRT. REG[1FCh] is set to 1.

Sample code demonstrating how to enable the CRT display is provided in the file 56_CRT.c. This file is available on the internet at www.eea.epson.com.

11.2 TV Considerations

TV timings are based on either the NTSC or PAL specifications. The TV display can be output in either composite video or S-video format.

11.2.1 NTSC Timings

NTSC timings require a 14.318MHz input clock. With the correct input clock the following resolutions are supported.

- 640 × 480
- 696 × 436
- 752 × 484

11.2.2 PAL Timings

PAL timings require a 17.734MHz input clock. With the correct input clock the following resolutions are supported.

- 640 × 480
- 800 × 572
- 856 × 518
- 920 × 572

11.2.3 TV Filters

The S1D13806 is designed with three filters which improve TV picture quality.

- Flicker Filter.
- Chrominance Filter.
- Luminance Filter.

Each filter is independent and can be enabled/disabled separately. The TV picture quality varies depending on the actual picture displayed (static image, moving image, number of colors etc.) and may be improved using the filters.

Flicker Filter

The Flicker Filter is controlled by the Display Mode Select bits (REG[1FCh] bits 2-0). It reduces the "flickering" effect seen on interlaced displays caused by sharp vertical image transitions that occur over one line (e.g. one pixel high lines, edges of window boxes, etc.). The Flicker Filter may be used to for both composite video and S-video formats.

Note: The CRT/TV PCLK 2X Enable bit (REG[018h] bit 7) must be set to 1 when the Flicker Filter is enabled.

Chrominance Filter

The Chrominance Filter is controlled by the TV Chrominance Filter Enable bit (REG[05Bh] bit 5). It adjusts the color of the TV, reducing the "ragged edges" seen a the boundaries between sharp color transitions. The Chrominance Filter may improve the TV picture quality when in composite video format.

Luminance Filter

The Luminance Filter is controlled by the TV Luminance Filter Enable bit (REG[05Bh] bit 4). It adjusts the brightness of the TV, reducing the "rainbow-like" colors at the boundaries between sharp brightness transitions. The Luminance Filter may improve the TV picture quality when in composite video format.

For further information on the TV filters, see the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx.

11.2.4 Examples

Example 21

Enable the TV display and set the Flicker Filter. Assume the TV timing registers are already programmed.

1. Enable the TV with Flicker Filter enabled. REG[1FCh] is set to 06h.
2. Enable the CRT/TV PCLK 2X bit (REG[018h] bit 7). REG[018h] bit 7 is set to 1b.

11.3 Simultaneous Display

The S1D13806 supports simultaneous display of an LCD panel and CRT or an LCD panel and TV. Both display images are completely independent. Each display can show separate areas of the display buffer and display different color depths. There are separate Look-Up Tables and Hardware Cursors/Ink Layers for both the LCD and CRT/TV. If desired, the LUTs for the LCD and CRT/TV may be written to simultaneously (REG[1E0h] bit 0 = 0).

Note: Not all combinations of panel and CRT/TV display resolutions are possible. For further information, see the “*S1D13806 Hardware Functional Specification*,” document number X28B-A-001-xx.

12 MEDIAPLUG

The S1D13806 is designed with support for MediaPlug. MediaPlug is a digital interface supporting the Winnov Videum camera. The Videum camera supports simultaneous video and audio capture of streaming (real-time) and still images. It also supports streaming live video at speeds near 30 frames per second on fast host systems (i.e. Pentium-2 300MHz or faster).

12.1 Programming

MediaPlug and the Winnov Videum camera are a proprietary design of Winnov. Due to the complexity of the digital interface, all software and drivers for the camera are provided by Winnov. Customers intending to use the MediaPlug interface in their design should contact Epson Electronics America to obtain the latest S1D13806 MediaPlug drivers for testing purposes.

The MediaPlug interface on the S1D13806 must be enabled to function correctly. To enable the MediaPlug interface, CONF7 must be high (1) on the rising edge of RESET#. When the MediaPlug interface is enabled, **GPIO12 is controlled by the MediaPlug LCMD register, and the GPIO12 bits in both REG[005h] and REG[009h] have no effect**. Also when the MediaPlug interface is enabled, the camera power (VMPEPWR) is controlled by GPIO12 pin.

The MediaPlug LCMD 16-bit register REG[1000h] contains status bits which can be read by software. For further information on these status bits, see the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx.

The MediaPlug IC Revision bits (REG[1000h] bits 11-8) contain the revision of the interface. The 16-bit value read from REG[1000h] should be masked with 0F00h and compared with 0300h (the current revision of the interface).

The MediaPlug Cable Detected Status bit (REG[1000h] bit 7) determines if a camera is connected to the MediaPlug interface. When this bit returns a 0, a camera is connected. When this bit returns a 1, a camera is not connected.

The MediaPlug Power Enable to Remote bit (REG[1000h], bit 1) controls the power to the remote camera. GPIO12 is controlled by this bit when the MediaPlug interface is enabled. Writing this bit is necessary only when software needs to control the GPIO12 pin.

12.2 Considerations

Software can determine if the MediaPlug interface is enabled or disabled by reading the Config Status Register (REG[00Ch]) and masking the data with 80h. If the masked result equals 80h, the MediaPlug Interface is enabled.

The MediaPlug interface requires a source clock between 8MHz and 19MHz to operate (optimal is 14.318MHz). By default, the MediaPlug software assumes a 14.318MHz frequency is available on CLKI2. If the frequency of CLKI2 is changed, software should reprogram the MediaPlug Clock Register (REG[01Ch]) to select a clock source that is suitable, or program the clock divide bits to obtain a frequency within the correct range.

If the S5U13806B00C evaluation board is used, the clock chip should be programmed to support a valid clock for the MediaPlug interface.

13 IDENTIFYING THE S1D13806

The S1D13806 can only be identified once the Memory/Register Select bit is set to 0. The steps to identify the S1D13806 are:

1. Set the Memory/Register Select bit to 0 by writing 00h to REG[001h].
2. Read REG[000h].
3. The production version of the S1D13806 will return a value of 1Dh (00011101b).
4. The product code is 7 (000111b based on bits 7-2).
5. The revision code is 1 (01b based on bits 1-0).

S1D13806 Series

Embedded DRAM Graphics Controller

Utilities

Table of Contents

List of Figures

List of Tables

1 1386CFG

1386CFG is an interactive Windows® 9x/NT program that calculates the S1D13806 register values for a user-defined LCD/CRT/TV configuration. 1386CFG can edit and set the configurations of the S1D13806 utilities. The configuration information can be saved directly into the utility or into a text header file for use by the software/hardware developer.

1386CFG is designed to work with the S1D13806 utilities, or any programs designed using the Hardware Abstraction Layer (HAL) library.

Note: It is possible to override recommended register settings and select incorrect panel timings using 1386CFG. Seiko Epson does not assume liability for any damage done to the display device as a result of configuration errors.

1.1 S1D13806 Supported Evaluation Platforms

1386CFG runs on a PC system running Windows 9x/NT.

1386CFG can edit the executable files for the following S1D13806 evaluation platforms:

- PC system with an Intel 80 × 86 processor.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- MC68030IDP (Integrated Development Platform) board, revision 3.0, with a Motorola MC68030 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.
- MPC821ADS (Applications Development System) board, revision B, with a Motorola MPC821 processor.

1.2 Installation

Copy the file **1386cfg.exe** to a directory in the path. If running Windows 9x/NT create a shortcut to the file **1386cfg.exe**. Copy the file **panels.cfg** (which contains some common panel types) to the same directory as **1386cfg.exe.**

1.3 Usage

In Windows 9x/NT, double-click the following icon:

Or, at the Windows DOS Prompt, type **`1386cfg`**.

1.4 1386CFG Configuration Tabs

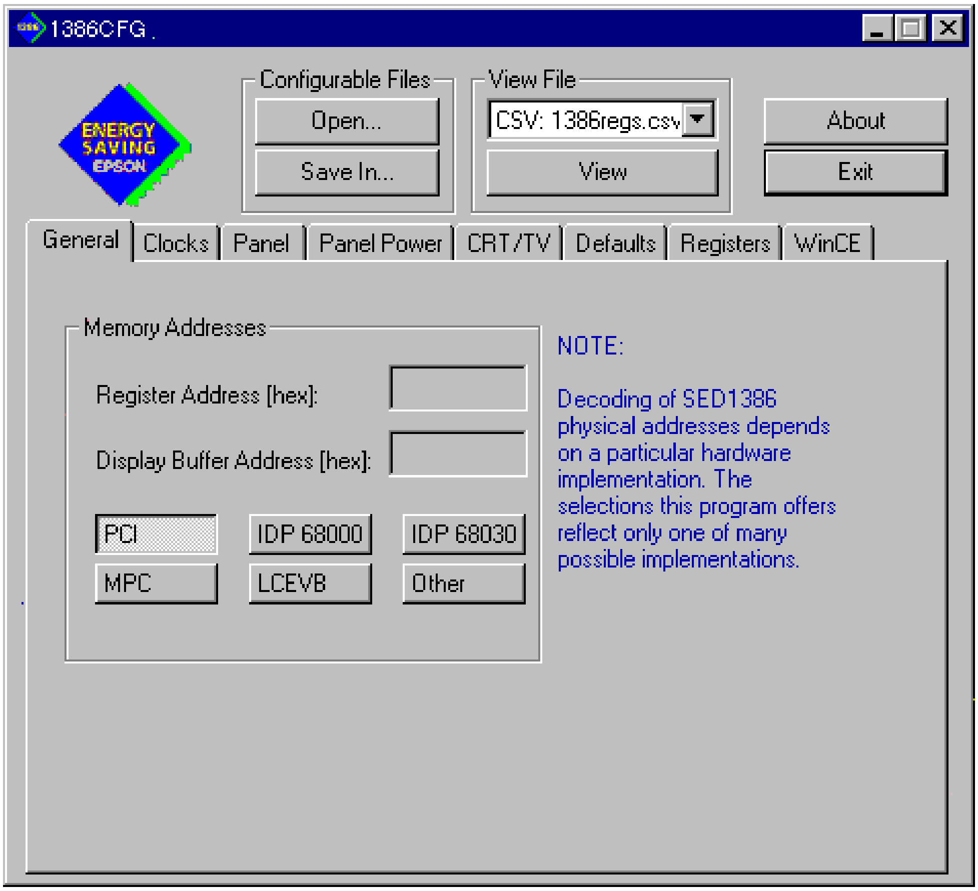
1386CFG provides a series of tabs which can be selected at the top of the main window. The tabs are “General”, “Memory”, “Clocks”, “Panel”, “CRT/TV”, “Defaults”, “Registers”, and “WinCE”. There are also buttons allowing the user to edit and save the configuration of a utility.

The basic procedure for using 1386CFG is as follows:

1. Click on the “Open...” button to load the configuration values from a S1D13806 utility (this step is optional).
2. Edit the configuration values as required for the specific implementation (see each tab description for configuration details).
3. Click on the “Save In...” button option to save the configuration values to the desired utilities, or into an ASCII header file. Each utility must be configured separately.

Note: 1386CFG is designed to work with utilities programmed using a given version of the HAL. If the configuration structure is of a different version, an error message is displayed.

1.4.1 General Tab



NOTE:

Decoding of SED1386 physical addresses depends on a particular hardware implementation. The selections this program offers reflect only one of many possible implementations.

Figure 1-1 General Tab

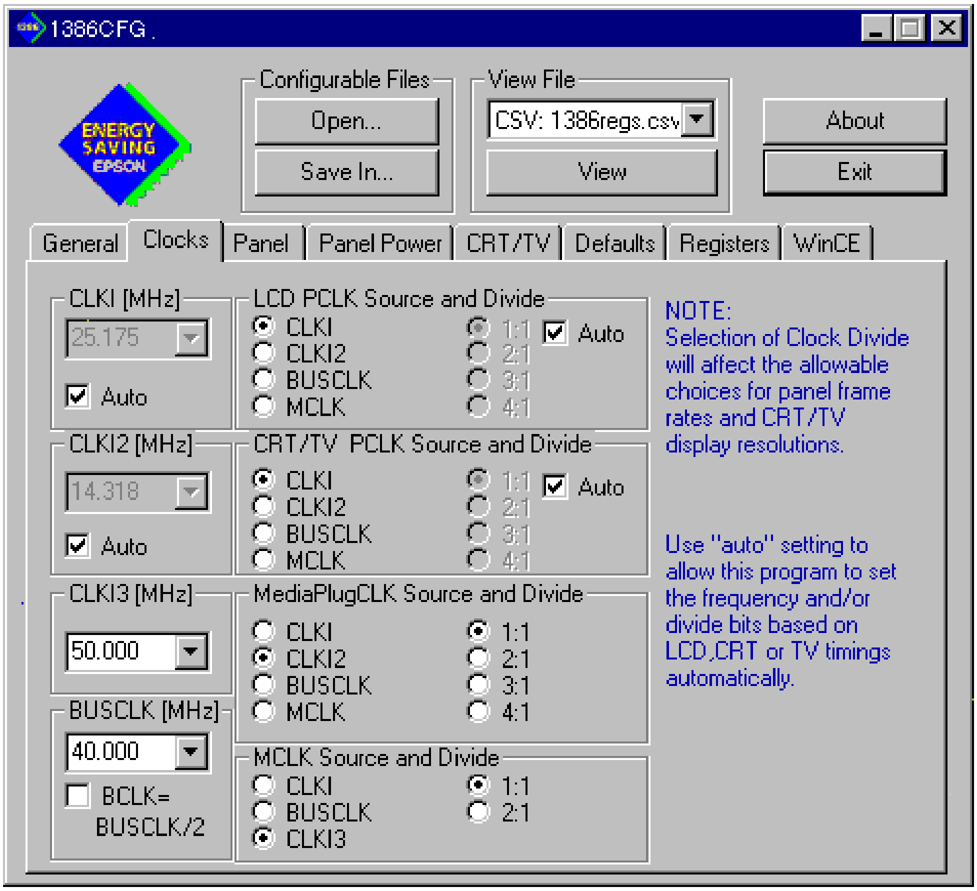
The General Tab selects the following general platform settings.

General Tab	
Register Address	Starting address of the registers (in hexadecimal).
Display Buffer Address	Starting address of the display buffer (in hexadecimal).
Platform Type	Hardware platform to be configured for.

Note: The settings used for PCI, IDP 68000, IDP 68030, MPC, and LCEVB platforms are examples of possible implementations and may not reflect your particular hardware implementation.

Note: PCI addresses are assigned by the system BIOS, and are not an option (grayed out in the example). For further information, see the “*S1D13XX Windows 95/98/NT Device Driver Installation Guide*,” document number X00A-E-003-xx.

1.4.2 Clocks Tab



View File: CSV: 1386regs.csv

CLKI [MHz]: 25.175 ☑ Auto

LCD PCLK Source and Divide: ● CLKI ○ CLKI2 ○ BUSCLK ○ MCLK; ● 1:1 ○ 2:1 ○ 3:1 ○ 4:1 ☑ Auto

CLKI2 [MHz]: 14.318 ☑ Auto

CRT/TV PCLK Source and Divide: ● CLKI ○ CLKI2 ○ BUSCLK ○ MCLK; ● 1:1 ○ 2:1 ○ 3:1 ○ 4:1 ☑ Auto

CLKI3 [MHz]: 50.000

MediaPlugCLK Source and Divide: ○ CLKI ● CLKI2 ○ BUSCLK ○ MCLK; ● 1:1 ○ 2:1 ○ 3:1 ○ 4:1

BUSCLK [MHz]: 40.000 □ BCLK= BUSCLK/2

MCLK Source and Divide: ○ CLKI ○ BUSCLK ● CLKI3; ● 1:1 ○ 2:1

NOTE: Selection of Clock Divide will affect the allowable choices for panel frame rates and CRT/TV display resolutions.

Use "auto" setting to allow this program to set the frequency and/or divide bits based on LCD,CRT or TV timings automatically.

Figure 1-2 Clocks Tab

The Clocks Tab allows manual selection of either the clocks or the required timings. From this information 1386CFG calculates the required timings (if clocks are specified) or the required clocks (if timings are specified). The program calculates the clocks automatically when the “Auto” setting is checked.

If clock selections are manually changed, it may modify or invalidate CRT/TV or panel timings selected previously. Changing the settings on the “Clocks” tab alters the settings on the “Panel” and “CRT/TV” tabs. Confirm all settings on these two tabs after manually changing any clock settings.

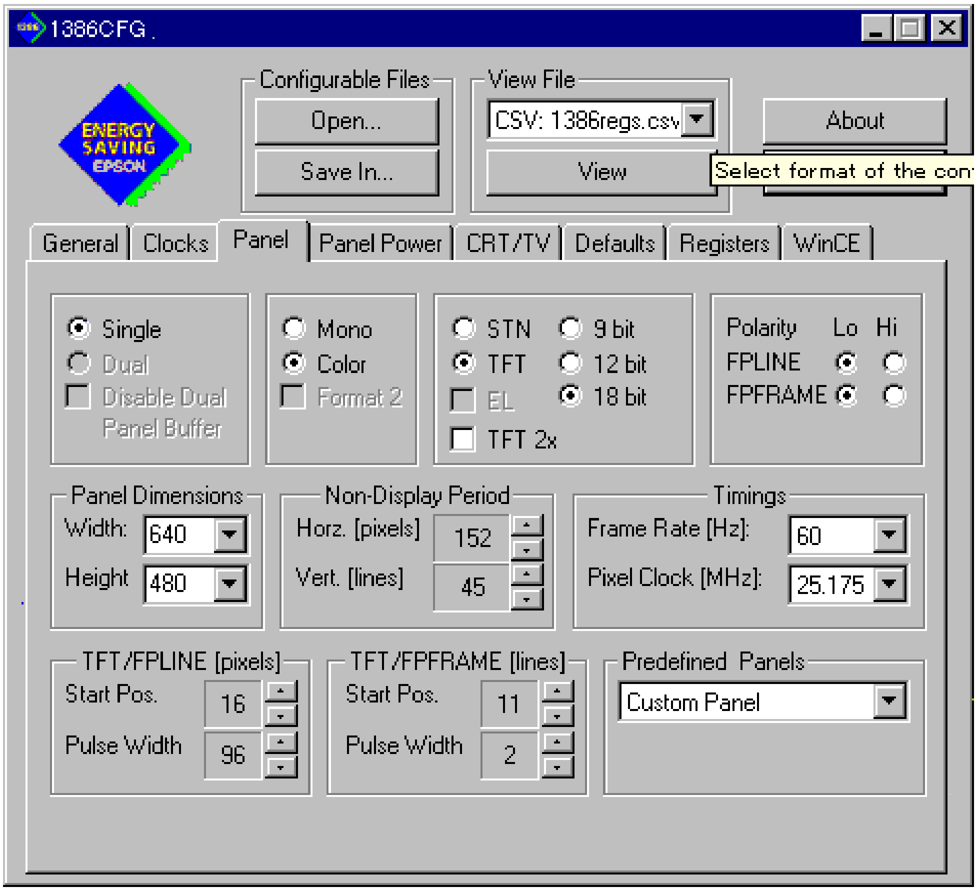
Note: If the same source clock is selected for use by both CRT/TV and LCD panels, the available LCD pixel clock selections are limited due to the more stringent CRT/TV timings.

The Clocks Tab allows the user to select the following settings.

Clocks Tab	
CLKI	Selects the frequency of CLKI.
LCD PCLK Source	Selects the LCD PCLK source.
LCD PCLK Divide	Selects the divide ratio for LCD PCLK. Selecting auto will automatically set the frequency and divide bits.
CLKI2	Selects the frequency of CLKI2.
CRT/TV PCLK Source	Selects the CRT/TV PCLK source.
CRT/TV PCLK Divide	Selects the divide ratio for CRT/TV PCLK. Selecting auto will automatically set the frequency and divide bits.
CLKI3	Selects the frequency of CLKI3.
MediaPlugCLK Source	Selects the source for MediaPlug clock.
MediaPlugCLK Divide	Selects the divide ratio for the MediaPlug clock.
BUSCLK	Selects the frequency of BUSCLK in KHz.
BCLK = BUSCLK/2	Sets BCLK to the BUSCLK source divided by 2.
MCLK Source	Selects the source for MCLK.
MCLK Divide	Selects the divide ratio for MCLK.

Note: Under normal circumstances BCLK = BUSCLK. This option is only set for Toshiba/Philips when DCLKOUT is connected to the S1D13806 BUSCLK signal.

1.4.3 Panel Tab



CSV: 1386regs.csv

● Single
○ Dual
□ Disable Dual Panel Buffer

○ Mono
● Color
□ Format 2

○ STN ○ 9 bit
● TFT ○ 12 bit
□ EL ● 18 bit
□ TFT 2x

Polarity	Lo	Hi
FPLINE	●	○
FPFRAME	●	○

Panel Dimensions
Width: 640
Height 480

Non-Display Period
Horz. [pixels] 152
Vert. [lines] 45

Timings
Frame Rate [Hz]: 60
Pixel Clock [MHz]: 25.175

TFT/FPLINE [pixels]
Start Pos. 16
Pulse Width 96

TFT/FPFRAME [lines]
Start Pos. 11
Pulse Width 2

Predefined Panels
Custom Panel

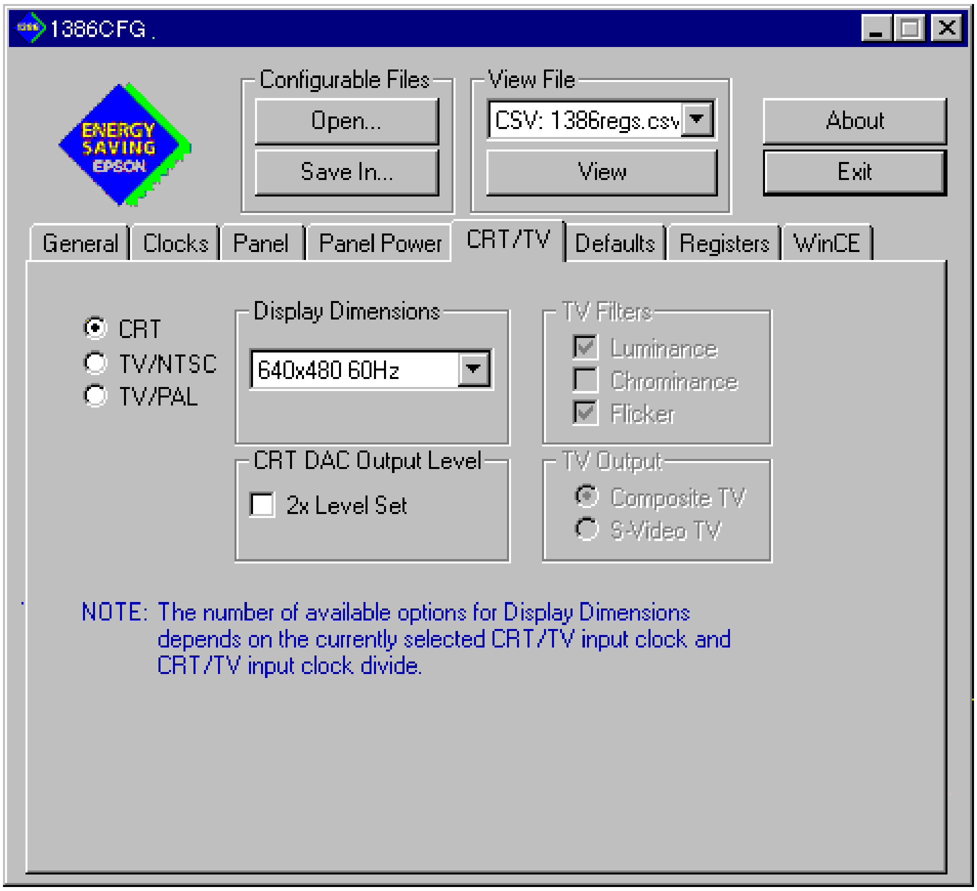
Figure 1-3 Panel Tab

This Panel Tab allows configuration of panel dimensions, type and timings. If the file PANELS.CFG is present in the same directory as 1386CFG.EXE, specific panels can be selected from a list of predefined panels.

The Panel Tab allows selection of the following settings.

Panel Tab	
Single/Dual	Select between a single or dual panel. If no panel exists, select single.
Disable Dual Panel Buffer	The Dual Panel Buffer is used only for dual panels. Disabling the Dual Panel Buffer is not recommended as this will reduce the display quality.
Mono/Color	Selects between a monochrome and color panel. If no panel exists, select color.
Format 2	Selects color passive LCD panel format 2. See the *S1D13806 Hardware Functional Specification*, document number X28B-A-001-xx, for format 1/format 2 descriptions.
STN/TFT	Selects between a passive LCD and TFT/D-TFD panel.
TFT 2x	Enable 2x TFT support.
EL	Enable EL panel support.
Panel Interface	Selects the panel interface width in bits. The bit width values will change when selecting between STN and TFT/D-TFD panels.
FPline Polarity	Selects the polarity of the FPLINE pulse.
FPframe Polarity	Selects the polarity of the FPFRAME pulse.
Dimensions	Selects the width and height of the panel in pixels.
Non-Display Period	Selects the HNDP in pixels and VNDP in lines.
Frame Rate	Selects the desired frame rate in Hz.
Pixel Clock	Selects the desired pixel clock in KHz.
TFT/FPLINE	Selects the start position and pulse width in pixels.
TFT/FPFRAME	Selects the start position and pulse width in lines.
Predefined Panels	Selects from a list of predefined panels.

1.4.4 CRT/TV Tab



● CRT
○ TV/NTSC
○ TV/PAL

Display Dimensions: 640x480 60Hz

CRT DAC Output Level: □ 2x Level Set

NOTE: The number of available options for Display Dimensions depends on the currently selected CRT/TV input clock and CRT/TV input clock divide.

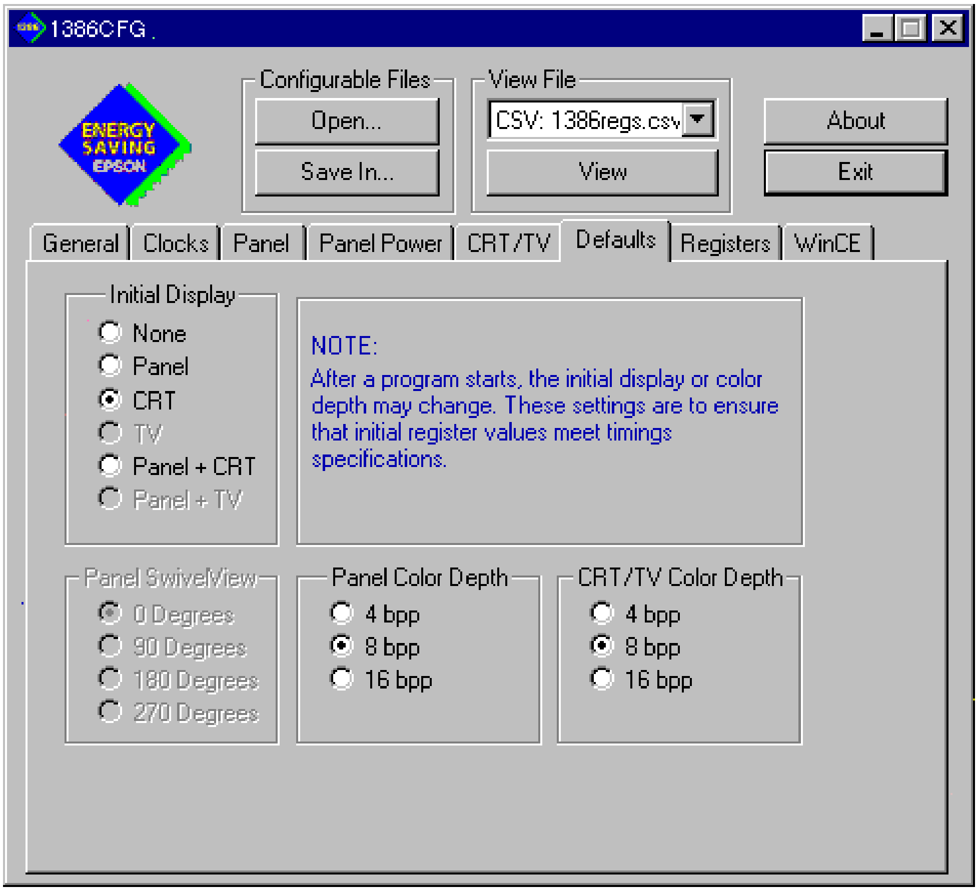
Figure 1-4 CRT/TV Tab

The CRT/TV Tab allows selection of the following settings.

CRT/TV Tab	
CRT/TV Display Selection	Selects the display to configure for.
Display Dimensions	Selects the desired resolution. If there is no selection possible, verify the CRT/TV pixel clock settings.
CRT DAC Output Level	Selects the CRT DAC output level.
TV Filters	Selects which filters will be enabled.
TV Output	Selects the format of the TV output (Composite or S-Video).

Note: For CRT operations, 1386CFG supports VESA timings only. For TV operations, 1386CFG supports NTSC and PAL timings only. Overiding these register values may cause the CRT or TV to display incorrectly.

1.4.5 Defaults Tab



Initial Display

○ None
○ Panel
● CRT
○ TV
○ Panel + CRT
○ Panel + TV

NOTE:
After a program starts, the initial display or color depth may change. These settings are to ensure that initial register values meet timings specifications.

Panel SwivelView

● 0 Degrees
○ 90 Degrees
○ 180 Degrees
○ 270 Degrees

Panel Color Depth

○ 4 bpp
● 8 bpp
○ 16 bpp

CRT/TV Color Depth

○ 4 bpp
● 8 bpp
○ 16 bpp

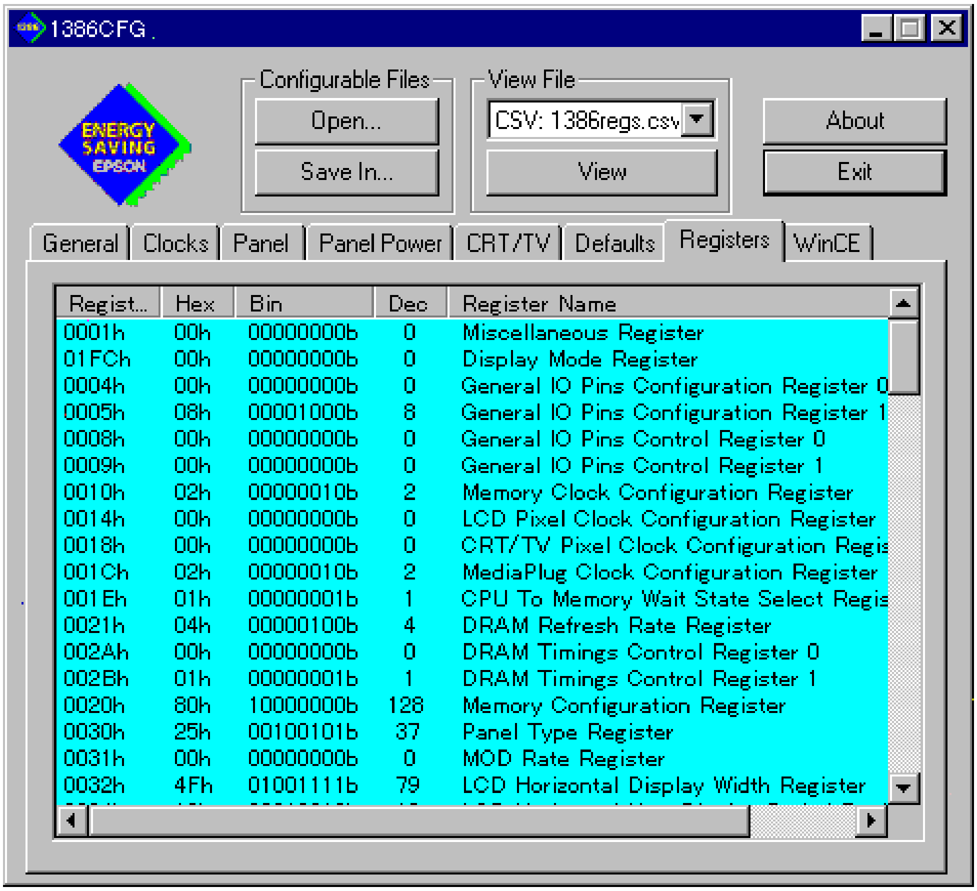
Figure 1-5 Defaults Tab

The Defaults Tab allows selection of the following default settings.

Default Tab	
Initial Display	Sets the default display device.
Panel SwivelView™	Sets the SwivelView™ default setting. This option is only available when “None” or “Panel” is selected.
Panel Color Depth	Sets the initial color depth (4/8/16 bpp) for the LCD panel.
CRT/TV Color Depth	Sets the initial color depth (4/8/16 bpp) for the CRT/TV.

Note: 1386CFG doesn’t check for bandwidth limitations. It is possible to select modes which the S1D13806 doesn’t have enough bandwith to support. For a list of example modes, refer to the “*S1D13806 Hardware Functional Specification*,” document number X28B-A-001-xx.

1.4.6 Registers Tab



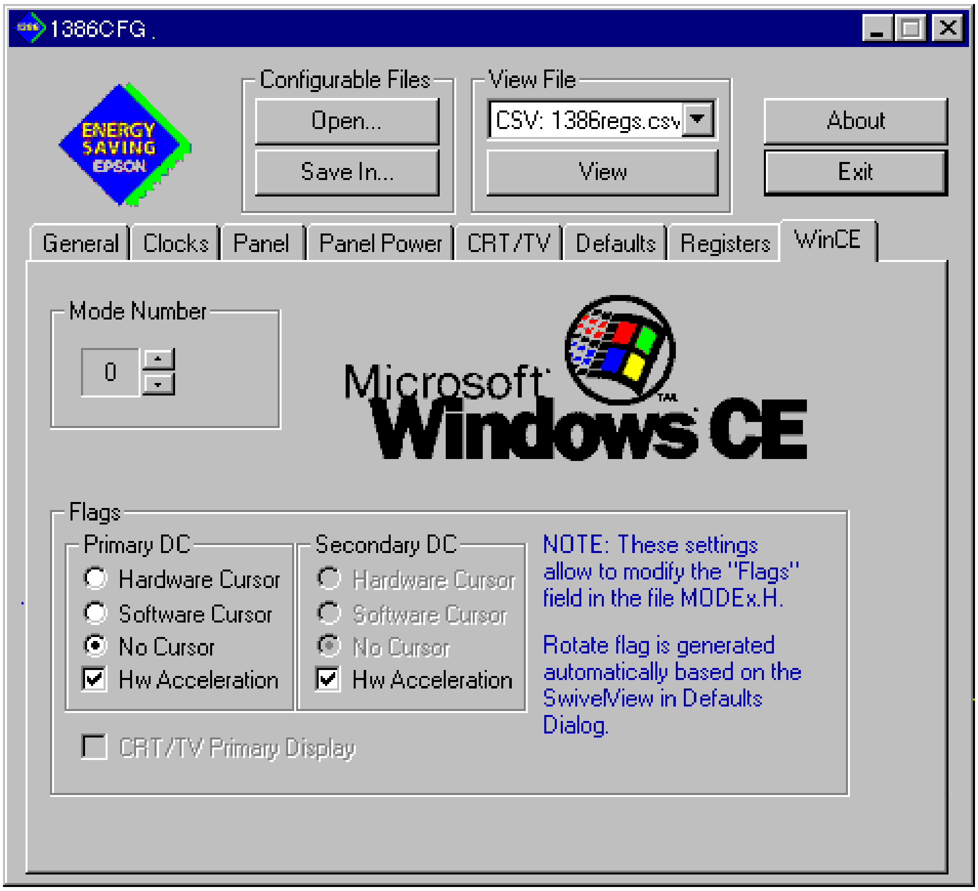
Regist...	Hex	Bin	Dec	Register Name
0001h	00h	00000000b	0	Miscellaneous Register
01FCh	00h	00000000b	0	Display Mode Register
0004h	00h	00000000b	0	General IO Pins Configuration Register 0
0005h	08h	00001000b	8	General IO Pins Configuration Register 1
0008h	00h	00000000b	0	General IO Pins Control Register 0
0009h	00h	00000000b	0	General IO Pins Control Register 1
0010h	02h	00000010b	2	Memory Clock Configuration Register
0014h	00h	00000000b	0	LCD Pixel Clock Configuration Register
0018h	00h	00000000b	0	CRT/TV Pixel Clock Configuration Regis
001Ch	02h	00000010b	2	MediaPlug Clock Configuration Register
001Eh	01h	00000001b	1	CPU To Memory Wait State Select Regis
0021h	04h	00000100b	4	DRAM Refresh Rate Register
002Ah	00h	00000000b	0	DRAM Timings Control Register 0
002Bh	01h	00000001b	1	DRAM Timings Control Register 1
0020h	80h	10000000b	128	Memory Configuration Register
0030h	25h	00100101b	37	Panel Type Register
0031h	00h	00000000b	0	MOD Rate Register
0032h	4Fh	01001111b	79	LCD Horizontal Display Width Register

Figure 1-6 Registers Tab

The Registers Tab lists the register settings that are generated from the chosen configuration. Individual register settings may be changed by clicking on the register listing. **Manual changes to the registers are not checked for errors, so caution is warranted when directly editing these values.** The manually entered values may be changed by 1386CFG if further configuration changes are made on the other tabs. In this case, the user is notified.

Note: Manual changes to the registers may have unpredictable results if incorrect values are entered.

1.4.7 WinCE Tab



Mode Number: 0

Flags

Primary DC
○ Hardware Cursor
○ Software Cursor
● No Cursor
☑ Hw Acceleration

Secondary DC
○ Hardware Cursor
○ Software Cursor
● No Cursor
☑ Hw Acceleration

□ CRT/TV Primary Display

NOTE: These settings allow to modify the "Flags" field in the file MODEx.H.

