S1D13506F00A
Technical Manual

EPSON

MF1255-02

Color LCD/CRT/TV Controller
S1D13506F00A
Technical Manual

S1D13506F00A Technical Manual

EPSON

SEIKO EPSON CORPORATION

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ EPSON Electronic Devices Website

http://www.epson.co.jp/device/

F43

This manual was made with recycle papaer,
and printed using soy-based inks.

PRINTED WITH
SOY INK

ENERGY
SAVING
EPSON
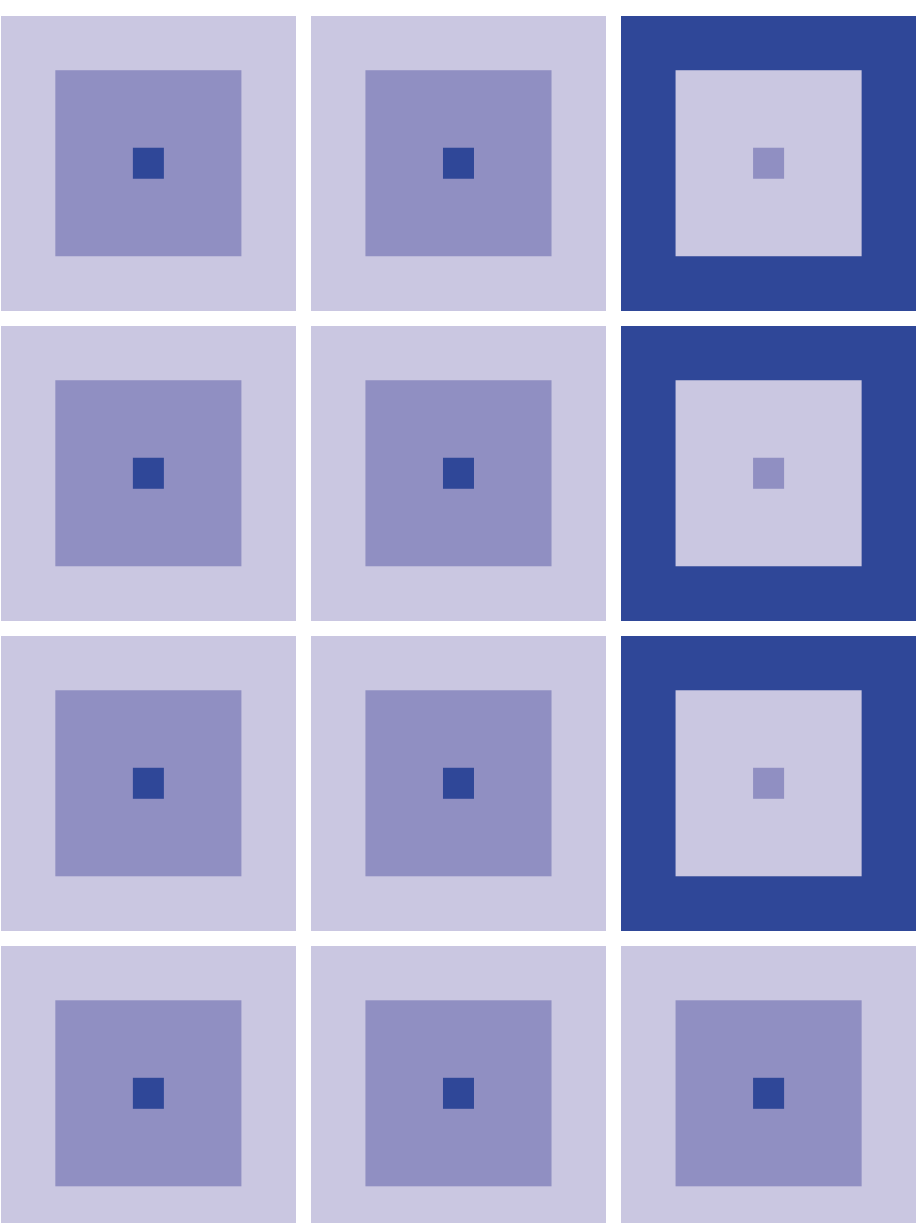
NOTICE

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.

MS-DOS and Windows are registered trademarks of Microsoft Corporation, U.S.A.
PC-DOS, PC/AT, VGA, EGA and IBM are registered trademarks of International Business Machines Corporation, U.S.A.
All other product names mentioned herein are trademarks and/or registered trademarks of their respective owners.
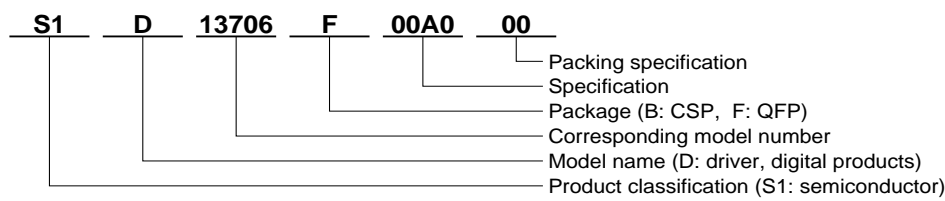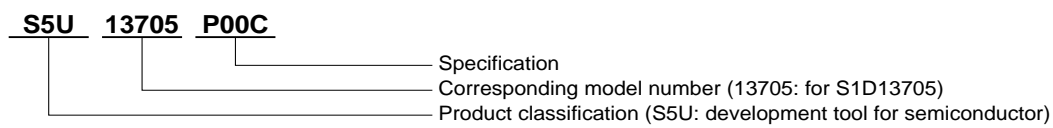
The information of the product number change

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

Configuration of product number

● 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 S1D13705)
  └────────────────── Product classification (S5U: development tool for semiconductor)

Comparison table between new and previous number

• S1D13305 Series

Previous No.	New No.
SED1335 Series	S1D13305 Series
SED1335D0A	S1D13305D00A
SED1335F0A	S1D13305F00A
SED1335F0B	S1D13305F00B

• S1D1350x Series

Previous No.	New No.
SED135x Series	S1D1350x Series
SED1353D0A	S1D13503D00A
SED1353F0A	S1D13503F00A
SED1353F1A	S1D13503F01A
SED1354F0A	S1D13504F00A
SED1354F1A	S1D13504F01A
SED1354F2A	S1D13504F02A
SED1355F0A	S1D13505F00A
SED1356F0A	S1D13506F00A

• S1D1370x Series

Previous No.	New No.
SED137x Series	S1D1370x Series
SED1374F0A	S1D13704F00A
SED1375F0A	S1D13705F00A
SED1376B0A	S1D13706B00A
SED1376F0A	S1D13706F00A
SED1378 Series	S1D13708 Series

• S1D13A0x Series

Previous No.	New No.
SED13Ax Series	S1D13A0x Series
SED13A3F0A	S1D13A03F00A
SED13A3B0B	S1D13A03B00B
SED13A4B0B	S1D13A04B00B

• S1D1380x Series

Previous No.	New No.
SED138x Series	S1D1380x Series
SED1386F0A	S1D13806F00A

Comparison table between new and previous number of Evaluation Boards

• S1D1350x Series

Previous No.	New No.
SDU1353#0C	S5U13503P00C
SDU1354#0C	S5U13504P00C
SDU1355#0C	S5U13505P00C
SDU1356#0C	S5U13506P00C

• S1D1370x Series

Previous No.	New No.
SDU1374#0C	S5U13704P00C
SDU1375#0C	S5U13705P00C
SDU1376#0C	S5U13706P00C
SDU1376BVR	S5U13706B32R
SDU1378#0C	S5U13708P00C

• S1D13A0x Series

Previous No.	New No.
SDU13A3#0C	S5U13A03P00C
SDU13A4#0C	S5U13A04P00C

• S1D1380x Series

Previous No.	New No.
SDU1386#0C	S5U13806P00C

S1D13506F00A Technical Manual

HARDWARE FUNCTIONAL SPECIFICATION

PROGRAMMING NOTES AND EXAMPLES

UTILITIES

S5U13506P00C PCI BUS EVALUATION BOARD USER'S MANUAL

APPLICATION NOTES

Windows® CE DISPLAY DRIVERS

S1D13506F00A
Color LCD/CRT/TV Controller

Hardware Functional
Specification

CONTENTS

List of Figures

CONTENTS

List of Tables

CONTENTS

EPSON
S1D13506F00A HARDWARE FUNCTIONAL
SPECIFICATION (X25B-A-001-08)

CONTENTS

1 INTRODUCTION

1.1 Scope

This is the Hardware Functional Specification for the S1D13506 Color LCD/CRT/TV 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 S1D13506 is a color LCD/CRT/TV graphics controller interfacing to a wide range of CPUs and display devices. The S1D13506 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 S1D13506 supports multiple CPUs, all LCD panel types, CRT, TV, and additionally provides a number of differentiating features. Products requiring digital camera input can take advantage of the directly supported WINNOV VideumCam™ digital interface. The EPSON Independent Simultaneous Display (EISD) capability allows the user to configure two different images on two different displays, while the SviwelView™, Hardware Cursor, Ink Layer, and BitBLT engine offer substantial performance benefits. These features, combined with the S1D13506's Operating System independence, make it an ideal display solution for a wide variety of applications.

2 FEATURES

2.1 Memory Interface

- 16-bit DRAM interface:
  - EDO-DRAM up to 40MHz data rate (80M Bytes/s).
  - FPM-DRAM up to 25MHz data rate (50M Bytes/s).
- Memory size options:
  - 512K bytes using one 256K×16 device.
  - 2M bytes using one 1M×16 device.
- A configuration register can be programmed to enhance performance by tailoring the memory control output timing to the DRAM device.
- The complete 2M byte display buffer address space is directly and contiguously available through the 21-bit address bus.

2.2 CPU Interface

- Supports the following interfaces:
  - Epson S1C33 (16-bit interface to 32-bit microprocessor).
  - Hitachi SH-4 bus interface.
  - Hitachi SH-3 bus interface.
  - MIPS/ISA.
  - Motorola MC68000 (16-bit interface to 16/32-bit microprocessor/microcontroller).
  - Motorola MC68030 (16-bit interface to 16/32-bit microprocessor/microcontroller).
  - Motorola PowerPC MPC82x (16-bit interface to 32-bit microprocessor).
  - MPU bus interface with programmable READY.
  - NEC MIPS VR41xx.
  - PC Card (PCMCIA).
  - Philips MIPS PR31500/31700.
  - Toshiba MIPS TX39xx.
  - StrongARM (PC Card).
- One-stage write buffer for minimum wait-state CPU writes.
- Registers are memory-mapped – the M/R# pin selects between display buffer and register address space.

2.3 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 9/12-bit TFT/D-TFD, 18-bit TFT/D-TFD is supported up to 64K colors.

- Direct support for CRT up to 64K colors using Embedded RAMDAC.

- Direct support for NTSC/PAL TV output using Embedded RAMDAC.

2.4 Display Modes

- 4/8/15/16 bit-per-pixel (bpp) color depths.

- Up to 64 shades of gray on monochrome passive LCD panels using Frame Rate Modulation (FRM) and Dithering.

- Up to 32K/64K colors in 15/16 bpp modes on color passive LCD panels using dithering.

- Up to 64K colors on TFT/D-TFD, CRT and TV.

- 4/8 bit-per-pixel color depths are mapped using three 256×4 Look-Up Tables (LUT) allowing 16/256 out of a possible 4096 colors.

- Separate LUTs for LCD and CRT/TV.

- 15/16 bit-per-pixel color depths are mapped directly, bypassing the LUT.

- Example Resolutions:
  320×240 at a color depth of 16 bpp.
  640×240 at a color depth of 16 bpp.
  640×480 at a color depth of 16 bpp.
  800×600 at a color depth of 16 bpp.

2.5 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/D-TFD panel).

- Virtual Display Support: displays images larger than the panel size through the use of panning and scrolling.

- Hardware Cursor/Ink Layer: separate 64×64×2 hardware cursor or 2-bit ink layer for both LCD and CRT/TV.

- Double Buffering/Multi-pages: for smooth animation and instantaneous screen update.

2.6 Clock Source

- Memory clock can be derived from CLKI or BUSCLK pin. It can be internally divided by 2.
- Pixel clock can be derived from CLKI, CLKI2, or BUSCLK pin. It can be internally divided by 2, 3 or 4.
- Bus clock can be BUSCLK or (BUSCLK)/2, i.e. a 2x clock may be used.

2.7 Acceleration

- 2D Engine including the following 2 ROP BitBlts:
    Write BLT.
    Move BLT.
    Solid Fill.
    Pattern Fill.
    Transparent Write BLT.
    Transparent Move BLT.
    Read BLT.
    Color Expansion.
    Move BLT with Color Expansion.

2.8 MediaPlug Interface

- Built-in WINNOV MediaPlug interface.
- VideumCam support at resolution of 320×240×256 color at 30fps.

2.9 Miscellaneous

- The memory data bus, MD[15:0], is used to configure the chip at power-on.
- Three General Purpose Input/Output pins, GPIO[3:1], are available if upper Memory Address pins are not required for asymmetric DRAM support.
- Power save mode is initiated by software.
- Operating voltage from 2.7 volts to 5.5 volts.
- 128-pin QFP15 surface mount package.

3 TYPICAL SYSTEM IMPLEMENTATION DIAGRAMS

For the pin mapping of each system implementation, see Table 5-7, "CPU Interface Pin Mapping" on page 1-24.
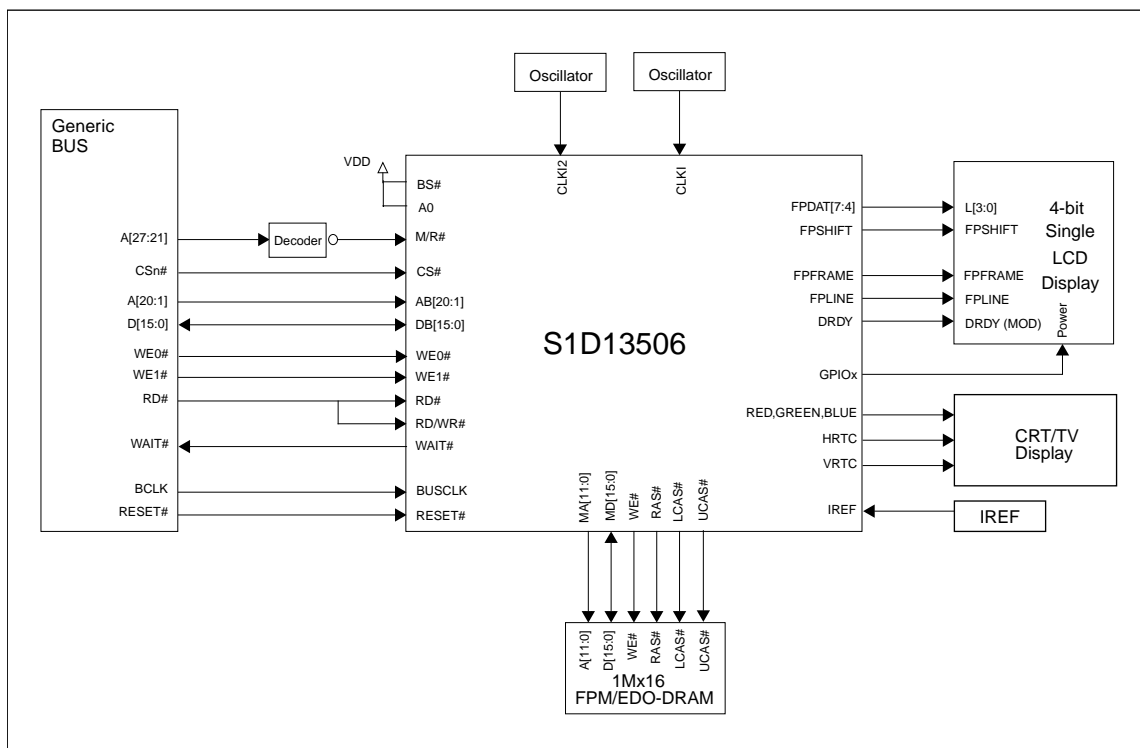
Figure 3-1 Typical System Diagram (Generic Bus)
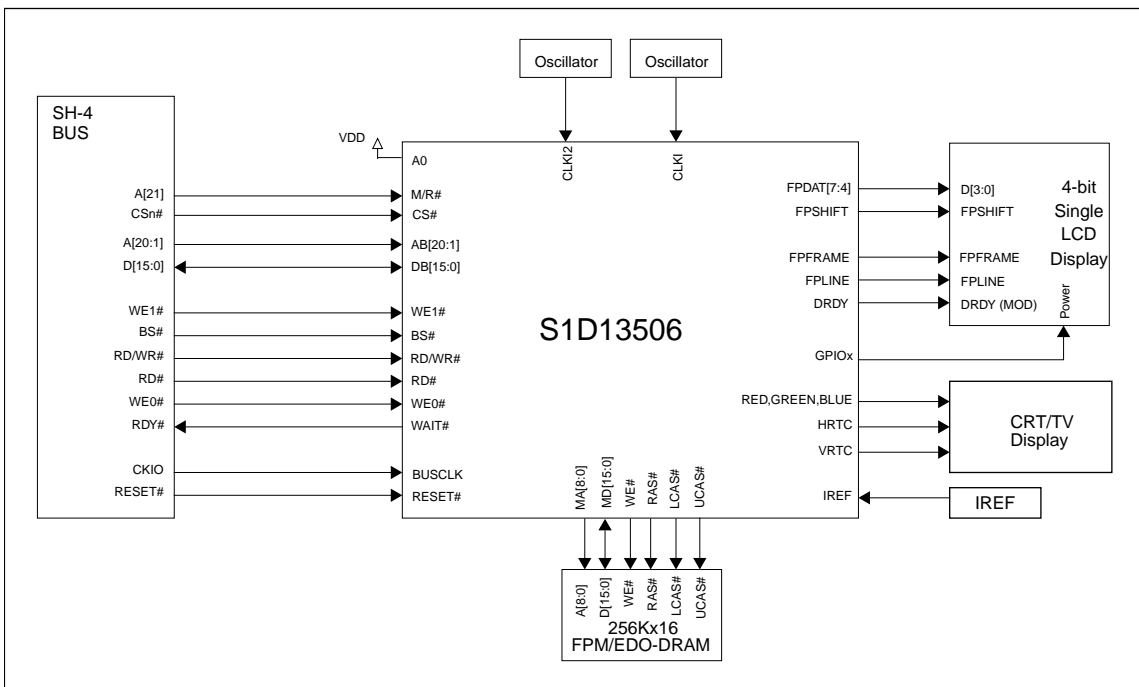
Figure 3-2 Typical System Diagram (Hitachi SH-4 Bus)
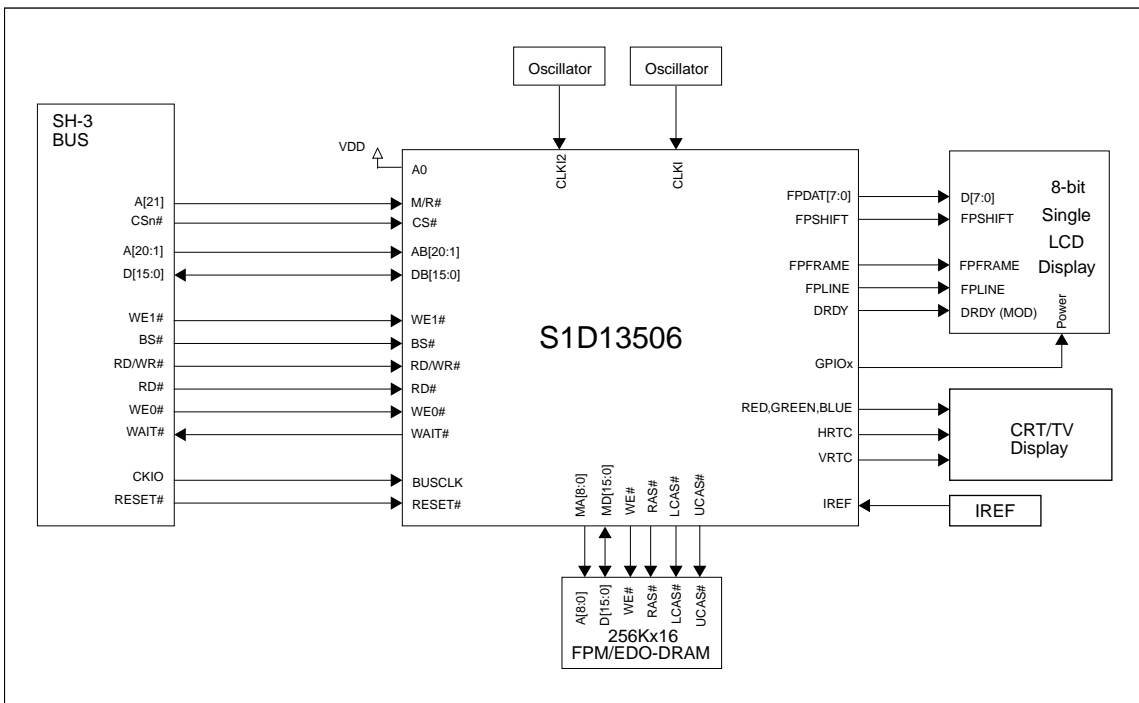
Figure 3-3 Typical System Diagram (Hitachi SH-3 Bus)

EPSON
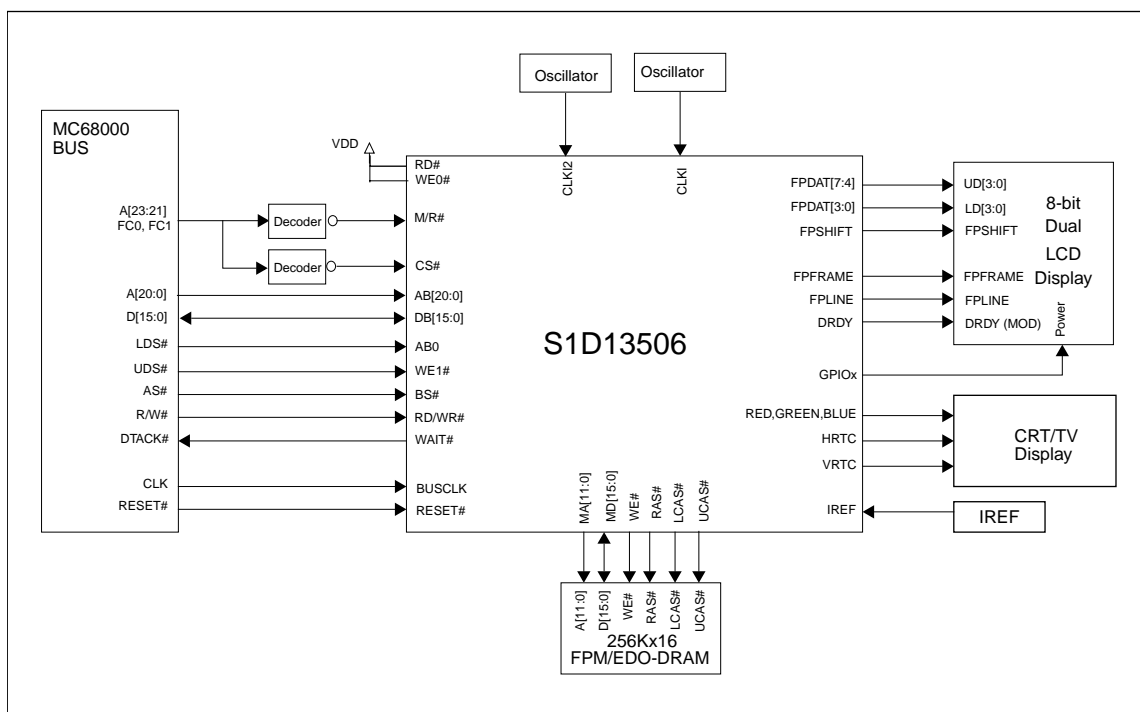
Figure 3-4 Typical System Diagram (MC68K Bus 1)
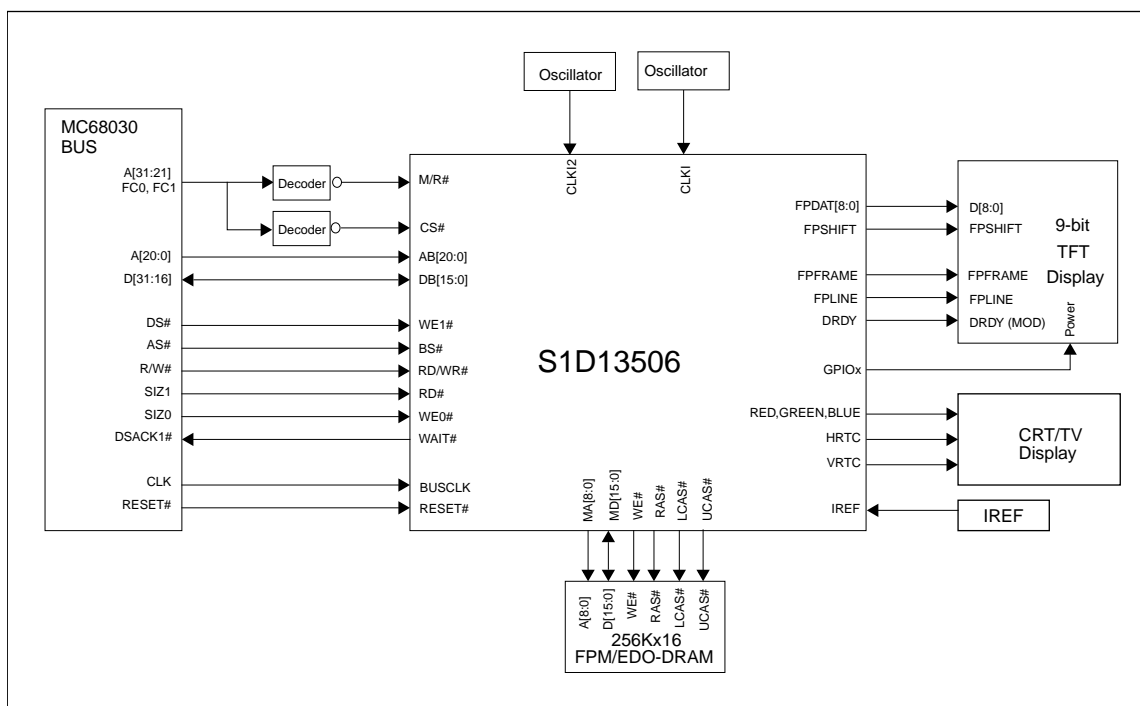
Figure 3-5 Typical System Diagram (MC68K Bus 2, Motorola 32-Bit 68030)
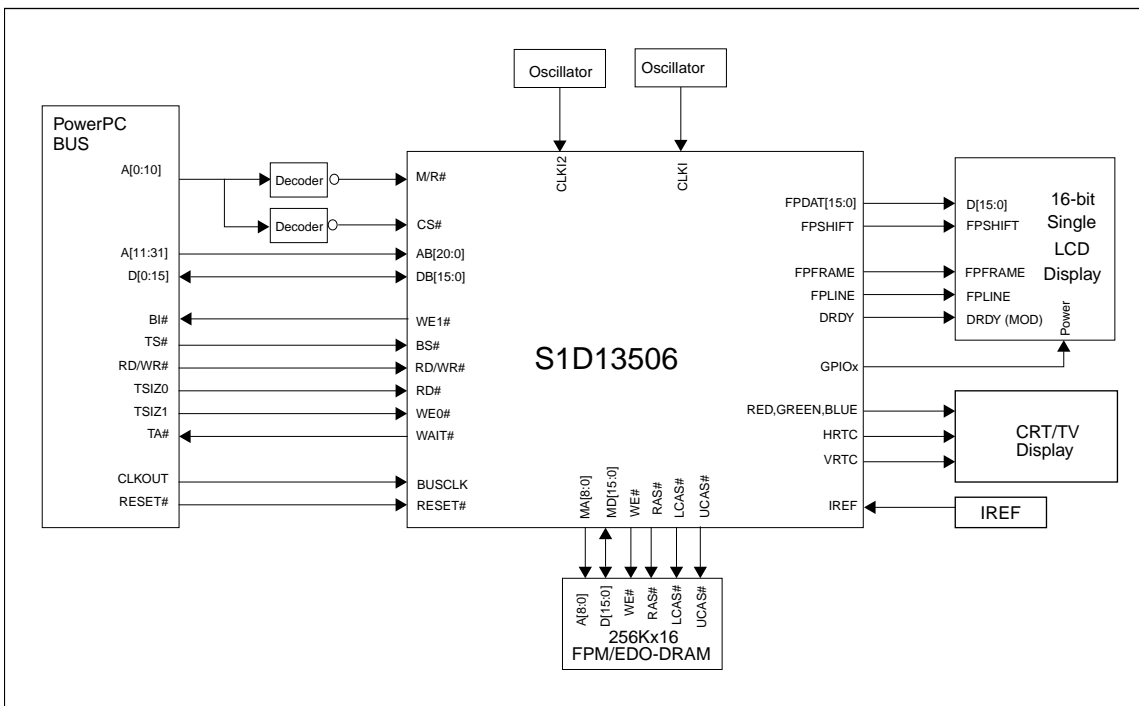
Figure 3-6 Typical System Diagram (Motorola PowerPC Bus)
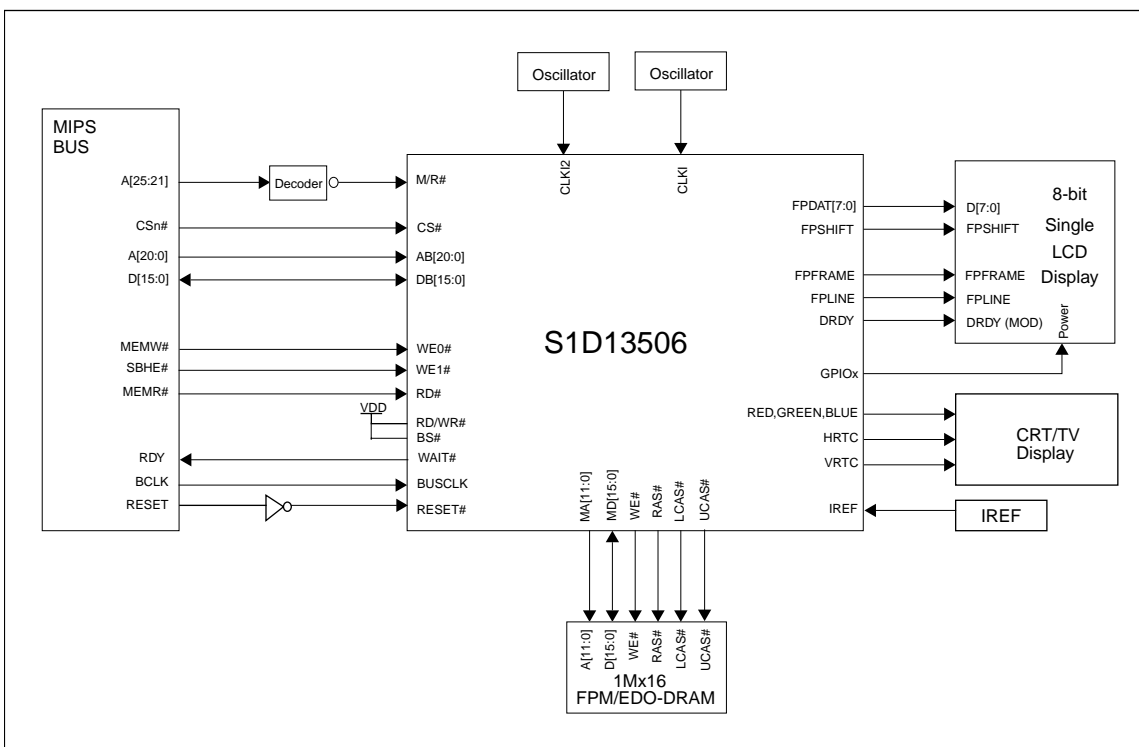
Figure 3-7 Typical System Diagram (NECVR41xx MIPS Bus)

Figure 3-8 Typical System Diagram (PC Card Bus)
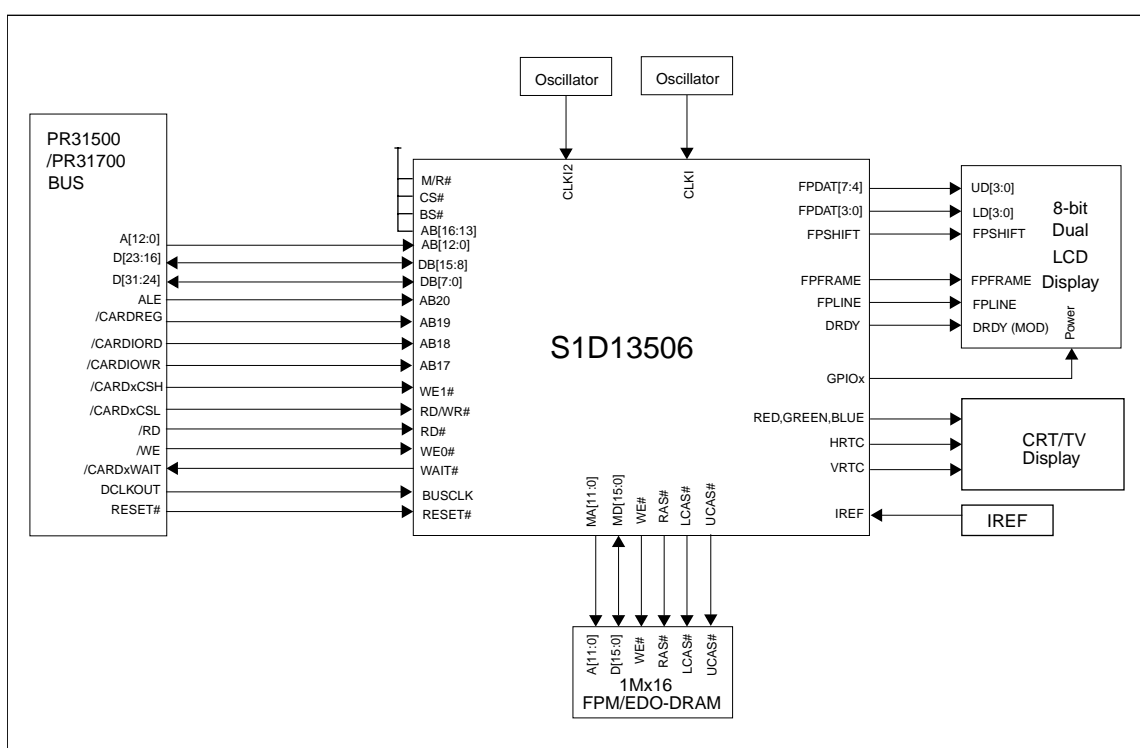
Figure 3-9 Typical System Diagram (Philips MIPS PR31500/PR31700 Bus)

Figure 3-10 Typical System Diagram (Toshiba MIPS TX3912 Bus)

4 INTERNAL DESCRIPTION

4.1 Block Diagram Showing Pipelines
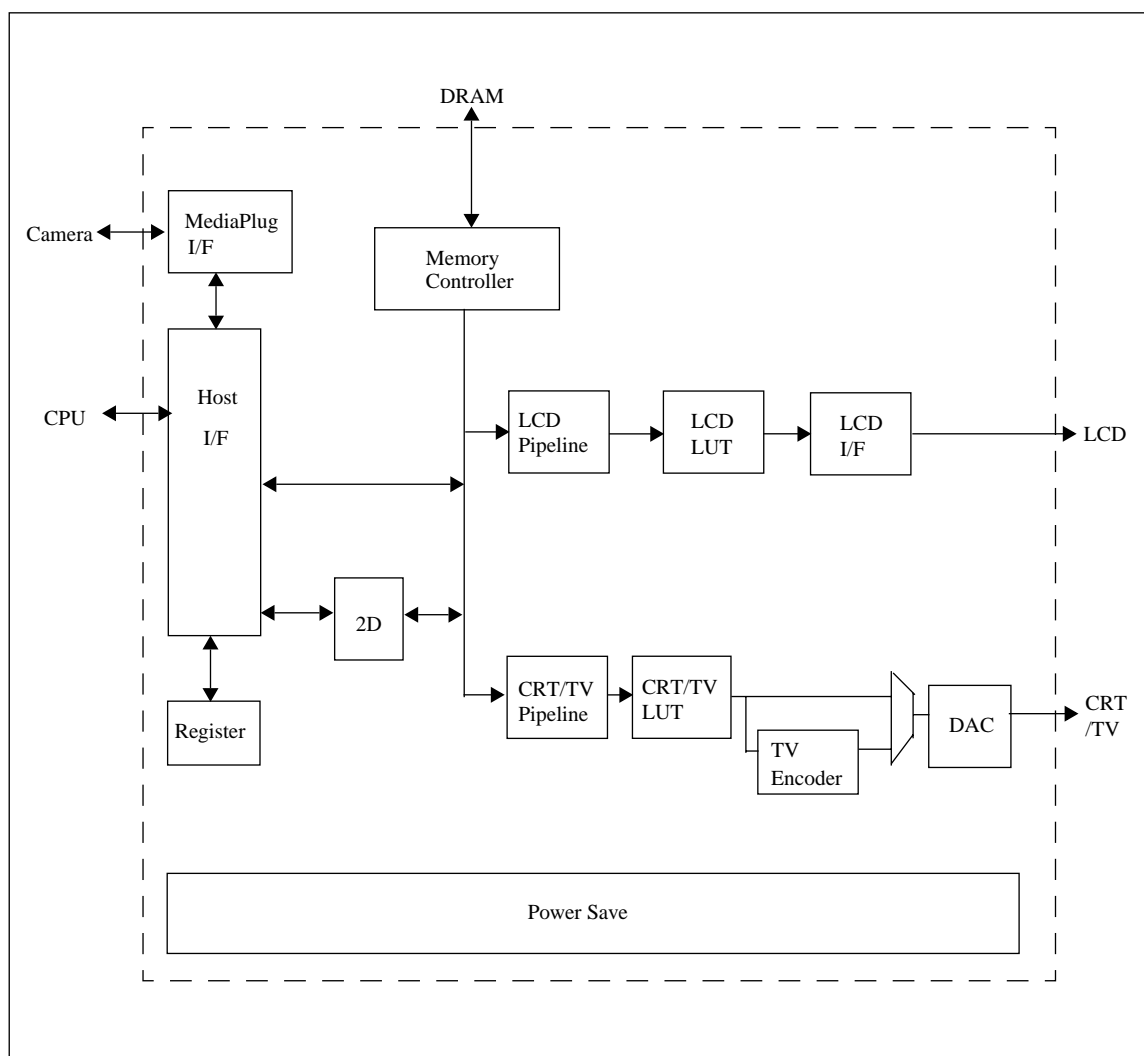
Figure 4-1 S1D13506 Block Diagram

5 PINS

5.1 Pinout Diagram
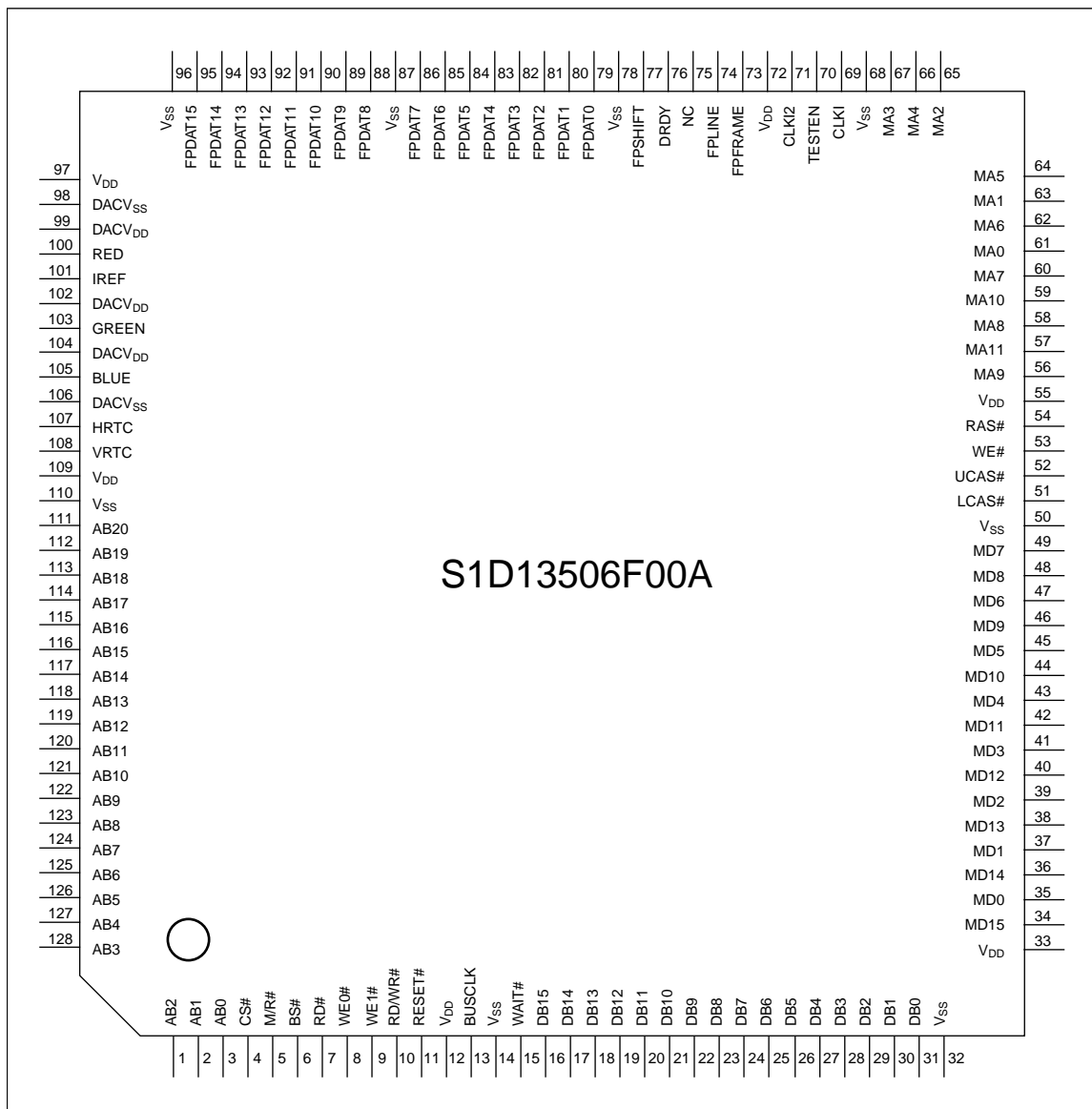
Figure 5-1 Pinout Diagram

128-pin QFP15 surface mount package

5.2 Pin Description

Key:

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 (Typ. values of 50kΩ/90kΩ at 5V/3.3V respectively)
CS	=	CMOS level Schmitt input
COx	=	CMOS output driver, x denotes driver type (1=4/-4mA, 2=8/-8mA, 3=12/-12mA at 5V)
TSx	=	Tri-state CMOS output driver, x denotes driver type (1=4/-4mA, 2=8/-8mA, 3=12/-12mA at 5V), x denotes driver type (1=4/-4mA, 2=8/-8mA, 3=12/-12mA at 5V)
TSu	=	TSx with pull up resistor (Typ. values of 100kΩ/180kΩ at 5V/3.3V respectively)
TSxD	=	TSx with pull down resistor, x denotes driver type (1=4/-4mA, 2=8/-8mA, 3=12/-12mA at 5V) (Typ. values of 100kΩ/180kΩ at 5V/3.3V)
CNx	=	CMOS low-noise output driver, x denotes driver type (1=4/-4mA, 2=8/-8mA, 3=12/-12mA at 5V)
CNxU	=	CNx with pull up resistor, x denotes driver type (1=4/-4mA, 2=8/-8mA, 3=12/-12mA at 5V)
CNxD	=	CNx with pull down resistor, x denotes driver type (1=4/-4mA, 2=8/-8mA, 3=12/-12mA at 5V)

5.2.1 Host Bus Interface

Table 5-1 Host Bus Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
AB0	I	3	CS	Hi-Z	• For SH-3/SH-4 Bus, this pin must be connected to V_{SS} or V_{DD}. • 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 must be connected to V_{SS} or V_{DD}. • 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 TX3912 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 must be connected to V_{SS} or V_{DD}. SeeTable 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
AB[12:1]	I	119-128, 1, 2	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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.

Table 5-1 Host Bus Interface Pin Descriptions (Continued)

Pin Name	Type	Pin #	Cell	RESET# State	Description
AB[16:13]	I	115-118	C	Hi-Z	• For Philips PR31500/31700 Bus, these pins are connected to V_{DD}. • For Toshiba TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
AB17	I	114	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin inputs the IO write command (/CARDIOWR). • For Toshiba TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
AB18	I	113	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin inputs the IO read command (/CARDIORD). • For Toshiba TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
AB19	I	112	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin inputs the card control register access (/CARDREG). • For Toshiba TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
AB20	I	111	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 TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.

Table 5-1 Host Bus Interface Pin Descriptions (Continued)

Pin Name	Type	Pin #	Cell	RESET# State	Description
DB[15:0]	IO	16-31	C/TS2	Hi-Z	These pins are the system data bus. For 8-bit bus modes, unused data pins should be tied to V_{DD}. • 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 TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
WE1#	IO	9	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 TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
M/R#	I	5	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin is connected to V_{DD}. • For Toshiba TX3912 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 S1D13506. M/R# is set high to access the display buffer and low to access the registers. See *Register Mapping*. See Table 5-7, "CPU Interface Pin Mapping" on page 1-24.
CS#	I	4	C	Hi-Z	• For Philips PR31500/31700 Bus, this pin is connected to V_{DD}. • For Toshiba TX3912 Bus, this pin is connected to V_{DD}. • For all other busses, this is the Chip Select input. See Table 5-7, "CPU Interface Pin Mapping" on page 1-24. See the respective AC Timing diagram for detailed functionality.

Table 5-1 Host Bus Interface Pin Descriptions (Continued)

Pin Name	Type	Pin #	Cell	RESET# State	Description
BUSCLK	I	13	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 MD12 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 TX3912 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). See Table 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
BS#	I	6	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 TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
RD/WR#	I	10	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 S1D13506 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 TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.

Table 5-1 Host Bus Interface Pin Descriptions (Continued)

Pin Name	Type	Pin #	Cell	RESET# State	Description
RD#	I	7	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 TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
WE0#	I	8	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 TX3912 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 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.

Table 5-1 Host Bus Interface Pin Descriptions (Continued)

Pin Name	Type	Pin #	Cell	RESET# State	Description
WAIT#	O	15	TS2	Hi-Z*a or 1*b or 0*c	The active polarity of the WAIT# output is configurable; the state of MD5 on the rising edge of RESET# defines the active polarity of WAIT# - see *"Summary of Configuration Options"*. • For SH-3 Bus, this pin outputs the wait request signal (WAIT#); MD5 must be pulled low during reset by the internal pull-down resistor. • For SH-4 Bus, this pin outputs the ready signal (RDY#); MD5 must be pulled high during reset by an external pull-up resistor. • For MC68K Bus 1, this pin outputs the data transfer acknowledge signal (DTACK#); MD5 must be pulled high during reset by an external pull-up resistor. • For MC68K Bus 2, this pin outputs the data transfer and size acknowledge bit 1 (DSACK1#); MD5 must be pulled high during reset by an external pull-up resistor. • For Generic Bus, this pin outputs the wait signal (WAIT#); MD5 must be pulled low during reset by the internal pull-down resistor. • For MIPS/ISA Bus, this pin outputs the IO channel ready signal (IOCHRDY); MD5 must be pulled low during reset by the internal pull-down resistor. • For Philips PR31500/31700 Bus, this pin outputs the wait state signal (/CARDxWAIT). MD5 must be pulled low during reset by the internal pull-down resistor. • For Toshiba TX3912 Bus, this pin outputs the wait state signal (CARDxWAIT*). MD5 must be pulled low during reset by the internal pull-down resistor. • For PowerPC Bus, this pin outputs the transfer acknowledge signal (TA#); MD5 must be pulled high during reset by an external pull-up resistor. • For PC Card (PCMCIA) Bus, this pin outputs the wait signal (-WAIT); MD5 must be pulled low during reset by the internal pull-down resistor. See Table 5-7, "CPU Interface Pin Mapping" on page 1-24 for summary. See the respective AC Timing diagram for detailed functionality.
RESET#	I	11	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.

* a When the MD configuration at RESET# is set such that WAIT# can be tristated.

   b When the MD configuration at RESET# is set such that WAIT# is always driven and active low.

   c When the MD configuration at RESET# is set such that WAIT# is always driven and active high.

5.2.2 Memory Interface

Table 5-2 Memory Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
LCAS#	O	51	CO1	1	• For dual-CAS# DRAM, this is the column address strobe for the lower byte (LCAS#). • For single-CAS DRAM, this is the column address strobe (CAS#). See Table 5-8, "Memory Interface Pin Mapping" on page 1-25 for summary. See "Memory Interface Timing" on page 1-53 for detailed functionality.
UCAS#	O	52	CO1	1	This is a multi-purpose pin: • For dual-CAS# DRAM, this is the column address strobe for the upper byte (UCAS#). • For single-CAS# DRAM, this is the write enable signal for the upper byte (UWE#). See Table 5-8, "Memory Interface Pin Mapping" on page 1-25 for summary. See "Memory Interface Timing" on page 1-53 for detailed functionality.
WE#	O	53	CO1	1	• For dual-CAS# DRAM, this is the write enable signal (WE#). • For single-CAS# DRAM, this is the write enable signal for the lower byte (LWE#). See Table 5-8, "Memory Interface Pin Mapping" on page 1-25 for summary. See "Memory Interface Timing" on page 1-53 for detailed functionality.
RAS#	O	54	CO1	1	Row address strobe - see "Memory Interface Timing" on page 1-53 for detailed functionality.
MD[15:0]	IO	34, 36, 38, 40, 42, 44, 46, 48, 49, 47, 45, 43, 41, 39, 37, 35	C/ TS1D	Hi-Z (pull 0)	Bi-directional memory data bus. During reset, these pins are inputs and their states at the rising edge of RESET# are used to configure the chip - see "Summary of Configuration Options" on page 1-23. Internal pull-down resistors (Typ. values of 100kΩ/180kΩ at 5V/3.3V respectively) pull the reset states to 0. External pull-up resistors can be used to pull the reset states to 1. See "Memory Interface Timing" on page 1-53 for detailed functionality.
MA[8:0]	O	58, 60, 62, 64, 66, 67, 65, 63, 61	CO1	0	Multiplexed memory address - see "Memory Interface Timing" on page 1-53 for detailed functionality.
MA9	IO	56	C/TS1	0*a or Hi-Z*b	This is a multi-purpose pin: • For 2M byte DRAM, this is memory address bit 9 (MA9). • For asymmetrical 512K byte DRAM, this is memory address bit 9 (MA9). • For symmetrical 512K byte DRAM, this pin can be used as general purpose IO pin 3 (GPIO3). Note that unless configured otherwise, this pin defaults to an input and must be driven to a valid logic level. See Table 5-8, "Memory Interface Pin Mapping" on page 1-25 for summary. See "Memory Interface Timing" on page 1-53 for detailed functionality.
MA10	IO	59	C/TS1	0*c or Hi-Z*d	This is a multi-purpose pin: • For asymmetrical 2M byte DRAM this is memory address bit 10 (MA10). • For symmetrical 2M byte DRAM and all 512K byte DRAM this pin can be used as general purpose IO pin 1 (GPIO1). Note that unless configured otherwise, this pin defaults to an input and must be driven to a valid logic level. See Table 5-8, "Memory Interface Pin Mapping" on page 1-25 for summary. See "Memory Interface Timing" on page 1-53 for detailed functionality.

EPSON

Table 5-2 Memory Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
MA11	IO	57	C/TS1	0*e or Hi-Z*f or 1*g	This is a multi-purpose pin: • For asymmetrical 2M byte DRAM this is memory address bit 11 (MA11). • For symmetrical 2M byte DRAM and all 512K byte DRAM this pin can be used as general purpose IO pin 2 (GPIO2). • Note that unless configured otherwise, this pin defaults to an input and must be driven to a valid logic level. See Table 5-8, "Memory Interface Pin Mapping" on page 1-25 for summary. See "Memory Interface Timing" on page 1-53 for detailed functionality. This pin can also be configured as the MediaPlug power pin VMPEPWR - see Table 5-10, "MA11, MA10, MA9, and DRDY Pin Mapping" on page 1-27 for details.

* a When the MD configuration at RESET# is set such that MA9 is used as MA9.
b When the MD configuration at RESET# is set such that MA9 is used as GPIO3.
c When the MD configuration at RESET# is set such that MA10 is used as MA10.
d When the MD configuration at RESET# is set such that MA10 is used as GPIO1.
e When the MD configuration at RESET# is set such that MA11 is used as MA11.
f When the MD configuration at RESET# is set such that MA11 is used as GPIO2.
g When the MD configuration at RESET# is set such that MA11 is used as VMPEPWR.

5.2.3 LCD Interface

Table 5-3 LCD Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
FPDAT[8:0]	O	88, 86-79	CN3	0	Panel data bus. Not all pins are used for some panels - see Table 5-9, "LCD Interface Pin Mapping" on page 1-26 for details. Unused pins are driven low. FPDAT[15:8] can be configured for MediaPlug interface - see Table 5-11, "MediaPlug Interface Pin Mapping" on page 1-27 for details.
FPDAT9	O	89	CN3D	0*a or Hi-Z*b	Panel data bus. Not all pins are used for some panels - see Table 5-9, "LCD Interface Pin Mapping" on page 1-26 for details. Unused pins are driven low. FPDAT[15:8] can be configured for MediaPlug interface - see Table 5-11, "MediaPlug Interface Pin Mapping" on page 1-27 for details.
FPDAT[13:10]	IO	93-90	C/ TS3U	0*c or Hi-Z*d	Panel data bus. Not all pins are used for some panels - see Table 5-9, "LCD Interface Pin Mapping" on page 1-26 for details. Unused pins are driven low. FPDAT[15:8] can be configured for MediaPlug interface - see Table 5-11, "MediaPlug Interface Pin Mapping" on page 1-27 for details.
FPDAT[15:14]	O	95,94	CN3	0	Panel data bus. Not all pins are used for some panels - see Table 5-9, "LCD Interface Pin Mapping" on page 1-26 for details. Unused pins are driven low. FPDAT[15:8] can be configured for MediaPlug interface - see Table 5-11, "MediaPlug Interface Pin Mapping" on page 1-27 for details.
FPFRAME	O	73	CN3	0	Frame pulse
FPLINE	O	74	CN3	0	Line pulse
FPSHIFT	O	77	CO3	0	Shift clock
DRDY	O	76	CO3	0*e or 1*f	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 5-9, "LCD Interface Pin Mapping" on page 1-26 and REG[030h] for details. This pin can also be configured as the MediaPlug power pin VMPEPWR - see Table 5-10, "MA11, MA10, MA9, and DRDY Pin Mapping" on page 1-27 for details.

* a When the MD configuration at RESET# is set such that FPDAT9 is used as FPDAT9.
  b When the MD configuration at RESET# is set such that FPDAT9 is used as VMPRCTL.
  c When the MD configuration at RESET# is set such that FPDAT[13:10] is used as FPDAT[13:10].
  d When the MD configuration at RESET# is set such that FPDAT[13:10] is used as VMPD[3:0].
  e When the MD configuration at RESET# is set such that DRDY is used as DRDY (MOD).
  f When the MD configuration at RESET# is set such that DRDY is used as VMPEPWR.

5.2.4 CRT Interface

Table 5-4 CRT Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
HRTC	O	107	CN3	0	Horizontal retrace signal for CRT
VRTC	O	108	CN3	0	Vertical retrace signal for CRT
RED	O	100	A	no output current	Analog output for CRT color Red / S-Video Luminance
GREEN	O	103	A	no output current	Analog output for CRT color Green / Composite Video Out
BLUE	O	105	A	no output current	Analog output for CRT color Blue / S-Video Chrominance
IREF	I	101	A	–	Current reference for DAC. This pin must be connected to V_{SS} if the DAC is not needed.

5.2.5 Miscellaneous

Table 5-5 Miscellaneous Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	RESET# State	Description
CLKI	I	69	C	–	Selectable input clock. Can be used for the internal pixel clock (PCLK), memory clock (MCLK), and MediaPlug Clock.
CLKI2	I	71	C	–	Selectable input clock. Can be used for the internal pixel clock (PCLK) and MediaPlug Clock.
TESTEN	I	70	CD	–	Test Enable. This pin should be connected to V_{SS} for normal operation.
V_{DD}	P	12, 33, 55, 72, 97, 109	P	–	V_{DD}
DACV$_{DD}$	P	99, 102, 104	P	–	DAC V_{DD}
V_{SS}	P	14, 32, 50, 68, 78, 87, 96, 110	P	–	V_{SS}
DACV$_{SS}$	P	98, 106	P	–	DAC V_{SS}
NC	-	75		–	Not connected

5.3 Summary of Configuration Options

Table 5-6 Summary of Power-On/Reset Options

Pin Name	value of this pin at rising edge of RESET# is used to configure:(1/0)	
	1	0
MD0	Not used, value of this pin at rising edge of RESET# can be read at REG[00Ch] bit 0	
MD11, MD[3:1]	Select Host Bus Interface as follows : **MD11 MD3 MD2 MD1 Host Bus** 0 0 0 0 SH-4/SH-3 Bus interface 0 0 0 1 MC68K Bus 1 0 0 1 0 MC68K Bus 2 0 0 1 1 Generic 0 1 0 0 Reserved 0 1 0 1 MIPS/ISA 0 1 1 0 PowerPC 0 1 1 1 PC Card (PCMCIA) 1 1 1 1 Philips PR31500/PR31700 / Toshiba TX3912	
MD4	Little Endian	Big Endian
MD5	WAIT# is active high (1 = insert wait state)	WAIT# is active low (0 = insert wait state)
MD[7:6]	Memory Address/GPIO configuration: (See Table 5-10, "MA11, MA10, MA9, and DRDY Pin Mapping" on page 1-27) 00 = symmetrical 256K×16 DRAM. MA[8:0] = DRAM address. MA[11:9] can be used as GPIO2,1,3 pins. 01 = symmetrical 1M×16 DRAM. MA[9:0] = DRAM address. MA[11:10] can be used as GPIO2,1 pins. 10 = asymmetrical 256K×16 DRAM. MA[9:0] = DRAM address. MA[11:10] can be used as GPIO2,1 pins. 11 = asymmetrical 1M×16 DRAM. MA[11:0] = DRAM address.	
MD8	Not used, value of this pin at rising edge of RESET# can be read at REG[00Dh] bit 0	
MD9	Not used, value of this pin at rising edge of RESET# can be read at REG[00Dh] bit 1	
MD10	Not Used, value of this pin at rising edge of RESET# can be read at REG[00Dh] bit 2	
MD11	Alternate Host Bus Interface Selected	Primary Host Bus Interface Selected
MD12	BUSCLK input divided by 2	BUSCLK input not divided
MD13	Configure FPDAT[15:8] for MediaPlug I/F. External latches required to support 16-bit passive panels.	Support 16-bit passive panels directly
MD14	DRDY or MA11 is configured as MediaPlug power down pin (VMPEPWR). (See Table 5-10, "MA11, MA10, MA9, and DRDY Pin Mapping" on page 1-27)	DRDY is configured as a normal LCD I/F output pin. MA11 is configured as either a memory address or GPIO2. (See Table 5-10, "MA11, MA10, MA9, and DRDY Pin Mapping" on page 1-27)
MD15	WAIT# is always driven	WAIT# is tristated when the chip is not accessed by the host

5.4 Multiple Function Pin Mapping

Table 5-7 CPU Interface Pin Mapping

S1D13506 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}	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	Connected to V_{DD}[1]	Connected to V_{DD}[1]	SA0	LDS#	A0	A31	Connected to V_{DD}[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#	$\overline{BI}$	CE2#	/CARDxCSH	CARDxCSH*
M/R#	External Decode							connected to V_{DD}	connected to V_{DD}
CS#	External Decode							connected to V_{DD}	connected to V_{DD}
BUSCLK	BCLK	CKIO	CLK	CLK	CLK	CLKOUT	External Oscillator[2]	DCLKOUT	DCLKOUT
BS#	connected to V_{DD}	BS#	connected to V_{DD}	AS#	AS#	$\overline{TS}$	connected to V_{DD}	connected to V_{DD}	connected to V_{DD}
RD/WR#	RD1#	RD/WR#	connected to V_{DD}	R/W#	R/W#	$\overline{RD/WR}$	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#	RDY# /WAIT#	IOCHRDY	DTACK#	DSACK1#	$\overline{TA}$	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, must be connected to either V_{SS} or IO V_{DD} if not used.

[1]: AB0 is not used internally for these busses and must be connected to either V_{SS} or V_{DD}.

[2]: For further information on interfacing the S1D13506 to the PC Card bus, see Interfacing to the PC Card Bus, document number X25B-G-005-xx

Table 5-8 Memory Interface Pin Mapping

S1D13506 Pin Names	FPM/EDO-DRAM							
	Sym 256Kx16		Asym 256Kx16		Sym 1Mx16		Asym 1Mx16	
	2-CAS#	2-WE#	2-CAS#	2-WE#	2-CAS#	2-WE#	2-CAS#	2-WE#
MD[15:0]	D[15:0]							
MA[8:0]	A[8:0]							
MA9*1	GPIO3*2		A9				A9	
MA10*1	GPIO1 2						A10	
MA11*1	GPIO2 2						A11	
UCAS#	UCAS#	UWE#	UCAS#	UWE#	UCAS#	UWE#	UCAS#	UWE#
LCAS#	LCAS#	CAS#	LCAS#	CAS#	LCAS#	CAS#	LCAS#	CAS#
WE#	WE#	LWE#	WE#	LWE#	WE#	LWE#	WE#	LWE#
RAS#	RAS#							

*1 For MA9, MA10, and MA11 functionality see Table 5-10, "MA11, MA10, MA9, and DRDY Pin Mapping" on page 1-27.

*2 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 5-9 LCD Interface Pin Mapping

S1D13506 Pin Names	Monochrome Passive Panel			Color Passive Panel						Color TFT/D-TFD 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
FPFRAME	FPFRAME											
FPLINE	FPLINE											
FPSHIFT	FPSHIFT											
DRDY	MOD			FPSHIFT2	MOD					DRDY		
FPDAT0	driven 0	D0	LD0	driven 0	D0	D0	D0	LD0	LD0	R2	R3	R5
FPDAT1	driven 0	D1	LD1	driven 0	D1	D1	D1	LD1	LD1	R1	R2	R4
FPDAT2	driven 0	D2	LD2	driven 0	D2	D2	D2	LD2	LD2	R0	R1	R3
FPDAT3	driven 0	D3	LD3	driven 0	D3	D3	D3	LD3	LD3	G2	G3	G5
FPDAT4	D0	D4	UD0	D0	D4	D4	D4	UD0	UD0	G1	G2	G4
FPDAT5	D1	D5	UD1	D1	D5	D5	D5	UD1	UD1	G0	G1	G3
FPDAT6	D2	D6	UD2	D2	D6	D6	D6	UD2	UD2	B2	B3	B5
FPDAT7	D3	D7	UD3	D3	D7	D7	D7	UD3	UD3	B1	B2	B4
FPDAT8	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D8	driven 0	LD4	B0	B1	B3
FPDAT9	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D9	driven 0	LD5	driven 0	R0	R2
FPDAT10	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D10	driven 0	LD6	driven 0	driven 0	R1
FPDAT11	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D11	driven 0	LD7	driven 0	G0	G2
FPDAT12	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D12	driven 0	UD4	driven 0	driven 0	G1
FPDAT13	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D13	driven 0	UD5	driven 0	driven 0	G0
FPDAT14	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D14	driven 0	UD6	driven 0	B0	B2
FPDAT15	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	D15	driven 0	UD7	driven 0	driven 0	B1

Note: DRDY and FPDAT[15:8] may be used by the MediaPlug interface when the MediaPlug is enabled. For MediaPlug Interface pin mapping, see Table 5-11.