Rotate flag is generated automatically based on the SwivelView in Defaults Dialog.

Figure 1-7 WinCE Tab

The WinCE Tab generates the header files used to write Windows CE display drivers. Two files are generated: MODE.H and CHIP.H.

WinCE Tab	
Mode Number	Selects the mode number used for the generation of the Windows CE header files.
Cursor	Selects between Hardware Cursor, software cursor and no cursor support.
Hardware Acceleration	Selects whether 2D BitBlt hardware acceleration is used.

1.5 Open File Dialog Box

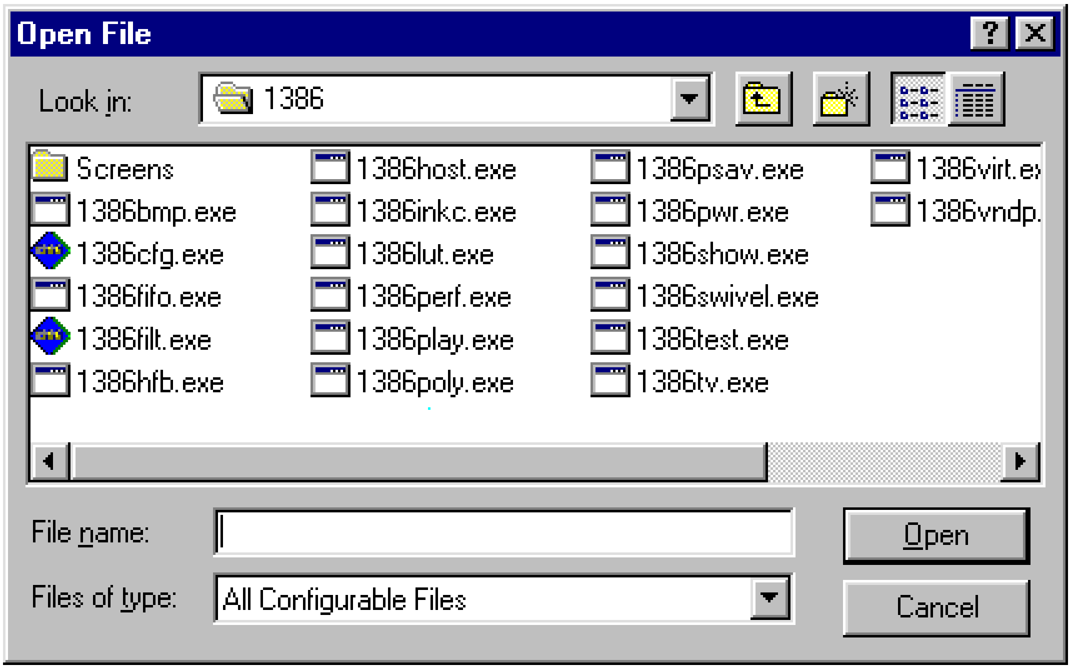
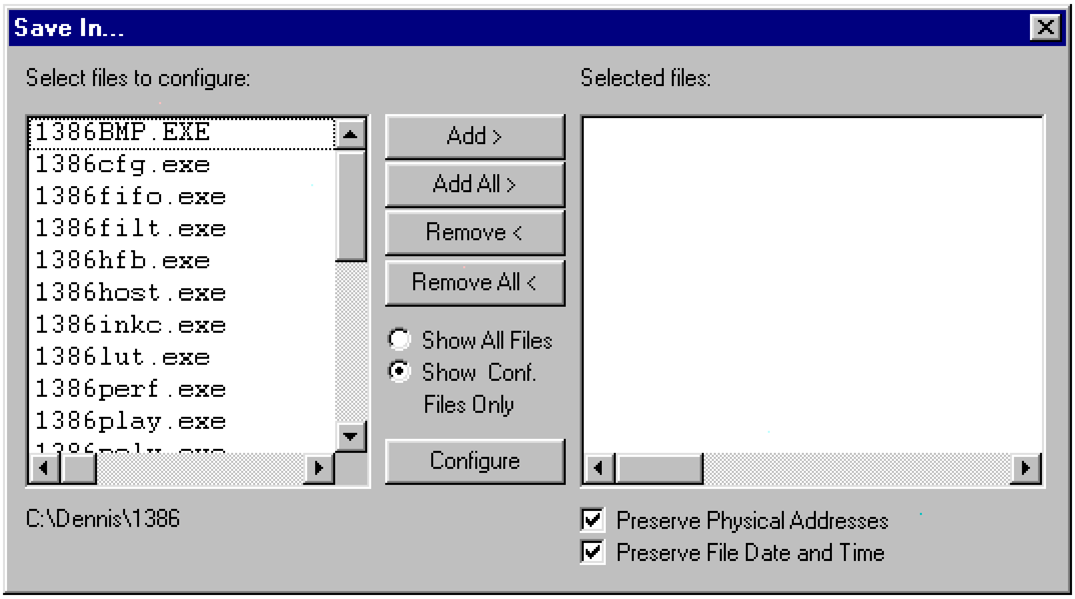


Figure 1-8 Open File Dialog Box

Clicking the “Open...” button displays the Open File Dialog Box. 1386CFG reads the configuration values from a specific .EXE file for Intel platforms, and from a specific S9 or ELF file for non-Intel platforms. The file must have been compiled using a valid version of the 13806 HAL library.

1.6 *Save In Dialog Box*



Save In...

Select files to configure:

1386BMP.EXE
1386cfg.exe
1386fifo.exe
1386filt.exe
1386hfb.exe
1386host.exe
1386inkc.exe
1386lut.exe
1386perf.exe
1386play.exe

Selected files:

C:\Dennis\1386

☑ Preserve Physical Addresses
☑ Preserve File Date and Time

Figure 1-9 Save In Dialog Box

The Save In Dialog Box is shown when the “Save In” button is clicked or a file is dragged onto the 1386CFG window. The left pane lists files available for configuration; the right pane lists the selected files. Files can be selected by clicking the “Add” button, double clicking any file in the left pane, or by using Drag & Drop from Windows Explorer.

The configuration values can be saved to a specific EXE file for Intel platforms, or to a specific S9 or ELF file for non-Intel platforms. The file must have been compiled using a valid version of the 13806 HAL library. The configuration values can also be saved to an ASCII header file (i.e. 13806reg.h) for use by the software/hardware developer. After selecting the files, click the “Configure” button to configure the files.

Note: 1386CFG itself can be configured, allowing changes to the default configuration settings. To configure 1386CFG, copy the existing program into a different directory as it is not possible to configure the file as it is running.

Checking “Preserve Physical Addresses” forces the program to retain the Physical Addresses for the display buffer and registers. This means the addresses specified in the General Tab are ignored. This is useful when configuring several programs for various hardware platforms at the same time. For example, if configuring PCI, MPC and IDP based programs at the same time for a new panel type, the physical addresses for each are retained.

Checking “Preserve File Date and Time” saves the files without changing the date or time stamp of the file.

1.7 Comments

- It is assumed that the user is familiar with the S1D13806 and associated software utilities. For further information on the S1D13806, refer to the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx. For further information on programming the S1D13806, refer to the "*S1D13806 Programming Notes and Examples*," document number X28B-G-003-xx.
- 1386CFG itself can be configured, allowing changes to the default configuration settings. To configure 1386CFG, copy the existing program into a different directory as it is not possible to configure the file as it is running.
- Grayed out options are not available under the current configuration constraints.
- Manual changes of the registers may be changed if further configuration is done, but notification is provided.
- The file PANELS.CFG is a text file containing some supported panels. This file can be edited and is available to 1386CFG if stored in the same directory.
- 1386CFG allows manual entry of values that violate the "*S1D13806 Hardware Functional Specification*" (see document number X28B-A-001-xx) memory and LCD timings. If this is done, unpredictable results may occur.
- To see the current configuration options in condensed form, use the "View File" option and select APPCFG.H.

2 1386SHOW

1386SHOW is designed to demonstrate and test some of the S1D13806 display capabilities. The program can cycle through all color depths and display a pattern showing all available colors or shades of gray. Alternately, the user can specify a color depth and display configuration. 1386SHOW supports SwivelView™ (0°, 90°, 180°, and 270° hardware rotation of the display image).

The 1386SHOW demonstration program must be configured and/or compiled to work with your hardware platform. The utility 1386CFG.EXE can be used to configure 1386SHOW. For further information on 1386CFG, refer to the “*1386CFG Users Manual,*” document number X28B-B-001-xx.

This software is designed to work in both embedded and personal computer (PC) environments. For the embedded environment, it is assumed that the system has a means of downloading software from the PC to the target platform. Typically this is done by serial communications, where the PC uses a terminal program to send control commands and information to the target processor. Alternatively, the PC can program an EPROM, which is then placed in the target platform. Some target platforms can also communicate with the PC via a parallel port connection, or an Ethernet connection.

2.1 S1D13806 Supported Evaluation Platforms

1386SHOW supports the following S1D13806 evaluation platforms:

- PC system with an Intel 80 × 86 processor running Windows® 9x/NT.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- MC68030IDP (Integrated Development Platform) board, revision 3.0, with a Motorola MC68030 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.
- MPC821ADS (Applications Development System) board, revision B, with a Motorola MPC821 processor.

2.2 Installation

PC platform:
Copy the file **1386show.exe** to a directory specified in the path (e.g. PATH=C:\13806).

Embedded platform:
Download the program **1386show** to the system.

2.3 Usage

PC Platform

At the prompt, type:

```
1386SHOW [/a] [bl=n] [bc=n] [ds=n | ds=?] [/g] [/noinit] [/r90 | /r180 | /r270] [/read]
[/s] [/write] [/?]
```

Embedded platform

Execute `1386show` and type the command line argument at the prompt.

Where:

/a — Cycles through all video modes automatically.

bl=n — Shows the LCD display at a user specified color depth (bpp) where n = (4, 8, 16).

bc=n — Shows the CRT/TV display at a user specified color depth (bpp) where n = (4, 8, 16).

ds=n — Selects display surfaces (see Section 2.4, “Display Surfaces” on page 17).

ds=? — Shows the available display surfaces (see Section 2.4, “Display Surfaces” on page 17).

/g — Shows the image overlaid with a 20 pixel wide grid.

/noinit — Skips full register initialization. Only registers used for changing the color depth (bpp) and programming the clock synthesizer are updated. Additionally, some registers are read to determine information such as display size and type (LCD, CRT, TV).

/r90 — Enables SwivelView 90° mode, clockwise hardware rotation of LCD image by 90 degrees.

/r180 — Enables SwivelView 180° mode, clockwise hardware rotation of LCD image by 180 degrees.

/r270 — Enables SwivelView 270° mode, clockwise hardware rotation of LCD image by 270 degrees.

/read	After drawing the image, continually reads from the screen (for testing purposes).
/s	Displays a vertical stripe pattern.
/write	Continually writes to one word of offscreen memory (for testing purposes only).
/?	Displays the help screen.

Note: Pressing the *Esc* key will exit the program.

2.4 Display Surfaces

A surface is a block of memory assigned to one or more physical display devices. 13806SHOW provides seven display surfaces (0-6) which cover the possible combinations of display types. Table 2-1, "Display Surfaces" lists the predefined display surfaces that may be selected.

Table 2-1 Display Surfaces

Display Surfaces (ds =)	Display Device(s) using Memory Block 0	Display Device(s) using Memory Block 1
0	LCD	
1	CRT	
2	TV	
3	LCD & CRT	
4	LCD & TV	
5	LCD	CRT
6	LCD	TV

Display surfaces 0 through 2 display data from a single memory block to an individual display device (LCD, CRT, or TV).

Display surfaces 3 and 4 output to two separate display devices, but generate the output from the same memory block. This may be useful when the same image is to be displayed on both display devices. It also reduces the total amount of display buffer required.

Display surfaces 5 and 6 output to two separate devices from different memory blocks. This allows two independent images to be displayed at the same time. When using display surfaces 5 or 6, some combinations of display modes with a high resolution and/or high color depth may not be supported within a 1.25MB display buffer.

Note: Only Surfaces 5 and 6 support SwivelView as it requires a separate memory block for the LCD. Surfaces 3 and 4 use the same memory block for both displays.

2.5 1386SHOW Examples

1386SHOW is designed to demonstrate and test some of the features of the S1D13806. The following examples show how to use the program in both instances.

2.5.1 Using 1386SHOW For Demonstration

1. To show color patterns which must be manually stepped through, type the following:

   1386SHOW

   The program displays the default color depth and display surface as selected by 1386CFG. Press any key to go to the next screen. Once all screens are shown the program exits. To exit the program immediately press the *Esc* key.

2. To show color patterns which automatically step through, type the following:

   1386SHOW /a

   The program displays the default color depth and display surface as selected by 1386CFG. Each screen is shown for approximately 1 second before the next screen is automatically shown. The program exits after the last screen is shown. To exit the program immediately press *CTRL+BREAK*.

3. To show a color pattern for a specific color depth on the LCD, type the following:

   1386SHOW bl=[mode]

   where:
   mode = 4, 8, or 16

   The program displays the requested color depth for the default display surface and then exits.

Note: If configured for a default display surface including CRT/TV, the color pattern for the default CRT/TV color depth is displayed also.

Note: If a monochrome LCD panel is used, the image is formed using only the green
component of the Look-Up Table for 4 and 8 bpp color depths. For 16 bpp color depths the green component of the pixel value is used.

4. To show a color pattern for a specific color depth on the CRT, type the following:

   1386SHOW bc=[mode]

   where:
   mode = 4, 8, or 16

   The program displays the requested color depth for the default display surface and then exits.

Note: If configured for a default display surface including LCD, the color pattern for the default LCD color depth is displayed also.

5. To show the color patterns in SwivelView 90° mode, type the following:

   1386SHOW /r90

   The program displays the default color depth and display surface as selected by 1386CFG. Press any key to go to the next screen. Since SwivelView 90° is limited to color depths of 8 and 16 bpp the program exits. To exit the program immediately press the *Esc* key.

   The “/r90”, “/r180”, and “/r270” switches can be used in combination with other command line switches.

Note: If configured for a default display surface including CRT/TV, the color pattern for the default CRT/TV color depth is displayed also but non-rotated. Note that display surfaces 3 and 4 do not support SwivelView since the LCD requires a cerebrate memory block.

6. To show solid vertical stripes, type the following:

   1386SHOW /s

   The program displays the default color depth and display surface as selected by 1386CFG. Press any key to go to the next screen. Once all screens are shown the program exits. To exit the program immediately press the *Esc* key.

   The “/s” switch can be used in combination with other command line switches.

2.5.2 Using 1386SHOW For Testing

1. To show a test grid over the 8 bpp color pattern on an LCD, type the following:

   1386SHOW bl=8 /g

   The program displays the 8 bpp color pattern overlaid with a white grid 20 pixels wide and then exits. The grid makes it obvious if the image is shifted or if pixels are missing. **Note the grid is not aligned with the color pattern**, therefore the color boxes will not match the grid boxes.

   The "/g" switch can be used in combination with other command line switches.

Note: If 1386SHOW is configured for a default display surface which includes CRT/TV, the color pattern for the default CRT/TV color depth is displayed as well as the specified LCD color depth.

2. To test background memory reads to the CRT, type the following:

   1386SHOW bc=16 /read

   The program tests screen reads. If a problem exists with memory access, the displayed color pattern appears different than when the "/read" switch is not used. When a problem is detected, check the configuration parameters of 1386SHOW using the utility 1386CFG. For further information on 1386CFG, "*1386CFG Users Manual*," document number X28B-B-001-xx.

   The "/read" switch should be used in combination with the "bl=" or "bc=" setting. Otherwise the test always starts with the default color depth and display surface as selected by 1386CFG. To exit the program after using "/read", press the *Esc* key and wait for a couple of seconds (the keystroke is checked after reading a full screen).

Note: If 1386SHOW is configured for a default display surface which includes LCD, the color pattern for the default LCD color depth is displayed as well as the specified CRT/TV color depth.

2.6 Comments

- If 1386SHOW is started without specifying the color depth (bl= or bc=), the program automatically cycles through the available color depths from highest to lowest. The first color depth shown is the default color depth value saved to 1386SHOW using 1386CFG. This approach avoids showing color depths not supported by a given hardware configuration.
- 1386SHOW checks if the display(s) selected using the DS= option have been previously configured by 1386CFG. If these display(s) have not been configured, 1386SHOW displays an error message. For example, if 1386SHOW is configured for the CRT, an error is displayed if the user selects the TV with the DS= option.
- If the DS= option is used to combine two displays of different resolutions into the same surface, the program will display an error message.
- 1386SHOW cannot show a greater color depth than the display device allows.
- SwivelView 90° and 270° modes (/r90, /r270) are available only for color depths of 8 and 16 bpp.
- SwivelView 180° mode (/r180) is available for color depths of 4, 8, and 16 bpp.

2.7 Program Messages

ERROR: Could not detect S1D13806.
The ID register did not indicate the presence of the 13806.

ERROR: Could not map memory from evaluation board to host platform.
This message should only be shown for DOS platforms. In this case the DOS extender could not be initialized, or was unable to get the linear address of the display buffer.

ERROR: In the given display surface configuration, the user must select the same BPP for both LCD and CRT/TV.
When two displays are using an image from the same display memory block, both displays must be configured for the same color depth (bpp).

ERROR: Invalid display surface number.
The "ds=" command line option included an invalid value. The parameter "ds=?" lists the valid numbers.

WARNING: LCD and CRT resolutions not same but use same display memory block.
LCD: (width, height)
CRT: (width, height)
When the LCD and CRT are displaying an image from the same display memory block, the higher resolution device displays the entire image and the lower resolution device displays a portion of the image using a virtual display.

WARNING: LCD and TV resolutions not same but use same display memory block.
LCD: (width, height)
TV: (width, height)
When the LCD and TV are displaying an image from the same display memory block, the higher resolution device displays the entire image and the lower resolution device displays a portion of the image using a virtual display.

ERROR: LCD must be in landscape mode.
The LCD panel must be configured for SwivelView 0° mode (landscape) when using display surfaces 3 and 4 (both the LCD display and CRT/TV are active using the same memory block).

ERROR: Not enough display buffer memory.
There was insufficient display buffer for the given configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a Dual Panel Buffer is required.
- the number of displays active (LCD **or** LCD and CRT/TV).

WARNING: Cannot use 4 bpp LCD in SwivelView 90° or SwivelView 270° modes.
A color depth of 4 bpp is not supported in SwivelView 90° or SwivelView 270° modes.

ERROR: Do not select 4 bpp LCD in SwivelView 90° or SwivelView 270°.
The “bl=” option selected a color depth not supported with SwivelView enabled.

ERROR: Not enough memory for LCD/CRT/TV in 4/8/16 bits-per-pixel.
1386SHOW is unable to change the color depth due to insufficient display buffer. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a Dual Panel Buffer is required.
- the number of displays active (LCD **or** LCD and CRT/TV).

ERROR: PCI bridge adapter not found.
The Windows PCI driver did not find the PCI Bridge Adapter.

ERROR: PCI driver not found.
The Windows PCI driver is not loaded.

ERROR: Program not configured for LCD/CRT/TV. Run 1386CFG and configure for LCD/CRT/TV.
The program was configured by 1386CFG for a display device that is not available. This typically occurs if the wrong command line was entered for the current configuration.

WARNING: CLKI frequency not in HAL table. Program assumes that external oscillator is used.
WARNING: CLKI2 frequency not in HAL table. Program assumes that external oscillator is used.
The correct frequency was not found in the HAL table used to program the clock synthesizer. An external oscillator may be in use. This warning message will not stop the program.

WARNING: CRT/TV only available in LANDSCAPE mode.
SwivelView is only available on configurations where the LCD uses a separate memory block.

ERROR: bl= and bc= option cannot be used with /noinit.
The command line options “bl=” and “bc=” and /noinit are contradictory, since “bl=” and “bc=” instruct the program to change the color depth and /noinit indicates that no register changes are to be made.

ERROR: Continual screen read will not work with the /a switch.
ERROR: Continual screen write will not work with the /a switch.
The /a switch automatically cycles through the different color depths, whereas the continual screen read/write goes into an infinite loop to read/write memory.

ERROR: Do not select 4 BPP LCD in SwivelView 90 or SwivelView 270 degrees.
SwivelView 90 and SwivelView 270 are available only in 8 and 16 bits-per-pixel modes.

ERROR: Not enough memory for virtual display.
Insufficient memory for the lower resolution display to create a virtual display of the image shown on the higher resolution display.

ERROR: Could not initialize virtual display.
Could not set up virtual image.

3 1386PLAY

1386PLAY is a diagnostic utility allowing a user to read/write to all the S1D13806 Registers, Look-Up Tables and Display Buffer. 1386PLAY is similar to the DOS DEBUG program; commands are received from the standard input device, and output is sent to the standard output device (console for Intel, terminal for embedded platforms). This utility requires the target platform to support standard IO (stdio).

1386PLAY commands can be entered interactively by a user, or be executed from a script file. Scripting is a powerful feature which allows command sequences to be used repeatedly without re-entry.

The 1386PLAY diagnostic utility must be configured and/or compiled to work with your hardware platform. The program 1386CFG.EXE can be used to configure 1386PLAY. For further information on 1386CFG, refer to the "*1386CFG Users Manual*," document number X28B-B-001-xx.

This software is designed to work in both embedded and personal computer (PC) environments. For the embedded environment, it is assumed that the system has a means of downloading software from the PC to the target platform. Typically this is done by serial communications, where the PC uses a terminal program to send control commands and information to the target processor. Alternatively, the PC can program an EPROM, which is then placed in the target platform. Some target platforms can also communicate with the PC via a parallel port connection, or an Ethernet connection.

3.1 S1D13806 Supported Evaluation Platforms

1386PLAY supports the following S1D13806 evaluation platforms:

- PC with an Intel 80 × 86 processor running Windows® 9x/NT.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- MC68030IDP (Integrated Development Platform) board, revision 3.0, with a Motorola MC68030 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.
- MPC821ADS (Applications Development System) board, revision B, with a Motorola MPC821 processor.

3.2 Installation

PC platform

Copy the file **1386play.exe** to a directory in the path (e.g. PATH=C:\S1D13806).

Embedded platform

Download the program **1386play** to the system.

3.3 Usage

PC platform

At the prompt, type:

```
1386play [/?]
```

Where:

/? displays program version information.

Embedded platform

Execute `1386play` and at the prompt, type the command line argument **/?**.

Where:

/? displays program version information.

3.4 Commands

The following commands are designed to be used from within the 1386PLAY program. However, simple commands can also be executed from the command line. If a command with multiple arguments is executed from the command line, it must be enclosed in double quotes (e.g. **1386play "f 0 1FFFFF AB" q**).

Note: If the endian mode of the host platform is big endian, reading/writing words and dwords to/from the registers and display buffer will be incorrect. It is necessary for the user to manually swap the bytes in order to perform the IO correctly.

CLKI [?] iFreq

Selects a preset clock frequency (MHz) for CLKI. If the "?" option is used, the list of available frequencies for CLKI is displayed.

Where:

? Displays a list of available frequencies for CLKI (MHz).

iFreq Sets CLKI to a preset frequency (MHz) specified by iFreq. iFreq is based on the table provided with the command: **CLKI ?**.

CLKI2 [?] iFreq

Selects a preset clock frequency (MHz) for CLKI2. If the "?" option is used, the list of available frequencies for CLKI2 is displayed.

Where:

? Displays a list of available frequencies for CLKI2 (MHz).

iFreq Sets CLKI2 to a preset frequency (MHz) specified by iFreq. iFreq is based on the table provided with the command: **CLKI2 ?**.

CW word

Sends a 24-bit hexadecimal value to the programmable clock. Note that the programmable clock documentation uses the term "word" to describe the 24-bit value. The use of "word" does not imply a 16-bit value in this case.

F addr addr data...

Fills a specified address range with 8-bit data (bytes).

Where:

addr Start and end addresses which define the range to be filled (hex).

data Data to be written (hex). Data can be a list of bytes that will be repeated for the duration of the fill. To use decimal values, attach a "t" suffix to the value. (e.g. 100t is 100 decimal)

FD addr addr data...

Fills a specified address range with 32-bit data (dwords).

Where:

addr	Start and end addresses which define the range to be filled (hex).
data	Data to be written (hex). Data can be a list of dwords that will be repeated for the duration of the fill. To use decimal values, attach a “t” suffix to the value. (e.g. 100t is 100 decimal)

FW addr addr data...

Fills a specified address range with 16-bit data (words).

Where:

addr	Start and end addresses which define the range to be filled (hex).
data	Data to be written (hex). Data can be a list of words that will be repeated for the duration of the fill. To use decimal values, attach a “t” suffix to the value. (e.g. 100t is 100 decimal).

H [lines]

Sets the number of lines of data that will be displayed at a time. The display will be halted after the specified number of lines. Setting the number of lines to 0 will disable the halt function and allow the data to continue displaying until all data has been shown.

Where:

lines	Number of lines that will be shown before halting the displayed data (decimal value).

I [?] [LCD|CRT|TV] [d=iCrtTv] [COMP | SVIDEO] [FLICKER=ON | OFF]

Initializes the S1D13806 registers and enables a a given display type.

Where:

?	Displays a help message.
LCD	Initializes the LCD registers.
CRT	Initializes the CRT registers.
TV	Initializes the TV registers.
d = iCrtTv	Initializes the CRT/TV for a display configuration based on the following table.

Table 3-1 iCrtTv Selection

iCrtTv	Resolution	Display Type
0	640 × 480	CRT
1	800 × 600	CRT
2	752 × 484	TV (NTSC)
3	696 × 436	TV (NTSC)
4	640 × 480	TV (NTSC)
5	920 × 572	TV (PAL)
6	856 × 518	TV (PAL)
7	800 × 572	TV (PAL)
8	640 × 480	TV (PAL)

COMP	Initializes for Composite video output (TV only).
SVIDEO	Initializes for SVideo output (TV only).
FLICKER=ON	Initializes for Flicker Filter enabled (TV only).
FLICKER=OFF	Initializes for Flicker Filter disabled (TV only).

IC {LCD|CRT|TV}

Initializes the Hardware Cursor for a given display type.

Where:

LCD	Initializes for the LCD display.
CRT	Initializes for the CRT display.
TV	Initializes for the TV display.

II {LCD|CRT|TV}

Initializes the Ink Layer for a given display type.

Where:

LCD	Initializes for the LCD display.
CRT	Initializes for the CRT display.
TV	Initializes for the TV display.

L {LCD|CRT|TV} index [red green blue]

Writes red, green, and blue Look-Up Table (LUT) components for a given display type. If the red, green, and blue components are not specified, reads the components at the given index.

Where:

LCD	LUT used by the LCD display.
CRT	LUT used by the CRT display.
TV	LUT used by the TV display.
index	Index into the LUT (hex).
red	Red component of the LUT (hex).
green	Green component of the LUT (hex).
blue	Blue component of the LUT (hex).

Note: Only bits 7-4 of each color are used in the LUT. For example, 10h is the first color intensity after 00h. Valid LUT colors follow the pattern 00h, 10h, 20h, 30h,...E0h, F0h.

LA {LCD|CRT|TV}

Reads all LUT values for a given display.

Where:

LCD	Reads LUT values for LCD display.
CRT	Reads LUT values for CRT display.
TV	Reads LUT values for TV display.

Note: Only bits 7-4 of each color are used in the LUT. For example, 10h is the first color intensity after 00h. Valid LUT colors follow the pattern 00h, 10h, 20h, 30h,...E0h, F0h.

M [?] [LCD|CRT|TV] [bpp]

Sets the color depth (bpp) for the specified display type. If no color depth is provided, information about the current setting on the specified display are listed.

Where:

?	Displays help information for the M and MC commands.
LCD	Sets the color depth of the LCD display.
CRT	Sets the color depth of the CRT display.
TV	Sets the color depth of the TV display.
bpp	Color depth to be set (4/8/16 bpp).

MC [?] [LCD|CRT|TV] [bpp]

Gets extended information and sets the color depth (bpp). If no color depth is specified, further information on the current mode is displayed.

Where:

?	Displays help information for the M and MC commands.
LCD	Sets the color depth of the LCD display.
CRT	Sets the color depth of the CRT display.
TV	Sets the color depth of the TV display.
bpp	Color depth to be set (4/8/16 bpp).

Q

Quits the program.

R addr [count]

Reads a certain number of bytes from the specified address. If no value is provided for count, it defaults to 10h.

Where:

addr	Address from which byte(s) will be read (hex).
count	Number of bytes to be read (hex).

RD addr [count]

Reads a certain number of dwords from the specified address. If no value is provided for count, it defaults to 10h.

Where:

addr	Address from which dword(s) will be read (hex).
count	Number of dwords to be read (hex).

RW addr [count]

Reads a certain number of words from the specified address. If no value is provided for count, it defaults to 10h.

Where:

addr	Address from which word(s) will be read (hex).
count	Number of words to be read (hex).

S {CLKI | CLKI2 | BUSCLK} freq

Sets PCLK source frequency (in kHz).

Where:

CLKI	Sets PCLK source to CLKI.
CLKI2	Sets PCLK source to CLKI2.
BUSCLK	Sets PCLK source to BUSCLK.
freq	Sets the frequency of the PCLK source (decimal value).

V

Calculates the current frame rate for all enabled display devices. The frame rate is calculated from the VNDP count.

W addr data ...

Writes byte(s) of data to specified memory address.

Where:

addr	Address data is written to
data	Data to be written (hex). Data can be a list of bytes that will be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111’b).

WD addr data ...

Writes dword(s) of data to specified memory address.

Where:

addr Address data is written to

data Data to be written (hex). Data can be a list of dwords that will be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111’b).

WW addr data ...

Writes word(s) of data to specified memory address.

Where:

addr Address data is written to

data Data to be written (hex). Data can be a list of words that will be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111’b).

X index [data]

Writes byte data to the register at index. If no data is specified, reads the 8-bit (byte) data from the register at index.

Where:

index Index into the registers (hex).

data Data to be written to/read from register (hex). Data can be a list of bytes that will be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111’b).

XA

Reads all the S1D13806 registers.

XD index [data]

Writes dword data to the register at index. If no data is specified, reads the 32-bit (dword) data from the register at index.

Where:

index Index into the registers (hex).

data Data to be written to/read from register (hex). Data can be a list of dwords that will be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111’b).

XW index [data]

Writes word data to the register at index. If no data is specified, reads the 16-bit (word) data from the register at index.

Where:

index Index into the registers (hex).

data Data to be written to/read from register (hex). Data can be a list of words that will be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111’b).

?

Displays the help screen.

3.5 1386PLAY Example

1. Type **1386PLAY** to start the program.
2. Type **?** for help.
3. Type **i LCD** to initialize the registers.
4. Type **xa** to display the contents of the registers.
5. Type **x 34** to read register 34h.
6. Type **x 34 10** to write 10h to register 34h.
7. Type **f 0 ffff aa** to fill the first FFFFh bytes of the display buffer with AAh.
8. Type **f 0 1fffff aa** to fill 2M bytes of the display buffer with AAh.
9. Type **r 0 100** to read the first 100h bytes of the display buffer.
10. Type **q** to exit the program.

3.6 Scripting

1386PLAY can be driven by a script file. This is useful when:

- there is no display output and a current register status is required.
- various registers must be quickly changed to view results.

A script file is an ASCII text file with one 1386PLAY command per line. All scripts must end with a "q" (quit) command.

On a PC platform, a typical script command line might be: "1386PLAY < dumpregs.scr > results."

This causes the file "dumpregs.scr" to be interpreted as commands by 1386PLAY and the results to be sent to the file "results."

Example 1
Create an ASCII text file that contains the commands `i`, xa, and q.

```
; This file initializes the S1D13806 and reads the registers.

; Note: after a semicolon (;), all characters on a line are ignored.

; Note: all script files must end with the "q" command.

i

xa

q
```

3.7 Comments

- All displayed numeric values are considered to be hexadecimal unless identified otherwise. For example:

  •10 = 10h = 16 decimal.

  •10t = 10 decimal.

  •010'b = 2 decimal.

- Redirecting commands from a script file (PC platform) allows those commands to be executed as if entered by a user.

3.8 Program Messages

ERROR: Could not map memory from evaluation board to host platform.
This message should only be shown for DOS platforms. In this case the DOS extender could not be initialized, or was unable to get the linear address of the display buffer.

ERROR: Not enough display buffer memory.
There was insufficient display buffer for the given configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a Dual Panel Buffer is required.
- the number of displays active (LCD **or** LCD and CRT/TV).

ERROR: Not enough memory for LCD/CRT/TV in 4/8/16 bits-per-pixel.
1386BMP is unable to change the color depth due to insufficient display buffer. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a Dual Panel Buffer is required.
- the number of displays active (LCD **or** LCD and CRT/TV).

ERROR: PCI bridge adapter not found.
The Windows PCI driver did not find the PCI Bridge Adapter.

ERROR: PCI driver not found.
The Windows PCI driver is not loaded.

ERROR: Program not configured for LCD/CRT/TV. Run 1386CFG and configure for LCD/CRT/TV.
The program was configured by 1386CFG for a display device that is not available. This typically occurs if the wrong command line was entered for the current configuration.

WARNING: CLKI frequency not in HAL table. Program assumes that external oscillator is used.
WARNING: CLKI2 frequency not in HAL table. Program assumes that external oscillator is used.
The correct frequency was not found in the HAL table used to program the clock synthesizer. An external oscillator may be in use. This warning message will not stop the program.

ERROR: At least one of the displays must be enabled.
This message is shown when 1386PLAY received the V command, but no display is enabled. At least one display must be enabled for the V command to function (5 seconds of VNDP pulses are counted to calculate the frame rate).

ERROR: Invalid iFreq value.
The CLKI and/or CLKI2 commands were used with an invalid iFreq value. To display a list of iFreq values, type **CLKI ?** or **CLKI2 ?**.

ERROR: Not enough display buffer memory for LCD/CRT/TV cursor/ink layer.
There was insufficient display buffer for the given Hardware Cursor/Ink Layer configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a Dual Panel Buffer is required.
- the number of displays active (LCD **or** LCD and CRT/TV).

WARNING: FEATCLK cannot be multiplexed to CLKI. Clock synthesizer programmed instead.
In 1386PLAY, the CLKI command was used to select the FEATCLK frequency. Since the FEATCLK can only be multiplexed to CLKI2, the clock synthesizer is programmed instead.

4 1386BMP

1386BMP is a demonstration utility used to show the S1D13806 display capabilities by rendering bitmap images on the display device(s). The program will display any bitmap stored in Windows BMP file format and then exit. 1386BMP supports SviwelView™ (0°, 90°, 180°, and 270° hardware rotation of the display image).

1386BMP is designed to operate on a personal computer (PC) within a 32-bit environment only (Windows® 9x/NT). Other embedded platforms are not supported due to the possible lack of system memory or structured file system.

The 1386BMP demonstration utility must be configured and/or compiled to work with your hardware configuration. The program 1386CFG.EXE can be used to configure 1386BMP. For further information on 1386CFG, refer to the "*1386CFG Users Manual*," document number X28B-B-001-xx.

4.1 S1D13806 Supported Evaluation Platforms

1386BMP supports the following S1D13806 evaluation platforms:

- PC with an Intel 80 × 86 processor running Windows 9x/NT.

Note: The 1386BMP source code may be modified by the OEM to support other evaluation platforms.

4.2 Installation

Copy the file **1386bmp.exe** to a directory in the path (e.g. PATH=C:\S1D13806).

4.3 Usage

At the prompt, type:

1386bmp bmpfile1 [bmpfile2] [ds=n | ds=?] [/noinit] [/r90 | /r180 | /r270] [/v] [/?]

Where:

bmpfile1 Specifies filename of the windows format bmp image used for the first display surface (display surface 0).

bmpfile2 Specifies filename of the windows format bmp image used for the second display surface (display surface 1). If bmpfile2 is not specified, bmpfile1 is used for the second display surface also.

ds=n Selects display surfaces (see Section 4.4, “Display Surfaces” on page 38).

ds=? Shows available display surfaces (see Section 4.4, “Display Surfaces” on page 38).

/noinit Skips full register initialization. Only registers used for changing the color depth (bpp) and programming the clock synthesizer are updated. Additionally, some registers are read to determine information such as display size and type (LCD, CRT, TV).

/r90 Enables SwivelView 90° mode, clockwise hardware rotation of LCD image by 90 degrees.

/r180 Enables SwivelView 180° mode, clockwise hardware rotation of LCD image by 180 degrees.

/r270 Enables SwivelView 270° mode, clockwise hardware rotation of LCD image by 270 degrees.

/v Verbose mode (provides information about the displayed image).

/? Displays the help message.

Note: 1386BMP displays the bmpfile image(s) and returns to the prompt.

4.4 Display Surfaces

A surface is a block of memory assigned to one or more physical display devices. 1386BMP provides seven display surfaces (0-6) which cover the possible combinations of display types. Table 2-1, “Display Surfaces” lists the predefined display surfaces that may be selected.

Table 4-1 Display Surfaces

Display Surfaces (ds=)	Display Device(s) using Memory Block 0	Display Device(s) using Memory Block 1
0	LCD	
1	CRT	
2	TV	
3	LCD & CRT	
4	LCD & TV	
5	LCD	CRT
6	LCD	TV

Display surfaces 0 through 2 each display data from a single memory block to an individual display device (LCD, CRT, or TV).

Display surfaces 3 and 4 output to two separate display devices, but generate the output from the same memory block. This may be useful when the same image is to be displayed on both display devices. It also reduces the total amount of display memory required.

Display surfaces 5 and 6 output to two separate devices from different memory blocks. This allows two independent images to be displayed at the same time. When using display surfaces 5 or 6, some combinations of display modes with a high resolution and/or high color depth may not be supported within a 1.25MB display buffer.

Note: Only Surfaces 5 and 6 support SwivelView as it requires a separate memory block for the LCD. Surfaces 3 and 4 use the same memory block for both displays.

4.5 1386BMP Examples

To display a bmp image on an LCD, type the following:
1386bmp bmpfile1.bmp ds=0

To display a bmp image on a CRT, type the following:
1386bmp bmpfile1.bmp ds=1

To display a bmp image on an LCD with 90° SwivelView™ enabled, type the following:
1386bmp bmpfile1.bmp ds=0 /r90

To display the same bmp image on both the LCD and CRT, type the following:
1386bmp bmpfile1.bmp ds=3

To display the same bmp image independently on the LCD and TV, type the following:
1386bmp bmpfile1.bmp ds=6

To display different bmp images independently on the LCD and CRT, type the following:
1386 bmpfile1.bmp bmpfile2.bmp ds=5

4.6 Comments

- 1386BMP displays only Windows BMP format images.
- A 24-bit true color bitmap is displayed at a color depth of 16 bit-per-pixel.
- Only the green component of the image is seen on a monochrome panel.
- When display devices of different resolutions are used, the image on the smaller display is displayed using a virtual display. Therefore, only a portion of the image is viewable. To show a complete image on the smaller display, specify two separate bmpfiles with resolutions matching the intended display device.

4.7 Program Messages

ERROR: Could not detect S1D13806.
The ID register did not indicate the presence of the S1D13806.

ERROR: Could not map memory from evaluation board to host platform.
This message should only be shown for DOS platforms. In this case the DOS extender could not be initialized, or was unable to get the linear address of the display buffer.

ERROR: Failed to open BMP file: 'filename'
Could not open the BMP file.

ERROR: 'filename' is not a valid bitmap file.
The filename given on the command line is not a valid BMP file.

ERROR: In the given display surface configuration, the user must select the same BPP for both LCD and CRT/TV.
When two displays are using an image from the same display memory block, both displays must be configured for the same color depth (bpp).

ERROR: Invalid display surface number.
The "ds=" command line option included an invalid value. The parameter "ds=?" lists the valid numbers.

ERROR: LCD and CRT resolutions must be identical.
LCD: (width, height)
CRT: (width, height)
When the LCD and CRT are using an image from the same display memory block, both displays must be the same resolution.

ERROR: LCD and TV resolutions must be identical.
LCD: (width, height)
TV: (width, height)
When the LCD and TV are using an image from the same display memory block, both displays must be the same resolution.

ERROR: LCD must be in landscape mode.
The LCD panel must be configured for SwivelView 0° mode (landscape) if both the LCD display and CRT/TV are active.

ERROR: Not enough display buffer memory.
There was insufficient display buffer for the given configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a Dual Panel Buffer is required.
- the number of displays active (LCD **or** LCD and CRT/TV).

ERROR: Not enough memory for LCD/CRT/TV in 4/8/16 bit-per-pixel.
1386BMP is unable to change the color depth due to insufficient display buffer. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a Dual Panel Buffer is required.
- the number of displays active (LCD **or** LCD and CRT/TV).

ERROR: Not enough memory for virtual display.
A virtual display is required for SwivelView. This error message indicates there is insufficient display buffer for the given configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a Dual Panel Buffer is required.
- the number of displays active (LCD **or** LCD and CRT/TV).

ERROR: PCI bridge adapter not found.
The Windows PCI driver did not find the PCI Bridge Adapter.

ERROR: PCI driver not found.
The Windows PCI driver is not loaded.

ERROR: Program not configured for LCD/CRT/TV. Run 1386CFG and configure for LCD/CRT/TV.
The program was configured by 1386CFG for a display device that is not available. This typically occurs if the wrong command line was entered for the current configuration.

WARNING: CLKI frequency not in HAL table. Program assumes that external oscillator is used.
WARNING: CLKI2 frequency not in HAL table. Program assumes that external oscillator is used.
The correct frequency was not found in the HAL table used to program the clock synthesizer. An external oscillator may be in use. This warning message will not stop the program.

WARNING: CRT/TV only available in LANDSCAPE mode.
SwivelView is only available on LCD only configurations.

5 1386SWIVEL

1386SWIVEL is a command line utility that demonstrates the SwivelView™ feature of the S1D13806. 1386SWIVEL provides hardware rotation of a predefined image by 0°, 90°, 180°, and 270° in a clockwise direction. 1386SWIVEL cycles through each SwivelView mode, advancing to the next mode when a key is pressed.

1386SWIVEL is designed to operate on a personal computer (PC) within a 32-bit environment only (Windows® 9x/NT). Other embedded platforms are not supported.

The 1386SWIVEL utility must be configured and/or compiled to work with your hardware configuration. The utility 1386CFG.EXE can be used to configure 1386SWIVEL. For further information on 1386CFG, refer to the "*1386CFG Users Manual*," document number X28B-B-001-xx.

5.1 S1D13806 Supported Evaluation Platforms

1386SWIVEL supports the following S1D13806 evaluation platforms.

- PC with an Intel 80 × 86 processor running Windows 9x/NT.

5.2 Installation

Copy the file **1386swivel.exe** to a directory in the path. If desired, create a shortcut on the Windows 9x/NT desktop to the file **1386swivel.exe**.

5.3 Usage

At the Windows DOS Prompt, type:

```
1386swivel
```

Note: Pressing the *ESC* key exits the program.

5.4 Example

1. Run the utility 1386SWIVEL. At the Windows DOS prompt type:

   1386swivel

Note: 1386SWIVEL displays colored lines of text and initially appears in SwivelView 0° mode (normal landscape).

2. Press any key to enable SwivelView 90° mode. The pattern is rotated by 90°.
3. Press any key to enable SwivelView 180° mode. The pattern is rotated by 180°.
4. Press any key to enable SwivelView 270° mode. The pattern is rotated by 270°.
5. Press any key to return to SwivelView 0° mode (landscape).
6. Press the *ESC* key to exit the program.

Note: 1386SWIVEL will continue to cycle through the SwivelView modes in the above order until the *ESC* key is pressed.

5.5 Comments

- 1386SWIVEL supports LCD panels only (no CRT or TV).
- 1386SWIVEL must be configured for LCD only using the utility 1386CFG. For further information on 1386CFG, refer to the "*1386CFG Users Manual*," document number X28B-B-001-xx.
- 1386SWIVEL supports 8 and 16 bpp color depths only.
- For further information on SwivelView™, refer to the "*S1D13806 Hardware Functional Specification*" (document number X28B-A-001-xx) and the "*S1D13806 Programming Notes and Examples*" (document number X28B-G-003-xx).

6 1386FILT

1386FILT is an interactive Windows® 9x/NT program that enables/disables the S1D13806 TV Filters. It demonstrates the effect each filter has on a pre-loaded TV image. 1386FILT is particularly useful in a test or demonstration environment where 1386FILT is run on one display and the effects of enabling/disabling each filter are seen on a TV. An OEM may find this useful in determining the appropriate filters for their application.

Note: The 1386FILT dialog box doesn’t appear on any display device controlled by the S1D13806. The 1386FILT dialog box appears on the Windows 9x/NT desktop.

1386FILT is designed to operate on a personal computer (PC) within a 32-bit environment only (Windows 9x/NT). Other embedded platforms are not supported.

6.1 S1D13806 Supported Evaluation Platforms

1386FILT supports the following S1D13806 evaluation platforms:

- PC with an Intel 80 × 86 processor running Windows 9x/NT.

6.2 Installation

Copy the file **1386filt.exe** to a directory in the path. If desired, create a shortcut on the Windows 9x/NT desktop to the file **1386filt.exe**.

6.3 Usage

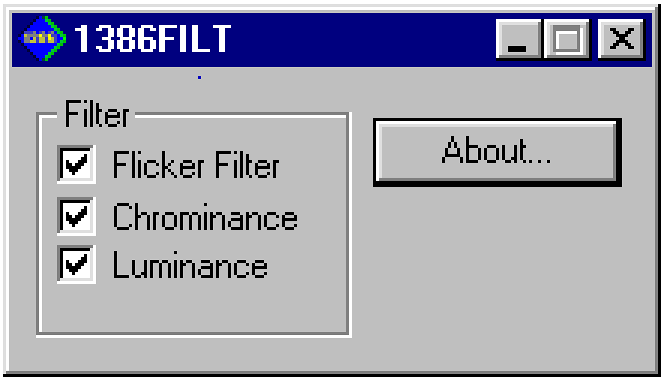
In Windows 9x/NT, double-click the following icon:

Or, at the Windows DOS Prompt, type **`1386filt`**.

6.4 Filter Dialog Box

The filter dialog box controls which TV filters are enabled/disabled during NTSC or PAL output. The check box for each filter determines if the filter is enabled or disabled. When the box is checked the filter is enabled. When the box is unchecked the filter is disabled. In the example below:

- the flicker filter is enabled.
- the chrominance filter is enabled.
- the luminance filter is enabled.



☑ Flicker Filter
☑ Chrominance
☑ Luminance

Figure 6-1 Filter Dialog Box

6.5 Filter Descriptions

When displaying computer images on a TV, several image distortions are likely to arise.

- flickering.
- cross-chrominance distortion.
- cross-luminance distortion.

These distortions are caused by the high-resolution nature of computer images which typically contain sharp chrominance (color) transitions, and sharp luminance (brightness) transitions. Three filters are available to reduce these distortions.

6.5.1 Anti-flicker Filter

The “flickering” effect seen on interlaced displays is caused by sharp vertical image transitions that occur over one line (1 vertical pixel). For example, flickering may occur where there are one pixel high lines, edges of window boxes, etc. Flickering occurs because these high resolution lines are effectively displayed at half the refresh frequency due to interlacing. To reduce flickering, the anti-flicker filter averages adjacent lines on the TV display.

6.5.2 Chrominance Filter

The chrominance filter adjusts the color of the TV by limiting the bandwidth of the chrominance signal (reducing cross-luminance distortion). This reduces the “ragged edges” seen at boundaries between sharp color transitions. This filter is intended for use with composite video output.

6.5.3 Luminance Filter

The luminance filter adjusts the brightness of the TV by limiting the bandwidth of the luminance signal (reducing cross-chrominance distortion). This reduces the “rainbow-like” colors at boundaries between sharp luminance transitions. This filter is intended for use with composite video output.

6.6 Comments

- The Flicker Filter can’t be enabled unless a TV is present and active.
- 1386FILT is designed to show the effects of the filters on a pre-loaded TV image. 1386BMP may be used to display a static image on the TV (see the “*1386BMP Users Manual*,” document number X28B-B-004-xx).
- The chrominance and luminance filters are intended for use with composite output.
- For information on manually enabling/disabling the TV filters, refer to the “*S1D13806 Hardware Functional Specification*” (document number X28B-A-001-xx) and the “*S1D13806 Programming Notes and Examples*” (document number X28B-G-003-xx).

THIS PAGE IS BLANK.

S1D13806 Series

Embedded DRAM Graphics Controller

S5U13806B00C
Rev. 1.0
Evaluation Board
User's Manual

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

This manual describes the setup and operation of the S5U13806B00C Evaluation Board. The S5U13806B00C is designed as an evaluation platform for the S1D13806 Embedded Memory Display Controller chip.

This user manual will be updated as appropriate. Please check the latest revision of this document before beginning any development.

2 FEATURES

The S5U13806B00C features the following:

- S1D13806 Embedded Memory Display Controller chip.
- Headers for connecting to host bus interface.
- 1280K bytes embedded SDRAM.
- Configuration options.
- Adjustable positive LCD bias power supply from +24V to +40V.
- Adjustable negative LCD bias power supply from -24V to -14V.
- 4/8-bit 3.3V or 5V monochrome passive LCD panel support.
- 4/8/16-bit 3.3V or 5V color passive LCD panel support.
- 9/12/18/2x9/2x12-bit 3.3V or 5V TFT/D-TFD LCD panel support.
- Embedded DAC for CRT and TV support.
- WINNOV VideumCam^{TM} digital camera support at 320 × 240 × 256 color at up to 30fps with on-chip MediaPlug interface.
- Programmable Clock synthesizer for maximum clock flexibility (CLKI and CLKI2).
- Software initiated Power Save Mode.

3 INSTALLATION AND CONFIGURATION

The S5U13806B00C is designed to support as many platforms as possible. The S5U13806B00C incorporates a DIP switch and three jumpers which allow both evaluation board and S1D13806 LCD controller to be configured for a specified evaluation platform.

3.1 Configuration DIP Switches

The S1D13806 has configuration inputs (CONF[7:0]) which are read on the rising edge of RESET#. In order to configure the S1D13806 for multiple host bus interfaces an eight-position DIP switch (S1) is required. The following figure shows the location of DIP switch S1 on the S5U13806B00C.

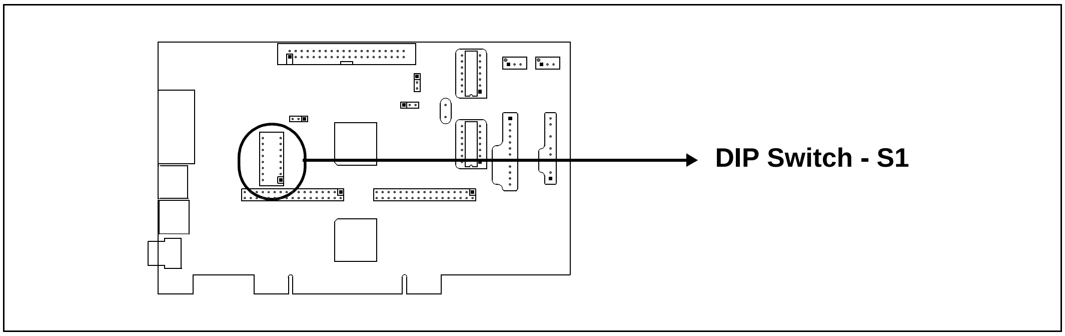


Figure 3-1 Configuration DIP Switch (S1) Location

The following DIP switch settings configure the S1D13806.

Table 3-1 Configuration DIP Switch Settings

Switch	S1D13806 Signal	PCI Bridge Adapter FPGA Signal	Closed/On = 1	Open/Off = 0
S1-[4:1]	CONF[3:0]	—	0000 = Generic; Little Endian; Active Low WAIT#	
			0001 = Generic; Little Endian; Active High WAIT#	
			0010 = Generic; Big Endian; Active Low WAIT#	
			0011 = Generic; Big Endian; Active High WAIT#	
			0100 = MIPS/ISA; Little Endian; Active Low WAIT#	
			0101 = MIPS/ISA; Little Endian; Active High WAIT#	
			0110 = MC68000; Big Endian; Active High WAIT#	
			0111 = MC68030; Big Endian; Active High WAIT#	
			1000 = PR31500/Tx39xx; Little Endian; Active Low WAIT#	
			1001 = PCMCIA; Little Endian; Active Low WAIT#	
			1010 = Reserved	
			1011 = MPC821; Big Endian; Active High WAIT#	
			1100 = SH3; Little Endian; Active Low WAIT#	
			1101 = SH4; Little Endian; Active High WAIT#	
			1110 = SH3; Big Endian; Active Low WAIT#	
			1111 = SH4; Big Endian; Active High WAIT#	
S1-5	CONF5	—	BUSCLK input divided by 2	BUSCLK input not divided
S1-6	CONF6	—	WAIT# always driven	WAIT tristated when the chip is not accessed by the host
S1-7	CONF7	—	Configures GPIO12 for use as MediaPlug output pin VMPEPWR and enables MediaPlug Register access	Configured GPIO12 for normal use and disables MediaPlug register access
S1-8	—	nCONFIG	Disable FPGA for non-PCI host	Enable FPGA for PCI host

▨ = Required configuration when used in a PCI environment

3.2 Configuration Jumpers

The S5U13806B00C has three jumper blocks which configure various board settings. The jumper positions for each function are shown below.

Table 3-2 Jumper Summary

Jumper	Function	Position 1-2	Position 2-3
JP1	I_{REF} for DAC	4.6mA	9.2mA
JP2	LCD Panel Voltage	5.0V	3.3V
JP3	GPIO11 Polarity on H3	Normal (Active High)	Inverted (Active Low)

▨ = Default setting

JP1 - I_{REF} for DAC

JP1 sets the internal reference current (I_{REF}) used by the embedded DAC on the S1D13806.

Position 2-3 is used for CRT and TV displays. (default setting)

Position 1-2 is used for CRT mode only and must be set in conjunction with the DAC Output Level Select bit (REG[05Bh] bit 3). Choosing position 1-2 and setting REG[05Bh] bit 3 to 1, lowers IREF which reduces DAC power consumption. Refer to the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx for details.

Note: For LCD only, JP1 should be left at the default setting (position 2-3). IREF is not required for LCD displays.

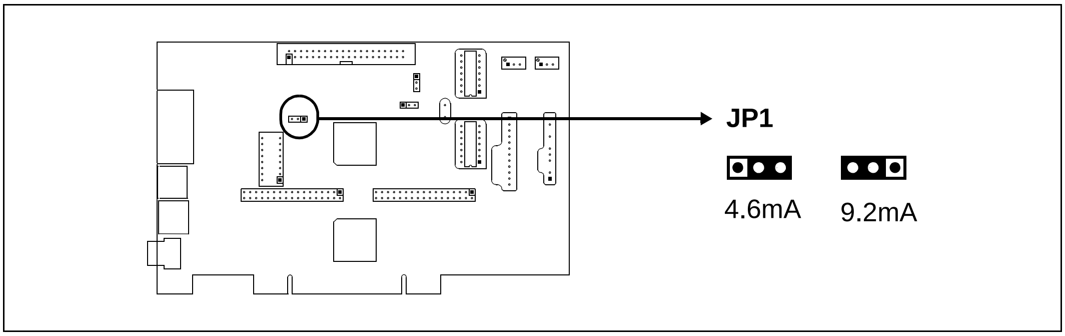


Figure 3-2 Configuration Jumper (JP1) Location

JP2 - LCD Panel Voltage

JP2 sets the voltage level to the LCD panel.
Position 1-2 sets the voltage level to 5.0 volts.
Position 2-3 sets the voltage level to 3.3 volts.

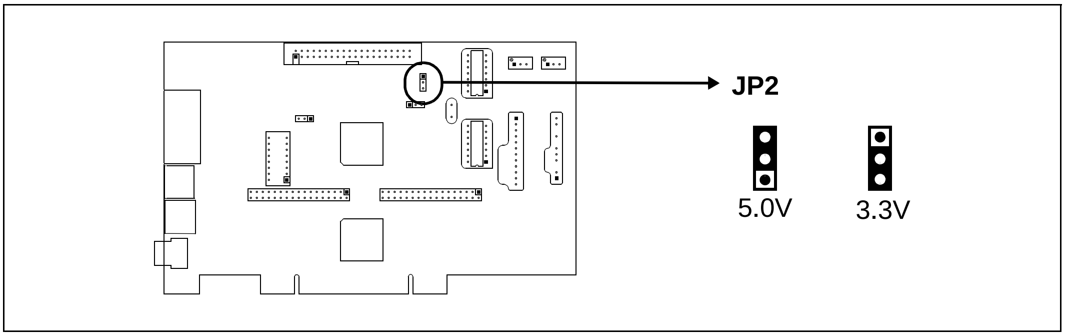


Figure 3-3 Configuration Jumper (JP2) Location

JP3 - GPIO11 Polarity on H3

By default, the S5U13806B00C uses GPIO11 to control the on-board LCD bias power supplies. If the LCD panel requires GPIO11 for other purposes, it is supplied on the LCD connector H3. JP3 controls the polarity of GPIO11 **going to LCD connector H3. This jumper has no effect on the polarity of GPIO11 controlling the on-board LCD bias power supplies.**
Position 1-2 sets the polarity of GPIO11 to active high.
Position 2-3 sets the polarity of GPIO11 to active low. (default setting)

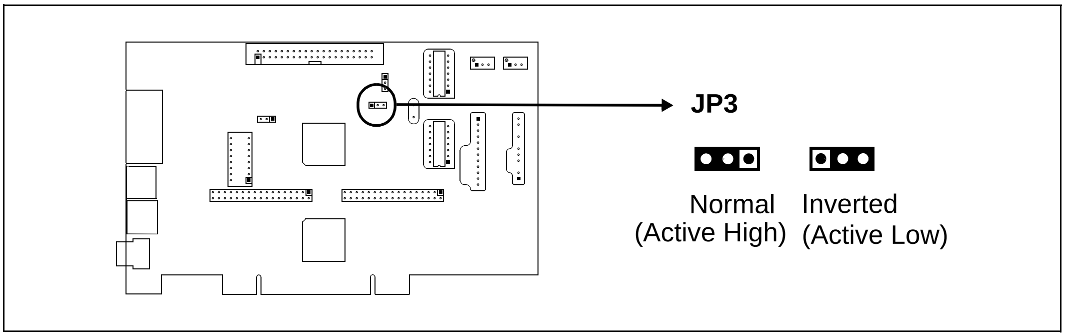


Figure 3-4 Configuration Jumper (JP3) Location

For further information on the LCD bias power supplies, refer to Section 5.5, “LCD Power Sequencing” on page 14 for details.

4 CPU HOST INTERFACE

The S5U13806B00C operates with both PCI and non-PCI evaluation platforms. It supports display types such as, passive LCD panels (4/8/16-bit), TFT/D-TFD panels (9/12/18/2x9/2x12), CRT and TV (NTSC and PAL). Additionally, it supports a variety of clock options.

4.1 PCI Bus Support

The S1D13806 LCD controller does not support the PCI bus directly. However, for ease of evaluation, the S5U13806B00C is designed using the PCI Bridge Adapter FPGA. This allows the S5U13806B00C to run on any PC with a PCI slot. The FPGA translates PCI accesses into PC Card accesses which are decoded by the S1D13806.

As a PCI device, the S5U13806B00C has the following characteristics:

- 33MHz bus clock.
- Target with no interrupts.
- Non-cacheable memory read and write.
- 3.3V or 5V PCI signalling.

A 4M byte PCI address range is allocated to the S5U13806B00C by the system BIOS. The S1D13806 uses this address range to map the internal registers and the 1.25M byte display buffer. The following table shows the memory mapping of the PCI address block.

Table 4-1 S1D13806 Memory Mapping onto 4M byte PCI Address Block

PCI Memory Offset	Description	S1D13806 M/R#	S1D13806 AB[20:0]
00 0000h to 00 01FFh	General registers (512 byte)	0	00 0000h to 00 01FFh
00 1000h to 00 1FFFh	MediaPlug registers (4K byte)	0	00 1000h to 00 1FFFh
10 0000h to 1F FFFFh	BitBlt data registers (1M byte)	0	10 0000h to 1F FFFFh
20 0000h to 33 FFFFh	Display Buffer (1.25M byte)	1	00 0000h to 13 FFFFh

4.1.1 On-Board PCI Configuration Registers

Read-Only Registers

The PCI Bridge Adapter FPGA provides configuration registers which contain identification information required by the PCI interface. The following values are hard-wired into these registers.

Table 4-2 PCI Configuration Register Read Values

Name	Address	Register size	Value
Vendor ID	0h	16 bits	10F4h
Device ID	2h	16 bits	1300h
Status	6h	16 bits	400h
Revision ID	8h	8 bits	1
Class Code	9h	24 bits	FF 0000h
Subsystem Vendor ID	2Ch	16 bits	10F4h
Subsystem ID	2Dh	16 bits	8000h
Header Type	Eh	8 bits	0
n/a	Fh-FFh	32 bits	0

Read/Write Registers

The PCI Bridge Adapter FPGA provides two read/write registers which are used for access enabling and memory mapping as follows.

Table 4-3 PCI Configuration Register Write Values

Name	Address	Register size	Valid bits	Meaning
Command	4h	16 bits	Bit 1 only; other bits are zero.	Access enabled if high
Base Address	10h	32 bits	Bits 31 to 22; other bits are zero.	Position of 4M byte reserved window

4.1.2 Utility Software

All utility software for the S5U13806B00C evaluation board is fully PCI compliant and handles the PCI configuration registers automatically.

4.2 *Non-PCI Host Interface Support*

The S5U13806B00C is specifically designed to support a standard PCI bus environment (using the PCI Bridge Adapter FPGA). However, the S5U13806B00C directly supports many other Host Bus Interfaces. When the FPGA is disabled (using DIP switch S1-8), headers H1 and H2 provide the necessary IO pins to interface the Host Bus Interfaces listed in Table 4-4, "CPU Interface Pin Mapping".

4.2.1 *CPU Interface Pin Mapping*

When connecting the S5U13806B00C to other evaluation platforms, the following host interface pin mapping applies. Note the PCI Bridge Adapter FPGA must be disabled (using DIP switch S1-8) before setting the S5U13806B00C for use with a non-PCI host interface.

Table 4-4 CPU Interface Pin Mapping

S1D13806 Pin Names	Generic	Hitachi SH-4/SH-3	MIPS/ISA	Motorola MC68K Bus 1	Motorola MC68K Bus 2	Motorola PowerPC	PC Card	Philips PR31500 /PR31700	Toshiba TX3912
AB20	A20	A20	LatchA20	A20	A20	A11	A20	ALE	ALE
AB19	A19	A19	SA19	A19	A19	A12	A19	/CARDREG	CARDREG*
AB18	A18	A18	SA18	A18	A18	A13	A18	/CARDIORD	CARDIORD*
AB17	A17	A17	SA17	A17	A17	A14	A17	/CARDIOWR	CARDIOWR*
AB[16:13]	A[16:13]	A[16:13]	SA[16:13]	A[16:13]	A[16:13]	A[15:18]	A[16:13]	Connected to V_{DD}[1]	
AB[12:1]	A[12:1]	A[12:1]	SA[12:1]	A[12:1]	A[12:1]	A[19:30]	A[12:1]	A[12:1]	A[12:1]
AB0	A0[1]	A0[1]	SA0	LDS#	A0	A31	A0[1]	A0	A0
DB[15:8]	D[15:0]	D[15:8]	SD[15:0]	D[15:8]	D[31:24]	D[0:7]	D[15:0]	D[23:16]	D[23:16]
DB[7:0]	D[7:0]	D[7:0]	SD[7:0]	D[7:0]	D[23:16]	D[8:15]	D[7:0]	D[31:24]	D[31:24]
WE1#	WE1#	WE1#	SBHE#	UDS#	DS#	$\overline{\text{BI}}$	-CE2	/CARDxCSH	CARDxCSH*
M/R#	External Decode							Connected to V_{DD}[1]	
CS#	External Decode							Connected to V_{DD}[1]	
BUSCLK	BCLK	CKIO	CLK	CLK	CLK	CLKOUT	CLK	DCLKOUT	DCLKOUT
BS#	Connected to V_{DD}[1]	BS#	Connected to V_{DD}[1]	AS#	AS#	$\overline{\text{TS}}$	Connected to V_{DD}[1]	Connected to V_{DD}[1]	
RD/WR#	RD1#	RD/WR#	Connected to V_{DD}[1]	R/W#	R/W#	$\overline{\text{RD/WR}}$	-CE1	/CARDxCSL	CARDxCSL*
RD#	RD0#	RD#	MEMR#	Connected to V_{DD}[1]	SIZ1	TSIZ0	-OE	/RD	RD*
WE0#	WE0#	WE0#	MEMW#	Connected to V_{DD}[1]	SIZ0	TSIZ1	-WE	/WE	WE*
WAIT#	WAIT#	RDY# /WAIT#	IOCHRDY	DTACK#	DSACK1#	$\overline{\text{TA}}$	-WAIT	/CARDxWAIT	CARDxWAIT*
RESET#	RESET#	RESET#	inverted RESET	RESET#	RESET#	RESET#	inverted RESET	RESET#	PON*

Note: Pull-up resistors are not provided on the S5U13806B00C. However, these pins are not used in their corresponding CPU interface mode and systems are responsible for connecting them to V_{DD} using external pull-up resistors.

4.3 External CPU Host Connector Pin Mapping

The pinouts for Connector H1 are listed in the following table.

Table 4-5 External CPU Host Connector (H1) Pinout

Connector Pin No.	Comments
1	Connected to DB0 of the S1D13806
2	Connected to DB1 of the S1D13806
3	Connected to DB2 of the S1D13806
4	Connected to DB3 of the S1D13806
5	Ground
6	Ground
7	Connected to DB4 of the S1D13806
8	Connected to DB5 of the S1D13806
9	Connected to DB6 of the S1D13806
10	Connected to DB7 of the S1D13806
11	Ground
12	Ground
13	Connected to DB8 of the S1D13806
14	Connected to DB9 of the S1D13806
15	Connected to DB10 of the S1D13806
16	Connected to DB11 of the S1D13806
17	Ground
18	Ground
19	Connected to DB12 of the S1D13806
20	Connected to DB13 of the S1D13806
21	Connected to DB14 of the S1D13806
22	Connected to DB15 of the S1D13806
23	Connected to RESET# of the S1D13806
24	Ground
25	Ground
26	Ground
27	+12 volt supply
28	+12 volt supply
29	Connected to WE0# of the S1D13806
30	Connected to WAIT# of the S1D13806
31	Connected to CS# of the S1D13806
32	Connected to MR# of the S1D13806
33	Connected to WE1# of the S1D13806
34	Not connected

The pinouts for Connector H2 are listed in the following table.

Table 4-6 External CPU Host Connector (H2) Pinout

Connector Pin No.	Comments
1	Connected to A0 of the S1D13806
2	Connected to A1 of the S1D13806
3	Connected to A2 of the S1D13806
4	Connected to A3 of the S1D13806
5	Connected to A4 of the S1D13806
6	Connected to A5 of the S1D13806
7	Connected to A6 of the S1D13806
8	Connected to A7 of the S1D13806
9	Ground
10	Ground
11	Connected to A8 of the S1D13806
12	Connected to A9 of the S1D13806
13	Connected to A10 of the S1D13806
14	Connected to A11 of the S1D13806
15	Connected to A12 of the S1D13806
16	Connected to A13 of the S1D13806
17	Ground
18	Ground
19	Connected to A14 of the S1D13806
20	Connected to A15 of the S1D13806
21	Connected to A16 of the S1D13806
22	Connected to A17 of the S1D13806
23	Connected to A18 of the S1D13806
24	Connected to A19 of the S1D13806
25	Ground
26	Ground
27	+5 volt supply
28	+5 volt supply
29	Connected to RD/WR# of the S1D13806
30	Connected to BS# of the S1D13806
31	Connected to BUSCLK of the S1D13806
32	Connected to RD# of the S1D13806
33	Connected to A20 of the S1D13806
34	Not connected

5 LCD INTERFACE

The S1D13806 supports 4/8-bit single and dual passive monochrome panels, 4/8/16-bit single and dual passive color, and 9/12/18/2x9/2x12-bit active matrix TFT/D-TFD panels. All necessary signals are provided on the 40-pin LCD connector (H3).

5.1 LCD Connector Pin Mapping

Table 5-1 LCD Signal Connector (H3)

S1D13806 Pin Names	Connector Pin No.	Monochrome Passive Panel			Color Passive Panel						Color Active (TFT) Panel				
		Single		Dual	Single	Single Format 1	Single Format 2	Single	Dual						
		4-bit	8-bit	8-bit	4-bit	8-bit	8-bit	16-Bit	8-bit	16-bit	9-bit	12-bit	18-bit	2x9-bit	2x12-bit
FPDAT0	1	driven 0	D0	LD0	driven 0	D0	D0	D0	LD0	LD0	R2	R3	R5	R02	R03
FPDAT1	3	driven 0	D1	LD1	driven 0	D1	D1	D1	LD1	LD1	R1	R2	R4	R01	R02
FPDAT2	5	driven 0	D2	LD2	driven 0	D2	D2	D2	LD2	LD2	R0	R1	R3	R00	R01
FPDAT3	7	driven 0	D3	LD3	driven 0	D3	D3	D3	LD3	LD3	G2	G3	G5	G02	G03
FPDAT4	9	D0	D4	UD0	D0	D4	D4	D8	UD0	UD0	G1	G2	G4	G01	G02
FPDAT5	11	D1	D5	UD1	D1	D5	D5	D9	UD1	UD1	G0	G1	G3	G00	G01
FPDAT6	13	D2	D6	UD2	D2	D6	D6	D10	UD2	UD2	B2	B3	B5	B02	B03
FPDAT7	15	D3	D7	UD3	D3	D7	D7	D11	UD3	UD3	B1	B2	B4	B01	B02
FPDAT8	17	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D4	driven 0	LD4	B0	B1	B3	B00	B01
FPDAT9	19	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D5	driven 0	LD5	driven 0	R0	R2	driven 0	R00
FPDAT10	21	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D6	driven 0	LD6	driven 0	driven 0	R1	R12	R13
FPDAT11	23	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D7	driven 0	LD7	driven 0	G0	G2	driven 0	G00
FPDAT12	25	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D12	driven 0	UD4	driven 0	driven 0	G1	G12	G13
FPDAT13	27	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D13	driven 0	UD5	driven 0	driven 0	G0	G11	G12
FPDAT14	29	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D14	driven 0	UD6	driven 0	B0	B2	driven 0	B00
FPDAT15	31	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D15	driven 0	UD7	driven 0	driven 0	B1	B12	B13
FPDAT16	4	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B0	B11	B12
FPDAT17	6	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	R0	R11	R12
FPDAT18	10	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	R10	R11
FPDAT19	12	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	R10
FPDAT20	16	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	G10	G11
FPDAT21	18	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	G10
FPDAT22	22	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B10	B11
FPDAT23	24	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B10
FPSHIFT	33	FPSHIFT													
DRDY	35 & 38	MOD				FPSHIFT 2	MOD				DRDY				
FPLINE	37	FPLINE													
FPFRAME	39	FPFRAME													
GND	2, 8, 14, 20, 26	GND													
N/C	28	N/C													
VLCD	30	Adjustable -24 to -14 negative LCD bias													
LCDVCC	32	+5.0V or +3.3V according to JP1													
+12V	34	+12V													
VDDH	36	Adjustable +24 to +40 positive LCD bias													
GPIO11	40	GPIO11[1]													

Note: 1. The polarity of GPIO11 sent to pin 40 can be inverted using jumper JP3. However, JP3 does not affect the polarity of the signals controlling the LCD bias power supplies.

5.2 Voltage Translation Buffers

The S5U13806B00C is designed with voltage translation buffers. This allows both 3.3V and 5.0V panels to be supported. Jumper JP1 selects the panel voltage (see Section 3.2, “Configuration Jumpers” on page 5).

5.3 Adjustable LCD Panel Positive Supply (V_{DDH})

For LCD panels which require a positive bias voltage between +24V and +40V (I_{out} = 45mA), a power supply has been provided as an integral part of the S5U13806B00C design. The voltage on V_{DDH} can be adjusted using R22 (200K potentiometer) to provide an output voltage from +24V to +40V.

The voltage on V_{DDH} is enabled/disabled using the S1D13806 pin GPIO11 (active high). A pull-down resistor holds GPIO11 low at reset and, unless initialized at power-up, the bias voltage to the panel is off. For further information on controlling the LCD bias voltage, refer to Section 5.5, “LCD Power Sequencing” on page 14.

Note: **Before connecting the panel.** set the potentiometer according to the panel’s specific voltage requirements.

5.4 Adjustable LCD Panel Negative Power Supply (V_{LCD})

For LCD panels which require a negative bias voltage between -14V and -24V (I_{out} = 25mA), a power supply has been provided as an integral part of the S5U13806B00C design. The voltage on V_{LCD} can be adjusted using R29 (100K potentiometer) to provide an output voltage from -14V to -24V.

The voltage on V_{LCD} is enabled/disabled using the S1D13806 pin GPIO11 (active high). A pull-down resistor holds GPIO11 low at reset and, unless initialized at power-up, the bias voltage to the panel is off. For further information on controlling the LCD bias voltage, refer to Section 5.5, “LCD Power Sequencing” on page 14.

Note: **Before connecting the panel.** set the potentiometer according to the panel’s specific voltage requirements.

5.5 LCD Power Sequencing

LCD power sequencing ensures that bias voltage is not supplied to the LCD panel while the control signals are inactive. This prevents long term damage to the panel and avoids unsightly “lines” at power-on/power-off. LCD power sequencing for power-off requires a delay between disabling of the LCD power and disabling the LCD signals. Power-on requires the LCD signals to be active prior to providing power to the LCD. The time intervals vary depending on the LCD bias power supply design. The LCD bias power supplies on the S5U13806B00C require approximately 1.2 seconds to fully discharge (depending on the load).

The S5U13806B00C design uses GPIO11 (active high) to enable/disable the bias power supplies. A pull-down resistor is added to GPIO11 ensuring that the bias power supplies are off before the S1D13806 is initialized. GPIO11 is configured using REG[005h] and controlled using REG[009h]. For further information on the S1D13806 registers, refer to the “*S1D13806 Hardware Functional Specification*,” document number X28B-A-001-xx.

Note: The software test utilities accompanying the S5U13806B00C handle the LCD power sequencing automatically and comply with most LCD panel power-on/power-off requirements. When connecting the S5U13806B00C to other platforms, ensure that the software is designed to handle the LCD power sequencing for the panel under test.

6 CRT/TV INTERFACE

6.1 CRT/TV Connectors Pin Mapping

The S1D13806 is designed with an embedded DAC for CRT and TV support. CRT/TV signals are supplied through an analog CRT connector (J3), Composite Video connector (J2), and S-Video connector (J4) as follows.

Table 6-1 CRT/TV Connectors Pin Mapping

S1D13806 Pin Names	CRT (J3)		Composite Video (J2)		S-Video (J4)	
	Connector Pin No.	Signal Name	Connector Pin No.	Signal Name	Connector Pin No.	Signal Name
HRTC	13	Horizontal retrace	—	N/A	—	N/A
VRTC	14	Vertical retrace	—	N/A	—	N/A
RED	1	Red	—	N/A	3	Luminance
GREEN	2	Green	1	Composite	—	N/A
BLUE	3	Blue		N/A	4	Chrominance
(GND)	5,6,7,8,10	GND	2	GND	1,2	GND

Note: Either CRT **or** TV can be active at any given time.