EPSON

Table 5-10 MA11, MA10, MA9, and DRDY Pin Mapping

MD14, MD7, MD6	MA11	MA10	MA9	DRDY
000	GPIO2	GPIO1	GPIO3	DRDY
001	GPIO2	GPIO1	MA9	DRDY
010	GPIO2	GPIO1	MA9	DRDY
011	MA11	MA10	MA9	DRDY
100	VMPEPWR	GPIO1	GPIO3	DRDY
101	VMPEPWR	GPIO1	MA9	DRDY
110	VMPEPWR	GPIO1	MA9	DRDY
111	MA11	MA10	MA9	VMPEPWR

Table 5-11 MediaPlug Interface Pin Mapping

S1D13506 Pin Names	IO Type	MediaPlug I/F (MD13=1 at RESET)
FPDAT8	O	VMPLCTL
FPDAT9	I	VMPRCTL
FPDAT10	IO	VMPD0
FPDAT11	IO	VMPD1
FPDAT12	IO	VMPD2
FPDAT13	IO	VMPD3
FPDAT14	O	VMPCLK
FPDAT15	O	VMPCLKN
DRDY or MA11[1]	O	VMPEPWR

[1] Either DRDY or MA11 may be used for VMPEPWR (see Table 5-10, "MA11, MA10, MA9, and DRDY Pin Mapping"). If DRDY is required by the LCD interface and MA11 is required by the DRAM interface then VMPEPWR is not available.

5.5 *CRT/TV Interface*

The following figure shows external circuitry for the CRT/TV interface.
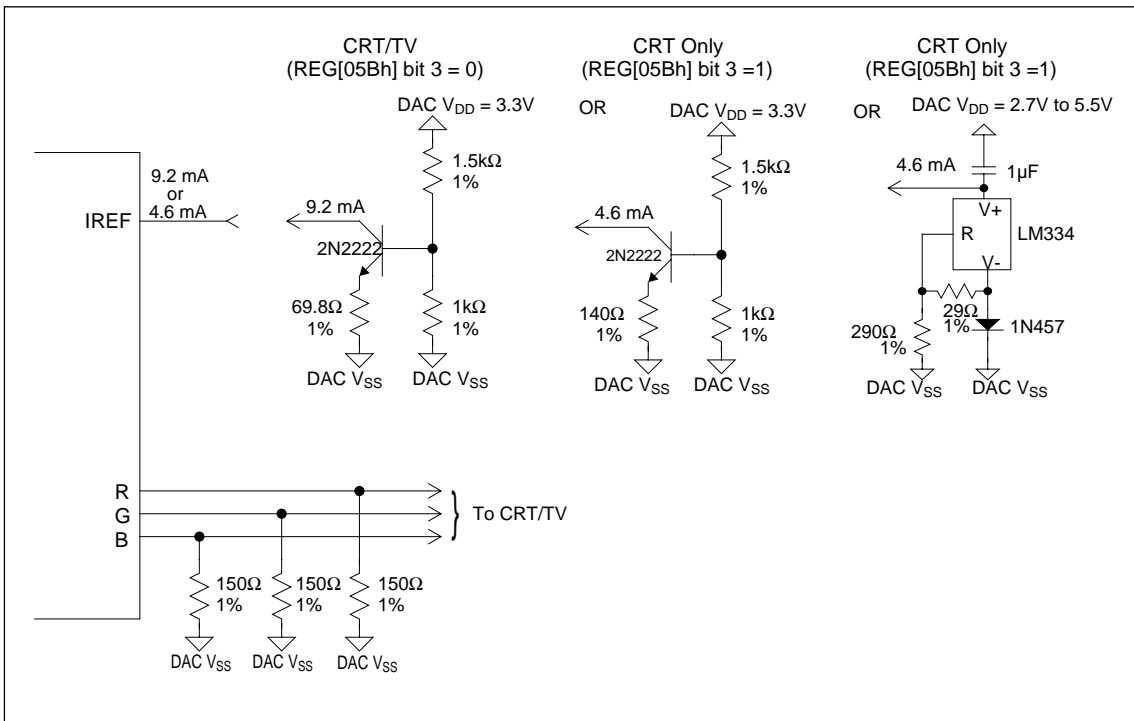
Figure 5-2 External Circuitry for CRT/TV Interface

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

EPSON   **S1D13506F00A HARDWARE FUNCTIONAL**
SPECIFICATION (X25B-A-001-08)

6 D.C. CHARACTERISTICS

Table 6-1 Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
V_{DD}	Supply Voltage	V_{SS} - 0.3 to 6.0	V
DAC V_{DD}	Supply Voltage	V_{SS} - 0.3 to 6.0	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 6-2 Recommended Operating Conditions

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{DD}	Supply Voltage	V_{SS} = 0 V	2.7	3.0/3.3/5.0	5.5	V
V_{IN}	Input Voltage		V_{SS}		V_{DD}	V
T_{OPR}	Operating Temperature		-40	25	85	° C

Table 6-3 Electrical Characteristics for V_{DD} = 5.0V typical

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{DDS}	Quiescent Current	Quiescent Conditions			400	uA
I_{IZ}	Input Leakage Current		-1		1	µA
I_{OZ}	Output Leakage Current		-1		1	µA
V_{OH}	High Level Output Voltage	V_{DD} = Min. I_{OL} = -4mA (Type1), -8mA (Type2) -12mA (Type3)	V_{DD} - 0.4			V
V_{OL}	Low Level Output Voltage	V_{DD} = Min. I_{OL} = 4mA (Type1), 8mA (Type2) 12mA (Type3)			0.4	V
V_{IH}	High Level Input Voltage	CMOS level, V_{DD} = Max.	3.5			V
V_{IL}	Low Level Input Voltage	CMOS level, V_{DD} = Min.			1.0	V
V_{T+}	High Level Input Voltage	CMOS Schmitt, V_{DD} = 5.0V			4.0	V
V_{T-}	Low Level Input Voltage	CMOS Schmitt, V_{DD} = 5.0V	0.8			V
V_{H1}	Hysteresis Voltage	CMOS Schmitt, V_{DD} = 5.0V	0.3			V
R_{PD}	Pull Down Resistance	$V_I = V_{DD}$	50	100	200	kΩ
C_I	Input Pin Capacitance				12	pF
C_O	Output Pin Capacitance				12	pF
C_{IO}	Bi-Directional Pin Capacitance				12	pF

Table 6-4 Electrical Characteristics for V_{DD} = 3.3V typical

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{DDS}	Quiescent Current	Quiescent Conditions			290	uA
I_{IZ}	Input Leakage Current		-1		1	μA
I_{OZ}	Output Leakage Current		-1		1	μA
V_{OH}	High Level Output Voltage	V_{DD} = Min. I_{OL} = -2mA (Type1), -4mA (Type2) -6mA (Type3)	V_{DD} - 0.3			V
V_{OL}	Low Level Output Voltage	V_{DD} = Min. I_{OL} = 2mA (Type1), 4mA (Type2) 6mA (Type3)			0.3	V
V_{IH}	High Level Input Voltage	CMOS level, V_{DD} = Max.	2.2			V
V_{IL}	Low Level Input Voltage	CMOS level, V_{DD} = Min.			0.8	V
V_{T+}	High Level Input Voltage	CMOS Schmitt, V_{DD} = 3.3V			2.4	V
V_{T-}	Low Level Input Voltage	CMOS Schmitt, V_{DD} = 3.3V	0.6			V
V_{H1}	Hysteresis Voltage	CMOS Schmitt, V_{DD} = 3.3V	0.1			V
R_{PD}	Pull Down Resistance	$V_I = V_{DD}$	90	180	360	kΩ
C_I	Input Pin Capacitance				12	pF
C_O	Output Pin Capacitance				12	pF
C_{IO}	Bi-Directional Pin Capacitance				12	pF

Table 6-5 Electrical Characteristics for V_{DD} = 3.0V typical

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{DDS}	Quiescent Current	Quiescent Conditions			260	uA
I_{IZ}	Input Leakage Current		-1		1	μA
I_{OZ}	Output Leakage Current		-1		1	μA
V_{OH}	High Level Output Voltage	V_{DD} = Min. I_{OL} = -1.8mA (Type1), -3.5mA (Type2) -5mA (Type3)	V_{DD} - 0.3			V
V_{OL}	Low Level Output Voltage	V_{DD} = Min. I_{OL} = 1.8mA (Type1), 3.5mA (Type2) 5mA (Type3)			0.3	V
V_{IH}	High Level Input Voltage	CMOS level, V_{DD} = Max.	2.0			V
V_{IL}	Low Level Input Voltage	CMOS level, V_{DD} = Min.			0.8	V
V_{T+}	High Level Input Voltage	CMOS Schmitt, V_{DD} = 3.0V			2.3	V
V_{T-}	Low Level Input Voltage	CMOS Schmitt, V_{DD} = 3.0V	0.5			V
V_{H1}	Hysteresis Voltage	CMOS Schmitt, V_{DD} = 3.0V	0.1			V
R_{PD}	Pull Down Resistance	$V_I = V_{DD}$	100	200	400	kΩ
C_I	Input Pin Capacitance				12	pF
C_O	Output Pin Capacitance				12	pF
C_{IO}	Bi-Directional Pin Capacitance				12	pF

7 A.C. CHARACTERISTICS

Conditions: $V_{DD} = 3.0V \pm 10\%$ and $V_{DD} = 5.0V \pm 10\%$
$T_A = -40°C$ to $85°C$
T_{rise} and T_{fall} for all inputs must be ≤ 5 ns (10% to 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)

7.1 CPU Interface Timing

7.1.1 Generic Timing
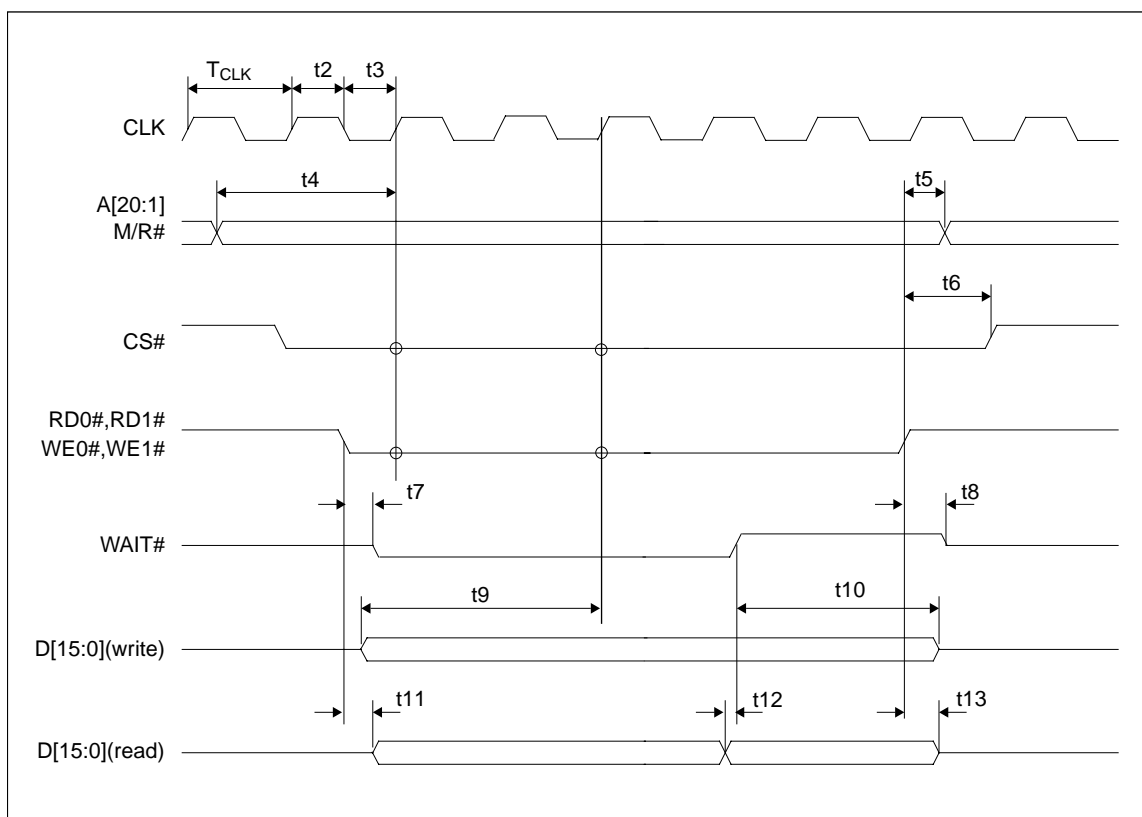
Figure 7-1 Generic Timing

Note: The above timing diagram is not applicable if MD12 = 1 (BUSCLK divided by 2).

Table 7-1 Generic Timing

Symbol	Parameter	3.0V		5.0V		Units
		Min.	Max.	Min.	Max.	
f_{CLK}	Clock frequency		50		50	MHz
T_{CLK}	Clock period	$1/f_{CLK}$		$1/f_{CLK}$		ns
t2	Clock pulse width high	6		6		ns
t3	Clock pulse width low	6		6		ns
t4	A[20:1], M/R# setup to first CLK where CS# = 0 and either RD0#, RD1#= 0 or WE0#, WE1#= 0	4		3		ns
t5	A[20:1], M/R# hold from rising edge of either RD0#, RD1# or WE0#, WE1#	0		0		ns
t6	CS# hold from rising edge of either RD0#, RD1# or WE0#, WE1#	0		0		ns
t7	Falling edge of either RD0#, RD1# or WE0#, WE1# to WAIT# driven low	4	21	3	13	ns
t8	Rising edge of either RD0#, RD1# or WE0#, WE1# to WAIT# tri-state	3	14	2	7	ns
t9	D[15:0] setup to third CLK where CS# = 0 and WE0#, WE1# = 0 (write cycle)	0		0		ns
t10	D[15:0] hold (write cycle)	0		0		ns
t11	Falling edge RD0#, RD1# to D[15:0] driven (read cycle)	3		3		ns
t12	D[15:0] setup to rising edge WAIT# (read cycle)	0		0		ns
t13	Rising edge of RD0#, RD1# to D[15:0] tri-state (read cycle)	7	31	4	15	ns

7.1.2 Hitachi SH-4 Interface Timing
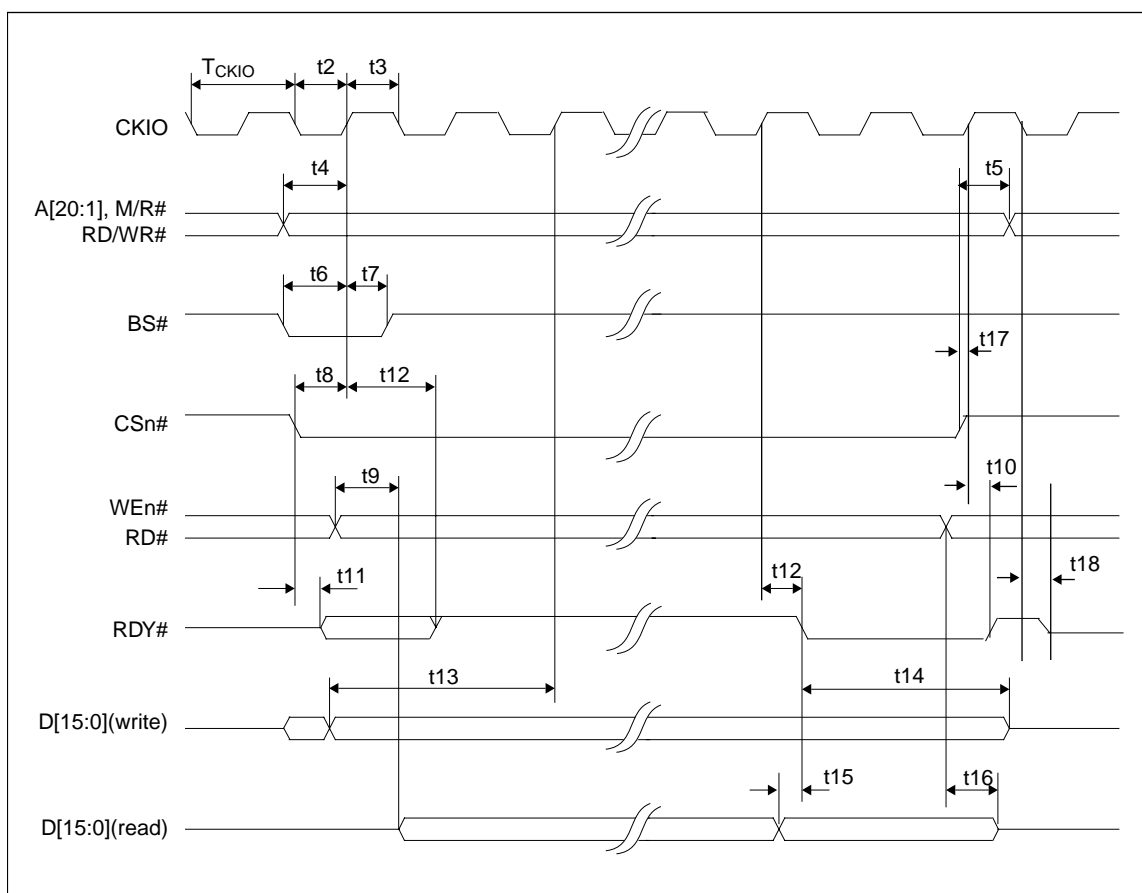
Figure 7-2 Hitachi SH-4 Timing

Note: 1. The above timing diagram is not applicable if MD12 = 1 (BUSCLK divided by 2).

2. The SH-4 Wait State Control Register for the area in which the S1D13506 resides must be set to a non-zero value. The SH-4 read-to-write idle cycle transition must be set to a non-zero value (with reference to BUSCLK).

Table 7-2 Hitachi SH-4 Timing

Symbol	Parameter	3.0V[1]		5.0V[2]		Units
		Min.	Max.	Min.	Max.	
f_{CKIO}	Clock frequency	0	66	0	66	MHz
T_{CKIO}	Clock period	$1/f_{CKIO}$		$1/f_{CKIO}$		
t2	Clock pulse width low	6		16		ns
t3	Clock pulse width high	6		6		ns
t4	A[20:1], M/R#, RD/WR# setup to CKIO	4		3		ns
t5	A[20:1], M/R#, RD/WR# hold from CSn#	0		0		ns
t6	BS# setup	4		3		ns
t7	BS# hold	3		2		ns
t8	CSn# setup	3		2		ns
t9	Falling edge RD# to D[15:0] driven	3		3		ns
t10	CKIO to RDY# high	4	21	3	13	ns
t11	Falling edge CSn# to RDY# driven	3	11	2	7	ns
t12	CKIO to RDY# delay	4	20	3	13	ns
t13	D[15:0] setup to 2nd CKIO after BS# (write cycle)	0		0		ns
t14	D[15:0] hold (write cycle)	0		0		ns
t15	D[15:0] valid to RDY# falling edge (read cycle)	0		0		ns
t16	Rising edge RD# to D[15:0] tri-state (read cycle)	6	30	3	16	ns
t17	CSn# high setup to CKIO	3		2		ns
t18	Falling edge CKIO to RDY# tri-state	3	14	2	10	ns

Note: 1. Two software WAIT states are required.

2. One software WAIT state is required

7.1.3 Hitachi SH-3 Interface Timing
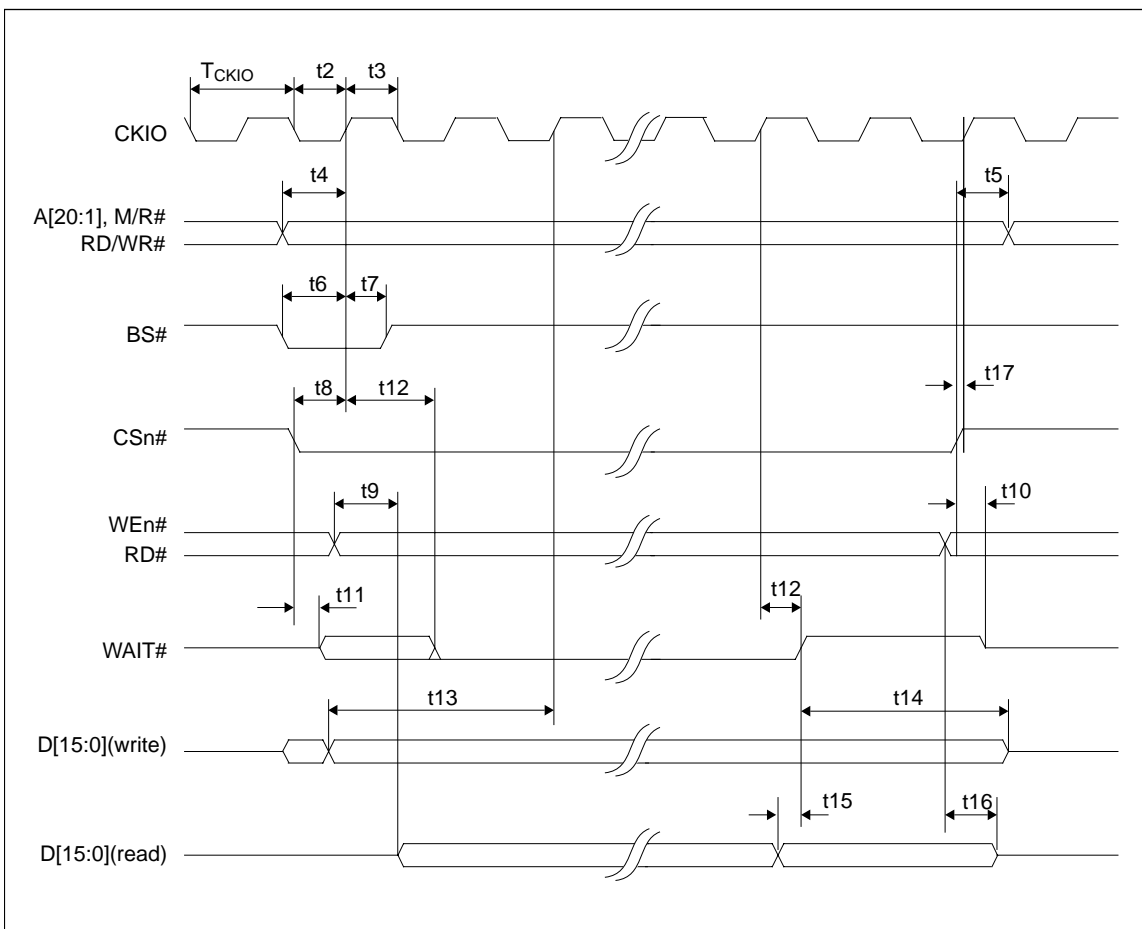
Figure 7-3 Hitachi SH-3 Timing

Note: 1. The above timing diagram is not applicable if MD12 = 1 (BUSCLK divided by 2).

2. The SH-3 Wait State Control Register for the area in which the S1D13506 resides must be set to a non-zero value.

Table 7-3 Hitachi SH-3 Timing

Symbol	Parameter	3.0V[1]		5.0V[2]		Units
		Min.	Max.	Min.	Max.	
f_{CKIO}	Clock frequency		66		66	MHz
T_{CKIO}	Clock period	$1/f_{CKIO}$		$1/f_{CKIO}$		ns
t2	Clock pulse width low	6		6		ns
t3	Clock pulse width high	6		6		ns
t4	A[20:1], M/R#, RD/WR# setup to CKIO	4		3		ns
t5	A[20:1], M/R#, RD/WR# hold from CSn#	0		0		ns
t6	BS# setup	4		3		ns
t7	BS# hold	3		2		ns
t8	CSn# setup	3		3		ns
t9	Falling edge RD# to D[15:0] driven	3		2		ns
t10	Rising edge CSn# to WAIT# tri-state	2	10	1	6	ns
t11	Falling edge CSn# to WAIT# driven	3	16	2	10	ns
t12	CKIO to WAIT# delay	4	20	3	13	ns
t13	D[15:0] setup to 2nd CKIO after BS# (write cycle)	0		0		ns
t14	D[15:0] hold (write cycle)	0		0		ns
t15	D[15:0] valid to WAIT# rising edge (read cycle)	0		0		ns
t16	Rising edge RD# to D[15:0] tri-state (read cycle)	6	30	3	15	ns
t17	CSn# high setup to CKIO	3		2		ns

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

2. One software WAIT state is required when f_{CKIO} is greater than 33MHz.

7.1.4 MIPS/ISA Interface Timing (e.g. NEC Vʀ41xx)
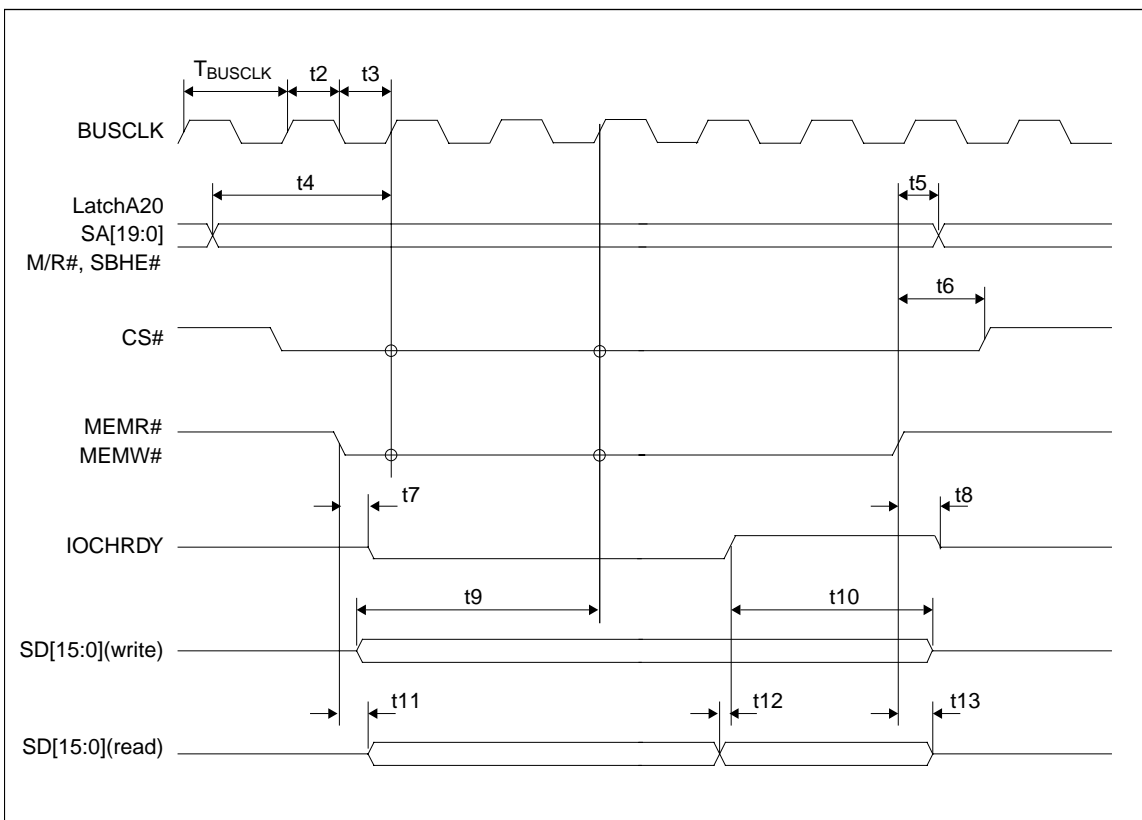
Figure 7-4 MIPS/ISA Timing

Note: The above timing diagram is not applicable if MD12 = 1 (BUSCLK divided by 2).

EPSON S1D13506F00A HARDWARE FUNCTIONAL
SPECIFICATION (X25B-A-001-08)

Table 7-4 MIPS/ISA Timing

Symbol	Parameter	3.0V		5.0V		Units
		Min.	Max.	Min.	Max.	
f_{BUSCLK}	Clock frequency		50		50	MHz
T_{BUSCLK}	Clock period	$1/f_{BUSCLK}$		$1/f_{BUSCLK}$		ns
t2	Clock pulse width high	6		6		ns
t3	Clock pulse width low	6		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		3		ns
t5	LatchA20, SA[19:0], M/R#, SBHE# hold from rising edge of either MEMR# or MEMW#	0		0		ns
t6	CS# hold from rising edge of either MEMR# or MEMW#	0		0		ns
t7	Falling edge of either MEMR# or MEMW# to IOCHRDY# driven low	2	17	2	10	ns
t8	Rising edge of either MEMR# or MEMW# to IOCHRDY# tri-state	2	12	1	7	ns
t9	SD[15:0] setup to third BUSCLK where CS# = 0 MEMW# = 0 (write cycle)	0		0		ns
t10	SD[15:0] hold (write cycle)	0		0		ns
t11	Falling edge MEMR# to SD[15:0] driven (read cycle)	4		3		ns
t12	SD[15:0] setup to rising edge IOCHRDY# (read cycle)	0		0		ns
t13	Rising edge of MEMR# toSD[15:0] tri-state (read cycle)	7	31	4	15	ns

7.1.5 Motorola MC68K Bus 1 Interface Timing (e.g. MC68000)
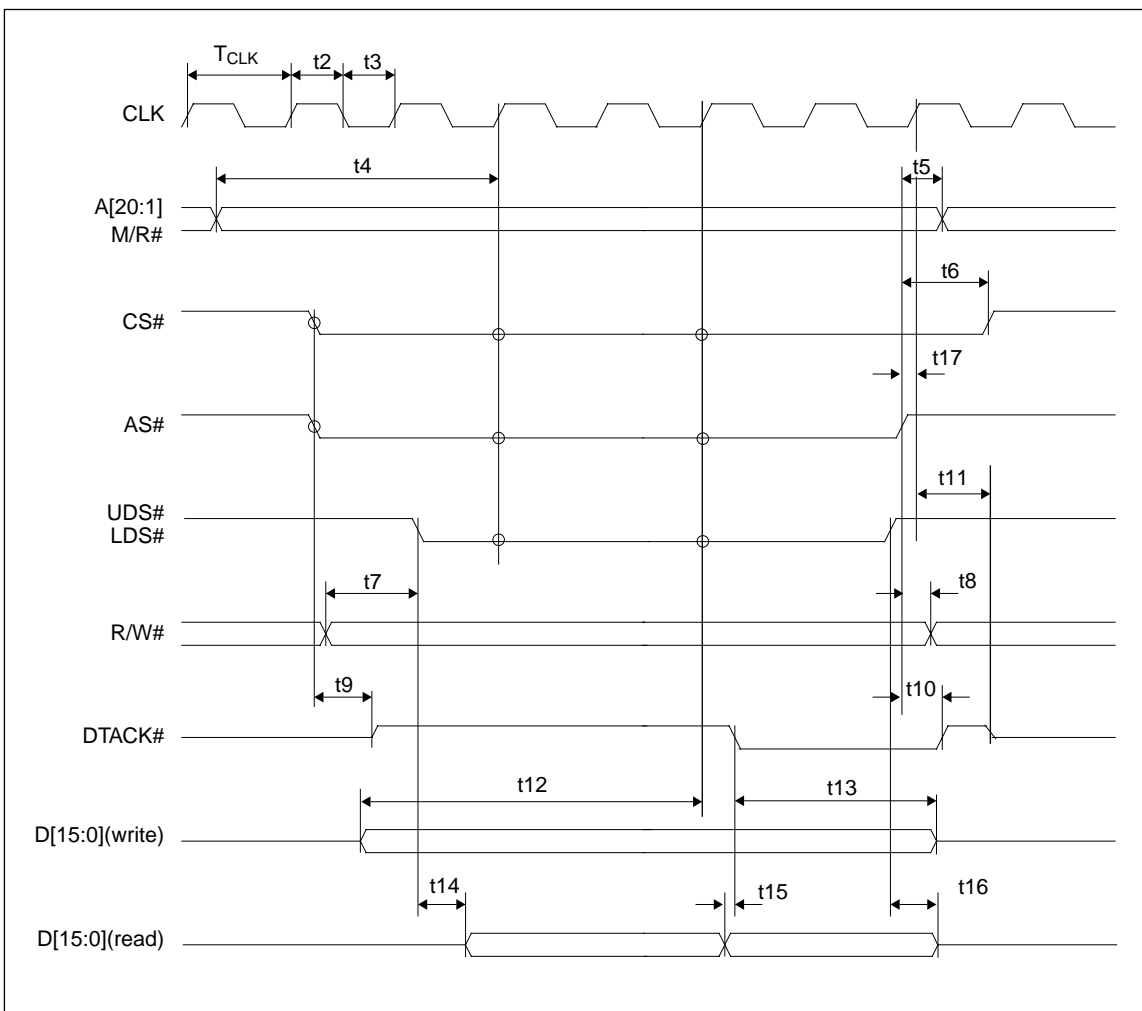
Figure 7-5 Motorola MC68000 Timing

Note: The above timing diagram is not applicable if MD12 = 1 (BUSCLK divided by 2).

Table 7-5 Motorola MC68000 Timing

Symbol	Parameter	3.0V		5.0V		Units
		Min.	Max.	Min.	Max.	
f_{CLK}	Clock frequency		50		50	MHz
T_{CLK}	Clock period	$1/f_{CLK}$		$1/f_{CLK}$		ns
t2	Clock pulse width high	6		6		ns
t3	Clock pulse width low	6		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		3		ns
t5	A[20:1], M/R# hold from AS#	0		0		ns
t6	CS# hold from AS#	0		0		ns
t7	R/W# setup to before to either UDS#=0 or LDS# = 0	10		10		ns
t8	R/W# hold from AS#	0		0		ns
t9	AS# = 0 and CS# = 0 to DTACK# driven high	1		1		ns
t10	AS# high to DTACK# high	4	18	3	11	ns
t11	First BCLK where AS# = 1 to DTACK# high impedance	3	15	2	10	ns
t12	D[15:0] valid to third CLK where CS# = 0 AS# = 0, and either UDS#=0 or LDS# = 0 (write cycle)	0		0		ns
t13	D[15:0] hold from falling edge of DTACK# (write cycle)	0		0		ns
t14	Falling edge of UDS#=0 or LDS# = 0 to D[15:0] driven (read cycle)	3		3		ns
t15	D[15:0] valid to DTACK# falling edge (read cycle)	0		0		ns
t16	UDS# and LDS# high to D[15:0] invalid/high impedance (read cycle)	6	31	4	15	ns
t17	AS# high setup to CLK	4		3		ns

7.1.6 Motorola MC68K Bus 2 Interface Timing (e.g. MC68030)
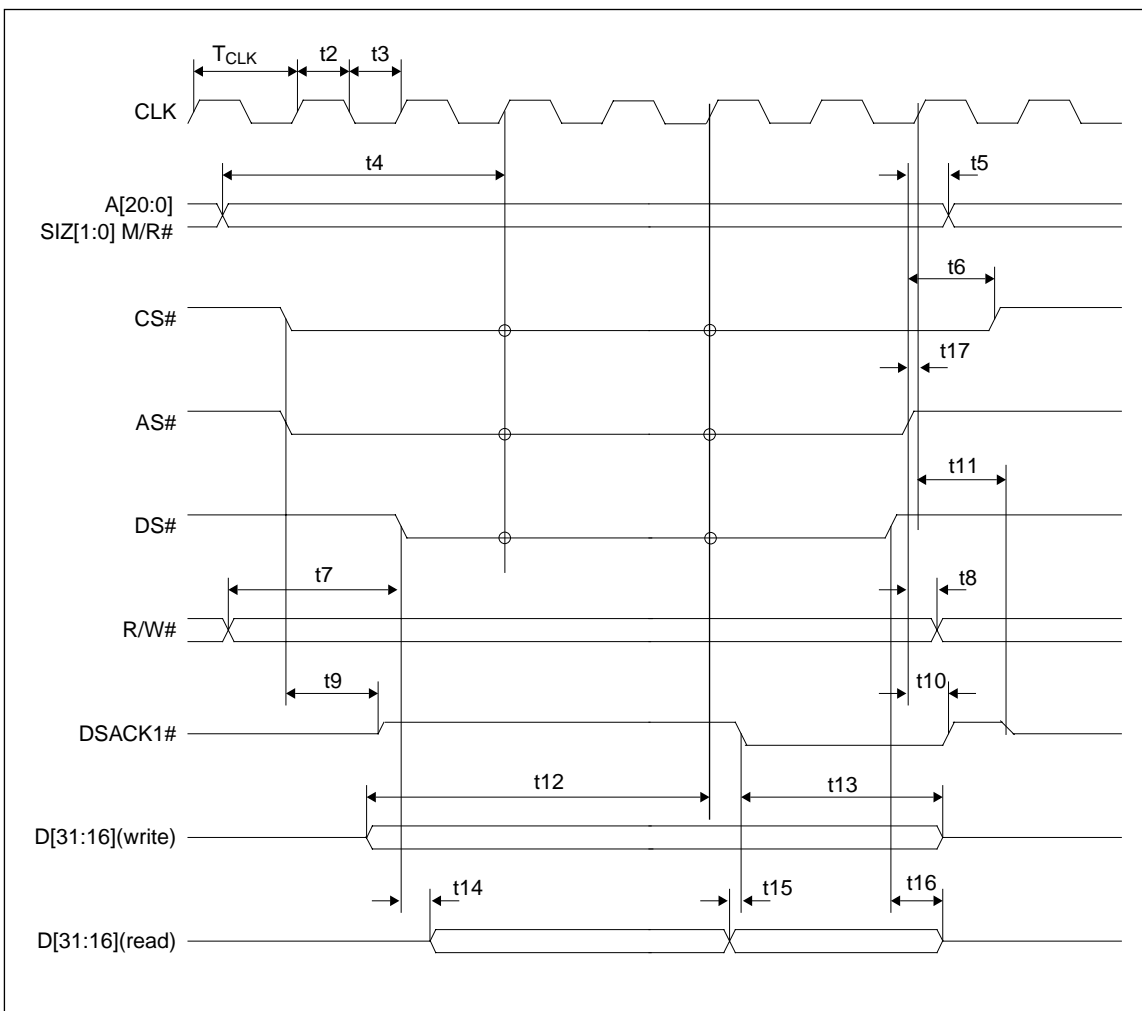
Figure 7-6 Motorola MC68030 Timing

Note: The above timing diagram is not applicable if MD12 = 1 (BUSCLK divided by 2).

Table 7-6 Motorola MC68030 Timing

Symbol	Parameter	3.0V		5.0V		Units
		Min.	Max.	Min.	Max.	
f_{CLK}	Clock frequency		50		50	MHz
T_{CLK}	Clock period	$1/f_{CLK}$		$1/f_{CLK}$		ns
t2	Clock pulse width high	6		6		ns
t3	Clock pulse width low	6		6		ns
t4	A[20:0], SIZ[1:0], M/R# setup to first CLK where CS# = 0, AS# = 0, and DS# = 0	5		3		ns
t5	A[20:0], SIZ[1:0], M/R# hold from AS#	0		0		ns
t6	CS# hold from AS#	0		0		ns
t7	R/W# setup to DS#	10		10		ns
t8	R/W# hold from AS#	0		0		ns
t9	AS# = 0 and CS# = 0 to DSACK1# driven high	1		1		ns
t10	AS# high to DSACK1# high	4	18	3	12	ns
t11	First BCLK where AS# = 1 to DSACK1# high impedance	3	15	2	14	ns
t12	D[31:16] valid to third CLK where CS# = 0, AS# = 0, and DS# = 0 (write cycle)	0		0		ns
t13	D[31:16] hold from falling edge of DSACK1# (write cycle)	0		0		ns
t14	Falling edge of DS# = 0 to D[31:16] driven (read cycle)	3		3		ns
t15	D[31:16] valid to DSACK1# falling edge (read cycle)	0		0		ns
t16	DS# high to D[31:16] invalid/high impedance (read cycle)	6	31	4	15	ns
t17	AS# high setup to CLK	4		3		ns

7.1.7 Motorola PowerPC Interface Timing (e.g. MPC8xx, MC68040, Coldfire)
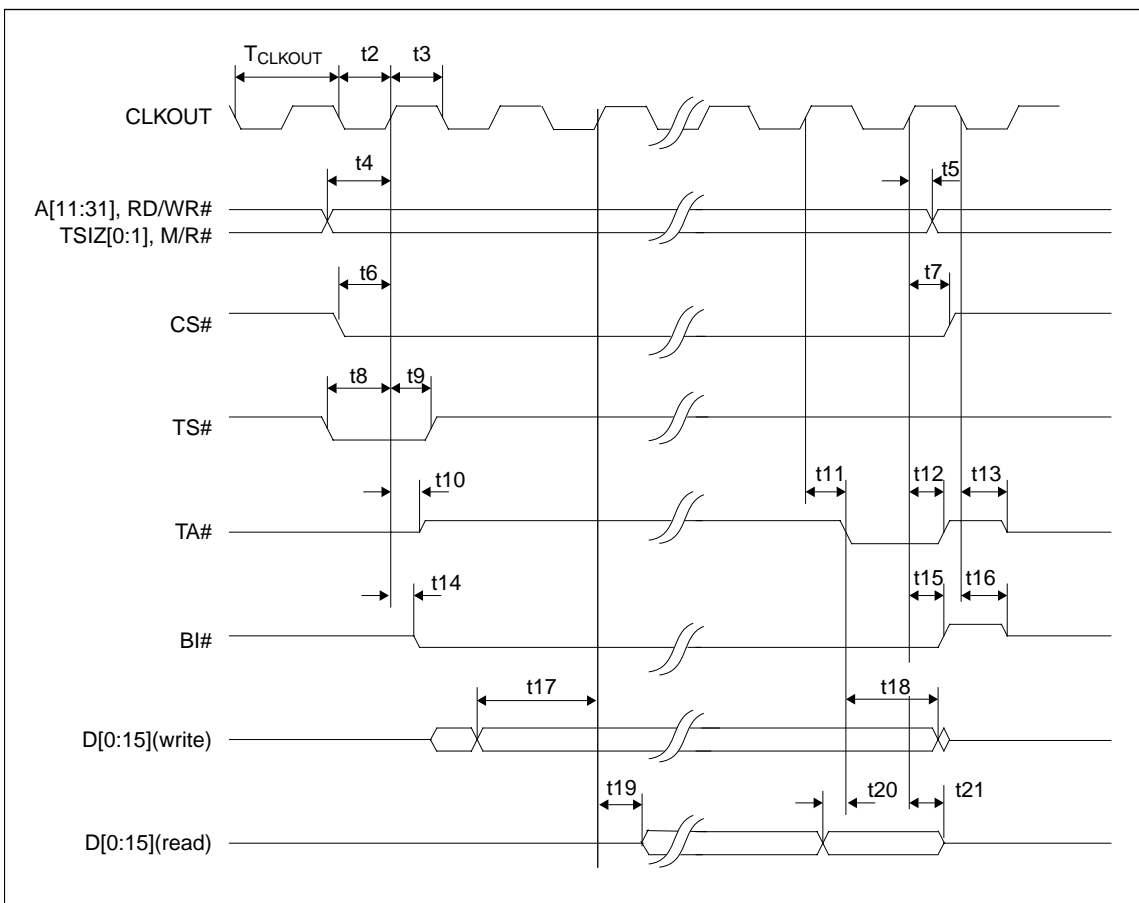
Figure 7-7 Motorola PowerPC Timing

Note: The above timing diagram is not applicable if MD12 = 1 (BUSCLK divided by 2).

Table 7-7 Motorola PowerPC Timing

Symbol	Parameter	3.0V		5.0V		Units
		Min.	Max.	Min.	Max.	
f_{CLKOUT}	Clock frequency		45		50	MHz
T_{CLKOUT}	Clock period	$1/f_{CLKOUT}$		$1/f_{CLKOUT}$		ns
t2	Clock pulse width low	6		6		ns
t3	Clock pulse width high	6		6		ns
t4	AB[11:31], RD/WR#, TSIZ[0:1], M/R# setup	0		0		ns
t5	AB[11:31], RD/WR#, TSIZ[0:1], M/R# hold	0		0		ns
t6	CS# setup	1		0		ns
t7	CS# hold	0		1		ns
t8	TS# setup	1		1		ns
t9	TS# hold	0		1		ns
t10	CLKOUT to TA# driven	2		1		ns
t11	CLKOUT to TA# low	4	19	3	12	ns
t12	CLKOUT to TA# high	4	20	3	12	ns
t13	negative edge CLKOUT to TA# tri-state	3	15	2	10	ns
t14	CLKOUT to BI# driven	3	18	3	11	ns
t15	CLKOUT to BI# high	3	17	3	11	ns
t16	negative edge CLKOUT to BI# tri-state	3	13	2	9	ns
t17	DB[15:0] setup to 2nd CLKOUT after TS# = 0 (write cycle)	0		0		ns
t18	DB[15:0] hold (write cycle)	0		0		ns
t19	CLKOUT to DB driven (read cycle)	0		4		ns
t20	DB[15:0] valid to TA# falling edge (read cycle)	0		0		ns
t21	CLKOUT to DB[15:0] tri-state (read cycle)	7	15	5	18	ns

7.1.8 PC Card Timing (e.g. StrongARM)
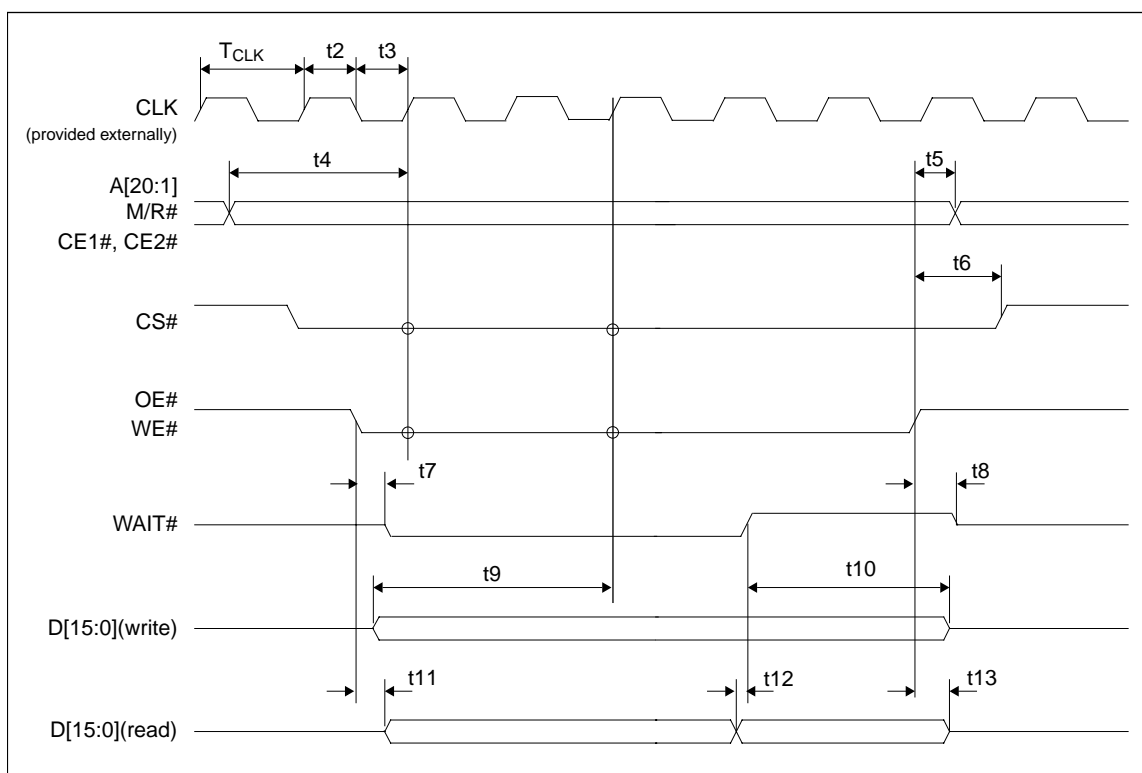
Figure 7-8 PC Card Timing

Note: The above timing diagram is not applicable if MD12 = 1 (BUSCLK divided by 2).

Table 7-8 PC Card Timing

Symbol	Parameter	3.0V		5.0V		Units
		Min.	Max.	Min.	Max.	
f_{CLK}	Clock frequency		50		50	MHz
T_{CLK}	Clock period	$1/f_{CLK}$		$1/f_{CLK}$		ns
t2	Clock pulse width high	6		6		ns
t3	Clock pulse width low	6		6		ns
t4	A[20:1], M/R# setup to first CLK where CE1# = 0 or CE2# = 0 and either OE# = 0 or WE# = 0	4		3		ns
t5	A[20:1], M/R# hold from rising edge of either OE #or WE#	0		0		ns
t6	CS# hold from rising edge of either OE# or WE#	0		0		ns
t7	Falling edge of either OE# or WE# to WAIT# driven low	2	21	2	9	ns
t8	Rising edge of either OE# or WE# to WAIT# tri-state	3	14	2	9	ns
t9	D[15:0] setup to third CLK where CE1# = 0, CE2# = 0 and WE# = 0 (write cycle)	0		0		ns
t10	D[15:0] hold (write cycle)	0		0		ns
t11	Falling edge OE# to D[15:0] driven (read cycle)	10		8		ns
t12	D[15:0] setup to rising edge WAIT# (read cycle)	0		0		ns
t13	Rising edge of OE# to D[15:0] tri-state (read cycle)	7	34	5	17	ns

7.1.9 Philips Interface Timing (e.g. PR31500/PR31700)
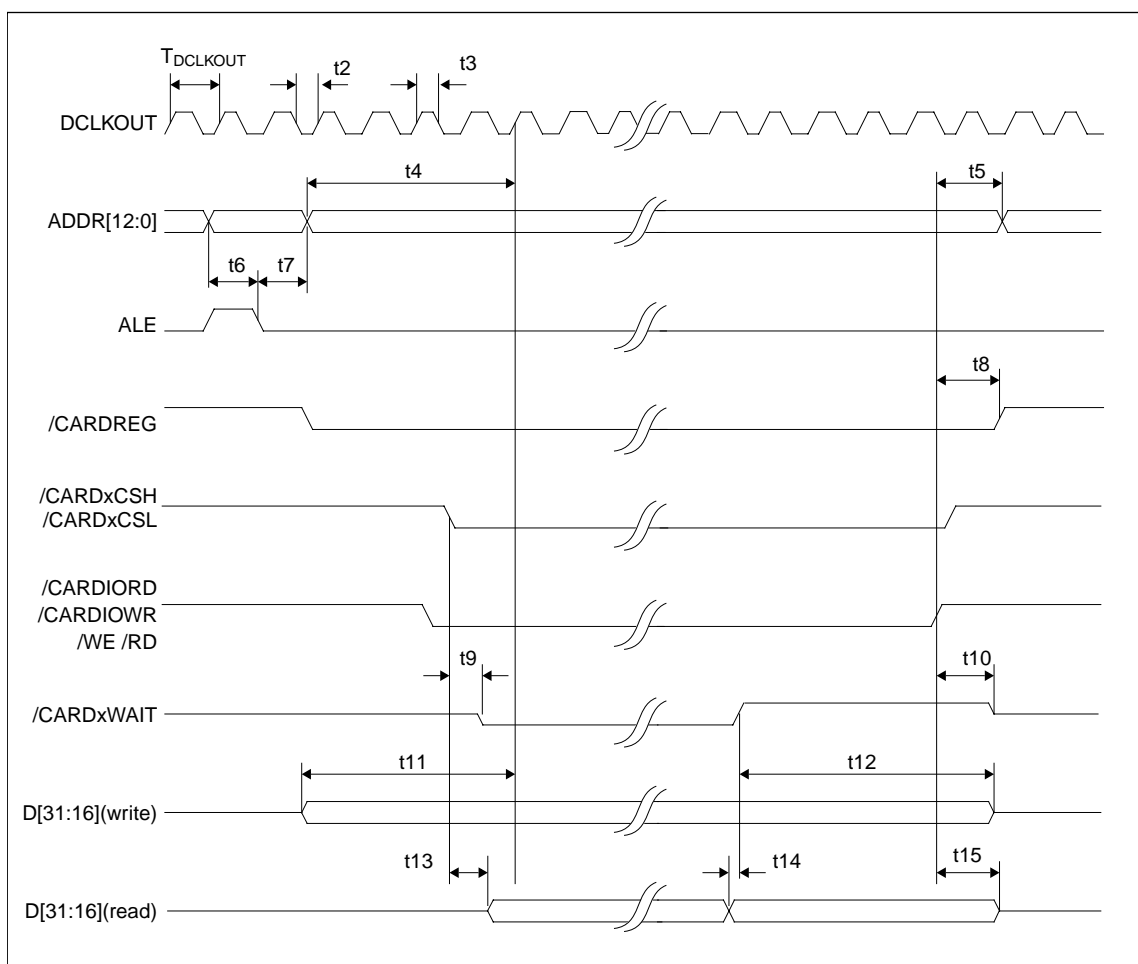
Figure 7-9 Philips Timing

Table 7-9 Philips Timing

Symbol	Parameter	3.0V		5.0V		Units
		Min.	Max.	Min.	Max.	
$f_{DCLKOUT}$	Clock frequency		75		75	MHz
$T_{DCLKOUT}$	Clock period	$1/f_{DCLKOUT}$		$1/f_{DCLKOUT}$		ns
t2	Clock pulse width low	6		6		ns
t3	Clock pulse width high	6		6		ns
t4	ADDR[12:0] setup to first CLK of cycle	10		10		ns
t5	ADDR[12:0] hold from command invalid	0		0		ns
t6	ADDR[12:0] setup to falling edge ALE	10		10		ns
t7	ADDR[12:0] hold from falling edge ALE	5		5		ns
t8	/CARDREG hold from command invalid	0		0		ns
t9	Falling edge of chip select to /CARDxWAIT driven	2	14	1	9	ns
t10	Command invalid to /CARDxWAIT tri-state	2	13	2	12	ns
t11	D[31:16] valid to first CLK of cycle (write cycle)	10		10		ns
t12	D[31:16] hold from rising edge of /CARDxWAIT	0		0		
t13	Chip select to D[31:16] driven (read cycle)	4		3		ns
t14	D[31:16] setup to rising edge /CARDxWAIT (read cycle)	0		0		ns
t15	Command invalid to D[31:16] tri-state (read cycle)	7	30	4	16	ns

Note: If BUSCLK exceeds 37.5MHz, it must be divided by 2 using MD12 (see Table 5-6, "Summary of Power-On/Reset Options" on page 1-23).

EPSON

7.1.10 Toshiba Interface Timing (e.g. TX39xx)
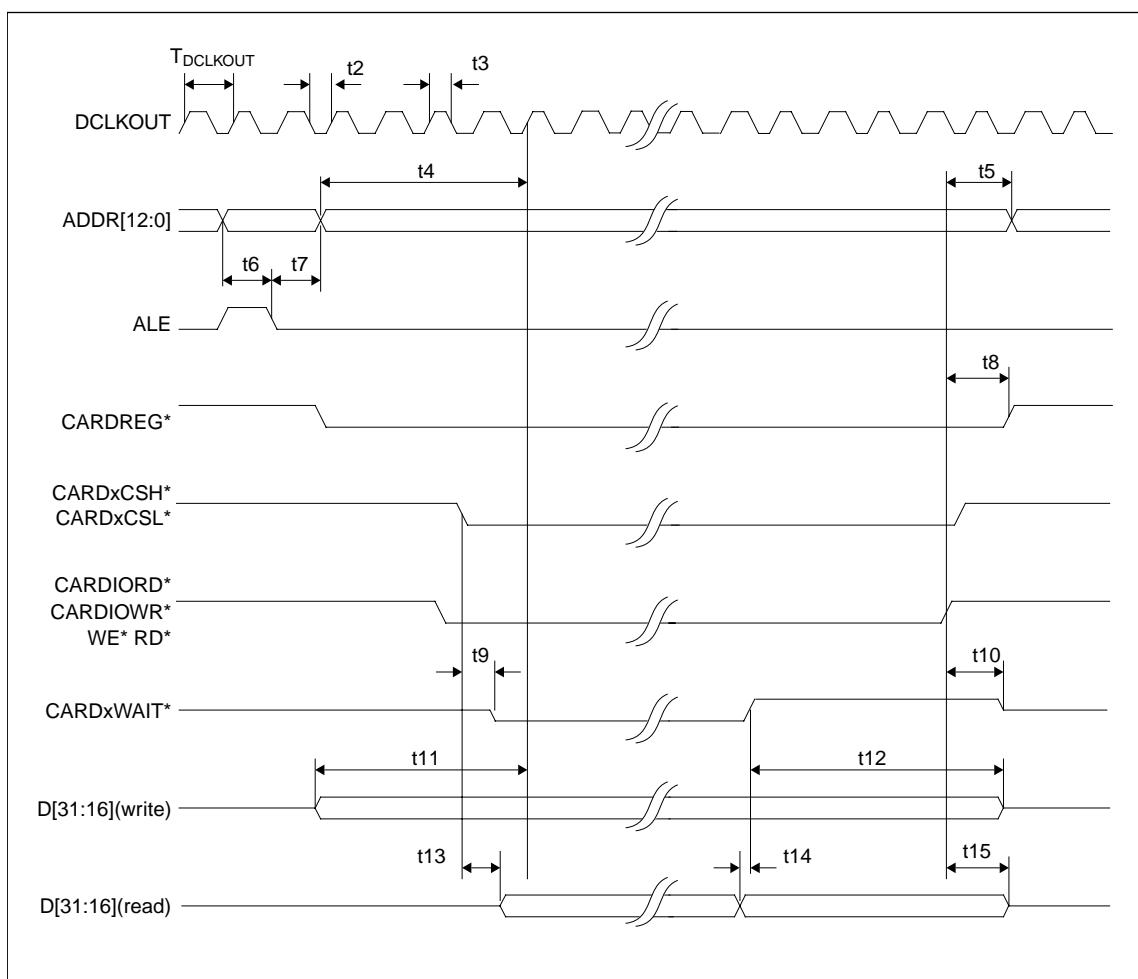
Figure 7-10 Toshiba Timing

Table 7-10 Toshiba Timing

Symbol	Parameter	3.0V		5.0V		Units
		Min.	Max.	Min.	Max.	
f$_{DCLKOUT}$	Clock frequency		75		75	MHz
T$_{DCLKOUT}$	Clock period	1/ f$_{DCLKOUT}$		1/ f$_{DCLKOUT}$		ns
t2	Clock pulse width low	6		6		ns
t3	Clock pulse width high	6		6		ns
t4	ADDR[12:0] setup to first CLK of cycle	10		10		ns
t5	ADDR[12:0] hold from command invalid	0		0		ns
t6	ADDR[12:0] setup to falling edge ALE	10		10		ns
t7	ADDR[12:0] hold from falling edge ALE	5		5		ns
t8	CARDREG* hold from command invalid	0		0		ns
t9	Falling edge of chip select to CARDxWAIT* driven	2	14	1	9	ns
t10	Command invalid to CARDxWAIT* tri-state	2	13	2	12	ns
t11	D[31:16] valid to first CLK of cycle (write cycle)	10		10		ns
t12	D[31:16] hold from rising edge of CARDxWAIT*	0		0		
t13	Chip select to D[31:16] driven (read cycle)	4		3		ns
t14	D[31:16] setup to rising edge CARDxWAIT* (read cycle)	0		0		ns
t15	Command invalid to D[31:16] tri-state (read cycle)	7	30	4	16	ns

Note: If BUSCLK exceeds 37.5MHz, it must be divided by 2 using MD12 (see Table 5-6, "Summary of Power-On/Reset Options" on page 1-23).

7.2 Clock Timing

7.2.1 Input Clocks
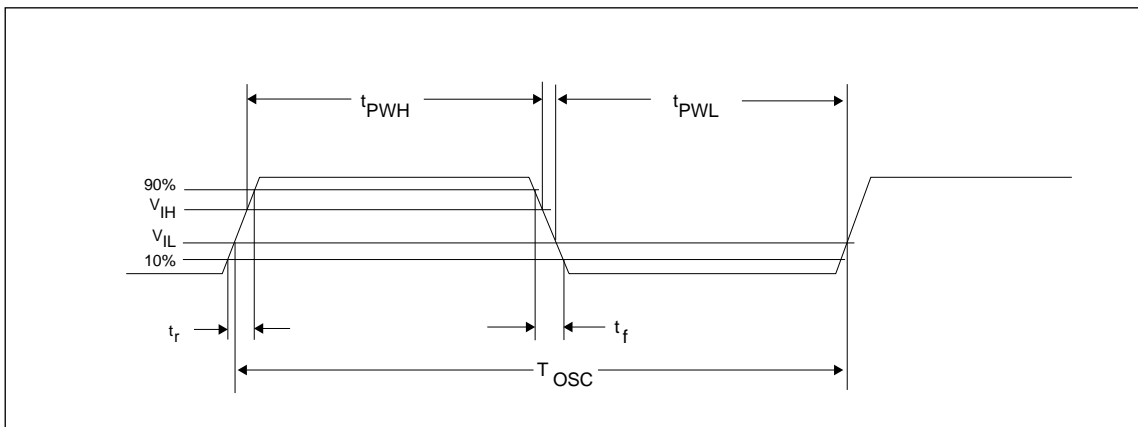
Figure 7-11 CLKI Clock Input Requirements

Table 7-11 Clock Input Requirements for CLKI/CLKI2/BUSCLK divided down internally

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 7-12 Clock Input Requirements for CLKI or BUSCLK if used directly for MCLK[1]

Symbol	Parameter	Min.	Max.	Units
f_{OSC}	Input Clock Frequency		40	MHz
T_{OSC}	Input Clock Period	$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High	11.3[1]		ns
t_{PWL}	Input Clock Pulse Width Low	11.3[1]		ns
t_f	Input Clock Fall Time (10% - 90%)		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5	ns

[1] MCLK must have a duty cycle of 50% ± 5%.

7.2.2 Internal Clocks

Table 7-13 Internal Clock Requirements

Symbol	Parameter	Min.	Max.	Units
f_{MCLK}	Memory Clock Frequency	0	40	MHz
$f_{LCD\ PCLK}$	LCD Pixel Clock Frequency	0	40	MHz
$f_{CRT/TV\ PCLK}$	CRT/TV Pixel Clock Frequency	0	Note 1	MHz
$f_{MediaPlug\ Clock}$	MediaPlug Clock Frequency	0	10	MHz

Note: 1. The maximum CRT pixel clock is 40MHz.
The TV pixel clock for NTSC output is fixed at 14.318MHz.
The TV pixel clock for PAL output is fixed at 17.734MHz.