Note: For TV, either PAL or NTSC format **and** Composite or S-Video output can be enabled at any given time.

6.2 DAC Output Level Select for CRT

When CRT mode is enabled, the S1D13806 does not use the whole range of the DAC. Therefore, while in CRT mode the DAC inputs may be shifted up by 1 (controlled by REG[05Bh] bit 3). This allows I_{REF} (current reference for the DAC) to be reduced by half, thus reducing the DAC power consumption.

This option can be configured using the configuration utility 1386CFG only when CRT output is selected. For further information on 1386CFG, refer to the "*1386CFG Configuration Utility Users Manual*," document number X28B-B-001-xx.

Note: When this option is selected, S5U13806B00C jumper JP1 (I_{REF} for DAC) must be set to position 1-2. For further information, see Section 3.2, "Configuration Jumpers" on page 5.

7 MediaPlug Interface (for WINNOV Videum®Cam)

The S5U13806B00C is designed with a MediaPlug connector (J1) for connecting to the WINNOV Videum®Cam digital camera.

When using the WINNOV Videum®Cam digital camera, configuration dip switch S1-7 (CONF7) must be set before power-up or reset to enable the MediaPlug interface. For further information on configuring DIP switch S1, refer to Section 3.1, “Configuration DIP Switches” on page 3.

7.1 MediaPlug Interface Pin Mapping

Table 7-1 MediaPlug Connector (J1) Pin Mapping

S1D13806 Pin Names	Connector Pin No.	MediaPlug I/F
VMP0	7	VMPCLKN
VMP1	8	VMPCLK
VMP2	5	VMPD3
VMP3	6	VMPD2
VMP4	4	VMPD1
VMP5	3	VMPD0
VMP6	2	VMPRCTL
VMP7	1	VMPLCTL
GPIO12[1]	9	VMPEPWR
(GND)	10-12	Ground

Note: 1. When the MediaPlug interface is enabled using S1-7 (CONF7), GPIO12 is configured as the MediaPlug output pin VMPEPWR and cannot be controlled by REG[005h] and REG[009h]. It must be controlled using the MediaPlug LCMD register (REG[1000h] bit 1).

8 Clock Synthesizer and Clock Options

For maximum flexibility, the S5U13806B00C implements a Cypress ICD2061A Clock Generator. MCLKOUT from the clock synthesizer is connected to CLKI of the S1D13806, and VCLKOUT from the clock synthesizer is connected to CLKI2 of the S1D13806. A 14.31818MHz crystal (Y1) is connected to XTALIN and XTALOUT of the clock synthesizer and a 17.734475MHz oscillator (U8) is connected to FEATCLK of the clock synthesizer. The diagram below shows a simplified representation of the clock synthesizer connections.

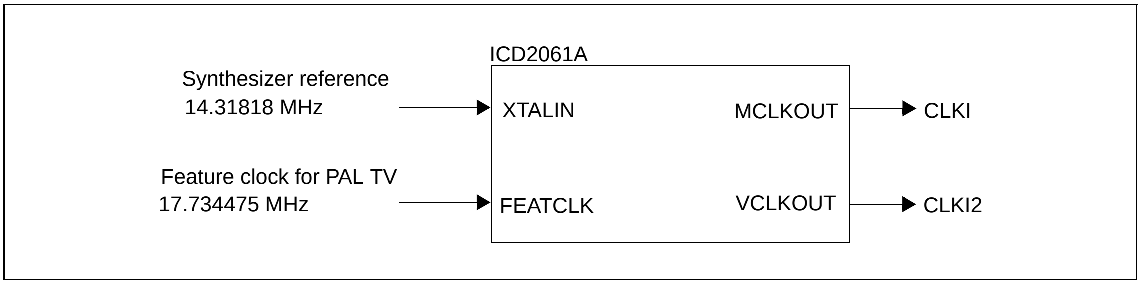


Figure 8-1 Symbolic Clock Synthesizer Connections

Upon power-up, CLKI (MCLKOUT) is configured to be 40MHz and CLKI2 (VCLKOUT) is configured to be 25.175MHz. CLKI3 of the S1D13806 is connected to a 50MHz oscillator (U10) as the MCLK clock source. CLKI and BUSCLK can also be selected as the clock source for MCLK using the Memory Clock Configuration Register (REG[010h]).

8.1 Clock Programming

The S1D13806 utilities automatically program the clock generator.

For further information on programming the clock generator, refer to the "*Cypress ICD2061A specification*."

Note: When CLKI and CLKI2 are programmed to multiples of each other (e.g. CLKI = 20MHz, CLKI2 = 40MHz), the clock output signals from the Cypress clock generator may jitter. Refer to the Cypress ICD2061A specification for details.

To avoid this problem, set CLKI and CLKI2 to different frequencies and configure both LCD PCLK and CRT/TV PCLK to use the same clock input (CLKI or CLKI2).Then use the S1D13806 internal clock divides (LCD PCLK Divide REG[014h], CRT/TV PCLK Divide REG[018h]) to obtain the lower frequencies.

9 REFERENCES

- SEIKO EPSON CORP., "*S1D13806 Hardware Functional Specification*," Document Number X28B-A-001-xx.
- SEIKO EPSON CORP., "*S1D13806 Programming Notes and Examples*," Document Number X28B-G-003-xx.
- Cypress Semiconductor Corporation, *ICD2061A Data Sheet.*

10 PARTS LIST

Table 10-1 Parts List

Item	Quantity	Reference	Part	Description	Manufacturer / Part No. / Assembly Instructions
1	1	C1	22u 10V	"Tantalum C-size, 10V +/-10%"	
2	2	"C2,C34"	10u 10V	"Tantalum C-size, 10V +/-10%"	
3	26	"C3-16,C31-33,C35,C36,C39,C40,C42,C58-61"	0.1u	"50V X7R +/-5%, 1206 pckg"	
4	9	C17-25	0.22u	"50V X7R +/-5%, 1206 pckg"	
5	7	"C26,C28,C29,C37,C38,C41,C46"	68u 10V	"Tantalum D-size, 10V +/-10%"	
6	2	"C27,C30"	33u 20V	"Tantalum D-size, 20V +/-10%"	
7	3	C43-C45	10uF/63V	"Electrolytic, Radial Lead 63V +/-20%"	NIPPON/UNITED CHEMI-CON KMF63VB10RM5X11LL or equivalent
8	1	C47	56uF/35V	"Electrolytic, Radial Lead 35V +/-20%"	NIPPON/UNITED CHEMI-CON LXV35VB56RM6X11LL or equivalent
9	6	"C48-50,C52,C54,C55"	33pF	"50V X7R +/-5%, 1206 pckg"	
10	1	C51	0.01uF	"50V X7R +/-5%, 1206 pckg"	
11	1	C53	10uF 16V	"Tantalum C-size, 16V +/-10%"	
12	2	"C57,C56"	220pF	"50V X7R +/-5%, 1206 pckg"	
13	2	"C62,C63"	n/p	1206 pckg	Do not populate
14	1	D1	BAT54	Schottky Barrier Diode / SOT-23	"Liteon BAT54 or equivalent, NOT BAT54A !"
15	4	D2-5	BAV99	Dual Diode / SOT-23	
16	2	"H1,H2"	HEADER 17X2	"2x17 .1""x.1"" pitch unshrouded header"	
17	1	H3	HEADER 20X2	"2x20, .025"" sq. shrouded header, keyed"	Thomas&Betts P/N:636-4207 or equivalent
18	3	JP1-3	HEADER 3	"1x3 .1"" pitch unshrouded header"	
19	1	JP5	HEADER 3	"1x3 .1"" pitch unshrouded header"	Do not populate--mask
20	1	JP4	HEADER 2	"1x2 .1"" pitch unshrouded header"	Do not populate--mask
21	1	J1	MediaPlug Conn.	9-pin Right Angle PCB Mini DIN Socket	CUI Stack P/N:MD-90S or Digi-Key P/N:CP-2490-ND

Table 10-1 Parts List (Continued)

Item	Quantity	Reference	Part	Description	Manufacturer / Part No. / Assembly Instructions
22	1	J2	C-VIDEO	P.C. Phono Jack	Keystone Electronics Cat. No. 901 or equivalent
23	1	J3	CRT	"Receptacle, Right Angle Mini. Circular DIN"	AMP 749264 or equivalent
24	1	J4	S-VIDEO	"DB15 Female, Right Angle, PCB Mount"	Assman A-HDF 15 A KG/T or equivalent
25	6	"L1-3,L11-13"	Ferrite	Ferrite Bead / SMT	Philips BDS3/3/8.9-4S2
26	1	L4	1uH	1812 SMT inductor	RCD MCI-1812 1uH MT or MSI-1812 1uH MT
27	6	L5-10	150nH	1008 SMT inductor	Panasonic ELJNCR15JF (Digi-Key PCD1210CT-ND) or Delevan 1008-151K (Digi-Key DN08151CT-ND)
28	3	"Q1,Q3,Q5"	MMBT2222A	NPN Transistor / SOT-23	Motorola or equivalent
29	1	Q2	MMBT3906	PNP Transistor / SOT-23	Motorola or equivalent
30	1	Q4	NDS9400A	P-Channel FET / SO-8	"Fairchild, National or equivalent"
31	14	"R1-7,R15-18,R23,R24,R46"	15K	"1206 resistor, 5%"	
32	2	"R8,R32"	1.5K 1%	"1206 resistor, 1%"	
33	1	R9	1K 1%	"1206 resistor, 1%"	
34	1	R10	133 1%	"1206 resistor, 1%"	
35	1	R11	63.4 1%	"1206 resistor, 1%"	
36	3	"R12,R27,R28"	100K	"1206 resistor, 5%"	
37	4	"R13,R14,R19,R20"	1K	"1206 resistor, 5%"	
38	1	R21	470K	"1206 resistor, 5%"	
39	1	R22	200K Pot	"Trim POT, Knob Adjust"	Spectrol 63S204T607 or equivalent
40	2	"R25,R26"	22K	"1206 resistor, 5%"	
41	1	R29	100K Pot	"Trim POT, Knob Adjust"	Spectrol 63S104T607 or equivalent
42	1	R30	6.04K 1%	"1206 resistor, 1%"	
43	6	"R31,R34,R37,R40,R44,R50"	68 Ohms	"1206 resistor, 5%"	
44	3	"R33,R39,R42"	316 1%	"1206 resistor, 1%"	
45	3	"R35,R41,R45"	357 1%	"1206 resistor, 1%"	
46	2	"R36,R38"	137 1%	"1206 resistor, 1%"	
47	2	"R43,R47"	10K	"1206 resistor, 5%"	
48	2	"R48,R51"	22 Ohms	"1206 resistor, 5%"	
49	2	"R49,R52"	33 Ohms	"1206 resistor, 5%"	
50	3	R53-55	150 1%	"1206 resistor, 1%"	
51	8	R56-63	330K	"1206 resistor, 5%"	
52	1	S1	SW DIP-8	"DIP switch, 8-position"	
53	1	S2	SW DIP-4	"DIP switch, 4-position"	Do not populate--mask
54	1	U1	S1D13806F00A	144-pin QFP	

Table 10-1 Parts List (Continued)

Item	Quantity	Reference	Part	Description	Manufacturer / Part No. / Assembly Instructions
55	1	U2	LT1117CST-3.3	Regulator Fixed 3.3V / SOT-223	Linear Technology LT1117CST-3.3
56	1	U3	EPF6016TC144-2	144-pin QFP	Altera EPF6016TC144-2
57	1	U4	EPC1441PC8	8-pin DIP pckg	"Altera EPC1441PC8, programmed, socketed"
58	1	(U4)	8-pin DIP socket	8-pin DIP socket	"Machined socket, 8-pin"
59	1	U5	LT1117CST-5	Regulator Fixed 5V / SOT-223	Linear Technology LT1117CST-5
60	1	U6	74AHC04	SO-14 package	"NS 74VHC04 or TI 74AHC04, SO-14 package"
61	1	U7	ICD2061A	Wide SO-16 package	Cypress ICD2061A
62	1	U8	17.734475MHz	"14-DIP, 4-pin metal can oscillator"	
63	1	U9	LT1117CM-3.3	DPAK SMT regulator	Linear Technology LT1117CM-3.3
64	1	U10	50MHz	"14-DIP, 4-pin metal can oscillator"	
65	1	U11	RD-0412	Xentek RD-0412	Xentek RD-0412
66	1	U12	EPN001	Xentek EPN001	Xentek EPN001
67	4	"U13,U14, U15,U16"	74HCT244	SO-20 package	
68	1	Y1	14.31818MHz	"Fundamental Mode, Parallel Resonant Crystal, HC49 Low Profile pckg."	FOXS/143-20 or equivalent
69	3	"(JP1,JP2,JP3)"	Shunt	".1" shunt for .025"" square-pin jumpers"	"Place at JP1: 1-2, JP2: 1-2 and JP3: 1-2"
70	1		Bracket	PCI bracket	
71	2		Screw	"Pan head, #4-40 x 1/4"""	"Screw, pan head, #4-40 x 1/4""--please assemble bracket onto board"

11 SCHEMATIC DIAGRAMS

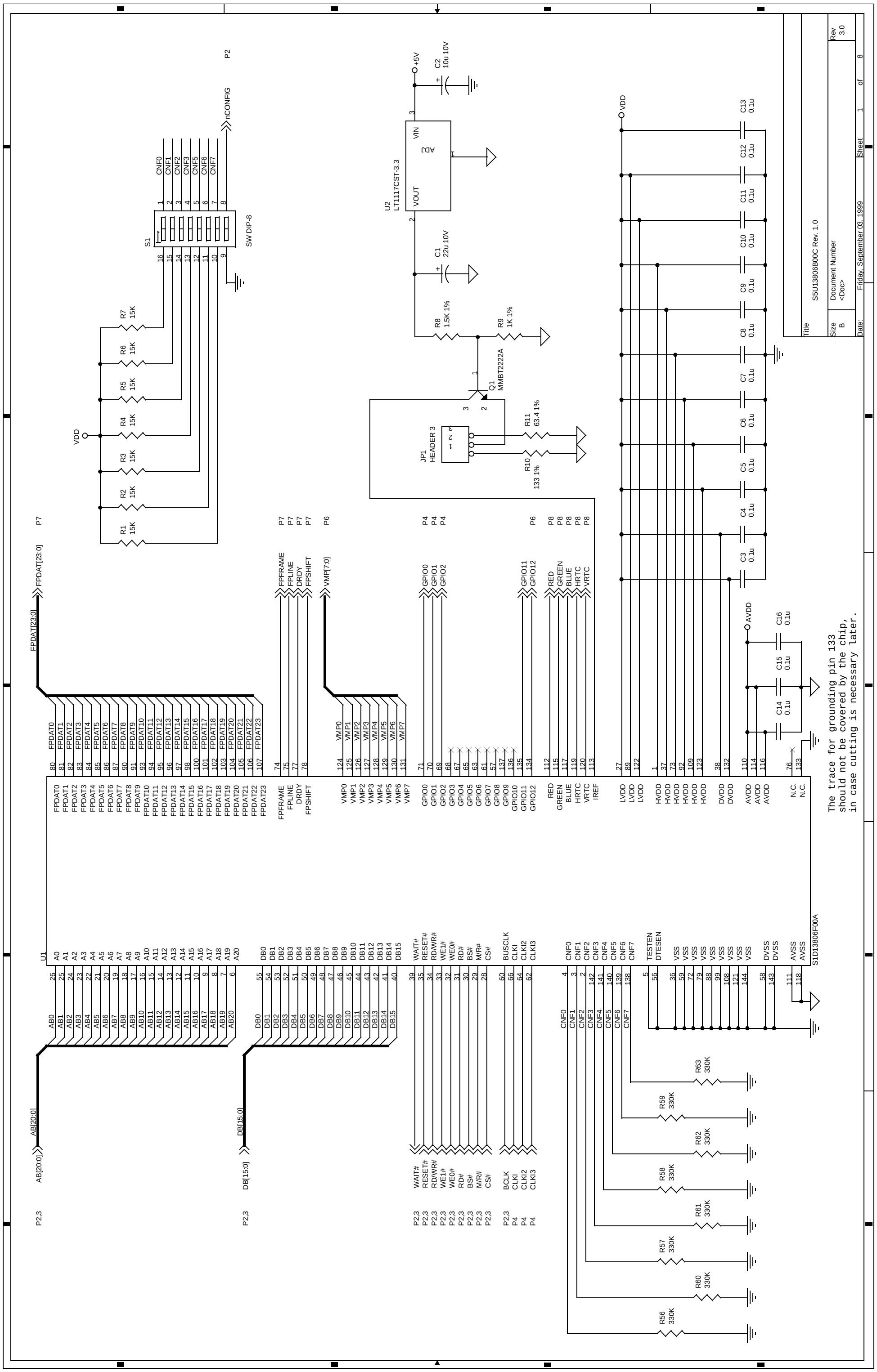


Figure 11-1 S5U13806B00C Schematics (1 of 8)

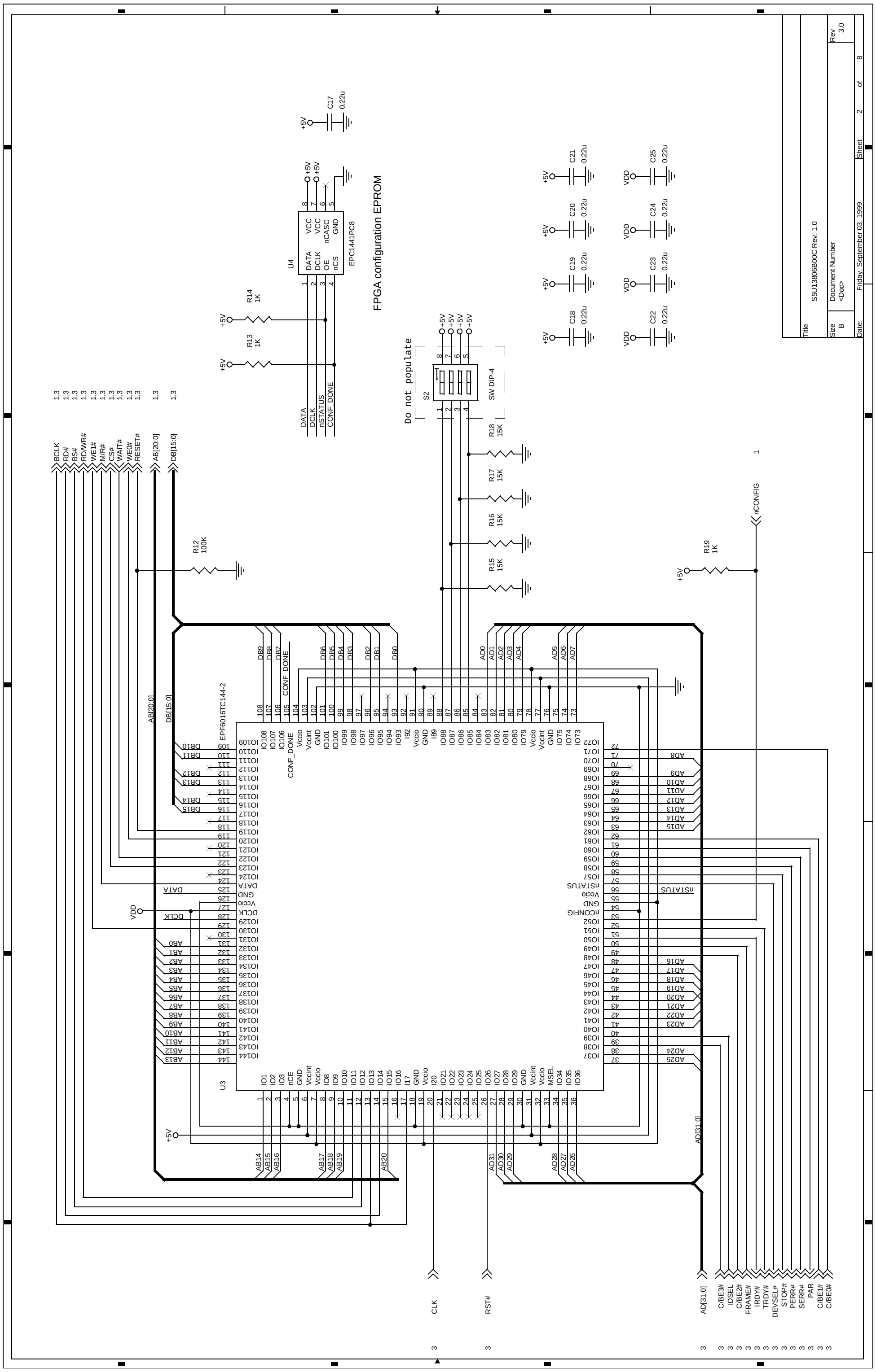


Figure 11-2 S5U13806B00C Schematics (2 of 8)

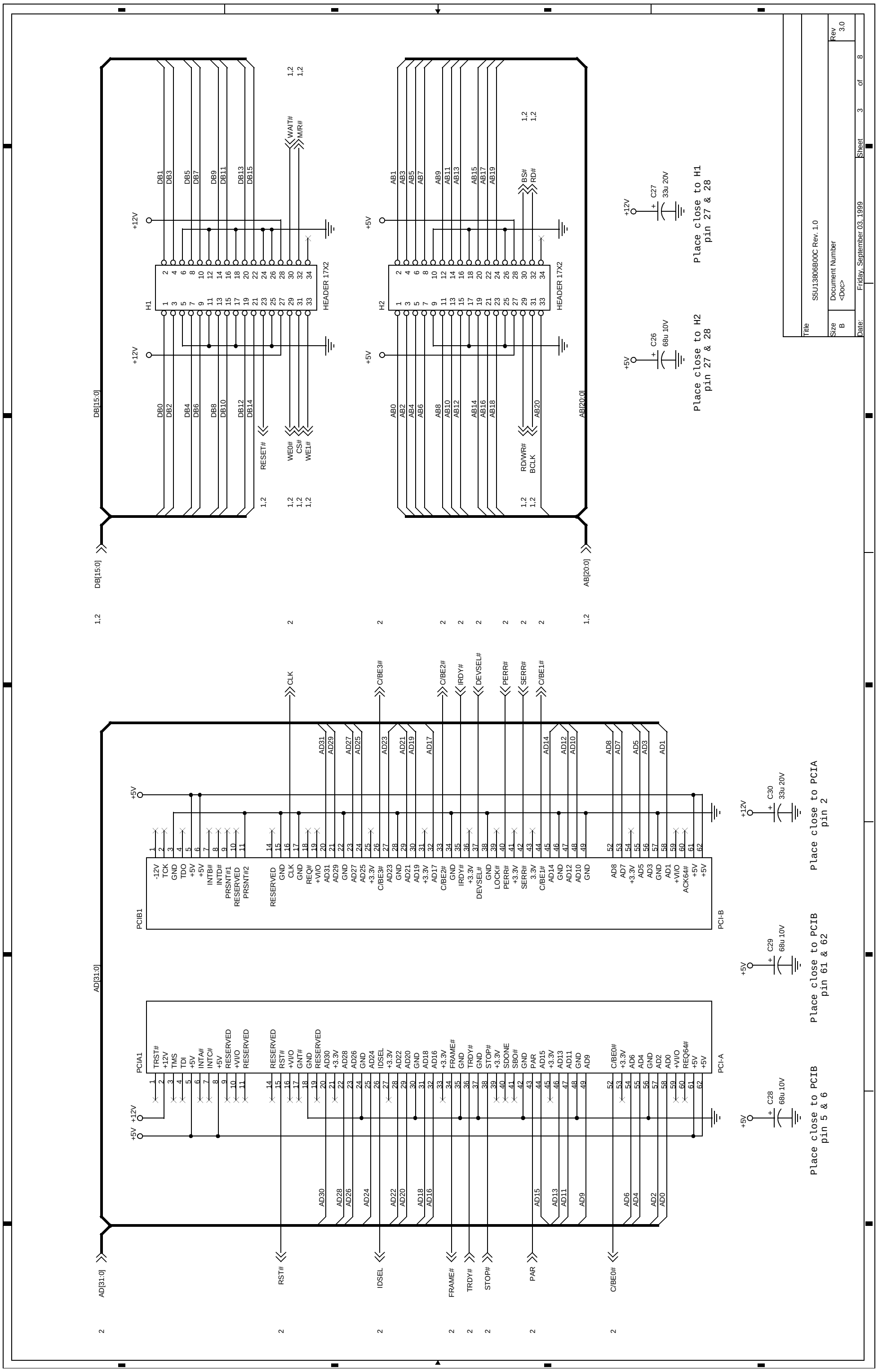


Figure 11-3 S5U13806B00C Schematics (3 of 8)

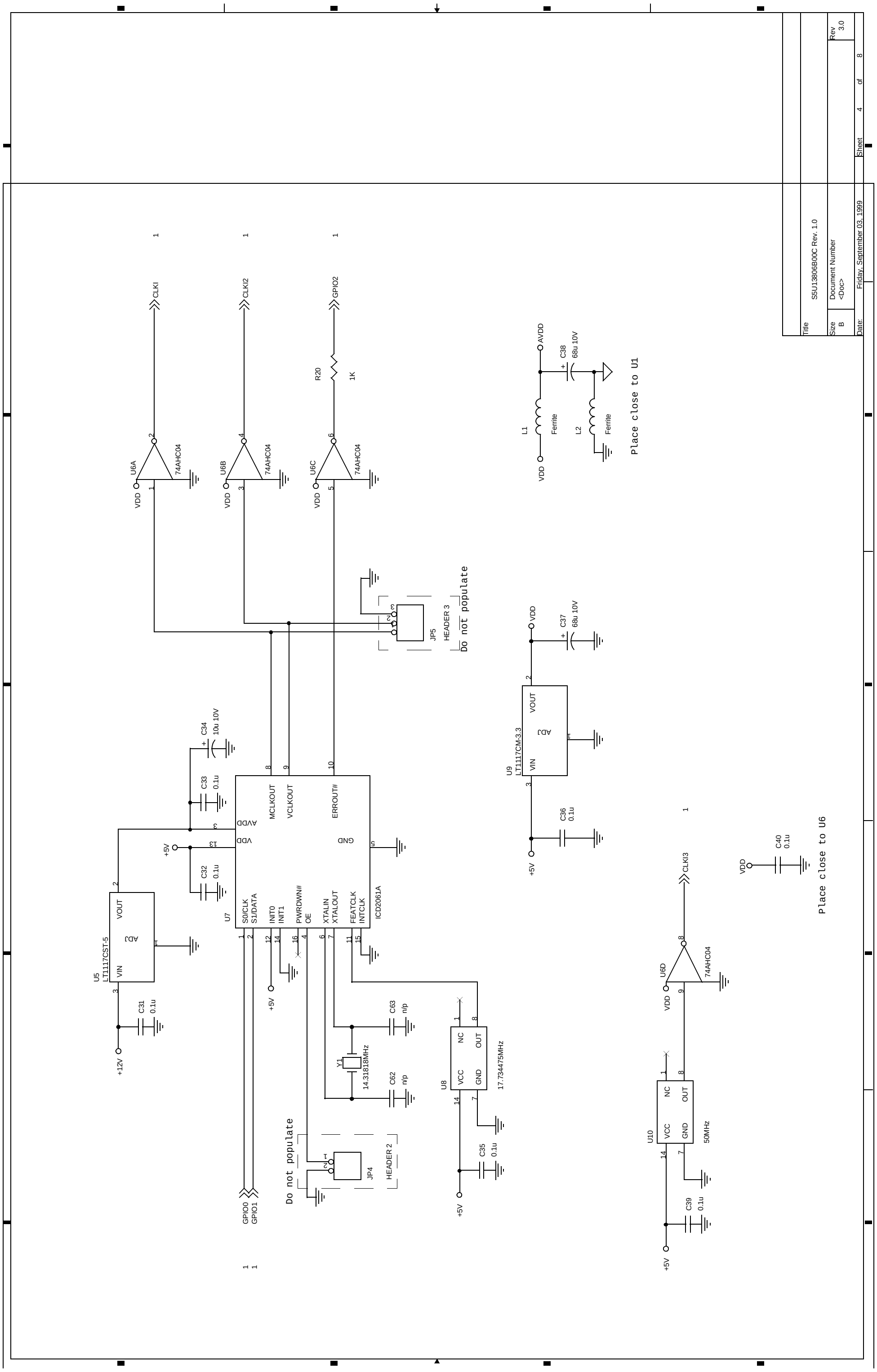


Figure 11-4 S5U13806B00C Schematics (4 of 8)

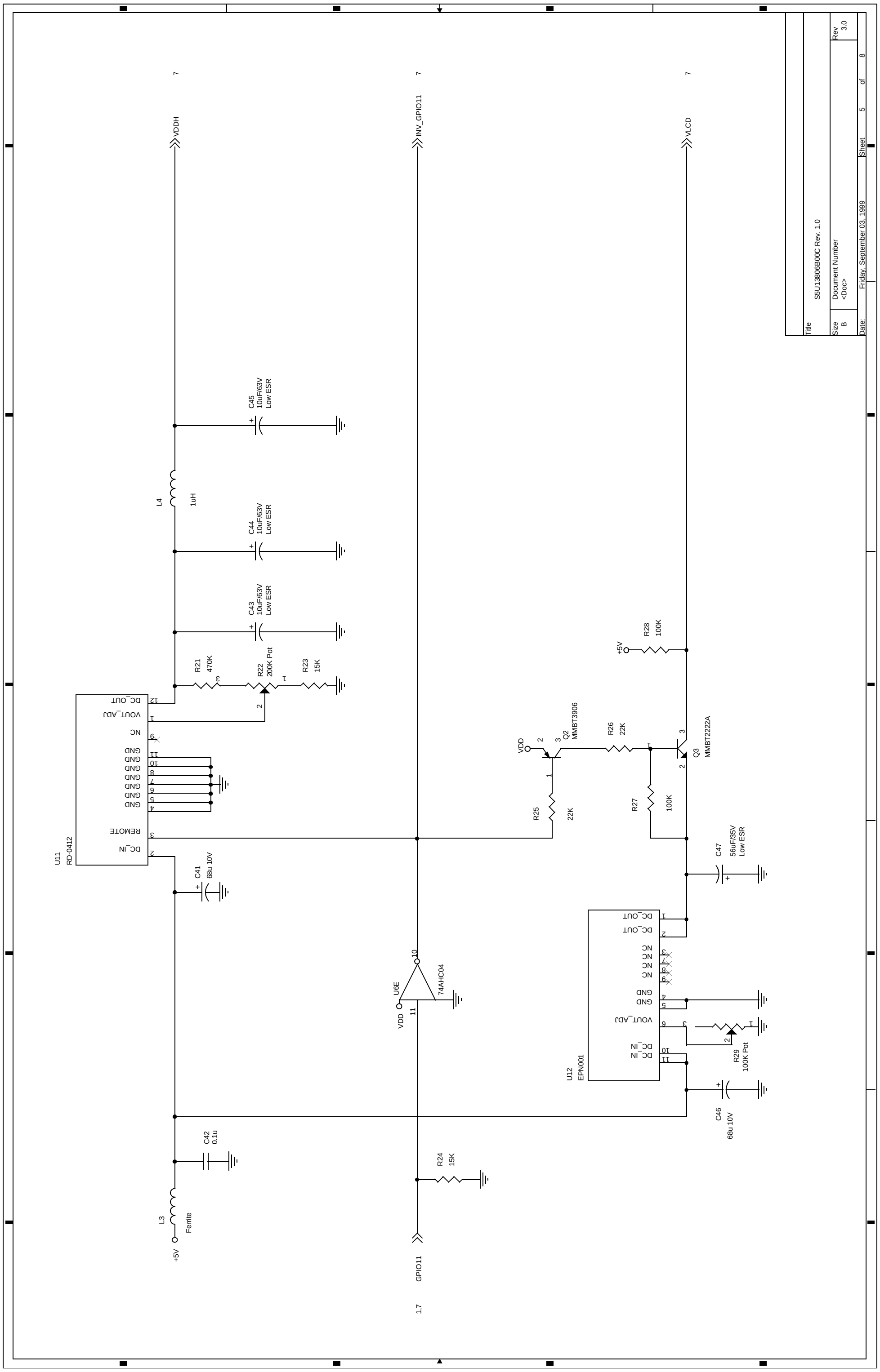


Figure 11-5 S5U13806B00C Schematics (5 of 8)

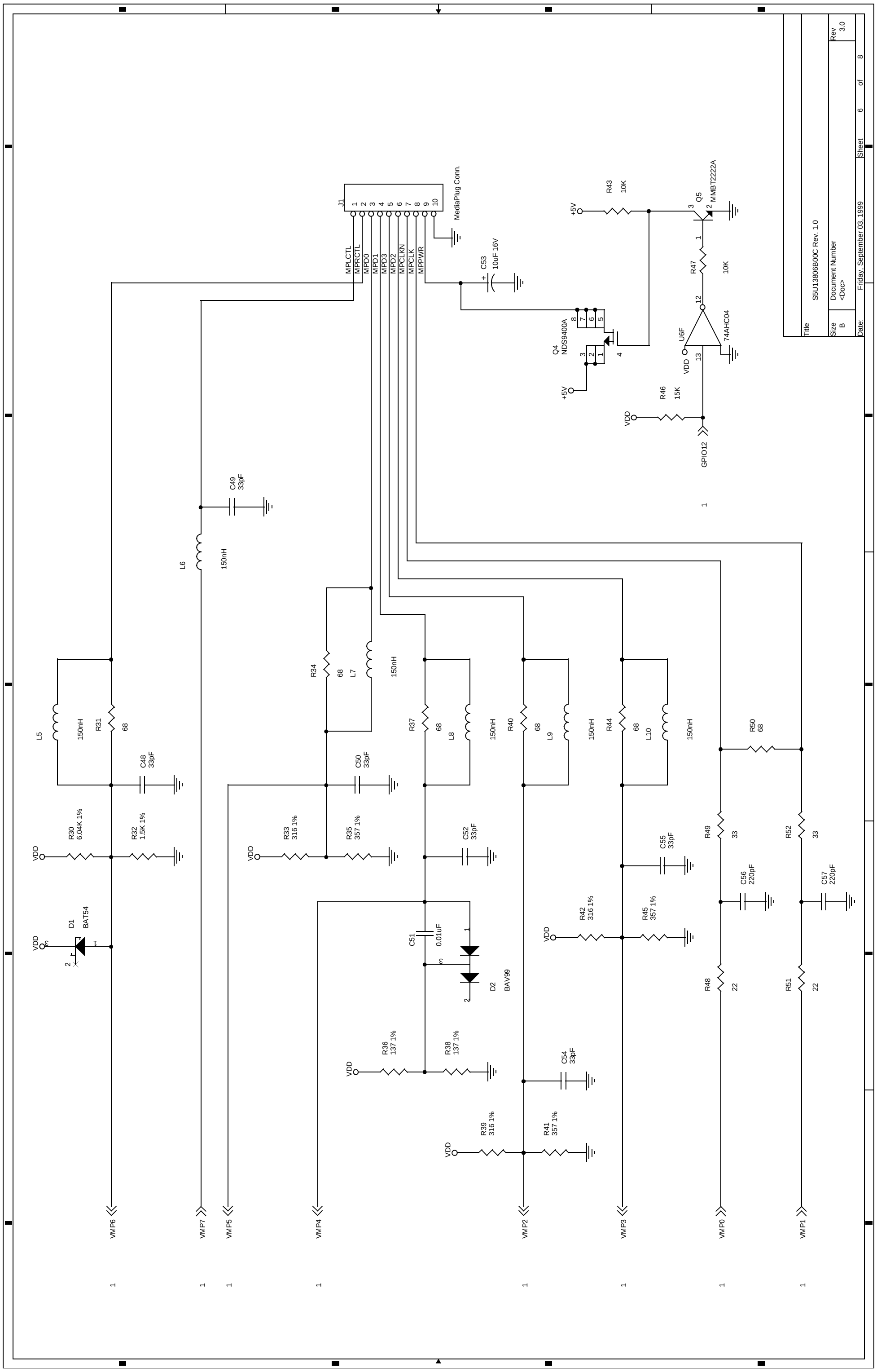


Figure 11-6 S5U13806B00C Schematics (6 of 8)

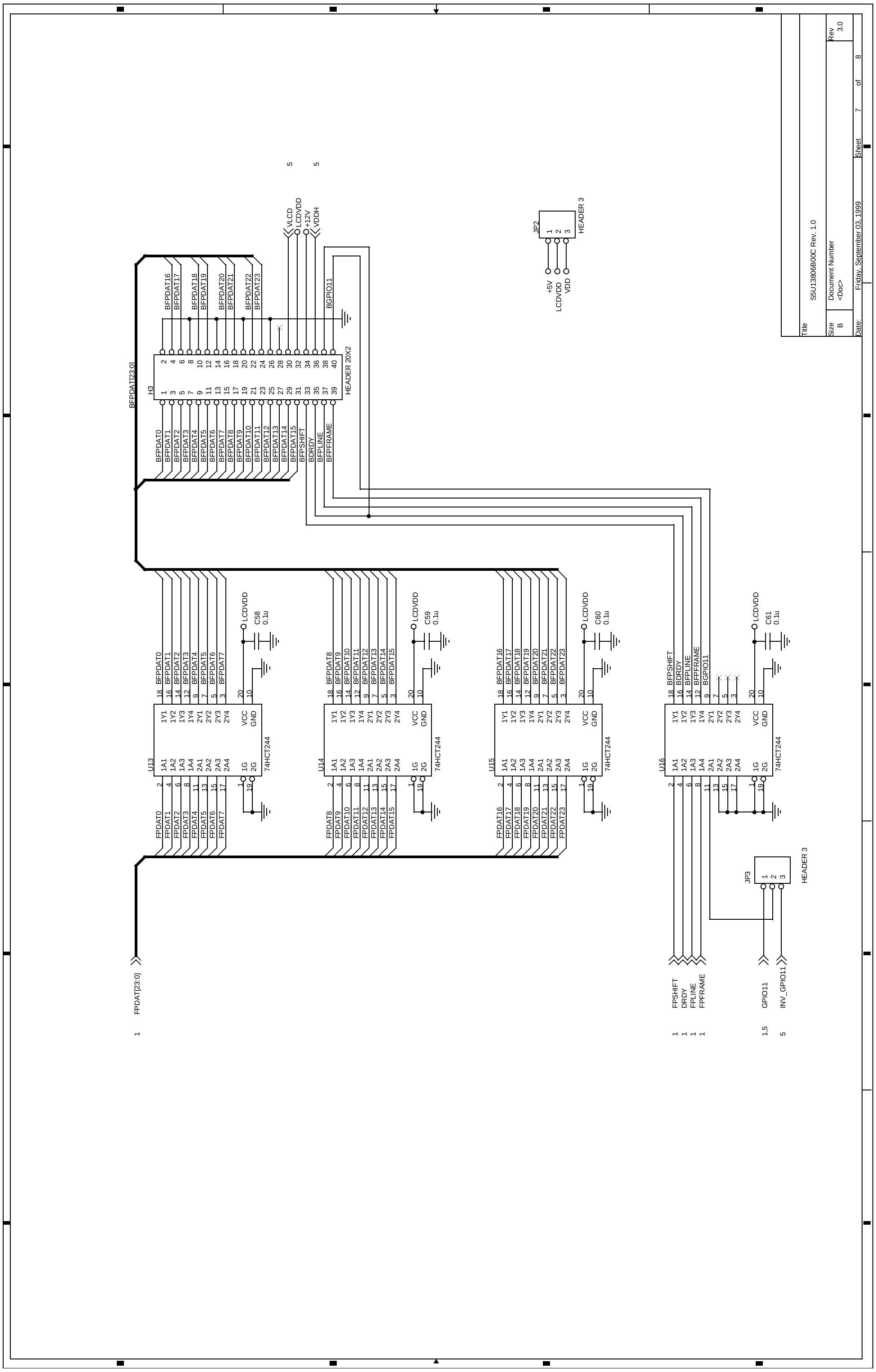


Figure 11-7 S5U13806B00C Schematics (7 of 8)

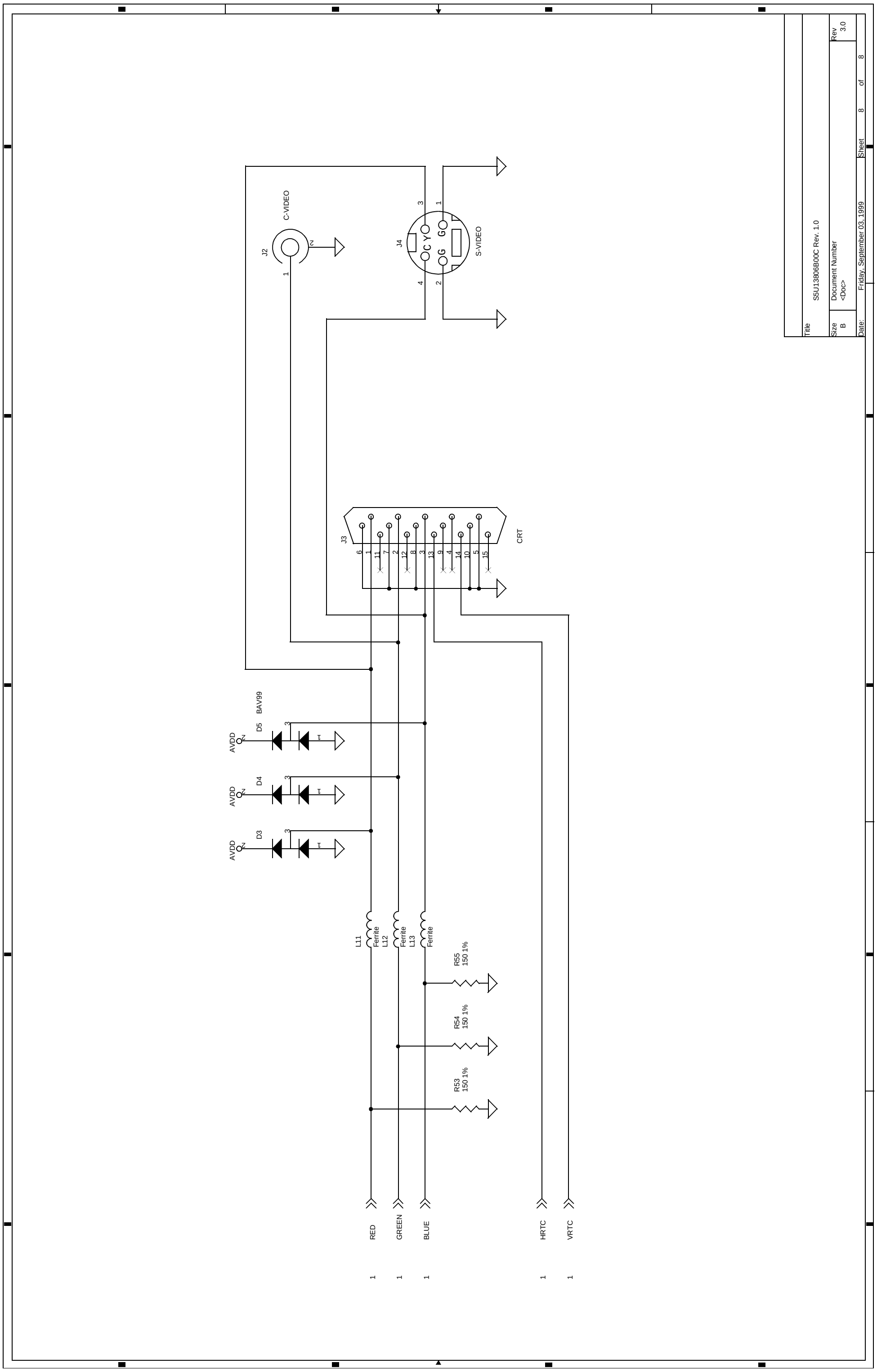


Figure 11-8 S5U13806B00C Schematics (8 of 8)

12 PCB LAYOUT

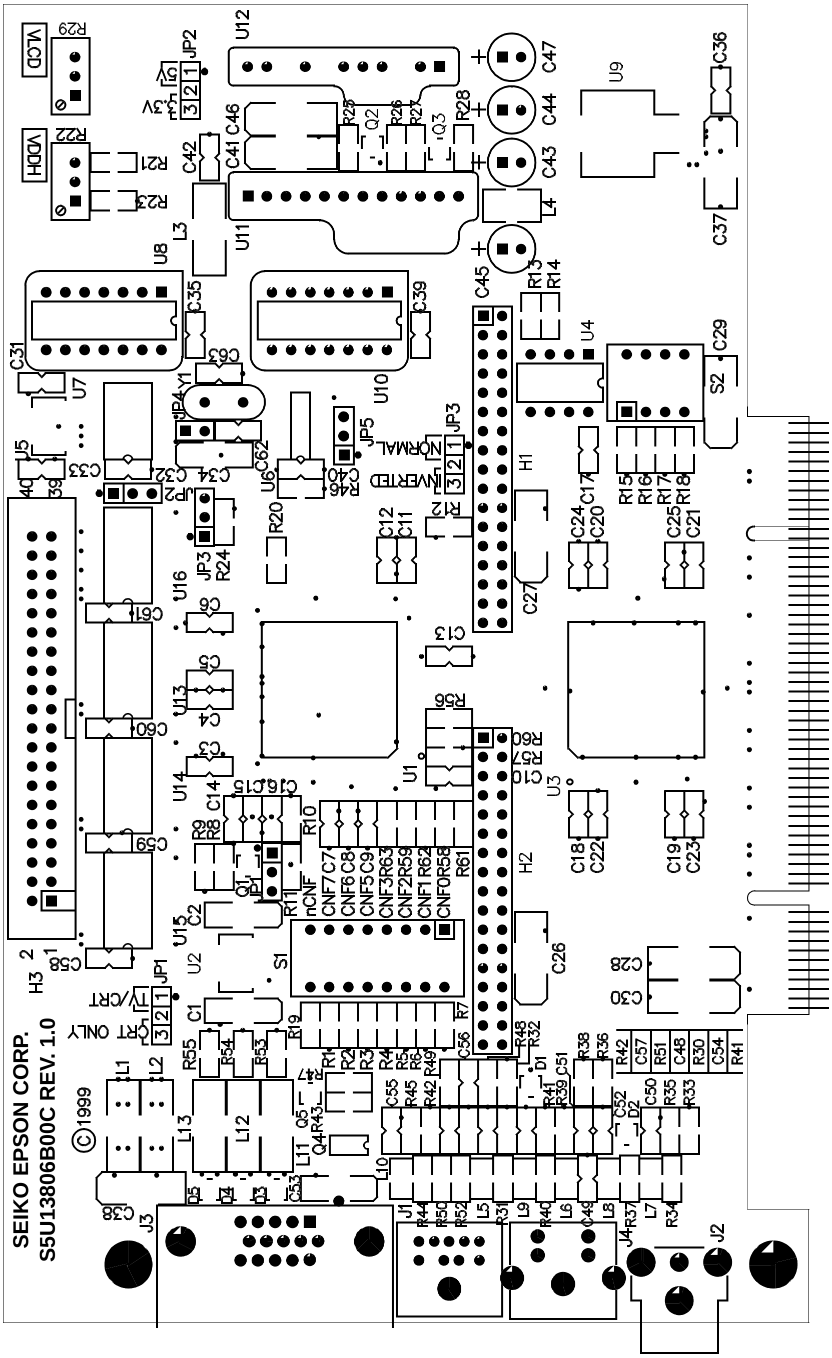


Figure 12-1 PCB Layout

S1D13806 Series

Embedded DRAM Graphics Controller

Application Notes

Table of Contents

List of Figures

List of Tables

1 INTERFACING TO THE PC CARD BUS

1.1 Introduction

This application note describes the hardware and software environment required to provide an interface between the S1D13806 Embedded Memory Display Controller and the PC Card (PCMCIA) bus.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the latest revision of this document before beginning any development.

1.2 Interfacing to the PC Card Bus

1.2.1 The PC Card System Bus

PC Card technology has gained wide acceptance in the mobile computing field as well as in other markets due to its portability and ruggedness. This section is an overview of the operation of the 16-bit PC Card interface conforming to the PCMCIA 2.0/JEIDA 4.1 Standard (or later).

PC Card Overview

The 16-bit PC Card provides a 26-bit address bus and additional control lines which allow access to three 64M byte address ranges. These ranges are used for common memory space, IO space, and attribute memory space. Common memory may be accessed by a host system for memory read and write operations. Attribute memory is used for defining card specific information such as configuration registers, card capabilities, and card use. IO space maintains software and hardware compatibility with hosts such as the Intel x86 architecture, which address peripherals independently from memory space.

Bit notation follows the convention used by most micro-processors, the high bit being the most significant. Therefore, signals A25 and D15 are the most significant bits for the address and data busses respectively.

PC Card bus signals are asynchronous to the host CPU bus signals. Bus cycles are started with the assertion of the CE1# and/or the CE2# card enable signals. The cycle ends once these signals are de-asserted. Bus cycles can be lengthened using the WAIT# signal.

Note: The PCMCIA 2.0/JEIDA 4.1 PC Card Standard supports the two signals WAIT# and RESET which are not supported in earlier versions of the standard. The WAIT# signal allows for asynchronous data transfers for memory, attribute, and IO access cycles. The RESET signal allows resetting of the card configuration by the reset line of the host CPU.

Memory Access Cycles

A data transfer is initiated when a memory address is placed on the PC Card bus and one, or both, of the card enable signals (CE1# and CE2#) are driven low. REG# must be inactive. If only CE1# is driven low, 8-bit data transfers are enabled and A0 specifies whether the even or odd data byte appears on data bus lines D[7:0]. If both CE1# and CE2# are driven low, a 16-bit word transfer takes place. If only CE2# is driven low, an odd byte transfer occurs on data lines D[15:8].

During a read cycle, OE# (output enable) is driven low. A write cycle is specified by driving OE# high and driving the write enable signal (WE#) low. The cycle can be lengthened by driving WAIT# low for the time needed to complete the cycle.

Figure 1-1 illustrates a typical memory access read cycle on the PC Card bus.

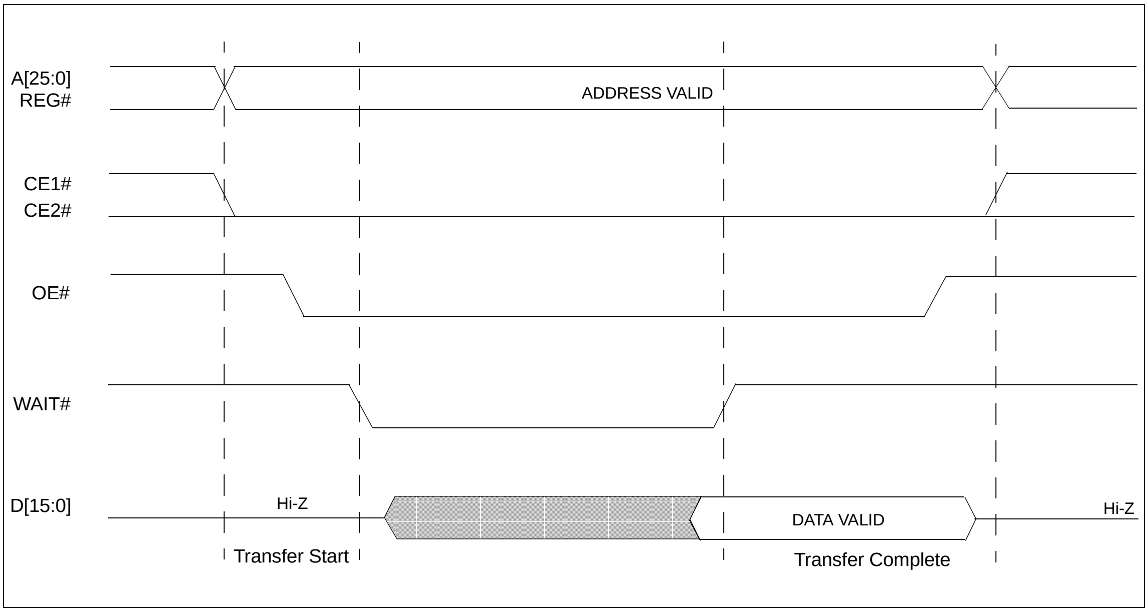


Figure 1-1 PC Card Read Cycle

Figure 1-2 illustrates a typical memory access write cycle on the PC Card bus.

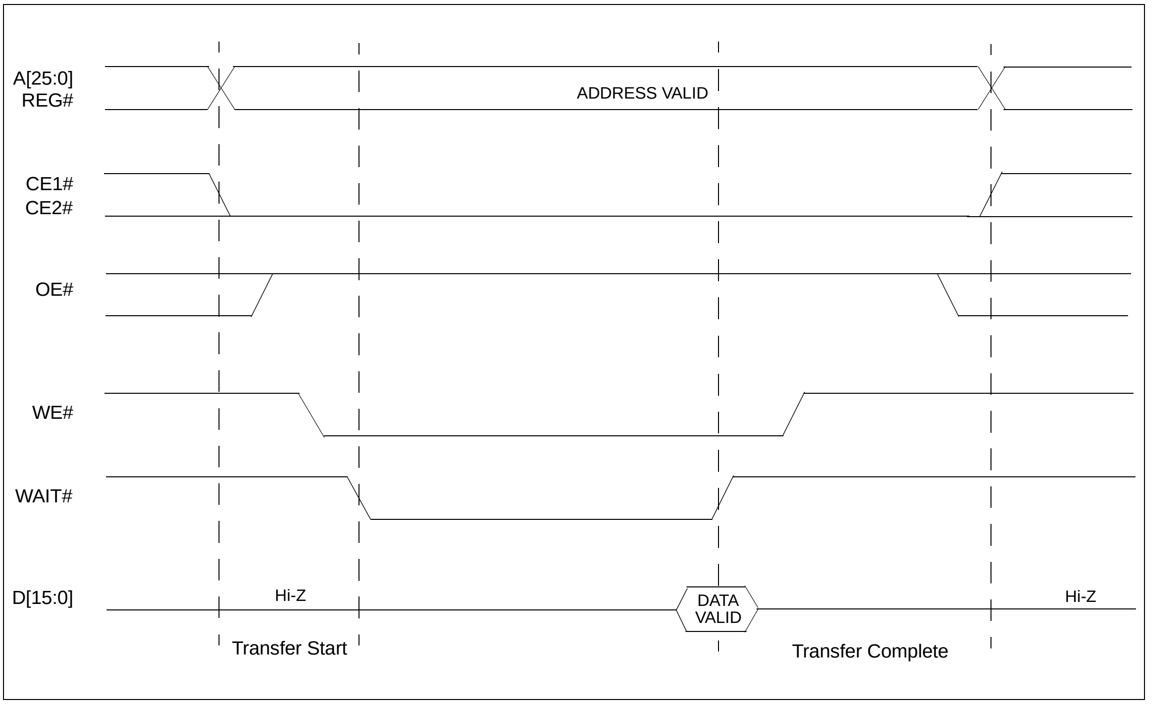


Figure 1-2 PC Card Write Cycle

1.3 S1D13806 Host Bus Interface

The S1D13806 implements a 16-bit PC Card (PCMCIA) Host Bus Interface which is used to interface to the PC Card bus.

The PC Card Host Bus Interface is selected by the S1D13806 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13806 configuration, see Section 1.4.2, “S1D13806 Hardware Configuration” on page 9.

Note: At reset, the Register/Memory Select bit in the Miscellaneous Register (REG[001h] bit 7) is set to 1. This means that only REG[000h] (read-only) and REG[001h] are accessible **until a write to REG[001h] sets bit 7 to 0 making all registers accessible**. When debugging a new hardware design, this can sometimes give the appearance that the interface is not working, so it is important to remember to clear this bit before proceeding with debugging.

1.3.1 PC Card Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 1-1 PC Card Host Bus Interface Pin Mapping

S1D13806 Pin Name	PC Card (PCMCIA)
AB[20:0]	A[20:0][1]
DB[15:0]	D[15:0]
WE1#	-CE2
M/R#	External Decode
CS#	External Decode
BUSCLK	n/a[2]
BS#	V_{DD}
RD/WR#	-CE1
RD#	-OE
WE0#	-WE
WAIT#	-WAIT
RESET#	Inverted RESET

Note: 1. The bus signal A0 is not used by the S1D13806 internally.

2. Although a clock is not directly supplied by the PC Card interface, one is required by the S1D13806 PC Card Host Bus Interface. For an example of how this can be accomplished see the discussion on BUSCLK in Section 1.3.2, “PC Card Host Bus Interface Signals” on page 6.

1.3.2 PC Card Host Bus Interface Signals

The S1D13806 PC Card Host Bus Interface is designed to support processors which interface the S1D13806 through the PC Card bus.

The S1D13806 PC Card Host Bus Interface requires the following signals from the PC Card bus.

- BUSCLK is a clock input which is required by the S1D13806 Host Bus Interface. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock. Since PC Card signalling is independent of any clock, BUSCLK can come from any oscillator already implemented. For example, the source for the CLKI input of the S1D13806 may be used.
- The address inputs AB[20:0], and the data bus DB[15:0], connect directly to the PC Card address (A[20:0]) and data bus (D[15:0]), respectively. CONF[3:0] must be set to select the PC Card Host Bus Interface with little endian mode.
- M/R# (memory/register) selects between memory or register access. It may be connected to an address line, allowing system address A21 to control M/R#.
- Chip Select (CS#) must be driven low whenever the S1D13806 is accessed by the PC Card bus.
- WE1# and RD/WR# connect to -CE2 and -CE1 (the byte enables for the high-order and low-order bytes). They are driven low when the PC Card bus is accessing the S1D13806.
- RD# connects to -OE (the read enable signal from the PC Card bus).
- WE0# connects to -WE (the write enable signal from the PC Card bus).
- WAIT# is a signal output from the S1D13806 that indicates the PC Card bus must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since PC Card bus accesses to the S1D13806 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13806 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Start (BS#) signal is not used for the PC Card Host Bus Interface and should be tied high (connected to V_{DD}).
- The RESET# (active low) input of the S1D13806 may be connected to the PC Card RESET (active high) using an inverter.

1.4 PC Card to S1D13806 Interface

1.4.1 Hardware Description

The S1D13806 is designed to directly support a variety of CPUs, providing an interface to the "local bus" of each processor. However, in order to provide support for processors not having an appropriate local bus, the S1D13806 supports a specific PC Card interface.

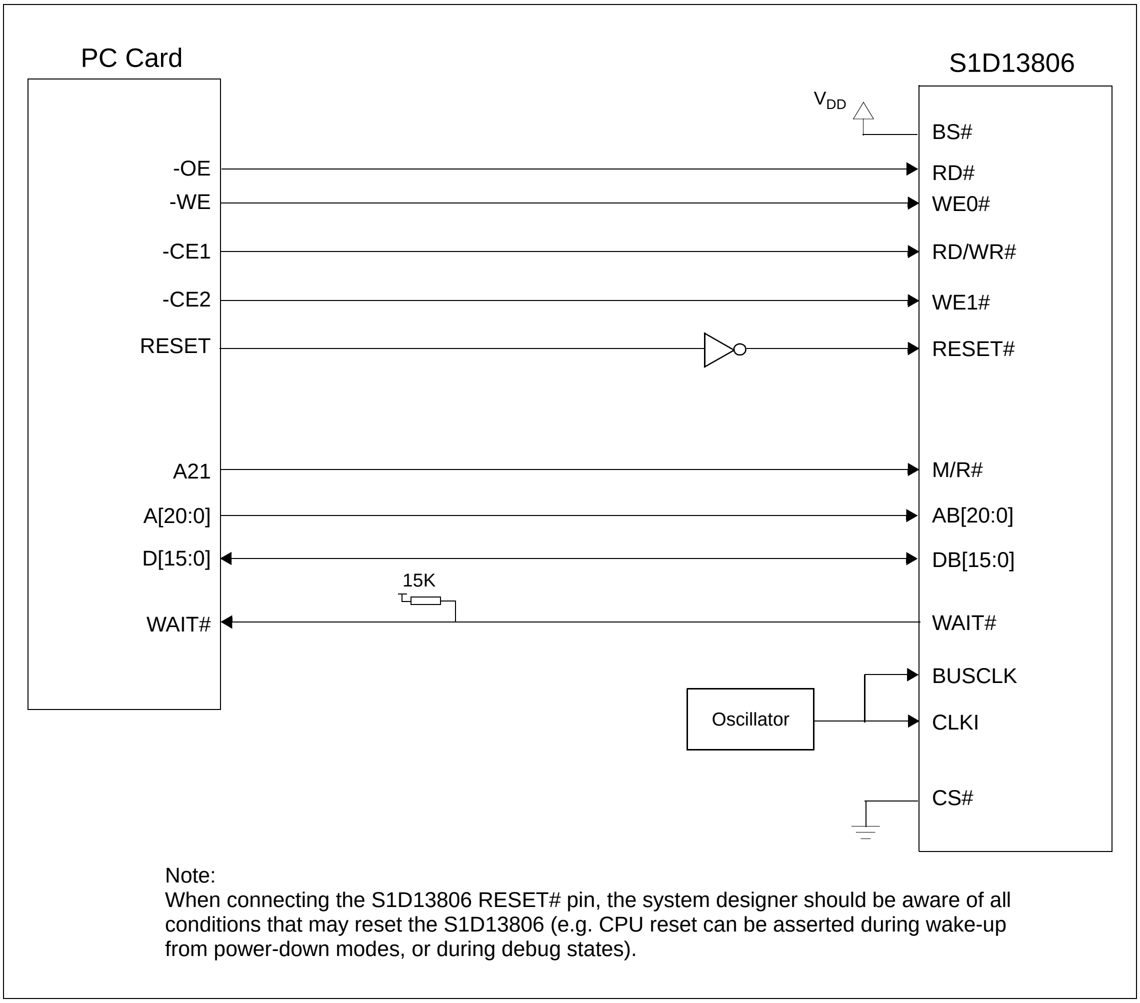
The S1D13806 provides a "glueless" interface to the PC Card bus except for the following.

- The RESET# signal on the S1D13806 is active low and must be inverted to support the active high RESET provided by the PC Card interface.
- Although the S1D13806 supports an asynchronous bus interface, a clock source is required on the BUSCLK input pin.

In this implementation, the address bus (AB[20:0]) and data bus (DB[15:0]) connect directly to the CPU address (A[20:0]) and data bus (D[15:0]). M/R# is treated as an address line so that it can be controlled using system address A21.

The PC Card interface does not provide a bus clock, so one must be supplied for the S1D13806. Since the bus clock frequency is not critical, nor does it have to be synchronous to the bus signals, it may be the same as CLKI. BS# (bus start) is not used and should be tied high (connected to V_{DD}).

The following shows a typical implementation of the PC Card to S1D13806 interface.



Note:
When connecting the S1D13806 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13806 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 1-3 Typical Implementation of PC Card to S1D13806 Interface

1.4.2 S1D13806 Hardware Configuration

The S1D13806 latches CONF7 through CONF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx.

The table below shows only those configuration settings important to the PC Card Host Bus Interface.

Table 1-2 Summary of Power-On/Reset Options

S1D13806 Pin Name	value on this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
CONF[3:0]	1001 = PC Card Host Bus Interface; Little Endian; Active Low WAIT# selected	
CONF4	LCDPWR Active Low	LCDPWR Active High
CONF5	BUSCLK input divided by two	BUSCLK input not divided by two
CONF6	WAIT# is always driven	WAIT# is floating if S1D13806 is not selected
CONF7	Configure GPIO12 as MediaPlug output pin VMPEPWR	Configure GPIO12 for normal use

▨ = configuration for PC Card Host Bus Interface

1.4.3 Performance

The S1D13806 PC Card Interface specification supports a BCLK up to 50MHz, and therefore can provide a high performance display solution.

1.4.4 Register/Memory Mapping

The S1D13806 is a memory-mapped device. The internal registers are mapped in the lower PC Card memory address space starting at zero. The display buffer requires 1.25M bytes and is mapped in the third and fourth megabytes of the PC Card address space (ranging from 20 0000h to 33 FFFFh).

A typical implementation as shown in Figure 1-3 "Typical Implementation of PC Card to S1D13806 Interface" on page 8 has Chip Select (CS#) connected to ground (always enabled) and the Memory/ Register select pin (M/R#) connected to address bit A21. This implementation decodes as shown in the following table.

Table 1-3 Register/Memory Mapping for Typical Implementation

A21 (M/R#)	A20	A12	Address Range	Function
0	0	0	0 to 1FFh	Control Registers Decoded
0	0	1	1000h to 1FFFh	MediaPlug Registers Decoded
0	1	x	10 0000h to 1F FFFh	BitBLT Registers Decoded
1	x	x	20 0000h to 33 FFFFh	Display Buffer Decoded

x = don't care

The PC Card socket provides 64M byte of address space. Since the PC Card address bits A[25:22] are ignored, the S1D13806 registers and display buffer are aliased within the allocated address space. If aliasing is undesirable, the address space must be fully decoded.

2 INTERFACING TO THE NEC VR4102/ VR4111TM MICROPROCESSORS

2.1 Introduction

This application note describes the hardware and software environment necessary to provide an interface between the S1D13806 Embedded Memory Display Controller and the NEC VR4102TM (*µ*PD30102) or VR4111TM (*µ*PD30111) Microprocessors.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the latest revision of this document before beginning any development.

2.2 Interfacing to the VR4102/VR4111

2.2.1 The NEC VR4102/VR4111 System Bus

The VR-Series family of microprocessors features a high-speed synchronous system bus typical of modern microprocessors. Designed with external LCD controller support and Windows CE based embedded consumer applications in mind, the VR4102/VR4111 offers a highly integrated solution for portable systems. This section provides an overview of the operation of the CPU bus in order to establish interface requirements.

Overview

The NEC VR4102/VR4111 is designed around the RISC architecture developed by MIPS. This microprocessor is based on the 66MHz VR4100 CPU core which supports 64-bit processing. The CPU communicates with the Bus Control Unit (BCU) using its internal SysAD bus. The BCU in turn communicates with external devices using its ADD and DAT buses which can be dynamically sized for 16 or 32-bit operation.

The NEC VR4102/VR4111 has direct support for an external LCD controller. Specific control signals are assigned for an external LCD controller providing an easy interface to the CPU. A 16M byte block of memory is assigned for the LCD controller and its own chip select and ready signals are available. Word or byte accesses are controlled by the system high byte signal (SHB#).

LCD Memory Access Cycles

Once an address in the LCD block of memory is placed on the external address bus (ADD[25:0]), the LCD chip select (LCDCS#) is driven low. The read or write enable signals (RD# or WR#) are driven low for the appropriate cycle and LCDRDY is driven low to insert wait states into the cycle. The high byte enable (SHB#) in conjunction with address bit 0 allows for byte steering.

The following figure illustrates typical NEC VR4102/VR4111 memory read and write cycles to the LCD controller interface.

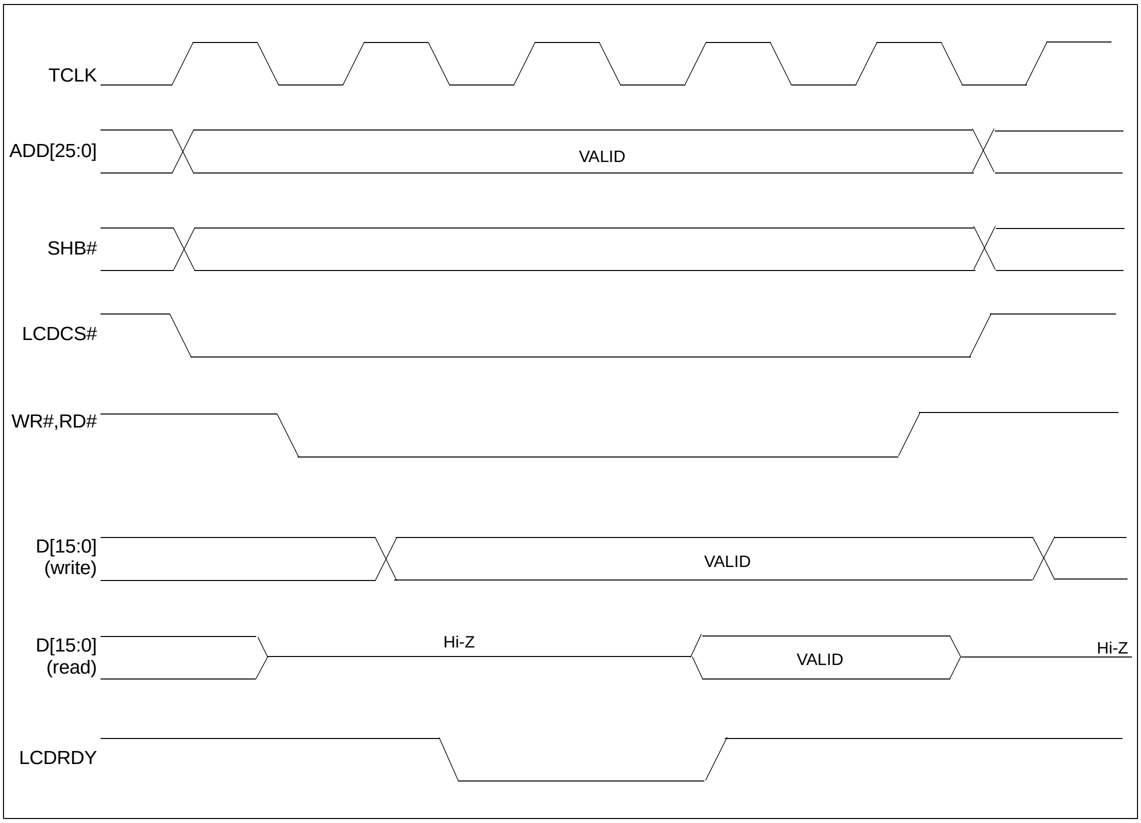


Figure 2-1 NEC VR4102/VR4111 Read/Write Cycles

2.3 S1D13806 Host Bus Interface

The S1D13806 directly supports multiple processors. The S1D13806 implements a 16-bit MIPS/ ISA Host Bus Interface which is most suitable for direct connection to the VR4102/VR4111 microprocessor.

The MIPS/ISA Host Bus Interface is selected by the S1D13806 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13806 configuration, see Section 2.4.2, “S1D13806 Hardware Configuration” on page 17.

Note: At reset, the Register/Memory Select bit in the Miscellaneous Register (REG[001h] bit 7) is set to 1. This means that only REG[000h] (read-only) and REG[001h] are accessible **until a write to REG[001h] sets bit 7 to 0 making all registers accessible**. When debugging a new hardware design, this can sometimes give the appearance that the interface is not working, so it is important to remember to clear this bit before proceeding with debugging.

2.3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 2-1 Host Bus Interface Pin Mapping

S1D13806 Pin Name	NEC VR4102/VR4111 Pin Name
AB[20:0]	ADD[20:0]
DB[15:0]	DAT[15:0]
WE1#	SHB#
M/R#	ADD21
CS#	LCDCS#
BUSCLK	BUSCLK
BS#	V_{DD}
RD/WR#	V_{DD}
RD#	RD#
WE0#	WR#
WAIT#	LCDRDY
RESET#	connected to system reset

2.3.2 *Host Bus Interface Signal Descriptions*

The S1D13806 MIPS/ISA Host Bus Interface requires the following signals.

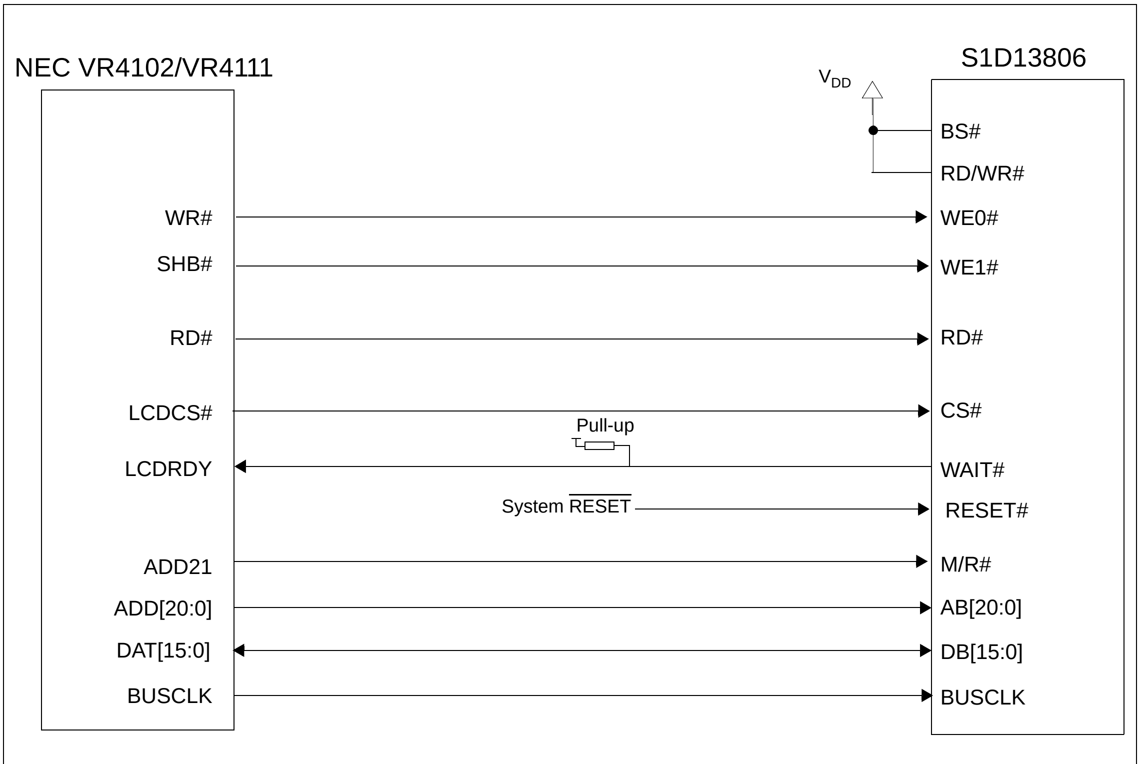
- BUSCLK is a clock input which is required by the S1D13806 Host Bus Interface. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB[20:0], and the data bus DB[15:0], connect directly to the VR4102/VR4111 address (ADD[20:0]) and data bus (DAT[15:0]), respectively. CONF[3:0] must be set to select the MIPS/ISA Host Bus Interface with little endian mode.
- M/R# (memory/register) selects between memory or register access. It may be connected to an address line, allowing system address ADD21 to be connected to the M/R# line.
- Chip Select (CS#) must be driven low by LCDCS# whenever the S1D13806 is accessed by the VR4102/VR4111.
- WE1# connects to SHB# (the high byte enable signal from the VR4102/VR4111) which in conjunction with address bit 0 allows byte steering of read and write operations.
- WE0# connects to WR# (the write enable signal from the VR4102/VR4111) and must be driven low when the VR4102/VR4111 is writing data to the S1D13806.
- RD# connects to RD# (the read enable signal from the VR4102/VR4111) and must be driven low when the VR4102/VR4111 is reading data from the S1D13806.
- WAIT# connects to LCDRDY and is a signal output from the S1D13806 that indicates the VR4102/VR4111 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since VR4102/VR4111 accesses to the S1D13806 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13806 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The BS# and RD/WR# signals are not used for the MIPS/ISA Host Bus Interface and should be tied high (connected to V_{DD}).