7.3 Memory Interface Timing

7.3.1 EDO-DRAM Read, Write, Read-Write Timing
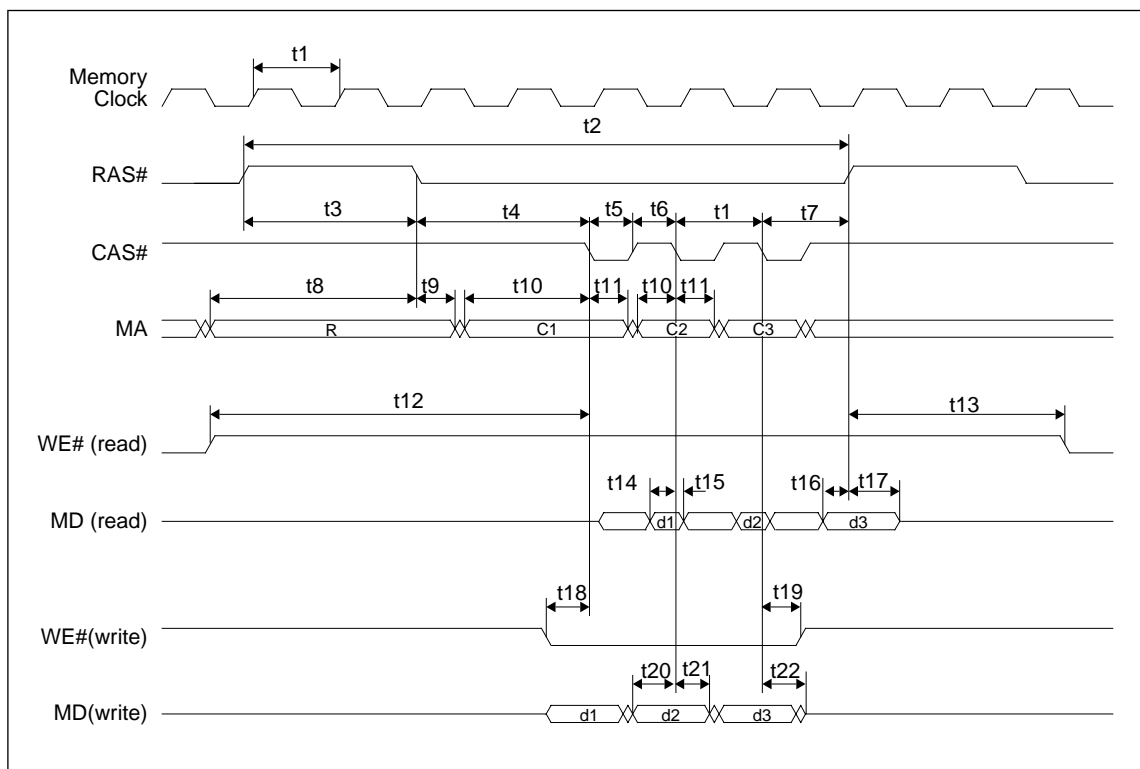
Figure 7-12 EDO-DRAM Page Mode Timing
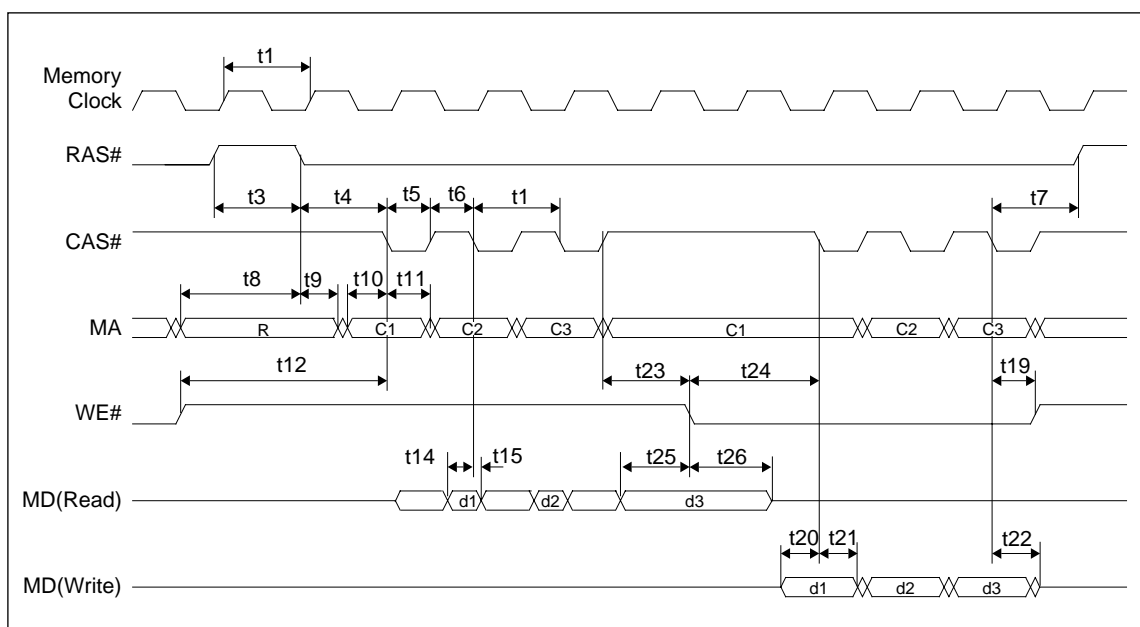
Figure 7-13 EDO-DRAM Read-Write Timing

EPSON

Table 7-14 EDO-DRAM Read, Write, Read-Write Timing

Symbol	Parameter	Min.	Max.	Units
t1	Memory clock period	25		ns
t2	Random read or write cycle time (REG[02Bh] bits 1-0 = 00)	5 t1		ns
	Random read or write cycle time (REG[02Bh] bits 1-0 = 01)	4 t1		ns
	Random read or write cycle time (REG[02Bh] bits 1-0 = 10)	3 t1		ns
t3	RAS# precharge time (REG[02Ah] bits 1-0 = 00)	2 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 01)	1.45 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 10)	t1		ns
t4	RAS# to CAS# delay time (REG[02Ah] bit 4 = 0 and bits 1-0 = 00 or 10)	2 t1 - 3	2 t1	ns
	RAS# to CAS# delay time (REG[02Ah] bit 4 = 1 and bits 1-0 = 00 or 10)	t1 - 3	t1	ns
	RAS# to CAS# delay time (REG[02Ah] bits 1-0 = 01)	1.45 t1 - 3	1.55 t1	ns
t5	CAS# precharge time	0.45 t1		ns
t6	CAS# pulse width	0.45 t1 - 1		ns
t7	RAS# hold time	t1		ns
t8	Row address setup time (REG[02Ah] bits 1-0 = 00)	2.45 t1 - 3		ns
	Row address setup time (REG[02Ah] bits 1-0 = 01)	2 t1 - 3		
	Row address setup time (REG[02Ah] bits 1-0 = 10)	1.45 t1 - 3		
t9	Row address hold time (REG[02Ah] bits 1-0 = 00 or 10)	0.45 t1 - 1		ns
	Row address hold time (REG[02Ah] bits 1-0 = 01)	t1 - 3		ns
t10	Column address setup time	0.45 t1 - 3		ns
t11	Column address hold time	0.45 t1 - 1		ns
t12	Read Command setup (REG[02Ah] bit 4 = 0 and bits 1-0 = 00)	4.45 t1 - 1		ns
	Read Command setup (REG[02Ah] bit 4 = 1 and bits 1-0 = 10)	2.45 t1 - 1		
	Read Command setup (all other REG[02Ah] values)	3.45 t1 - 1		
t13	Read Command hold (REG[02Ah] bit 4 = 0 and bits 1-0 = 00)	3.45 t1 - 1		
	Read Command hold (REG[02Ah] bit 4 = 1 and bits 1-0 = 10)	1.45 t1 - 1		
	Read Command hold (all other REG[02Ah] values)	2.45 t1 - 1		ns
t14	Read data setup referenced from CAS#	4		
t15	Read data hold referenced from CAS#	2		
t16	Last read data setup referenced from RAS#	3		
t17	Bus turn-off from RAS#	2		
t18	Write command setup time	0.45 t1 - 1		ns
t19	Write command hold time	0.45 t1 - 1		ns
t20	Write Data setup time	0.45 t1 - 4		ns
t21	Write Data hold time	0.45 t1		ns
t22	MD tri-state	0.45 t1	0.55 t1 + 19	ns
t23	CAS# to WE# active during read-write cycle	t1		ns
t24	Write command setup during read-write cycle	1.45 t1 - 1		ns
t25	Last read data setup referenced from WE# during read-write cycle	18		ns
t26	Bus tri-state from WE# during read-write cycle	0	t1 - 6	ns

7.3.2 EDO-DRAM CAS Before RAS Refresh Timing
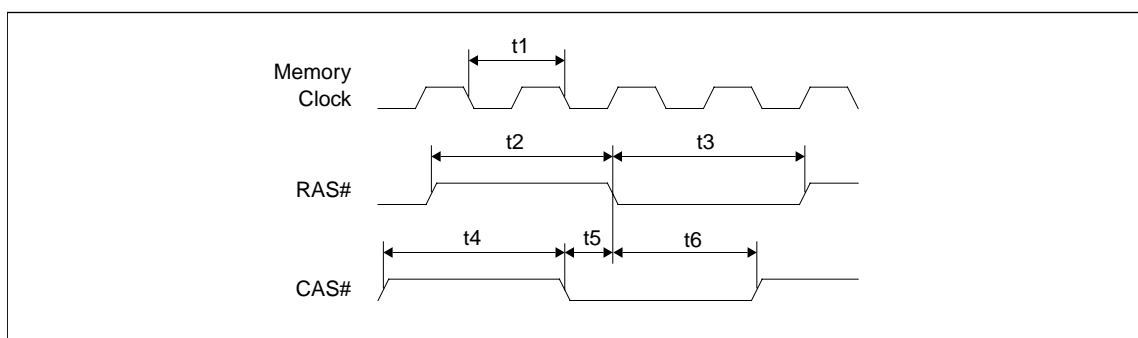
Figure 7-14 EDO-DRAM CAS Before RAS Refresh Timing

Table 7-15 EDO-DRAM CAS Before RAS Refresh Timing

Symbol	Parameter	Min.	Max.	Units
t1	Memory clock period	25		ns
t2	RAS# precharge time (REG[02Ah] bits 1-0 = 00)	2 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 01)	1.45 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 10)	t1		ns
t3	RAS# pulse width (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 00)	3 t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 01)	3.45 t1 - 1		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 10)	4 t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 00)	2 t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 01)	2.45 t1 - 1		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 10)	3 t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 00)	t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 01)	1.45 t1 - 1		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 10)	2 t1 - 7		ns
t4	CAS# precharge time (REG[02Ah] bits 1-0 = 00)	2 t1		ns
	CAS# precharge time (REG[02Ah] bits 1-0 = 01 or 10)	t1		ns
t5	CAS# setup time (REG[02Ah] bits 1-0 = 00 or 10)	0.45 t1		ns
	CAS# setup time (REG[02Ah] bits 1-0 = 01)	t1 - 4		ns
t6	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 00)	2.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 01)	3 t1		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 10)	3.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 00)	1.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 01)	2 t1		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 10)	2.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 00)	0.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 01)	t1		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 10)	1.45 t1 - 4		ns

7.3.3 EDO-DRAM Self-Refresh Timing
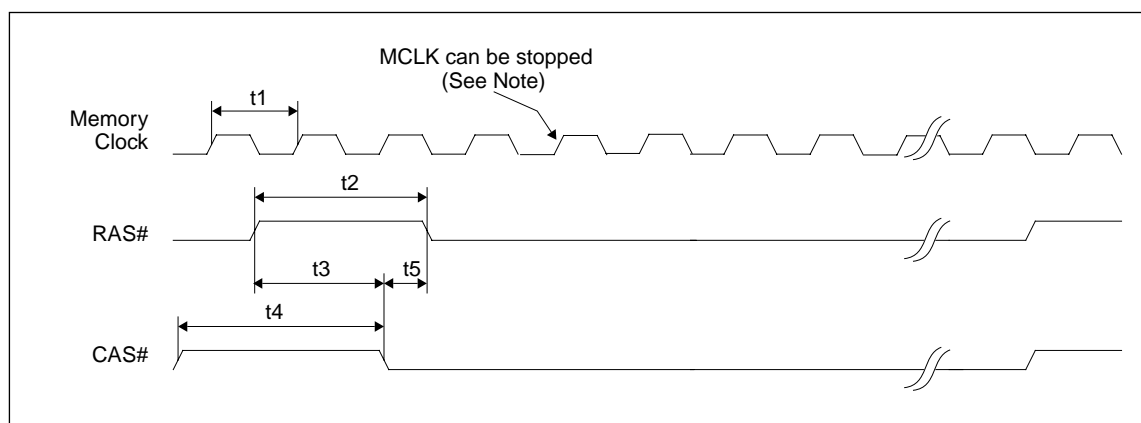
Figure 7-15 EDO-DRAM Self-Refresh Timing

Note: MCLK can be stopped. For timing see Section 7.4.2, "Power Save Mode" on page 1-62.

Table 7-16 EDO-DRAM Self-Refresh Timing

Symbol	Parameter	Min.	Max.	Units
t1	Memory clock period	25		ns
t2	RAS# precharge time (REG[02Ah] bits 1-0 = 00)	2 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 01)	1.45 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 10)	t1		ns
t3	RAS# to CAS# precharge time (REG[02Ah] bits 1-0 = 00)	1.45 t1		ns
	RAS# to CAS# precharge time (REG[02Ah] bits 1-0 = 01 or 10)	0.45 t1		ns
t4	CAS# precharge time (REG[02Ah] bits 1-0 = 00)	2 t1		ns
	CAS# precharge time (REG[02Ah] bits 1-0 = 01 or 10)	t1		ns
t5	CAS# setup time (REG[02Ah] bits 1-0 = 00 or 10)	0.45 t1		ns
	CAS# setup time (REG[02Ah] bits 1-0 = 01)	t1 - 4		ns

7.3.4 FPM-DRAM Read, Write, Read-Write Timing
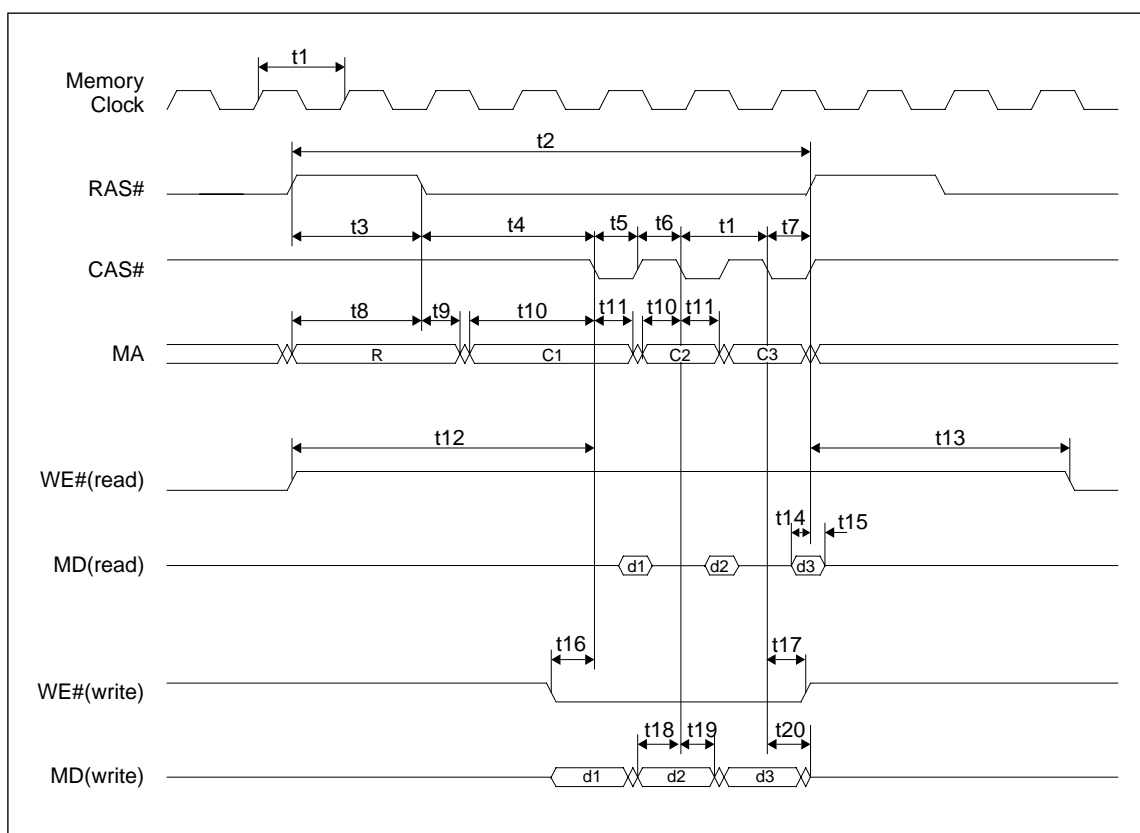
Figure 7-16 FPM-DRAM Page Mode Timing
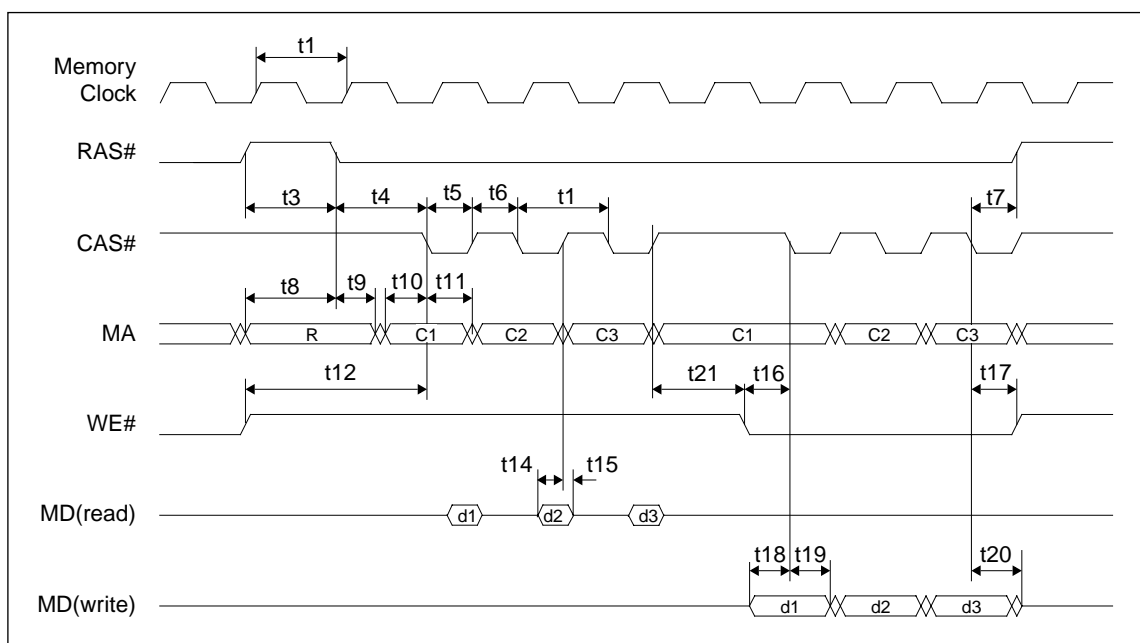
Figure 7-17 FPM-DRAM Read-Write Timing

Table 7-17 FPM-DRAM Read, Write, Read-Write Timing

Symbol	Parameter	Min.	Max.	Units
t1	Memory clock	40		ns
t2	Random read or write cycle time (REG[02Bh] bits 1-0 = 00)	5 t1		ns
	Random read or write cycle time (REG[02Bh] bits 1-0 = 01)	4 t1		ns
	Random read or write cycle time (REG[02Bh] bits 1-0 = 10)	3 t1		ns
t3	RAS# precharge time (REG[02Ah] bits 1-0 = 00)	2 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 01)	1.45 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 10)	t1		ns
t4	RAS# to CAS# delay time (REG[02Ah] bit 4 = 0 and bits 1-0 = 00 or 10)	2.45 t1 - 3	2.55 t1	ns
	RAS# to CAS# delay time (REG[02Ah] bit 4 = 1 and bits 1-0 = 00 or 10)	1.45 t1 - 3	1.55 t1	ns
	RAS# to CAS# delay time (REG[02Ah] bit 4 = 0 and bits 1-0 = 01)	2 t1 - 3	2 t1	ns
	RAS# to CAS# delay time (REG[02Ah] bit 4 = 1 and bits 1-0 = 01)	t1 - 3	t1	ns
t5	CAS# precharge time	0.45 t1		ns
t6	CAS# pulse width	0.45 t1 - 1		ns
t7	RAS# hold time	0.45 t1		ns
t8	Row address setup time (REG[02Ah] bits 1-0 = 00)	2 t1 - 2		ns
	Row address setup time (REG[02Ah] bits 1-0 = 01)	1.45 t1 - 2		ns
	Row address setup time (REG[02Ah] bits 1-0 = 10)	t1 - 2		ns
t9	Row address hold time (REG[02Ah] bits 1-0 = 00 or 10)	t1 - 3		ns
	Row address hold time (REG[02Ah] bits 1-0 = 01)	0.45 t1 - 3		ns
t10	Column address set-up time	0.45 t1 - 3		ns
t11	Column address hold time	0.45 t1 - 1		ns
t12	Read Command setup (REG[02Ah] bit 4 = 0 and bits 1-0 = 00)	4.45 t1 - 1		ns
	Read Command setup (REG[02Ah] bit 4 = 1 and bits 1-0 = 01 or 10)	2.45 t1 - 1		ns
	Read Command setup (all other REG[02Ah] values)	3.45 t1 - 1		ns
t13	Read Command hold (REG[02Ah] bit 4 = 0 and bits 1-0 = 00)	4 t1 - 1		ns
	Read Command hold (REG[02Ah] bit 4 = 1 and bits 1-0 = 01 or 10)	2 t1 - 1		ns
	Read Command hold (all other REG[02Ah] values)	3 t1 - 1		ns
t14	Read data setup referenced from CAS#	3		ns
t15	Read Data turn-off from CAS#	3		ns
t16	Write command setup time	0.45 t1 - 1		ns
t17	Write command hold time	0.45 t1 - 1		ns
t18	Write Data setup time	0.45 t1 - 4		ns
t19	Write Data hold time	0.45 t1		ns
t20	MD tri-state	0.45 t1	0.55 t1 + 19	ns
t21	CAS# to WE# active during read-write cycle	0.45 t1		ns

7.3.5 FPM-DRAM CAS Before RAS Refresh Timing
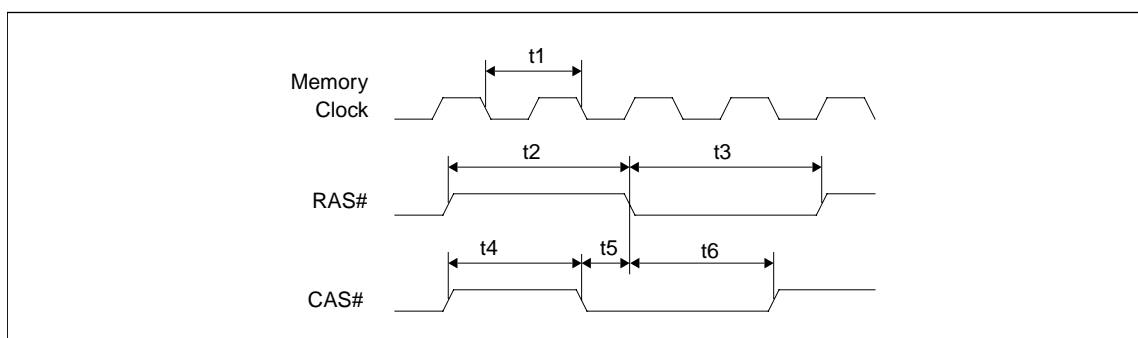
Figure 7-18 FPM-DRAM CAS Before RAS Refresh Timing

Table 7-18 FPM-DRAM CAS Before RAS Refresh Timing

Symbol	Parameter	Min.	Max.	Units
t1	Memory clock	40		ns
t2	RAS# precharge time (REG[02Ah] bits 1-0 = 00)	2.45 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 01 or 10)	1.45 t1		ns
t3	RAS# pulse width (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 00)	2.45 t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 01 or 10)	3.45 t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 00)	1.45 t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 01 or 10)	2.45 t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 00)	0.45 t1 - 7		ns
	RAS# pulse width (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 01 or 10)	1.45 t1 - 7		ns
t4	CAS# precharge time (REG[02Ah] bits 1-0 = 00)	2 t1		ns
	CAS# precharge time (REG[02Ah] bits 1-0 = 01 or 10)	t1		ns
t5	CAS# setup time	0.45 t1		ns
t6	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 00)	2.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 00, REG[02Ah] bits 1-0 = 01 or 10)	3.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 00)	1.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 01, REG[02Ah] bits 1-0 = 01 or 10)	2.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 00)	0.45 t1 - 4		ns
	CAS# hold to RAS# (REG[02Bh] bits 1-0 = 10, REG[02Ah] bits 1-0 = 01 or 10)	1.45 t1 - 4		ns

7.3.6 FPM-DRAM Self-Refresh Timing
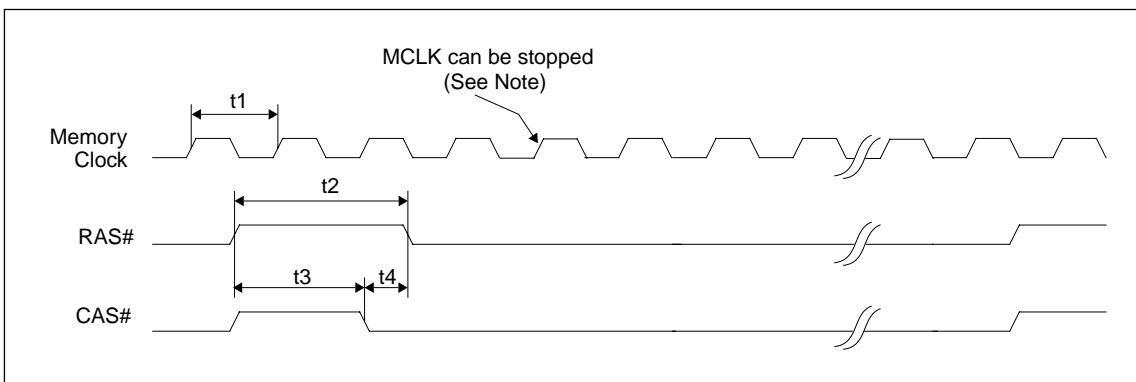
Figure 7-19 FPM-DRAM Self-Refresh Timing

Note: MCLK can be stopped. For timing see Section 7.4.2, "Power Save Mode" on page 1-62.

Table 7-19 FPM-DRAM Self-Refresh Timing

Symbol	Parameter	Min.	Max.	Units
t1	Memory clock	40		ns
t2	RAS# precharge time (REG[02Ah] bits 1-0 = 00)	2.45 t1		ns
	RAS# precharge time (REG[02Ah] bits 1-0 = 01 or 10)	1.45 t1		ns
t3	RAS# to CAS# precharge time (REG[02Ah] bits 1-0 = 00)	2 t1		ns
	RAS# to CAS# precharge time (REG[02Ah] bits 1-0 = 01 or 10)	t1		ns
t4	CAS# setup time	0.45 t1		ns

7.4 Power Sequencing

7.4.1 LCD Power Sequencing
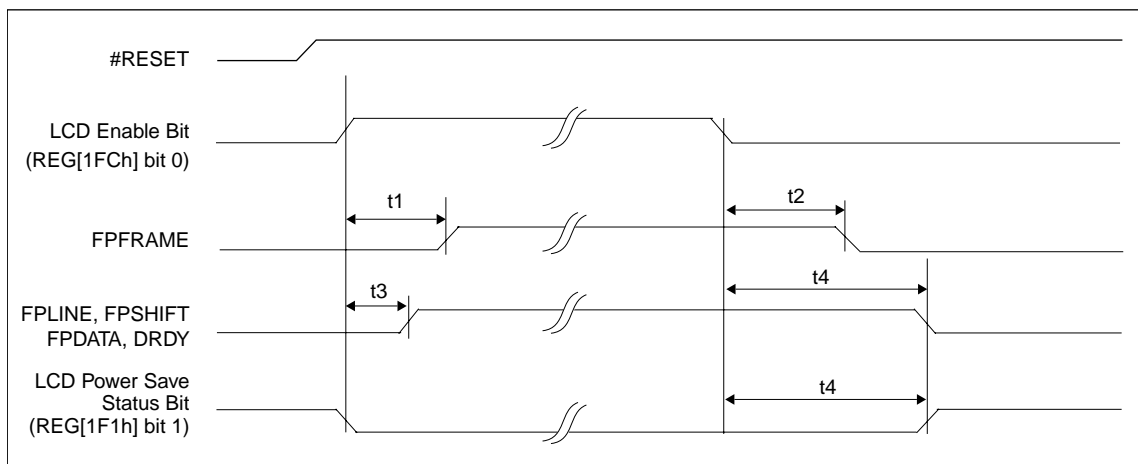
Figure 7-20 LCD Panel Power-off/Power-on Timing

Table 7-20 LCD Panel Power-off/Power-on Timing

Symbol	Parameter	Min.	Max.	Units
t1	LCD Enable Bit high to FPFRAME active		$T_{FPFRAME}$	ns
t2	FPFRAME inactive to LCD Power Save Status bit high		$5T_{FPFRAME}$	ns
t3	LCD Enable Bit high to FPLINE, FPSHIFT, FPDATA, DRDY active		$3T_{FPLINE}$	ns
t4	LCD Enable Bit low to FPLINE, FPSHIFT, FPDATA, DRDY active **and** LCD Power Save Status bit high		note 1	ns

Note: 1.   t4 $= 130T_{FPFRAME}$ for dual panels
$= 65T_{FPFRAME}$ for single panels

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

7.4.2 *Power Save Mode*
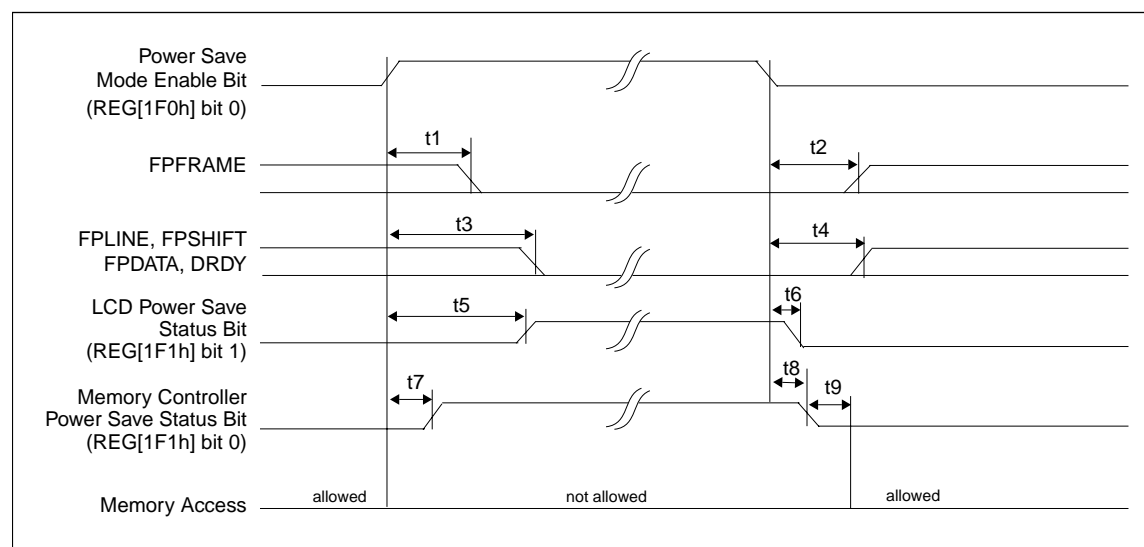
Figure 7-21 Power Save Mode Timing

Note: Memory accesses cannot be performed after a Power Save Mode has been initiated.

The Memory Controller Power Save Status Bit will go high only if the Refresh Select Bits (REG[021h] bits 7-6) are set to Self-Refresh or No Refresh.

Table 7-21 Power Save Mode Timing

Symbol	Parameter	Min.	Max.	Units
t1	Power Save Mode Enable Bit high to FPFRAME inactive		$T_{FPFRAME} + T_{FPLINE}$	ns
t2	Power Save Mode Enable Bit low to FPFRAME active		$3T_{FPLINE}$	ns
t3	Power Save Mode Enable Bit high to FPLINE, FPSHIFT, FPDATA, DRDY inactive		$129T_{FPFRAME} + T_{FPLINE}$	ns
t4	Power Save Mode Enable Bit low to FPLINE, FPSHIFT, FPDATA, DRDY active		$T_{FPFRAME}$	ns
t5	Power Save Mode Enable Bit high to LCD Power Save Status Bit high	$128T_{FPFRAME}$	$129T_{FPFRAME}$	ns
t6	Power Save Mode Enable Bit low to LCD Power Save Status Bit low		T_{PCLK}	ns
t7	Power Save Mode Enable Bit high to Memory Controller Power Save Status Bit high (self-refresh or no refresh selected)		note 1	ns
t8	Power Save Mode Enable Bit low to Memory Controller Power Save Status Bit low (self-refresh or no refresh selected)		$12T_{MCLK}$	ns
t9	Memory Controller Power Save Status Bit low to the earliest time where memory access is allowed (self-refresh or no refresh selected)		$8T_{MCLK}$	ns

Note: 1. $t14_{max}$= (1 DRAM refresh clock period) + 12 MCLK periods

Note: Where $T_{FPFRAME}$ is the period of FPFRAME, T_{FPLINE} is the period of FPLINE, T_{PCLK} is the period of the pixel clock, and T_{MCLK} is the period of the memory clock.

The DRAM refresh clock period is programmed using REG[021h].

7.5 Display Interface

7.5.1 Single Monochrome 4-Bit Panel Timing
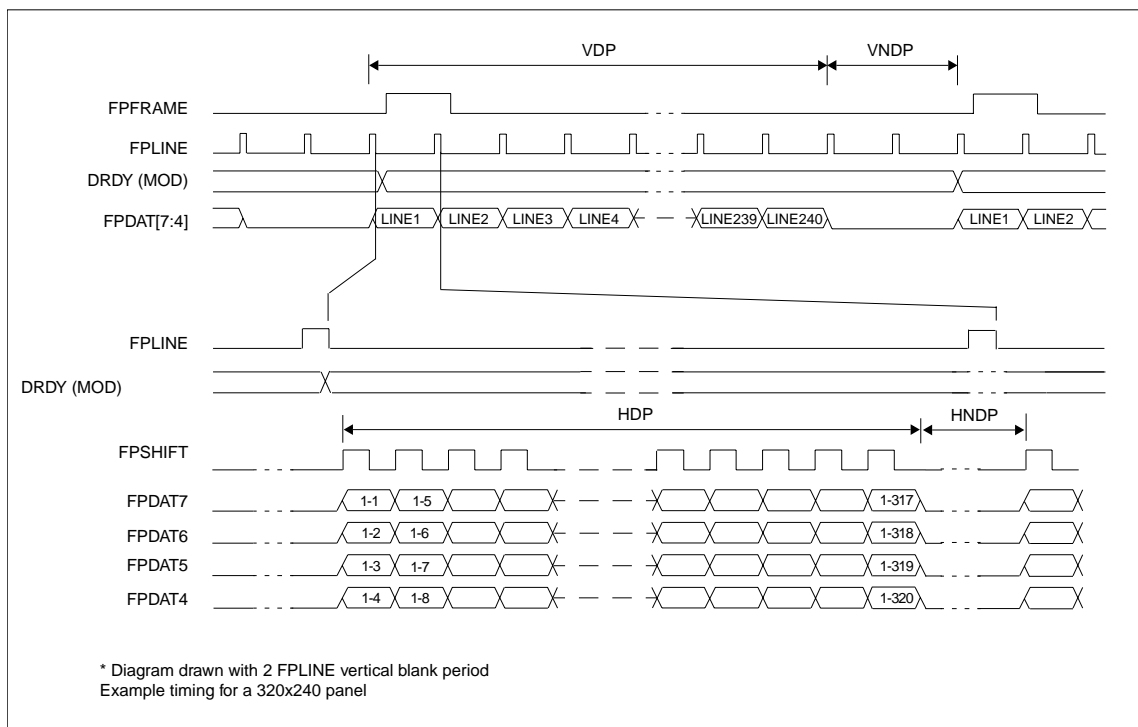
Figure 7-22 Single Monochrome 4-Bit Panel Timing

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) $\times$ 8 Ts
HNDP = Horizontal Non-Display Period = ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts

Figure 7-23 Single Monochrome 4-Bit Panel A.C. Timing

Table 7-22 Single Monochrome 4-Bit Panel A.C. Timing

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3	note 4	259	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	5	note 5	229	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	7	note 5	231	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		20		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		18		Ts
t8	FPSHIFT period		4		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	16	note 6	240	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	18	note 6	242	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		18		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		16		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

Note: 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. $t1 = t4 - 12$ Ts

3. $t4 = [((REG[032h]\ bits\ [6:0]) + 1) \times 8 + ((REG[034h]\ bits\ [4:0]) + 1) \times 8]$ Ts

4. $t5 = [((REG[034h]\ bits\ [6:0]) + 1) \times 8 + 3]$ Ts for (REG[031h] bits [5:0]) $\neq$ 0
   $= 3$ Ts   for (REG[031h] bits [5:0]) = 0

5. $t6 = [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 27]$ Ts for 4 bpp or 8 bpp color depth
   $= [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 25]$ Ts for 15/16 bpp color depth

6. $t9 = [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 16]$ Ts for 4 bpp or 8 bpp color depth
   $= [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 14]$ Ts for 15/16 bpp color depth

7.5.2 *Single Monochrome 8-Bit Panel Timing*

Figure 7-24 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) $\times$ 8 Ts
HNDP = Horizontal Non-Display Period = ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts
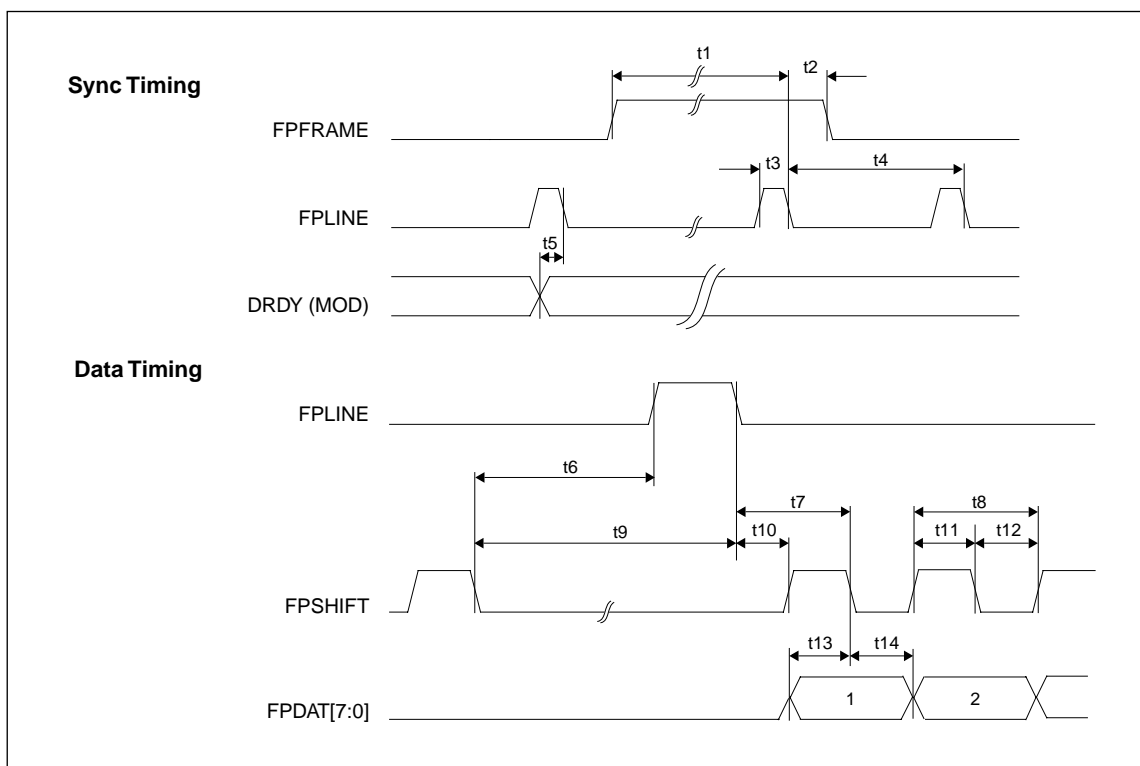
Figure 7-25 Single Monochrome 8-Bit Panel A.C. Timing

Table 7-23 Single Monochrome 8-Bit Panel A.C. Timing

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3	note 4	259	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	7	note 5	231	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	9	note 5	233	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		22		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		20		Ts
t8	FPSHIFT period		8		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	18	note 6	242	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	20	note 6	244	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		18		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		16		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

Note: 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 = t4 - 12 Ts

3. t4 = [((REG[032h] bits [6:0]) + 1) $\times$ 8 + ((REG[034h] bits [4:0]) + 1) $\times$ 8] Ts

4. t5 = [((REG[034h] bits [6:0]) + 1) $\times$ 8 + 3] Ts for (REG[031h] bits [5:0]) $\neq$ 0
   = 3 Ts   for (REG[031h] bits [5:0]) = 0

5. t6 = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 25] Ts for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 23] Ts for 15/16 bpp color depth

6. t9 = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 14] Ts for 4 bpp or 8 bpp color depth
   = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 12] Ts for 15/16 bpp color depth

7.5.3 Single Color 4-Bit Panel Timing

Figure 7-26 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) $\times$ 8 Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts

Figure 7-27 Single Color 4-Bit Panel A.C. Timing

Table 7-24 Single Color 4-Bit Panel A.C. Timing

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3	note 4	259	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	4.5	note 5	228.5	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	6.5	note 5	230.5	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		19.5		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		17.5		Ts
t8	FPSHIFT period		1		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	15.5	note 6	239.5	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	17.5	note 6	241.5	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		19		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		17		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

Note: 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. $t1 = t4 - 12$ Ts

3. $t4 = [((REG[032h] \text{ bits } [6:0]) + 1) \times 8 + ((REG[034h] \text{ bits } [4:0]) + 1) \times 8]$ Ts

4. $t5 = [((REG[034h] \text{ bits } [6:0]) + 1) \times 8 + 3]$ Ts for $(REG[031h] \text{ bits } [5:0]) \neq 0$
    $= 3$ Ts   for $(REG[031h] \text{ bits } [5:0]) = 0$

5. $t6 = [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 27.5]$ Ts for 4 bpp or 8 bpp color depth
    $= [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 25.5]$ Ts for 15/16 bpp color depth

6. $t9 = [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 16.5]$ Ts for 4 bpp or 8 bpp color depth
    $= [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 14.5]$ Ts for 15/16 bpp color depth

7.5.4 Single Color 8-Bit Panel Timing (Format 1)
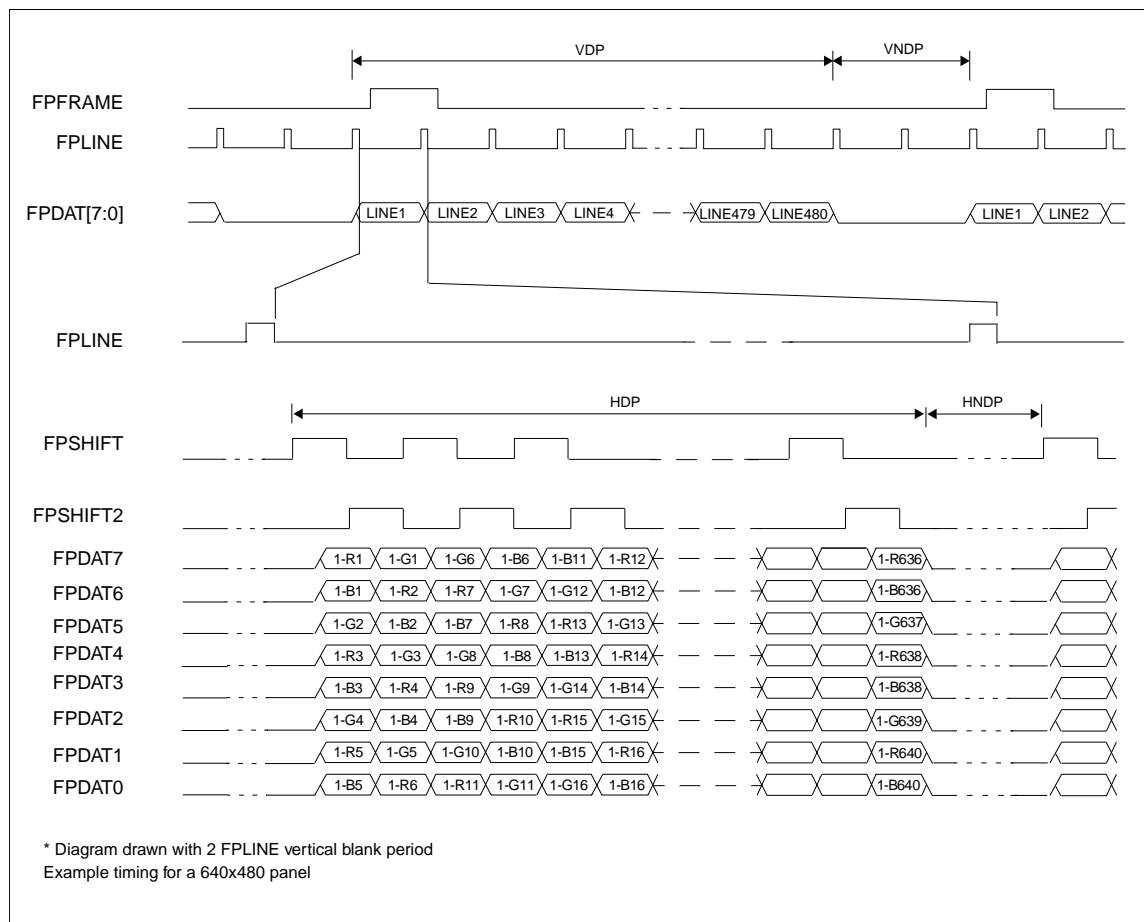
Figure 7-28 Single Color 8-Bit Panel Timing (Format 1)

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) $\times$ 8 Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts
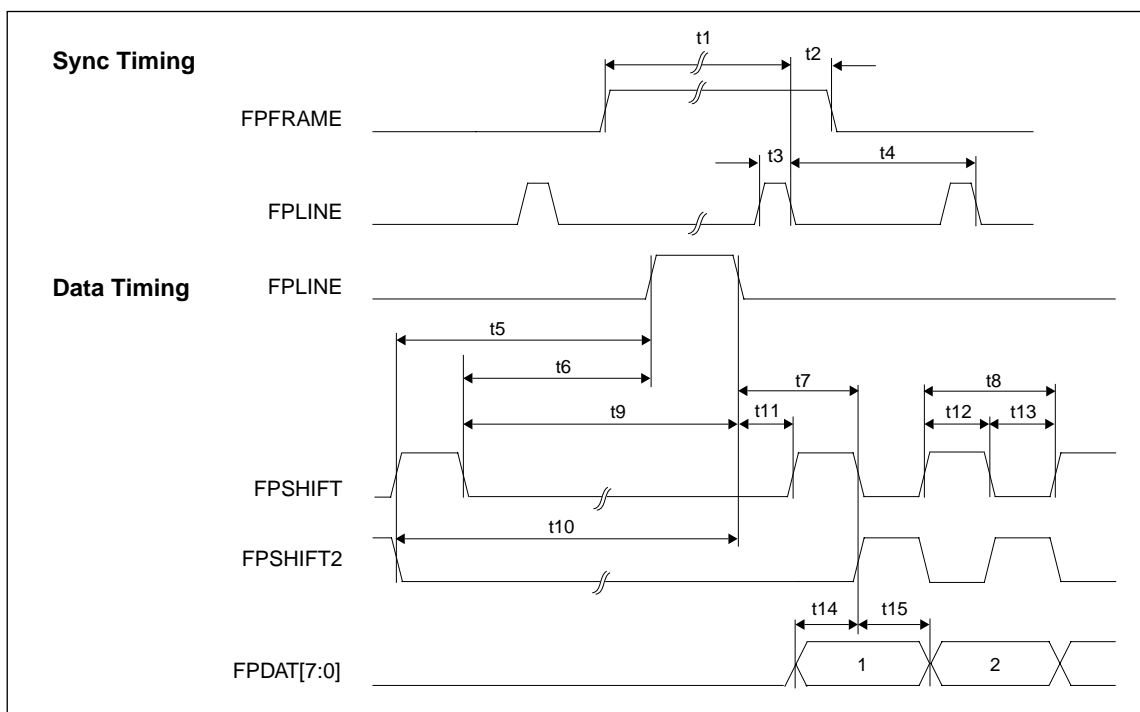
Figure 7-29 Single Color 8-Bit Panel A.C. Timing (Format 1)

Table 7-25 Single Color 8-Bit Panel A.C. Timing (Format 1)

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts
t2	FPFRAME hold from FPLINE falling edge		12		Ts (note 1)
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5a	FPSHIFT2 falling edge to FPLINE rising edge, 4 bpp or 8 bpp	5	note 4	229	Ts
t5b	FPSHIFT2 falling edge to FPLINE rising edge, 15/16 bpp	7	note 4	231	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	3	note 5	227	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	5	note 5	229	Ts
t7a	FPLINE falling edge to FPSHIFT2 rising, FPSHIFT falling edge, 4/8 bpp		20		Ts
t7b	FPLINE falling edge to FPSHIFT2 rising, FPSHIFT falling edge, 15/16 bpp		18		Ts
t8	FPSHIFT2, FPSHIFT period		4		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	14	note 6	238	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	16	note 6	240	Ts
t10a	FPSHIFT2 falling edge to FPLINE falling edge, 4 bpp or 8 bpp	16	note 7	240	Ts
t10b	FPSHIFT2 falling edge to FPLINE falling edge, 15/16 bpp	18	note 7	242	Ts
t11a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		18		Ts
t11b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		16		Ts
t12	FPSHIFT2, FPSHIFT pulse width high		2		Ts
t13	FPSHIFT2, FPSHIFT pulse width low		2		Ts
t14	FPDAT[7:0] setup to FPSHIFT2 rising, FPSHIFT falling edge		1		Ts
t15	FPDAT[7:0] hold from FPSHIFT2 rising, FPSHIFT falling edge		1		Ts

Notes 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 = t4 - 12 Ts

3: t4 = [((REG[032h] bits [6:0]) + 1) $\times$ 8 + ((REG[034h] bits [4:0]) + 1) $\times$ 8] Ts

4: t5 = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 27] Ts for 4 bpp or 8 bpp color depth
= [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 25] Ts for 15/16 bpp color depth

5: t6 = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 29] Ts for 4 bpp or 8 bpp color depth
= [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 27] Ts for 15/16 bpp color depth

6: t9 = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 18] Ts for 4 bpp or 8 bpp color depth
= [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 16] Ts for 15/16 bpp color depth

7: t10 = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 16] Ts for 4 bpp or 8 bpp color depth
= [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 14] Ts for 15/16 bpp color depth

7.5.5 Single Color 8-Bit Panel Timing (Format 2)
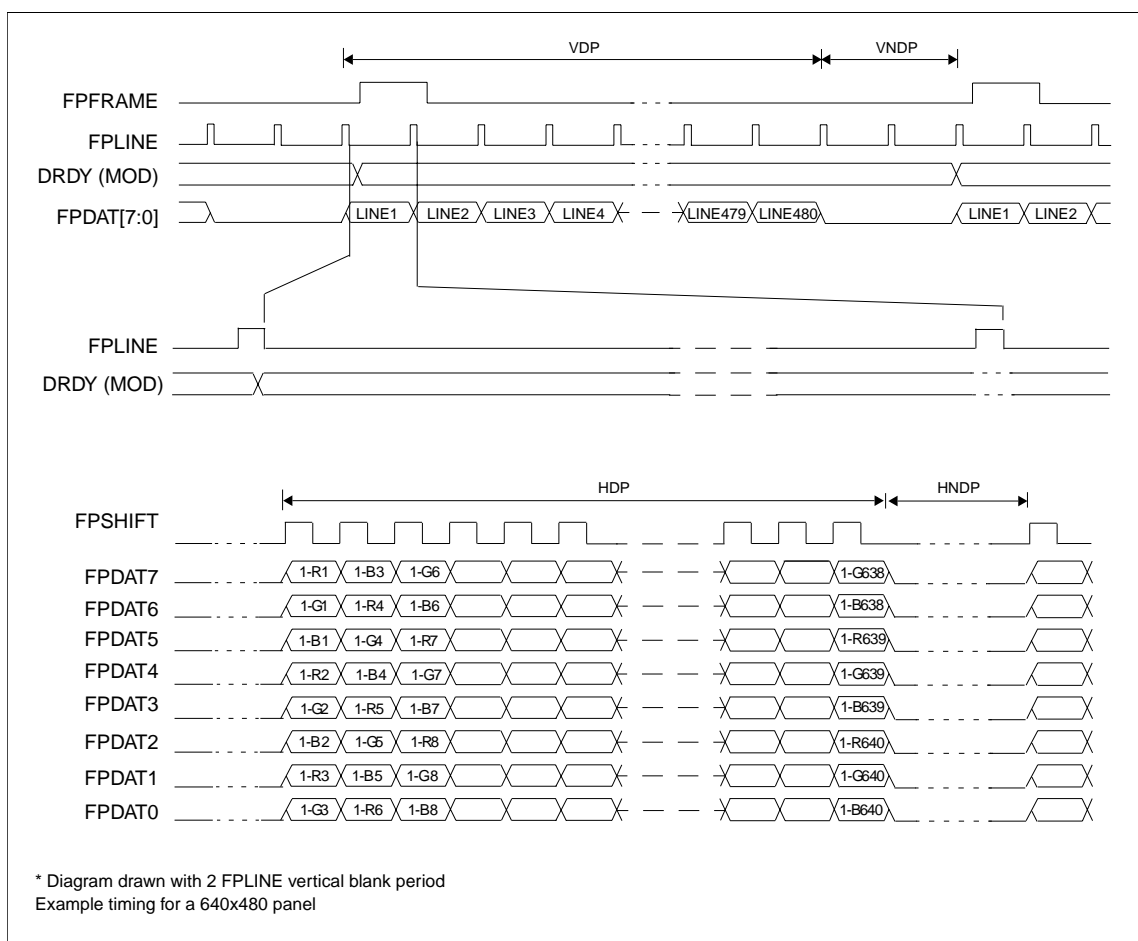
Figure 7-30 Single Color 8-Bit Panel Timing (Format 2)

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) $\times$ 8 Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts
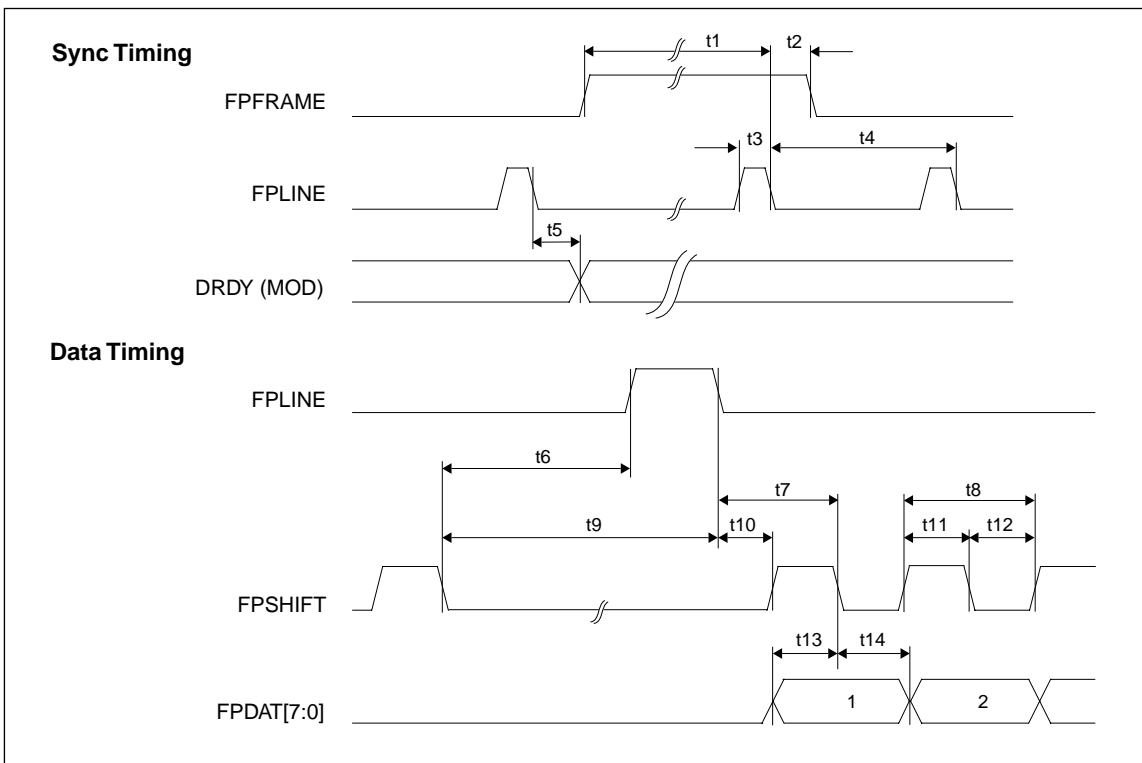
Figure 7-31 Single Color 8-Bit Panel A.C. Timing (Format 2)

EPSON

Table 7-26 Single Color 8-Bit Panel A.C. Timing (Format 2)

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3	note 4	259	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	4	note 5	228	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	6	note 5	230	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		20		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		18		Ts
t8	FPSHIFT period		2		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	15	note 6	239	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	17	note 6	241	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		18		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		16		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

Note: 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 = t4 - 12 Ts

3. t4 = $[((REG[032h] \text{ bits } [6:0]) + 1) \times 8 + ((REG[034h] \text{ bits } [4:0]) + 1) \times 8]$ Ts

4. t5 = $[((REG[034h] \text{ bits } [6:0]) + 1) \times 8 + 3]$ Ts for (REG[031h] bits [5:0]) $\neq$ 0
     = 3 Ts   for (REG[031h] bits [5:0]) = 0

5. t6 = $[((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 28]$ Ts for 4 bpp or 8 bpp color depth
     = $[((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 26]$ Ts for 15/16 bpp color depth

6. t9 = $[((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 17]$ Ts for 4 bpp or 8 bpp color depth
     = $[((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 15]$ Ts for 15/16 bpp color depth

7.5.6 Single Color 16-Bit Panel Timing
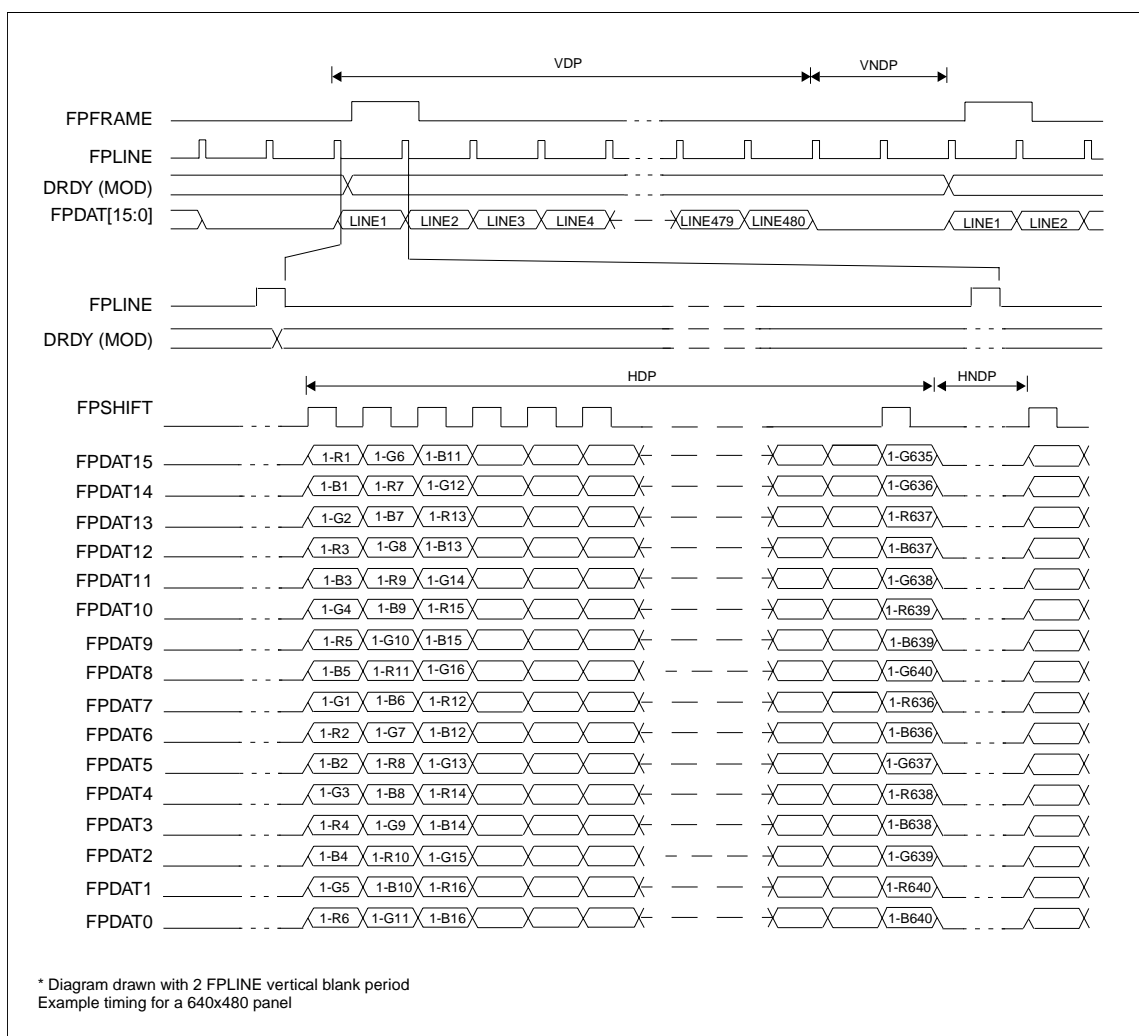
Figure 7-32 Single Color 16-Bit Panel Timing

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) $\times$ 8 Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts
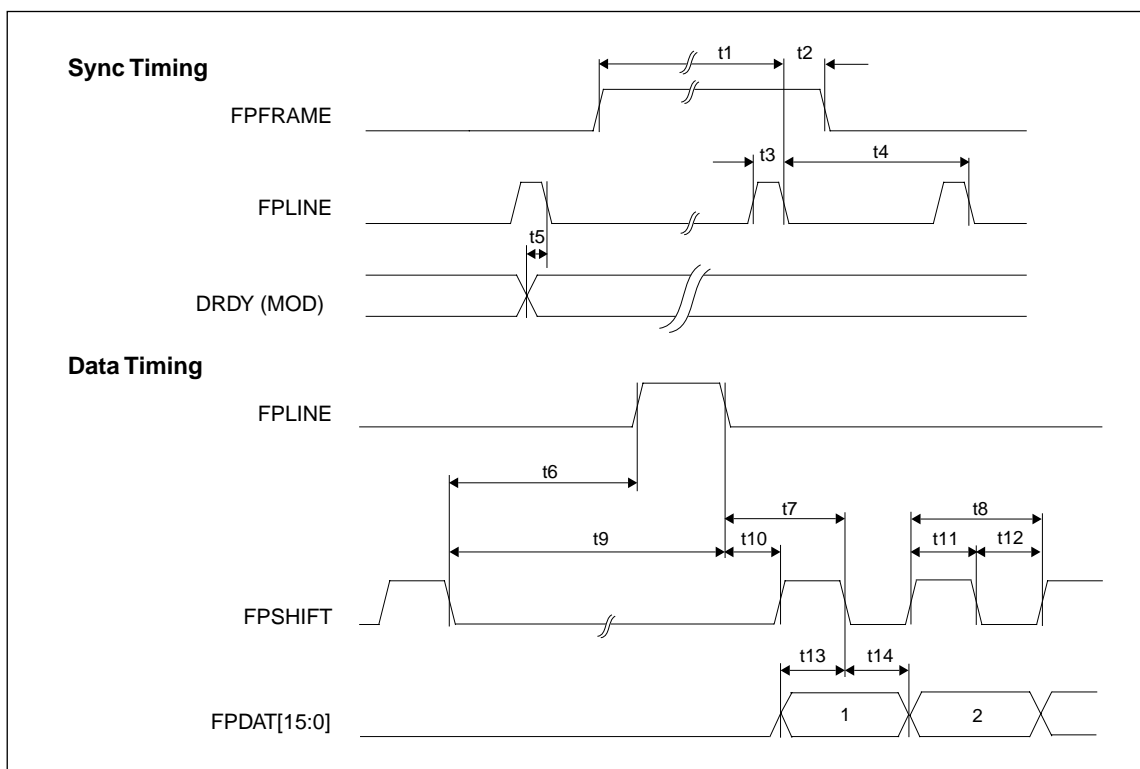
Figure 7-33 Single Color 16-Bit Panel A.C. Timing

Table 7-27 Single Color 16-Bit Panel A.C. Timing

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3	note 4	259	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	5	note 5	229	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	7	note 5	231	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		21		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		19		Ts
t8	FPSHIFT period		5		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	16	note 6	240	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	18	note 6	242	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		18		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		16		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

Note: 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. $t1 = t4 - 12$ Ts

3. $t4 = [((REG[032h]\ bits\ [6:0]) + 1) \times 8 + ((REG[034h]\ bits\ [4:0]) + 1) \times 8]$ Ts

4. $t5 = [((REG[034h]\ bits\ [6:0]) + 1) \times 8 + 3]$ Ts for (REG[031h] bits [5:0]) $\neq$ 0
   $= 3$ Ts   for (REG[031h] bits [5:0]) = 0

5. $t6 = [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 27]$ Ts for 4 bpp or 8 bpp color depth
   $= [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 25]$ Ts for 15/16 bpp color depth

6. $t9 = [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 16]$ Ts for 4 bpp or 8 bpp color depth
   $= [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 14]$ Ts for 15/16 bpp color depth