2.4 VR4102/VR4111 to S1D13806 Interface

2.4.1 Hardware Description

The NEC VR4102/VR4111 Microprocessors are specifically designed to support an external LCD controller. They provide the necessary internal address decoding and control signals.

The diagram below shows a typical implementation utilizing the S1D13806.



Note:
When connecting the S1D13806 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13806 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 2-2 Typical Implementation of NEC VR4102/VR4111 to S1D13806 Interface

Note: For pin mapping, see Table 1-1, “PC Card Host Bus Interface Pin Mapping,” on page 5.

2.4.2 S1D13806 Hardware Configuration

The S1D13806 latches CONF7 through CONF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx.

The table below shows the configuration settings important to the MIPS/ISA Host Bus Interface used by the NEC VR4102/VR4111 microprocessor.

Table 2-2 Summary of Power-On/Reset Options

S1D13806 Pin Name	state of this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
CONF[3:0]	0100 = MIPS/ISA Host Bus Interface; Little Endian; Active Low WAIT# selected	
CONF4	LCDPWR Active Low	LCDPWR Active High
CONF5	BUSCLK input divided by two	BUSCLK input not divided by two
CONF6	WAIT# is always driven	WAIT# is floating if S1D13806 is not selected
CONF7	Configure GPIO12 as MediaPlug output pin VMPEPWR	Configure GPIO12 for normal use

▨ = configuration for NEC VR4102/VR4111 microprocessor

2.4.3 NEC VR4102/VR4111 Configuration

In the NEC VR4102/VR4111 register BCUCNTREG1, the ISAM/LCD bit must be set to 0. A 0 indicates that the reserved address space is used by the external LCD controller rather than the high-speed ISA memory.

In the register BCUCNTREG2, the GMODE bit must be set to 1 indicating that LCD controller accesses use a non-inverting data bus.

In the VR4111 register BCUCNTREG3, the LCD32/ISA32 bit must be set to 0. This sets the LCD interface to operate using a 16-bit data bus.

Note: Setting the LCD32/ISA32 bit to 0 in the BCUCNTREG3 register affects both the LCD controller and high-speed ISA memory access.

The frequency of BUSCLK output is programmed from the state of pins TxD/CLKSEL2, RTS#/CLKSEL1 and DTR#/CLKSEL0 during reset, and from the PMU (Power Management Unit) configuration registers of the NEC VR4102/VR4111. The S1D13806 works at any of the frequencies provided by the NEC VR4102/VR4111.

2.4.4 Register/Memory Mapping

The NEC VR4102/VR4111 provides the internal address decoding necessary to map an external LCD controller. Physical address 0A00 0000h to 0AFF FFFFh (16M bytes) is reserved for an external LCD controller such as the S1D13806.

The S1D13806 is a memory-mapped device. The internal registers are mapped in the lower address space starting at zero. The display buffer requires 1.25M bytes and is mapped in the third and fourth megabytes (0A20 0000h to 0A33 FFFFh).

A typical implementation as shown in Figure 2-2 “Typical Implementation of NEC VR4102/VR4111 to S1D13806 Interface” on page 16 has the Memory/Register select pin (M/R#) connected to NEC VR4102/VR4111 address line ADD21. ADD21 selects between the S1D13806 display buffer (ADD21=1) and internal registers (ADD21=0). This implementation decodes as shown in the following table.

Table 2-3 Register/Memory Mapping for Typical Implementation

ADD21 (M/R#)	ADD20	ADD12	Physical Address Range	Function
0	0	0	0A00 0000h to 0A00 01FFh	Control Registers Decoded
0	0	1	0A00 1000h to 0A00 1FFFh	MediaPlug Registers Decoded
0	1	x	0A10 0000h to 0A1F FFFFh	BitBLT Registers Decoded
1	x	x	0A20 0000h to 0A33 FFFFh	Display Buffer Decoded

x = don’t care

The NEC VR4102/VR4111 provides 16M byte of address space. Since the NEC VR4102/VR4111 address bits ADD[25:22] are ignored, the S1D13806 registers and display buffer are aliased within the allocated address space. If aliasing is undesirable, the address space must be fully decoded.

3 INTERFACING TO THE MOTOROLA MPC821 MICROPROCESSOR

3.1 Introduction

This application note describes the hardware and software environment required to provide an interface between the S1D13806 Embedded Memory Display Controller and the Motorola MPC821 processor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the latest revision of this document before beginning any development.

3.2 Interfacing to the MPC821

3.2.1 The MPC8xx System Bus

The MPC8xx family of processors feature a high-speed synchronous system bus typical of RISC microprocessors. This section provides an overview of the operation of the CPU bus in order to establish interface requirements.

3.2.2 MPC821 Bus Overview

The MPC8xx microprocessor family uses a synchronous address and data bus. All IO is synchronous to a square-wave reference clock called CLKOUT. This clock runs at the machine cycle speed of the CPU core (typically 25 to 50MHz). Most outputs from the processor change state on the rising edge of this clock. Similarly, most inputs to the processor are sampled on the rising edge.

Note: The external bus can be run at one-half the CPU core speed using the clock control register. This is typically done when the CPU core is operated above 50MHz.

The MPC821 can generate up to eight independent chip select outputs, each of which may be controlled by one of two types of timing generators: the General Purpose Chip Select Module (GPCM) or the User-Programmable Machine (UPM). Examples are given using the GPCM.

All Power PC microprocessors, including the MPC8xx family, use bit notation which is opposite from the convention used by most other microprocessor systems. Bit numbering for the MPC8xx always starts with zero as the most significant bit, and increments in value to the least-significant bit. For example, the most significant bits of the address bus and data bus are A0 and D0, while the least significant bits are A31 and D31.

The MPC8xx uses both a 32-bit address and data bus. A parity bit is supported for each of the four byte lanes on the data bus. Parity checking is done when data is read from external memory or peripherals, and generated by the MPC8xx bus controller on write cycles. There is no separate IO space in the Power PC architecture because all IO accesses are memory-mapped.

The bus can support both normal and burst cycles. Burst memory cycles are used to fill on-chip cache memory and for certain on-chip DMA operations. Normal cycles are used for all other data transfers.

Normal (Non-Burst) Bus Transactions

The bus master initiates a data transfer by placing the memory address on address lines A0 through A31 and driving $\overline{\text{TS}}$ (Transfer Start) low for one clock cycle. Several control signals are provided with the memory address:

- TSIZ[0:1] (Transfer Size) – indicates bus cycle size (8, 16, or 32-bit).
- RD/$\overline{\text{WR}}$ – set high for read cycles and low for write cycles.
- AT[0:3] (Address Type Signals) – provides detail on the type of transfer being attempted.

When the peripheral device being accessed completes its bus transfer, it asserts $\overline{\text{TA}}$ (Transfer Acknowledge) for one clock cycle to complete the bus transaction. Once $\overline{\text{TA}}$ has been asserted, the MPC821 doesn't start another bus cycle until $\overline{\text{TA}}$ is de-asserted. The minimum length of a bus transaction is two bus clocks.

Figure 3-1 "Power PC Memory Read Cycle" on page 21 illustrates a typical memory read cycle on the Power PC system bus.

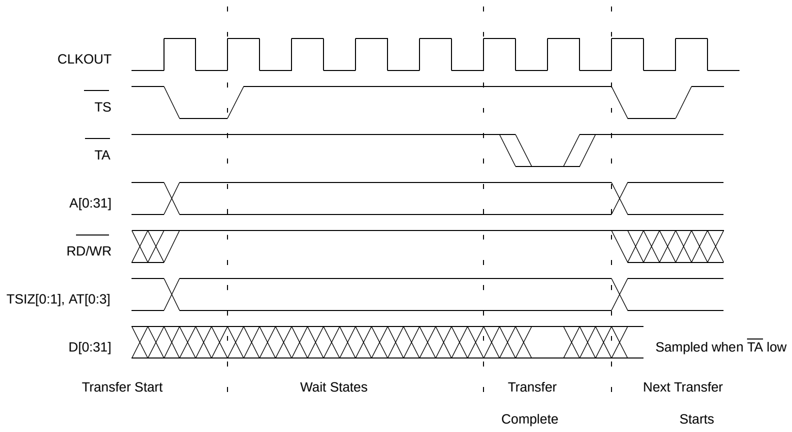


Figure 3-1 Power PC Memory Read Cycle

Figure 3-2 “Power PC Memory Write Cycle” on page 22 illustrates a typical memory write cycle on the Power PC system bus.

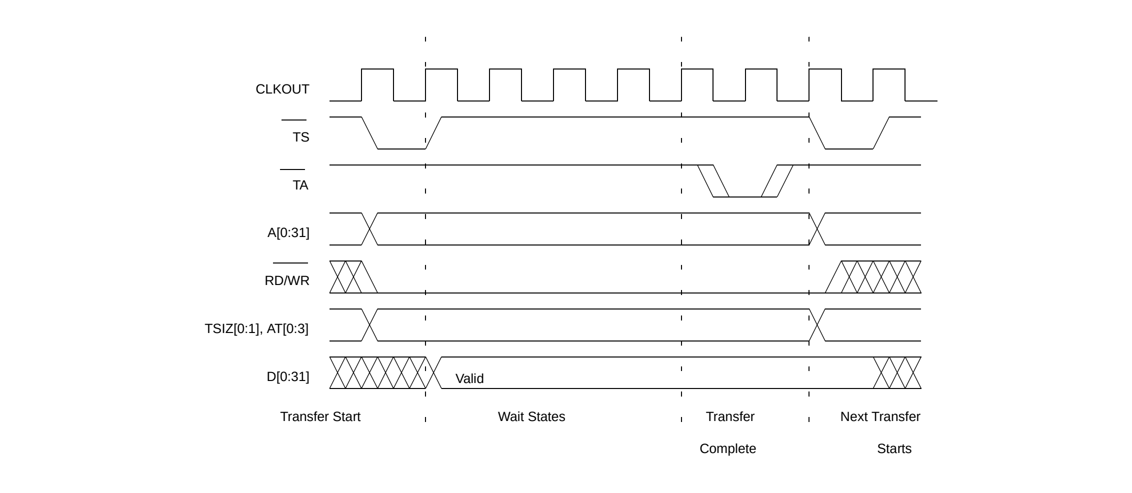


Figure 3-2 Power PC Memory Write Cycle

If an error occurs, $\overline{\text{TEA}}$ (Transfer Error Acknowledge) is asserted and the bus cycle is aborted. For example, a peripheral device may assert $\overline{\text{TEA}}$ if a parity error is detected, or the MPC821 bus controller may assert $\overline{\text{TEA}}$ if no peripheral device responds at the addressed memory location within a bus time-out period.

For 32-bit transfers, all data lines (D[0:31]) are used and the two low-order address lines A30 and A31 are ignored. For 16-bit transfers, data lines D[0:15] are used and address line A30 is ignored. For 8-bit transfers, data lines D[0:7] are used and all address lines (A[0:31]) are used.

Note: This assumes that the Power PC core is operating in big endian mode (typically the case for embedded systems).

Burst Cycles

Burst memory cycles are used to fill on-chip cache memory and to carry out certain on-chip DMA operations. They are very similar to normal bus cycles with the following exceptions:

- Always 32-bit.
- Always attempt to transfer four 32-bit words sequentially.
- Always address longword-aligned memory (i.e. A30 and A31 are always 0:0).
- Do not increment address bits A28 and A29 between successive transfers; the addressed device must increment these address bits internally.

If a peripheral is not capable of supporting burst cycles, it can assert Burst Inhibit ($\overline{\text{BI}}$) simultaneously with $\overline{\text{TA}}$ and the processor reverts to normal bus cycles for the remaining data transfers.

Burst cycles are mainly intended to facilitate cache line fills from program or data memory. They are normally not used for transfers to/from IO peripheral devices such as the S1D13806, therefore the interfaces described in this document do not attempt to support burst cycles. However, the example interfaces include circuitry to detect the assertion of $\overline{\text{BDIP}}$ and respond with $\overline{\text{BI}}$ if caching is accidently enabled for the S1D13806 address space.

3.2.3 Memory Controller Module

General-Purpose Chip Select Module (GPCM)

The General-Purpose Chip Select Module (GPCM) is used to control memory and peripheral devices which do not require special timing or address multiplexing. In addition to the chip select output, it can generate active-low Output Enable ($\overline{\text{OE}}$) and Write Enable ($\overline{\text{WE}}$) signals compatible with most memory and x86-style peripherals. The MPC821 bus controller also provides a Read/ Write (RD/$\overline{\text{WR}}$) signal which is compatible with most 68K peripherals.

The GPCM is controlled by the values programmed into the Base Register (BR) and Option Register (OR) of the respective chip select. The Option Register sets the base address, the block size of the chip select, and controls the following timing parameters:

- The ACS bit field allows the chip select assertion to be delayed by 0, ¼, or ½ clock cycle with respect to the address bus valid.
- The CSNT bit causes chip select and $\overline{\text{WE}}$ to be negated ½ clock cycle earlier than normal.
- The TRLX (relaxed timing) bit will insert an additional one clock delay between assertion of the address bus and chip select. This accommodates memory and peripherals with long setup times.
- The EHTR (Extended hold time) bit will insert an additional 1 clock delay on the first access to a chip select.
- Up to 15 wait states may be inserted, or the peripheral can terminate the bus cycle itself by asserting $\overline{\text{TA}}$ (Transfer Acknowledge).
- Any chip select may be programmed to assert $\overline{\text{BI}}$ (Burst Inhibit) automatically when its memory space is addressed by the processor core.

User-Programmable Machine (UPM)

The UPM is typically used to control memory types, such as Dynamic RAMs, which have complex control or address multiplexing requirements. The UPM is a general purpose RAM-based pattern generator which can control address multiplexing, wait state generation, and five general-purpose output lines on the MPC821. Up to 64 pattern locations are available, each 32 bits wide. Separate patterns may be programmed for normal accesses, burst accesses, refresh (timer) events, and exception conditions. This flexibility allows almost any type of memory or peripheral device to be accommodated by the MPC821.

In this application note, the GPCM is used instead of the UPM, since the GPCM has enough flexibility to accommodate the S1D13806 and it is desirable to leave the UPM free to handle other interfacing duties, such as EDO DRAM.

3.3 S1D13806 Host Bus Interface

The S1D13806 implements a 16-bit native PowerPC host bus interface which is used to interface to the MPC821 microprocessor.

The PowerPC host bus interface is selected by the S1D13806 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13806 configuration, see Section 3.4.3, “S1D13806 Hardware Configuration” on page 30.

Note: At reset, the Register/Memory Select bit in the Miscellaneous Register (REG[001h] bit 7) is set to 1. This means that only REG[000h] (read-only) and REG[001h] are accessible **until a write to REG[001h] sets bit 7 to 0 making all registers accessible**. When debugging a new hardware design, this can sometimes give the appearance that the interface is not working, so it is important to remember to clear this bit before proceeding with debugging.

3.3.1 PowerPC Host Bus Interface Pin Mapping

The following table shows the functions of each host bus interface signal.

Table 3-1 PowerPC Host Bus Interface Pin Mapping

S1D13806 Pin Names	PowerPC
AB[20:0]	A[11:31]
DB[15:0]	D[0:15]
WE1#	$\overline{\text{BI}}$
M/R#	A10
CS#	PowerPC Internal Chip Select
BUSCLK	CLKOUT
BS#	$\overline{\text{TS}}$
RD/WR#	$\overline{\text{RD/WR}}$
RD#	TSIZ0
WE0#	TSIZ1
WAIT#	$\overline{\text{TA}}$
RESET#	RESET#

3.3.2 PowerPC Host Bus Interface Signals

The S1D13806 PowerPC host bus interface is designed to support processors which interface the S1D13806 through the PowerPC bus.

The S1D13806 PowerPC host bus interface requires the following signals:

- BUSCLK is a clock input which is required by the S1D13806 host bus interface. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB[20:0], and the data bus DB[15:0], connect directly to the PowerPC bus address (A[11:31]) and data bus (D[0:15]), respectively. CONF[3:0] must be set to select the PowerPC Host Bus Interface with big endian mode.
- M/R# (memory/register) selects between memory or register access. It may be connected to an address line, allowing the PowerPC bus address A10 to be connected to the M/R# line.
- Chip Select (CS#) must be driven low whenever the S1D13806 is accessed by the PowerPC bus.
- RD/WR# connects to $\overline{\mathrm{RD/WR}}$ which indicates whether a read or a write access is being performed on the S1D13806.
- WE1# connects to $\overline{\mathrm{BI}}$ (burst inhibit signal). WE# is output by the S1D13806 to indicate whether the S1D13806 is able to perform burst accesses.
- WE0# and RD# connect to TSIZ1 and TSIZ0 (high and low byte enable signals). These signals must be driven by the PowerPC bus to indicate the size of the transfer taking place on the bus.
- WAIT# connects to $\overline{\mathrm{TA}}$ and is output from the S1D13806 to indicate the PowerPC bus must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since PowerPC bus accesses to the S1D13806 may occur asynchronously to the display update, it is possible that contention may occur while accessing the S1D13806 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until resource arbitration is complete.
- The Bus Start (BS#) signal connects to $\overline{\mathrm{TS}}$ (the transfer start signal).

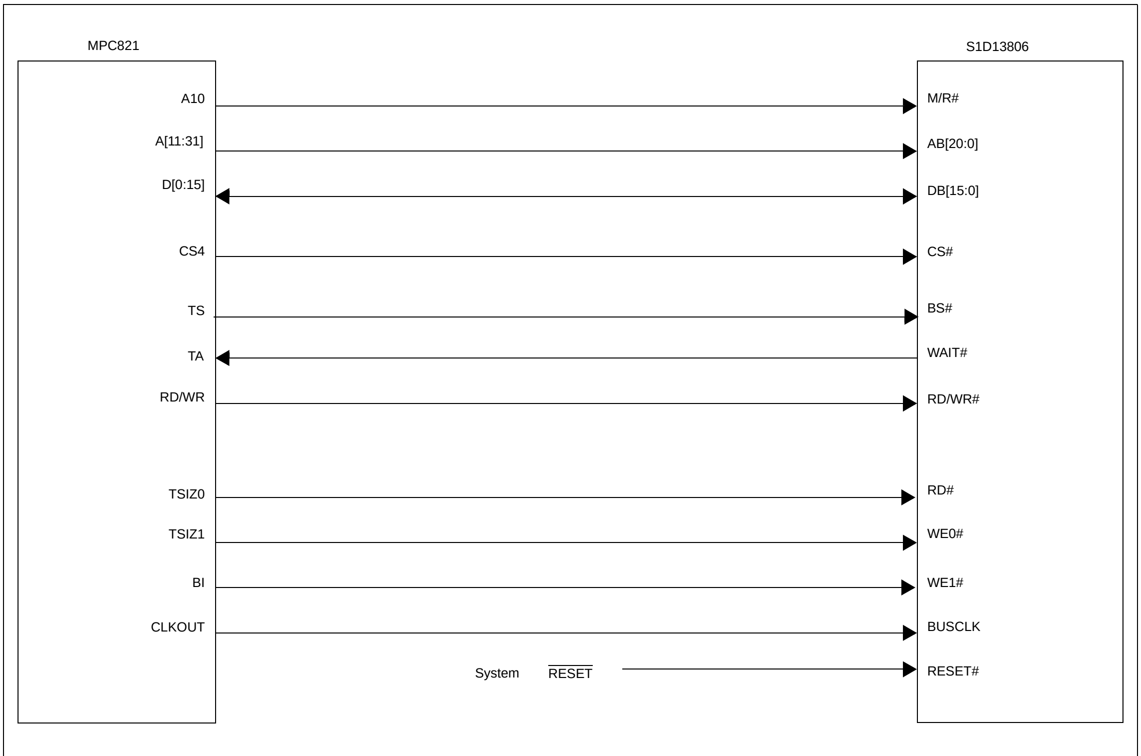
3.4 MPC821 to S1D13806 Interface

3.4.1 Hardware Description

The S1D13806 provides native PowerPC bus support, making it very simple to interface the two devices. This application note describes the environment necessary to connect the S1D13806 to the MPC821 native system bus and the connections between the S5U13806P00C Evaluation Board and the Motorola MPC821 Application Development System (ADS).

The S1D13806, by implementing a dedicated display buffer, reduces system power consumption, improves image quality, and increases system performance as compared to the on-chip LCD controller of the MPC821.

The S1D13806, through the use of the MPC821 chip selects, shares the system bus with all other MPC821 peripherals. The following figure demonstrates a typical implementation of the S1D13806 to MPC821 interface.



Note: When connecting the S1D13806 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13806 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 3-3 Typical Implementation of MPC821 to S1D13806 Interface

Table 3-2, “List of Connections from MPC821ADS to S1D13806,” on page 28 shows the connections between the pins and signals of the MPC821 and the S1D13806.

Note: The interface was designed using a Motorola MPC821 Application Development System (ADS). The ADS board has 5 volt logic connected to the data bus, so the interface included two 74F245 octal buffers on D[0:15] between the ADS and the S1D13806. In a true 3 volt system, no buffering is necessary.

3.4.2 Hardware Connections

The following table details the connections between the pins and signals of the MPC821 and the S1D13806.

Table 3-2 List of Connections from MPC821ADS to S1D13806

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13806 Signal Name
V_{CC}	P6-A1, P6-B1	V_{CC}
A10	P6-C23	M/R#
A11	P6-A22	AB20
A12	P6-B22	AB19
A13	P6-C21	AB18
A14	P6-C20	AB17
A15	P6-D20	AB16
A16	P6-B24	AB15
A17	P6-C24	AB14
A18	P6-D23	AB13
A19	P6-D22	AB12
A20	P6-D19	AB11
A21	P6-A19	AB10
A22	P6-D28	AB9
A23	P6-A28	AB8
A24	P6-C27	AB7
A25	P6-A26	AB6
A26	P6-C26	AB5
A27	P6-A25	AB4
A28	P6-D26	AB3
A29	P6-B25	AB2
A30	P6-B19	AB1
A31	P6-D17	AB0
D0	P12-A9	DB15
D1	P12-C9	DB14
D2	P12-D9	DB13
D3	P12-A8	DB12
D4	P12-B8	DB11
D5	P12-D8	DB10
D6	P12-B7	DB9
D7	P12-C7	DB8

Table 3-2 List of Connections from MPC821ADS to S1D13806 (Continued)

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13806 Signal Name
D8	P12-A15	DB7
D9	P12-C15	DB6
D10	P12-D15	DB5
D11	P12-A14	DB4
D12	P12-B14	DB3
D13	P12-D14	DB2
D14	P12-B13	DB1
D15	P12-C13	DB0
SRESET	P9-D15	RESET#
SYSCLK	P9-C2	BUSCLK
CS4	P6-D13	CS#
TS	P6-B7	BS#
TA	P6-B6	WAIT#
R/W	P6-D8	RD/WR#
TSIZ0	P6-B18	RD#
TSIZ1	P6-C18	WE0#
BI	P6-B9	WE1#
Gnd	P12-A1, P12-B1, P12-A2, P12-B2, P12-A3, P12-B3, P12-A4, P12-B4, P12-A5, P12-B5, P12-A6, P12-B6, P12-A7	Vss

Note: Note that the bit numbering of the Power PC bus signals is reversed. e.g. the most significant address bit is A0, the next is A1, A2, etc.

3.4.3 S1D13806 Hardware Configuration

The S1D13806 latches CONF7 through CONF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx.

The table below shows the configuration settings important to the PowerPC Host Bus Interface used by the MPC821 microprocessor.

Table 3-3 Summary of Power-On/Reset Options

S1D13806 Pin Name	state of this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
CONF[3:0]	1011 = MPC821 Host Bus Interface; Big Endian; Active High WAIT# selected	
CONF4	Reserved	
CONF5	BUSCLK input divided by 2	BUSCLK input not divided
CONF6	WAIT# is always driven	WAIT# is floating if S1D13806 is not selected
CONF7	Configure GPIO12 as MediaPlug output pin VMPEPWR and enable MediaPlug functionality	Configure GPIO12 for normal use and disable MediaPlug functionality

▨ = configuration for MPC821 microprocessor

3.4.4 Register/Memory Mapping

The DRAM on the MPC821 ADS board extends from address 0 through 3F FFFFh, so the S1D13806 is addressed starting at 40 0000h. A total of 4M bytes of address space is used, The S1D13806 is a memory-mapped device. The internal registers are mapped in the lower address space starting at zero. The display buffer requires 1.25M bytes and is mapped in the third and fourth megabytes of allocated memory (60 0000h to 73 FFFFh).

A typical implementation as shown in Figure 3-3 "Typical Implementation of MPC821 to S1D13806 Interface" on page 27 has the Memory/Register select pin (M/R#) connected to MPC821 address line A10. A10 selects between the S1D13806 display buffer (A10 = 1) and internal registers (A10 = 0). This implementation decodes as shown in the following table.

Table 3-4 Register/Memory Mapping for Typical Implementation

A10 (M/R#)	A11	A19	Physical Address Range	Function
0	0	0	40 0000h to 40 01FFh	Control Registers Decoded
0	0	1	40 1000h to 40 1FFFh	MediaPlug Registers Decoded
0	1	x	50 0000h to 5F FFFFh	BitBLT Registers Decoded
1	x	x	60 0000h to 73 FFFFh	Display Buffer Decoded

x = don't care

3.4.5 MPC821 Chip Select Configuration

Chip select 4 is used to control the S1D13806. The following options are selected in the base address register (BR4):

- BA[0:16] = 0000 0000 0100 0000 0 – set starting address of S1D13806 to 40 0000h.
- AT[0:2] = 0 – ignore address type bits.
- PS[0:1] = 1:0 – memory port size is 16-bit.
- PARE = 0 – disable parity checking.
- WP = 0 – disable write protect.
- MS[0:1] = 0:0 – select General Purpose Chip Select module to control this chip select.
- V = 1 – set valid bit to enable chip select.

The following options were selected in the option register (OR4):

- AM[0:16] = 1111 1111 1100 0000 0 – mask all but upper 10 address bits; S1D13806 consumes 4M byte of address space.
- ATM[0:2] = 0 – ignore address type bits.
- CSNT = 0 – normal $\overline{\text{CS}}$/$\overline{\text{WE}}$ negation.
- ACS[0:1] = 1:1 – delay $\overline{\text{CS}}$ assertion by ½ clock cycle from address lines.
- BI = 0 – do not assert Burst Inhibit.
- SCY[0:3] = 0 – wait state selection; this field is ignored since external transfer acknowledge is used; see SETA below.
- SETA = 1 – the S1D13806 generates an external transfer acknowledge using the WAIT# line.
- TRLX = 0 – normal timing.
- EHTR = 0 – normal timing.

3.4.6 Test Software

The test software is very simple. It configures chip select 4 (CS4) on the MPC821 to map the S1D13806 to an unused 4M byte block of address space. Next, it loads the appropriate values into the option register for CS4 and writes the value 0 to the S1D13806 register REG[001h] to enable full S1D13806 memory/register decoding. Lastly, the software runs a tight loop that reads the S1D13806 Revision Code Register REG[000h]. This allows monitoring of the bus timing on a logic analyzer.

The following source code was entered into the memory of the MPC821ADS using the line-by-line assembler in MPC8BUG (the debugger provided with the ADS board). Once the program was executed on the ADS, a logic analyzer was used to verify operation of the interface hardware.

It is important to note that when the MPC821 comes out of reset, the on-chip caches and MMU are disabled. If the data cache is enabled, then the MMU must be set so that the S1D13806 memory block is tagged as non-cacheable. This ensures the MPC821 does not attempt to cache any data read from, or written to, the S1D13806 or its display buffer.

```
BR4         equ$120 ; CS4 base register
OR4         equ$124 ; CS4 option register
MemStart    equ$40  ; upper word of S1D13806 start address
DisableReg  equ$1   ; address of S1D13806 Disable Register
RevCodeReg  equ$0   ; address of Revision Code Register

Start       mfsprr1,IMMR; get base address of internal registers
            andis.r1,r1,$ffff; clear lower 16 bits to 0
            andis.r2,r0,0; clear r2
            orisr2,r2,MemStart; write base address
            orir2,r2,$0801; port size 16 bits; select GPCM; enable
            stwr2,BR4(r1); write value to base register
            andis.r2,r0,0; clear r2
            orisr2,r2,$ffc0; address mask – use upper 10 bits
            orir2,r2,$0608; normal CS negation; delay CS ½ clock;
                    ; no burst inhibit (1386 does this)
            stwr2,OR4(r1); write to option register
            andis.r1,r0,0; clear r1
            orisr1,r1,MemStart; point r1 to start of S1D13806 mem space
            stbr1,DisableReg(r1); write 0 to disable register
Loop        lbzr0,RevCodeReg(r1); read revision code into r1
            bLoop   ; branch forever

            end
```

Note: MPC8BUG does not support comments or symbolic equates; these have been added for clarity.

4 INTERFACING TO THE PHILIPS MIPS PR31500/PR31700 PROCESSOR

4.1 Introduction

This application note describes the hardware and software environment necessary to provide an interface between the S1D13806 Embedded Memory Display Controller and the Philips MIPS PR31500/PR31700 Processor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the latest revision of this document before beginning any development.

4.2 Interfacing to the PR31500/PR31700

The Philips MIPS PR31500/PR31700 processor supports up to two PC Card (PCMCIA) slots. It is through this Host Bus Interface that the S1D13806 connects to the PR31500/PR31700 processor.

The S1D13806 can be successfully interfaced using one of the following configurations:

- Direct connection to the PR31500/PR31700 (see Section 4.4, “Direct Connection to the Philips PR31500/PR31700” on page 37).
- System design using the ITE IT8368E PC Card/GPIO buffer chip (see Section 4.5, “System Design Using the IT8368E PC Card Buffer” on page 41).

4.3 S1D13806 Host Bus Interface

The S1D13806 implements a 16-bit Host Bus Interface specifically for interfacing to the PR31500/PR31700 microprocessor.

The PR31500/PR31700 Host Bus Interface is selected by the S1D13806 on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. For details on S1D13806 configuration, see Section 4.4.2, “S1D13806 Configuration” on page 39.

Note: At reset, the Register/Memory Select bit in the Miscellaneous Register (REG[001h] bit 7) is set to 1. This means that only REG[000h] (read-only) and REG[001h] are accessible **until a write to REG[001h] sets bit 7 to 0 making all registers accessible**. When debugging a new hardware design, this can sometimes give the appearance that the interface is not working, so it is important to remember to clear this bit before proceeding with debugging.

4.3.1 PR31500/PR31700 Host Bus Interface Pin Mapping

The following table shows the function of each Host Bus Interface signal.

Table 4-1 PR31500/PR31700 Host Bus Interface Pin Mapping

S1D13806 Pin Name	Philips PR31500/PR31700
AB20	ALE
AB19	/CARDREG
AB18	/CARDIORD
AB17	/CARDIOWR
AB[16:13]	V_{DD}
AB[12:0]	A[12:0]
DB[15:8]	D[23:16]
DB[7:0]	D[31:24]
WE1#	/CARDxCSH
M/R#	V_{DD}
CS#	V_{DD}
BUSCLK	DCLKOUT
BS#	V_{DD}
RD/WR#	/CARDxCSL
RD#	/RD
WE0#	/WE
WAIT#	/CARDxWAIT
RESET#	RESET#

4.3.2 PR31500/PR31700 Host Bus Interface Signals

When the S1D13806 is configured to operate with the PR31500/PR31700, the Host Bus Interface requires the following signals:

- BUSCLK is a clock input required by the S1D13806 Host Bus Interface. It is separate from the input clock (CLKI) and should be driven by the PR31500/PR31700 bus clock output DCLKOUT.
- Address input AB20 corresponds to the PR31500/PR31700 signal ALE (address latch enable) whose falling edge indicates that the most significant bits of the address are present on the multiplexed address bus (AB[12:0]).
- Address input AB19 should be connected to the PR31500/PR31700 signal /CARDREG. This signal is active when either IO or configuration space of the PR31500/PR31700 PC Card slot is being accessed.
- Address input AB18 should be connected to the TX3912 signal CARDIORD*. Either AB18 or the RD# input must be asserted for a read operation to take place. The presence of two pins which provide the same function, maps the S1D13806 to both IO and attribute space (see Section 4.4.3, “Memory Mapping and Aliasing” on page 40).
- Address input AB17 should be connected to the TX3912 signal CARDIOWR*. Either AB17 or the WE0# input must be asserted for a write operation to take place. The presence of two pins which provide the same function, maps the S1D13806 to both IO and attribute space (see Section 4.4.3, “Memory Mapping and Aliasing” on page 40).
- Address inputs AB[16:13] and control inputs M/R#, CS# and BS# must be tied to V_{DD} as they are not used in this interface mode.
- Address inputs AB[12:0], and the data bus DB[15:0], connect directly to the PR31500/PR31700 address and data bus, respectively. MD4 must be set to select the proper endian mode on reset (see Section 4.4.2, “S1D13806 Configuration” on page 39). **Because of the PR31500/PR31700 data bus naming convention and endian mode, S1D13806 DB[15:8] must be connected to PR31500/PR31700 D[23:16], and S1D13806 DB[7:0] must be connected to PR31500/PR31700 D[31:24].**
- Control inputs WE1# and RD/WR# should be connected to the PR31500/PR31700 signals /CARDxCSH and /CARDxCSL respectively for byte steering.
- Input RD# should be connected to the PR31500/PR31700 signal /RD. Either RD# or the AB18 input (/CARDIORD) must be asserted for a read operation to take place.
- Input WE0# should be connected to the PR31500/PR31700 signal /WR. Either WE0# or the AB17 input (/CARDIOWR) must be asserted for a write operation to take place.
- WAIT# is a signal output from the S1D13806 that indicates the host CPU must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since the host CPU accesses to the S1D13806 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13806 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.

4.4 Direct Connection to the Philips PR31500/PR31700

The S1D13806 was specifically designed to support the Philips MIPS PR31500/PR31700 processor. When configured, the S1D13806 will utilize one of the PC Card slots supported by the processor.

4.4.1 Hardware Description

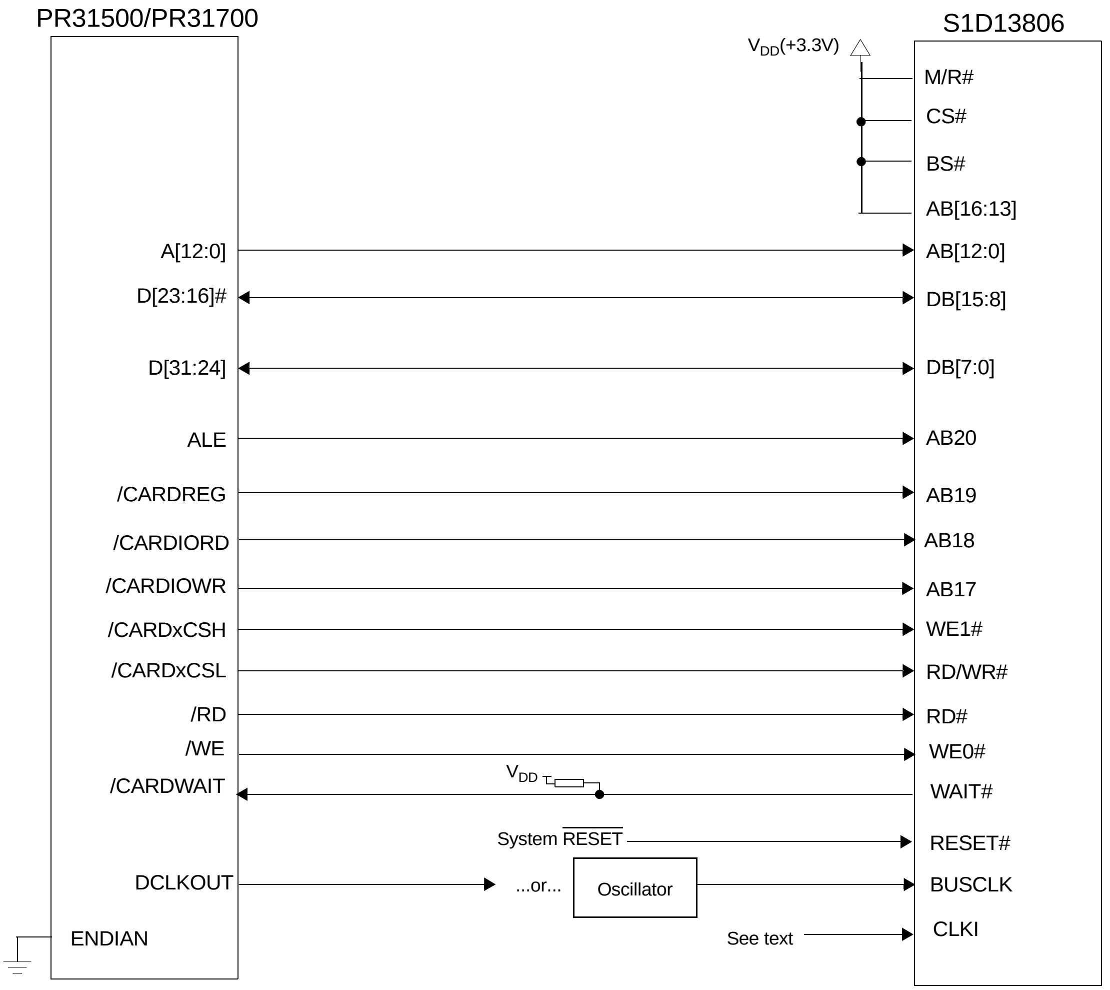
In this example implementation, the S1D13806 occupies one PC Card slot and resides in the Attribute and IO address range. The processor provides address bits A[12:0], with A[23:13] being multiplexed and available on the falling edge of ALE. Peripherals requiring more than 8K bytes of address space would require an external latch for these multiplexed bits. However, the S1D13806 has an internal latch specifically designed for this processor making additional logic unnecessary. To further reduce the need for external components, the S1D13806 has an optional BUSCLK divide-by-2 feature, allowing the high speed DCLKOUT from the processor to be directly connected to the BUSCLK input of the S1D13806. An optional external oscillator may be used for BUSCLK since the S1D13806 will accept host bus control signals asynchronously with respect to BUSCLK.

The host interface control signals of the S1D13806 are asynchronous with respect to the S1D13806 bus clock. This gives the system designer full flexibility to choose the appropriate source (or sources) for CLKI and BUSCLK. The choice of whether both clocks should be the same, whether to use DCLKOUT as clock source, and whether an external or internal clock divider is needed, should be based on the following criteria.

- pixel and frame rates.
- power budget.
- part count.
- maximum S1D13806 clock frequencies.

The S1D13806 also has internal CLKI dividers providing additional flexibility.

The following diagram shows a typical implementation of the interface.



Note:
When connecting the S1D13806 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13806 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 4-1 Typical Implementation of Direct Connection

4.4.2 S1D13806 Configuration

The S1D13806 latches CONF7 through CONF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the “*S1D13806 Hardware Functional Specification*,” document number X28B-A-001-xx.

The table below shows only those configuration settings important to the PR31500/PR31700 Host Bus Interface.

Table 4-2 Summary of Power-On/Reset Options

S1D13806 Pin Name	state of this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
CONF[3:0]	1000 = PR31500/PR31700 Host Bus Interface; Little Endian; Active Low WAIT# selected	
CONF4	LCDPWR Active Low	LCDPWR Active High
CONF5	BUSCLK input divided by two: use with DCLKOUT	BUSCLK input not divided: use with external oscillator
CONF6	WAIT# is always driven	WAIT# is floating if S1D13806 is not selected
CONF7	Configure GPIO12 as MediaPlug output pin VMPEPWR	Configure GPIO12 for normal use

▨ = configuration for PR31500/PR31700 Host Bus Interface

4.4.3 Memory Mapping and Aliasing

The PR31500/PR31700 uses a portion of the PC Card Attribute and IO space to access the S1D13806. The S1D13806 responds to both PC Card Attribute and IO bus accesses, thus freeing the programmer from having to set the PR31500/PR31700 Memory Configuration Register 3 bit CARD1IOEN (or CARD2IOEN if slot 2 is used). As a result, the PR31500/PR31700 sees the S1D13806 on its PC Card slot as described in the table below.

Table 4-3 PR31500/PR31700 to PC Card Slots Address Remapping for Direct Connection

S1D13806 Uses PC Card Slot #	Philips Address	Size	Function
1	0800 0000h	16M byte	Card 1 IO or Attribute
	0900 0000h	8M byte	S1D13806 registers, aliased 4 times at 2M byte intervals
	0980 0000h	8M byte	S1D13806 display buffer, aliased 4 times at 2M byte intervals
	0A00 0000h	32M byte	Card 1 IO or Attribute
	6400 0000h	64M byte	Card 1 Memory
2	0C00 0000h	16M byte	Card 2 IO or Attribute
	0D00 0000h	8M byte	S1D13806 registers, aliased 4 times at 2M byte intervals
	0D80 0000h	8M byte	S1D13806 display buffer, aliased 4 times at 2M byte intervals
	0E00 0000h	32M byte	Card 2 IO or Attribute
	6800 0000h	64M byte	Card 2 Memory

4.5 System Design Using the IT8368E PC Card Buffer

In a system design using one or two ITE IT8368E PC Card and multiple-function IO buffers, the S1D13806 can be interfaced so as to share one of the PC Card slots.

4.5.1 Hardware Description

The IT8368E can be programmed to allocate the same portion of the PC Card Attribute and IO space to the S1D13806 as in the direct connection implementation described in Section 4.4, “Direct Connection to the Philips PR31500/PR31700” on page 37.

Following is a block diagram showing an implementation using the IT8368E PC Card buffer.

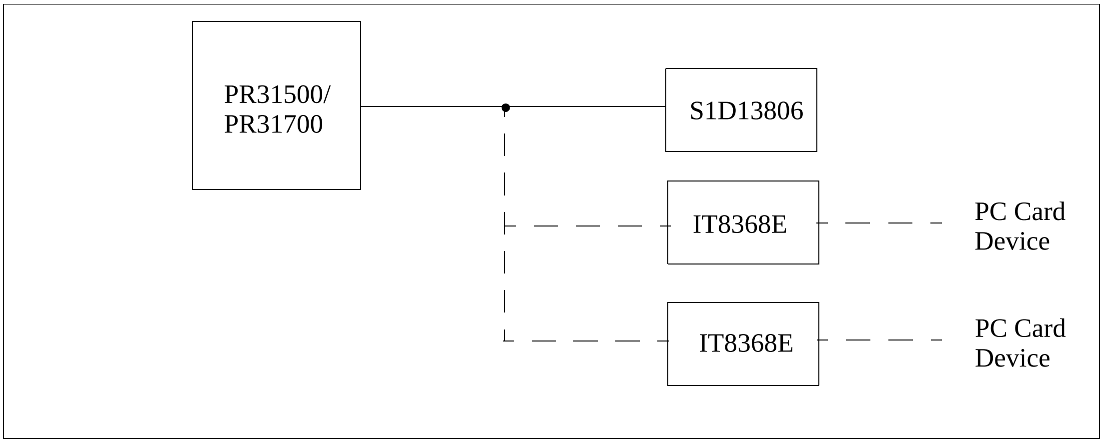


Figure 4-2 IT8368E Implementation Block Diagram

4.5.2 IT8368E Configuration

The ITE IT8368E is specifically designed to support EPSON LCD/CRT controllers. Older EPSON Controllers not supporting a direct interface to the Philips processor can utilize the IT8368E MFIO pins to provide the necessary control signals, however when using the S1D13806 this is not necessary as the Direct Connection described in Section 4.4, “Direct Connection to the Philips PR31500/PR31700” on page 37 can be used.

The IT8368E must have both “Fix Attribute/IO” and “VGA” modes enabled. When both these modes are enabled a 16M byte portion of the system PC Card attribute and IO space is allocated to address the S1D13806.

When the IT8368E senses that the S1D13806 is being accessed, it does not propagate the PC Card signals to its PC Card device. This makes S1D13806 accesses transparent to any PC Card device connected to the same slot.

For mapping details, refer to Section 4.4.3, “Memory Mapping and Aliasing” on page 40. For further information on configuring the IT8368E, refer to the “*IT8368E PC Card/GPIO Buffer Chip Specification*.”

4.5.3 S1D13806 Configuration

For S1D13806 configuration, refer to Section 4.4.2, “S1D13806 Configuration” on page 39.

5 INTERFACING TO THE TOSHIBA MIPS TX3912 PROCESSOR

5.1 Introduction

This application note describes the hardware and software environment necessary to provide an interface between the S1D13806 Embedded Memory Display Controller and the Toshiba MIPS TX3912 Processor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the latest revision of this document before beginning any development.

5.2 Interfacing to the TX3912

The Toshiba MIPS TX3912 processor supports up to two PC Card (PCMCIA) slots. It is through this Host Bus Interface that the S1D13806 connects to the TX3912 processor.

The S1D13806 can be successfully interfaced using one of the following configurations:

- Direct connection to the TX3912 (see Section 4.4, “Direct Connection to the Philips PR31500/ PR31700” on page 37).
- System design using the ITE IT8368E PC Card/GPIO buffer chip (see Section 4.5, “System Design Using the IT8368E PC Card Buffer” on page 41).

5.3 S1D13806 Host Bus Interface

The S1D13806 implements a 16-bit Host Bus Interface specifically for interfacing to the TX3912 microprocessor.

The TX3912 Host Bus Interface is selected by the S1D13806 on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. For details on S1D13806 configuration, see Section 4.4.2, "S1D13806 Configuration" on page 39.

Note: At reset, the Register/Memory Select bit in the Miscellaneous Register (REG[001h] bit 7) is set to 1. This means that only REG[000h] (read-only) and REG[001h] are accessible **until a write to REG[001h] sets bit 7 to 0 making all registers accessible**. When debugging a new hardware design, this can sometimes give the appearance that the interface is not working, so it is important to remember to clear this bit before proceeding with debugging.

5.3.1 TX3912 Host Bus Interface Pin Mapping

The following table shows the function of each Host Bus Interface signal.

Table 5-1 TX3912 Host Bus Interface Pin Mapping

S1D13806 Pin Names	Toshiba TX3912
AB20	ALE
AB19	CARDREG*
AB18	CARDIORD*
AB17	CARDIOWR*
AB[16:13]	V_{DD}
AB[12:0]	A[12:0]
DB[15:8]	D[23:16]
DB[7:0]	D[31:24]
WE1#	CARDxCSH*
M/R#	V_{DD}
CS#	V_{DD}
BUSCLK	DCLKOUT
BS#	V_{DD}
RD/WR#	CARDxCSL*
RD#	RD*
WE0#	WE*
WAIT#	CARDxWAIT*
RESET#	PON*

5.3.2 TX3912 Host Bus Interface Signals

When the S1D13806 is configured to operate with the TX3912, the Host Bus Interface requires the following signals:

- BUSCLK is a clock input required by the S1D13806 Host Bus Interface. It is separate from the input clock (CLKI) and should be driven by the TX3912 bus clock output DCLKOUT.
- Address input AB20 corresponds to the TX3912 signal ALE (address latch enable) whose falling edge indicates that the most significant bits of the address are present on the multiplexed address bus (AB[12:0]).
- Address input AB19 should be connected to the TX3912 signal CARDREG*. This signal is active when either IO or configuration space of the TX3912 PC Card slot is being accessed.
- Address input AB18 should be connected to the TX3912 signal CARDIORD*. Either AB18 or the RD# input must be asserted for a read operation to take place. The presence of two pins which provide the same function, maps the S1D13806 to both IO and attribute space (see Section 4.4.3, “Memory Mapping and Aliasing” on page 40).
- Address input AB17 should be connected to the TX3912 signal CARDIOWR*. Either AB17 or the WE0# input must be asserted for a write operation to take place. The presence of two pins which provide the same function, maps the S1D13806 to both IO and attribute space (see Section 4.4.3, “Memory Mapping and Aliasing” on page 40).
- Address inputs AB[16:13] and control inputs M/R#, CS# and BS# must be tied to V_{DD} as they are not used in this interface mode.
- Address inputs AB[12:0], and the data bus DB[15:0], connect directly to the TX3912 address and data bus, respectively. The S1D13806 supports the TX3912 in little endian mode only. **Because of the TX3912 data bus naming convention and endian mode, S1D13806 DB[15:8] must be connected to TX3912 D[23:16], and S1D13806 DB[7:0] must be connected to TX3912 D[31:24].**
- Control inputs WE1# and RD/WR# should be connected to the TX3912 signals CARDxCSH* and CARDxCSL* respectively for byte steering.
- Input RD# should be connected to the TX3912 signal RD*. Either RD# or the AB18 input (CARDIORD*) must be asserted for a read operation to take place.
- Input WE0# should be connected to the TX3912 signal WR*. Either WE0# or the AB17 input (CARDIOWR*) must be asserted for a write operation to take place.
- WAIT# is a signal output from the S1D13806 that indicates the TX3912 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since the TX3912 accesses to the S1D13806 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13806 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.

5.4 Direct Connection to the Toshiba TX3912

The S1D13806 was specifically designed to support the Toshiba MIPS TX3912 processor. When configured, the S1D13806 will utilize one of the PC Card slots supported by the processor.

5.4.1 Hardware Description

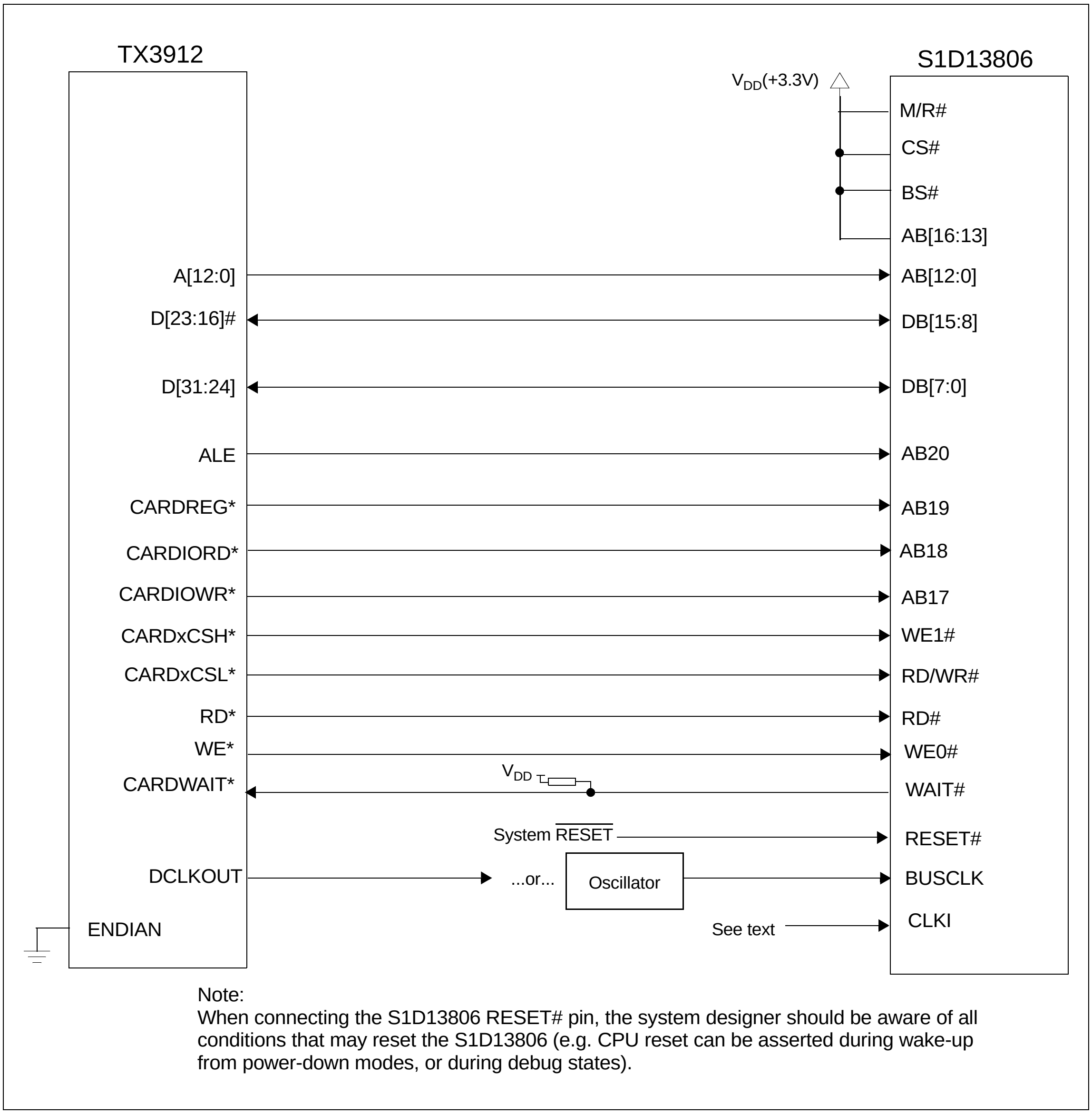
In this example implementation, the S1D13806 occupies one PC Card slot and resides in the Attribute and IO address range. The processor provides address bits A[12:0], with A[23:13] being multiplexed and available on the falling edge of ALE. Peripherals requiring more than 8K bytes of address space would require an external latch for these multiplexed bits. However, the S1D13806 has an internal latch specifically designed for this processor making additional logic unnecessary. To further reduce the need for external components, the S1D13806 has an optional BUSCLK divide-by-2 feature, allowing the high speed DCLKOUT from the processor to be directly connected to the BUSCLK input of the S1D13806. An optional external oscillator may be used for BUSCLK since the S1D13806 will accept host bus control signals asynchronously with respect to BUSCLK.

The host interface control signals of the S1D13806 are asynchronous with respect to the S1D13806 bus clock. This gives the system designer full flexibility to choose the appropriate source (or sources) for CLKI and BUSCLK. The choice of whether both clocks should be the same, whether to use DCLKOUT as clock source, and whether an external or internal clock divider is needed, should be based on the following criteria.

- pixel and frame rates.
- power budget.
- part count.
- maximum S1D13806 clock frequencies.

The S1D13806 also has internal CLKI dividers providing additional flexibility.

The following diagram shows a typical implementation of the interface.



Note:
When connecting the S1D13806 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13806 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 5-1 Typical Implementation of Direct Connection

5.4.2 S1D13806 Configuration

The S1D13806 latches CONF7 through CONF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx.

The table below shows the configuration settings important to the TX3912 Host Bus Interface.

Table 5-2 Summary of Power-On/Reset Options

S1D13806 Pin Name	state of this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
CONF[3:0]	1000 = TX3912 Host Bus Interface; Little Endian; Active Low WAIT# selected	
CONF4	LCDPWR Active Low	LCDPWR Active High
CONF5	BUSCLK input divided by two: use with DCLKOUT	BUSCLK input not divided: use with external oscillator
CONF6	WAIT# is always driven	WAIT# is floating if S1D13806 is not selected
CONF7	Configure GPIO12 as MediaPlug output pin VMPEPWR	Configure GPIO12 for normal use

▨ = configuration for TX3912 Host Bus Interface

5.4.3 *Memory Mapping and Aliasing*

The TX3912 uses a portion of the PC Card Attribute and IO space to access the S1D13806. The S1D13806 responds to both PC Card Attribute and IO bus accesses, thus freeing the programmer from having to set the TX3912 Memory Configuration Register 3 bit CARD1IOEN (or CARD2IOEN if slot 2 is used). As a result, the TX3912 sees the S1D13806 on its PC Card slot as described in the table below.

Table 5-3 TX3912 to PC Card Slots Address Remapping for Direct Connection

S1D13806 Uses PC Card Slot #	Toshiba Address	Size	Function
1	0800 0000h	16M byte	Card 1 IO or Attribute
	0900 0000h	8M byte	S1D13806 registers, aliased 4 times at 2M byte intervals
	0980 0000h	8M byte	S1D13806 display buffer, aliased 4 times at 2M byte intervals
	0A00 0000h	32M byte	Card 1 IO or Attribute
	6400 0000h	64M byte	Card 1 Memory
2	0C00 0000h	16M byte	Card 2 IO or Attribute
	0D00 0000h	8M byte	S1D13806 registers, aliased 4 times at 2M byte intervals
	0D80 0000h	8M byte	S1D13806 display buffer, aliased 4 times at 2M byte intervals
	0E00 0000h	32M byte	Card 2 IO or Attribute
	6800 0000h	64M byte	Card 2 Memory

5.5 System Design Using the IT8368E PC Card Buffer

In a system design using one or two ITE IT8368E PC Card and multiple-function IO buffers, the S1D13806 can be interfaced so as to share one of the PC Card slots.

5.5.1 Hardware Description

The IT8368E can be programmed to allocate the same portion of the PC Card Attribute and IO space to the S1D13806 as in the direct connection implementation described in Section 4.4, “Direct Connection to the Philips PR31500/PR31700” on page 37.

Following is a block diagram showing an implementation using the IT8368E PC Card buffer.

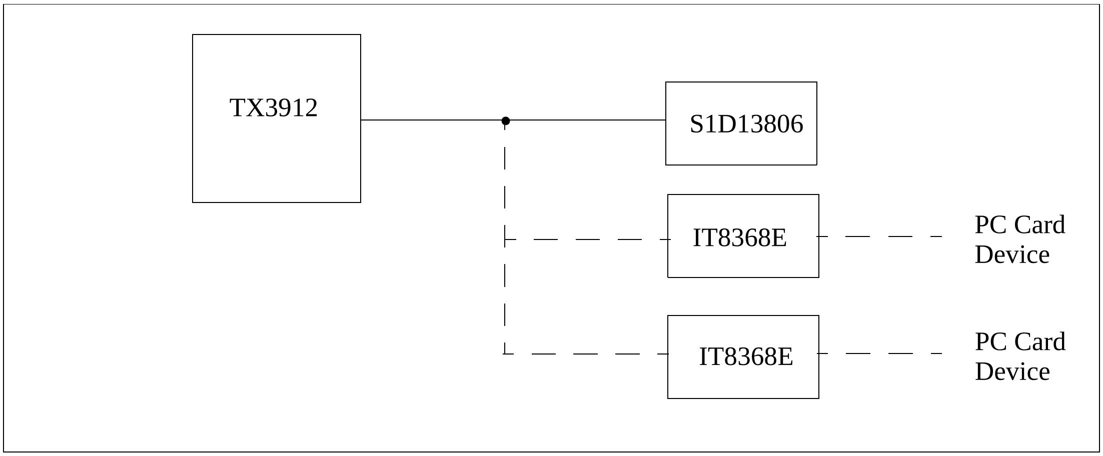


Figure 5-2 IT8368E Implementation Block Diagram

5.5.2 IT8368E Configuration

The ITE IT8368E is specifically designed to support EPSON LCD/CRT controllers. Older EPSON Controllers not supporting a direct interface to the Toshiba processor can utilize the IT8368E MFIO pins to provide the necessary control signals, however when using the S1D13806 this is not necessary as the Direct Connection described in Section 4.4, “Direct Connection to the Philips PR31500/ PR31700” on page 37 can be used.

The IT8368E must have both “Fix Attribute/IO” and “VGA” modes enabled. When both these modes are enabled a 16M byte portion of the system PC Card attribute and IO space is allocated to address the S1D13806.

When the IT8368E senses that the S1D13806 is being accessed, it does not propagate the PC Card signals to its PC Card device. This makes S1D13806 accesses transparent to any PC Card device connected to the same slot.

For mapping details, refer to Section 4.4.3, “Memory Mapping and Aliasing” on page 40. For further information on configuring the IT8368E, refer to the “*IT8368E PC Card/GPIO Buffer Chip Specification*.”

5.5.3 S1D13806 Configuration

For S1D13806 configuration, refer to Section 4.4.2, “S1D13806 Configuration” on page 39.

6 INTERFACING TO THE NEC VR4121^{TM} MICROPROCESSOR

6.1 Introduction

This application note describes the hardware and software environment necessary to provide an interface between the S1D13806 Embedded Memory Display Controller and the NEC VR4121^{TM} (µPD30121) microprocessor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the latest revision of this document before beginning any development.

6.2 Interfacing to the NEC VR4121

6.2.1 The NEC VR4121 System Bus

The VR-Series family of microprocessors features a high-speed synchronous system bus typical of modern microprocessors. Designed with external LCD controller support and Windows CE based embedded consumer applications in mind, the VR4121 offers a highly integrated solution for portable systems. This section provides an overview of the operation of the CPU bus in order to establish interface requirements.

Overview

The NEC VR4121 is designed around the RISC architecture developed by MIPS. This microprocessor is based on the 166MHz VR4120 CPU core which supports 64-bit processing. The CPU communicates with the Bus Control Unit (BCU) using its internal SysAD bus. The BCU in turn communicates with external devices using its ADD and DATA buses which can be dynamically sized to 16 or 32-bit operation.

The NEC VR4121 has direct support for an external LCD controller. Specific control signals are assigned for an external LCD controller providing an easy interface to the CPU. A 16M byte block of memory is assigned for the LCD controller and its own chip select and ready signals are available. Word or byte accesses are controlled by the system high byte signal (SHB#).

LCD Memory Access Cycles

Once an address in the LCD block of memory is placed on the external address bus (ADD[25:0]), the LCD chip select (LCDCS#) is driven low. The read or write enable signals (RD# or WR#) are driven low for the appropriate cycle and LCDRDY is driven low to insert wait states into the cycle. The high byte enable (SHB#) in conjunction with address bit 0 allows for byte steering.

The following figure illustrates typical NEC VR4121 memory read and write cycles to the LCD controller interface.

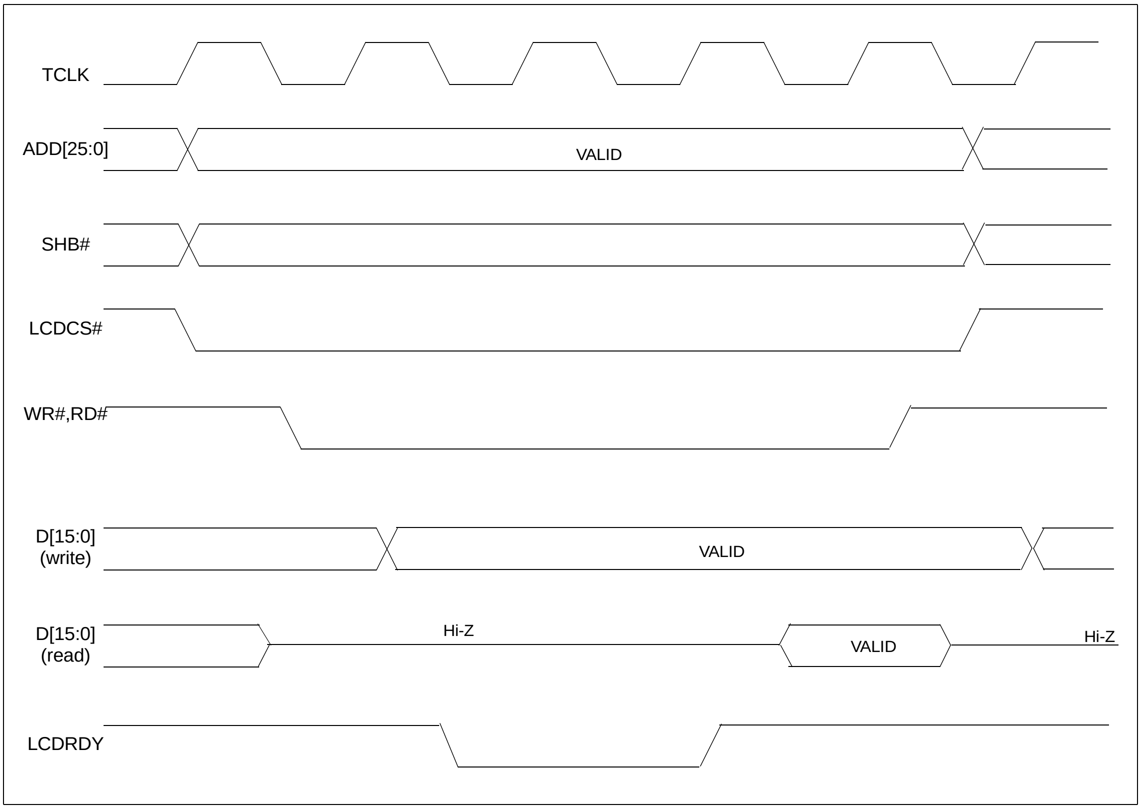


Figure 6-1 NEC VR4121 Read/Write Cycles

6.3 S1D13806 Host Bus Interface

The S1D13806 directly supports multiple processors. The S1D13806 implements a 16-bit MIPS/ ISA Host Bus Interface which is most suitable for direct connection to the VR4121 microprocessor.

The MIPS/ISA Host Bus Interface is selected by the S1D13806 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13806 configuration, see Section 6.4.2, "S1D13806 Configuration" on page 59.

Note: At reset, the Register/Memory Select bit in the Miscellaneous Register (REG[001h] bit 7) is set to 1. This means that only REG[000h] (read-only) and REG[001h] are accessible **until a write to REG[001h] sets bit 7 to 0 making all registers accessible**. When debugging a new hardware design, this can sometimes give the appearance that the interface is not working, so it is important to remember to clear this bit before proceeding with debugging.

6.3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 6-1 Host Bus Interface Pin Mapping

S1D13806 Pin Name	NEC VR4121 Pin Name
AB[20:0]	ADD[20:0]
DB[15:0]	DAT[15:0]
WE1#	SHB#
M/R#	ADD21
CS#	LCDCS#
BUSCLK	BUSCLK
BS#	V_{DD}
RD/WR#	V_{DD}
RD#	RD#
WE0#	WR#
WAIT#	LCDRDY
RESET#	connected to system reset

6.3.2 Host Bus Interface Signal Descriptions

The S1D13806 MIPS/ISA Host Bus Interface requires the following signals.

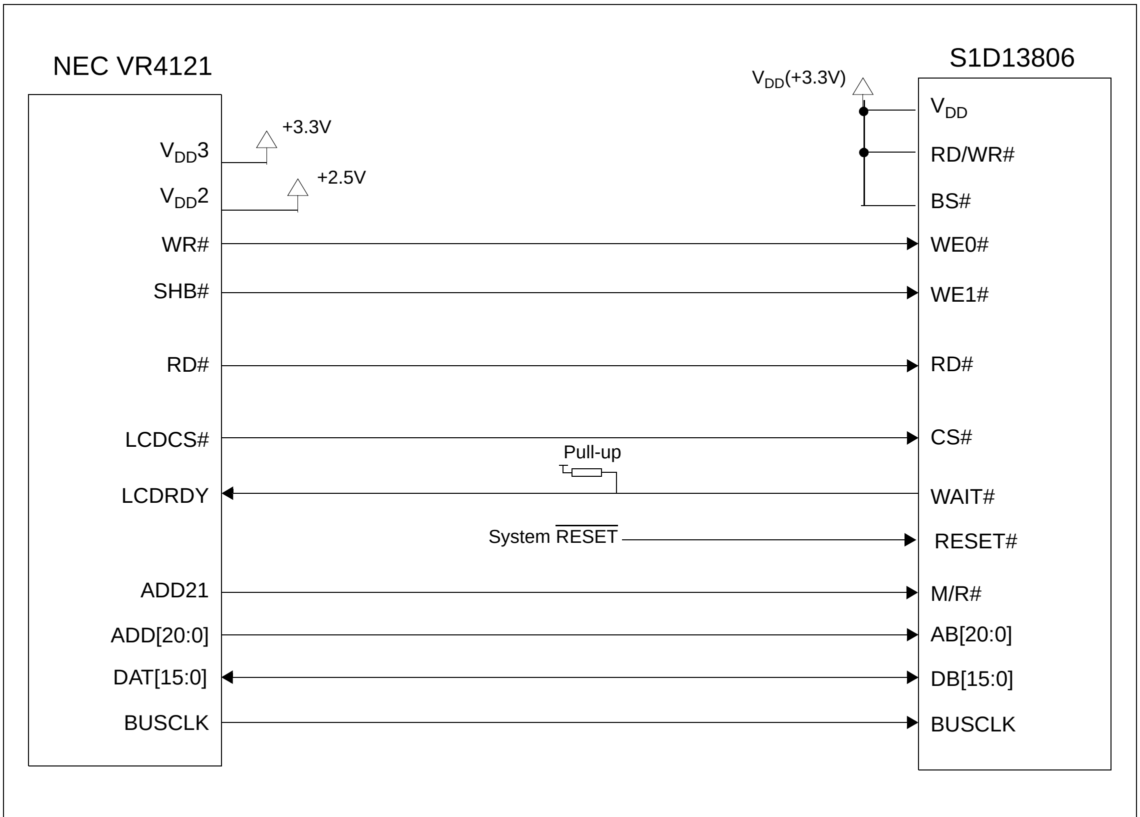
- BUSCLK is a clock input which is required by the S1D13806 Host Bus Interface. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB[20:0], and the data bus DB[15:0], connect directly to the VR4121 address (ADD[20:0]) and data bus (DAT[15:0]), respectively. CONF[3:0] must be set to select the MIPS/ ISA Host Bus Interface with little endian mode.
- M/R# (memory/register) selects between memory or register access. It may be connected to an address line, allowing system address ADD21 to be connected to the M/R# line.
- Chip Select (CS#) must be driven low by LCDCS# whenever the S1D13806 is accessed by the VR4121.
- WE1# connects to SHB# (the high byte enable signal from the VR4121) which in conjunction with address bit 0 allows byte steering of read and write operations.
- WE0# connects to WR# (the write enable signal from the VR4121) and must be driven low when the VR4121 bus is writing data to the S1D13806.
- RD# connects to RD# (the read enable signal from the VR4121) and must be driven low when the VR4121 bus is reading data from the S1D13806.
- WAIT# connects to LCDRDY and is a signal output from the S1D13806 that indicates the VR4121 bus must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since VR4121 bus accesses to the S1D13806 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13806 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The BS# and RD/WR# signals are not used for the MIPS/ISA Host Bus Interface and should be tied high (connected to V_{DD}).

6.4 VR4121 to S1D13806 Interface

6.4.1 Hardware Description

The NEC VR4121 microprocessor is specifically designed to support an external LCD controller. It provides all the necessary internal address decoding and control signals required by the S1D13806.

The diagram below shows a typical implementation utilizing the S1D13806.



Note:
When connecting the S1D13806 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13806 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 6-2 NEC VR4121 to S1D13806 Configuration Schematic

Note: For pin mapping see Table 1-1, “PC Card Host Bus Interface Pin Mapping,” on page 5.

6.4.2 S1D13806 Configuration

The S1D13806 latches CONF7 through CONF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx.

The table below shows the configuration settings important to the MIPS/ISA Host Bus Interface used by the NEC VR4121 microprocessor.

Table 6-2 Summary of Power-On/Reset Options

S1D13806 Pin Name	state of this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
CONF[3:0]	0100 = MIPS/ISA Host Bus Interface; Little Endian; Active Low WAIT# selected	
CONF4	LCDPWR Active Low	LCDPWR Active High
CONF5	BUSCLK input divided by two	BUSCLK input not divided by two
CONF6	WAIT# is always driven	WAIT# is floating if S1D13806 is not selected
CONF7	Configure GPIO12 as MediaPlug output pin VMPEPWR	Configure GPIO12 for normal use

▨ = configuration for NEC VR4121 microprocessor

6.4.3 NEC VR4121 Configuration

In the NEC VR4121 register BCUCNTREG1, the ISAM/LCD bit must be set to 0. A 0 indicates that the reserved address space is used by the external LCD controller rather than the high-speed ISA memory.

In the register BCUCNTREG2, the GMODE bit must be set to 1 indicating that LCD controller accesses use a non-inverting data bus.

In the register BCUCNTREG3, the LCD32/ISA32 bit must be set to 0. This sets the LCD interface to operate using a 16-bit data bus.

Note: Setting the LCD32/ISA32 bit to 0 in the BCUCNTREG3 register affects both the LCD controller and high-speed ISA memory access.

The frequency of BUSCLK output is programmed from the state of pins TxD/CLKSEL2, RTS#/CLKSEL1 and DTR#/CLKSEL0 during reset, and from the PMU (Power Management Unit) configuration registers of the NEC VR4121. The S1D13806 works at any of the frequencies provided by the NEC VR4121.

6.4.4 Register/Memory Mapping

The NEC VR4121 provides the internal address decoding necessary to map an external LCD controller. Physical address 0A00 0000h to 0AFF FFFFh (16M bytes) is reserved for an external LCD controller such as the S1D13806.

The S1D13806 is a memory-mapped device. The internal registers are mapped in the lower address space starting at zero. The display buffer requires 1.25M bytes and is mapped in the third and fourth megabytes (0A20 0000h to 0A33 FFFFh).

A typical implementation as shown in Figure 6-2 “NEC VR4121 to S1D13806 Configuration Schematic” on page 58 has the Memory/Register select pin (M/R#) connected to NEC VR4121 address line ADD21. ADD21 selects between the S1D13806 display buffer (ADD21 = 1) and internal registers (ADD21 = 0). This implementation decodes as shown in the following table.

Table 6-3 Register/Memory Mapping for Typical Implementation

ADD21 (M/R#)	ADD20	ADD12	Physical Address Range	Function
0	0	0	0A00 0000h to 0A00 01FFh	Control Registers Decoded
0	0	1	0A00 1000h to 0A00 1FFFh	MediaPlug Registers Decoded
0	1	x	0A10 0000h to 0A1F FFFFh	BitBLT Registers Decoded
1	x	x	0A20 0000h to 0A33 FFFFh	Display Buffer Decoded

x = don’t care

The NEC VR4121 provides 16M bytes of address space. Since the NEC VR4121 address bits ADD[23:22] are ignored, the S1D13806 registers and display buffer are aliased within the allocated address space. If aliasing is undesirable, the address space must be fully decoded.

Note: Address lines ADD[25:24] are set at 10b and never change while the LCD controller is being addressed.

7 INTERFACING TO THE STRONGARM SA-1110 PROCESSOR

7.1 Introduction

This application note describes the hardware and software environment required to provide an interface between the S1D13806 Embedded Memory Display Controller and the Intel StrongARM SA-1110.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the latest revision of this document before beginning any development.

7.2 Interfacing to the StrongARM SA-1110 Bus

7.2.1 The StrongARM SA-1110 System Bus

The StrongARM SA-1110 microprocessor is a highly integrated communications microcontroller that incorporates a 32-bit StrongARM RISC processor core. The SA-1110 is ideally suited to interface to the S1D13806 LCD controller and provides a high performance, power efficient solution for embedded systems.

StrongARM SA-1110 Overview

The SA-1110 system bus can access both variable-latency IO and memory devices. The SA-1110 uses a 26-bit address bus and a 32-bit data bus which can be used to access 16-bit devices. A chip select module with six chip select signals (each accessing 64M bytes of memory) allows selection of external devices. Only chip selects 3 through 5 (nCS[5:3]) may be used to select variable-latency devices which use RDY to extend access cycles. These chip selects are individually programmed in the SA-1110 memory configuration registers and can be configured for either a 16 or 32-bit data bus.