7.5.7 Single Color 16-Bit Panel Timing with External Circuit
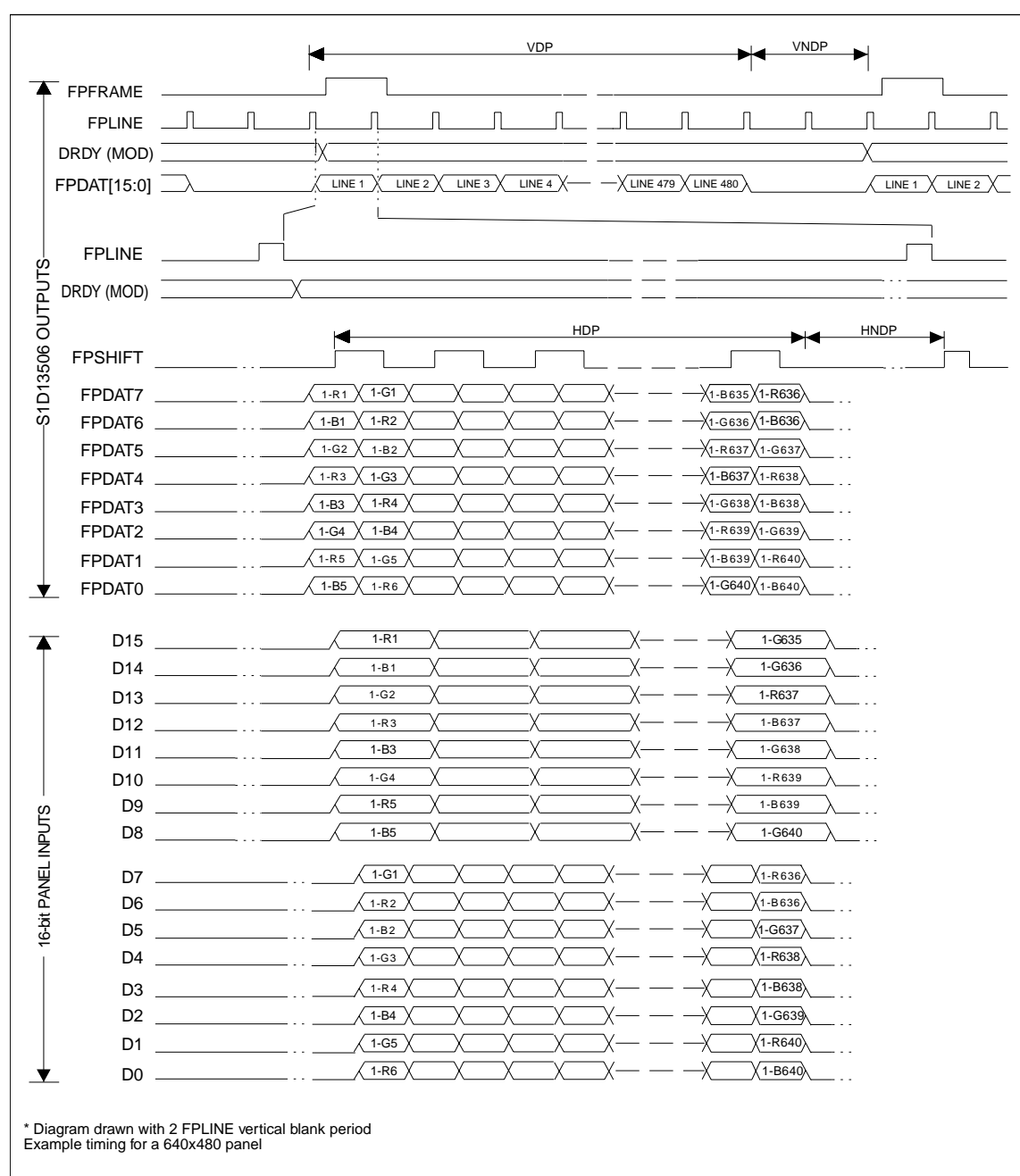
Figure 7-34 16-Bit Single Color Panel Timing with External Circuit

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) $\times$ 8 Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts
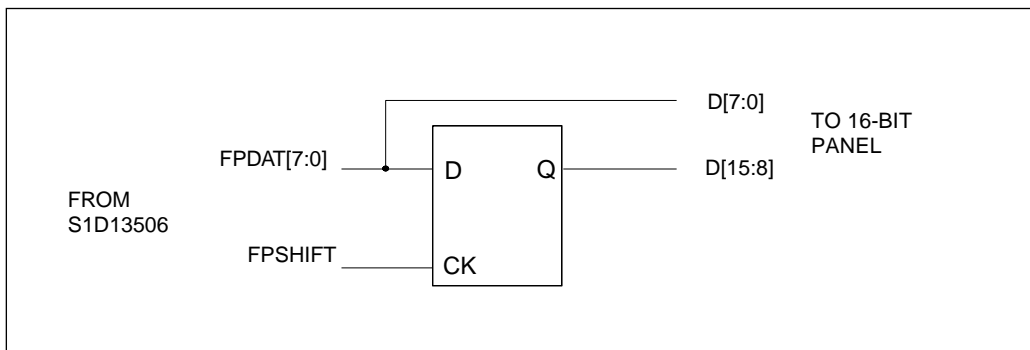
Figure 7-35 External Circuit for Color Single 16-Bit Panel When the Media Plug is Enabled
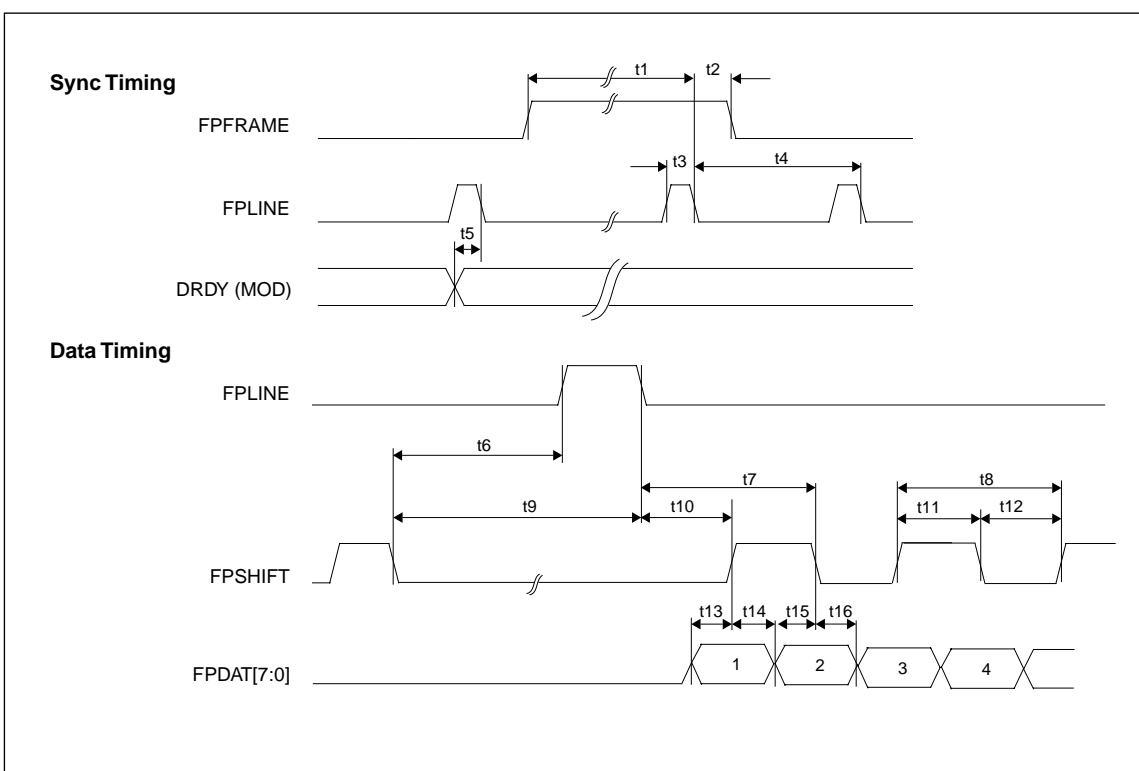
Figure 7-36 Single Color 16-Bit Panel (with External Circuit) A.C. Timing

Table 7-28 Single Color 16-Bit Panel (with External Circuit) A.C. Timing

Symbol	Parameter	Min. Setting	Nominal	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28 Ts	note 2	1268 Ts	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40 Ts	note 3	1280 Ts	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3 Ts	note 4	259 Ts	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	5 Ts	note 5	229 Ts	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	7 Ts	note 5	231 Ts	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		22		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		20		Ts
t8	FPSHIFT period		4		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	16 Ts	note 6	240 Ts	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	18 Ts	note 6	242 Ts	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		20		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		18		Ts
t11	FPSHIFT pulse width high		2		Ts
t12	FPSHIFT pulse width low		2		Ts
t13	FPDAT[7:0] setup to FPSHIFT rising edge		1		Ts
t14	FPDAT[7:0] hold to FPSHIFT rising edge		1		Ts
t15	FPDAT[7:0] setup to FPSHIFT falling edge		1		Ts
t16	FPDAT[7:0] hold to FPSHIFT falling edge		1		Ts

Note: 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. $t1 = t4 - 12\ Ts$

3. $t4 = [((REG[032h]\ bits\ [6:0]) + 1) \times 8 + ((REG[034h]\ bits\ [4:0]) + 1) \times 8]\ Ts$

4. $t5 = [((REG[034h]\ bits\ [6:0]) + 1) \times 8 + 3]\ Ts$ for (REG[031h] bits [5:0]) $\neq$ 0
   $= 3\ Ts$ for (REG[031h] bits [5:0]) = 0

5. $t6 = [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 27]\ Ts$ for 4 bpp or 8 bpp color depth
   $= [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 25]\ Ts$ for 15/16 bpp color depth

6. $t9 = [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 16]\ Ts$ for 4 bpp or 8 bpp color depth
   $= [((REG[034h]\ bits\ [4:0]) + 1) \times 8 - 14]\ Ts$ for 15/16 bpp color depth

7.5.8 Dual Monochrome 8-Bit Panel Timing

Figure 7-37 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) $\times$ 8 Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts
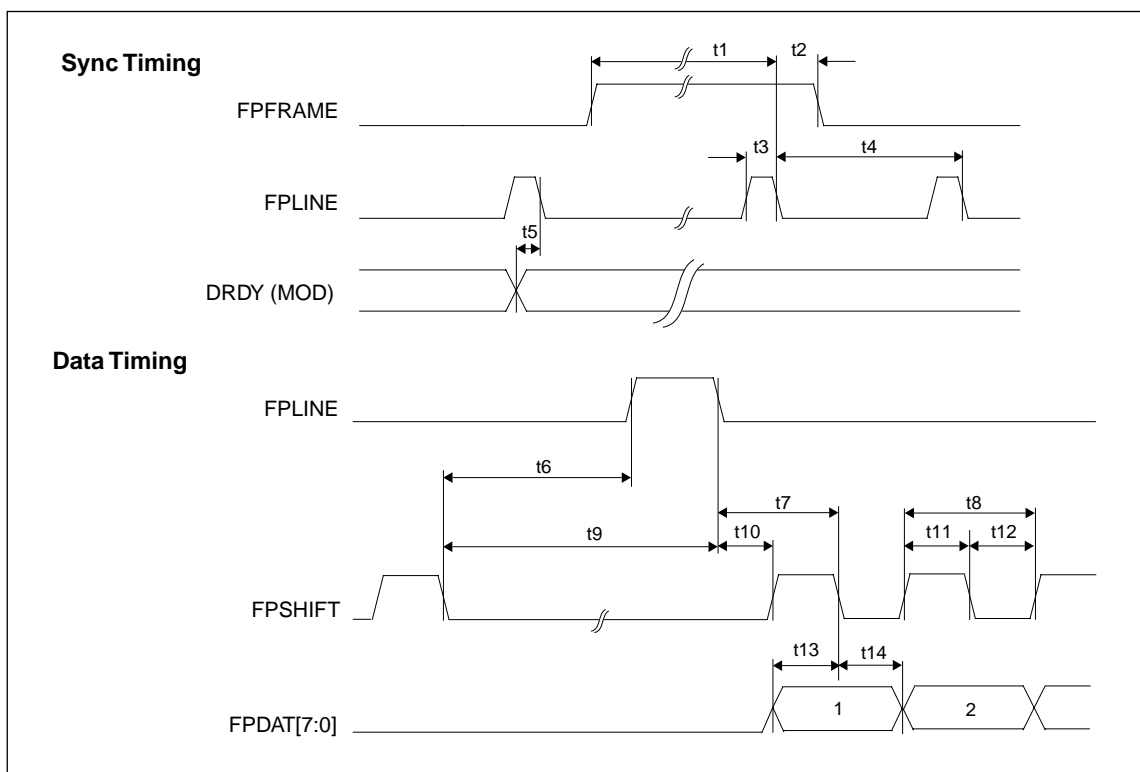
Figure 7-38 Dual Monochrome 8-Bit Panel A.C. Timing

EPSON

Table 7-29 Dual Monochrome 8-Bit Panel A.C. Timing

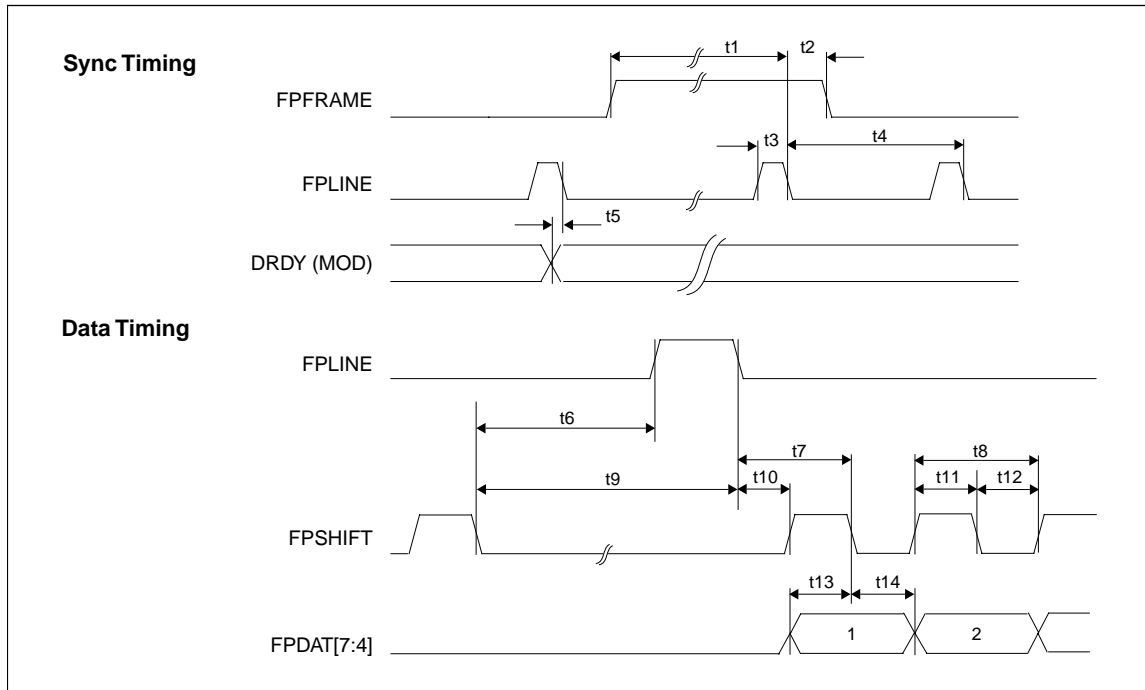| Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3	note 4	259	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	13	note 5	237	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	15	note 5	239	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		12		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		10		Ts
t8	FPSHIFT period		4		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	24	note 6	248	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	26	note 6	250	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		10		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		8		Ts
t11	FPSHIFT pulse width low		2		Ts
t12	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

Note: 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 = t4 - 12 Ts

3. t4 = [((REG[032h] bits [6:0]) + 1) $\times$ 8 + ((REG[034h] bits [4:0]) + 1) $\times$ 8] Ts

4. t5 = [((REG[034h] bits [6:0]) + 1) $\times$ 8 + 3] Ts for (REG[031h] bits [5:0]) $\neq$ 0
    = 3 Ts   for (REG[031h] bits [5:0]) = 0

5. t6 = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 19] Ts for 4 bpp or 8 bpp color depth
    = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 17] Ts for 15/16 bpp color depth

6. t9 = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 8] Ts for 4 bpp or 8 bpp color depth
    = [((REG[034h] bits [4:0]) + 1) $\times$ 8 - 6] Ts for 15/16 bpp color depth

7.5.9 Dual Color 8-Bit Panel Timing

* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 640x480 panel
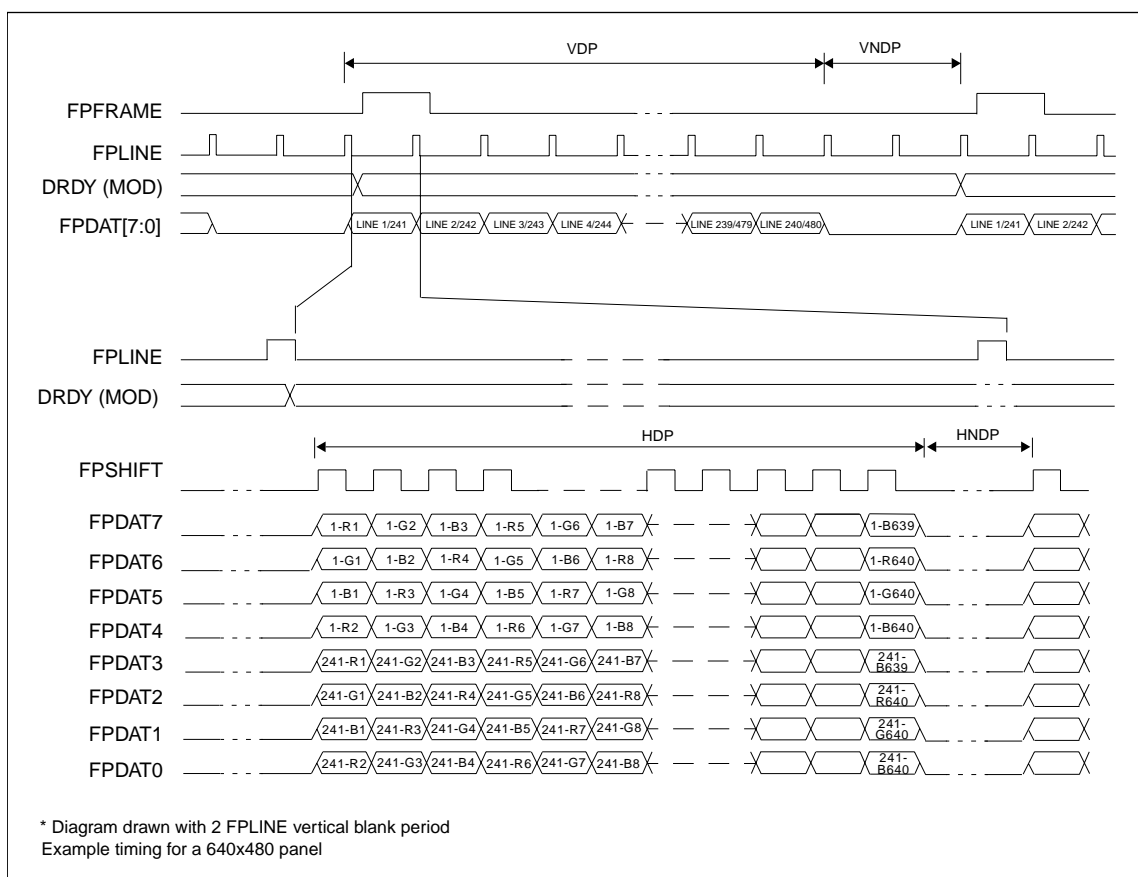
Figure 7-39 Dual Color 8-Bit Panel Timing

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) $\times$ 8 Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts
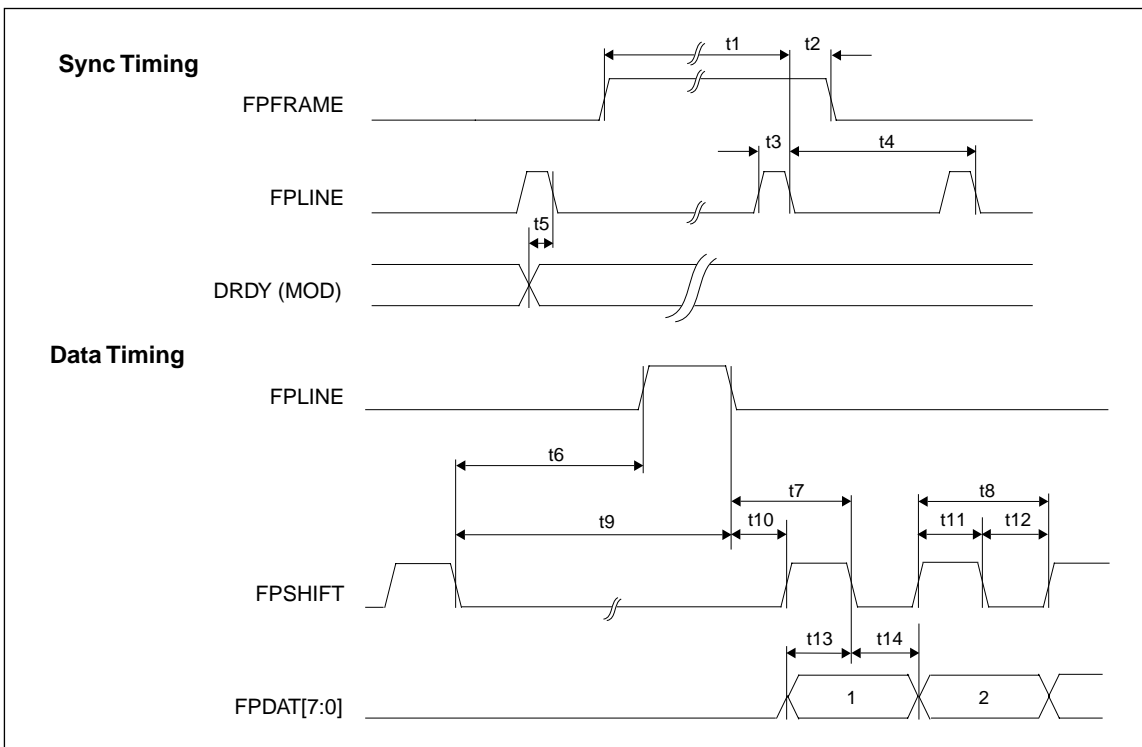
Figure 7-40 Dual Color 8-Bit Panel A.C. Timing

Table 7-30 Dual Color 8-Bit Panel A.C. Timing

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3	note 4	259	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	12	note 5	236	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	14	note 5	238	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		10.5		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		8.5		Ts
t8	FPSHIFT period		1		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	23	note 6	247	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	25	note 6	249	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		11		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		9		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

Note: 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. $t1 = t4 - 12$ Ts

3. $t4 = [((REG[032h] \text{ bits } [6:0]) + 1) \times 8 + ((REG[034h] \text{ bits } [4:0]) + 1) \times 8]$ Ts

4. $t5 = [((REG[034h] \text{ bits } [6:0]) + 1) \times 8 + 3]$ Ts for (REG[031h] bits [5:0]) $\neq$ 0
   $= 3$ Ts   for (REG[031h] bits [5:0]) = 0

5. $t6 = [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 20]$ Ts for 4 bpp or 8 bpp color depth
   $= [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 18]$ Ts for 15/16 bpp color depth

6. $t9 = [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 9]$ Ts for 4 bpp or 8 bpp color depth
   $= [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 7]$ Ts for 15/16 bpp color depth

7.5.10 Dual Color 16-Bit Panel Timing

Figure 7-41 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) × 8 Ts

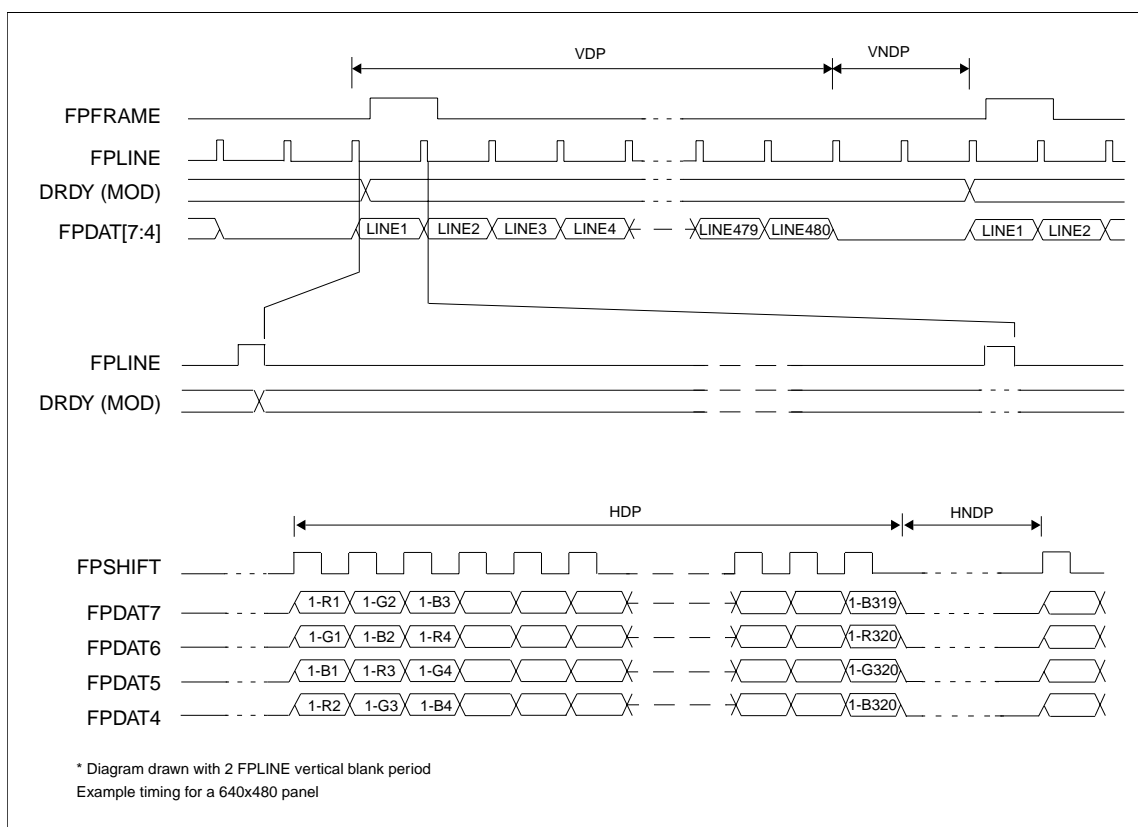HNDP   = Horizontal Non-Display Period   = ((REG[034h] bits [4:0]) + 1) × 8 Ts
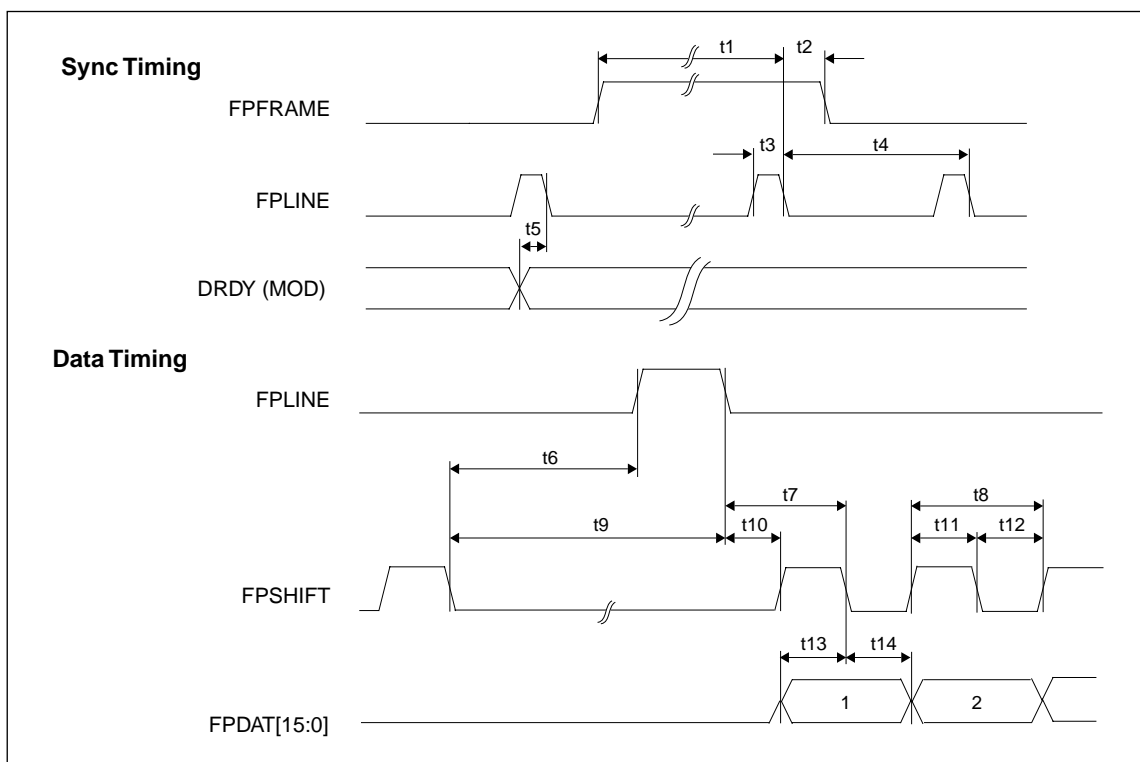
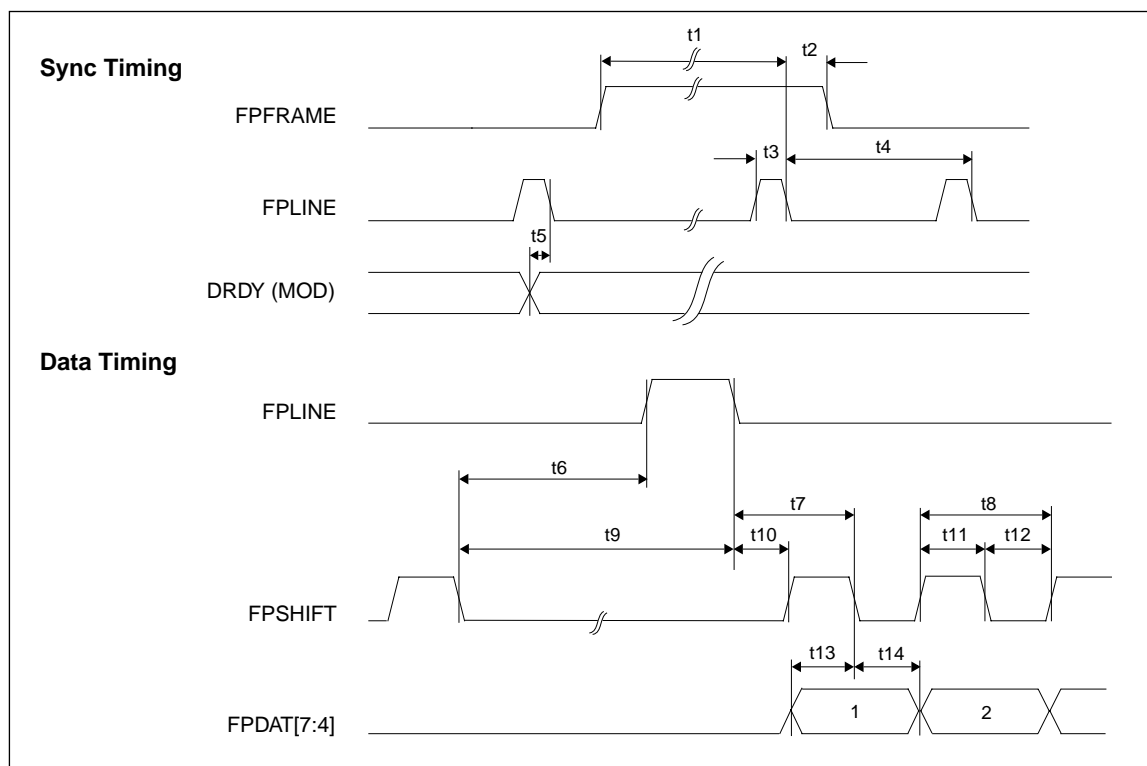Figure 7-42 Dual Color 16-Bit Panel A.C. Timing

Table 7-31 Dual Color 16-Bit Panel A.C. Timing

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3	note 4	259	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	12	note 5	236	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	14	note 5	238	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		12		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		10		Ts
t8	FPSHIFT period		2		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	23	note 6	247	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	25	note 6	249	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		10		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		8		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

Note: 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. $t1 = t4 - 12$ Ts

3. $t4 = [((REG[032h] \text{ bits } [6:0]) + 1) \times 8 + ((REG[034h] \text{ bits } [4:0]) + 1) \times 8]$ Ts

4. $t5 = [((REG[034h] \text{ bits } [6:0]) + 1) \times 8 + 3]$ Ts for (REG[031h] bits [5:0]) $\neq$ 0
   $= 3$ Ts   for (REG[031h] bits [5:0]) = 0

5. $t6 = [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 20]$ Ts for 4 bpp or 8 bpp color depth
   $= [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 18]$ Ts for 15/16 bpp color depth

6. $t9 = [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 9]$ Ts for 4 bpp or 8 bpp color depth
   $= [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 7]$ Ts for 15/16 bpp color depth

7.5.11 Dual Color 16-Bit Panel Timing with External Circuit
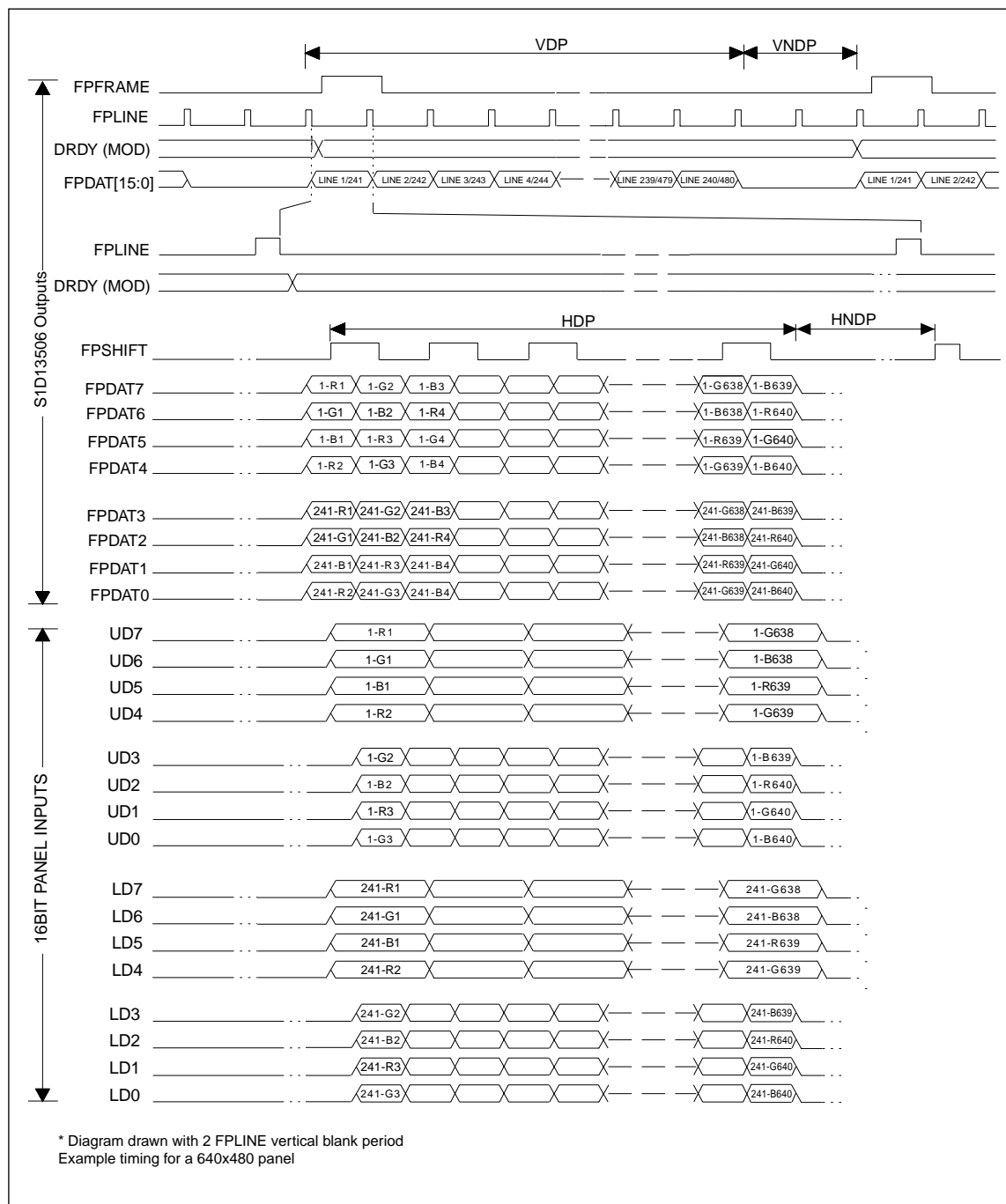
Figure 7-43 16-Bit Dual Color Panel Timing with External Circuit

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) $\times$ 8 Ts
HNDP	= Horizontal Non-Display Period	= ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts
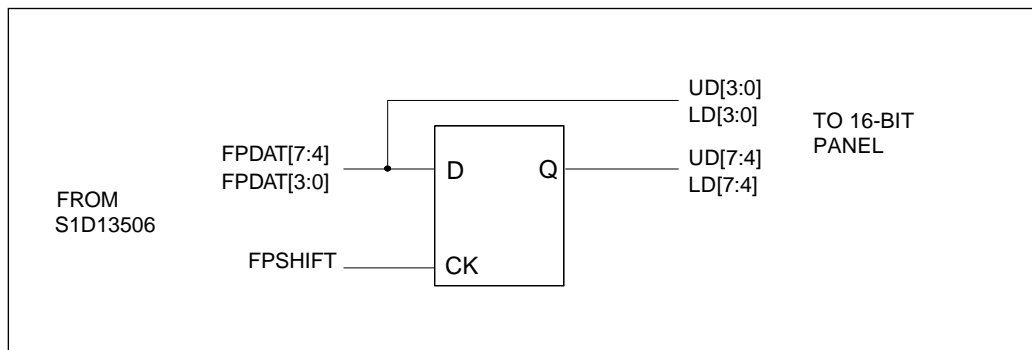
Figure 7-44 External Circuit for Color Dual 16-Bit Panel When the Media Plug is Enabled
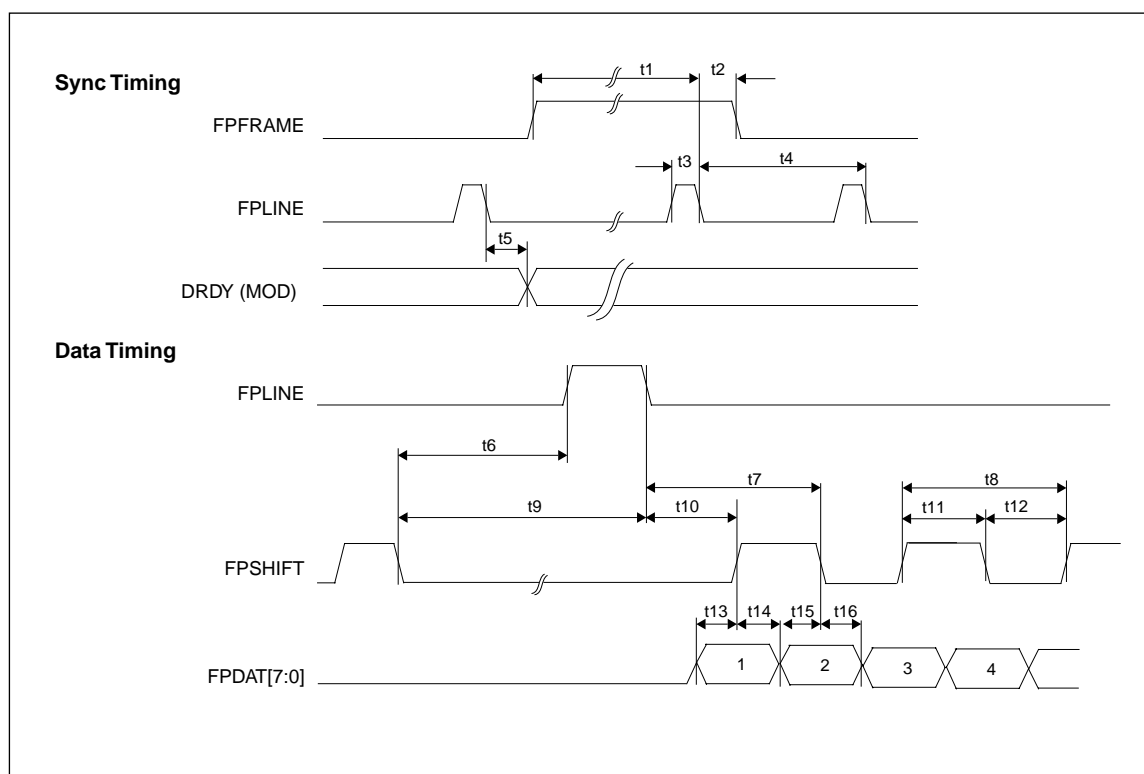
Figure 7-45 Dual Color 16-Bit Panel (with External Circuit) A.C. Timing

EPSON

S1D13506F00A HARDWARE FUNCTIONAL
SPECIFICATION (X25B-A-001-08)

Table 7-32 Dual Color 16-Bit Panel (with External Circuit) A.C. Timing

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPFRAME setup to FPLINE falling edge	28	note 2	1268	Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge		12		Ts
t3	FPLINE pulse width		11		Ts
t4	FPLINE period	40	note 3	1280	Ts
t5	DRDY (MOD) delay from FPLINE falling edge	3	note 4	259	Ts
t6a	FPSHIFT falling edge to FPLINE rising edge, 4 bpp or 8 bpp	12.5	note 5	236.5	Ts
t6b	FPSHIFT falling edge to FPLINE rising edge, 15/16 bpp	14.5	note 5	238.5	Ts
t7a	FPLINE falling edge to FPSHIFT falling edge, 4 bpp or 8 bpp		12.5		Ts
t7b	FPLINE falling edge to FPSHIFT falling edge, 15/16 bpp		10.5		Ts
t8	FPSHIFT period		2		Ts
t9a	FPSHIFT falling edge to FPLINE falling edge, 4 bpp or 8 bpp	23.5	note 6	247.5	Ts
t9b	FPSHIFT falling edge to FPLINE falling edge, 15/16 bpp	25.5	note 6	249.5	Ts
t10a	FPLINE falling edge to FPSHIFT rising edge, 4 bpp or 8 bpp		11.5		Ts
t10b	FPLINE falling edge to FPSHIFT rising edge, 15/16 bpp		9.5		Ts
t11	FPSHIFT pulse width high		1		Ts
t12	FPSHIFT pulse width low		1		Ts
t13	FPDAT[7:0] setup to FPSHIFT rising edge		0.5		Ts
t14	FPDAT[7:0] hold to FPSHIFT rising edge		0.5		Ts
t15	FPDAT[7:0] setup to FPSHIFT falling edge		0.5		Ts
t16	FPDAT[7:0] hold to FPSHIFT falling edge		0.5		Ts

Note: 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. $t1 = t4 - 12$ Ts

3. $t4 = [((REG[032h] \text{ bits } [6:0]) + 1) \times 8 + ((REG[034h] \text{ bits } [4:0]) + 1) \times 8]$ Ts

4. $t5 = [((REG[034h] \text{ bits } [6:0]) + 1) \times 8 + 3]$ Ts for (REG[031h] bits [5:0]) ≠ 0
   = 3 Ts   for (REG[031h] bits [5:0]) = 0

5. $t6 = [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 19.5]$ Ts for 4 bpp or 8 bpp color depth
   = $[((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 17.5]$ Ts for 15/16 bpp color depth

6. $t9 = [((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 8.5]$ Ts for 4 bpp or 8 bpp color depth
   = $[((REG[034h] \text{ bits } [4:0]) + 1) \times 8 - 6.5]$ Ts for 15/16 bpp color depth

7.5.12 16-Bit TFT/D-TFD Panel Timing
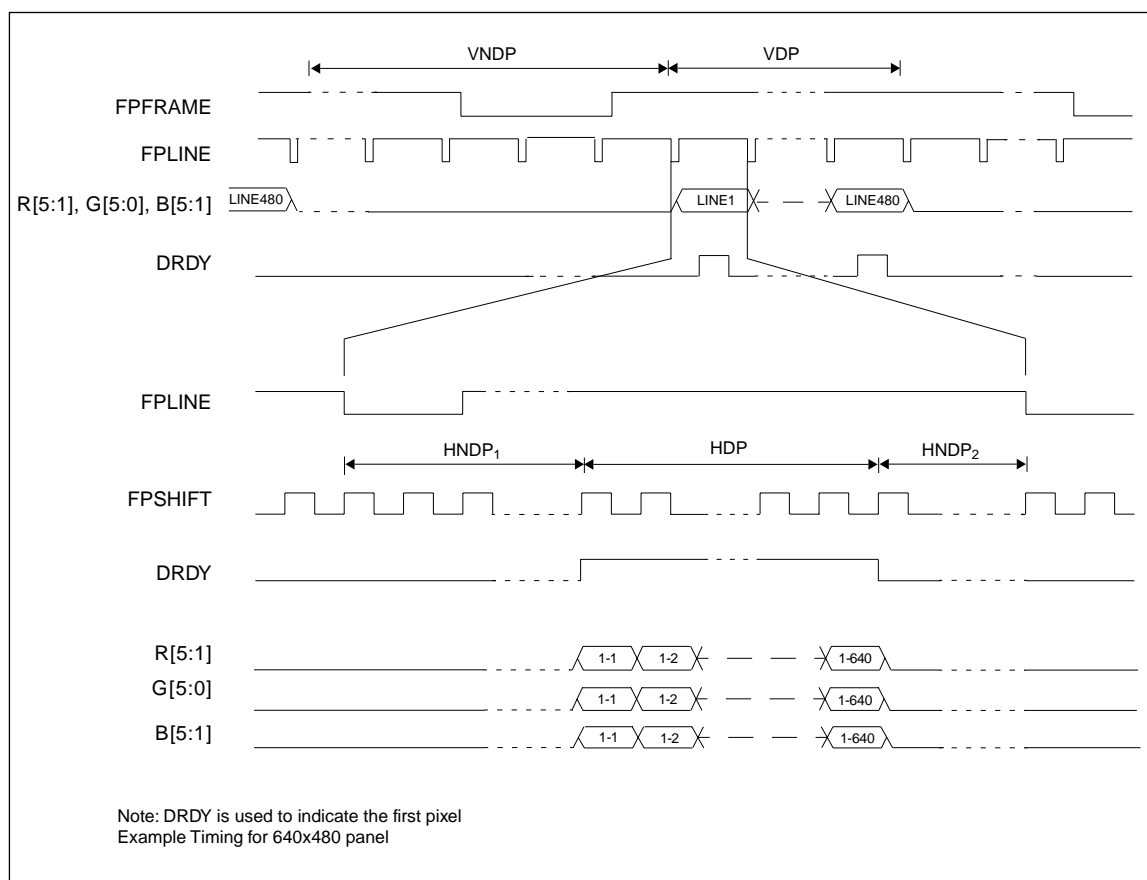
Figure 7-46 16-Bit TFT/D-TFD Panel Timing

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) $\times$ 8 Ts

HNDP   = Horizontal Non-Display Period = $HNDP_1 + HNDP_2$ = ((REG[034h] bits [4:0]) + 1) $\times$ 8 Ts

EPSON   S1D13506F00A HARDWARE FUNCTIONAL
SPECIFICATION (X25B-A-001-08)
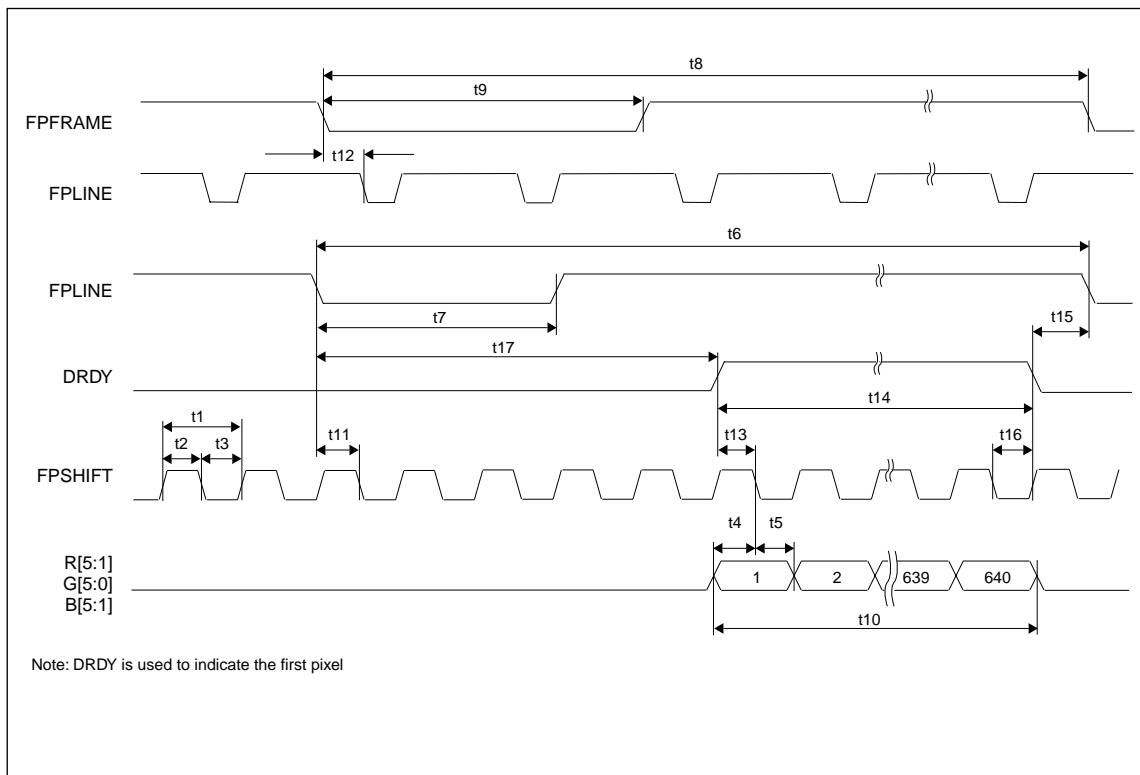
Figure 7-47 TFT/D-TFD A.C. Timing

Table 7-33 TFT/D-TFD A.C. Timing

Symbol	Parameter	Min. Setting	Typ.	Max. Setting	Units
t1	FPSHIFT period		1		Ts (note 1)
t2	FPSHIFT pulse width high		0.5		Ts
t3	FPSHIFT pulse width low		0.5		Ts
t4	Data setup to FPSHIFT falling edge		0.5		Ts
t5	Data hold from FPSHIFT falling edge		0.5		Ts
t6	FPLINE cycle time	40 Ts	note 2	1280 Ts	
t7	FPLINE pulse width low	8 Ts	note 3	128 Ts	
t8	FPFRAME cycle time	2	note 4	1088	lines
t9	FPFRAME pulse width low	1	note 5	8	lines
t10	Horizontal display period	8 Ts	note 6	1024 Ts	
t11	FPLINE setup to FPSHIFT falling edge		0.5		Ts
t12	FPFRAME falling edge to FPLINE falling edge phase difference	Ts	note 7	249 Ts	
t13	DRDY to FPSHIFT falling edge setup time		0.5		Ts
t14	DRDY pulse width	8 Ts	note 8	1024 Ts	
t15a	DRDY falling edge to FPLINE falling edge, 4 bpp or 8 bpp	4 Ts	note 9	252 Ts	
t15b	DRDY falling edge to FPLINE falling edge, 15/16 bpp	6 Ts	note 9	254 Ts	
t16	DRDY hold from FPSHIFT falling edge		0.5		Ts
t17a	FPLINE Falling edge to DRDY active, 4 bpp or 8 bpp	-6 Ts	note 10	250 Ts	
t17b	FPLINE Falling edge to DRDY active, 15/16 bpp	-8 Ts	note 10	248 Ts	

Note: 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. t6 = [((REG[032h] bits [6:0]) + 1) × 8 + ((REG[034h] bits [4:0]) + 1) × 8] Ts

3. t7 = [((REG[036h] bits [3:0]) + 1) × 8] Ts

4. t8 = [((REG[039h] bits [1:0], REG[038h] bits [7:0]) + 1) + ((REG[03Ah] bits [5:0]) + 1)] lines

5. t9 = [((REG[03Ch] bits [2:0]) + 1)] lines

6. t10 = [((REG[032h] bits [6:0]) + 1) × 8] Ts

7. t12 = [(REG[035h] bits [4:0]) × 8 + 1] Ts

8. t14 = [((REG[032h] bits [6:0]) + 1) × 8] Ts

9. t15 = [(REG[035h] bits [4:0]) × 8 + 4] Ts for 4 bpp or 8 bpp color depth= [(REG[035h] bits [4:0]) × 8 + 6] Ts for 15/16 bpp color depth

10. t17 = [((REG[034h] bits [4:0]) + 1) × 8 - (REG[035h] bits [4:0]) × 8 - 6] Ts for 4 bpp or 8 bpp color depth

   = [((REG[034h] bits [4:0]) + 1) × 8 - (REG[035h] bits [4:0]) × 8 - 8] Ts for 15/16 bpp color depth

7.5.13 CRT Timing
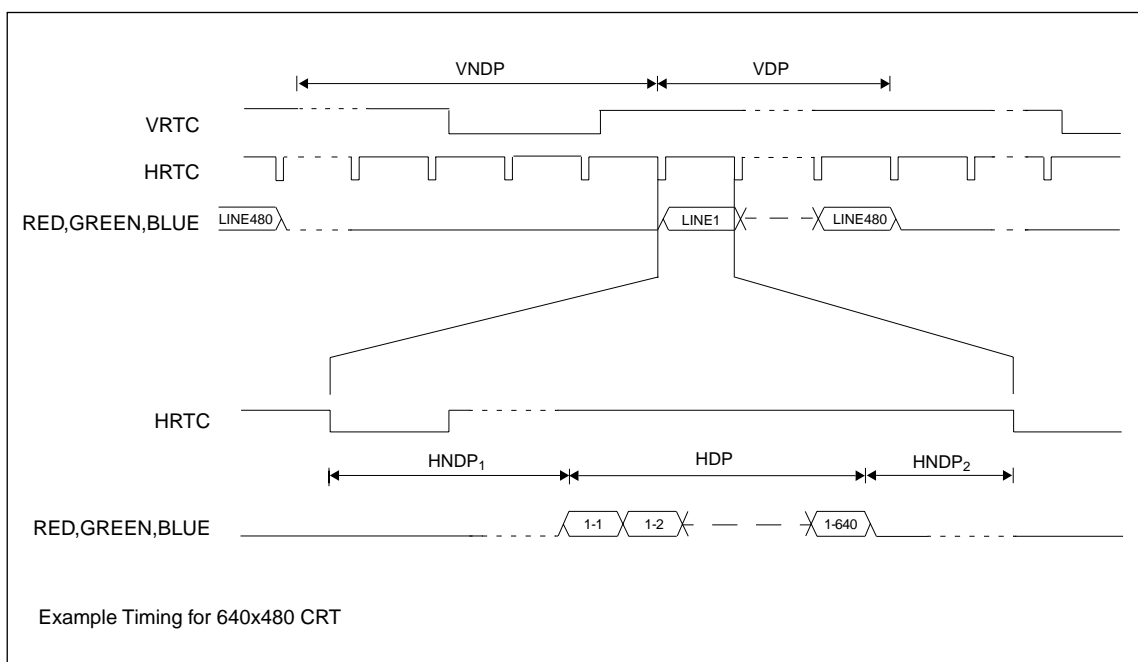
Figure 7-48 CRT Timing

VDP = Vertical Display Period = (REG[057h] bits [1:0], REG[056h] bits [7:0]) + 1
VNDP = Vertical Non-Display Period = (REG[058h] bits [5:0]) + 1
HDP = Horizontal Display Period = ((REG[050h] bits [6:0]) + 1) $\times$ 8 Ts
HNDP = Horizontal Non-Display Period = $HNDP_1$ + $HNDP_2$ = ((REG[052h] bits [5:0]) + 1) $\times$ 8 Ts

Note: The signals RED, GREEN and BLUE are analog signals from the embedded DAC and represent the color components which make up each pixel.
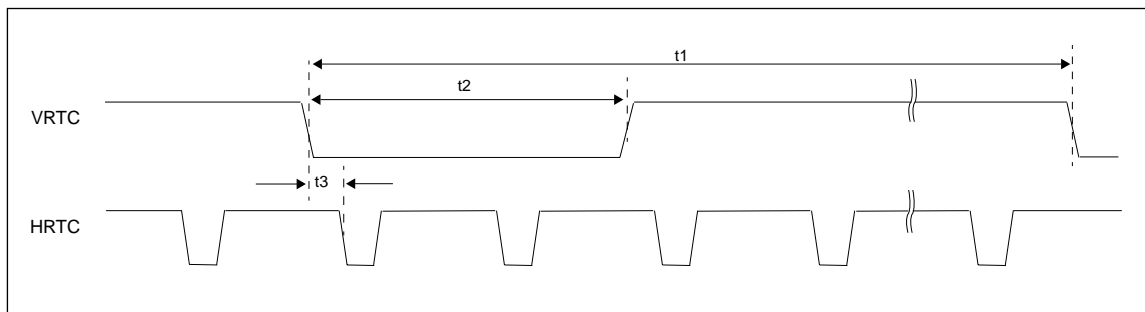
Figure 7-49 CRT A.C. Timing

Table 7-34 CRT A.C. Timing

Symbol	Parameter	Min. Setting	Typical	Max. Setting	Units
t1	VRTC cycle time	2	note 1	1152	lines
t2	VRTC pulse width low	1	note 2	8	lines
t3	VRTC falling edge to FPLINE falling edge phase difference	8	note 3	512	Ts

Note: 1. t1 = [((REG[057h] bits [1:0], REG[056h] bits [7:0]) + 1) + ((REG[058h] bits [6:0]) + 1)] lines

2. t2 = [((REG[05Ah] bits [2:0]) + 1)] lines

3. t3 = [((REG[053h] bits [5:0]) + 1) $\times$ 8] Ts

7.6 TV Timing

7.6.1 TV Output Timing

The overall NTSC and PAL video timing is shown in Figure 7-50 and Figure 7-51 respectively.
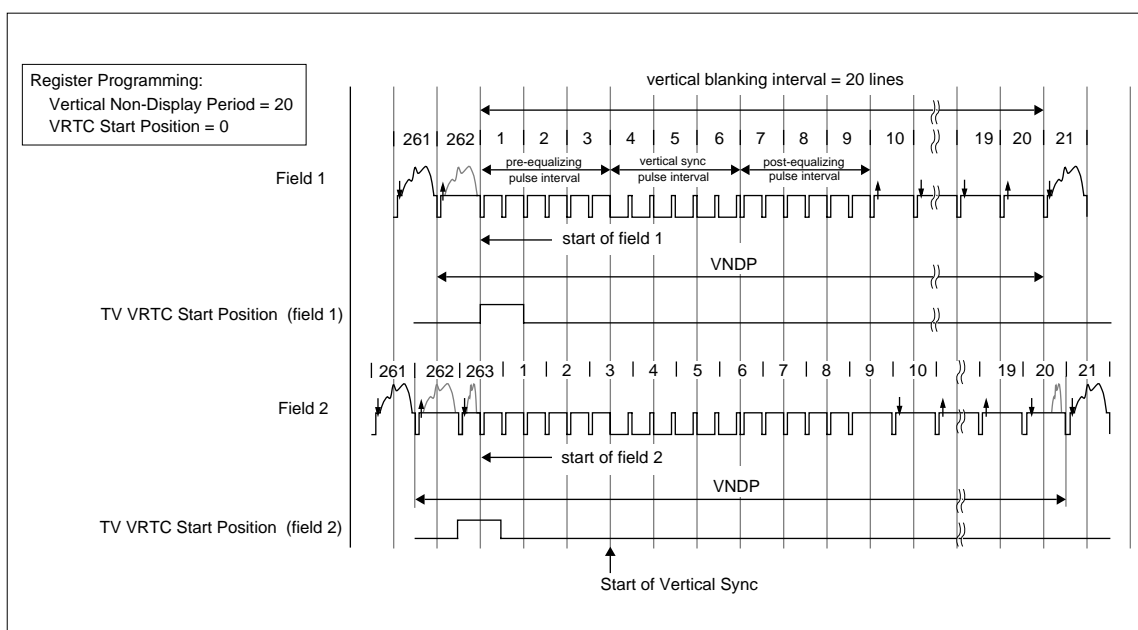
Figure 7-50 NTSC Video Timing
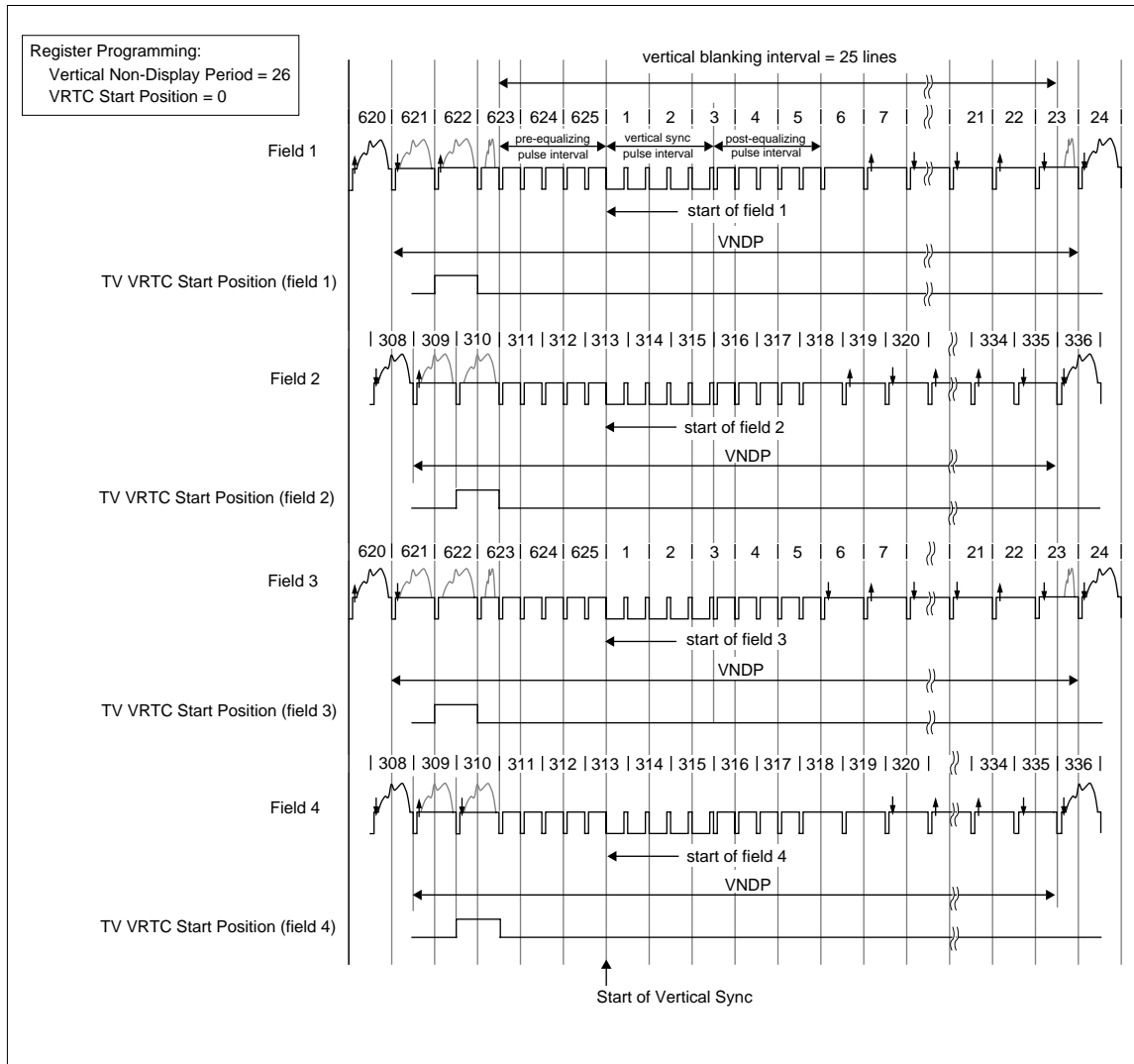
Figure 7-51 PAL Video Timing
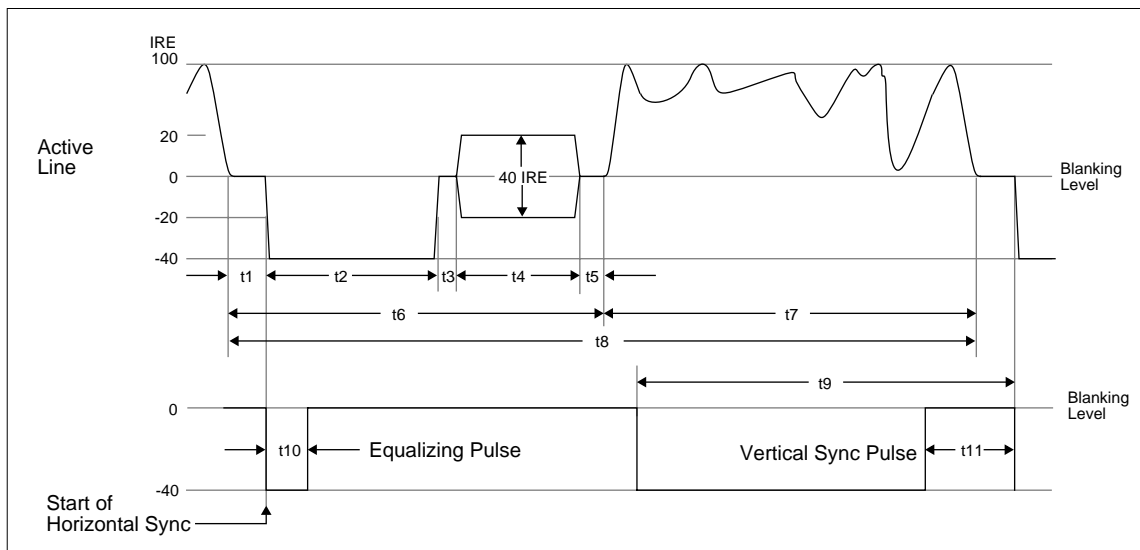
Figure 7-52 Horizontal Timing for NTSC/PAL

Table 7-35 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}

Note:
1. $t1$ = $((REG[053] \, bits[5:0]) + 1) \times 8 - 7$ (4 bpp, 8 bpp modes)

 = $((REG[053] \, bits[5:0]) + 1) \times 8 - 5$ (15/16 bpp modes)

2. $t5_{NTSC}$ = $(((REG[052] \, bits[5:0]) \times 8) + 6) - (((REG[053] \, bits[5:0]) + 1) \times 8) - 108$ (4 bpp, 8 bpp modes)

 = $(((REG[052] \, bits[5:0]) \times 8) + 6) - (((REG[053] \, bits[5:0]) + 1) \times 8) - 110$ (15/16 bpp modes)

3. $t5_{PAL}$ = $(((REG[052] \, bits[5:0]) \times 8) + 7) - (((REG[053] \, bits[5:0]) + 1) \times 8) - 136$ (4 bpp, 8 bpp modes)

 = $(((REG[052] \, bits[5:0]) \times 8) + 7) - (((REG[053] \, bits[5:0]) + 1) \times 8) - 138$ (15/16 bpp modes)

4. $t6_{NTSC}$ = $((REG[052] \, bits[5:0]) \times 8) + 6$
5. $t6_{PAL}$ = $((REG[052] \, bits[5:0]) \times 8) + 7$
6. $t7$ = $((REG[050] \, bits[6:0]) + 1) \times 8$

Important:

REG[050] and REG[052] must be programmed to satisfy the Line Period (t8).

For NTSC, $(((REG[050] \, bits[6:0]) + 1) \times 8) + (((REG[052] \, bits[5:0]) \times 8) + 6) = 910$.

For PAL, $(((REG[050] \, bits[6:0]) + 1) \times 8) + (((REG[052] \, bits[5:0]) \times 8) + 7) = 1135$.
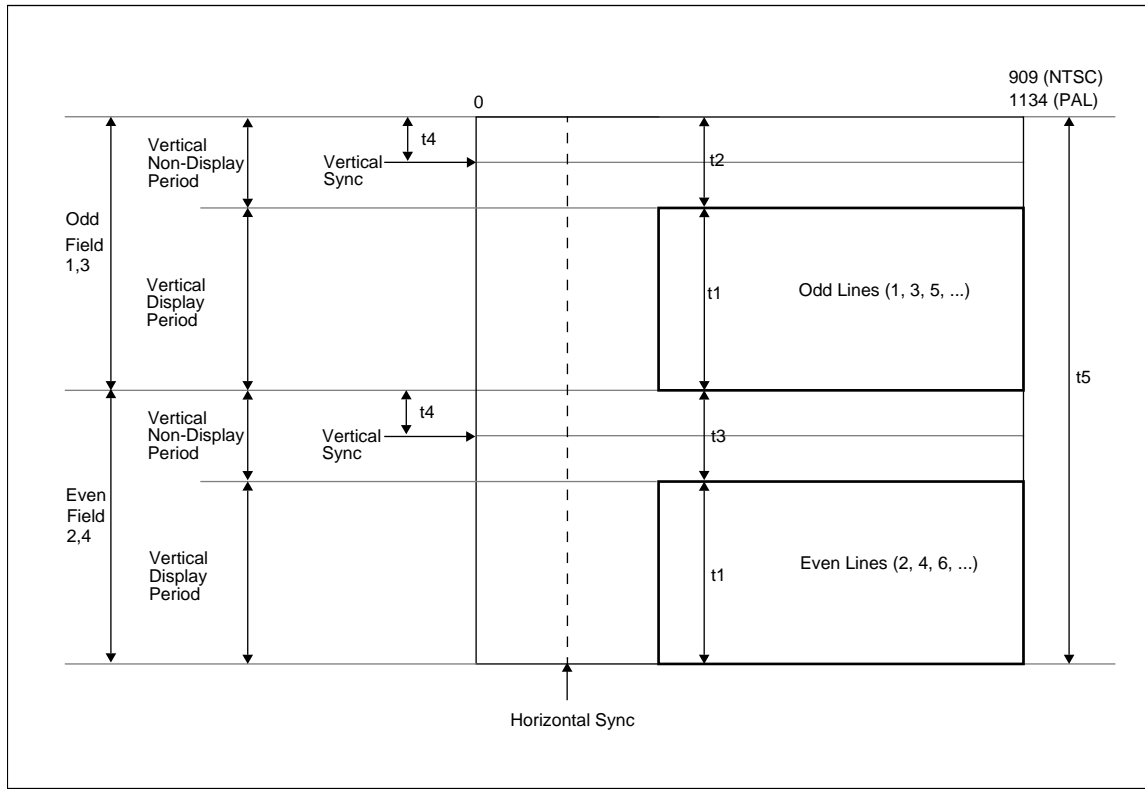
Figure 7-53 Vertical Timing for NTSC/PAL

Table 7-36 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 Blanking (Fields 1, 3)	note 2	note 2	T_{LINE}
t3	Vertical Blanking (Fields 2, 4)	note 3	note 3	T_{LINE}
t4	Vertical Sync Position	note 4	note 5	T_{LINE}
t5	Frame Period	525	625	T_{LINE}

Note: 1. t1   = ({(REG[057] bits[1:0]), (REG[056] bits[7:0])} + 1) / 2

2. t2   = ((REG[058] bits[6:0]) + 1) for fields 1 & 3

3. t3   = ((REG[058] bits[6:0]) + 2) for fields 2 & 4

4. $t4_{NTSC}$ = ((REG[059] bits[6:0]) + 4)   for field 1

= ((REG[059] bits[6:0]) + 4.5)   for field 2

5. $t4_{PAL}$ = ((REG[059] bits[6:0]) + 5)   for field 1 and field 3

= ((REG[059] 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) $\times$ 2 + 1) = 525

For PAL, ({(REG[057] bits[1:0]), (REG[056] bits[7:0])} + 1) + ((REG[058] bits[6:0]) + 1) $\times$ 2 + 1) = 625

7.7 MediaPlug Interface Timing
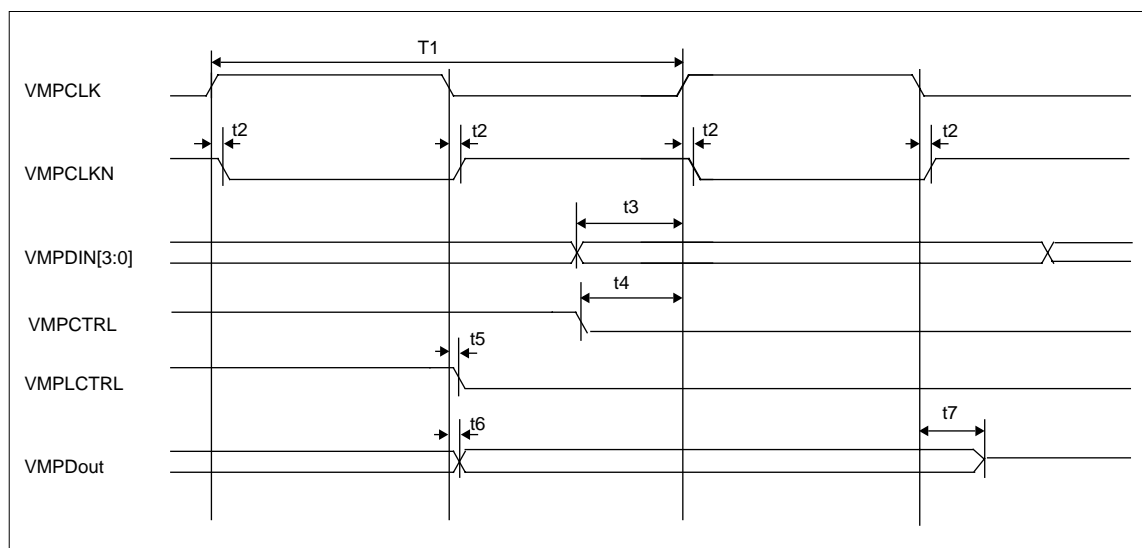
Figure 7-54 MediaPlug A.C. Timing

Note: The above timing diagram assumes no load.

Table 7-37 MediaPlug A.C. Timing

Symbol	Parameter	Min.	Max.	Units
T1	MediaPlug clock period	100		ns
t2	VMPCLKN delay from VMPCLK	0	3	ns
t3	Input data setup	6		ns
t4	VMPCTRL setup	6		
t5	Local control signal delay from VMPCLK falling edge		2	ns
t6	Output data delay from VMPCLK falling edge		1	ns
t7	Output data tristate delay from VMPCLK falling edge		14	ns

EPSON

8 REGISTERS

This section discusses how and where to access the S1D13506 registers. It also provides detailed information about the layout and usage of each register.

8.1 Register Mapping

The S1D13506 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 A20-A0.

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 A19-A0. When A20 = 0 and A12 = 0 the registers are decoded using A8-A0 as an index. When A20 = 0 and A12 = 1 the MediaPlug register ports are decoded using A11-A0.

The MediaPlug register ports are defined only when configuration input MD13 = 1 on reset. When MD13 = 0 on reset, A12 is always treated as 0 and the MediaPlug register space is not available - see Table 5-6, "Summary of Power-On/Reset Options" on page 1-23.

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)	100000h to 1FFFFFh
MediaPlug registers (4K bytes)	1000h to 1FFFh
Main 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.

8.2 Register Descriptions

8.2.1 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 is a read-only register that indicates the product code of the chip. The product code for S1D13506 is 000100b.

bits 1-0 Revision Code Bits [1:0]
This is a read-only register that indicates the revision code of the chip. The revision code is 01b.

Miscellaneous Register
REG[001h]   RW

Register/ Memory Select	n/a	n/a	n/a	n/a	Reserved	Reserved	Reserved

bit 7 Register Memory Select
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.

8.2.2 General IO Pins Registers

General IO Pins Configuration Register
REG[004h]   RW

Reserved	Reserved	Reserved	Reserved	GPIO3 Pin IO Config.	GPIO2 Pin IO Config.	GPIO1 Pin IO Config.	Reserved

bit 3   GPIO3 Pin IO Configuration
When this bit = 1, GPIO3 is configured as an output pin.
When this bit = 0 (default), GPIO3 is configured as an input pin.

Note: Note that MD[7:6] must be properly configured at the rising edge of RESET# to enable GPIO3, otherwise GPIO3 will be used as MA[9] for the DRAM and this bit will have no hardware effect. (See Table 8-2, "MA[11:9]/GPIO[1:3] Pin Functionality").

bit 2   GPIO2 Pin IO Configuration
When this bit = 1, GPIO2 is configured as an output pin.
When this bit = 0 (default), GPIO2 is configured as an input pin.

Note: Note that MD[14] and MD[7:6] must be properly configured at the rising edge of RESET# to enable GPIO2, otherwise GPIO2 will be used as MA[11] for the DRAM or as the MediaPlug VMPEPWR output and this bit will have no hardware effect. (See Table 8-2, "MA[11:9]/GPIO[1:3] Pin Functionality").

bit 1   GPIO1 Pin IO Configuration
When this bit = 1, GPIO1 is configured as an output pin.
When this bit = 0 (default), GPIO1 is configured as an input pin.

Note: Note that MD[7:6] must be properly configured at the rising edge of RESET# to enable GPIO1, otherwise GPIO1 will be used as MA[10] for the DRAM and this bit will have no hardware effect. (See Table 8-2, "MA[11:9]/GPIO[1:3] Pin Functionality").

Table 8-2 MA[11:9]/GPIO[1:3] Pin Functionality

MD[14] on Reset	MD[7:6] on Reset	Pin		
		MA[9]/GPIO3	MA[10]/GPIO1	MA[11]/GPIO2
0	00	GPIO3	GPIO1	GPIO2
0	01	MA9	GPIO1	GPIO2
0	10	MA9	GPIO1	GPIO2
0	11	MA9	MA10	MA11
1	00	GPIO3	GPIO1	VMPEPWR
1	01	MA9	GPIO1	VMPEPWR
1	10	MA9	GPIO1	VMPEPWR
1	11	MA9	MA10	MA11

General IO Pins Control Register

REG[008h]							RW
Reserved	Reserved	Reserved	Reserved	GPIO3 Pin IO Status	GPIO2 Pin IO Status	GPIO1 Pin IO Status	Reserved

bit 3   GPIO3 Pin IO Status

When GPIO3 is configured as an output, writing a 1 to this bit drives GPIO3 high and writing a 0 to this bit drives GPIO3 low.

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

Note: Note that MD[7:6] must be properly configured at the rising edge of RESET# to enable GPIO3, otherwise GPIO3 will be used as MA9 for the DRAM and this bit will have no hardware effect. (See Table 8-2, "MA[11:9]/GPIO[1:3] Pin Functionality").

bit 2   GPIO2 Pin IO Status

When GPIO2 is configured as an output, writing a 1 to this bit drives GPIO2 high and writing a 0 to this bit drives GPIO2 low.

When GPIO2 is configured as an input, a read from this bit returns the status of GPIO2.

Note: Note that MD[14] and MD[7:6] must be properly configured at the rising edge of RESET# to enable GPIO2, otherwise GPIO2 will be used as MA11 for the DRAM or as the MediaPlug VMPEPWR output and this bit will have no hardware effect. (See Table 8-2, "MA[11:9]/GPIO[1:3] Pin Functionality").

bit 1   GPIO1 Pin IO Status

When GPIO1 is configured as an output, writing a 1 to this bit drives GPIO1 high and writing a 0 to this bit drives GPIO1 low.

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

Note: Note that MD[7:6] must be properly configured at the rising edge of RESET# to enable GPIO1, otherwise GPIO1 will be used as MA10 for the DRAM and this bit will have no hardware effect. (See Table 8-2, "MA[11:9]/GPIO[1:3] Pin Functionality").

8.2.3 MD Configuration Readback Registers

MD Configuration Status Register 0

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

MD Configuration Status Register 1

REG[00Dh]							RO
MD[15] Config. Status	MD[14] Config. Status	MD[13] Config. Status	MD[12] Config. Status	MD[11] Config. Status	MD[10] Config. Status	MD[9] Config. Status	MD[8] Config. Status

REG[00Ch] bits 7-0   MD[15:0] Configuration Status Bits [15:0]

REG[00Dh] bits 7-0   These read-only bits return the status of MD[15:0] at the rising edge of RESET#.

8.2.4 Clock Configuration Registers

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

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

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

bit 0   MCLK Source Select
When this bit = 1, the source for the internal MCLK is derived from BUSCLK.
When this bit = 0, the source for MCLK is derived from CLKI.

Table 8-3 MCLK Source Select

MCLK Source Select	MCLK Source
0	CLKI
1	BUSCLK

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

For further information on MCLK, refer to Section 20.2, "Clock Descriptions" on page 1-199.

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

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 8-4 LCD PCLK Divide Selection

LCD PCLK Divide Select Bits	LCD PCLK Source to LCD PCLK 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 LCD pixel clock for the LCD display.

Table 8-5 LCD PCLK Source Selection

LCD PCLK Source Select Bits	LCD PCLK Source
00	CLKI
01	BUSCLK
10	CLKI2
11	MCLK (see note)

Note: MCLK may be a previously divided down version of CLKI, CLKI2 or BUSCLK.

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

CRT/TV PCLK 2X 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

bit 7   CRT/TV PCLK 2X Enable
            This bit multiplies the CRT/TV pixel clock by 2.
            This bit must be set to 1 when TV with flicker filter is enabled. 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 8-6 CRT/TV PCLK Divide Selection

CRT/TV PCLK Divide Select Bits	CRT/TV PCLK Source to CRT/TV PCLK 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 CRT/TV pixel clock for the CRT/TV display.

Table 8-7 CRT/TV PCLK Source Selection

CRT/TV PCLK Source Select Bits	CRT/TV PCLK Source
00	CLKI
01	BUSCLK
10	CLKI2
11	MCLK (see note)

Note: MCLK may be a previously divided down version of CLKI, CLKI2 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

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

Table 8-8 MediaPlug Clock Divide Selection

MediaPlug Clock Divide Select Bits	MediaPlug Clock Source to CRT/TV Pixel 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 7.7, "MediaPlug Interface Timing" on page 1-105 for AC Timing.

Table 8-9 Video Clock Source Selection

MediaPlug Clock Source Select Bits	MediaPlug Clock Source
00	CLKI
01	BUSCLK
10	CLKI2
11	MCLK (see note)

Note: MCLK may be a previously divided down version of CLKI, CLKI2 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 MD12 (see
            Table 5-6, "Summary of Power-On/Reset Options" on page 1-23).

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

Table 8-10 Minimum Memory Timing Selection

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

8.2.5 Memory Configuration Registers

Memory Configuration Register

REG[020h]   RW

n/a	n/a	n/a	n/a	n/a	n/a	Memory Type Bit 1	Memory Type Bit 0

bits 1-0   Memory Type Bits [1:0]
             These bits specify the memory type.

Table 8-11 Memory Type Selection

Memory Type Bits [1:0]	Memory Type
00	EDO-DRAM with 2-CAS#
01	FPM-DRAM with 2-CAS#
10	EDO-DRAM with 2-WE#
11	FPM-DRAM with 2-WE#

DRAM Refresh Rate Register

REG[021h]   RW

Refresh Select Bit 1	Refresh Select Bit 0	n/a	n/a	n/a	DRAM Refresh Rate Bit 2	DRAM Refresh Rate Bit 1	DRAM Refresh Rate Bit 0

bits 7-6   Refresh Select Bits [1:0]
             These bits specify the type of DRAM refresh used while in power save mode.

Table 8-12 Refresh Selection

Refresh Select Bits [1:0]	DRAM Refresh Type
00	CBR Refresh
01	Self-Refresh
1X	No Refresh

Note: These bits should not be changed while power save mode is enabled.

bits 2-0   DRAM Refresh Rate Select Bits [2:0]
These bits specify the divide used to generate the DRAM refresh clock rate, which is equal to $2^{(\text{ValueOfTheseBits} + 6)}$, from the MCLK source (either BUSCLK or CLKI as determined by REG[010h] bit 0).

Table 8-13 DRAM Refresh Rate Selection

DRAM Refresh Rate Bits [2:0]	MCLK Source Divide Amount	Refresh Rate for 40MHz MCLK Source	DRAM Refresh Time/256 cycles
000	64	625 kHz	0.4 ms
001	128	312 kHz	0.8 ms
010	256	156 kHz	1.6 ms
011	512	78 kHz	3.3 ms
100	1024	39 kHz	6.6 ms
101	2048	20 kHz	13.1 ms
110	4096	10 kHz	26.2 ms
111	8192	5 kHz	52.4 ms

DRAM Timing Control Register 0
REG[02Ah]   RW

DRAM Timing Control Register Bit 7	DRAM Timing Control Register Bit 6	DRAM Timing Control Register Bit 5	DRAM Timing Control Register Bit 4	DRAM Timing Control Register Bit 3	DRAM Timing Control Register Bit 2	DRAM Timing Control Register Bit 1	DRAM Timing Control Register Bit 0

DRAM Timing Control Register 1
REG[02Bh]   RW

n/a	n/a	n/a	n/a	n/a	n/a	DRAM Timing Control Register Bit 9	DRAM Timing Control Register Bit 8

REG[02Ah] bits 4-0   DRAM Timing Control Bits [9:0]

REG[02Bh] bits 1-0   The DRAM Timing Control registers must be set based on the type of DRAM, speed of DRAM, and MCLK frequency used. The following table provides the optimal values for each register.

Table 8-14 DRAM Timing Control Selection

DRAM Type	DRAM Speed (ns)	MCLK Frequency (MHz)	DRAM Timing Control Register 0	DRAM Timing Control Register 1
EDO	50	40	01h	01h
	50	33	01h	01h
	60		01h	01h
	50	30	12h	02h
	60		01h	01h
	70		00h	00h
	50	25	12h	02h
	60		12h	02h
	70		01h	01h
	80		00h	01h
	50	20	12h	02h
	60		12h	02h
	70		12h	02h
	80		01h	01h
FPM	50	25	12h	02h
	60		01h	01h
	50	20	12h	02h
	60		12h	02h
	70		11h	02h
	80		01h	01h

8.2.6 Panel Configuration Registers

Panel Type Register
REG[030h]   RW

EL Panel Mode Enable	n/a	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, the EL Panel support circuit is enabled.
When this bit = 0, there is no hardware effect.
This bit enables the S1D13506 built-in circuit for EL panels which require the Frame Rate Modulation (FRM) to remain static for one frame after 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.

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

Table 8-15 Panel Data Width Selection

Panel Data Width Bits [1:0]	Passive LCD Panel Data Width Size	TFT/D-TFD Panel Data Width Size
00	4-bit	9-bit
01	8-bit	12-bit
10	16-bit	18-bit (64K color)
11	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 7.5.5, "Single Color 8-Bit Panel Timing (Format 2)" on page 1-75.
When this bit = 0, 8-bit single color passive LCD panel data format 1 is selected. For AC timing see Section 7.5.4, "Single Color 8-Bit Panel Timing (Format 1)" on page 1-72.

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

bit 1   Dual/Single Panel Select
When this bit = 1, dual LCD panel is selected.
When this bit = 0, single 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 = $((\text{ContentsOfThisRegister}) + 1) \times 8$

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 will result in undefined behavior.

Table 8-16 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).

EPSON   S1D13506F00A HARDWARE FUNCTIONAL
                                                                                 SPECIFICATION (X25B-A-001-08)

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 horizontal non-display period width in 8 pixel resolution.

Horiz. non-display period width in number of pixels = $((\text{ContentsOfThisRegister}) + 1) \times 8$

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

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 panel 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 4/8 bpp color depth:
FPLINE start position in number of pixels = $[(\text{ContentsOfThisRegister}) \times 8 + 4]$

For 15/16 bpp color depth:
FPLINE start position in number of pixels = $[(\text{ContentsOfThisRegister}) \times 8 + 6]$

Note: For TFT/D-TFD only:
REG[034h] + 1 $\geq$ (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 8-17 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 panel 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) $\times$ 8

            The maximum FPLINE pulse width is 128 pixels.

Note: For TFT/D-TFD only:
           REG[034h] + 1 $\geq$ (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 panel 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: For TFT/D-TFD only:
             (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 8-18 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 panel 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: For TFT/D-TFD only:
            (REG[03Ah] bits 5-0 + 1) $\geq$ (REG[03Bh] + 1) + (REG[03Ch] bits 2-0 + 1)

8.2.7 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 1-178 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 8-19 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: 15/16 bpp color depths bypass the LUT. Passive panels are supported up to 32K/64K
     colors (4096 colors if dithering disabled, see REG[041h] bit 1). TFT/D-TFD panels are
     supported up to 32K/64K colors.

Table 8-20 LCD 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	15 bpp
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 15/16 bpp mode is disabled allowing a maximum of 4096 colors (2^{12}) or 16 gray shades.
When this bit = 0, dithering on the passive LCD panel for 15/16 bpp mode is enabled allowing a maximum of 64K colors (2^{16}) or 64 gray shades.
The dithering algorithm provides more shades of each primary color when in 15/16 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.

The Dual Panel Buffer is needed to fully support dual panels. Disabling the Dual Panel Buffer will improve performance, reduce power consumption, and allow higher resolution/ color display modes than would otherwise be possible; however, disabling the Dual Panel Buffer will reduce image contrast and overall display quality. This mode is not normally used except in special circumstances such as simultaneous display on a CRT/ TV and dual panel LCD. For details on Frame Rate Calculation, see Section 17.

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 for the starting word of the LCD image in the display buffer.

REG[044h] bits 3-0   **Note that this is a word address.** An entry of 00000h into these registers represents the first word of display memory, an entry of 00001h represents the second word of the display memory, 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	Reserved	Reserved	LCD Pixel Panning Bit 1	LCD Pixel Panning Bit 0

bits 3-2   Reserved.
            Must be set to 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 8-21 LCD Pixel Panning Selection

Color Depth (bpp)	Screen 2 Pixel Panning Bits Used
4 bpp	Bits [1:0]
8 bpp	Bit 0
15/16 bpp	–

Smooth horizontal panning can be achieved by a combination of this register and the LCD Display Start Address register.

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 1-190).

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 16.2, "Bandwidth Limitation" on page 1-186).

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

bits 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

8.2.8 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 = $((\text{ContentsOfThisRegister}) + 1) \times 8$

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 =
$((\text{ContentsOfThisRegister}) + 1) \times 8$ for CRT mode
$(\text{ContentsOfThisRegister}) \times 8 + 6$ for TV mode with NTSC output
$(\text{ContentsOfThisRegister}) \times 8 + 7$ for TV mode with PAL output

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

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=:
$[(\text{ContentsOfThisRegister}) \times 8 + 3]$ for CRT with 4/8 bpp color depth
$[(\text{ContentsOfThisRegister}) \times 8 + 5]$ for CRT in 15/16 bpp color depth
$[((\text{ContentsOfThisRegister}) + 1) \times 8 - 7]$ for TV-NTSC in 4/8 bpp color depth
$[((\text{ContentsOfThisRegister}) + 1) \times 8 - 5]$ for TV-NTSC in 15/16 bpp color depth
$[((\text{ContentsOfThisRegister}) + 1) \times 8 - 7]$ for TV-PAL in 4/8 bpp color depth
$[((\text{ContentsOfThisRegister}) + 1) \times 8 - 5]$ for TV-PAL in 15/16 bpp color depth

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

CRT HRTC Pulse Width Register
REG[054h]   RW

			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 NTSC/PAL modes, this bit must be set to 0b.

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 = $((\text{ContentsOfThisRegister}) + 1) \times 8$

Note: For NTSC/PAL modes, these bits must be set to 0000b.

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　　CRT/TV Vertical Display Height Bits [9:0]

REG[057h] bits 1-0　　These bits specify the CRT/TV vertical display height, in 1 line resolution.

Vertical display height in number of lines = $(\text{ContentsOfThisRegister}) + 1$

　　　　　EPSON　　　　　**S1D13506F00A HARDWARE FUNCTIONAL SPECIFICATION (X25B-A-001-08)**

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 5-0 + 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 5-0 + 1) ≥ (REG[059h] + 1) + (REG[05Ah] bits 2-0 + 1)

CRT/TV 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 PAL/NTSC, this bit must be set to 0b.

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 NTSC/PAL modes, these bits should be set to 000b.

            (REG[058h] bits 5-0 + 1) ≥ (REG[059h] + 1) + (REG[05Ah] bits 2-0 + 1)

CRT/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, there is no hardware effect.
          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, there is no hardware effect.
          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
          This bit should be set based on the conditions in the following table. When this bit is set
          to 1 it allows IREF to be reduced. For an example implementation of the required external
          CRT/TV circuitry, see Figure 5-2 "External Circuitry for CRT/TV Interface," on page 1-
          28.

Table 8-22 DAC Output Level Selection

LCD	CRT	TV	REG[05Bh] bit 3	IREF (mA)
Supported	Not Supported	Not Supported	x	x
x	Supported	Not Supported	1	4.6
x	Supported	Supported	0	9.2

x	= don't care

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 mode is enabled.

8.2.9 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 (i.e., the screen is blanked).
When this bit = 0 the CRT/TV 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: 15/16 bpp color depths bypass the LUT.

Table 8-23 CRT/TV Bit-per-pixel Selection

Bit-per-pixel Select Bits [2:0]	Color Depth (bpp)
000	Reserved
001	Reserved
010	4 bpp
011	8 bpp
100	15 bpp
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 display buffer. **Note that this is a word address.** An entry of 00000h into these registers

REG[064h] bits 3-0   represents the first word of display memory, an entry of 00001h represents the second word of the display memory, 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.

EPSON   S1D13506F00A HARDWARE FUNCTIONAL
SPECIFICATION (X25B-A-001-08)

CRT/TV Pixel Panning Register
REG[068h]   RW

n/a	n/a	n/a	n/a	Reserved	Reserved	CRT/TV Pixel Panning Bit 1	CRT/TV Pixel Panning Bit 0

bits 3-2   Reserved.
            Must be set to 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 8-24 CRT/TV Pixel Panning Selection

Color Depth (bpp)	Screen 2 Pixel Panning Bits Used
4 bpp	Bits [1:0]
8 bpp	Bit 0
15/16 bpp	–

Smooth horizontal panning can be achieved by a combination of this register and the
CRT/TV Display Start Address register.

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 1-190).

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 16.2, "Bandwidth
            Limitation" on page 1-186).

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[04Ah] from its default value, then the following formula must be maintained:

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

8.2.10 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 8-25 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 & 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, where

$M = (\text{Panel Height} \times \text{Panel Width} / 16) \times c$, $c = 1$ for monochrome, 4 for color panel.

Table 8-26   LCD Ink/Cursor Start Address Encoding

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

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

See Section 10, "Display Buffer" on page 1-161 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.

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[074h] 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.

8.2.11 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 8-27 CRT/TV Ink/Cursor Selection

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

Note: During Ink mode, the Cursor X & Y Position registers must be programmed to zero.

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 M memory locations in bytes, where

$M = (\text{Panel Height} \times \text{Panel Width} / 16) \times c$, $c = 1$ for monochrome, 4 for color panel.

Table 8-28 CRT/TV Ink/Cursor Start Address Encoding

CRT/TV Ink/Cursor Start Address Bits [7:0]	Start Address
0	Memory Size - 1024
n = 255...1	Memory Size - $n \times 8192$

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

See Section 10, "Display Buffer" on page 1-161 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 shall 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.

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 Y Sign
When this bit = 1, it defines the CRT/TV Cursor Y Position register as a negative number. The negative number shall not exceed 63 decimal.
When this bit = 0, it defines the CRT/TV Cursor Y Position register as 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 5-0   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.

8.2.12 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

bits 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 will initiate 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 8-29 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 = 1, the BitBlt FiFO has at least one data.
When this bit = 0, the BitBlt FIFO is empty.
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 data available in BitBlt FIFO under different status conditions.

Table 8-30 BitBlt FIFO Data 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 Data 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.
Software can use this bit to optimize BitBlt write burst operations.
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.
Software can use this bit to optimize BitBlt write burst operations.
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 Blit is stored as a contiguous linear block of memory.
When this bit = 0, the Destination Blit 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 Blit is stored as a contiguous linear block of memory.
When this bit = 0, the Source Blit 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.
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 Blit and Move Blit. Bits 2-0 also specify the start bit position for Color Expansion.

Table 8-31 BitBlt ROP Code/Color Expansion Function Selection

BitBlt ROP Code Bits [3:0]	Boolean Function for Write Blit and Move Blit	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 8-32 BitBlt Operation Selection

BitBlt Operation Bits [3:0]	Blit Operation
0000	Write Blit with ROP.
0001	Read Blit.
0010	Move Blit in positive direction with ROP.
0011	Move Blit in negative direction with ROP.
0100	Transparent Write Blit.
0101	Transparent Move Blit in positive direction.
0110	Pattern Fill with ROP.
0111	Pattern Fill with transparency.
1000	Color Expansion.
1001	Color Expansion with transparency.
1010	Move Blit with Color Expansion.
1011	Move Blit 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 $\geq$ 2. 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 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 8-33 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]

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 Blit 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.

Note: The BitBlt operations Pattern Fill with ROP and Pattern Fill with transparency require a BitBlt width ≥ 2.

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 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 Blit. For 16 bpp mode (REG[101h] bit 0 = 1), all 16 bits are used. For 8 bpp mode (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 mode (REG[101h] bit 0 = 1), all 16 bits are used. For 8 bpp mode (REG[101h] bit 0 = 0), only bits 7-0 are used.

     EPSON

8.2.13 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. Additionally, access to the LUT registers is not permitted during power save mode. For further information on the clocks, see Section 20, "Clocks" on page 1-198.

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 Look-Up Tables (LCD and CRT/TV) are accessible by REG[1E2h] and REG[1E4h].

Table 8-34 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 S1D13506 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 1-165 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.

EPSON

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.

8.2.14 Power Save Configuration Registers

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 0.

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

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.

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 is either in self refresh or no refresh mode.
When this bit = 0, the memory controller is powered up and is either in CBR refresh or normal mode.

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

8.2.15 *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

A non-zero value in this register enables the watchdog timer for CPU-to-memory access. When enabled, any CPU-to-memory access cycle will be 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 MD12 = 1, see Table 5-6 on page 23)

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 blit access.

8.2.16 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. Please refer to Section 15, "SwivelView™" on page 1-178 for application and limitations.
When this bit = 0, there is no hardware effect.
This bit in conjunction with SwivelView™ Enable Bit 1 achieves the following hardware rotations.

Table 8-35 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   Display Mode Select Bits [2:0]
These bits select the display mode according to the following table. The LCD display mode is enabled/disabled using bit 0. Programming this bit from a 0 to a 1 starts the power-on sequence. Programming this bit from a 1 to a 0 starts the power-off sequence.

Table 8-36 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.

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.

8.2.17 MediaPlug Register Descriptions

The S1D13506 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 1-187.

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 8-37 MediaPlug LCMD Read/Write Descriptions

Data	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Write	TO[2:0]		Xxxxxxxxxx										IC	MC	P	W
Read	TO[2:0]		00b		Rev[3:0]				Rstat[2:0]			0b	IC	MC	P	W

bits 15-14   Timeout Option (MediaPlug Parameter TO)
These bits select the timeout delay in MediaPlug clock cycles:

Table 8-38 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 will always return 00b.
A write to these bits has no hardware effect.

bits 11-8   MediaPlug IC Revision (MediaPlug Parameter Rev)
The revision for this MediaPlug IC is "0011b".
A write to these bits has no hardware effect.

bit 7   Cable Detected Status (MediaPlug Parameter Rstat)
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 will always return 0b.
A write to this bit has no hardware effect.

bit 5   Remote Powered Status (MediaPlug Parameter Rstat)
               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 8-39 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 will always return 0b.
               A write to this bit has no hardware effect.

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

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

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

bit 0   Watchdog Disable (MediaPlug Parameter W)
               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 8-40 MediaPlug CMD Read/Write Descriptions

Data	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Write	I[12:0]													C[2:0]		
Read	D	T	I[10:0]											C[2:0]		

bit 15   Dirty Bit (MediaPlug Parameter D)
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 (MediaPlug Parameter T)
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 (MediaPlug Parameter I)
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 (MediaPlug Parameter C)
Selects the command as follows:

Table 8-41 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.

EPSON   S1D13506F00A HARDWARE FUNCTIONAL
                                         SPECIFICATION (X25B-A-001-08)

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.

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.

8.2.18 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

The S1D13506 has a built-in 2D BitBlt engine which increases the performance of Bit Block Transfers (BitBlt). This section will discuss the BitBlt engine design and functionality.

9.1 Functional Description

The 2D BitBlt engine is designed using a 16-bit architecture. It implements a 16-bit data bus and supports both 8 and 16 bit-per-pixel color depths. The design does not support VGA planar mode.

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

Note: For details on the operation of the BitBlt registers, see Section 8.2.12, "BitBlt Configuration Registers" on page 1-142.

Write Blit

The Write Blit provides 16, two operand, ROP functions.

Move Blit

The Move Blit provides 16, two operand, ROP functions and is supported in both a positive and negative direction.

Read Blit

The Read Blit supports bit block transfers from the display buffer to the host. No ROP function is applied.

Solid Fill

The Solid Fill Blit fills a specified blit area with a solid color as defined in the Foreground Color Register. In 8 bpp mode, only the low byte of the Foreground Color is used for solid fill.

Pattern Fill

The Pattern Fill Blit fills a specified blit area with an 8 pixel by 8 line pattern in full color defined in off-screen display buffer. The pattern data has to be stored in a contiguous address (i.e. for 8 and 16 bpp, the pattern data will occupy 64 and 128 bytes respectively starting from the base address).

Any pixel within the 8x8 pattern can be used to start the fill area. The least significant bits of the source address start register are used to specify the starting pixel.

The 2D engine can detects the end of each line and continues from the beginning of the next line. When the last line of pattern is encountered, the first line of the pattern will be drawn on the following line.

Supports two full 16-bit operand ROP functions.

Note: The BitBlt operation Pattern Fill with ROP requires a BitBlt width ≥ 2. The BitBlt width is set in REG[110h], REG[111h].

Transparent Pattern Fill

The Transparent Pattern Fill fills a specified blit area with an 8 pixel by 8 line pattern in full color defined in off-screen display buffer. The pattern data has to be stored in a contiguous address (i.e. for 8 and 16 bpp, the pattern data will occupy 64 and 128 bytes respectively starting from the base address).

When the pattern color is equal to the key color, which is defined in Background Color Register, the destination area is not updated. In 8 bpp mode, only the low byte of the key color is used for comparison.

For this blit no raster operation is applied.

Note: The BitBlt operation Pattern Fill with transparency requires a BitBlt width ≥ 2. The BitBlt width is set in REG[110h], REG[111h].

Transparent Write Blit

The Transparent Write Blit supports bit block transfers from the host to display buffer.

When the source color is equal to key color, which is defined in Background Color Register, the destination area is not updated. In 8 bpp mode, only the low byte of the key color is used for comparison.

For this blit no raster operation is applied.

Transparent Move Blit

The Transparent Move Blit supports bit block transfers from display buffer to display buffer in positive direction only.

When the source color is equal to key color, which is defined in Background Color Register, the destination area is not updated. In 8 bpp mode, only the low byte of key color is used for comparison.

For this blit no raster operation is applied.

Color Expansion

The Color Expansion Blit expands the host's monochrome data to 8 or 16 bpp color format.

A 1 expands to the color defined in the Foreground Color Register. In 8 bpp mode, only the low byte of the Foreground Color Register is used.

A 0 expands to the color defined in the Background Color Register. In 8 bpp mode, only the low byte of the Background Color Register is used. If background transparency is enabled, then the destination color will remain untouched.

The host will be continuously feeding a 16-bit data package. When the end of the line is reached, any unused bits will be discarded. The data for the next line will be taken from the next data package. The low byte write data will be used first for the operation. Each bit is serially expanded to the destination data starting from MSB (Bit 7) to LSB (Bit 0).

This blit supports any bit alignment, but supports in a positive direction only.

Move Blit with Color Expansion

The Move Blit with Color Expansion expands off-screen source's monochrome data to 8 or 16 bpp color format.

A 1 expands to the color defined in the Foreground Color Register.

A 0 expands to the color defined in the Background Color Register. If background transparency is enabled, then the destination color will remain untouched.

In 8 bpp mode, only the low byte of the Foreground Color Register and Background Color Register are used for color expansion. The low byte write data will be used first for the operation. Each bit is serially expanded to the destination data starting from MSB (Bit 7) to LSB (Bit 0).

This blit supports byte alignment only and supports in a positive direction only.

10 DISPLAY BUFFER

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

Table 10-1 S1D13506 Addressing

CS#	M/R#	Access
0	0	Register access - see Section 8.1, "Register Mapping" on page 1-106. • 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 2M byte display buffer is addressed by AB[20:0]
1	X	S1D13506 not selected

The display buffer address space is always 2M bytes. However, the physical display buffer may be either 512K bytes or 2M bytes – see Section 5.3, "Summary of Configuration Options" on page 1-23.

A 512K byte display buffer is replicated in the 2M byte address space – see Figure 10-1 "Display Buffer Addressing," on page 1-161.

The display buffer can contain an image buffer, one or more Ink Layer/Hardware Cursor buffers, and a Dual Panel Buffer.
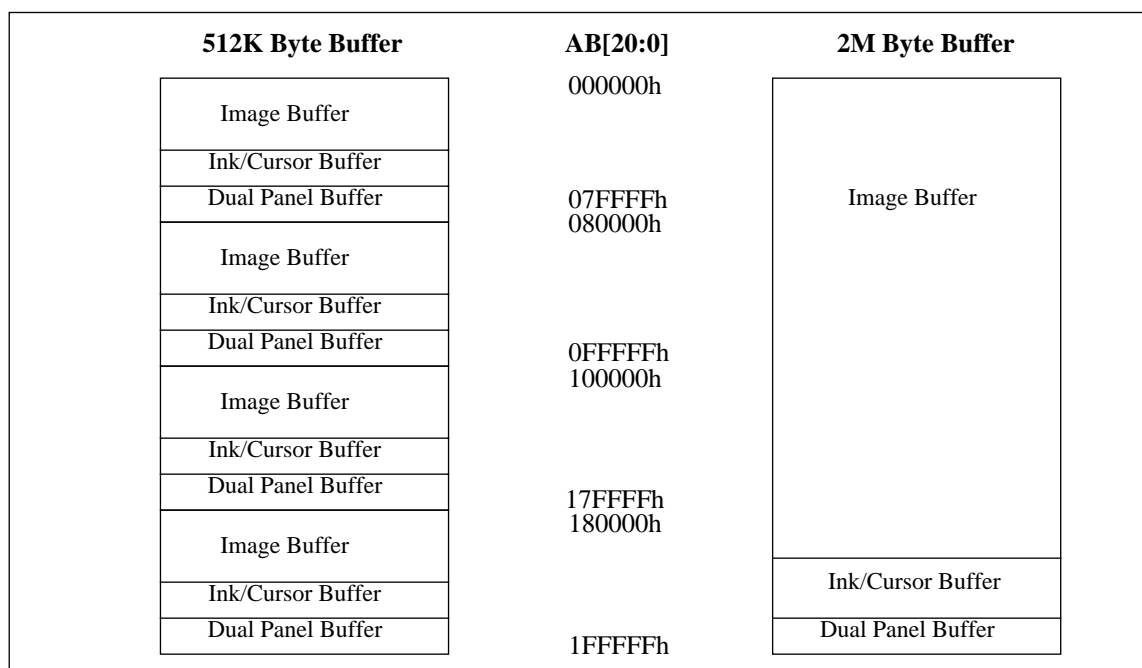
Figure 10-1 Display Buffer Addressing

10.1 Image Buffer

The image buffer contains the formatted display mode data – see Section 11.1, "Display Mode Data Format" on page 1-163.

The displayed image(s) may occupy only a portion of this space; the remaining area may be used for multiple images – possibly for animation or general storage. Section 11, "Display Configuration" on page 1-163 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 details see Section 14, "Ink Layer/Hardware Cursor Architecture" on page 1-174.

10.3 Dual Panel Buffer

In dual panel mode a buffer is required and allocated by hardware. With this 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 whether the panel is color or monochrome:

Dual Panel Buffer Size (in bytes) = (panel width x panel height) x 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 640×480 8 bpp color panel the Dual Panel Buffer size is 75K bytes. In a 512K byte display buffer, the Dual Panel Buffer resides from 6D400h to 7FFFFh. In a 2M byte display buffer, the Dual Panel Buffer resides from 1ED400h to 1FFFFFh.*

11 DISPLAY CONFIGURATION

11.1 Display Mode Data Format

The following diagrams show the display mode data formats for a little endian system:
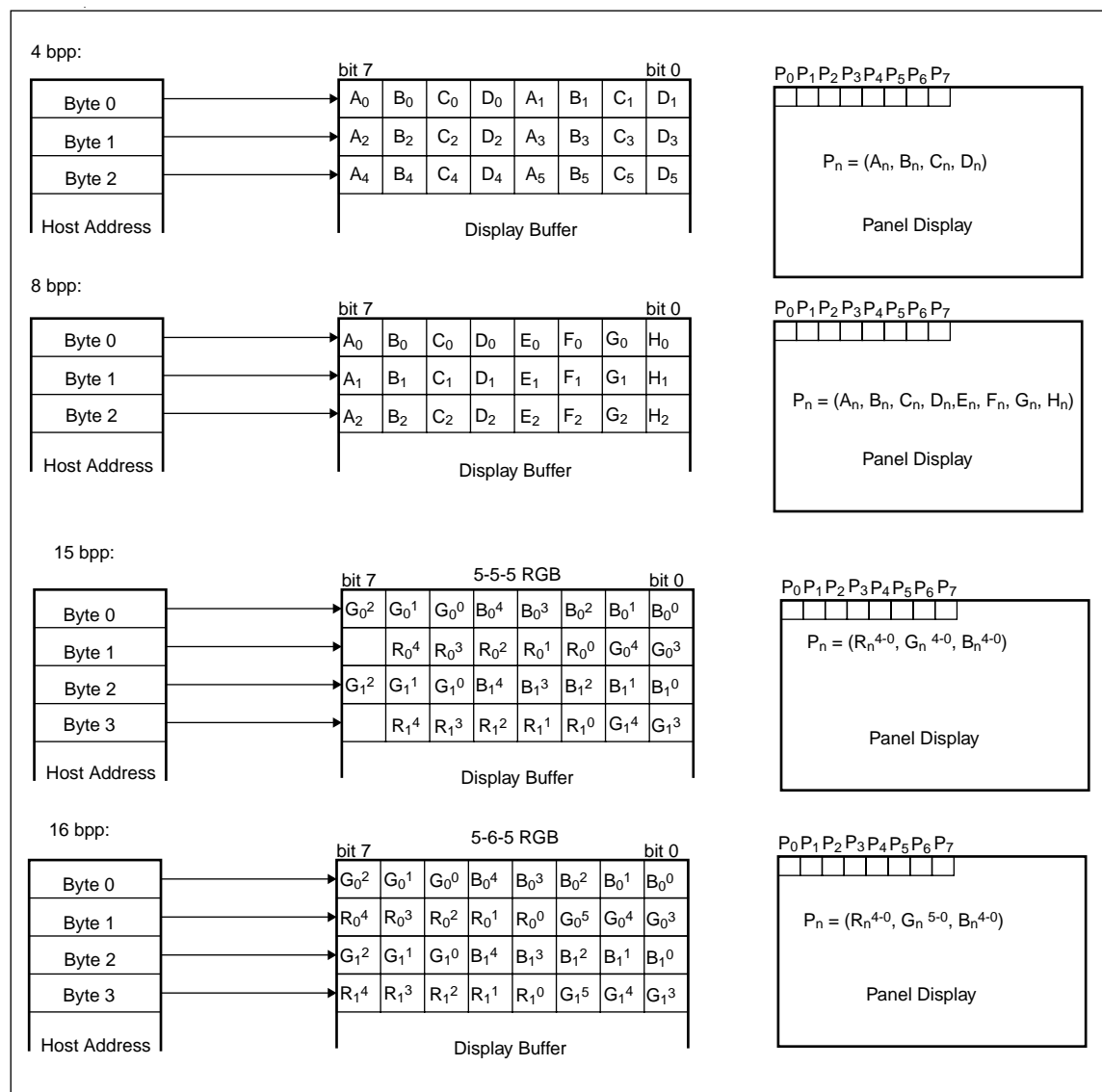
Figure 11-1 4/8/15/16 Bit-per-pixel Format Memory Organization

Note: 1. The Host-to-Display mapping shown here is for a little endian system.
2. For 15/16 bit-per-pixel formats, 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 1-185), the images on the LCD and on the CRT/TV are independent of each other.

- For LCD: REG[047h], REG[046h] defines the width of the virtual image.
  For CRT/TV: REG[067h], REG[066h] defines the width of the virtual image.

- For LCD: REG[044h], REG[043h], REG[042h] defines the starting word of the displayed image.
  For CRT/TV: REG[064h], REG[063h], REG[062h] defines 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] defines the height of the LCD display.
  For CRT/TV: REG[057h], REG[056h] defines the height of the CRT/TV display.
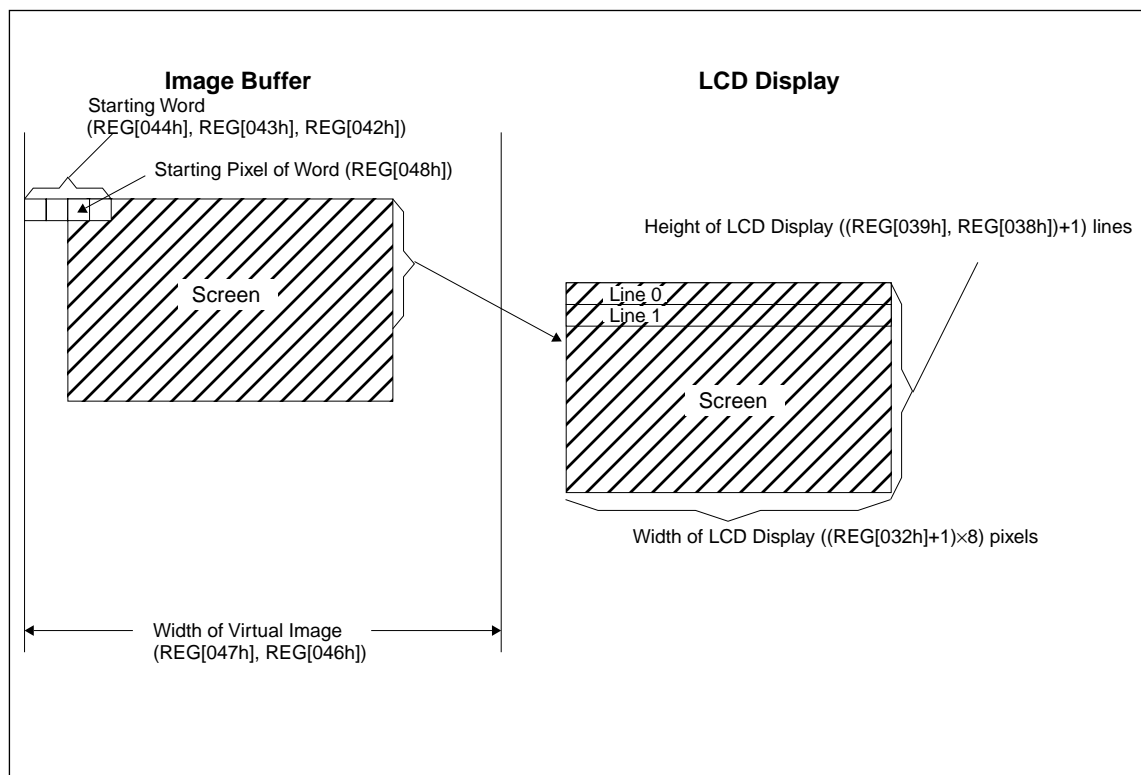
Figure 11-2 Image Manipulation

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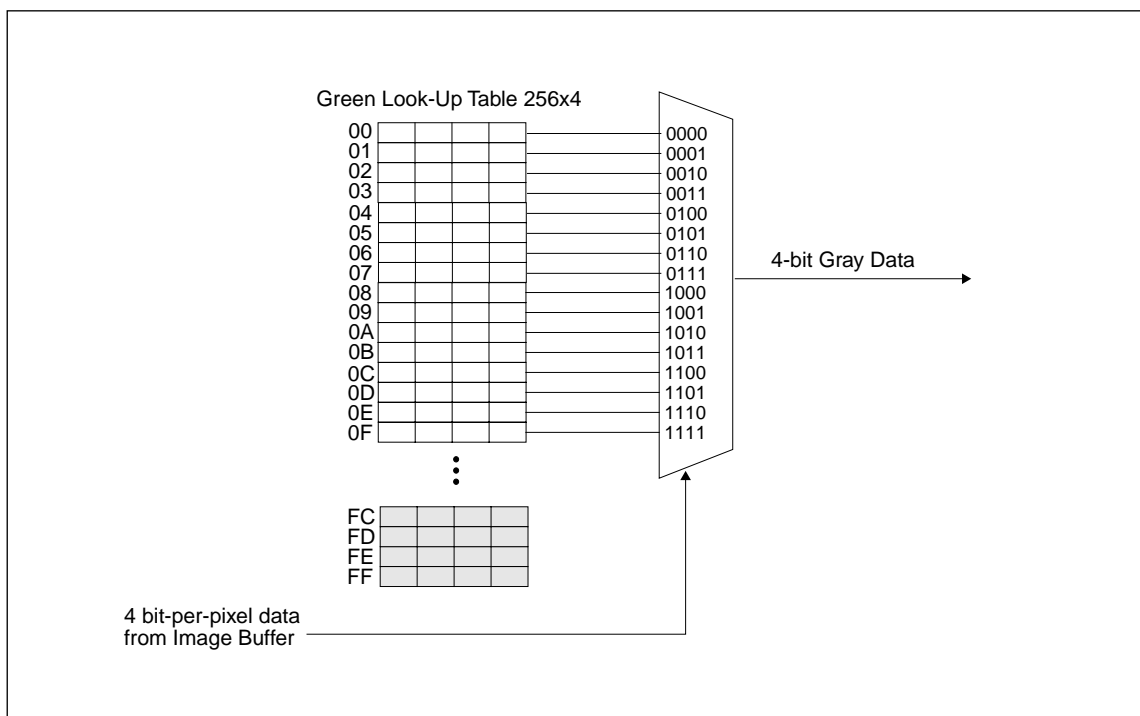
Figure 12-1 4 Bit-Per-Pixel Monochrome Mode Data Output Path

12.2 Color Modes

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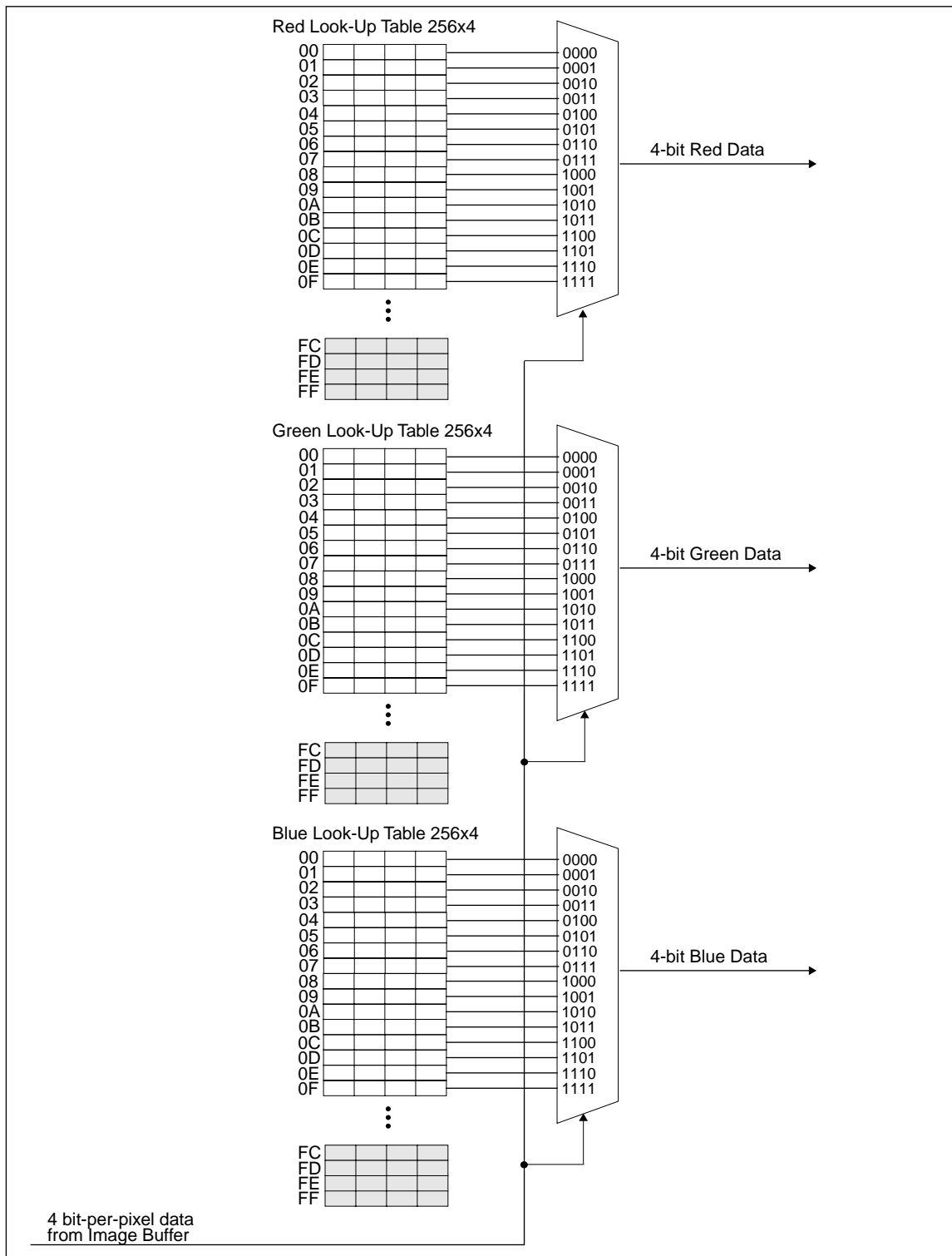
Figure 12-2 4 Bit-Per-Pixel Color Mode Data Output Path

EPSON

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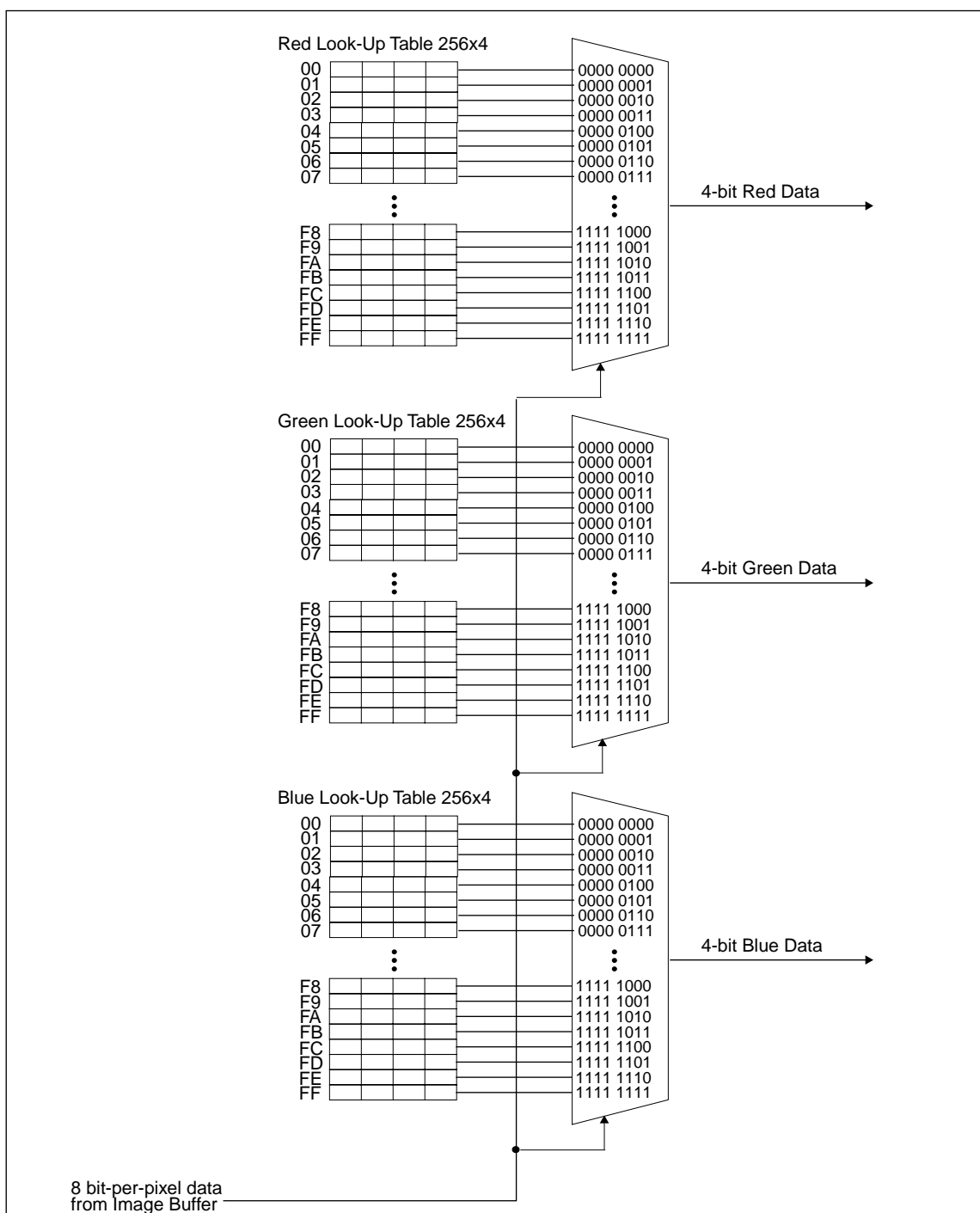
Figure 12-3 8 Bit-Per-Pixel Color Mode Data Output Path

15/16 Bit-Per-Pixel Color Modes

The LUT is bypassed and the color data is directly mapped for these color depths– See *"11 Display Configuration"* on page 1-163.