Byte steering is implemented using the four signals nCAS[3:0]. Each signal selects a byte on the 32-bit data bus. For example, nCAS0 selects bits D[7:0] and nCAS3 selects bits D[31:24]. For a 16-bit data bus, only nCAS[1:0] are used with nCAS0 selecting the low byte and nCAS1 selecting the high byte. The SA-1110 can be configured to support little or big endian mode.

Variable-Latency IO Access Overview

A data transfer is initiated when a memory address is placed on the SA-1110 system bus **and** a chip select signal (nCS[5:3]) is driven low. If all byte enable signals (nCAS[3:0]) are driven low, then a 32-bit transfer takes place. If only nCAS[1:0] are driven low, then a word transfer takes place through a 16-bit bus interface. If only one byte enable is driven low, then a byte transfer takes place on the respective data lines.

During a read cycle, the output enable signal (nOE) is driven low. A write cycle is specified by driving nOE high and driving the write enable signal (nWE) low. The cycle can be lengthened by driving RDY high for the time needed to complete the cycle.

Variable-Latency IO Access Cycles

The first nOE assertion occurs two memory cycles after the assertion of chip select (nCS3, nCS4, or nCS5). Two memory cycles prior to the end of minimum nOE or nWE assertion (RDF+1 memory cycles), the SA-1110 starts sampling the data ready input (RDY). Samples are taken every half memory cycle until **three consecutive samples** (at the rising edge, falling edge, and following rising edge of the memory clock) indicate that the IO device is ready for data transfer. Read data is latched one-half memory cycle after the third successful sample (on falling edge). Then nOE or nWE is deasserted on the next rising edge and the address may change on the subsequent falling edge. Prior to a subsequent data cycle, nOE or nWE remains deasserted for RDN+1 memory cycles. The chip select and byte selects (nCAS/DQM[1:0] for 16-bit data transfers), remain asserted for one memory cycle after the final nOE or nWE deassertion of the burst.

The SA-1110 is capable of burst cycles during which the chip select remains low while the read or write command is asserted, precharged and reasserted repeatedly.

Figure 7-1 illustrates a typical variable-latency IO access read cycle on the SA-1110 bus.

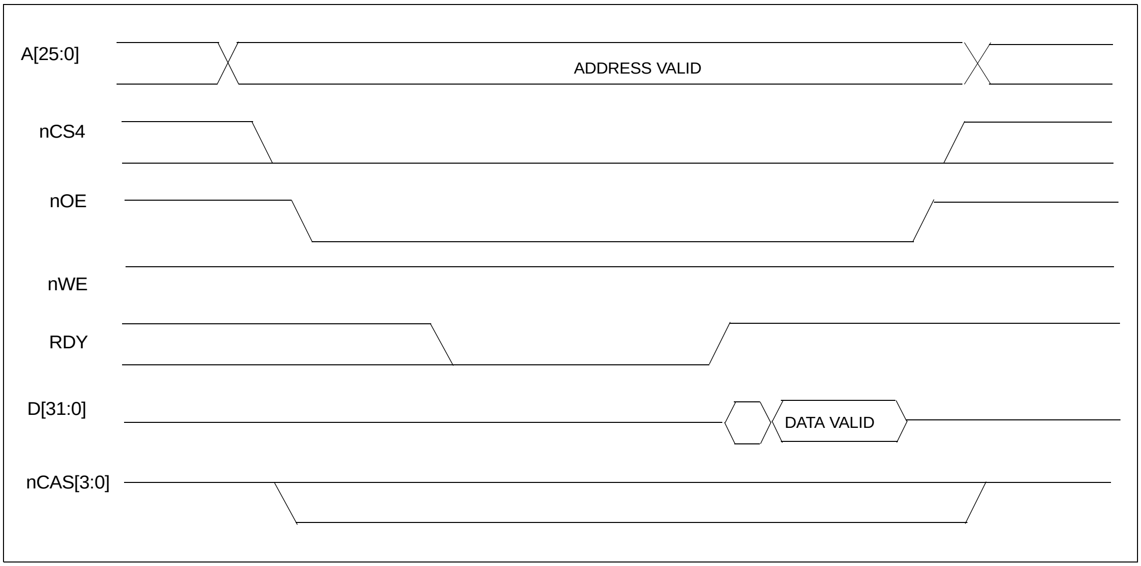


Figure 7-1 SA-1110 Variable-Latency IO Read Cycle

Figure 1-2 illustrates a typical variable-latency IO access write cycle on the SA-1110 bus.

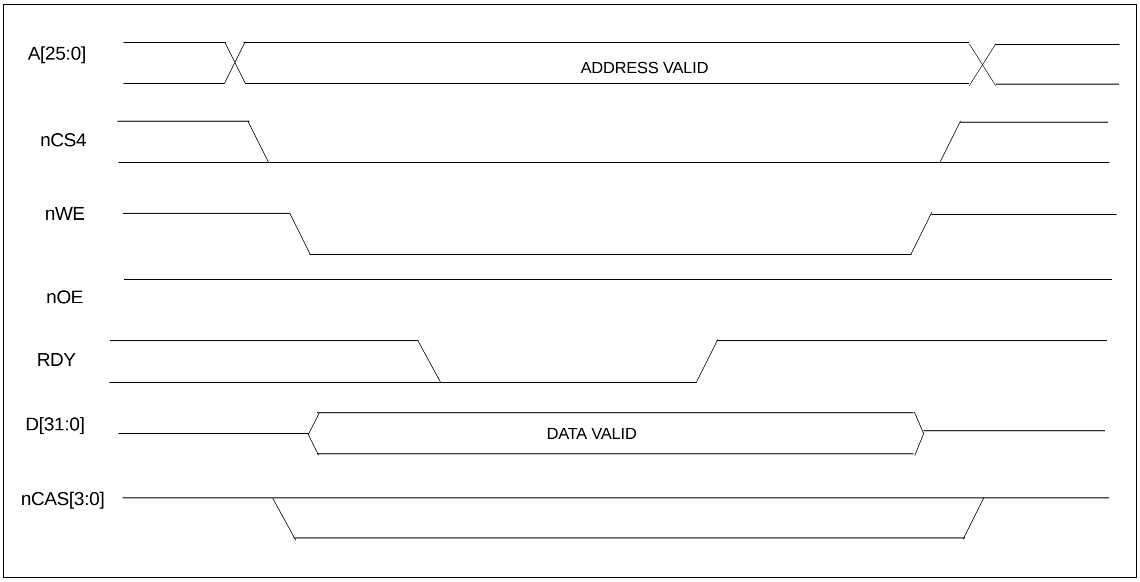


Figure 7-2 SA-1110 Variable-Latency IO Write Cycle

7.3 S1D13806 Host Bus Interface

The S1D13806 directly supports multiple processors. The S1D13806 implements a 16-bit PC Card (PCMCIA) Host Bus Interface which is most suitable for direct connection to the SA-1110.

The PC Card Host Bus Interface is selected by the S1D13806 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13806 configuration, see Section 1.4.2, “S1D13806 Hardware Configuration” on page 9.

Note: At reset, the Register/Memory Select bit in the Miscellaneous Register (REG[001h] bit 7) is set to 1. This means that only REG[000h] (read-only) and REG[001h] are accessible **until a write to REG[001h] sets bit 7 to 0 making all registers accessible**. When debugging a new hardware design, this can sometimes give the appearance that the interface is not working, so it is important to remember to clear this bit before proceeding with debugging.

7.3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 7-1 Host Bus Interface Pin Mapping

S1D13806 Pin Name	SA-1110
AB[20:0][1]	A[20:0]
DB[15:0]	D[15:0]
WE1#	nCAS1
M/R#	A21
CS#	nCS4
BUSCLK	SDCLK2
BS#	V_{DD}
RD/WR#	nCAS0
RD#	nOE
WE0#	nWE
WAIT#	RDY
RESET#	system RESET

Note: 1. The bus signal A0 is not used by the S1D13806 internally.

7.3.2 Host Bus Interface Signal Descriptions

The S1D13806 PC Card Host Bus Interface requires the following signals.

- BUSCLK is a clock input which is required by the S1D13806 Host Bus Interface. It is driven by one of the SA-1110 signals SDCLK1 or SDCLK2 (the example implementation in this document uses SDCLK2). For further information, see Section 7.4.3, “Performance” on page 68.
- The address inputs AB[20:0], and the data bus DB[15:0], connect directly to the SA-1110 address (A[20:0]) and data bus (D[15:0]), respectively. CONF[3:0] must be set to select the PC Card Host Bus Interface with little endian mode.
- M/R# (memory/register) selects between memory or register access. It may be connected to an address line, allowing system address A21 to be connected to the M/R# line.
- Chip Select (CS#) must be driven low by nCSx (where x is the SA-1110 chip select used) whenever the S1D13806 is accessed by the SA-1110.
- WE1# and RD/WR# connect to nCAS1 and nCAS0 (the byte enables for the high-order and low-order bytes). They are driven low when the SA-1110 is accessing the S1D13806.
- RD# connects to nOE (the read enable signal from the SA-1110).
- WE0# connects to nWE (the write enable signal from the SA-1110).
- WAIT# is a signal output from the S1D13806 that indicates the SA-1110 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since SA-1110 accesses to the S1D13806 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13806 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Start (BS#) signal is not used for this Host Bus Interface and should be tied high (connected to V_{DD}).
- The RESET# (active low) input of the S1D13806 may be connected to the system RESET.

7.4 StrongARM SA-1110 to S1D13806 Interface

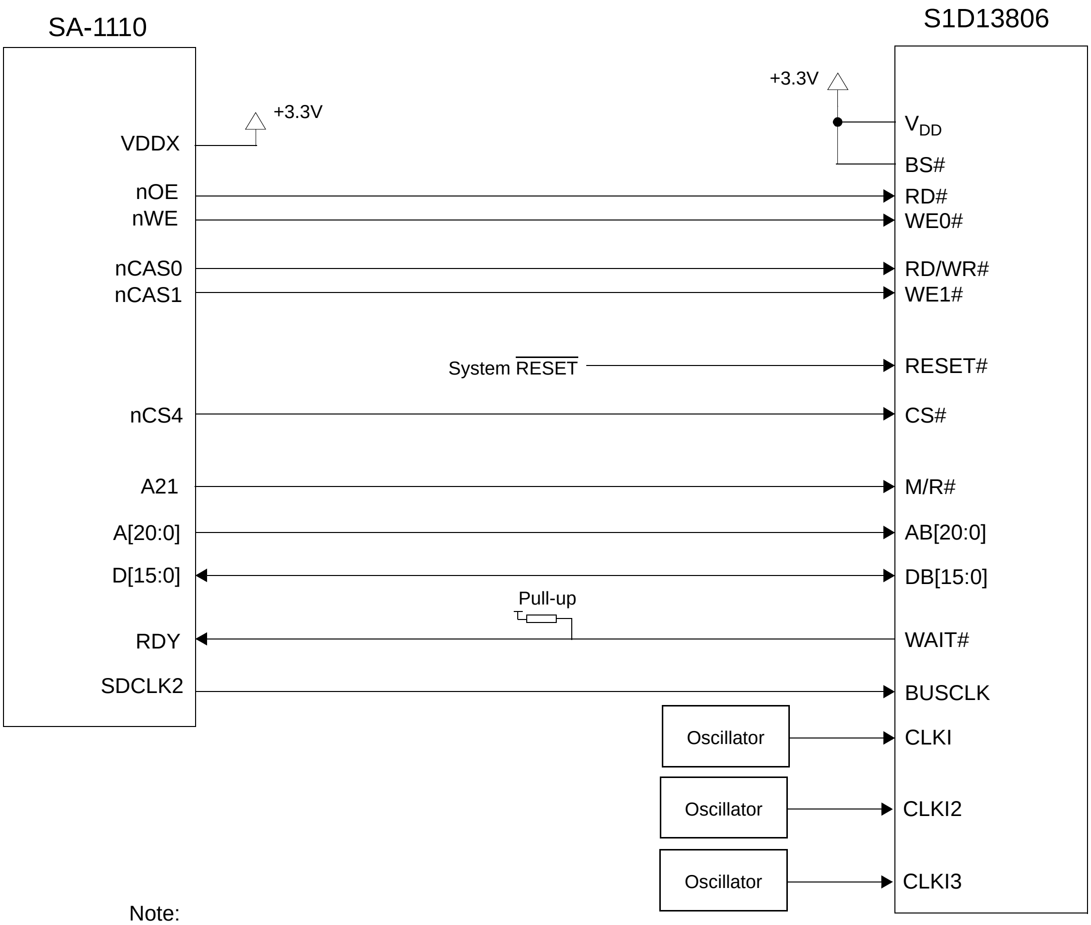
7.4.1 Hardware Description

The S1D13806 is designed to directly support a variety of CPUs, providing an interface to the unique “local bus” of each processor. The S1D13806’s PC Card Host Bus Interface provides a “glueless” interface to the SA-1110.

In this implementation, the address inputs (AB[20:0]) and data bus (DB[15:0]) connect directly to the CPU address (A[20:0]) and data bus (D[15:0]). M/R# is treated as an address line so that it can be controlled using system address A21.

BS# (Bus Start) is not used and should be tied high (connected to V_{DD}).

The following diagram shows a typical implementation of the SA-1110 to S1D13806 interface.



Note:
When connecting the S1D13806 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13806 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 7-3 Typical Implementation of SA-1110 to S1D13806 Interface

7.4.2 S1D13806 Hardware Configuration

The S1D13806 latches CONF7 through CONF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the "*S1D13806 Hardware Functional Specification*," document number X28B-A-001-xx.

The table below shows the configuration settings important to the PC Card Host Bus Interface used by the StrongARM SA-1110 microprocessor.

Table 7-2 Summary of Power-On/Reset Options

S1D13806 Pin Name	value on this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
CONF[3:0]	1001 = PC Card Host Bus Interface; Little Endian; Active Low WAIT# selected	
CONF4	LCDPWR Active Low	LCDPWR Active High
CONF5	BUSCLK input divided by two	BUSCLK input not divided by two
CONF6	WAIT# is always driven	WAIT# is floating if S1D13806 is not selected
CONF7	Configure GPIO12 as MediaPlug output pin VMPEPWR	Configure GPIO12 for normal use

▨ = configuration for StrongARM SA-1110 microprocessor

7.4.3 Performance

The S1D13806 PC Card Interface specification supports a BUSCLK up to 50MHz, and therefore provides a high performance display solution.

The BUSCLK signal input to the S1D13806 (from one of the SDCLK[2:1] pins) is a derivative of the SA-1110 internal processor speed. Since the PC Card Host Bus Interface on the S1D13806 has a maximum BUSCLK of 50MHz, the output clock from the SA-1110 must be a divided down from the processor clock. If the processor clock is higher than 100MHz, either divide the BUSCLK input by two (CONF5 = 1) or set SDCLK1/SDCLK2 to CPU clock divided by four using the DRAM Refresh Control Register (MDREFR) which determines the output of this signal.

- If SDCLK2 is used, bit 26 should be set to 1 to divide the CPU clock by 4.
- If SDCLK1 is used, bit 22 should be set to 1 to divide the CPU clock by 4.

7.4.4 StrongARM SA-1110 Register Configuration

The SA-1110 requires configuration of several of its internal registers to interface to the S1D13806 PC Card Host Bus Interface.

- The Static Memory Control Registers (MSC[2:0]) are read/write registers containing control bits for configuring static memory or variable-latency IO devices. These registers correspond to chip select pairs nCS[5:4], nCS[3:2], and nCS[1:0] respectively. Each of the three registers contains two identical CNFG fields, one for each chip select within the pair. Since only nCS[5:3] controls variable-latency IO devices, MSC2 and MSC1 should be programmed based on the chip select used.

  Parameter RTx<1:0> should be set to 01b (selects variable-latency IO mode).

  Parameter RBWx should be set to 1 (selects 16-bit bus width).

  Parameter RDFx<4:0> should be set according to the maximum desired CPU frequency as indicated in the table below.

Table 7-3 RDFx Parameter Value versus CPU Maximum Frequency

CPU Frequency (MHz)	RDFx
57.3 - 85.9	1
88.5 - 143.2	2
147.5 - 200.5	3
206.4 - 221.2	4

  Parameter RDNx<4:0> should be set to 0 (minimum command precharge time).

  Parameter RRRx<2:0> should be set to 0 (minimum nCSx precharge time).

- The S1D13806 endian mode is set to little endian. To program the SA-1110 for little endian set bit 7 of the control register (register 1) to 0.
- The BUSCLK signal input to the S1D13806 (from one of the SDCLK[2:1] pins) is a derivative of the SA-1110 internal processor speed. Since the PC Card Host Bus Interface on the S1D13806 has a maximum BUSCLK of 50MHz, the output clock from the SA-1110 must be a divided down from the processor clock. If the processor clock is higher than 100MHz, either divide the BUSCLK input by two (CONF5 = 1) or set SDCLK1/SDCLK2 to CPU clock divided by four using the DRAM Refresh Control Register (MDREFR) which determines the output of this signal.

  •If SDCLK2 is used, bit 26 should be set to 1 to divide the CPU clock by 4.

  •If SDCLK1 is used, bit 22 should be set to 1 to divide the CPU clock by 4.

7.4.5 Register/Memory Mapping

The S1D13806 is a memory-mapped device. The SA-1110 uses the memory assigned to a chip select (nCS4 in this example) to map the S1D13806 internal registers and display buffer. The internal registers are mapped in the lower SA-1110 memory address space starting at zero. The display buffer requires 1.25M bytes and is mapped in the third and fourth megabytes of the SA-1110 address space (ranging from 20 0000h to 33 FFFFh).

A typical implementation as shown in Figure 1-3 “Typical Implementation of PC Card to S1D13806 Interface” on page 8 has Chip Select (CS#) connected to ground (always enabled) and the Memory/Register select pin (M/R#) connected to address bit A21. This implementation decodes as shown in the following table.

Table 7-4 Register/Memory Mapping for Typical Implementation

A21 (M/R#)	A20	A12	Address Range	Function
0	0	0	0 to 1FFh	Control Registers Decoded
0	0	1	1000h to 1FFFh	MediaPlug Registers Decoded
0	1	x	10 0000h to 1F FFFFh	BitBLT Registers Decoded
1	x	x	20 0000h to 33 FFFFh	Display Buffer Decoded

x = don’t care

Each chip select on the SA-1110 provides 64M byte of address space. Since the SA-1110 address bits A[25:22] are ignored, the S1D13806 registers and display buffer are aliased within the allocated address space. If aliasing is undesirable, the address space must be fully decoded.

S1D13806 Series

Embedded DRAM Graphics Controller

Power Consumption

Table of Contents

List of Tables

1 S1D13806 Power Consumption

S1D13806 power consumption is affected by many system design variables.

- Input clock frequency (CLKI/CLKI2/CLKI3): the CLKI/CLKI2/CLKI3 frequency determines the LCD/CRT frame-rate, CPU performance to memory, and other functions – the higher the input clock frequency, the higher the frame-rate, performance and power consumption.
- CPU interface: the S1D13806 current consumption depends on the BUSCLK frequency, data width, number of toggling pins, and other factors – the higher the BUSCLK, the higher the CPU performance and power consumption.
- V_{DD} voltage level: the voltage level affects power consumption – the higher the voltage, the higher the consumption.
- Display mode: the resolution and color depth affect power consumption – the higher the resolution/color depth, the higher the consumption.
- Internal CLK divide: internal registers allow the input clock to be divided before going to the internal logic blocks – the higher the divide, the lower the power consumption.

There is a power save mode in the S1D13806. The power consumption is affected by various system design variables.

- Clock states during the power save mode: disabling the clocks during power save mode has substantial power savings.

1.1 Conditions

The following table gives an example of a specific environment and its effects on power consumption.

Table 1-1 S1D13806ES Total Power Consumption in mW

Test Condition All V_{DD} = 3.3V ISA Bus (8MHz)	Color Depth	S1D13806 Active (mW)	Power Save Mode	
			Clocks Active (mW)	Clocks Removed (mW)
CLKI = 6MHz LCD Panel = 60Hz 320 × 240 4-bit Single Monochrome	4 bpp	22.77	4.49	.43
CLKI = 6 MHz LCD Panel = 60Hz 320 × 240 8-bit Single Color	4 bpp	24.26	4.49	.43
	8 bpp	26.66	4.49	.43
	16 bpp	30.29	4.49	.43
CLKI = 25MHz LCD Panel = 60Hz 640 × 480 8-bit Dual Monochrome	4 bpp	67.82	11.19	.43
CLKI = 25MHz LCD Panel = 60Hz 640 × 480 16-bit Dual Color	4 bpp	85.07	11.19	.43
	8 bpp	90.78	11.19	.43
	16 bpp	96.10	11.19	.43

CLKI = 33.333MHz, CLKI2 = 25.175MHz
CRT = 60Hz 640 × 480 Color
	4 bpp	114.08	12.9	.43

	8 bpp	376.46	12.9	.43
	16 bpp	402.11	12.9	.43
CLKI = 33.333MHz, CLKI2 = 14.31818MHz NTSC TV = 640 × 480 Color, S-Video output, no filter	4 bpp	373.30	11.02	.43
	8 bpp	379.10	11.02	.43
	16 bpp	389.70	11.02	.43
CLKI = 33.333MHz, CLKI2 = 17.734475MHz PAL TV = 640 × 480 Color, S-Video output, no filter	4 bpp	375.94	11.62	.43
	8 bpp	380.95	11.62	.43
	16 bpp	389.43	11.62	.43

Note: 1. Conditions for power save mode with Clocks active:
- CPU interface active
- CLKI set to MCLK
- CLKI2 grounded when CRT/TV disabled
- CLKI3 grounded
- BUSCLK active
- Self-Refresh DRAM

2. Conditions for power save mode with Clocks inactive:
- CPU interface inactive
- CLKI, CLKI2, CLKI3, BUSCLK stopped
- Self-Refresh DRAM

2 SUMMARY

The system design variables in Section 1, “S1D13806 Power Consumption” and in Table 1-1 show that S1D13806 power consumption depends on the specific implementation. Active Mode power consumption depends on the desired CPU performance and LCD/CRT frame-rate, whereas power save mode consumption depends on the CPU Interface and Input Clock state.

In a typical design environment, the S1D13806 can be configured to be an extremely power-efficient LCD/CRT/TV Controller with high performance and flexibility.

THIS PAGE IS BLANK.

S1D13806 Series

Embedded DRAM Graphics Controller

Windows® CE Display Driver

Table of Contents

1 WINDOWS® CE DISPLAY DRIVERS

The Windows CE display drivers are designed to support the S1D13806 Color LCD/CRT/TV Controller running under the Microsoft Windows CE operating system. Available drivers include: 4, 8 and 16 bit-per-pixel landscape modes (no rotation), and 8 and 16 bit-per-pixel SwivelView™ 90 degree, 180 degree and 270 degree modes.

This document and the updated source code for the Windows CE drivers is updated as appropriate. Please check the latest revisions before beginning any development.

1.1 Program Requirements

Video Controller	:	S1D13806
Display Type	:	LCD or CRT
Windows Version	:	CE Version 2.0 and 2.11

1.2 *Example Driver Builds*

Build for CEPC (X86) on Windows CE 2.0

To build a Windows CE v2.0 display driver for the CEPC (X86) platform using a S5U13806B00C evaluation board, follow the instructions below:

1. Install Microsoft Windows NT v4.0.
2. Install Microsoft Visual C/C++ v5.0.
3. Install the Microsoft Windows CE Embedded Toolkit (ETK) by running SETUP.EXE from the ETK compact disc #1.
4. Create a new project by following the procedure documented in “Creating a New Project Directory” from the Windows CE ETK v2.0. Alternately, use the current “DEMO7” project included with the ETK v2.0. Follow the steps below to create a “X86 DEMO7” shortcut on the Windows NT v4.0 desktop which uses the current “DEMO7” project:
   a. Right click on the “Start” menu on the taskbar.
   b. Click on the item “Open All Users” and the “Start Menu” window will come up.
   c. Click on the icon “Programs”.
   d. Click on the icon “Windows CE Embedded Development Kit”.
   e. Drag the icon “X86 DEMO1” onto the desktop using the right mouse button.
   f. Click on “Copy Here”.
   g. Rename the icon “X86 DEMO1” on the desktop to “X86 DEMO7” by right clicking on the icon and choosing “rename”.
   h. Right click on the icon “X86 DEMO7” and click on “Properties” to bring up the “X86 DEMO7 Properties” window.
   i. Click on “Shortcut” and replace the string “DEMO1” under the entry “Target” with “DEMO7”.
   j. Click on “OK” to finish.

Note: You may need administrator privilege on your local developing machine to perform the above steps. If you only have a user privilege, you may refer to step 2 of the Section , “Build for CEPC (X86) on Windows CE Platform Builder 2.11” on page 4.

5. Create a sub-directory named S1D13806 under \wince\platform\cepc\drivers\display.
6. Copy the source code to the S1D13806 subdirectory.
7. Add an entry for the S1D13806 in the file \wince\platform\cepc\drivers\display\dirs.

8. Edit the file PLATFORM.BIB (located in X:\wince\platform\cepc\files) to set the default display driver to the file EPSON.DLL. (EPSON.DLL will be created during the build in step 12)

   You may replace the following lines in PLATFORM.BIB:

   ```
   IF CEPC_DDI_VGA2BPP
       ddi.dll     $(_FLATRELEASEDIR)\ddi_vga2.dll      NK SH
   ENDIF
   IF CEPC_DDI_VGA8BPP
       ddi.dll     $(_FLATRELEASEDIR)\ddi_vga8.dll      NK SH
   ENDIF
   IF CEPC_DDI_VGA2BPP !
   IF CEPC_DDI_VGA8BPP !
       ddi.dll     $(_FLATRELEASEDIR)\ddi_s364.dll      NK SH
   ENDIF
   ENDIF
   ```

   with this line:

   ```
       ddi.dll     $(_FLATRELEASEDIR)\EPSON.dll         NK SH
   ```

9. If the current MODE0.H is not appropriate for your project, generate a new MODE0.H file using the S1D13806 utility program 1386CFG.EXE. The file MODE0.H (located in X:\wince\platform\cepc\drivers\display\S1D13806) contains the register values required to set desired screen resolution, color depth (bpp), panel type, active display(LCD/CRT/TV), rotation, etc.

10. Edit the file PLATFORM.REG to match the screen resolution, color depth (bpp), active display(LCD/CRT/TV) and rotation information in MODE.H. PLATFORM.REG is located in X:\wince\platform\cepc\files. For example, the display driver section of PLATFORM.REG should be as follows when using a 640x480 LCD panel with a color depth of 8 bpp in SwivelView 0° (landscape) mode.

    ```
    ; Default for EPSON Display Driver
    ; 640 × 480 at 8bits/pixel, LCD display, no rotation
    ; Useful Hex Values
    ; 1024=0x400, 768=0x300 640=0x280 480=0x1E0 320=140 240=0xF0
    [HKEY_LOCAL_MACHINE\Drivers\Display\S1D13806]
    "Width"=dword:280
    "Height"=dword:1E0
    "Bpp"=dword:8
    “ActiveDisp”=dword:1
    “Rotation”=dword:0
    ```

11. Remove the X:\wince\release directory, and delete X:\wince\platform\cepc\*.bif
12. Generate the proper building environment by double-clicking on the sample project icon (i.e. X86 DEMO7).
13. Type BLDDEMO <ENTER> at the DOS prompt of the X86 DEMO7 window to generate a Windows CE image file (NK.BIN).

Build for CEPC (X86) on Windows CE Platform Builder 2.11

1. Install Microsoft Windows NT v4.0.
2. Install Platform Builder 2.11 by running SETUP.EXE from compact disk #1.
3. Follow the steps below to create a “Build Epson for x86” shortcut which uses the current “Minshell” project icon/shortcut on the Windows NT 4.0 desktop.
   a. Right click on the “Start” menu on the taskbar.
   b. Click on the item “Explore”, and “Exploring -- Start Menu” window will come up.
   c. Under “\Winnt\Profiles\All Users\Start Menu\Programs\Microsoft Windows CE Platform Builder\x86 Tools, find the icon “Build Minshell for x86”.
   d. Drag the icon “Build Minshell for x86” onto the desktop using the right mouse button.
   e. Choose “Copy Here”.
   f. Rename the icon “Build Minshell for x86” to “Build Epson for x86” by right clicking on the icon and choosing “rename”.
   g. Right click on the icon “Build Epson for x86” and click on “Properties” to bring up the “Build Epson for x86 Properties” window.
   h. Click on “Shortcut” and replace the string “Minshelll” under the entry “Target” with “Epson”.
   i. Click on “OK” to finish.
4. Create an EPSON project.
   a. Make an Epson directory under \WINCE211\PUBLIC.
   b. Copy MAXALL and its sub-directories (\WINCE211\PUBLIC\MAXALL) to the Epson directory.

      xcopy /s /e \wince211\public\maxall\*.* \wince211\public\epson
   c. Rename \WINCE211\PUBLIC\EPSON\MAXALL.BAT to EPSON.BAT.
   d. Edit EPSON.BAT to add the following lines to the end of the file:

      ```
      @echo on
      set CEPC_DDI_S1D13806=1
      @echo off
      ```

5. Make a S1D13806 directory under \WINCE211\PLATFORM\CEPC\DRIVERS\DISPLAY, and copy the S1D13806 driver source code into \WINCE211\PLATFORM\CEPC\DRIVERS\DISPLAY\S1D13806.

6. Add S1D13806 into the directory list in file --

   \WINCE211\PLATFORM\CEPC\DRIVERS\DISPLAY\dirs

7. Edit the file \WINCE211\PLATFORM\CEPC\FILES\platform.bib to add the following after the line “IF ODO_NODISPLAY!”:

   ```
   IF CEPC_DDI_S1D13806
       ddi.dll     $(_FLATRELEASEDIR)\epson.dll         NK SH
   ENDIF
   ```

   Replace the section:

   ```
   IF CEPC_DDI_S3VIRGE !
   IF CEPC_DDI_CT655X !
   IF CEPC_DDI_VGA8BPP !
       ddi.dll     $(_FLATRELEASEDIR)\ddi_s364.dll       NK  SH
   ENDIF
   ENDIF
   ENDIF
   ```

   with the following:

   ```
   IF CEPC_DDI_S1D13806 !
   IF CEPC_DDI_S3VIRGE !
   IF CEPC_DDI_CT655X !
   IF CEPC_DDI_VGA8BPP !
       ddi.dll     $(_FLATRELEASEDIR)\ddi_s364.dll      NK  SH
   ENDIF
   ENDIF
   ENDIF
   ENDIF
   ```

8. If the current MODE0.H is not appropriate for your project, generate a new MODE0.H using the S1D13806 utility program 1386CFG.EXE. The file MODE0.H (located in X:\wince\platform\cepc\drivers\display\S1D13806) contains the register values required to set desired screen resolution, color depth (bpp), panel type, active display(LCD/CRT/TV), rotation, etc.

9. Edit the file PLATFORM.REG match the screen resolution, color depth (bpp), active display(LCD/CRT/TV) and rotation information in MODE.H. PLATFORM.REG is located in X:\wince\platform\cepc\files. For example, the display driver section of PLATFORM.REG should be as follows when using a 640 × 480 LCD panel with a color depth of 8 bpp in SwivelView 0° (landscape) mode.

   ```
   ; Default for EPSON Display Driver
   ; 640 × 480 at 8bits/pixel, LCD display, no rotation
   ; Useful Hex Values
   ; 1024=0x400, 768=0x300 640=0x280 480=0x1E0 320=140 240=0xF0
   [HKEY_LOCAL_MACHINE\Drivers\Display\S1D13806]
   "Width"=dword:280
   "Height"=dword:1E0
   "Bpp"=dword:8
   “ActiveDisp”=dword:1
   “Rotation”=dword:0
   ```

10. Remove the \wince211\release directory and delete \wince211\platform\cepc\*.bif

11. Generate the proper building environment by double-clicking on the Epson project icon --”Build Epson for x86”.

12. Type BLDDEMO <ENTER> at the DOS prompt of the “Build Epson for x86” window to generate a Windows CE image file (NK.BIN).

1.3 Installation for CEPC Environment

Windows CE v2.0 can be loaded on a PC using a floppy drive or a hard drive. The two methods are described below:

1. To load CEPC from a floppy drive:
   a. Create a DOS bootable floppy disk.
   b. Edit CONFIG.SYS on the floppy disk to contain the following line only.

      device=a:\himem.sys

   c. Edit AUTOEXEC.BAT on the floppy disk to contain the following lines.
      mode com1:9600,n,8,1
      loadcepc /B:9600 /C:1 c:\wince\release\nk.bin
   d. Copy LOADCEPC.EXE from c:\wince\public\common\oak\bin to the bootable floppy disk.
   e. Confirm that NK.BIN is located in c:\wince\release.
   f. Reboot the system from the bootable floppy disk.
2. To load CEPC from a hard drive:
   a. Copy LOADCEPC.EXE to the root directory of the hard drive.
   b. Edit CONFIG.SYS on the hard drive to contain the following line only.

      device=c:\himem.sys

   c. Edit AUTOEXEC.BAT on the hard drive to contain the following lines.

      mode com1:9600,n,8,1

      loadcepc /B:9600 /C:1 c:\wince\release\nk.bin

   d. Confirm that NK.BIN is located in c:\wince\release.
   e. Reboot the system from the hard drive.

1.4 Comments

- The display driver is CPU independent allowing use of the driver for other Windows CE Platform Builder v2.11 supported platforms. The file EPSON.CPP may require editing to return the correct value for PhysicalPortAddr, PhysicalVmemAddr, etc.
- The sample code defaults to a 640 × 480 16-bit color dual passive LCD panel in SwivelView 0° mode (landscape) with a color depth of 8 bpp. To support other settings, use 1386CFG.EXE to generate the proper MODE0.H file. For further information, refer to the “*1386CFG user manual*,” document number X28B-B-001-xx.
- As the time of printing, the drivers have been tested on the x86 CPUs and have been run with version 2.0 of the ETK and Platform Builder v2.11. The drivers are will be updated as appropriate. Before beginning any development, check the latest revision of this document.

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.

- HEADQUARTERS -
150 River Oaks Parkway
San Jose, CA 95134, U.S.A.
Phone: +1-408-922-0200 Fax: +1-408-922-0238

- SALES OFFICES -

West
1960 E. Grand Avenue
El Segundo, CA 90245, U.S.A.
Phone: +1-310-955-5300 Fax: +1-310-955-5400

Central
101 Virginia Street, Suite 290
Crystal Lake, IL 60014, U.S.A.
Phone: +1-815-455-7630 Fax: +1-815-455-7633

Northeast
301 Edgewater Place, Suite 120
Wakefield, MA 01880, U.S.A.
Phone: +1-781-246-3600 Fax: +1-781-246-5443

Southeast
3010 Royal Blvd. South, Suite 170
Alpharetta, GA 30005, U.S.A.
Phone: +1-877-EEA-0020 Fax: +1-770-777-2637

EUROPE

EPSON EUROPE ELECTRONICS GmbH

- HEADQUARTERS -
Riesstrasse 15
80992 Munich, GERMANY
Phone: +49-(0)89-14005-0 Fax: +49-(0)89-14005-110

SALES OFFICE
Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone: +49-(0)2171-5045-0 Fax: +49-(0)2171-5045-10

UK BRANCH OFFICE
Unit 2.4, Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

FRENCH BRANCH OFFICE
1 Avenue de l' Atlantique, LP 915 Les Conquerants
Z.A. de Courtaboeuf 2, F-91976 Les Ulis Cedex, FRANCE
Phone: +33-(0)1-64862350 Fax: +33-(0)1-64862355

BARCELONA BRANCH OFFICE
Barcelona Design Center
Edificio Testa
Avda. Alcalde Barrils num. 64-68
E-08190 Sant Cugat del Vallès, SPAIN
Phone:+34-93-544-2490 Fax:+34-93-544-2491

ASIA

EPSON (CHINA) CO., LTD.
23F, Beijing Silver Tower 2# North RD DongSanHuan ChaoYang
District, Beijing, CHINA
Phone: 64106655 Fax: 64107319

SHANGHAI BRANCH
4F, Bldg., 27, No. 69, Gui Jing Road
Caohejing, Shanghai, CHINA
Phone: 21-6485-5552 Fax: 21-6485-0775

EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, Hong Kong
Phone: +852-2585-4600 Fax: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
10F, No. 287, Nanking East Road, Sec. 3
Taipei
Phone: 02-2717-7360 Fax: 02-2712-9164
Telex: 24444 EPSONTB

HSINCHU OFFICE
13F-3, No. 295, Kuang-Fu Road, Sec. 2
HsinChu 300
Phone: 03-573-9900 Fax: 03-573-9169

EPSON SINGAPORE PTE., LTD.
No. 1 Temasek Avenue, #36-00
Millenia Tower, SINGAPORE 039192
Phone: +65-337-7911 Fax: +65-334-2716

SEIKO EPSON CORPORATION
KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: 02-784-6027 Fax: 02-767-3677

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

Electronic Device Marketing Department
IC Marketing & Engineering Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5816 Fax: +81-(0)42-587-5624

ED International Marketing Department
Europe & U.S.A.
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5812 Fax: +81-(0)42-587-5564

ED International Marketing Department
Asia
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5814 Fax: +81-(0)42-587-5110

ENERGY
SAVING
EPSON

In pursuit of **“Saving” Technology**, Epson electronic devices.
Our lineup of semiconductors, liquid crystal displays and quartz devices
assists in creating the products of our customers’ dreams.
Epson IS energy savings.

S1D13806 Series
Technicl Manual

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ **EPSON Electronic Devices Website**

http://www.epson.co.jp/device/

This manual was made with recycle papaer,
and printed using soy-based inks.

Issue July,2001
Printed in Japan ⒸB