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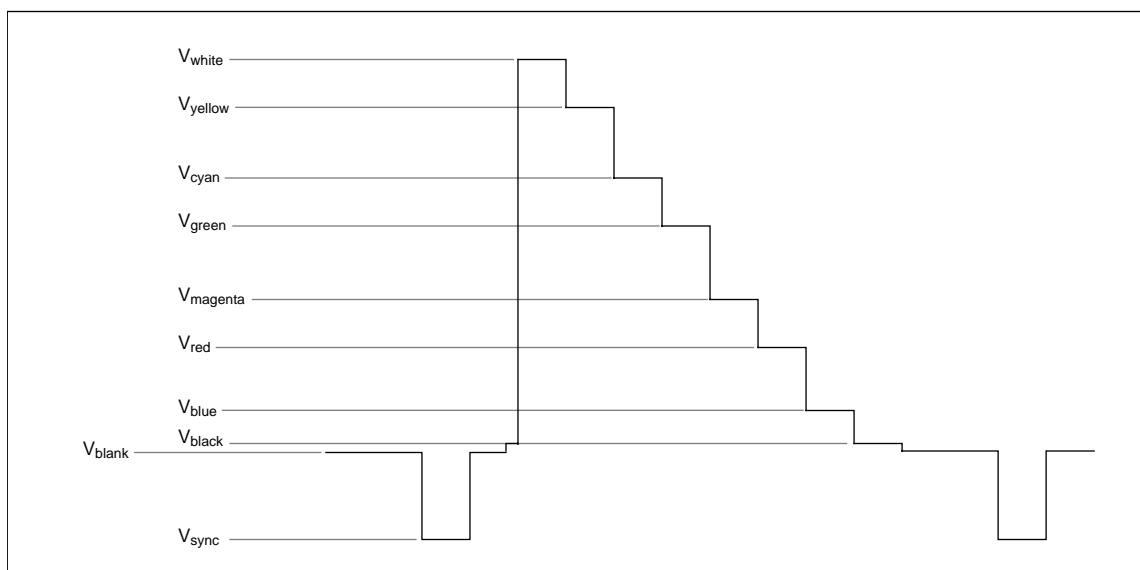
Figure 13-1 NTSC/PAL SVideo-Y (Luminance) Output Levels

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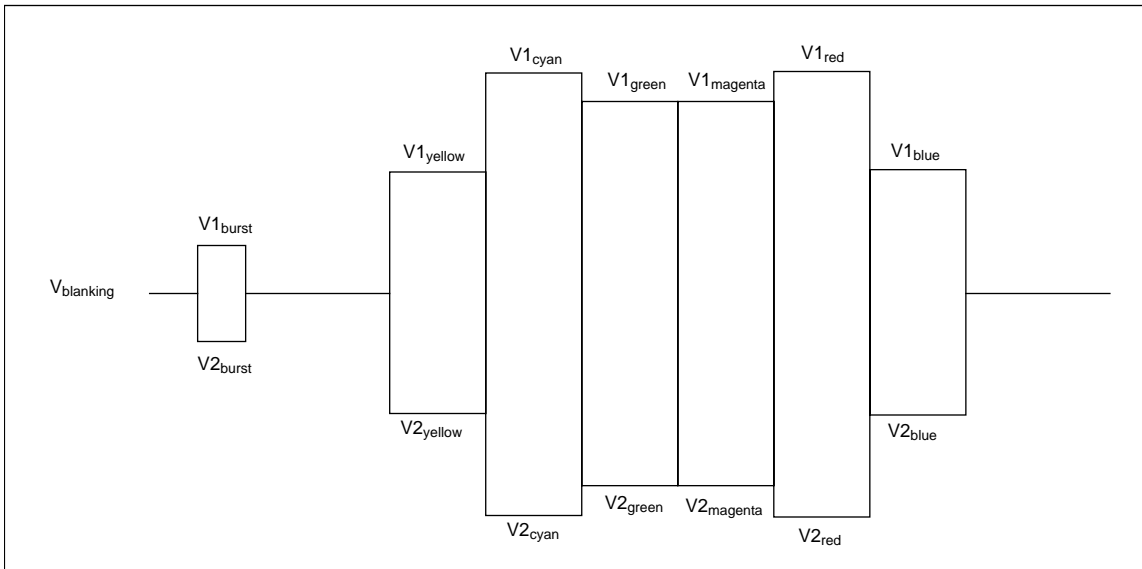
Figure 13-2 NTSC/PAL SVideo-C (Chrominance) Output Levels

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	
$V1_{yellow}$	Yellow positive peak	1F 3F 00	700	
$V1_{cyan}$	Cyan positive peak	00 3F 1F	815	
$V1_{green}$	Green positive peak	00 3F 00	751	
$V1_{magenta}$	Magenta positive peak	1F 00 1F	751	
$V1_{red}$	Red positive peak	1F 00 00	815	
$V1_{blue}$	Blue positive peak	00 00 1F	700	
$V_{blanking}$	Blanking	N.A.	410	
$V2_{burst}$	Burst negative peak	N.A.	268 / 279	
$V2_{yellow}$	Yellow negative peak	1F 3F 00	121	
$V2_{cyan}$	Cyan negative peak	00 3F 1F	5	
$V2_{green}$	Green negative peak	00 3F 00	70	
$V2_{magenta}$	Magenta negative peak	1F 00 1F	70	
$V2_{red}$	Red negative peak	1F 00 00	5	
$V2_{blue}$	Blue negative peak	00 00 1F	121	

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

EPSON

S1D13506F00A HARDWARE FUNCTIONAL
SPECIFICATION (X25B-A-001-08)
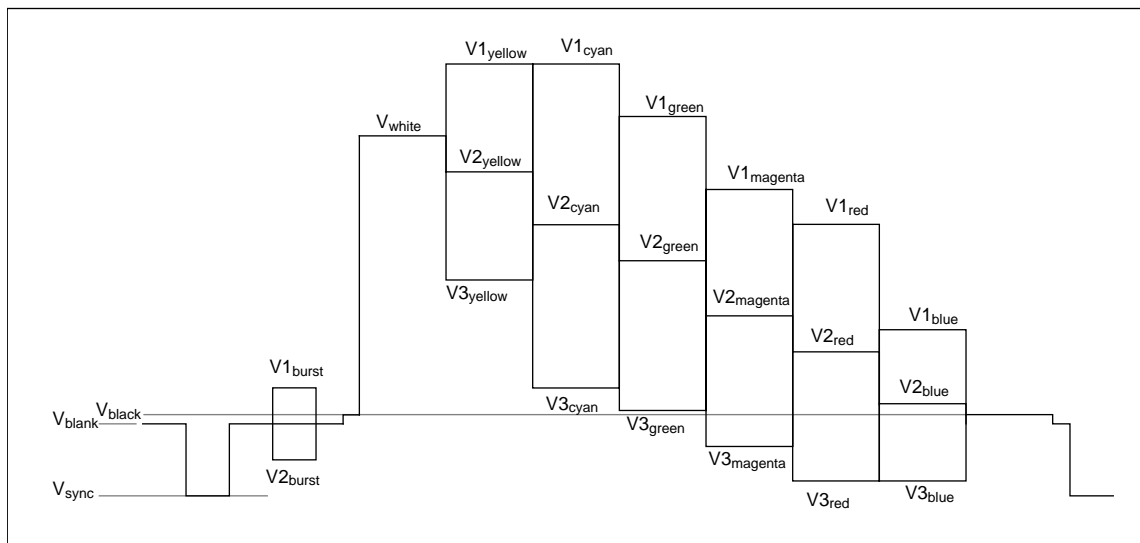
Figure 13-3 NTSC/PAL Composite Output Levels

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	1213	130
$V1_{cyan}$	Cyan chrominance positive peak	00 3F 1F	1203	128.5
$V1_{green}$	Green chrominance positive peak	00 3F 00	1066	109
$V1_{magenta}$	Magenta chrominance positive peak	1F 00 1F	949	93
$V1_{red}$	Red chrominance positive peak	1F 00 00	877	83
$V1_{blue}$	Blue chrominance positive peak	00 00 1F	700	58
V_{white}	White luminance level	1F 3F 1F	996	99.5
$V2_{yellow}$	Yellow luminance level	1F 3F 00	923	89
$V2_{cyan}$	Cyan luminance level	00 3F 1F	798	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	536	35
$V2_{blue}$	Blue luminance level	00 00 1F	410	17
V_{black}	Black luminance level	00 00 00	338	7.3
$V3_{yellow}$	Yellow chrominance negative peak	1F 3F 00	633	48.7
$V3_{cyan}$	Cyan chrominance negative peak	00 3F 1F	393	15
$V3_{green}$	Green chrominance negative peak	00 3F 00	384	13.8
$V3_{magenta}$	Magenta chrominance negative peak	1F 00 1F	267	-2.6
$V3_{red}$	Red chrominance negative peak	1F 00 00	195	-12.7
$V3_{blue}$	Blue chrominance negative peak	00 00 1F	121	-23
V_{blank}	Blank Level	N.A.	284	0
$V1_{burst}$	Burst positive peak	N.A.	426 / 415	19.7 / 18.1
$V2_{burst}$	Burst negative peak	N.A.	142 / 153	-20.1 / -18.6
V_{sync}	Sync Tip	N.A.	0	-40

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

13.5 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," on page 1-172 shows an image positioned on 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 Min. and Max. values for these registers are given in Table 13-5, "Min. and Max. Values for NTSC/PAL". The line period and frame period determined by these registers must also satisfy the following equations:

NTSC:

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

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

PAL:

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

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

The HRTC Start Position and VRTC Start Position registers position the image horizontally and vertically. The Min. and Max. register values for these registers are given in Table 13-5, "Min. and Max. Values for NTSC/PAL". Increasing the HRTC Start Position will move the image left, while increasing the VRTC Start Position will move the image up.
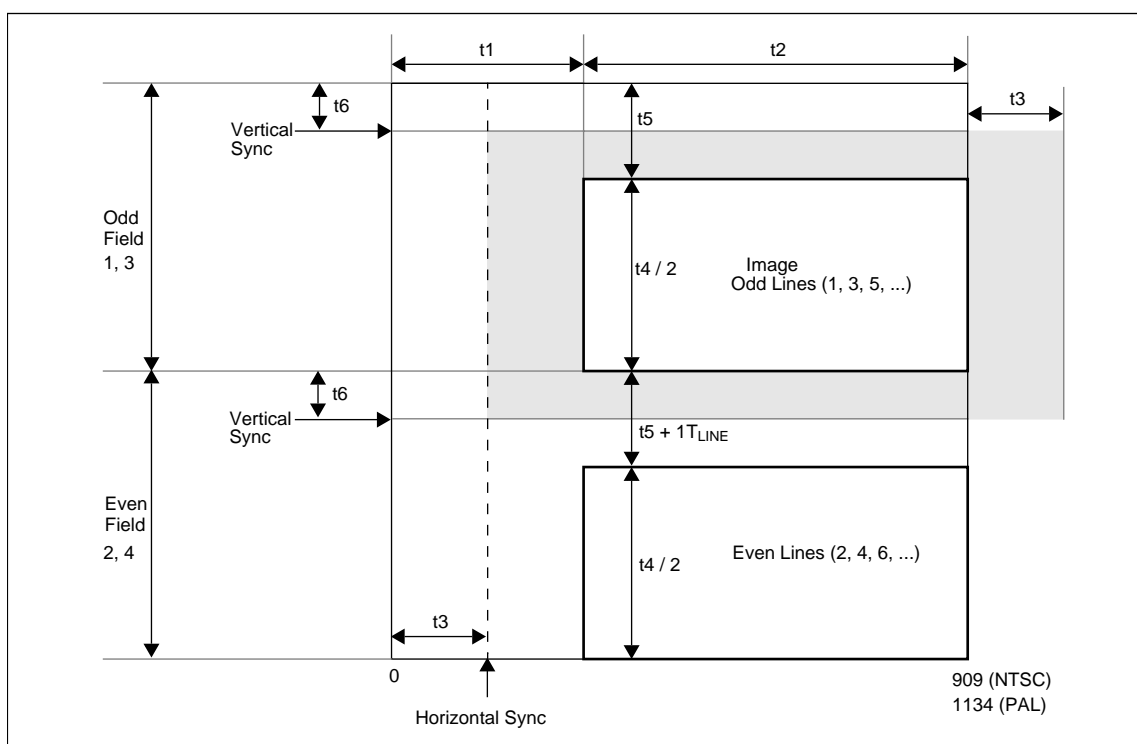
Figure 13-4 NTSC/PAL Image Positioning

The maximum Horizontal and Vertical Display Widths shown in Table 13-5, "Min. and Max. Values for NTSC/PAL" include display areas that are normally hidden by the edges of the TV. The visible display dimensions are shown in Figure 13-5 "Typical Total Display and Visible Display Dimensions for NTSC and PAL," on page 1-173 as a guideline. The actual visible display area for a particular TV may differ slightly from those dimensions given. Table 13-6, "Register Values for Example NTSC/PAL Images" lists register values for some example images.

Table 13-5 Min. and Max. Values for NTSC/PAL

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	396	484	496	572	T_{LINE}
t5	TV Vertical Non-Display Period	58	20	64	26	64	T_{LINE}
t6	TV VRTC Start Position	59	0	t5 - 20	0	t5 - 26	T_{LINE}
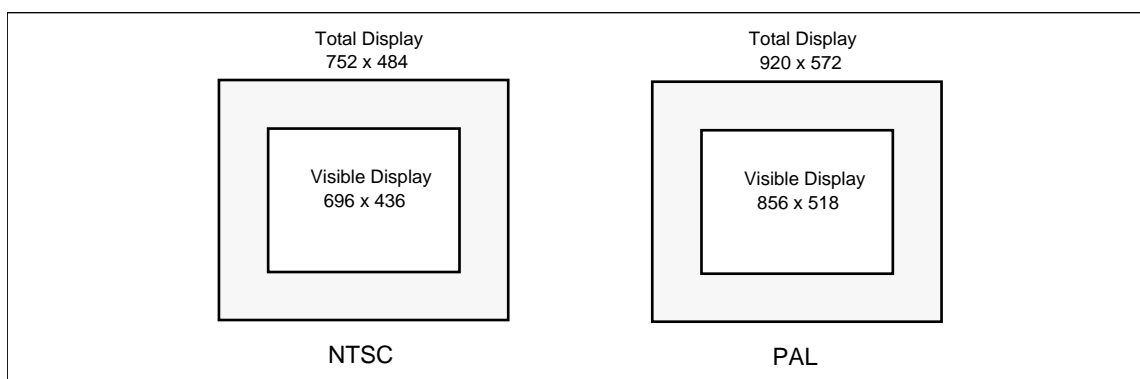
Figure 13-5 Typical Total Display and Visible Display Dimensions for NTSC and PAL

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			
		752x484	696x436	640x480	920x572	856x518	800x572	640x480
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	08h	02h	05h	09h	13h
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 VRTC Start Position	59	00h	0Ch	01h	00h	0Dh	00h	16h

13.6 TV Cursor Operation

See Section 14, "Ink Layer/Hardware Cursor Architecture" on page 1-174.

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	Display Buffer Size - 1024	This default value is suitable for a cursor when there is no Dual Panel Buffer.
n = 255...1	Display Buffer Size - (n $\times$ 8192)	These positions can be used to: • position an Ink buffer at the top of the display buffer; • position an Ink buffer between the image and Dual Panel Buffers; • position a Cursor buffer between the image and Dual Panel Buffers; • select from a multiple of Cursor buffers.

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.
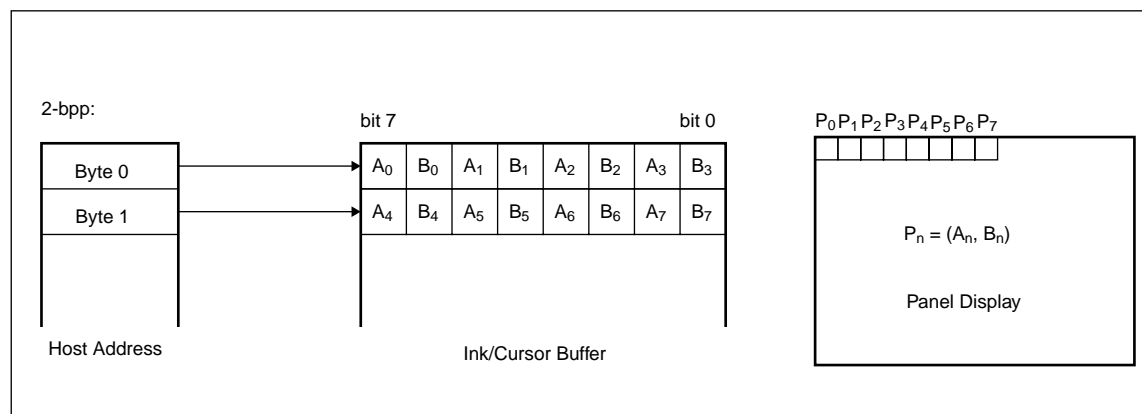
Figure 14-1 Ink/Cursor Data Format

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.
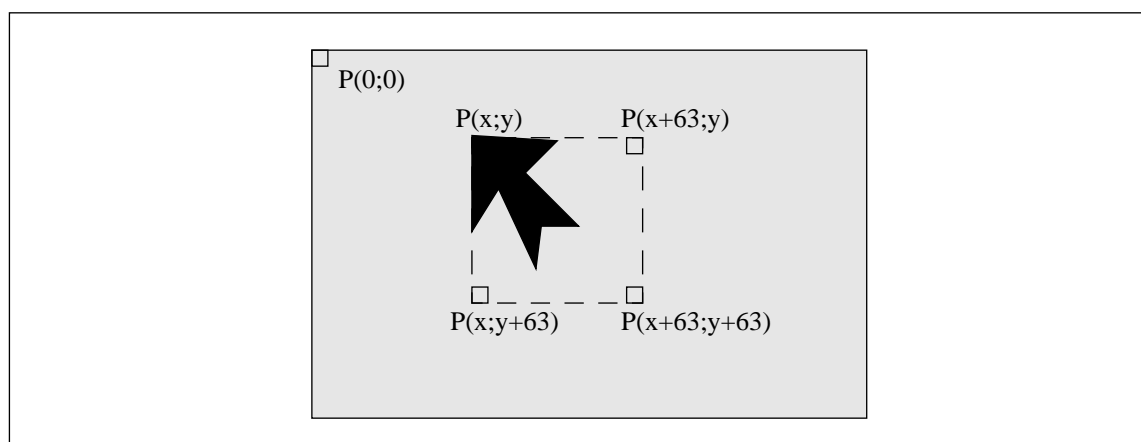
Figure 14-2 Unclipped Cursor Positioning

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.
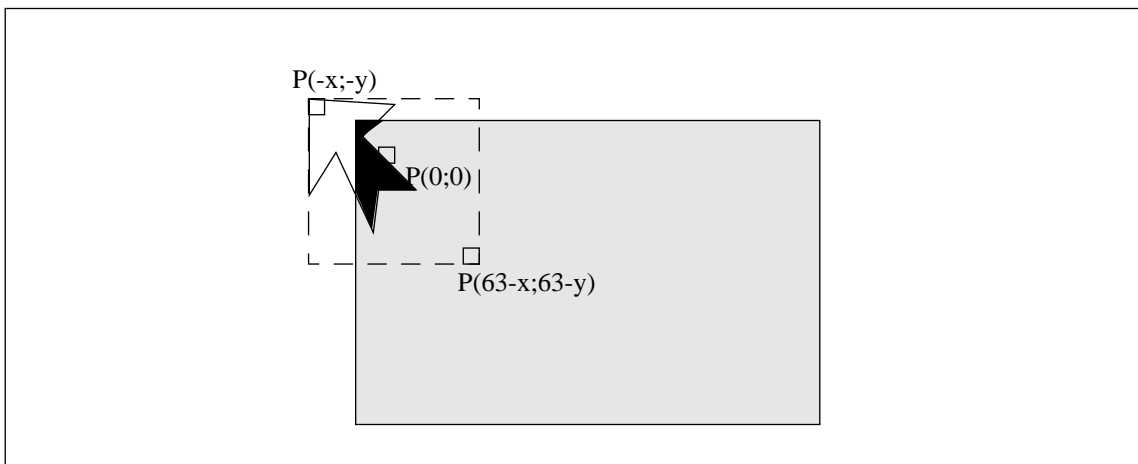
Figure 14-3 Clipped Cursor Positioning

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 S1D13506 in the following sense: A–B–C–D. The S1D13506 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 1-179).
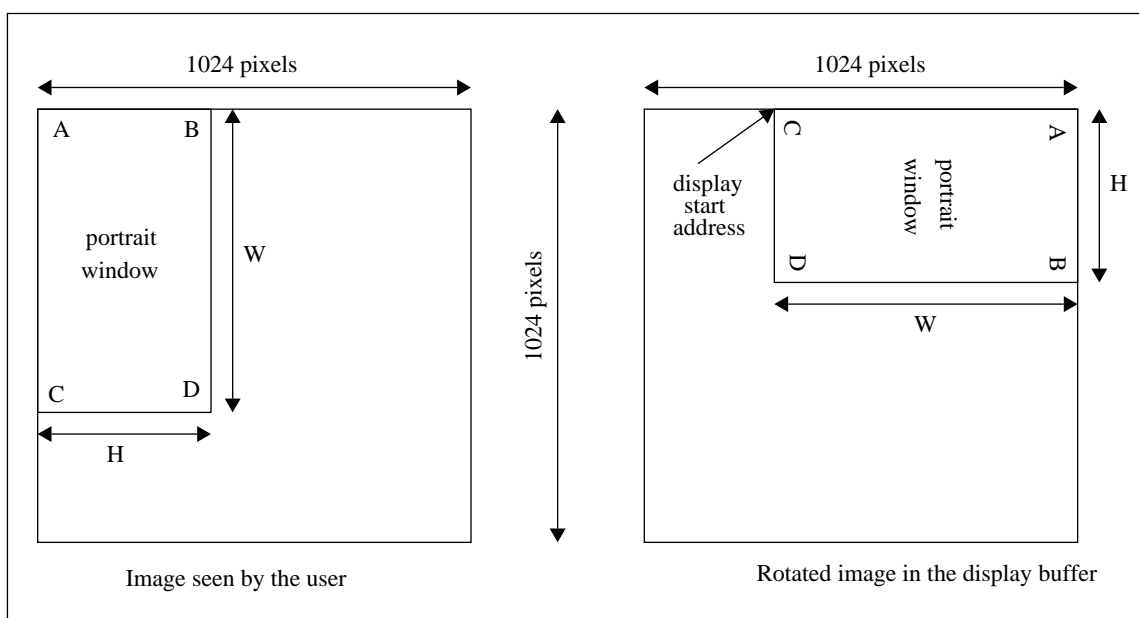
Figure 15-1 Relationship Between Screen Image and 90° Rotated Image in the Display Buffer

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 mode or 2048 bytes for 15/16 bpp mode) 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 bit 0 to 1. All CPU accesses to the display buffer are translated to provide 90° clockwise rotation of the display image. SwivelView™ Enable bit 1 should be set to 0.

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 15/16 bpp mode
        = 512   for 8 bpp mode

Display Start Address

As seen in Figure 15-1 "Relationship Between Screen Image and 90° Rotated Image in the Display Buffer," on page 1-178, 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 15/16 bpp mode
        = (1024 - W) / 2   for 8 bpp mode

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 (15/16 bpp mode) or 512 (8 bpp mode) 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 mode scrolls the display window up/down by 2 lines.

- Increment/decrement LCD Display Start Address register in 15/16 bpp mode scrolls the display window up/down by 1 line.

- Increment/decrement LCD Pixel Panning register in 8 bpp mode 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)

$= (1024 \times H) \times 2$ for 15/16 bpp mode
$= (1024 \times H)$ for 8 bpp mode

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. Its memory requirement is:

Dual Panel Buffer (bytes)

$=(W \times H) / 4$ for color mode
$=(W \times H) / 16$ 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.

The following table summarizes the DRAM size requirement for 90° SwivelView™ for different panel sizes and display modes. Note that DRAM size for the S1D13506 is limited to either 512K byte or 2M byte. The calculation is based on the minimum required image buffer size and the Dual Panel Buffer size. The Hardware Cursor/Ink Layer may or may not fit within this minimum DRAM configuration – this is noted in the table. The hardware cursor requires only 1KB of memory and so may resides at the last 1KB area if there is no Dual Panel Buffer, otherwise it must reside at an 8KB boundary. The 2-bit ink layer requires $(W \times H) / 4$ bytes of memory; it must reside at an 8KB boundary. The table shows only one possible Hardware Cursor/Ink Layer location – at the highest possible location without interfering with the Dual Panel Buffer. It is also assumed that CRT/TV is not used.

Table 15-1 Minimum DRAM Size Required for SwivelView™

Panel Size	Panel Type		Display Mode	Min. Image Buffer Size	Dual Panel Buffer Size	Minimum DRAM Size	Ink/Cursor Buffer Size	Ink/Cursor Location
320 × 240	Single	Color	8 bpp	240KB	0KB	512KB	18.75KB/1KB	488KB/511KB
			15/16 bpp	480KB				
		Mono	8 bpp	240KB				
			15/16 bpp	480KB				
640 × 480	Single	Color	8 bpp	480KB	0KB		75KB/1KB	–/511KB
			15/16 bpp	960KB		2MB		1968KB/2047KB
		Mono	8 bpp	480KB		512KB		–/511KB
			15/16 bpp	960KB				1968KB/2047KB
	Dual	Color	8 bpp	480KB	75KB	2MB		1896KB/1968KB
			15/16 bpp	960KB				
		Mono	8 bpp	480KB	18.75KB	512KB		–/488KB
			15/16 bpp	960KB				1952KB/2024KB
800 × 600	Single	Color	8 bpp	600KB	0KB	2MB	117.19KB/1KB	1928KB/2047KB
			15/16 bpp	1.2MB				
		Mono	8 bpp	600KB				
			15/16 bpp	1.2MB				
	Dual	Color	8 bpp	600KB	117.19KB			1808KB/1928KB
			15/16 bpp	1.2MB				
		Mono	8 bpp	600KB	29.30KB			1896KB/2016KB
			15/16 bpp	1.2MB				

Where KB = 1024 bytes and MB = 1024KB

15.2.3 *Limitations*

The following limitations apply to 90° SwivelView™:

- Only 8/15/16 bpp modes are supported – 4 bpp mode 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.

- 90° SwivelView™ does not support BitBlts.

15.3 180° SwivelView™

180° SwivelView™ is accomplished by fetching the display buffer image in the reverse address direction, starting at the bottom-right corner of the image. Unlike 90° SwivelView™, the 180° SwivelView™ image is not rotated in the display buffer. The image is simply **displayed** 180° clockwise rotated. Furthermore, a virtual window is not required and all color depths (4/8/15/16 bpp) are supported.

15.3.1 Register Programming

Reverse Display Buffer Fetching Address Direction

Set SwivelView™ Enable bit 1 to 1. 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. SwivelView™ Enable bit 0 should be set to 0.

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)

$= (\text{MA\_Offset} \times \text{H}) - (\text{MA\_Offset} - \text{W}) - 1$ for 15/16 bpp mode
$= (\text{MA\_Offset} \times \text{H}) - (\text{MA\_Offset} - \text{W}/2) - 1$ for 8 bpp mode
$= (\text{MA\_Offset} \times \text{H}) - (\text{MA\_Offset} - \text{W}/4) - 1$ for 4 bpp mode

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.
- 180° SwivelView™ does not support all BitBlts.

15.4 270° SwivelView™

270° SwivelView™ is a combination of 90° SwivelView™ and 180° SwivelView™. The image stored in the display buffer is 90° rotated, and the image is further 180° rotated during screen refresh, resulting in a 270° rotated display image. The user must use a 1024 × 1024 pixel virtual window as in 90° SwivelView™. See Figure 15-1 "Relationship Between Screen Image and 90° Rotated Image in the Display Buffer," on page 1-178.

15.4.1 Register Programming

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

Set SwivelView™ Enable bit 0 to 1. All CPU access to the display buffer is translated to provide 90° clockwise rotation of the display image.

Reverse Display Buffer Fetching Address Direction

Set SwivelView™ Enable bit 1 to 1. 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.

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 15/16 bpp mode
= 512		for 8 bpp mode

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 (15/16 bpp mode) or 512 (8 bpp mode) 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 mode scrolls the display window up/down by 2 lines.

- Increment/decrement LCD Display Start Address register in 15/16 bpp mode scrolls the display window up/down by 1 line.

- Increment/decrement LCD Pixel Panning register in 8 bpp mode 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 8/15/16 bpp modes are supported – 4 bpp mode 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.

- 270° SwivelView™ does not support BitBlts.

16 EPSON INDEPENDENT SIMULTANEOUS DISPLAY (EISD)

16.1 Introduction

EPSON Independent Simultaneous Display (EISD) allows the S1D13506 to display independent images on two different displays (LCD panel and CRT or TV). 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 also 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.

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.

Example 1: *Assuming a 2M byte display buffer, the LCD image may locate in the first 1M byte of the display buffer (AB[20:0] = 000000h-0FFFFFh), and the CRT/TV image may locate in the second 1M byte of the display buffer (AB[20:0] = 100000h-1FFFFFh).*

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. However, the display pipelines are still independent so the same image is fetched twice from the display buffer; once for the LCD refresh and once for the CRT/TV refresh.

16.2 Bandwidth Limitation

When EISD is enabled, the LCD and CRT/TV displays must share the total bandwidth available to the S1D13506. 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).

When the FIFO High Threshold Control register is set to 00h (default), the following settings are used:

- 11h for 4 bpp mode

- 21h for 8 bpp mode

- 23h for 15/16 bpp mode

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.

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. 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 1-190.

17 MEDIAPLUG INTERFACE

Winnov's MediaPlug Slave interface has been incorporated into the S1D13506. 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 uses the upper eight pins of the LCD data bus (FPDAT[15:8]) for the data bus, clock, and control lines. When pin MD13 is high at the rising edge of RESET#, FPDAT[15:8] are dedicated to the MediaPlug interface.

Table 17-1 MediaPlug Interface Pin Mapping

S1D13506 Pin Names	IO Type	MediaPlug I/F
FPDAT8	O	VMPLCTL
FPDAT9	I	VMPRCTL
FPDAT10	IO	VMPD0
FPDAT11	IO	VMPD1
FPDAT12	IO	VMPD2
FPDAT13	IO	VMPD3
FPDAT14	O	VMPCLK
FPDAT15	O	VMPCLKN
DRDY or MA11	O	VMPEPWR

Note: If MediaPlug is enabled, any 16-bit LCD panel must use an external circuit to support FPDAT[15:8].

Either pin MA11 or pin DRDY can be configured as the MediaPlug power control output, VMPEPWR. This is selected by the states of MD14, MD7, MD6 at the rising edge of RESET# - see Table 5-6, "Summary of Power-On/Reset Options" on page 1-23.

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.

$$\text{max. LCD Frame Rate} = \frac{\text{LCD PCLK}_{max}}{(LHDP + LHNDP) \times \left(\frac{LVDP}{n} + LVNDP\right)}$$

Where:

LCD PCLKmax   = 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.

$$\text{max. CRT Frame Rate} = \frac{\text{CRT PCLK}_{max}}{(CHDP + CHNDP) \times (CVDP + CVNDP)}$$

Where:

CRT PCLKmax = 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.

$$\text{max. TV Frame Rate} = \frac{\text{TV PCLK}_{max}}{(THDP + THNDP) \times (TVDP + TVNDP + 0.5)}$$

Where:

TV PCLKmax = 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
= (REG[052h] bits [5:0] x 8Ts) + 6 for NTSC output
= (REG[052h] bits [5:0] x 8Ts) + 7 for PAL output

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 40MHz.

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	–	–	–	–	–	–	–	–	–
	No	640	480	8	40	64	1	118	–	–	–	–	–	–	–	–	–
	No	640	480	16	34	56	1	101	–	–	–	–	–	–	–	–	–
Mono Passive Dual	No	640	480	4	40	64	1	235	–	–	–	–	–	–	–	–	–
	No	640	480	8	40	72	1	233	–	–	–	–	–	–	–	–	–
	No	640	480	16	27	56	1	161	–	–	–	–	–	–	–	–	–
Color Passive Dual	No	640	480	4	40	64	1	235	–	–	–	–	–	–	–	–	–
	No	640	480	8	33	64	1	194	–	–	–	–	–	–	–	–	–
	No	640	480	16	23	48	1	138	–	–	–	–	–	–	–	–	–
Passive Single / TFT	Yes	640	480	4	40	56	1	119	–	–	–	–	–	–	–	–	–
	Yes	640	480	8	40	64	1	118	–	–	–	–	–	–	–	–	–
	Yes	640	480	16	30	48	1	90	–	–	–	–	–	–	–	–	–
Mono Passive Dual	Yes	640	480	4	39	64	1	229	–	–	–	–	–	–	–	–	–
	Yes	640	480	8	31	56	1	184	–	–	–	–	–	–	–	–	–
	Yes	640	480	16	22	48	1	132	–	–	–	–	–	–	–	–	–
Color Passive Dual	Yes	640	480	4	31	56	1	184	–	–	–	–	–	–	–	–	–
	Yes	640	480	8	26	48	1	156	–	–	–	–	–	–	–	–	–
	Yes	640	480	16	20	40	1	122	–	–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–	–	CRT	No	640	480	4	36	192	29	85
–	–	–	–	–	–	–	–	–	CRT	No	640	480	8	36	192	29	85
–	–	–	–	–	–	–	–	–	CRT	No	640	480	16	36	192	29	85
–	–	–	–	–	–	–	–	–	NTSC TV	No	640	480	4	14.32	270	22	62
–	–	–	–	–	–	–	–	–	NTSC TV	No	640	480	8	14.32	270	22	62
–	–	–	–	–	–	–	–	–	NTSC TV	No	640	480	16	14.32	270	22	62
–	–	–	–	–	–	–	–	–	PAL TV	No	640	480	4	17.73	495	72	56
–	–	–	–	–	–	–	–	–	PAL TV	No	640	480	8	17.73	495	72	56
–	–	–	–	–	–	–	–	–	PAL TV	No	640	480	16	17.73	495	72	56
–	–	–	–	–	–	–	–	–	CRT	Yes	640	480	4	36	192	29	85

▢ = 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	8	36	192	29	85
–	–	–	–	–	–	–	–	–	CRT	Yes	640	480	16	31.5	200	20	75
–	–	–	–	–	–	–	–	–	NTSC TV	Yes	640	480	4	14.32	270	22	62
–	–	–	–	–	–	–	–	–	NTSC TV	Yes	640	480	8	14.32	270	22	62
–	–	–	–	–	–	–	–	–	NTSC TV	Yes	640	480	16	14.32	270	22	62
–	–	–	–	–	–	–	–	–	PAL TV	Yes	640	480	4	17.73	495	72	56
–	–	–	–	–	–	–	–	–	PAL TV	Yes	640	480	8	17.73	495	72	56
–	–	–	–	–	–	–	–	–	PAL TV	Yes	640	480	16	17.73	495	72	56

☐ = 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)
Color Passive Dual	No	800	600	4	40	64	1	153	–	–	–	–	–	–	–	–	–
Color Passive Dual	No	800	600	8	33	64	1	126	–	–	–	–	–	–	–	–	–
Color Passive Dual	No	800	600	16	23	48	1	90	–	–	–	–	–	–	–	–	–
Color Passive Dual	Yes	800	600	4	31	56	1	120	–	–	–	–	–	–	–	–	–
Color Passive Dual	Yes	800	600	8	26	48	1	101	–	–	–	–	–	–	–	–	–
Color Passive Dual	Yes	800	600	16	20	40	1	79	–	–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–	–	CRT	No	800	600	4	40	256	28	60
–	–	–	–	–	–	–	–	–	CRT	No	800	600	8	40	256	28	60
–	–	–	–	–	–	–	–	–	CRT	No	800	600	16	36	224	25	56
–	–	–	–	–	–	–	–	–	CRT	Yes	800	600	4	40	256	28	60
–	–	–	–	–	–	–	–	–	CRT	Yes	800	600	8	40	256	28	60
–	–	–	–	–	–	–	–	–	CRT	Yes	800	600	16	31.5	224	25	49

☐ = Example Frame Rates with Ink Layer Enabled

18.2.3 Frame Rates for LCD and CRT (640×480) with EISD Enabled

Table 18-3 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)	VND P (lines)	Frame Rate (MHz)
Passive Single / TFT	No	320	240	16	9.7	40	1	111	CRT	No	640	480	16	25.175	160	44	60
	No	640	240	16	9.7	40	1	59	CRT	No	640	480	16	25.175	160	44	60
	No	640	480	4	40	112	1	110	CRT	No	640	480	4	25.175	160	44	60
	No	640	480	8	27	96	1	76	CRT	No	640	480	8	25.175	160	44	60
Color Passive Dual	No	640	480	8	18	72	1	104	CRT	No	640	480	8	25.175	160	44	60
TFT	No	800	600	8	27	96	1	50	CRT	No	640	480	8	25.175	160	44	60
Color Passive Dual	No	800	600	4	27	80	1	101	CRT	No	640	480	4	25.175	160	44	60
	No	800	600	8	18	72	1	68	CRT	No	640	480	8	25.175	160	44	60
Passive Single / TFT	Yes	640	480	4	32	88	1	91	CRT	No	640	480	4	25.175	160	44	60
	Yes	640	480	8	20	72	1	58	CRT	No	640	480	8	25.175	160	44	60
Mono Passive Dual	Yes	640	480	4	25	80	1	144	CRT	No	640	480	4	25.175	160	44	60
	Yes	640	480	8	17	64	1	100	CRT	No	640	480	8	25.175	160	44	60
Color Passive Dual	Yes	640	480	4	22	64	1	129	CRT	No	640	480	4	25.175	160	44	60
	Yes	640	480	8	15	56	1	89	CRT	No	640	480	8	25.175	160	44	60
	Yes	800	600	4	22	64	1	84	CRT	No	640	480	4	25.175	160	44	60
Passive Single / TFT	No	640	240	8	20	72	1	116	CRT	Yes	640	480	8	25.175	160	44	60
	No	640	480	4	32	88	1	91	CRT	Yes	640	480	4	25.175	160	44	60
	No	640	480	8	20	72	1	58	CRT	Yes	640	480	8	25.175	160	44	60
Mono Passive Dual	No	640	480	4	24	72	1	139	CRT	Yes	640	480	4	25.175	160	44	60
	No	640	480	8	16	64	1	94	CRT	Yes	640	480	8	25.175	160	44	60
Color Passive Dual	No	640	480	4	21	64	1	123	CRT	Yes	640	480	4	25.175	160	44	60
	No	640	480	8	14	56	1	83	CRT	Yes	640	480	8	25.175	160	44	60
	No	800	600	4	21	64	1	80	CRT	Yes	640	480	4	25.175	160	44	60
Passive Single / TFT	Yes	640	240	8	16	56	1	95	CRT	Yes	640	480	8	25.175	160	44	60
	Yes	640	480	4	24	64	1	70	CRT	Yes	640	480	4	25.175	160	44	60
Mono Passive Dual	Yes	640	480	4	20	64	1	117	CRT	Yes	640	480	4	25.175	160	44	60
	Yes	640	480	8	13	56	1	77	CRT	Yes	640	480	8	25.175	160	44	60
Color Passive Dual	Yes	640	480	4	17	56	1	101	CRT	Yes	640	480	4	25.175	160	44	60
	Yes	640	480	8	12	48	1	72	CRT	Yes	640	480	8	25.175	160	44	60
	Yes	800	600	4	17	56	1	66	CRT	Yes	640	480	4	25.175	160	44	60

[ ] = Example Frame Rates with Ink Layer Enabled

18.2.4 Frame Rates for LCD and CRT (800×600) with EISD Enabled

Table 18-4 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 / TFT	No	640	240	8	20	72	1	116	CRT	No	800	600	8	40	256	28	60
Passive Single / TFT	No	640	480	4	40	112	1	110	CRT	No	800	600	4	40	256	28	60
Passive Single / TFT	No	640	480	8	20	72	1	58	CRT	No	800	600	8	40	256	28	60
Color Passive Dual [1]	No	640	480	8	13	56	1	77	CRT	No	800	600	8	40	256	28	60
Color Passive Dual	No	800	600	4	24	72	1	91	CRT	No	800	600	4	40	256	28	60
Color Passive Dual [2]	No	800	600	8	13	56	1	50	CRT	No	800	600	8	40	256	28	60
Passive Single / TFT	Yes	640	240	8	15	56	1	89	CRT	No	800	600	8	40	256	28	60
Passive Single / TFT	Yes	640	480	4	29	80	1	83	CRT	No	800	600	4	40	256	28	60
Mono Passive Dual	Yes	640	480	4	23	72	1	134	CRT	No	800	600	4	40	256	28	60
Mono Passive Dual	Yes	640	480	8	13	56	1	77	CRT	No	800	600	8	40	256	28	60
Color Passive Dual	Yes	640	480	4	19	56	1	113	CRT	No	800	600	4	40	256	28	60
Color Passive Dual	Yes	640	480	8	11	48	1	66	CRT	No	800	600	8	40	256	28	60
Passive Single / TFT	No	640	240	4	27	72	1	157	CRT	Yes	800	600	4	40	256	28	60
Passive Single / TFT	No	640	240	8	13	48	1	78	CRT	Yes	800	600	8	40	256	28	60
Passive Single / TFT	No	640	480	4	27	72	1	78	CRT	Yes	800	600	4	40	256	28	60
Mono Passive Dual	No	640	480	4	21	64	1	123	CRT	Yes	800	600	4	40	256	28	60
Mono Passive Dual	No	640	480	8	10	40	1	61	CRT	Yes	800	600	8	40	256	28	60
Color Passive Dual	No	640	480	4	17	56	1	101	CRT	Yes	800	600	4	40	256	28	60
Color Passive Dual	No	640	480	8	8.8	40	1	53	CRT	Yes	800	600	8	40	256	28	60
Passive Single / TFT	Yes	640	240	4	20	56	1	119	CRT	Yes	800	600	4	40	256	28	60
Passive Single / TFT	Yes	640	480	4	20	56	1	59	CRT	Yes	800	600	4	40	256	28	60
Mono Passive Dual	Yes	640	480	4	17	56	1	101	CRT	Yes	800	600	4	40	256	28	60
Mono Passive Dual	Yes	640	480	8	8.5	32	1	52	CRT	Yes	800	600	8	40	256	28	60
Color Passive Dual	Yes	640	480	4	14	48	1	84	CRT	Yes	800	600	4	40	256	28	60

☐ = Example Frame Rates with Ink Layer Enabled

The FIFO values for these display modes must be set as follows:

1. REG[04Ah] must be set to 1Ah.
2. REG[04Ah] must be set to Fh.

18.2.5 Frame Rates for LCD and NTSC TV with EISD Enabled

Table 18-5 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	5.3	32	1	62	NTSC TV	No	640	480	16	14.32	270	22	62
	No	640	480	4	31	120	1	84	NTSC TV	No	640	480	4	14.32	270	22	62
	No	640	480	8	18	88	1	51	NTSC TV	No	640	480	8	14.32	270	22	62
Mono Passive Dual	No	640	480	4	25	104	1	139	NTSC TV	No	640	480	4	14.32	270	22	62
	No	640	480	8	15	80	1	86	NTSC TV	No	640	480	8	14.32	270	22	62
Color Passive Dual	No	640	480	4	21	88	1	119	NTSC TV	No	640	480	4	14.32	270	22	62
	No	640	480	8	13	72	1	75	NTSC TV	No	640	480	8	14.32	270	22	62
Color Passive Dual	No	800	600	8	13	72	1	49	NTSC TV	No	640	480	8	14.32	270	22	62
Passive Single / TFT	Yes	640	480	4	24	96	1	67	NTSC TV	No	640	480	4	14.32	270	22	62
Mono Passive Dual	Yes	640	480	4	20	80	1	115	NTSC TV	No	640	480	4	14.32	270	22	62
	Yes	640	480	8	12	64	1	70	NTSC TV	No	640	480	8	14.32	270	22	62
Color Passive Dual	Yes	640	480	4	17	72	1	99	NTSC TV	No	640	480	4	14.32	270	22	62
	Yes	640	480	8	11	64	1	64	NTSC TV	No	640	480	8	14.32	270	22	62
Passive Single / TFT	No	640	240	4	19	72	1	110	NTSC TV	Yes	640	480	4	14.32	270	22	62
	No	640	240	8	12	64	1	70	NTSC TV	Yes	640	480	8	14.32	270	22	62
	No	640	480	4	19	72	1	55	NTSC TV	Yes	640	480	4	14.32	270	22	62
Mono Passive Dual	No	640	480	4	16	64	1	94	NTSC TV	Yes	640	480	4	14.32	270	22	62
	No	640	480	8	10	56	1	59	NTSC TV	Yes	640	480	8	14.32	270	22	62
Color Passive Dual	No	640	480	4	14	56	1	83	NTSC TV	Yes	640	480	4	14.32	270	22	62
	No	640	480	8	9.3	48	1	56	NTSC TV	Yes	640	480	8	14.32	270	22	62
Passive Single / TFT	Yes	640	240	4	16	64	1	94	NTSC TV	Yes	640	480	4	14.32	270	22	62
	Yes	640	240	8	10	56	1	59	NTSC TV	Yes	640	480	8	14.32	270	22	62
Mono Passive Dual	Yes	640	480	4	14	56	1	83	NTSC TV	Yes	640	480	4	14.32	270	22	62
	Yes	640	480	8	9	48	1	54	NTSC TV	Yes	640	480	8	14.32	270	22	62
Color Passive Dual	Yes	640	480	4	12	48	1	72	NTSC TV	Yes	640	480	4	14.32	270	22	62

☐ = Example Frame Rates with Ink Layer Enabled

18.2.6 Frame Rates for LCD and PAL TV with EISD Enabled

Table 18-6 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 / TFT	No	320	240	16	5.3	32	1	62	PAL TV	No	640	480	16	17.73	495	72	56
	No	640	480	4	31	120	1	84	PAL TV	No	640	480	4	17.73	495	72	56
	No	640	480	8	18	88	1	51	PAL TV	No	640	480	8	17.73	495	72	56
Mono Passive Dual	No	640	480	4	25	104	1	139	PAL TV	No	640	480	4	17.73	495	72	56
	No	640	480	8	15	80	1	86	PAL TV	No	640	480	8	17.73	495	72	56
Color Passive Dual	No	640	480	4	21	88	1	119	PAL TV	No	640	480	4	17.73	495	72	56
	No	640	480	8	13	72	1	75	PAL TV	No	640	480	8	17.73	495	72	56
Color Passive Dual	No	800	600	8	13	72	1	49	PAL TV	No	640	480	8	17.73	495	72	56
Passive Single / TFT	Yes	640	480	4	24	96	1	67	PAL TV	No	640	480	4	17.73	495	72	56
Mono Passive Dual	Yes	640	480	4	20	80	1	115	PAL TV	No	640	480	4	17.73	495	72	56
	Yes	640	480	8	12	64	1	70	PAL TV	No	640	480	8	17.73	495	72	56
Color Passive Dual	Yes	640	480	4	17	72	1	99	PAL TV	No	640	480	4	17.73	495	72	56
	Yes	640	480	8	11	64	1	64	PAL TV	No	640	480	8	17.73	495	72	56
Passive Single / TFT	No	640	240	4	19	72	1	110	PAL TV	Yes	640	480	4	17.73	495	72	56
	No	640	240	8	12	64	1	70	PAL TV	Yes	640	480	8	17.73	495	72	56
	No	640	480	4	19	72	1	55	PAL TV	Yes	640	480	4	17.73	495	72	56
Mono Passive Dual	No	640	480	4	16	64	1	94	PAL TV	Yes	640	480	4	17.73	495	72	56
	No	640	480	8	10	56	1	59	PAL TV	Yes	640	480	8	17.73	495	72	56
Color Passive Dual	No	640	480	4	14	56	1	83	PAL TV	Yes	640	480	4	17.73	495	72	56
	No	640	480	8	9.3	48	1	56	PAL TV	Yes	640	480	8	17.73	495	72	56
Passive Single / TFT	Yes	640	240	4	16	64	1	94	PAL TV	Yes	640	480	4	17.73	495	72	56
	Yes	640	240	8	10	56	1	59	PAL TV	Yes	640	480	8	17.73	495	72	56
Mono Passive Dual	Yes	640	480	4	14	56	1	83	PAL TV	Yes	640	480	4	17.73	495	72	56
	Yes	640	480	8	9	48	1	54	PAL TV	Yes	640	480	8	17.73	495	72	56
Color Passive Dual	Yes	640	480	4	12	48	1	72	PAL TV	Yes	640	480	4	17.73	495	72	56

[   ] = Example Frame Rates with Ink Layer Enabled

19 POWER SAVE MODE

The S1D13506 has been designed for very low-power applications. During normal operation, internal clock networks are dynamically disabled when not required. Similarly, the LCD and/or CRT/TV pipelines are shut down when not required in the selected display mode. Additionally, the S1D13506 has a software initiated power save mode.

19.1 Display Modes

The S1D13506 resets with both displays inactive, i.e. neither the LCD nor CRT/TV pipelines are active. The displays are independently enabled/disabled by REG[1FCh] bits 2-0: the CRT/TV is instantaneously enabled/disabled by these bits; the LCD is powered up/down according to the sequences in Section 7.4, "Power Sequencing" on page 1-61.

19.2 Power Save Mode

Power save mode is invoked by setting REG[1F0h] bit 0 to 1. In power save mode, both displays are disabled: the CRT/TV is instantaneously disabled; the LCD is powered down according to the sequences in Section 7.4, "Power Sequencing" on page 1-61. Access to memory is not allowed and the memory controller merely refreshes the memory in the method selected by REG[021h]. Register access is allowed.

19.3 Power Save Status Bits

LCD Power Save Status bit

The LCD Power Save Status bit (REG[1F1h] bit 0), when 1, indicates that the panel is powered down. When this bit is 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 is 1. This bit is 1 after chip reset.

Memory Controller Power Save Status bit

The Memory Controller Power Save Status bit (REG[1F1h] bit 1, when 1, indicates that the DRAM interface is powered down - the DRAM is either in self-refresh mode or completely idle. This condition occurs shortly after power save mode is invoked, provided Self-Refresh or No Refresh is pre-selected (see REG[021h] bits 7-6); this condition will never occur if CBR Refresh is selected. When this bit is 0, the DRAM interface is active. The system may disable the memory clock source when this bit is 1. This bit is 0 after chip reset.

19.4 Power Save Mode Summary

Table 19-1 Power Save Mode Summary

Function	Power Save Mode		
	LCD Disabled	CRT/TV Disabled	Power Save
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	–	No Output Current	No Output Current
DRAM interface	Active	Active	Refresh Only[3]
Host Interface	Active	Active	Active

Note: 1. LCD pixel clock required.
2. CRT/TV pixel clock required.
3. Selectable; may observe CBR refresh, self-refresh or no refresh at all.

20 CLOCKS

20.1 Clock Selection

The following diagram provides a logical representation of the S1D13506 internal clocks.
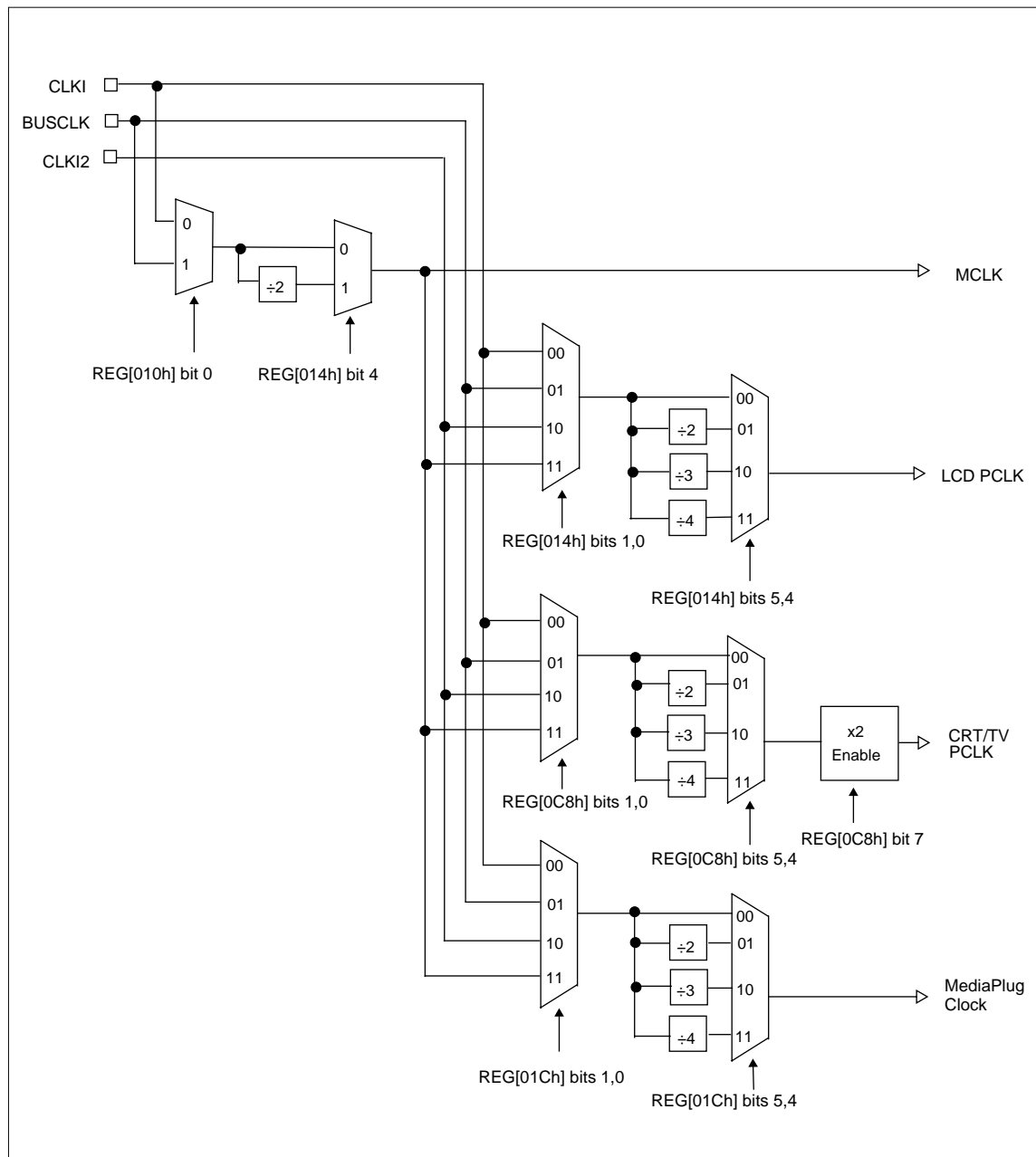
Figure 20-1 Clock Selection

20.2 Clock Descriptions

20.2.1 MCLK

MCLK should be configured as close to its maximum (40MHz) as possible. The S1D13506 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.

20.2.2 LCD PCLK

LCD PCLK should be chosen to match the optimum frame rate of the panel. See Section 18, "Clocking" on page 1-188 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. Panels typically have a range of permissible frame rates making it possible to choose a higher PCLK frequency and adjust the horizontal non-display period (see REG[052h]) to bring the frame-rate down to its optimal value.

20.2.3 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 BUSCLK or MCLK.

TV mode with flicker filter requires PCLK to be twice (2x) the standard NTSC (14.xxxMHz) and PAL (17.xxxMHz) clocks. A clock multiplier is used to create this clock, REG[018h] bit 7 is used to enable it. Note that the clock 2x clock could also be used for CRT support.

20.2.4 MediaPlug Clock

The MediaPlug Clock must be twice (2x) the frequency of VMPCLK. For timing see Section 7.7, "MediaPlug Interface Timing" on page 1-105. VMPCLK is typically in the range 6-8MHz so MediaPlug Clock must be in the range of 12-16MHz.

20.3 Clocks vs. Functions

The S1D13506 has five clock signals. Not all clock signals must be active for certain chip functions to be carried out. The following table shows which clocks are required for each chip function.

Table 20-1 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	–	–	–	–	–

Note: The S1D13506 contains sophisticated power management that dynamically shuts down clocks when not needed.

21 MECHANICAL DATA

128-pin QFP15 surface mount package
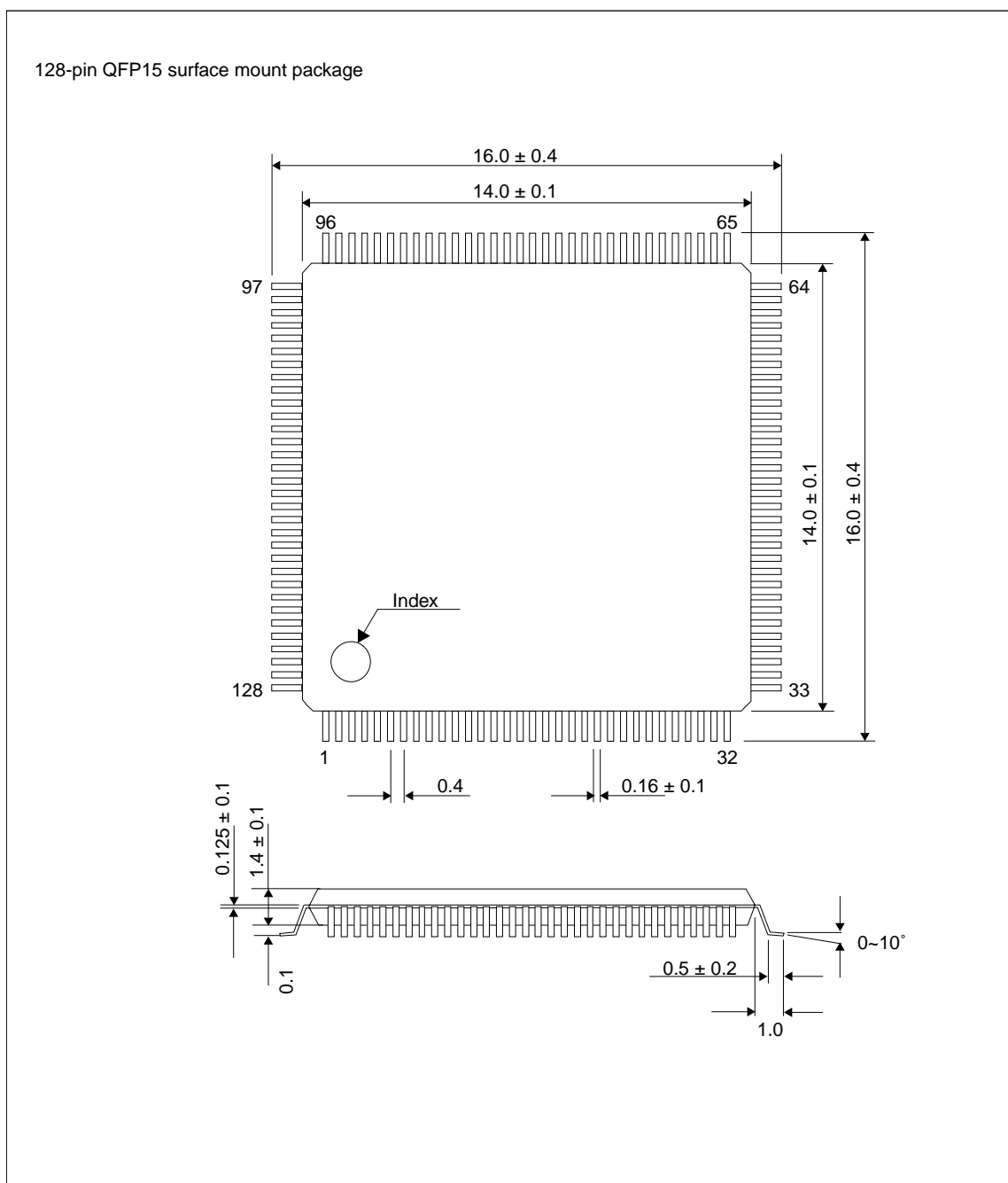
Figure 21-1 Mechanical Drawing QFP15

21: MECHANICAL DATA

THIS PAGE IS BLANK.

S1D13506F00A
Color LCD/CRT/TV Controller

Programming Notes and Examples

List of Figures

List of Tables

1 INTRODUCTION

This guide provides information on programming the S1D13506 Color LCD/CRT/TV Controller. Included are algorithms which demonstrate how to program the S1D13506. This guide discusses Power-on Initialization, Panning and Scrolling, LUT initialization, LCD Power Sequencing, SwivelView™, etc.

This guide also introduces the Hardware Abstraction Layer (HAL), which is designed to simplify the programming of the S1D13506. Most S1D1350x and S1D1370x products have HAL support, thus allowing OEMs to do multiple designs with a common code base.

2 INITIALIZATION

This section describes how to initialize the S1D13506.

S1D13506 initialization can be broken into three steps.

- Enable the S1D13506 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 S1D13506 is to use the utility program 13506CFG.EXE which to generates a header file that can be used by Windows CE or the HAL.

The following table represents the sequence and values written to the S1D13506 registers to control a configuration with these specifications:

- 640x480 color format 1 dual passive LCD at 78Hz.

- 16-bit data interface.

- 8 bit-per-pixel (bpp) color depth - 256 colors.

- 40 MHz input clock CLKI.

- CLKI used for BUSCLK (1:1); PCLK (2:1); MCLK (1:1).

- 50 ns EDO-DRAM, 2 CAS, 32 ms refresh.

Table 2-1 S1D13506 Initialization Sequence

Register	Value	Notes	See Also
[001h]	0000 0000	**Enable the Memory/Register Select Bit.**	
[1FCh]	0000 0000	**Disable the display outputs.**	
[004h] [008h]	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]	0000 0000 0000 0110 0000 0001 0000 0001	**Program the Frame Buffer Memory Configuration Registers.**	see REG[020h] - REG[02Bh] for details

Table 2-1 S1D13506 Initialization Sequence (Continued)

Register	Value	Notes	See Also
[030h] [031h] [032h] [034h] [035h] [036h] [038h] [039h] [03Ah] [03Bh] [03Ch]	0010 0110 0000 0000 0100 1111 0001 1111 0000 0000 0000 0000 1101 1111 0000 0001 0010 1100 0000 0000 0000 0000	**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.	
[040h] [041h] [042h] [043h] [044h] [046h] [047h] [048h] [04Ah] [04Bh]	0000 0011 0000 0000 0000 0000 0000 0000 0000 0000 0100 0000 0000 0001 0000 0000 0000 0000 0000 0000	**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.	
[050h] [052h] [053h] [054h] [056h] [057h] [058h] [059h] [05Ah] [05Bh]	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	**Program the CRT/TV Timing control registers.** All values are = don't care for this example.	
[060h] [062h] [063h] [064h] [066h] [067h] [068h] [06Ah] [06Bh]	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	**Program the CRT/TV Display Output Format and Configuration Registers including the FIFOs.** For this example, these values are = don't care.	
[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.	

Table 2-1 S1D13506 Initialization Sequence (Continued)

Register	Value	Notes	See Also
[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.	
[100h] [101h] [102h] [103h] [104h] [105h] [106h] [108h] [109h] [10Ah] [10Ch] [10Dh] [110h] [111h] [112h] [113h] [114h] [115h] [116h] [118h] [119h]	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 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	**Program the 2D acceleration (BitBLT) registers to a known state.**	
[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 init13506.c file shows the example.	see Section Section 4, "Look-Up Table (LUT)" on page 2-8.
[1F0h]]	0000 0000	**Turn off Power Save Mode**	
[1F4h]	0000 0000	**Disable Watchdog Timer**	
[1FCh]	0000 0001	**Enable the Display** For this example, enable LCD only	see REG[1FCh]

3 MEMORY MODELS

The S1D13506 is capable of several color depths. The memory model for each color depth is packed pixel. The S1D13506 supports 4, 8, 15 and 16 bit-per-pixel (bpp) memory models.

3.1 Display Buffer Location

The S1D13506 supports either a 512k byte or 2M byte display buffer. The display buffer is memory mapped and is accessible directly by software. The memory block location assigned to the S1D13506 display buffer varies with each individual hardware platform.

For further information on the display buffer, see the *"S1D13506 Hardware Fuctional Specification"*.

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)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Pixel 0 Bits 7-0							

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 S1D13506 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 15 Bpp (32768 Colors/32 Gray Shades)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	Red Component Bits 4-0					Green Component Bits 4-3	
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 15 Bpp in Two Bytes of Display Buffer

At a color depth of 15 bpp the S1D13506 is capable of displaying 32768 colors. The 32768 color pixel is divided into four parts: one reserved bit, five bits for red, five 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 a monochrome panel be used at this color depth, the output sends the five bits of the green LUT component to the modulator for a total of 32 possible gray shades. If dithering is disabled, the maximum number of gray shades is 16.

3.5 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 3-4 Pixel Storage for 16 Bpp in Two Bytes of Display Buffer

At a color depth of 16 bpp the S1D13506 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 S1D13506 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 "*S1D13506 Hardware Functional Specification*".

The S1D13506 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 15/16 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 S1D13506 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 "*S1D13506 Hardware Functional Specification*".

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 "*S1D13506 Hardware Functional Specification*".

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 S1D13506 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
15 bpp gray				16 gray shades	64 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
15 bpp color				4096 colors	32768 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 S1D13506 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

EPSON

8 bpp color

When the S1D13506 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 S1D13506 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 S1D13506 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

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
21	40	00	F0	61	B0	F0	C0	A1	70	50	50	E1	30	20	40
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

15 bpp color

The Look-Up Table is bypassed at this color depth, hence programming the LUT is not required.

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.

15 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 five bits of green are used to set the absolute intensity of the image. This results in 32 gray shades when dithering is enabled and 16 gray shades when dithering is disabled.

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 2-18.

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 2M bytes of display memory is 512. 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.
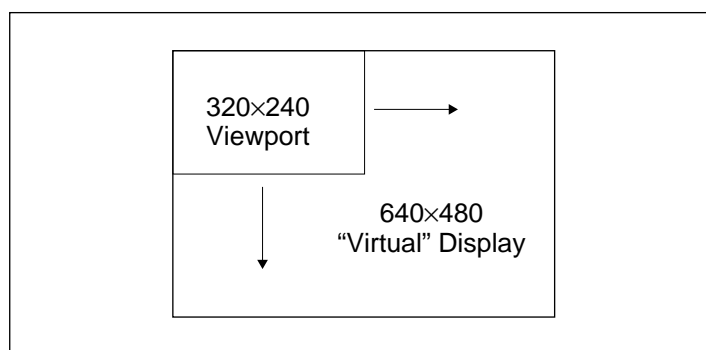
Figure 5-1 Viewport Inside a Virtual Display

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:

$$\text{Offset} = \text{PixelsPerVirtualLine} \div \text{PixelsPerWord}$$

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.

$$\begin{aligned} \text{PixelsPerWord} &= 16 \div \text{bpp} \\ &= 16 \div 8 \\ &= 2 \end{aligned}$$

To calculate the offset value for this example, the following formula is used.

$$\begin{aligned} \text{Offset} &= \text{PixelsPerVirtualLine} \div \text{PixelsPerWord} \\ &= 800 \div 2 \\ &= 400 \\ &= 190\text{h words} \end{aligned}$$

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 S1D13506 registers for a 640x480 panel. (See Section 2, "Initialization" on page 2-2).

2. Calculate the number of pixels per word.

$$\begin{aligned} \text{PixelsPerWord} &= 16 \div \text{bpp} \\ &= 16 \div 4 \\ &= 4 \end{aligned}$$

3. Determine the offset register value.

$$\begin{aligned} \text{Offset} &= \text{PixelsPerVirtualLine} \div \text{PixelsPerWord} \\ &= 800 \div 4 \\ &= 200 \text{ words} \\ &= 0\text{C8h words} \end{aligned}$$

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.

$$\begin{aligned} \text{MemoryRequired} &= \text{WordsPerVirtualLine} \times 2 \times \text{NumberOfLines} \\ &= 200 \times 2 \times 600 \\ &= 240,000 \text{ bytes} \end{aligned}$$

This display configuration is supported on a system with the minimum supported memory size of 512K bytes. 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 less than 15 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 S1D13506 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
15	1	1
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
15/16	none

Note: The pixel panning registers are not required for color depths of 15 or 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 2-2 and Section 5.1, "Virtual Display" on page 2-15 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 2-19.

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.

$$
\begin{aligned}
\text{OffsetWords} \quad &= \text{PixelsPerVirtualLine} \div \text{PixelsPerWord} \\
&= 800 \div 4 \\
&= 200 \\
&= \text{C8h}
\end{aligned}
$$

2. Increment the display start address by the number of words per virtual line.

$$
\begin{aligned}
\text{StartAddress} \quad &= \text{StartAddress} + \text{OffsetWords} \\
&= \text{StartAddress} + \text{C8h}
\end{aligned}
$$

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 S1D13506 has been designed for very low-power applications. During normal operation, the internal clocks are dynamically disabled when not required. The S1D13506 design also includes a Power Save Mode to further save power. When Power Save Mode is initiated, automatic LCD power sequencing takes place 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 2-26.

For Power Save Mode AC Timing, see the *"S1D13506 Hardware Functional Specification"*.

6.1 Overview

The S1D13506 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). The type of DRAM refresh used during Power Save Mode can also be selected by software.

While Power Save Mode is enabled the following conditions apply.

- Display(s) are inactive.
- Registers are accessible.
- Memory is in-accessible.
- LUT is 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 S1D13506 back to normal mode.

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 S1D13506 DRAM interface. When this bit returns a 1, the DRAM interface is powered down (the DRAM is either in self-refresh mode or completely idle). When this bit returns a 0, the DRAM interface is active. This bit will return a 0 after a chip reset.

For this bit to return a 1, the DRAM Refresh Select bits must select either self-refresh or no refresh. For information on the DRAM Refresh Select bits, see Section 6.2.3, "DRAM Refresh Selection" on page 2-25.

Note: The memory clock source may be disabled when this bit returns a 1.

6.2.3 DRAM Refresh Selection

REG[021h] DRAM Refresh Rate Register							
Refresh Select Bit 1	Refresh Select Bit 0	n/a	n/a	n/a	DRAM Refresh Rate Bit 2	DRAM Refresh Rate Bit 1	DRAM Refresh Rate Bit 0

The Refresh Select bits specify the type of DRAM refresh used while Power Save Mode is enabled. DRAM refresh selection is as follows.

Table 6-1 Refresh Selection

Refresh Select Bits [1:0]	DRAM Refresh Type
00	CAS-before-RAS (CBR) refresh
01	Self-Refresh
1X	No Refresh

The Refresh Select bits must be set before Power Save Mode is enabled. While CBR refresh is selected the memory controller cannot be powered down and the memory clock source must remain active. If either self-refresh or no refresh is selected the memory controller may be powered down and the memory clock source may be disabled. For further information, see Section 6.2.2, "Power Save Status Bits" on page 2-24.

Note: The Refresh Select bits must not be changed while in power save mode.

7 LCD POWER SEQUENCING

The S1D13506 is designed with internal circuitry which automates 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 varies depending on the LCD bias power supply design. For example, the LCD bias power supply on the S5U13506P00C Evaluation board requires 0.5 seconds to fully discharge. Other power supply designs may vary.

For most applications automatic power sequencing is the appropriate choice, see Section 7.1, "Automatic Sequencing". However, there may be situations where the internal time delay is insufficient to discharge the LCD bias power supply before the LCD signals are shut down. For the sequence used to manually power-off the LCD panel, see Section 7.2, "Manual Sequencing".

7.1 Automatic Sequencing

LCD power sequencing is automatically provided when the S1D13506 is powered-on or powered-off using the built-in Power Save Mode. The Power Save Mode is enabled using REG[1F0h] bit 0. For more information on Power Save Mode, see Section 6, "Power Save Mode".

For LCD power sequencing AC Timing, see the *"S1D13506 Hardware Functional Specification"*.

7.2 Manual Sequencing

In cases where automatic LCD power sequencing is not applicable, manual sequencing can be done. This section assumes the LCD bias power is controlled through GPIO1. The S1D13506 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 *"S1D13506 Hardware Functional Specification"*.

7.2.1 Registers

REG[040h] LCD Display Mode Register							
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

When set to 1, this bit disables the LCD display pipeline and forces all LCD data outputs to zero. This effectively blanks the screen.

> **Note:** If a dual panel is used, the Dual Panel Buffer must be disabled before blanking the LCD display. This is done by setting REG[041h] bit 0 to 1b.

7.2.2 Enabling the LCD Panel

If the LCD bias power supply timing requirements are different than those timings built into the S1D13506 automated LCD power sequencing, it may be necessary to manually enable the LCD panel. In such a case, the following procedure applies.

1. Enable the LCD signals - Set REG[040h] = 0.

   Note: If a dual panel is used, enable the Dual Panel Buffer by setting REG[041h] bit 0 = 0b.

2. Enable GPIO1 to activate the LCD bias power.

7.2.3 Disabling the LCD Panel

If the LCD bias power supply timing requirements are different than those timings built into the S1D13506 automated LCD power sequencing, it may be necessary to manually disable the LCD panel. In such a case, the following procedure applies.

1. Disable the LCD power using GPIO1.

2. Wait the required delay time for the LCD bias power supply to discharge.

3. Disable the LCD signals - Set REG[040h] = 1.

4. Disable the LCD pixel clock source if desired. (Optional)

8 HARDWARE CURSOR/INK LAYER

8.1 Introduction

The S1D13506 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 S1D13506 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 64x64 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 1024 bytes of display buffer and the Ink Layer requires (display width x 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 64x64 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	Display Buffer Size - 1024
1 - FFh	Display Buffer Size - $(n \times 8192)$

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.

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 Y 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.

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 *"13506 Hardware Functional Specification"*, document number X25B-A-001-04.

8.3 Initialization

This section describes the process of initializing the S1D13506 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 x Panel Height) x factor ÷ 16

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

Note: The dual panel buffer always starts at (Display Memory Size - Dual Panel Buffer Size).

The size of a hardware cursor is always 1024 bytes.
The size of the ink layer in bytes is (display width x 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 2M 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 2M 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 640x480 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	Display Buffer Size - 1024
1 - FFh	Display Buffer Size - (n x 8192)

$$
\begin{aligned}
n &= \text{InkLayerSize} \div \text{RequiredBlockSize} \\
&= 76{,}800 \div 8192 \\
&= 9.375
\end{aligned}
$$

Fractional values cannot be programmed, therefore round up to an address of 10 (0Ah). This reserves 10 x 8192 = 81,920 bytes 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 x 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 64x64 image at a color depth of 2 bpp. The Ink Layer is the same size as the virtual display (width x 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.
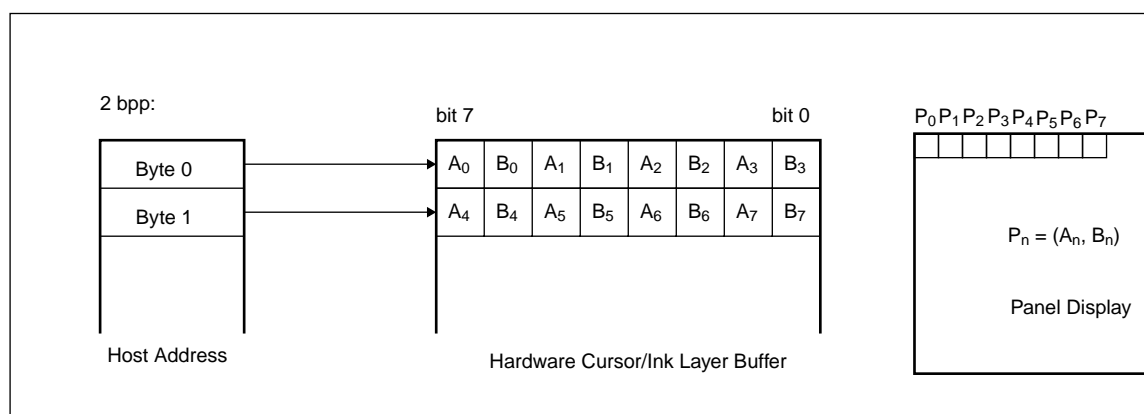
Figure 8-1 Hardware Cursor/Ink Layer Data Format

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 1024 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 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 1024 bytes.

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 8192 bytes of display buffer, not 1024 bytes, because the start address moves in steps of 8192 bytes.

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 x 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 x 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 high.

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 640x480 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 S1D13506 SwivelView™

The S1D13506 provides hardware support for SwivelView™ in 8, 15 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 (1024 bytes in 8 bpp, 2048 bytes in 15/16 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[01FCh]) 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		15/16 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 640x480 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 $\times$ 2).

Table 9-3 LCD Display Start Address Values

SwivelView™ Enable		Display Rotated	LCD Display Start Address Value	
Bit 1	Bit 0		15/16 Bpp mode	8 Bpp mode
0	0	0 degrees	0	0
0	1	90 degrees	(1024 - PW)	(1024 - PW) $\div$ 2
1	0	180 degrees	(Stride $\times$ PH - (Stride - 2 $\times$ PW)) $\div$ 2 - 1	(Stride $\times$ PH - (Stride - PW)) $\div$ 2 - 1
1	1	270 degrees	(Stride $\times$ PH) $\div$ 2 - 1	(Stride $\times$ PH) $\div$ 2 - 1

9.3 Limitations

The following limitations apply when SwivelView™ bit 0 is set to 1 (rotation by 90˚ or 270˚):

• Only 8/15/16 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 blit 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*

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.

    $$
    \begin{aligned}
    StartAddress \quad &= (ScanBytes \times PanelHeight - (ScanBytes - 2 \times PanelWidth)) \div 2 - 1 \\
    &= (1280 \times 480 - (1280 - 2 \times 640)) \div 2 - 1 \\
    &= (1280 \times 480) \div 2 - 1 \\
    &= 307199 \\
    &= 4\text{AFFFh}
    \end{aligned}
    $$

    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 S1D13506 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 S1D13506 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, 15 bpp, and 16 bpp.

10.1 Registers

The BitBLT control registers on the S1D13506 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 blit engine. When a read from this bit returns 0, the blit engine is idle and is ready for the next operation. When a read from this bit returns a 1, the blit 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.

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 blit. If this bit = 0, the destination blit is stored as a rectangular region of memory. If this bit = 1, the destination blit 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 blit. If this bit = 0, the source blit is stored as a rectangular region of memory. If this bit = 1, the source blit 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 blit, Move blit, 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 Blit and Move Blit	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]	Blit Operation
0000	Write Blit with ROP
0001	Read Blit
0010	Move Blit in positive direction with ROP
0011	Move Blit in negative direction with ROP
0100	Transparent Write Blit
0101	Transparent Move Blit in positive direction
0110	Pattern Fill with ROP
0111	Pattern Fill with transparency
1000	Color Expansion
1001	Color Expansion with transparency
1010	Move Blit with Color Expansion
1011	Move Blit 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.

EPSON

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 Blit 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 ≥ 2.

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 blits. 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 S1D13506 supports 13 fundamental BitBLT operations:

• Write Blit with ROP.

• Read Blit.

• Move Blit in positive direction with ROP.

• Move Blit in negative direction with ROP.

• Transparent Write Blit.

• Transparent Move Blit in positive direction.

• Pattern Fill with ROP.

• Pattern Fill with Transparency.

• Color Expansion.

• Color Expansion with Transparency.

• Move Blit with Color Expansion.

• Move Blit 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 Blit with ROP, Transparent Write Blit, Color Expansion, Color Expansion with Transparency, Read Blit) 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, then 32-bit CPU instructions are acceptable. Otherwise, two back to back 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 blit 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 S1D13506. If the S1D13506 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 Blit with ROP

The Write Blit increases the speed of transferring data from system memory to the display buffer.

The Write Blit with ROP fills a specified area of the display buffer with data supplied by the CPU. This blit is typically used to copy a bitmap image from system memory to the display buffer. The Write Blit 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 Blit requires the CPU to provide data. The blit engine expects to receive a certain number of WORDS. For 15/16 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 blit engine expects is calculated using the following formula.

$$\begin{aligned} \text{nWORDS} &= \text{nPixels} \\ &= \text{BlitWidth} \times \text{BlitHeight} \end{aligned}$$

For 8 bpp color depths, the formula must take into consideration that the blit engine accepts only WORD accesses and each pixel is one BYTE. The blit 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 Blit 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 blit engine expects for 8 bpp color depths is shown in the following formula.

$$\text{nWORDS} = ((\text{BlitWidth} + 1 + \text{SourcePhase}) \div 2) \times \text{BlitHeight}$$

Note: The blit engine counts WORD writes in the blit 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, two back to back 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 blit rectangle) using the following formula.

$$
\begin{aligned}
\text{DestinationAddress} \quad &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel}) \\
&= (38 \times 640) + (25 \times 1) \\
&= 24345 \\
&= \text{5F19h}
\end{aligned}
$$

where:
BytesPerPixel = 1 for 8 bpp
BytesPerPixel = 2 for 15/16 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 Blit 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:

$$
\begin{aligned}
\text{BltMemoryOffset} \quad &= \text{DisplayWidthInPixels} \div \text{BytesPerPixel} \\
&= 640 \div 2 \\
&= \text{140h}
\end{aligned}
$$

REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

9. Calculate the number of WORDS the blit engine expects to receive.

$$
\begin{aligned}
\text{nWORDS} \quad &= ((\text{BlitWidth} + 1 + \text{SourcePhase}) \div 2) \times \text{BlitHeight} \\
&= (100 + 1) \div 2 \times 20 \\
&= 1000 \\
&= \text{3E8h}
\end{aligned}
$$

10. Program the BitBLT Destination/Source Linear Select bits for a rectangular blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

Start the blit operation **and wait for the blit 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 Blit FIFO, confirm the Blit 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 Blit FIFO.

Table 10-4 Possible Blit 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)

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 blit engine must be word (16-bit) writes. **The blit 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
Blit Width = 2

The following bits are expanded.
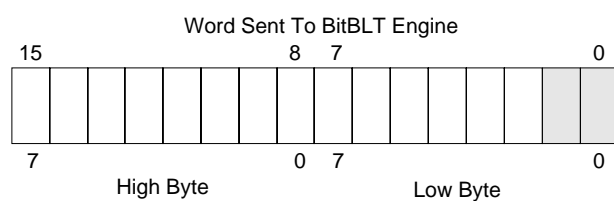
Word Sent To BitBLT Engine

2. To expand bits 0-15 of the word (entire word)

Source Address = 0
Start Bit Position = 7 (bit seven of the low byte)
Blit Width = 16

The following bits are expanded.

Word Sent To BitBLT Engine

```
15                  8  7              0
7           0  7              0
   High Byte       Low Byte
```
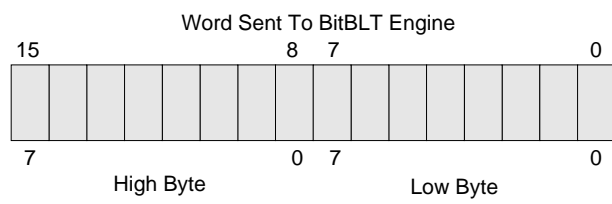
3. To expand bits 8-9 of the word

Source Address = 1
Start Bit Position = 1
Blit Width = 2

The following bits are expanded.

Word Sent To BitBLT Engine

```
15                  8  7              0
7           0  7              0
   High Byte       Low Byte
```
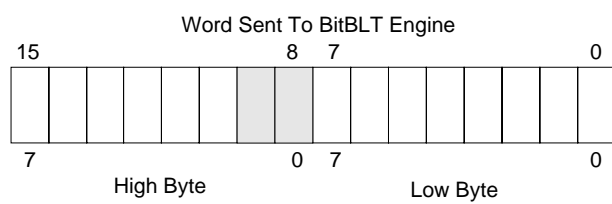
4. To expand bits 0,15-14 of the word

Source Address = 0
Start Bit Position = 0
Blit Width = 3

The following bits are expanded.

Word Sent To BitBLT Engine

```
15                  8  7              0
7           0  7              0
   High Byte       Low Byte
```
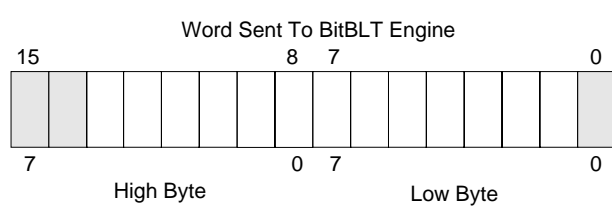
All subsequent WORDS in one blit line are then serially expanded starting at bit 7 of the low byte until the end of the blit line. All unused bits in the last WORD are discarded. It is extremely important that the exact number of WORDS is provided to the blit engine. The number of WORDS is calculated from the following formula. This formula is valid for all color depths (8/15/16 bpp).

$$nWords = ((Sx \bmod 16 + BltWidth + 15) \div 16) \times BltHeight$$

where:
Sx is the X coordinate of the starting pixel in a word aligned monochrome bitmap.
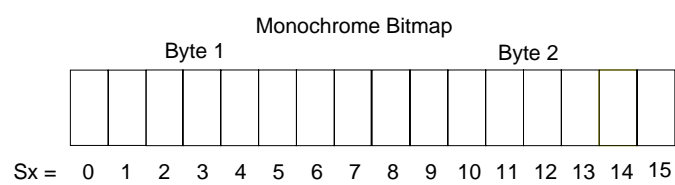
Example 10:   Color expand a rectangle of 12 × 18 starting at the coordinates Sx = 125, Sy = 17 using a 640×480 display at a color depth of 8 bpp.

This example assumes a monochrome, WORD aligned bitmap of dimensions 300 x 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.

$$
\begin{aligned}
SourceAddress \quad &= BitmapOrigin + (y \times SourceStride) + (x \div 8) \\
&= A + (Sy \times SourceStride) + (Sx \div 8) \\
&= A + (17 \times 38) + (125 \div 8) \\
&= A + 646 + 15 \\
&= A + 661
\end{aligned}
$$

where:

$$
\begin{aligned}
SourceStride \quad &= (BitmapWidth + 15) \div 16 \\
&= (300 + 15) \div 16 \\
&= 19 \text{ WORDS per line} \\
&= 38 \text{ BYTES per line}
\end{aligned}
$$

2. Calculate the destination address (upper left corner of the screen blit rectangle) using the following formula.

$$
\begin{aligned}
DestinationAddress \quad &= (Y \times ScreenStride) + (X \times BytesPerPixel) \\
&= (30 \times 640) + (20 \times 1) \\
&= 19220 \\
&= 4B14h
\end{aligned}
$$

where:

BytesPerPixel	= 1 for 8 bpp
BytesPerPixel	= 2 for 15/16 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 15/16 bpp color depths the Source address is guaranteed to be even.

6. Program the BitBLT Operation Register to select the Color Expand Blit. REG[103h] is set to 08h.

7. Program the Color Expansion Register. The formula for this example is as follows.

   $$\text{Reg[102h]} = 7 - (Sx \text{ MOD } 8)$$
   $$= 7 - (125 \text{ MOD } 8)$$
   $$= 7 - 5$$
   $$= 2$$

   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 15/16 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 15/16 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.

    $$\text{BltMemoryOffset} = \text{ScreenStride} \div 2$$
    $$= 640 \div 2$$
    $$= 140h$$

    REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

12. Calculate the number of WORDS the blit engine expects to receive.

    First, the number of WORDS in one blit line must be calculated as follows.

    $$nWordsOneLine = ((125 \text{ MOD } 16) + 12 + 15) \div 16$$
    $$= (13 + 12 + 15) \div 16$$
    $$= 40 \div 16$$
    $$= 2$$

    Therefore, the total WORDS the blit engine expects to receive is calculated as follows.

    $$nWords = nWordsOneLine \times 18$$
    $$= 2 \times 18$$
    $$= 36$$

13. Program the BitBLT Destination/Source Linear Select bits for a rectangular blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

    Start the blit operation and **wait for the Blit 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 Blit FIFO, confirm the Blit FIFO is not full (REG[100h] bit 4 returns a 0). One WORD expands into 16 pixels which fills all 16 FIFO words in 15/16 bpp or 8 FIFO words in 8 bpp.

    The following table summarizes how many words can be written to the Blit FIFO.

Table 10-5 Possible Blit 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	
1	1	0	0 (do not write)	0 (do not write)
1	1	1		

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.

$$
\begin{aligned}
\text{DestinationAddress} &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel}) \\
&= (10 \times (640 \times 2)) + (100 \times 2) \\
&= 13000 \\
&= 32\text{C8h}
\end{aligned}
$$

where:
BytesPerPixel = 1 for 8 bpp
BytesPerPixel = 2 for 15/16 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.

$$
\begin{aligned}
\text{BltMemoryOffset} &= \text{ScreenStride} \div 2 \\
&= 640 \\
&= 280\text{h}
\end{aligned}
$$

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 blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

Start the blit operation. REG[100h] is set to 80h.

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.

Destination Address less than Source Address
Use Move BitBLT in Positive Direction

Destination Address greater than Source Address
Use Move BitBLT in Negative Direction
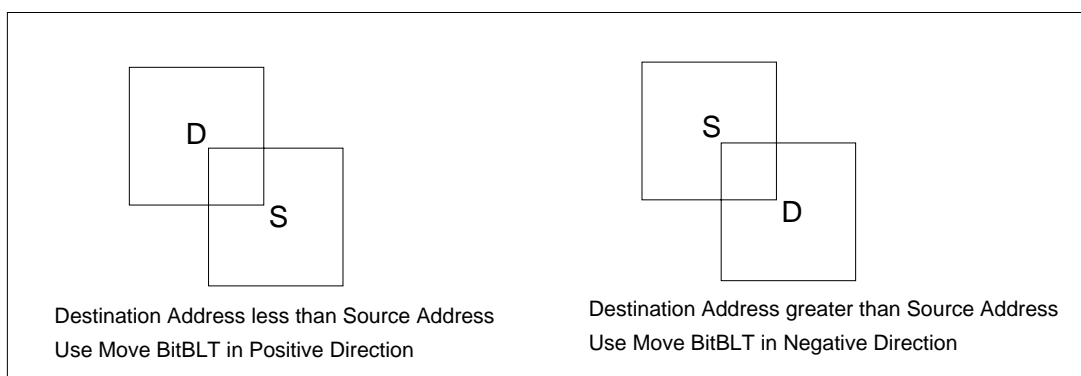
Figure 10-1 Move BitBLT Usage

Example 12:   Copy a 9 $\times$ 321 rectangle at the screen coordinates x = 100, y = 10 to screen coordinates x = 200, y = 20 using a 640$\times$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.

$$
\begin{aligned}
\text{SourceAddress} \quad &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel}) \\
&= (10 \times (640 \times 2)) + (100 \times 2) \\
&= 13000 \\
&= 32C8h
\end{aligned}
$$

$$
\begin{aligned}
\text{DestinationAddress} \ &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel}) \\
&= (20 \times (640 \times 2)) + (200 \times 2) \\
&= 26000 \\
&= 6590h
\end{aligned}
$$

where:
BytesPerPixel = 1 for 8 bpp
BytesPerPixel = 2 for 15/16 bpp
ScreenStride = DisplayWidthInPixels $\times$ 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.

   $$\begin{aligned} \text{BltMemoryOffset} &= \text{ScreenStride} \div 2 \\ &= 640 \\ &= 280h \end{aligned}$$

   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 blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the blit operation. REG[100h] is set to 80h.

   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 blit 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
    $$= ((y + \text{Height} - 1) \times \text{ScreenStride}) + ((x + \text{Width} - 1) \times \text{BytesPerPixel})$$
    $$= ((10 + 321 - 1) \times (640 \times 2)) + ((100 + 9 - 1) \times 2)$$
    $$= 422616$$
    $$= 672D8h$$

    DestinationAddress
    $$= ((Y + \text{Height} - 1) \times \text{ScreenStride}) + ((X + \text{Width} - 1) \times \text{BytesPerPixel})$$
    $$= ((20 + 321 - 1) \times (640 \times 2)) + ((105 + 9 - 1) \times 2)$$
    $$= 435426$$
    $$= 6A4E2h$$

    where:
    BytesPerPixel = 1 for 8 bpp
    BytesPerPixel = 2 for 15/16 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 Blit 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.

$$
\begin{aligned}
\text{BltMemoryOffset} \quad &= \text{ScreenStride} \div 2 \\
&= 640 \\
&= 280\text{h}
\end{aligned}
$$

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 blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

Start the blit operation. REG[100h] is set to 80h.

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 Blit

The Transparent Write Blit increases the speed of transferring data from system memory to the display buffer. Once the Transparent Write Blit begins, the blit engine remains active until all pixels have been written.

The Transparent Write Blit updates a specified area of the display buffer with data supplied by the CPU. This blit 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 Blit on the whole rectangle, the effect is a blit of the red circle only. The Transparent Write Blit supports both Destination Linear and Destination Rectangular modes.

This Transparent Write Blit requires the CPU to provide data. The blit engine expects to receive a certain number of WORDS. For 15/16 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 blit engine expects is calculated using the following formula.

$$
\begin{aligned}
\text{nWORDS} \quad &= \text{nPixels} \\
&= \text{Blit Width} \times \text{Blit Height}
\end{aligned}
$$

For 8 bpp color depths, the formula must take into consideration that the blit engine accepts only WORD accesses and each pixel is one BYTE. The blit 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 Blit 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 blit engine expects for 8 bpp color depths is shown in the following formula.

$$
\text{nWORDS} \quad = ((\text{BlitWidth} + 1 + \text{SourcePhase}) \div 2) \times \text{BlitHeight}
$$

Note: The blit engine counts WORD writes in the blit 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, two back to back 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 blit rectangle), using the formula:

    DestinationAddress $= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel})$
    $= (38 \times 640) + (25 \times 1)$
    $= 24345$
    $= 5F19h$

    where:
    BytesPerPixel = 1 for 8 bpp
    BytesPerPixel = 2 for 15/16 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 Blit. 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: Note that for 15/16 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 $= \text{ScreenStride} \div 2$
    $= 640 \div 2$
    $= 320$
    $= 140h$

    REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

9. Calculate the number of WORDS the blit engine expects to receive.

    nWORDS $= ((\text{BlitWidth} + 1 + \text{SourcePhase}) \div 2) \times \text{BlitHeight}$
    $= (100 + 1 + 0) \div 2 \times 20$
    $= 1000$
    $= 3E8h$

10. Program the BitBLT Destination/Source Linear Select bits for a rectangular blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

Start the blit operation **and wait for the blit 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 Blit FIFO, confirm the Blit 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 Blit FIFO.

Table 10-6 Possible Blit 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)

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 blit. 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 Blit on the whole rectangle, the effect is a blit 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 640x480 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:

$$
\begin{aligned}
\text{SourceAddress} \quad &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel}) \\
&= (10 \times (640 \times 2)) + (100 \times 2) \\
&= 13000 \\
&= 32\text{C8h}
\end{aligned}
$$

$$
\begin{aligned}
\text{DestinationAddress} \quad &= (Y \times \text{ScreenStride}) + (X \times \text{BytesPerPixel}) \\
&= (20 \times (640 \times 2)) + (200 \times 2) \\
&= 26000 \\
&= 6590\text{h}
\end{aligned}
$$

where:
BytesPerPixel = 1 for 8 bpp
BytesPerPixel = 2 for 15/16 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 Blit 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.

$$
\begin{aligned}
\text{BltMemoryOffset} \quad &= \text{ScreenStride} \div 2 \\
&= 640 \\
&= 280\text{h}
\end{aligned}
$$

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 blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

Start the blit operation. REG[100h] is set to 80h.

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 eight pixels square, this results in 64 consecutive bytes for 8 bpp color depths and 128 bytes for 15/16 bpp color depths. For 8 bpp color depths the pattern must begin on a 64 byte boundary, for 15/16 bpp color depths the pattern must begin on a 128 byte boundary.

To fill an area using the pattern BitBLT, the blit engine requires the location of the pattern, the destination rectangle position and size, and the ROP code. The blit 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 $\times$ 250 rectangle at the screen coordinates x = 10, y = 20 with the pattern in off-screen memory at offset 10 0000h using a 640$\times$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:

$$\begin{aligned} \text{DestinationAddress} \quad &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel}) \\ &= (20 \times 640) + (10 \times 1) \\ &= 12810 \\ &= 320\text{Ah} \end{aligned}$$

   where:
   BytesPerPixel = 1 for 8 bpp
   BytesPerPixel = 2 for 15/16 bpp
   ScreenStride = DisplayWidthInPixels $\times$ 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 15/16 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.

$$\begin{aligned} \text{BltMemoryOffset} \quad &= \text{ScreenStride} \div 2 \\ &= 640 \div 2 \\ &= 320 \\ &= 140\text{h} \end{aligned}$$

REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

9. Program the BitBLT Destination/Source Linear Select bits for a rectangular blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

    Start the blit operation. REG[100h] is set to 80h.

    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 128 bytes for 15/16 bpp color depths. For 8 bpp color depths the pattern must begin on a 64 byte boundary, for 15/16 bpp color depths the pattern must begin on a 128 byte boundary.

To fill an area using the Pattern Fill BitBLT with Transparency, the blit engine requires the location of the pattern, the destination rectangle position and size, and the transparency color. The blit 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 640x480 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 15/16 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 15/16 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 15/16 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.

   $$
   \begin{aligned}
   \text{BltMemoryOffset} \quad &= \text{ScreenStride} \div 2 \\
   &= 640 \div 2 \\
   &= 320 \\
   &= 140\text{h}
   \end{aligned}
   $$

   REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

9. Program the BitBLT Destination/Source Linear Select bits for a rectangular blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the blit operation. REG[100h] is set to 80h.

   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 because in the first word in a line color expansion is started with the most significant bit of the low byte.

Example 18: *Color expand a 9 × 16 rectangle using the pattern in off-screen memory at 10 0000h and move it to the screen coordinates x = 200, y = 20. Assume a 640x480 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.

$$
\begin{aligned}
\text{DestinationAddress} &= (y \times \text{ScreenStride}) + (x \times \text{BytesPerPixel}) \\
&= (20 \times (640 \times 2)) + (200 \times 2) \\
&= 26000 \\
&= 6590\text{h}
\end{aligned}
$$

where:
BytesPerPixel = 1 for 8 bpp
BytesPerPixel = 2 for 15/16 bpp
ScreenStride = DisplayWidthInPixels × BytesPerPixels = 1280 for 16 bpp

$$
\begin{aligned}
\text{SourceAddress} &= 1\text{M} \\
&= 100000\text{h}
\end{aligned}
$$

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 Blit 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.

$$\begin{aligned} \text{BltMemoryOffset} \quad &= \text{ScreenStride} \div 2 \\ &= 640 \\ &= 280h \end{aligned}$$

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 blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

Start the blit operation. REG[100h] is set to 80h.

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 Blit with Color Expansion*

The Transparent Move Blit with Color Expansion is virtually identical to the Move Blit 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 Blit*

This Read Blit increases the speed of transferring data from the display buffer to system memory. This blit complements the Write Blit and is typically used to save a part of the display buffer to the system memory. Once the Read Blit begins, the blit engine remains active until all the pixels have been read.

The blit engine requires the address to copy from and the size of the area to copy (width x height). The blit engine expects to read a certain number of words. For 15/16 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 blit engine expects is calculated using the following formula.

$$\begin{aligned} \text{nWORDS} \quad &= \text{nPixels} \\ &= \text{Blit Width} \times \text{Blit Height} \end{aligned}$$

For 8 bpp color depths, the formula must take into consideration that the blit engine accepts only WORD accesses and each pixel is one BYTE. The blit 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 Blit 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 blit engine expects for 8 bpp color depths is shown in the following formula.

$$\text{nWORDS} = ((\text{BlitWidth} + 1 + \text{DestinationPhase}) \div 2) \times \text{BlitHeight}$$

Example 19: Read 100 $\times$ 20 pixels at the screen coordinates x = 25, y = 38 and save to system memory. Assume a display of 640x480 at a color depth of 8 bpp.

1. Calculate the source address (upper left corner of the screen blit rectangle), using the formula.

    SourceAddress   = (y $\times$ ScreenStride) + (x $\times$ BytesPerPixel)
                      = (38 $\times$ 640) + (25 $\times$ 1)
                      = 24345
                      = 5F19h

    where:
    BytesPerPixel = 1 for 8 bpp
    BytesPerPixel = 2 for 15/16 bpp
    ScreenStride = DisplayWidthInPixels $\times$ 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 Blit. 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.

    BltMemoryOffset   = ScreenStride $\div$ 2
                        = 640 $\div$ 2
                        = 320
                        = 140h

    REG[10Dh] is set to 01h and REG[10Ch] is set to 40h.

8. Calculate the number of WORDS the blit engine expects to receive.

$$\begin{aligned} \text{nWORDS} \quad &= ((\text{BlitWidth} + 1 + \text{DestinationPhase}) \div 2) \times \text{BlitHeight} \\ &= (100 + 1 + 0) \div 2 \times 20 \\ &= 1000 \\ &= 3E8h \end{aligned}$$

9. Program the BitBLT Destination/Source Linear Select bits for a rectangular blit (BitBLT Destination Linear Select = 0, BitBLT Source Linear Select = 0).

   Start the blit operation **and wait for the blit 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 Blit FIFO, confirm the Blit 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 Blit FIFO.

Table 10-7 Possible Blit 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

Note: The order of register initialization is irrelevant as long as all relevant registers are programmed before the BitBLT is initiated.

10.3 S1D13506 BitBLT Synchronization

A BitBLT operation can only be started if the blit engine is not busy servicing another blit. 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 blit 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. 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 blit 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 blit 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. Program and start the new BitBLT operation.

3. Continue the program (CPU and blit engine work independently).

However, this approach can pose problems if the CPU writes a pixel while the blit engine is running a blit. If the CPU writes the pixel before the BitBLT finishes, the pixel may be overwritten by the blit. 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 S1D13506 BitBLT Known Limitations

The S1D13506 blit engine has the following limitations.

• BitBLT Width must be greater than 0.

• BitBLT Height must be greater than 0.

• The blit 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 15/16 bpp color depths and > 2 for 8 bpp.

11 CRT/TV CONSIDERATIONS

The S1D13506 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 13506CFG.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 13506CFG.EXE, see Chapter3, "2 13506CFG.EXE Configuration Program" on page 3-6.

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.

11.1.1 Generating CRT timings with 13506CFG.EXE

13506CFG.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:

- 640x480 at 60Hz (Input Clock = 25.175 MHz)
- 640x480 at 72Hz (Input Clock = 31.500 MHz)
- 640x480 at 75Hz (Input Clock = 31.500 MHz)
- 640x480 at 85 Hz (Input Clock = 36.000 MHz)
- 800x600 at 56 Hz (Input Clock = 36.000 MHz)
- 800x600 at 60 Hz (Input Clock = 40.000 MHz)

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 outpu 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.

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 14.318 MHz 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 17.734 MHz 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 S1D13506 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 *S1D13506 Hardware Functional Specification*.

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 S1D13506 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 *"S1D13506 Hardware Functional Specification"*.

12 MEDIAPLUG

The S1D13506 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 to obtain the latest S1D13506 MediaPlug drivers for testing purposes.

The MediaPlug interface on the S1D13506 must be enabled to function correctly. To enable the MediaPlug interface, MD13 and MD14 must be high (1) on the rising edge of RESET#. When the MediaPlug interface is enabled, **GPIO2 is controlled by the MediaPlug LCMD register, and the GPIO2 bits in both REG[004h] and REG[008h] have no effect**. Also when the MediaPlug interface is enabled, the camera power (VMPEPWR) is controlled by MA11/GPIO2 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 "*S1D13506 Hardware Functional Specification*".

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. GPIO2 is controlled by this bit when the MediaPlug interface is enabled. Therefore, this bit must be used instead of the GPIO2 control bits in REG[004h] and REG[008h] when programming the external ICD2061A clock chip used on the S5U13506P00C evaluation boards. Writing this bit is necessary only when software needs to control the MA11/GPIO2 pin.

12.2 Considerations

Software can determine if the MediaPlug interface is enabled or disabled by reading the MD Configuration Readback Register (REG[00Dh]) and masking the data with 60h. If the masked result equals 60h, the MediaPlug Interface is enabled.

When the MediaPlug interface is enabled, FPDAT[15:8] are used exclusively for the MediaPlug interface. Therefore, when the MediaPlug interface is enabled, Color 16-bit panels cannot be used without an external multiplexing circuit. For further information on the external circuit required, see the *"S1D13506 Hardware Functional Specification"*.

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 S5U13506P00C evaluation board is used, the clock chip should be programmed to support a valid clock for the MediaPlug interface. The ICD2061A clock chip selects frequencies based on the states of GPIO1 and GPIO2. Since the MediaPlug interface uses the GPIO2 pin for camera power, it is important to program the clock chip for the correct MediaPlug interface frequency when the camera is both on or off (GPIO2 low or high). A HAL function is available which programs the clock chip for the MediaPlug interface.

13 IDENTIFYING THE S1D13506

The S1D13506 can only be identified once the Memory/Register Select bit is set to 0. The steps to identify the S1D13506 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 S1D13506 will return a value of 11h (00010001b).

4. The product code is 4 (000100b based on bits 7-2).

5. The revision code is 1 (01b based on bits 1-0).

14 HARDWARE ABSTRACTION LAYER (HAL)

The HAL is a processor independent programming library designed to help port applications and utilities from one S1D13xx product to another. Epson has provided this library as a result of developing test utilities for the S1D13xx LCD controller products.

The HAL contains functions which are designed to be consistent between S1D13xx products, but as the semiconductor products evolve, so must the HAL; consequently there are some differences between HAL functions for different S1D13xx products.

> **Note:** As the S1D13xx line of products changes, the HAL may change significantly or cease to be a useful tool. Seiko Epson reserves the right to change the functionality of the HAL or discontinue its use if no longer required.

14.1 API for 13506HAL

This section is a description of the HAL library Application Programmers Interface (API). Updates and revisions to the HAL may include new functions not included in the following documentation.

Table 14-1 HAL Functions

Function	Description
Initialization	
seRegisterDevice	Registers the S1D13506 parameters with the HAL. seRegisterDevice MUST be the first HAL function called by an application.
seInitReg	Initializes the registers, LUT, and allocates memory for and assigns.
seGetHalVersion	Returns HAL library version information
seGetId	Identifies the controller by interpreting the revision code register.
seGetLibseVersion	Return version information on the LIBSE libraries (for non-x86 platforms)
General HAL Support:	
seGetInstalledMemorySize	Returns the total size of the display buffer memory.
seGetAvailableMemorySize	Determines the last byte of display memory, before the Dual Panel buffer, available to an application
seGetResolution seGetLcdResolution seGetCrtResolution seGetTvResolution	Retrieve the width and height of the physical display device.
seGetBytesPerScanline seGetLcdBytesPerScanline seGetCrtBytesPerScanline seGetTvBytesPerScanline	Returns the number of bytes in each scanline of the display including non-display size.
seSetSoftwareSuspend	Sets/resets software suspend mode.
seGetSoftwareSuspend	Returns the current software suspend state.
seCheckEndian	Retrieves the "endian-ness" of the host CPU platform.
seGetLcdOrientation	Returns the SwivelView™ orientation of the LCD panel.
seDelay	Delays the given number of seconds before returning.
seDisplayBlank seDisplayLcdBlank seDisplayCrtBlank seDisplayTvBlank	Blank/unblank the display by disabling the FIFO.

Table 14-1 HAL Functions (Continued)

Function	Description
seDisplayEnable seLcdDisplayEnable seCrtDisplayEnable seTvDisplayEnable	Enable/disable the display.
Advanced HAL Functions:	
seBeginHighPriority	Increase thread priority for time critical routines.
seEndHighPriority	Return thread priority to normal.
seSetClock	Set the programmable clock.
Surface Support	
seGetSurfaceDisplayMode	Returns the display surface associated with the active surface.
seGetSurfaceSize	Returns the number of bytes allocated to the active surface.
seGetSurfaceLinearAddress	Returns the linear address of the start of display memory for the active surface.
seGetSurfaceOffsetAddress	Returns the offset from the start of display memory to the start of surface memory.
seAllocLcdSurface seAllocCrtSurface> seAllocTvSurface	Use to manually allocate display buffer memory for a surface.
seFreeSurface	Free any allocated surface memory.
seSetLcdAsActiveSurface seSetCrtAsActiveSurface seSetTvAsActiveSurface	Call one of these function to change the active surface.
Register Access:	
seReadRegByte	Reads one register using a byte access.
seReadRegWord	Reads two registers using a word access.
seReadRegDword	Reads four registers using a dword access.
seWriteRegByte	Writes one register using a byte access.
seWriteRegWord	Writes two registers using a word access.
seWriteRegDword	Writes four registers using a dword access.
Memory Access	
seReadDisplayByte	Reads one byte from display memory.
seReadDisplayWord	Reads one word from display memory.
seReadDisplayDword	Reads one dword from display memory.
seWriteDisplayBytes	Writes one or more bytes to display memory.
seWriteDisplayWords	Writes one or more words to display memory.
seWriteDisplayDwords	Writes one or more dwords to display memory.
Color Manipulation:	
seWriteLutEntry seWriteLcdLutEntry seWriteCrtLutEntry seWriteTvLutEntry	Writes one RGB element to the lookup table.
seReadLutEntry seReadLcdLutEntry seReadCrtLutEntry seReadTvLutEntry	Reads one RGB element from the lookup table.
seWriteLut seWriteLcdLut seWriteCrtLut seWriteTvLut	Write the entire lookup table.

Table 14-1 HAL Functions (Continued)

Function	Description
seReadLut seReadLcdLut seReadCrtLut seReadTvLut	Read the entire lookup table.
seSetBitsPerPixel seSetLcdBitsPerPixel seSetCrtBitsPerPixel seSetTvBitsPerPixel seSetLcdCrtBitsPerPixel seSetLcdTvBitsPerPixel	Sets the color depth. In addition to setting the control bits to set the color depth this action sets a default lookup table for the selected color depth.
Virtual Display	
seVirtInit seLcdVirtInit seCrtVirtInit seTvVirtInit seLcdCrtVirtInit seLcdTvVirtInit	Initialize a surface to hold an image larger than the physical display size.
seVirtPanScroll seVirtPanScroll seVirtPanScroll seVirtPanScroll seVirtPanScroll seVirtPanScroll	Pan (right/left) and Scroll (up/down) the display device over the indicated virtual surface.
Drawing	
seSetPixel seSetLcdPixel seSetCrtPixel seSetTvPixel	Set one pixel at the specified x,y co-ordinate and color.
seGetPixel seGetLcdPixel seGetCrtPixel seGetTvPixel	Returns the color of the pixel at the specified x,y co-ordinate.
seDrawLine seDrawLcdLine seDrawCrtLine seDrawTvLine	Draws a line between two endpoints in the specified color
seDrawRect seDrawLcdRect seDrawCrtRect seDrawTvRect	Draws a rectangle. The rectangle can be outlined or filled.
seDrawCircle seDrawLcdCircle seDrawCrtCircle seDrawTvCircle	Draws a circle of given radius and color at the specified center point.
seDrawEllipse seDrawLcdEllipse seDrawCrtEllipse seDrawTvEllipse	Draws an ellipse centered on a given point with the specified horizontal and vertical radius.
Hardware Cursor	
seInitCursor seInitLcdCursor seInitCrtCursor seInitTvCursor	Prepares the hardware cursor for use.

Table 14-1 HAL Functions (Continued)

Function	Description
seFreeCursor seFreeLcdCursor seFreeCrtCursor seFreeTvCursor	Releases the memory allocated to the hardware cursor by the cursor init function.
seEnableCursor seEnableLcdCursor seEnableCrtCursor seEnableTvCursor	Enable (show) or disable (hide) the hardware cursor.
seGetCursorLinearAddress seGetLcdCursorLinearAddress seGetCrtCursorLinearAddress seGetTvCursorLinearAddress	Returns the linear address of the start of the cursor.
seGetCursorOffsetAddress seGetLcdCursorOffsetAddress seGetCrtCursorOffsetAddress seGetTvCursorOffsetAddress	Returns the offset from the start of display memory to the start of the cursor memory.
seMoveCursor seMoveLcdCursor seMoveCrtCursor seMoveTvCursor	Moves the top-left corner of the hardware cursor to the specified x,y co-ordinates.
seSetCursorColor seSetLcdCursorColor seSetCrtCursorColor seSetTvCursorColor	Allows the application to set the color values for either of the two changeable elements of the hardware cursor.
seSetCursorPixel seSetLcdCursorPixel seSetCrtCursorPixel seSetTvCursorPixel	Set one pixel at the specified x,y co-ordinate within the hardware cursor.
seDrawCursorLine seDrawLcdCursorLine seDrawCrtCursorLine seDrawTvCursorLine	Draws a line between two endpoints within the hardware cursor, in the specified color.
seDrawCursorRect seDrawLcdCursorRect seDrawCrtCursorRect seDrawTvCursorRect	Draws a hollow or filled rectangle within the hardware cursor.
Ink Layer	
seInitInk seInitLcdInk seInitCrtInk seInitTvInk	Prepares the hardware ink layer for use.
seFreeInk seFreeLcdInk seFreeCrtInk seFreeTvInk	Frees memory allocated to the hardware ink layer.
seEnableInk seEnableLcdInk seEnableCrtInk seEnableTvInk	Enable (show) or disable (hide) the hardware ink layer.
seGetInkLinearAddress seGetLcdInkLinearAddress seGetCrtInkLinearAddress seGetTvInkLinearAddress	Returns the linear address of the start of the hardware ink layer.

Table 14-1 HAL Functions (Continued)

Function	Description
seGetInkOffsetAddress seGetLcdInkOffsetAddress seGetCrtInkOffsetAddress seGetTvInkOffsetAddress	Returns the offset from the start of display memory to the start of ink layer memory.
seSetInkColor seSetLcdInkColor seSetCrtInkColor seSetTvInkColor	Allows the application to set the color values for either of the two changeable elements of the ink layer.
seSetInkPixel seSetLcdInkPixel seSetCrtInkPixel seSetTvInkPixel	Set one pixel at the x,y co-ordinate within the ink layer.
seDrawInkLine seDrawLcdInkLine seDrawCrtInkLine seDrawTvInkLine	Draws a line between two endpoints within the hardware ink layer.
seDrawInkRect seDrawLcdInkRect seDrawCrtInkRect seDrawTvInkRect	Draws an outlined or solid rectangle within the hardware ink layer.

14.2 Initialization

Initialization functions are normally the first functions in the HAL library that an application calls. These routine allow the application to learn a little about the controller and to prepare the HAL library for use.

int seRegisterDevice(const LPHAL_STRUC lpHalInfo)

Description:	This function registers the S1D13506 device parameters with the HAL library. The device parameters include such item as address range, register values, desired frame rate, and more which are stored in the HAL_STRUCT structure pointed to by lpHalInfo. Additionally this routine allocates system memory as address space for accessing registers and the display buffer.	
Parameters:	lpHalInfo	A pointer to a HAL_STRUCT structure. This structure must be filled with appropriate values prior to calling seRegisterDevice.
Return Value:	ERR_OK	operation completed with no problems
	ERR_UNKNOWN_DEVICE	The HAL was unable to locate the S1D13506.
	ERR_FAILED	The HAL was unable to map S1D13506 display memory to the host platform.

In addition, on Win32 platforms, the following two error values may be returned:

	ERR_PCI_DRIVER_-	The HAL was unable to locate file S1D13XX.VXD
	NOT_FOUND	
	ERR_PCI_BRIDGE_-	The driver file S1D13XX.VXD was unable to locate the
	ADAPTER_NOT_FOUND	S1D13506.

Note: seRegisterDevice() MUST be called before any other HAL functions.

int seInitReg(unsigned DisplayMode, unsigned Flags)

Description:		This function initializes the S1D13506 registers, the LUT, assigns default surfaces and allocates memory accordingly.
Parameters:	DisplayMode	Set this parameter according to the type of initialization desired. Valid values for DisplayMode are:

	0	Use the values configured by 13506CFG.EXE
	LCD	Initialize for use with an LCD panel.
	CRT	Initialize for use with a monitor.
	TV	Initialize for use with a TV
	LCD | CRT	Initialize for both LCD panel and monitor.
	LCD | TV	Initialize for both LCD panel and TV.

	Flags	Provides additional information about how to perform the initialization. Valid values for Flags are:
	CLEAR_MEM	Zero display memory as part of the initialization
	DISP_BLANK	Blank the display, for aesthetics, during initialization.

Return Value:	ERR_OK	The initialization completed with no problems.
	ERR_FAILED	seInitReg failed to initialize the system correctly.

void seGetHalVersion(const char ** pVersion, const char ** pStatus, const char **pStatus-Revision)

Description:		Retrieves the HAL library version information. By retrieving and displaying the HAL version information along with application version information it is possible to determine at a glance whether the latest version of the software is being run.
Parameters:	pVersion	A pointer to the array to receive the HAL version code.
	pStatus	A pointer to the array to receive the HAL status code
		A "B" designates a beta version of the HAL, a NULL indicates the release version
	pStatusRevision	A pointer to the array to receive the HAL revision status.
Return Value:		The version information is returned as the contents of the pointer arguments. A typical return might be:
		*pVersion == "1.01" (HAL version 1.01)
		*pStatus == "B" (BETA release)
		*pStatusRevision == "5" (fifth update of the beta)

int seGetId(int * pId)

Description: Reads the S1D13506 revision code register to determine the controller product and revision.

Parameters: pId   A pointer to an integer to receive the controller ID. The value returned is an interpreted version of the controller identification.

For the S1D13506 the return values are:

ID_S1D13506_REV0	S1D13506 Test Sample version.
ID_S1D13506_REV1	S1D13506 Production version
ID_UNKNOWN	The HAL was unable to identify the controller.

Return Value:

ERR_OK	The operation completed with no problems.
ERR_UNKNOWN_DEVICE	Return value when pID is ID_UNKNOWN.

Note: seGetId() will disable hardware suspend on x86 platforms, and enables the Memory/ Register Select bit (REG[001h] bit 7) on all platforms.

14.2.1 General HAL Support

This category of HAL functions provide several essential services which do not readily group with other functions.

DWORD seGetInstalledMemorySize(void)

Description: This function returns the size of display buffer memory in bytes.

Memory size is determined during the call to seRegisterDevice() by reading the status of MD6 and MD7. seGetInstalledMemorySize() returns the memory size determined during the HAL initialization.

Parameters: None

Return Value: The return value is the size of the video memory buffer in bytes and will be either 80000h (512kb) or 200000h (2 MB).

DWORD seGetAvailableMemorySize(void)

Description: This function returns an offset to the last byte memory, before the Dual Panel buffer, accessible to an application.

An application can directly access memory from offset zero to the offset returned by this function. On most systems the return value will be the last byte of physical display memory. On systems configured for a dual STN panel the return value will account for the presence of the Dual Panel buffer.

Parameters: None.

Return Value: The return value is an offset to the last byte memory directly accessible to an application.

int seGetResolution(unsigned *Width, unsigned *Height)
void seGetLcdResolution(unsigned *Width, unsigned *Height)
void seGetCrtResolution(unsigned *Width, unsigned *Height)
void seGetTvResolution(unsigned *Width, unsigned *Height)

Description:	These functions return the width and height of the physical display device. Virtual dimensions are not accounted for in the return value.
	seGetResolution() returns the width and height of the active surface. If there is more than one display associated with the surface then precedence is given to the LCD.
	seGetLcdResolution() returns the width and height of the LCD panel. The width and height are adjusted for SwivelView™ orientation.
	seGetCrtResolution() and seGetTvResolution() return the width and height of the display indicated by the function name.
Parameters:	Width A pointer to an unsigned integer which will receive the width, in pixels, for the indicated surface.
	Height A pointer to an unsigned integer which will receive the height, in pixels, for the indicated surface.
Return Value:	seGetResolution() returns one of the following:
	ERR_OK Function completed successfully
	ERR_FAILED Returned when there is not an active display surface.
	seGetLcdResolution(), seGetCrtResolution(), and seGetTvResolution() do not return any value.

unsigned seGetBytesPerScanline(void)
unsigned seGetLcdBytesPerScanline(void)
unsigned seGetTvBytesPerScanline(void)
unsigned seGetCrtBytesPerScanline(void)

Description:	These functions return the number of bytes per scanline, including both displayed and non-displayed for the current video mode.
	seGetBytesPerScanline() returns the number of bytes per scanline for the current active surface.
	seGetLcdBytesPerScanline(), seGetTvBytesPerScanline(), and seGetCrtBytesPerScanline() return the number of bytes per scanline for the surface indicated in the function name.
	To work correctly the S1D13506 registers have been initialized prior to calling any of these routines.
Parameters:	None.
Return Value:	The return value is the "stride" or number of bytes from the first byte of one scanline to the first byte of the next scanline. This value includes both the displayed and the non-displayed bytes on each logical scanline.
	For rotated display modes the return value will be either 1024 (8 bpp) or 2048 (15/16 bpp) to reflect the 1024×1024 virtual area of the rotated memory.

void seSetSoftwareSuspend(BOOL Enable)

Description: This function enables or disables software power suspend.

When software suspend is enabled the S1D13506 reduces power consumption by making the displays inactive and ignoring memory accesses. Disabling software suspend re-enables the video system to full functionality.

Parameters: Enable   Call with Enable set to TRUE to set software power suspend
Call with Enable set to FALSE to disable software power suspend.

Return Value: None.

BOOL seGetSoftwareSuspend(void)

Description: seGetSoftwareSuspend() returns the current state of software power suspend.

Parameters: None.

Return Value: The return value is TRUE if software suspend is enabled. The return FALSE if software suspend is not enabled.

int seCheckEndian(BOOL *ReverseBytes)

Description: This function returns the "endian-ness" of the CPU the application is running on.

Parameters: ReverseBytes   A pointer to boolean value to receive the endian-ness of the system. On return from this function ReverseBytes is FALSE if the CPU is little endian (i.e. Intel). ReverseBytes will be TRUE if the CPU is big-endian (i.e. Motorola)

Return Value: The return value is always ERR_OK.

unsigned seGetLcdOrientation(void)

Description: This function retrieves the SwivelView™ orientation of the LCD display.

The SwivelView™ status is read directly from the S1D13506 registers. Calling this function when the LCD display is not enabled will result in an erroneous return a value.

Note: Only the LCD interface supports SwivelView™. A CRT/TV is always assumed to be in LANDSCAPE mode.

Parameters: None.

Return Value: LANDSCAPE   Not rotated.
ROTATE90   Display is rotated 90 degrees clockwise.
ROTATE180   Display is rotated 180 degrees clockwise.
ROTATE270   Display is rotated 270 degrees.

int seDelay(DWORD Seconds)

Description:		This function, intended for non-Intel platforms, delays for the specified number of seconds then returns to the calling routine. On several evaluation platforms it was not readily apparent where to obtain an accurate source of time delays. seDelay() was the result of the need to delay a specified amount of time on these platforms.

seDelay works by calculating and counting the number of vertical non-display periods in the requested delay time. This implies two conditions for proper operation:
a) The S1D13506 control registers must be configured to correct values.
b) Either the CRT or LCD display interface must be enabled.

On Intel and other supported platforms seDelay() calls the C library time functions to delay the desired amount of time.

Parameters:	Seconds	The number of seconds to delay for.
Return Value:	ERR_OK	Returned by all platforms at the completion of a successful delay.
	ERR_FAILED	Returned by platforms which do not have time functions in their standard C library and either of the operating conditions is violated.

void seDisplayBlank(BOOL Blank)
void seDisplayLcdBlank(BOOL Blank)
void seDisplayCrtBlank(BOOL Blank)
void seDisplayTvBlank(BOOL Blank)

Description: These functions blank the display by disabling the FIFO for the specified surface. Blanking the display is a fast convenient means of temporarily shutting down a display device.

For instance updating the entire display in one write may produce a flashing or tearing effect. If the display is blanked prior to performing the update then the operation is perceived to be smoother and cleaner.

seDisplayBlank() will blank the display associated with the current active surface.

seDisplayLcdBlank(), seDisplayCrtBlank(), and seDisplayTvBlank() blank the display for the surface indicated in the function name.

Parameters:	Blank	Call with Blank set to TRUE to blank the display. Call with Blank set to FALSE to un-blank the display.
Return Value:	None.	

void seDisplayEnable(BOOL Enable)
void seLcdDisplayEnable(BOOL Enable)
void seCrtDisplayEnable(BOOL Enable)
void seTvDisplayEnable(BOOL Enable)

Description:	These functions enable or disable the selected display device.
	seDisplayEnable() enables or disables the display for the active surface.
	seLcdDisplayEnable() enables or disables the LCD display.
	seCrtDisplayEnable() enables or disables the CRT display. seCrtDisplayEnable() will disable CRT/TV PCLK 2X clock and as a side effect will disable TV, if the TV was enabled.
	seTvDisplayEnable() enables or disables the TV display. If the CRT is enabled then seTvDisplayEnable() disables it. When seTvDisplayEnable is called the TV flicker filter is enabled or disabled based on the values saved by the configuration program.
Parameters:	Enable Call with Enable set to TRUE to enable the display device. Call with Enable set to FALSE to disable the device.
Return Value:	None.

14.2.2 Advance HAL Functions

The advanced HAL functions include a level of access that most applications don't will never need to access.

int seBeginHighPriority(void)

Description:	Writing and debugging software under the Windows operating system greatly simplifies the developing process for the S1D13506 evaluation system. One issue which impedes application programming is that of latency. Time critical operations, performance measurement for instance, are not guaranteed any set amount of processor time.
	This function raises the priority of the thread and virtually eliminates the question of latency for programs running on a Windows platform.
Note:	The application should not leave it's thread running in a high priority state for long periods of time. As soon as a time critical operation is complete the application should call seEndHighPriorty().
Parameters:	None.
Return Value:	ERR_OK If the function completes successfully.

int seEndHighPriority(void)

Description:	The function returns the thread priority of the calling application to normal.
	After performing some time critical operation the application should call seEndHighPriority() to return the thread priority to a normal level.
Parameters:	None.
Return Value:	None.

int seSetClock(CLOCKSELECT ClockSelect, FREQINDEX FreqIndex)

Description:	Call seSetClock() to set the clock rate of the programmable clock.	
Parameters:	ClockSelect	The ICD2061A programmable clock chip supports two output clock signals. ClockSelect chooses which of the two output clocks to adjust.
		Valid ClockSelect values are defined by the HAL contents CLKI or CLKI2
	FreqIndex	FreqIndex is an enumerated constant and determines what the output frequency should be.

Valid values for FreqIndex are:

FREQ_6000	6.000 MHz
FREQ_10000	10.000 MHz
FREQ_14318	14.318 MHz
FREQ_17734	17.734 MHz
FREQ_20000	20.000 MHz
FREQ_24000	24.000 MHz
FREQ_25000	25.000 MHz
FREQ_25175	25.175 MHz
FREQ_28318	28.318 MHz
FREQ_30000	30.000 MHz
FREQ_31500	31.500 MHz
FREQ_32000	32.000 MHz
FREQ_33000	33.000 MHz
FREQ_33333	33.333 MHz
FREQ_34000	34.000 MHz
FREQ_35000	35.000 MHz
FREQ_36000	36.000 MHz
FREQ_40000	40.000 MHz
FREQ_49500	49.500 MHz
FREQ_50000	50.000 MHz
FREQ_56250	56.250 MHz
FREQ_65000	65.000 MHz
FREQ_80000	80.000 MHz

Return Value:	ERR_OK	The function completed with no problems.
	ERR_FAILED	seSetClock failed the cause may be an invalid ClockSelect or an invalid frequency index

Note: If seSetClock is called with a ClockSelect of CLKI2 and FreqIndex of FREQ_17734 then the HAL will bypass the programmable clock and select the Feature Clock as the input clock source. This is done with the assumption that the application is setting up for TV output and the Feature Clock oscillator will provide a more stable clock for use with TV. (The feature oscillator must be 17.734 MHz)

14.2.3 Surface Support

The S1D13506 HAL library depends heavily on the concept of surfaces. Through surfaces the HAL tracks memory requirements of the attached display devices, hardware cursor and ink layers, and the Dual Panel buffer.

Surfaces allow the HAL to permit or fail function calls which change the geometry of the S1D13506 display memory. Most HAL functions either allocate surface memory or manipulate a surface that has been allocated.

The functions in this sections allow the application programmer a little greater control over surfaces.

int seGetSurfaceDisplayMode(void)

Description: This function determines the type of display associated with the current active surface.

Parameters: None.

Return Value: The return value indicates the active surface display type. Return values will be one of:

LCD   The LCD panel is the active surface.
CRT   The CRT display is the active surface.
TV   The TV is the active display.

DWORD seGetSurfaceSize(void)

Description: This function returns the number of display memory bytes allocated to the current active surface. The return value does not account for the size for the hardware cursor or ink layer which may be associated with the surface.

Parameters: None.

Return Value: The return value is the number of bytes allocated to the current active surface.

The return value can be 0 if this function is called before initializing and making active a surface.

DWORD seGetSurfaceLinearAddress(void)

Description: This function returns the linear address of the start of memory for the active surface.

Parameters: None.

Return Value: The return value is the linear address to the start of memory for the active surface. A linear address is a 32-bit offset, in CPU address space.

The return value will be NULL if this function is called before a surface has been initialized and made active.

Note: On 16-bit Intel platforms, this function will always return NULL. The PCI memory may be located beyond the addressing range of 16-bit programs so all S1D13506 controller accesses must be made through the HAL.

DWORD seGetSurfaceOffsetAddress(void)

Description:	This function returns the offset, from the first byte of display memory to the first byte of memory associated with the active display surface.
Parameters:	None.
Return Value:	The return value is the offset, in bytes, from the start of display memory to the start of the active surface. An address of 0 indicates the surface starts in the first byte of display buffer memory.

> **Note:** This function also returns 0 if there is no memory allocated to an active surface. You must ensure that memory is allocated before calling seGetSurfaceOffsetAddress().

DWORD seAllocLcdSurface(DWORD Size)
DWORD seAllocCrtSurface(DWORD Size)
DWORD seAllocTvSurface(DWORD Size)

Description:	These functions allocate display buffer memory for a surface. If the surface previously had memory allocated then that memory is first released. Newly allocated memory is not cleared.	
	Call seAllocLcdSurface(), seAllocCrtSurface(), and seAllocTvSurface() to allocate the requested amount of display memory for the indicated surface.	
	These functions allow an application to bypass the automatic surface allocation which occurs when functions such as seInitReg() or seSetBitsPerPixel() are called.	
Parameters:	Size	The size in bytes of the requested memory block.
Return Value:	If the memory allocation succeeds then the return value is the linear address of the allocated memory. If the allocation fails then the return value is 0. A linear address is a 32-bit offset, in CPU address space.	

int seFreeSurface(DWORD LinearAddress)

Description:	This function can be called to free any previously allocated display buffer memory.	
	This function is intended to complement seAllocLcdSurface(), seAllocCrtSurface(), and seAllocTvSurface(). seFreeSurface can be used to free memory allocated for the hardware cursor and ink layer however it is recommended that seFreeCursor() or seFreeInk() be called for these surfaces.	
	After calling one of these functions the application must switch the active surface to one which has memory allocated before calling any drawing functions.	
Parameters:	LinearAddress	A valid linear address. The linear address is a dword returned to the application by any surface allocation call.
Return Value:	ERR_OK	Function completed successfully.
	ERR_FAILED	Function failed.

void seSetLcdAsActiveSurface(void)
void seSetCrtAsActiveSurface(void)
void seSetTvAsActiveSurface(void)

Description:	These functions set the active surface to the display indicated in the function name.
	Before calling one of these surface selection routines, that surface must have been allocated using any of the surface allocation methods.
Parameters:	None.
Return Value:	None.

14.2.4 Register Access

The Register Access functions provide convenient method of accessing the control registers of the S1D13506 controller using byte, word or dword widths.

To reduce the overhead of the function call as much as possible, two steps were taken:

• To gain maximum efficiency on all compilers and platforms, byte and word size arguments are passed between the application and the HAL as unsigned integers. This typically allows a compiler to produce more efficient code for the platform.

• Index alignment for word and dword accesses is not tested. On non-Intel platforms attempting to access a word or dword on a non-aligned boundary may result in a processor trap. It is the responsibility of the caller to ensure that the requested index offset is correctly aligned for the target platform.

unsigned seReadRegByte(DWORD Index)

Description:	This routine reads the register specified by Index and returns the value.	
Parameters:	Index	Offset, in bytes, to the register to read.
Return Value:	The return value is the byte read from the register.	

unsigned seReadRegWord(DWORD Index)

Description:	This routine read two consecutive registers as a word and returns the value.	
Parameters:	Index	Offset to the first register to read.
Return Value:	The return value is the word read from the S1D13506 registers.	

DWORD seReadRegDword(DWORD Index)

Description:	This routine reads four consecutive registers as a dword and returns the value.	
Parameters:	Index	Offset to the first of the four registers to read.
Return Value:	The return value is the dword read from the S1D13506 registers.	

int seWriteRegByte(DWORD Index, unsigned Value)

Description:	This routine writes Value to the register specified by Index.	
Parameters:	Index	Offset to the register to be written
	Value	The value, in the least significant byte, to write to the register
Return Value:	None	

void seWriteRegWord(DWORD Index, unsigned Value)

Description:	This routine writes the word contained in Value to the specified index.	
Parameters:	Index	Offset to the register pair to be written.
	Value	The value, in the least significant word, to write to the registers.
Return Value:	None.	

void seWriteRegDword(DWORD Index, DWORD Value)

Description:	This routine writes the value specified to four registers starting at Index.	
Parameters:	Index	Offset to the register to be written to.
	Value	The dword value to be written to the registers.
Return Value:	None.	

14.2.5 Memory Access

The Memory Access functions provide convenient method of accessing the display memory on an S1D13506 controller using byte, word or dword widths.

To reduce the overhead of these function calls as much as possible, two steps were taken:

- To gain maximum efficiency on all compilers and platforms, byte and word size arguments are passed between the application and the HAL as unsigned integers. This typically allows a compiler to produce more efficient code for the platform.

- Offset alignment for word and dword accesses is not tested. On non-Intel platforms attempting to access a word or dword on a non-aligned boundary may result in a processor trap. It is the responsibility of the caller to ensure that the requested offset is correctly aligned for the target platform.

unsigned seReadDisplayByte(DWORD Offset)

Description:	Reads a byte from the display buffer memory at the specified offset and returns the value.	
Parameters:	Offset	Offset, in bytes, from start of the display buffer to the byte to read.
Return Value:	The return value, in the least significant byte, is the byte read from display memory.	

unsigned seReadDisplayWord(DWORD Offset)

Description:	Reads one word from display buffer memory at the specified offset and returns the value.	
Parameters:	Offset	Offset, in bytes, from start of the display buffer to the word to read.
Return Value:	The return value, in the least significant word, is the word read from display memory.	

DWORD seReadDisplayDword(DWORD Offset)

Description:	Reads one dword from display buffer memory at the specified offset and returns the value.	
Parameters:	Offset	Offset, in bytes, from start of the display buffer to the dword to read.
Return Value:	The DWORD read from display memory.	

void seWriteDisplayBytes(DWORD Offset, unsigned Value, DWORD Count)

Description:	This routine writes one or more bytes to the display buffer at the offset specified by Offset.	
Parameters:	Offset	Offset, in bytes, from start of display memory to the first byte to be written.
	Value	An unsigned integer containing the byte to be written in the least significant byte.
	Count	Number of bytes to write. All bytes will have the same value.
Return Value:	None.	

Note: If ((Offset + Count) > memory size) then this function limits the writes to the end of display memory.

void seWriteDisplayWords(DWORD Offset, unsigned Value, DWORD Count)

Description:	This routine writes one or more words to display memory starting at the specified offset.	
Parameters:	Offset	Offset, in bytes, from the start of display memory to the first word to write.
	Value	An unsigned integer containing the word to written in the least significant word
	Count	Number of words to write. All words will have the same value.
Return Value:	None.	

Note: If ((Offset + (Count * 2)) > memory size) then this function limits the writes to the end of display memory.

void seWriteDisplayDwords(DWORD Offset, DWORD Value, DWORD Count)

Description:	This routine writes one or more dwords to display memory starting at the specified offset.	
Parameters:	Offset	Offset, in bytes, from the start of display memory to the first dword to write.
	Value	The value to be written to display memory.
	Count	Number of dwords to write. All dwords will have the same value.
Return Value:	None.	

> **Note:** If ((Offset + (Count * 4)) > memory size) then this function limits the writes to the end of display memory.

14.2.6 Color Manipulation

The functions in the Color Manipulation section deal with altering the color values in the Look-Up Table directly through the accessor functions and indirectly through the color depth setting functions.

Keep in mind that all lookup table data is contained in the upper nibble of each byte.

void seWriteLutEntry(int Index, BYTE *pRGB)
void seWriteLcdLutEntry(int Index, BYTE *pRGB)
void seWriteCrtLutEntry(int Index, BYTE *pRGB)
void seWriteTvLutEntry(int Index, BYTE *pRGB)

≠**Description:**	These routines write one lookup table entry to the specified index of the lookup table.	
	seWriteLutEntry() writes to the specified index of the current active surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.	
	seWriteLcdLutEntry(), seWriteCrtLutEntry() and seWriteTvLutEntry() modify one entry of the lookup table of the surface indicated in by the function name.	
Parameter:	Index	Offset to the lookup table entry to be modified. (i.e. a 0 will write the first entry and a 255 will write the last lookup table entry)
	pRGB	A pointer to byte array of data to write to the lookup table. The array must consist of three bytes; the first byte contains the red value the second byte contains the green value and the third byte contains the blue value.
Return Value:	None	

void seReadLutEntry(int Index, BYTE *pRGB)
void seReadLcdLutEntry(int Index, BYTE *pRGB)
void seReadCrtLutEntry(int Index, BYTE *pRGB)
void seReadTvLutEntry(int Index, BYTE *pRGB)

≠**Description:**	These routines read one lookup table entry and return the results in the byte array pointed to by pRGB.	
	seReadLutEntry() reads the specified index from the lookup table of the current active surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.	
	seReadLcdLutEntry(), seReadCrtLutEntry(), and seReadTvLutEntry() read one entry form the lookup table for the surface indicated by the function name.	
Parameter:	Index	Offset to the lookup table entry to be read. (i.e. setting index to 2 will return a the value of the third element of the lookup table)
	pRGB	A pointer to an array to receive the lookup table data. The array must be at least three bytes long. On return from this function the first byte of the array will contain the red data, the second byte will contain the green data and the third byte will contain the blue data.
Return Value:	None.	

void seWriteLut(BYTE *pRGB, int Count)
void seWriteLcdLut(BYTE *pRGB, int Count)
void seWriteCrtLut(BYTE *pRGB, int Count)
void seWriteTvLut(BYTE *pRGB, int Count)

≠**Description:**	These routines write one or more lookup table entries starting at offset zero.	
	seWriteLut() modifies *Count* entries in the current active surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.	
	seWriteLcdLut(), seWriteCrtLut(), and seWriteTvLut() modifies the lookup table for the surface indicated in the function name.	
	This routine is intended to allow setting as many lookup table entries as the current color depth allows in call.	
Parameter:	pRGB	A pointer to an array of lookup table entry values to write to the LUT. Each lookup table entry must consist of three bytes. The first byte must contain the red value, the second byte must contain the green value and the third byte must contain the blue value to modify the lookup table with.
	Count	The number of lookup table entries to modify.
Return Value:	None.	

void seReadLut(BYTE *pRGB, int Count)
void seReadLcdLut(BYTE *pRGB, int Count)
void seReadCrtLut(BYTE *pRGB, int Count)
void seReadTvLut(BYTE *pRGB, int Count)

≠**Description:**   This routine reads one or more lookup table entries and returns the result in the array pointed to by pRGB. The read always begins at the first lookup table entry.

seReadLut() reads the first *Count* lookup table entries from the current active surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.

seReadLcdLut(), seReadCrtLut(), and seReadTvLut() read the first *Count* entries from the surface indicated by the function name.

This routine allows reading all the lookup table elements used by the current color depth in one library call.

Parameters: pRGB   A pointer to an array of bytes large enough to hold the requested number of lookup table entries. Each lookup table entry consists of three bytes; the first byte will contain the red data, the second the green data and the third the blue data.

Count   The number of lookup table entries to read.

Return Value: None.

DWORD seSetBitsPerPixel(unsigned BitsPerPixel)
DWORD seSetLcdBitsPerPixel(unsigned BitsPerPixel)
DWORD seSetCrtBitsPerPixel(unsigned BitsPerPixel)
DWORD seSetTvBitsPerPixel(unsigned BitsPerPixel)
DWORD seSetLcdCrtBitsPerPixel(unsigned BitsPerPixel)
DWORD seSetLcdTvBitsPerPixel(unsigned BitsPerPixel)

Description: These functions change the color depth of the display and update the appropriate LUT. Display memory is automatically released and then reallocated as necessary for the display size.

seSetBitsPerPixel() changes the bits-per-pixel mode for the active surface. Memory is reassigned according to the descriptions for each of the following mode sets.

seSetLcdBitsPerPixel() changes the bits-per-pixel mode for the panel display. This function uses the current register settings for SwivelView™ to determine the amount of memory to allocate, and what starting register addresses are required.

Note: seSetLcdBitsPerPixel() frees CRT/TV memory in order to guarantee the LCD image starts at the beginning of display buffer memory.

seSetCrtBitsPerPixel() and seSetTvBitsPerPixel() change the bits-per-pixel mode for the indicated display device. These functions ignore the rotate90 and rotate180 register bits. Memory is allocated only for the landscape mode.

seSetLcdCrtBitsPerPixel() and seSetLcdTvBitsPerPixel() change the color depth for a surface which combines LCD and CRT/TV. Portrait (SwivelView™ 90) is disabled. If the display resolution is not the same for the two displays then memory is allocated based on the larger of the two.

IMPORTANT

When the LCD color depth is changed, memory allocated for the display buffer and ink layer/hardware cursors is freed and the display buffer memory is reassigned. The application must redraw the display and re-initialize the cursor/ink and redraw after calling seSetBitsPerPixel().

seSetLcdCrtBitsPerPixel(), and seSetLcdTvBitsPerPixel() will free all allocated memory for all displays and all ink layers/hardware cursors, then allocate memory only for the display(s) mentioned in the function name. The cursor/ink must be re-initialized and restored after making one of these calls.

If the active surface is the panel then seSetBitsPerPixel() will free all allocated memory for all displays and all ink layers/hardware cursors, then allocate memory ONLY for the active surface (LCD). If the active surface is the CRT or TV, seSetBitsPerPixel() will free memory only for the active surface (CRT or TV), and then reallocate memory for this surface as required.

Parameters: BitsPerPixel   The new color depth. BitsPerPixel can be one of the following: 4, 8, 15, 16.

Return Value: The return value is the thirty-two bit offset to the start of the surface display memory. If there is an error the return value is 0. A linear address is the 32-bit offset, in CPU address space, where the application can directly read or write display memory.

The thirty-two bit address must be converted to a segment:offset for use with a 16-bit Intel platform.

unsigned seGetBitsPerPixel(void)
unsigned seGetLcdBitsPerPixel(void)
unsigned seGetCrtBitsPerPixel(void)
unsigned seGetTvBitsPerPixel(void);

Description: These functions returns the current color depth for a the associated display surface.

seGetBitsPerPixel() returns the color depth for the current active surface.

seGetLcdBitsPerPixel(), seGetCrtBitsPerPixel(), and seGetTvBitsPerPixel() return the color depth for the surface indicated in the function name.

Parameters: None.

Return Value: The color depth of the surface. This value will be 4, 8, 15 or 16.

14.2.7 Virtual Display

int seVirtInit(DWORD Width, DWORD Height)
int seLcdVirtInit(DWORD Width, DWORD Height)
int seCrtVirtInit(DWORD Width, DWORD Height)
int seTvVirtInit(DWORD Width, DWORD Height)
int seLcdCrtVirtInit(DWORD Width, DWORD Height)
int seLcdTvVirtInit(DWORD Width, DWORD Height)

Parameters:	None.
Description:	These functions prepare the S1D13506 for displaying a virtual image.
	"Virtual Image" describes the condition where the image contained in display memory is larger than the physical display. In this situation the display surface is used as a window into the larger display memory area. Panning (right/left) and scrolling (up/down) are used move the display surface in order to view the entire image a portion at a time.
	seVirtInit() prepares the current active surface for virtual image display. Memory is allocated based on width, height and the current color depth.
	seLcdVirtInit initializes and allocates memory for the LCD based on width and height and color depth. If the panel surface is rotated 90 or 270 degrees then the height is limited to a maximum 1024 lines.
	seCrtVirtInit and seTvVirtInit initialize and allocate memory for the given display based on current width and height and color depth.
	seLcdCrtVirtInit and seLcdTvVirtInit initialize and allocate memory for a surface which combines both LCD and CRT/TV. Memory is allocated based on the requirements of the larger of the two surfaces (if different). If the panel surface is rotated 90 or 270 degrees then the height is limited to a maximum of 1024 lines.
	Memory previously allocated for this surface is released then reallocated to the larger size.
Parameters:	Width	The desired virtual width of the display in pixels.

		Width must be a multiple of the number of pixels contained in one word of display memory. At 15/16 bit per pixel Width may be any value. At 8 bit per pixel Width must be a multiple of two and at 4 bit per pixel Width must be a multiple of four.
	Height	The desired virtual height of the display in pixels.
		The HAL performs internal memory management to ensure that all display surfaces and cursor/ink layer have sufficient memory for operation. The Height parameter is required so the HAL can determine the amount of memory the application requires for the virtual image.
Return Value:	ERR_OK	The function completed successfully.
	ERR_HAL_BAD_ARG	The requested virtual dimensions are smaller than the physical display size.
	ERR_NOT_ENOUGH_MEMORY	There is insufficient free display memory to set the requested virtual display size.

void seVirtPanScroll(DWORD x, DWORD y)
void seLcdVirtPanScroll(DWORD x, DWORD y)
void seCrtVirtPanScroll(DWORD x, DWORD y)
void seTvVirtPanScroll(DWORD x, DWORD y)
void seLcdCrtVirtPanScroll(DWORD x, DWORD y)
void seLcdTvVirtPanScroll(DWORD x, DWORD y)

Description: When displaying a virtual image the display surface is smaller than the virtual image contained in display memory. In order to view the entire image the display is treated as a window into the virtual image.

These functions allow an application to pan (right and left) and scroll (up and down) the display over the virtual image.

seVirtPanScroll() will pan and scroll the current active surface.

seLcdVirtPanScroll(), seCrtVirtPanScroll(), seTvVirtPanScroll(), seLcdCrtVirtPanScroll(), and seLcdTvVirtPanScroll() will pan and scroll the surface indicated in the function name.

Parameters: x   The new x offset, in pixels, of the upper left corner of the display.

y   The new y offset, in pixels, of the upper left corner of the display.

Return Value: None.

14.2.8 Drawing

Functions in this category perform primitive drawing on the specified display surface. Supported drawing primitive include pixels, lines, rectangles, ellipses and circles.

void seSetPixel(long x, long y, DWORD Color)
void seSetLcdPixel(long x, long y, DWORD Color)
void seSetCrtPixel(long x, long y, DWORD Color)
void seSetTvPixel(long x, long y, DWORD Color)

Description:	These routines set a pixel at the location x,y with the specified color.
	Use seSetPixel() to set one pixel on the current active surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.
	Use seSetLcdPixel(), seSetCrtPixel(), and seSetTvPixel() to set one pixel on the surface indicated in the function name.
Parameters:	x The X co-ordinate, in pixels, of the pixel to set.
	y The Y co-ordinate, in pixels, of the pixel to set.
	Color Specifies the color to draw the pixel with. Color is interpreted differently at different color depths.
	At 4 and 8 bit per pixel display colors are derived from the lookup table values. The least significant byte of Color forms an index into the lookup table.
	At 15 and 16 bit per pixel the lookup table is bypassed and each word of display memory forms the color to display. At 15 bit per pixel the least significant word directly represents the color to draw the pixel with as a 5-5-5 RGB value with the most significant bit of the word discarded. In 16 bit per pixel display mode the least significant word describes the color to draw the pixel with in 5-6-5 RGB format.
Return Value:	None.

DWORD seGetPixel(long x, long y)
DWORD seGetLcdPixel(long x, long y)
DWORD seGetCrtPixel(long x, long y)
DWORD seGetTvPixel(long x, long y)

Description:	Returns the pixel color at the specified display location
	Use seGetPixel() to read the pixel color at the specified x,y co-ordinates on the current active surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.
	Use seGetLcdPixel(), seGetCrtPixel(), and seGetTvPixel() to read the pixel color at the specified x,y co-ordinate on the display surface referenced in the function name.
Parameters:	x The X co-ordinate, in pixels, of the pixel to read
	y The Y co-ordinate, in pixels, of the pixel to read
Return Value:	The return value is a dword describing the color read at the x,y co-ordinate. Color is interpreted differently at different color depths.
	At 4 and 8 bit per pixel, display colors are derived from the lookup table values. The return value is an index into the lookup table. The red, green and blue components of the color can be determined by reading the lookup table values at the returned index.
	At 15 and 16 bit per pixel the lookup table is bypassed and each word of display memory form the color to display. At 15 bit per pixel the least significant word of the return value directly represents the color of the pixel as a 5-5-5 RGB value with the most significant bit of the word discarded. In 16 bit per pixel display mode the least significant word of the return value describes the color as a 5-6-5 RGB value.

void seDrawLine(long x1, long y1, long x2, long y2, DWORD Color)
void seDrawLcdLine(long x1, long y1, long x2, long y2, DWORD Color)
void seDrawCrtLine(long x1, long y1, long x2, long y2, DWORD Color)
void seDrawTvLine(long x1, long y1, long x2, long y2, DWORD Color)

Description:		These functions draw a line between two points in the specified color.
		Use seDrawLine() to draw a line on the current active surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.
		Use seDrawLcdLine(), seDrawCrtLine(), and seDrawTvLine() to draw a line on the surface referenced by the function name
Parameters:	x1	The X co-ordinate, in pixels, of the first endpoint of the line to be drawn.
	y1	The Y co-ordinate, in pixels, or the first endpoint of the line to be drawn.
	x2	The X co-ordinate, in pixels, of the second endpoint of the line to be drawn.
	y2	The Y co-ordinate, in pixels, of the second endpoint of the line to be drawn.
	Color	Specifies the color to draw the line with. Color is interpreted differently at different color depths.
		At 4 and 8 bit per pixel display colors are derived from the lookup table values. The least significant byte of Color is an index into the lookup table.
		At 15 and 16 bit per pixel the lookup table is bypassed and each word of display memory forms the color to display. At 15 bit per pixel the least significant word directly represents the color to draw the line with as a 5-5-5 RGB value with the most significant bit of the word discarded. In 16 bit per pixel display mode the least significant word describes the color to draw the line with in 5-6-5 RGB format.
Return Value:		None.

void seDrawRect(long x1, long y1, long x2, long y2, DWORD Color, BOOL SolidFill)
void seDrawLcdRect(long x1, long y1, long x2, long y2, DWORD Color, BOOL SolidFill)
void seDrawCrtRect(long x1, long y1, long x2, long y2, DWORD Color, BOOL SolidFill)
void seDrawTvRect(long x1, long y1, long x2, long y2, DWORD Color, BOOL SolidFill)

Description:	These routines draw a rectangle on the screen in the specified color. The rectangle is bounded on the upper left by the co-ordinate x1,y1 and on the lower right by the co-ordinate x2,y2. The SolidFill parameter allows the programmer to select whether to fill the interior or the rectangle or to only draw the border.
	Use seDrawRect() to draw a rectangle on the current active display surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.
	Use seDrawLcdRect(), seDrawCrtRect(), and seDrawTvRect() to draw a rectangle on the display surface indicated by the function name.

Parameters:	x1	The X co-ordinate, in pixels, of the upper left corner of the rectangle.
	y1	The Y co-ordinate, in pixels, of the upper left corner of the rectangle.
	x2	The X co-ordinate, in pixels, of the lower right corner of the rectangle.
	y2	The Y co-ordinate, in pixels, of the lower right corner of the rectangle.
	Color	Specifies the color to draw the line with. Color is interpreted differently at different color depths.
		At 4 and 8 bit per pixel display colors are derived from the lookup table values. The least significant byte of Color is an index into the lookup table.
		At 15 and 16 bit per pixel the lookup table is bypassed and each word of display memory forms the color to display. At 15 bit per pixel the least significant word directly represents the color to draw the line with as a 5-5-5 RGB value with the most significant bit of the word discarded. In 16 bit per pixel display mode the least significant word describes the color to draw the line with in 5-6-5 RGB format.
	SolidFill	A boolean value specifying whether to fill the interior of the rectangle.
		Set to FALSE "0" to draw only the rectangle border. Set to TRUE "non-zero" to instruct this routine to fill the interior of the rectangle.
Return Value:	None	

void seDrawCircle(long xCenter, long yCenter, long Radius, DWORD Color)
void seDrawLcdCircle(long xCenter, long yCenter, long Radius, DWORD Color)
void seDrawCrtCircle(long xCenter, long yCenter, long Radius, DWORD Color)
void seDrawTvCircle(long xCenter, long yCenter, long Radius, DWORD Color)

Description:	These routines draw a circle on the screen in the specified color. The circle is centered at the co-ordinate x,y and is drawn with the specified radius and Color. Circles cannot be solid filled.
	Use seDrawCircle() to draw the circle on the current active display surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.
	Use seDrawLcdCircle(), seDrawCrtCircle(), seDrawTvCircle() draw the circle on the display surface indicated by the function name

Parameters:

x	The X co-ordinate, in pixels, of the center of the circle.
y	The Y co-ordinate, in pixels, or the center of the circle.
Radius	Specifies the radius of the circle in pixels.
Color	Specifying the color to draw the circle. Color is interpreted differently at different color depths.
	At 4 and 8 bit per pixel display colors are derived from the lookup table values. The least significant byte of Color is an index into the lookup table.
	At 15 and 16 bit per pixel the lookup table is bypassed and each word of display memory forms the color to display. At 15 bit per pixel the least significant word directly represents the color to draw the circle with as a 5-5-5 RGB value with the most significant bit of the word discarded. In 16 bit per pixel display mode the least significant word describes the color to draw the circle with in 5-6-5 RGB format.

Return Value: None.

void seDrawEllipse(long xc, long yc, long xr, long yr, DWORD Color)
void seDrawLcdEllipse(long xc, long yc, long xr, long yr, DWORD Color)
void seDrawCrtEllipse(long xc, long yc, long xr, long yr, DWORD Color)
void seDrawTvEllipse(long xc, long yc, long xr, long yr, DWORD Color)

Description: These routines draw an ellipse on the screen in the specified color. The circle is centered at the co-ordinate x,y and is drawn in the specified color with the indicated radius for the x and y axis. Ellipses cannot be solid filled.

Use seDrawEllipse() to draw the circle on the current active display surface. See seSetLcdAsActiveSurface(), seSetCrtAsActiveSurface() and seSetTvAsActiveSurface() for information about changing the active surface.

Use seDrawLcdEllipse(), seDrawCrtEllipse(), seDrawTvEllipse() draw the ellipse on the display surface indicated by the function name

Parameters: xc   The X co-ordinate, in pixels, of the center of the ellipse.

    yc   The Y co-ordinate, in pixels, of the center of the ellipse.

    xr   A long integer specifying the X radius of the ellipse, in pixels.

    yr   A long integer specifying the Y radius of the ellipse, in pixels.

    Color   A dword specifying the color to draw the ellipse. Color is interpreted differently at different color depths.

At 4 and 8 bit per pixel display colors are derived from the lookup table values. The least significant byte of Color is an index into the lookup table.

At 15 and 16 bit per pixel the lookup table is bypassed and each word of display memory forms the color to display. At 15 bit per pixel the least significant word directly represents the color to draw the ellipse with as a 5-5-5 RGB value with the most significant bit of the word discarded. In 16 bit per pixel display mode the least significant word describes the color to draw the circle with in 5-6-5 RGB format.

Return Value: None.

14.2.9 Hardware Cursor

The routines in this section support hardware cursor. Most of the calls look similar to normal drawing calls (i.e. seDrawCursorLine()); however, these calls remove the programmer from having to know the particulars of the cursor memory location, layout and whether portrait mode is enabled.

The same S1D13506 uses the same hardware for both hardware cursor and ink layer which means that only the cursor or the ink layer can be active at any given time.The difference between the hardware cursor and the ink layer is that in cursor mode the image is a maximum of 64x64 pixels and can be moved around the display while in ink layer mode the image is as large as the physical size of the display and is fixed in position. Both the ink layer and hardware cursor have the same number of colors and handle these colors identically.

DWORD seInitCursor(void)
DWORD seInitLcdCursor(void)
DWORD seInitCrtCursor(void)
DWORD seInitTvCursor(void)

Description:	These functions allocate cursor memory, fills the cursor image with a transparent block, and enable the cursor. If memory was previously allocated for the cursor, this memory is first released.
	The S1D13506 supports two independent hardware cursors, one on a panel surface and one on the CRT/TV surface.
	Use seInitCursor() to initialize the cursor for the active surface.
	Use seInitLcdCursor(), seInitCrtCursor(), and seInitTvCursor() initialize the cursor on the display surface indicated in the function name.
Parameters:	None.
Return Value:	The return value is the thirty-two bit offset to the start of the hardware cursor memory. If there is an error the return value is 0.
Note:	On a 16-bit DOS the return value must be converted into a segment:offset before use.

void seFreeCursor(void)
void seFreeLcdCursor(void)
void seFreeCrtCursor(void)
void seFreeTvCursor(void)

Parameters:	None.
Description:	These functions release memory allocated to the hardware cursor by seInitCursor() functions.
	Use seFreeCursor() to free the hardware cursor memory for the current active surface.
	Use seFreeLcdCursor(), seFreeCrtCursor(), and seFreeTvCursor() to free the resources associated with the surface indicated by the function name.
Parameters:	None.
Return Value:	None.

void seEnableCursor(int Enable)
void seEnableLcdCursor(int Enable)
void seEnableCrtCursor(int Enable)
void seEnableTvCursor(int Enable)

Description:	These functions enable or disable the hardware cursor. When enabled the cursor will be visible on the display surface. When disabled the cursor will not be displayed.	
	Call seEnableCursor() to enable/disable the hardware cursor of the active surface.	
	Call seEnableLcdCursor(), seEnableCrtCursor(), and seEnableTvCursor() to enable/disable the hardware cursor for the surface indicated by the function name.	
	Recall that the CRT and TV share the same cursor. Enabling/disabling the cursor for one device will affect the other display as well.	
Parameters:	Enable	A flag indicating whether to enable or disable the hardware cursor.
		Call with Enable set to FALSE "0" to disable the hardware cursor for the surface. Call with Enable set to TRUE "non-0" to enable the hardware cursor for the device.
Return Value:	None.	

DWORD seGetCursorLinearAddress(void)
DWORD seGetLcdCursorLinearAddress(void)
DWORD seGetCrtCursorLinearAddress(void)
DWORD seGetTvCursorLinearAddress(void)

Description:	These routines return address for the hardware cursor through which the application can directly access the cursor memory.
	Call seGetCursorLinearAddress() to retrieve the address of the hardware cursor associated with the current active surface.
	Call seGetLcdCursorLinearAddress(), seGetCrtCursorLinearAddress(), or seGetTvCursorLinearAddress() to retrieve the address of the hardware cursor associated with the display surface indicated by the function name.
Parameters:	None.
Return Value:	The return value is the linear address of the hardware cursor. A linear address is the 32 bit offset in CPU address space where the application can directly read or write the hardware cursor.

DWORD seGetCursorOffsetAddress(void)
DWORD seGetLcdCursorOffsetAddress(void)
DWORD seGetCrtCursorOffsetAddress(void)
DWORD seGetTvCursorOffsetAddress(void)

Description:	These routines return the offset in display memory of the hardware cursor. Using this offset the application can use HAL API calls such as seSetWriteDisplayBytes() to access the hardware cursor image.
	Call seGetCursorOffsetAddress() to get the offset to the hardware cursor associated with the current active surface.
	Call seGetLcdCursorOffsetAddress(), seGetCrtCursorOffsetAddress(), and seGetTvCursorOffsetAddress() to retrieve the offset to the hardware cursor for the surface indicated in the function name.
Parameters:	None.
Return Value:	The return value is the offset, in bytes, from the start of display memory to the start of the hardware cursor.

void seMoveCursor(long x, long y)
void seMoveLcdCursor(long x, long y)
void seMoveCrtCursor(long x, long y)
void seMoveTvCursor(long x, long y)

      Description: These routines are move where the hardware cursor is shown on the display surface.

                         Call seMoveCursor() to move the hardware cursor on the current active surface.

                         Call seMoveLcdCursor(), seMoveCrtCursor(), and seMoveTvCursor() to move the hardware cursor associated with the surface indicated in the function name.

      Parameter: x   The desired display surface X co-ordinate, in pixels, of the upper left corner of the cursor.

                         y   The desired display surface Y co-ordinate, in pixels, of the upper left corner of the cursor.

      Return Value: None.

void seSetCursorColor(int Index, DWORD Color)
void seSetLcdCursorColor(int Index, DWORD Color)
void seSetCrtCursorColor(int Index, DWORD Color)
void seSetTvCursorColor(int Index, DWORD Color)

      Description: These routines allow the user to set the either of the two user definable colors.

                         The hardware cursor can be thought of as a four color image. Two of the colors cannot be changed. Displaying these two colors in a cursor image will always result in transparent and inverse video being displayed.

                         The remaining two colors can be changed.

                         Call seSetCursorColor() to change the cursor colors for the current active surface.

                         Call seSetLcdCursorColor(), seSetCrtCursorColor(), or seSetTvCursorColor() to change the color for the surface associated with the function name.

      Parameters: Index   Specifies which of the two application changeable colors this operation is to affect.

                                       Legal values for Index are 0 and 1.

                         Color   The new color to set as the hardware cursor color.

                                       The color values in the dword are arranged as follows: xxxx xxxx xxxR RRRR xxGG GGGG xxxB BBB

                                       Where x is don't care (set to 0), R is five bits of red intensity, G is six bits of green intensity and B is five bits of blue intensity.

      Return Value: None.

void seSetCursorPixel(long x, long y, DWORD Color)
void seSetLcdCursorPixel(long x, long y, DWORD Color)
void seSetCrtCursorPixel(long x, long y, DWORD Color)
void seSetTvCursorPixel(long x, long y, DWORD Color)

Description:	These functions are intended for drawing in the hardware cursor area a pixel at a time.
	Call seSetCursorPixel() to set a pixel in the cursor associated with the current active surface.
	Call seSetLcdCursorPixel(), seSetCrtCursorPixel(), and seSetTvCursorPixel() to set pixels in the cursor associated with the display surface indicated in the function name.
Parameters:	x The X co-ordinate of the cursor, in pixels, at which to set the pixel color. Valid values for x range from 0 to 63.
	y The Y co-ordinate of the cursor, in pixels, a which to set the pixel color.
	Valid values for y range from 0 to 63.
	Color Specifies which of the four cursor colors set the pixel to. Valid values for Color are:

                                            0 - to set the pixel to the solid color 0
                                            1 - to set the pixel to the solid color 1
                                            2 - to set the pixel to the transparent color
                                            3 - to set the pixel to the inverted color.

Return Value: None.

void seDrawCursorLine(long x1, long y1, long x2, long y2, DWORD Color)
void seDrawLcdCursorLine(long x1, long y1, long x2, long y2, DWORD Color)
void seDrawCrtCursorLine(long x1, long y1, long x2, long y2, DWORD Color)
void seDrawTvCursorLine(long x1, long y1, long x2, long y2, DWORD Color)

Description:	These routines assist in defining the cursor shape by drawing a line in the hardware cursor between the specified points.
	Call seDrawCursorLine() to draw a line in the hardware cursor image associated with the current active surface.
	Call seDrawLcdCursorLine(), seDrawCrtCursorLine(), or seDrawTvCursorLine() to draw a line in the hardware cursor image associated with the display surface indicated in the function name.

Parameter:	x1	Specifies the X co-ordinate of the first endpoint of the line measured in pixels from the left edge of the cursor image.
	x2	Specifies the X co-ordinate of the second endpoint of the line measured in pixels from the left edge of the cursor image.
	y2	Specifies the Y co-ordinate of the second endpoint of the line measured n pixels from the top edge of the cursor image.
	Color	Specifies which of the four cursor colors to draw the line with. Valid values for Color are:

0 - to draw the line in solid color 0
1 - to draw the line in solid color 1
2 - to draw the line in the transparent color
3 - to draw the line in the inverted color.

Return Value: None.

void seDrawCursorRect(long x1, long y1, long x2, long y2, DWORD color, BOOL SolidFlll)
void seDrawLcdCursorRect(long x1, long y1, long x2, long y2, DWORD color, BOOL SolidFill)
void seDrawCrtCursorRect(long x1, long y1, long x2, long y2, DWORD color, BOOL SolidFill)
void seDrawTvCursorRect(long x1, long y1, long x2, long y2, DWORD color, BOOL SolidFill)

Description:	These routines draw rectangles on the hardware cursor surface. The rectangle may be drawn as just a border or as a solid filled area.
	Call seDrawCursorRect() to draw a rectangle in the hardware cursor image associated with the current active surface.
	Call seDrawLcdCursorRect(), seDrawCrtCursorRect(), or seDrawTvCursorRect() to draw a rectangle in the hardware cursor image associated with the display surface indicated by the function name.

Parameter:	x1	The X co-ordinate for the top left corner of the rectangle measured in pixels from the left edge of the cursor image.
	y1	The Y co-ordinate for the top left corner of the rectangle measured in pixels from the top of the cursor image.
	x2	The X co-ordinate for the bottom right corner of the rectangle measured in pixels from the left edge of the cursor image.
	y2	The Y co-ordinate for the bottom right corner of the rectangle measured in pixels from the top edge of the cursor image.
	Color	Specifies which of the four cursor colors to draw the line with. Valid values for Color are:
		0 - to draw the rectangle in solid color 0
		1 - to draw the rectangle in solid color 1
		2 - to draw the rectangle to the transparent color
		3 - to draw the rectangle in the inverted color.
	SolidFill	Flags whether to fill the rectangle as or to only draw the border.
		Set SolidFill to FALSE ("0") to draw only the outline of the rectangle.
		Set SolidFill to TRUE ("1") to fill the interior of the rectangle.

Return Value:	None.

14.2.10 Ink Layer

The functions in this section support the hardware ink layer. These functions are nearly identical to the routines to control the hardware cursor.

The same S1D13506 uses the same hardware for both hardware cursor and ink layer which means that only the cursor or the ink layer can be active at any given time.The difference between the hardware cursor and the ink layer is that in cursor mode the image is a maximum of 64x64 pixels and can be moved around the display while in ink layer mode the image is as large as the physical size of the display and is fixed in position. Both the ink layer and hardware cursor have the same number of colors and handle these colors identically.

DWORD seInitInk(void)
DWORD seInitLcdInk(void)
DWORD seInitCrtInk(void)
DWORD seInitTvInk(void)

Description:	These functions initialize the ink layer for use. The initialization includes: allocating ink layer memory, filling the ink layer image with a transparent color, and enabling the ink layer.
	If memory was previously allocated for the ink layer or a hardware cursor on the surface then this memory is first released.
	Call seInitInk() to initialize the ink layer for the current active surface.
	Call seInitLcdInk(), seInitCrtInk(), and seInitTvInk() to initialize the ink layer for the surface indicated in the display name.
Parameters:	None.
Return Value:	The return value is the thirty-two bit offset in CPU address space to the start of the ink layer memory. If there is an error the return value is 0.
Note:	On 16-bit DOS systems the return value must be converted into a segment:offset before use.

void seFreeInk(void)
void seFreeLcdInk(void)
void seFreeCrtInk(void)
void seFreeTvInk(void)

      Description: These functions release the memory allocations made by the call to the seInitInk() calls.

                     Prior to calling the seFreeInk() functions the application must make a call to seEnableInk() to hide the ink layer.

                     Call seFreeInk() to Free the ink layer memory associated with the current active surface.

                     Call seFreeLcdInk(), seFreeCrtInk(), or seFreeTvInk() to free the ink layer memory associated with the surface indicated in the function name.

      Parameters: None.

      Return Value: None.

void seEnableInk(int Enable)
void seEnableLcdInk(int Enable)
void seEnableCrtInk(int Enable)
void seEnableTvInk(int Enable)

      Description: These functions enable or disable the hardware ink layer. When enabled the ink layer will be visible and when disabled the ink layer will be hidden.

                     Call seEnableInk() to enable/disable the ink layer associated with the current active surface.

                     Call seEnableLcdInk(), seEnableCrtInk(), and seEnableTvInk() to enable/disable the hardware ink layer for the surface indicated by the function name.

                     Recall that the CRT and TV share the same ink layer. Enabling/disabling the ink layer for one device will affect the other display as well.

      Parameters: Enable   A flag indicating whether to enable or disable the ink layer.

                     Set Enable to FALSE (0) to disable the ink layer or set Enable to TRUE (non-0) to enable the ink layer.

      Return Value: None.

DWORD seGetInkLinearAddress(void)
DWORD seGetLcdInkLinearAddress(void)
DWORD seGetCrtInkLinearAddress(void)
DWORD seGetTvInkLinearAddress(void)

Description: These routines return address for the hardware ink layer through which the application can directly access the ink layer memory.

Call seGetInkLinearAddress() to retrieve the address of the ink layer associated with the current active surface.

Call seGetLcdInkLinearAddress(), seGetCrtInkLinearAddress(), or seGetTvInkLinearAddress() to retrieve the address of the ink layer associated with the display surface indicated in the function name.

Parameters: None.

Return Value: The return value is the linear address of the hardware cursor. A linear address is the 32 bit offset in CPU address space where the application can directly read or write the hardware ink layer memory

DWORD seGetInkOffsetAddress(void)
DWORD seGetLcdInkOffsetAddress(void)
DWORD seGetCrtInkOffsetAddress(void)
DWORD seGetTvInkOffsetAddress(void)

Description: These routines return the offset in display memory of the hardware ink layer. Using this offset an application can use HAL API calls such as seSetWriteDisplayBytes() to access the ink layer memory.

Call seGetInkOffsetAddress() to get the offset to the ink layer associated with the current active surface.

Call seGetLcdInkOffsetAddress(), seGetCrtInkOffsetAddress(), and seGetTvInkOffsetAddress() to retrieve the offset to the ink layer for the surface indicated in the function name.

Parameters: None.

Return Value: The return value is the offset, in bytes, from the start of display memory to the start of ink layer memory.

void seSetInkColor(int index, DWORD color)
void seSetLcdInkColor(int index, DWORD color)
void seSetCrtInkColor(int index, DWORD color)
void seSetTvInkColor(int index, DWORD color)

Description: These routines allow the user to set the either of the two user definable hardware ink layer colors.

The hardware ink layer can be thought of as a four color image of which only two colors can be changed. The non-changeable colors will displays as transparent and inverted colors.

Call seSetInkColor() to change the in colors for the current active surface.

Call seSetLcdInkColor(), seSetCrtInkColor(), or seSetTvInkColor() to change the color for the surface indicated by the function name.

Parameters: Index   Index to specifies which of the two changeable colors to access. Valid values for Index are 0 and 1.

Color   The new color to set as the ink layer color.

The color values in the dword are arranged as follows:
xxxx xxxx xxxR RRRR xxGG GGGG xxxB BBB

Where x is don't care (set to 0), R is five bits of red intensity, G is six bits of green intensity and B is five bits of blue intensity.

Return Value: None.

void seSetInkPixel(long x, long y, DWORD color)
void seSetLcdInkPixel(long x, long y, DWORD color)
void seSetCrtInkPixel(long x, long y, DWORD color)
void seSetTvInkPixel(long x, long y, DWORD color)

Description:	These functions are intended for drawing on the hardware ink layer a pixel at a time.
	Call seSetInkPixel() to set a pixel in the ink layer associated with the current active surface.
	Call seSetLcdInkPixel(), seSetCrtInkPixel(), and seSetTvInkPixel() to set pixels in the ink layer associated with the display surface indicated in the function name.
Parameters:	x The X co-ordinate of the ink layer, in pixels, at which to set the pixel color. Valid values for x range from 0 to display width - 1.
	y The Y co-ordinate of the ink layer, in pixels, a which to set the pixel color. Valid values for y range from 0 to display height - 1.
	Color Specifies which of the four cursor colors set the pixel to. Valid values for Color are:

                                             0 - to set the pixel to the solid color 0
                                           1 - to set the pixel to the solid color 1
                                         2 - to set the pixel to the transparent color
                                         3 - to set the pixel to the inverted color.

Return Value:	None.

void seDrawInkLine(long x1, long y1, long x2, long y2, DWORD color)
void seDrawLcdInkLine(long x1, long y1, long x2, long y2, DWORD color)
void seDrawCrtInkLine(long x1, long y1, long x2, long y2, DWORD color)
void seDrawTvInkLine(long x1, long y1, long x2, long y2, DWORD color)

Description:	These routines draw lines on the hardware ink layer between two points in the specified color.	
	Call seDrawInkLine() to draw a line in the hardware ink layer associated with the current active surface.	
	Call seDrawLcdInkLine(), seDrawCrtInkLine(), or seDrawTvInkLine() to draw a line in the hardware ink layer image associated with the display surface indicated in the function name.	
Parameter:	x1	Specifies the X co-ordinate of the first endpoint of the line measured in pixels from the left edge of the display.
	y1	Specifies the Y co-ordinate of the first endpoint of the line measured in pixels from the top edge of the display.
	x2	Specifies the X co-ordinate of the second endpoint of the line measured in pixels from the left edge of the display.
	y2	Specifies the Y co-ordinate of the second endpoint of the line measured in pixels from the top edge of the display.
	Color	Specifies which of the four ink layer colors to draw the line with. Valid values for Color are:
		0 - to draw the line in solid color 0
		1 - to draw the line in solid color 1
		2 - to draw the line in the transparent color
		3 - to draw the line in the inverted color.
Return Value:	None.	

void seDrawInkRect(long x1, long y1, long x2, long y2, DWORD color, BOOL SolidFill)
void seDrawLcdInkRect(long x1, long y1, long x2, long y2, DWORD color, BOOL SolidFill)
void seDrawCrtInkRect(long x1, long y1, long x2, long y2, DWORD color, BOOL SolidFill)
void seDrawTvInkRect(long x1, long y1, long x2, long y2, DWORD color, BOOL SolidFIll)

Description:		These routines draw rectangles on the hardware ink layer surface. The rectangle may be drawn as just a border or as a solid filled area.
		Call seDrawInkRect() to draw a rectangle in the hardware ink layer cursor image associated with the current active surface.
		Call seDrawLcdInkRect(), seDrawCrtInkRect(), or seDrawTvInkRect() to draw a rectangle in the hardware ink layer associated with the display surface indicated by the function name.
Parameter:	x1	The X co-ordinate for the top left corner of the rectangle measured in pixels from the left edge of the display surface.
	y1	The Y co-ordinate for the top left corner of the rectangle measured in pixels from the top edge of the display surface.
	x2	The X co-ordinate for the bottom right corner of the rectangle measured in pixels from the left edge of the display surface.
	y2	The Y co-ordinate for the bottom right corner of the rectangle measured in pixels from the top edge of the display surface.
	Color	Specifies which of the four ink layer colors to draw the line with. Valid values for Color are:
		0 - to draw the rectangle in solid color 0 1 - to draw the rectangle in solid color 1 2 - to draw the rectangle to the transparent color 3 - to draw the rectangle in the inverted color.
	SolidFill	Flags whether to fill the rectangle as or to only draw the border.
		Set SolidFill to FALSE ("0") to draw only the outline of the rectangle. Set SolidFill to TRUE ("1") to fill the interior of the rectangle.
Return Value:	None.	

14.2.11 PCI Support

The S1D13506 utilizes up to two megabytes of display memory. Addressing that much memory on Intel ISA based evaluation platforms is difficult and prone to introducing conflicts. To overcome these obstacles the standard PCI evaluation boar is PCI based.

By placing the S1D13506 evaluation board on a PCI bus the issue of address space is eliminated. In addition 32 bit software can be written and debugged in a Microsoft Windows environment before being ported to an embedded platform.

In order for an application to access the S1D13506 memory and register the following two functions are provided.

DWORD seGetLinearDisplayAddress(void)

Description:	This function returns the linear address for the start of physical display memory on a PCI system
Parameters:	None.
Return Value:	The return value is the linear address of the start of display memory. A linear address is a 32-bit offset, in CPU address space.

DWORD seGetLinearRegAddress(void)

Description:	This function returns the linear address of the start of S1D13506 control registers.
Parameters:	None.
Return Value:	The return value is the linear address of the start of S1D13506 control registers. A linear address is a 32-bit offset, in CPU address space.

14.3 Porting LIBSE to a new target platform

Building Epson applications like a simple HelloApp for a new target platform requires 3 things, the HelloApp code, the 13506HAL library, and a some standard C functions (portable ones are encapsulated in our mini C library LIBSE).


```
                        ┌──────── HelloApp Source code
   HelloApp    ◄────────┼──────── C Library Functions (LIBSE for embedded platforms)
                        └──────── 13506HAL Library
```
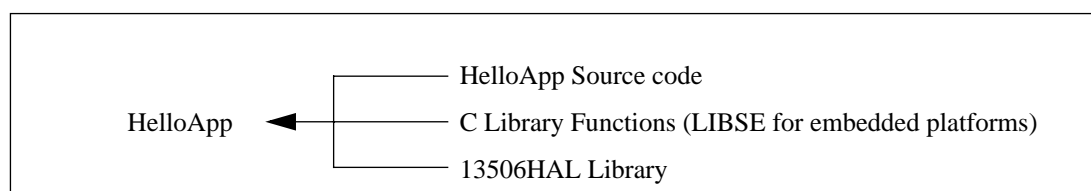
Figure 14-1 Components needed to build 13506 HAL application

For example, when building HELLOAPP.EXE for the x86 16-bit platform, you need the HELLOAPP source files, the 13506HAL library and its include files, and some Standard C library functions (which in this case would be supplied by the compiler as part of its run-time library). As this is a DOS .EXE application, you do not need to supply start-up code that sets up the chip selects or interrupts, etc... What if you wanted to build the application for an SH-3 target, one not running DOS?

Before you can build that application to load onto the target, you need to build a C library for the target that contains enough of the Standard C functions (like sprintf and strcpy) to let you build the application. Epson supplies the LIBSE for this purpose, but your compiler may come with one included. You also need to build the 13506HAL library for the target. This library is the graphics chip dependent portion of the code. Finally, you need to build the final application, linked together with the libraries described earlier. The following examples assume that you have a copy of the complete source code for the S1D13506 utilities, including the nmake makefiles, as well as a copy of the GNU Compiler v2.7-96q3a for Hitachi SH3.

14.3.1 Building the LIBSE library for SH3 target example

In the LIBSE files, there are three main types of files:

- C files that contain the library functions.

- assembler files that contain the target specific code.

- makefiles that describe the build process to construct the library.

The C files are generic to all platforms, although there are some customizations for targets in the form of #ifdef LCEVBSH3 code (the ifdef used for the example SH3 target Low Cost Eval Board SH3). The majority of this code remains constant whichever target you build for.

The assembler files contain some platform setup code (stacks, chip selects) and jumps into the main entry point of the C code that is contained in the C file entry.c. For our example, the assembler file is STARTSH3.S and it performs only some stack setup and a jump into the code at _mainEntry (entry.c).

In the embedded targets, printf (in file rprintf.c), putchar (putchar.c) and getch (kb.c) resolve to serial character input/output. For SH3, much of the detail of handling serial IO is hidden in the monitor of the evaluation board, but in general the primitives are fairly straight forward, providing the ability to get characters to/from the serial port.

For our target example, the nmake makefile is makesh3.mk. This makefile calls the Gnu compiler at a specific location (TOOLDIR), enumerates the list of files that go into the target and builds a .a library file as the output of the build process.

With nmake.exe in your path run:

nmake -fmakesh3.mk

S1D13506F00A
Color LCD/CRT/TV Controller

Utilities

CONTENTS

List of Figures

List of Tables

1 S1D13XX 32-BIT Windows® DEVICE DRIVER INSTALLATION GUIDE

This manual describes the installation of the Windows® 95/98 and Windows® NT device drivers for the S5U13xxB0x series of Epson Evaluation Boards.

The file S1D13XX.VXD is required for using the Epson supplied Intel32 evaluation and test programs for the S1D13xx family of LCD controllers.

1.1 Driver Requirements

Video Controller	:	S1D13xx
Display Type	:	N/A
BIOS	:	N/A
DOS Program	:	No
Dos Version	:	N/A
Windows Program	:	Yes, Windows® 95 / Windows® 98 / Windows® NT 4.0 device driver
Windows DOS Box	:	N/A
Windows Full Screen	:	N/A
OS/2	:	N/A

1.2 Installation

1.2.1 Windows® NT Version 4.0

Note: All evaluation boards require the driver to be installed as described in the following.

1. Install the evaluation board in the computer and boot the computer.

2. Copy the files S1D13XX.INF and S1D13XX.SYS to a directory on a local hard drive.

3. Right click your mouse on the file S1D13XX.INF and select INSTALL from the menu.

4. Windows will install the device driver and ask you to restart.

1.2.2 Windows® 98

All PCI Bus Evaluation Cards

1. Install the evaluation board in the computer and boot the computer.

2. Windows will detect the new hardware as a new PCI Device and bring up the ADD NEW HARDWARE dialog box.

3. Click NEXT

4. Windows will look for the driver. When Windows does not find the driver it will allow you to specify the location of it. Type the driver location or select BROWSE to find it.

5. Click NEXT

6. Windows 98 will locate the specified file and show it as EPSON S1D13XX PCI Bridge Card

7. Click FINISH

All ISA Bus Evaluation Cards

1. Install the evaluation board in the computer and boot the computer.

2. Go to the CONTROL PANEL and select ADD NEW HARDWARE. Windows® 98 will attempt to detect any new plug and play device and fail.

3. When the dialog box DETECT NON-PNP HARDWARE appears, select NO for WINDOWS DETECT and click NEXT.

4. Select OTHER HARDWARE DEVICES from HARDWARE TYPES and click NEXT.

5. Click HAVE DISK.

6. Specify the location of the driver files and click OK.

7. Select EPSON 13XX and Click NEXT.

8. Click FINISH.

Alternative Installation for ISA Bus Evaluation Cards

Copy the files S1D13XX.INF and S1D13XX.SYS to the WINDOWS\SYSTEM directory on your hard drive.

1.2.3 Windows® 95 OSR2

All PCI Bus Evaluation Cards

1. Install the evaluation board in the computer and boot the computer.

2. Windows® will detect the card as a new PCI Device and launch the UPDATE DEVICE DRIVER wizard.

If The Driver is on Floppy Disk

3. Place the disk into drive A: and click NEXT.

4. Windows® will find the EPSON S1D13XX PCI Adapter Card.

5. Click FINISH to install the driver

6. Windows® will ask you to restart the system.

If The Driver is not on Floppy Disk

7. Click NEXT, Windows® will search the floppy drive and fail.

8. Windows® will attempt to load the new hardware as a Standard VGA Card.

9. Click CANCEL. The Driver must be loaded from the CONTROL PANEL under ADD/NEW HARDWARE.

10. Select NO for Windows® to DETECT NEW HARDWARE.

11. Click NEXT.

12. Select OTHER DEVICES from HARDWARE TYPE and Click NEXT.

13. Click HAVE DISK.

14. Specify the location of the driver and click OK

15. Click OK

16. EPSON S1D13XX PCI Bridge Card will appear in the list.

17. Click NEXT

18. Windows® will install the driver

19. Click FINISH

20. Windows® will ask you to restart the system.

21. Windows® will re-detect the card and ask you to restart the system.

All ISA Bus Evaluation Cards

1. Install the evaluation board in the computer and boot the computer.

2. Go to the CONTROL PANEL and select ADD NEW HARDWARE.

3. Click NEXT.

4. Select NO and click NEXT.

5. Select OTHER DEVICES and click NEXT.

6. Click Have Disk.

7. Specify the location of the driver files and click OK.

8. Click Next.

9. Click Finish.

Alternative Installation for ISA Bus Evaluation Cards

Copy the files S1D13XX.INF and S1D13XX.SYS to the WINDOWS\SYSTEM directory on your hard drive.

1.2.4 Previous Versions of Windows® 95

All PCI Bus Evaluation Cards

1. Install the evaluation board in the computer and boot the computer.

2. Windows will detect the card.

3. Select DRIVER FROM DISK PROVIDED BY MANUFACTURER.

4. Click OK

5. Specify a path to the location of the driver files.

6. Click OK

7. Windows® will find the S1D13XX.INF file.

8. Click OK

9. Click OK and Windows® will install the driver

All ISA Bus Evaluation Cards

1. Install the evaluation board in the computer and boot the computer.

2. Go to the CONTROL PANEL and select ADD NEW HARDWARE.

3. Click NEXT.

4. Select NO and click NEXT.

5. Select OTHER DEVICES from the HARDWARE TYPES list.

6. Click HAVE DISK.

7. Specify the location of the driver files and click OK.

8. Select the file S1D13XX.INF and click OK.

9. Click OK.

10. The EPSON 13XX should be selected in the list window.

11. Click NEXT

12. Click NEXT

13. Click Finish.

Alternative Installation for ISA Bus Evaluation Cards

Copy the files S1D13XX.INF and S1D13XX.SYS to the WINDOWS\SYSTEM directory on your hard drive.

2 13506CFG.EXE CONFIGURATION PROGRAM

2.1 13506CFG.EXE

13506CFG.EXE is an interactive Windows® '9x/NT program that calculates the S1D13506 register values for a user-defined LCD/CRT/TV configuration. 13506CFG.EXE can edit and set the configurations of the S1D13506 utilities. Additionally, the program can present the configuration information in several file formats for processing in other programs.

13506CFG.EXE is designed to work with the S1D13506 utilities, or any program designed by a software/hardware developer using the Hardware Abstraction Layer (HAL) library. The configuration information can be saved directly into the utility or into a text header file for use by the software/hardware developer.

> **Note:** Seiko Epson does not assume liability for any damage done to the display device as a result of software configuration errors.

2.2 S1D13506 Supported Evaluation Platforms

13506CFG.EXE only runs on a PC system running Windows® '9x/NT.

13506CFG.EXE can edit the executable files for the following S1D13506 evaluation platforms:

* PC system with an Intel 80x86 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.

2.3 Installation

Copy the file **13506CFG.EXE** to a directory in the path. If running Windows® '9x/NT create a shortcut to the file **13506CFG.EXE**. Copy the file **PANELS.CFG** (which contains some common panel types) to the same directory as **13506CFG.EXE.**

2.4 Usage

In Windows® '9x/NT, double-click the following icon:

1356cfg.exe

Or, at the Windows® DOS Prompt, type `13506CFG.EXE`.

2.5 *13506CFG.EXE Configuration Pages*

13506CFG.EXE provides a series of pages which can be selected by a tab at the top of the main window. The pages are "General", "Defaults", "Memory", "Panel", "CRT/TV", "Clocks", "Registers", and "WinCE". There are also buttons allowing the user to edit and save the configuration of a utility.

The basic procedure for using 13506CFG.EXE is as follows:

1. Use the "Open File" option to load the configuration values from a current utility (this step is optional).

2. Edit the configuration values as required for the specific implementation (see each page description for configuration details).

3. Use the "Save In" option to save the configuration values into the desired utilities, or into an ASCII header file. Each utility must be configured separately.

   Note: 13506CFG.EXE 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.

2.5.1 General Page
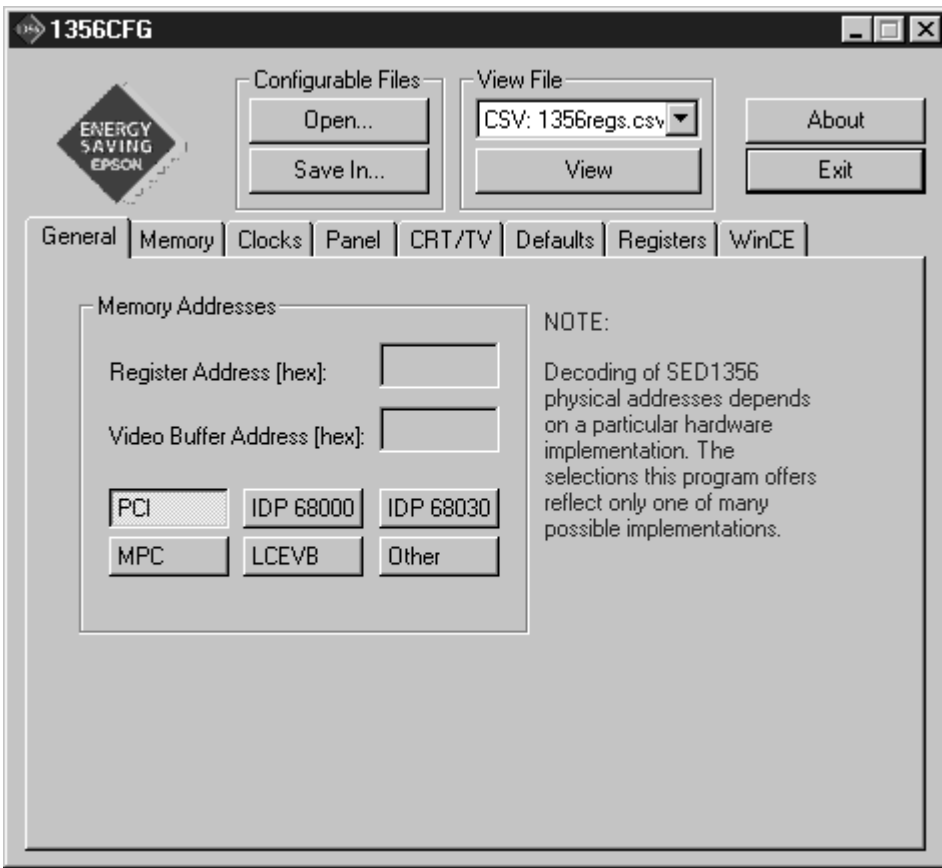
Figure 2-1 General Page

Note: The settings used for MPC, IDP 68000, IDP 68030 and LCEVB platforms are examples of possible implementations and may not reflect your particular hardware implementation.

The General Page allows the user to select the following general platform settings.

General Page	
Register Address	Starting address of the registers (in hexadecimal).
Video Buffer Address	Starting address of the display buffer (in hexadecimal).
Platform Type	Hardware platform to be configured for.

2.5.2 Memory Page
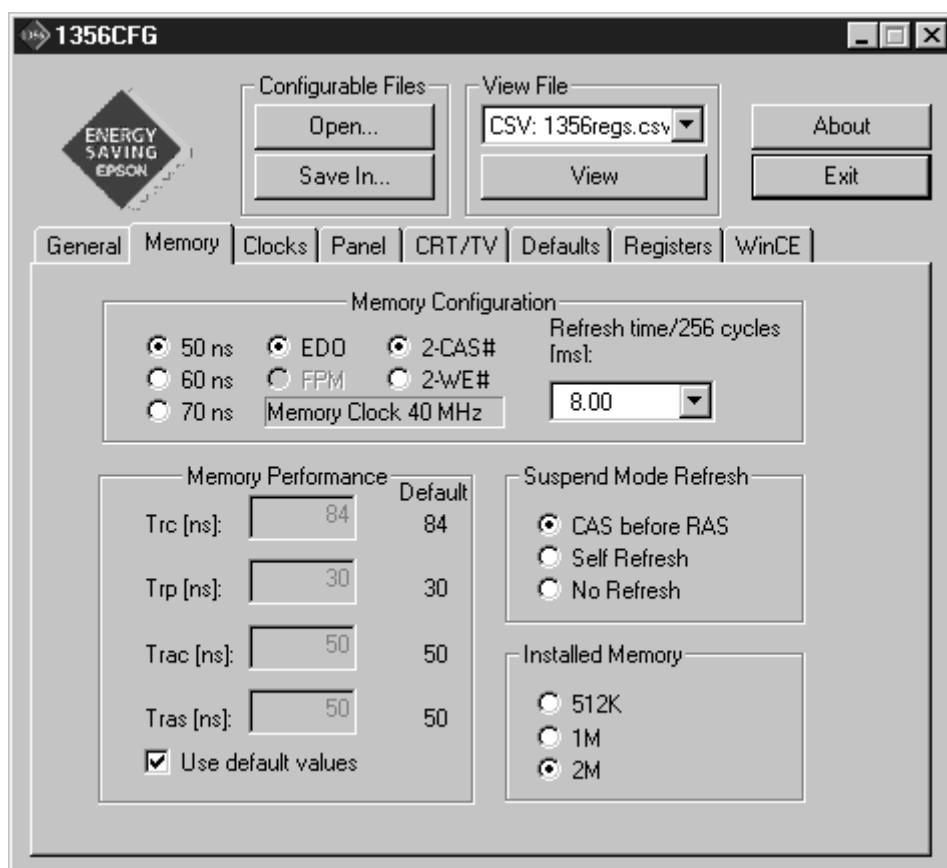
Figure 2-2 Memory Page

The Memory Page allows the user to select the following settings.

Memory Page	
Memory Speed	Access time for memory.
Memory Type	Sets either EDO or FPM memory.
DRAM Type	Selects between 2-CAS# and 2-WE# types of DRAM.
Refresh Time	Selects the correct refresh time in milliseconds.
Memory Performance	Sets the timing parameters for the display buffer. Default values are typical, however, precise values based on memory chip manufacturer specifications should be entered.
Suspend Mode Refresh	Sets the type of refresh during Power Save Mode.
Installed Memory	Selects the total amount of memory available.

2.5.3 Clocks Page
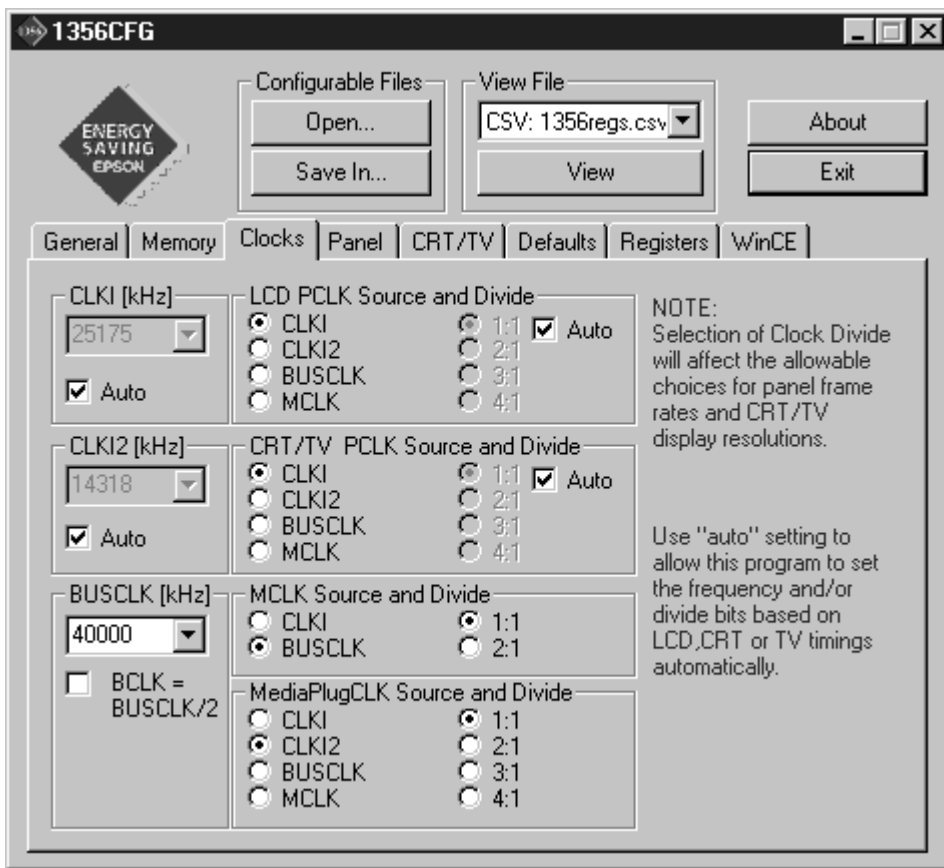
Figure 2-3 Clocks Page

The clocks page allows manual selection of either the clocks or the required timings. From this information 13506CFG.EXE calculates the required timings (if clocks were given) or the required clocks (if timings were given). The program calculates the clocks automatically when the "auto" is selected.

The Clocks Page allows the user to select the following settings.

Clocks Page	
CLKI	Selects the frequency of CLKI.
LCD PCLK Source	Selects the LCD PCLK source.
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 Source	Selects the CRT/TV PCLK source.
CRT/TV Divide	Selects the divide ratio for CRT/TV PCLK. Selecting auto will automatically set the frequency and divide bits.
BUSCLK	Selects the frequency of BUSCLK in KHz.
MCLK Source	Selects the source for MCLK.
MCLK Divide	Selects the divide ratio for MCLK.
MediaPlugCLK Source	Selects the source for MediaPlug clock.
MediaPlugCLK Divide	Selects the divide ratio for the MediaPlug clock.
BCLK = BUSCLK/2	Sets BCLK to the BUSCLK source divided by 2.

2.5.4 Panel Page
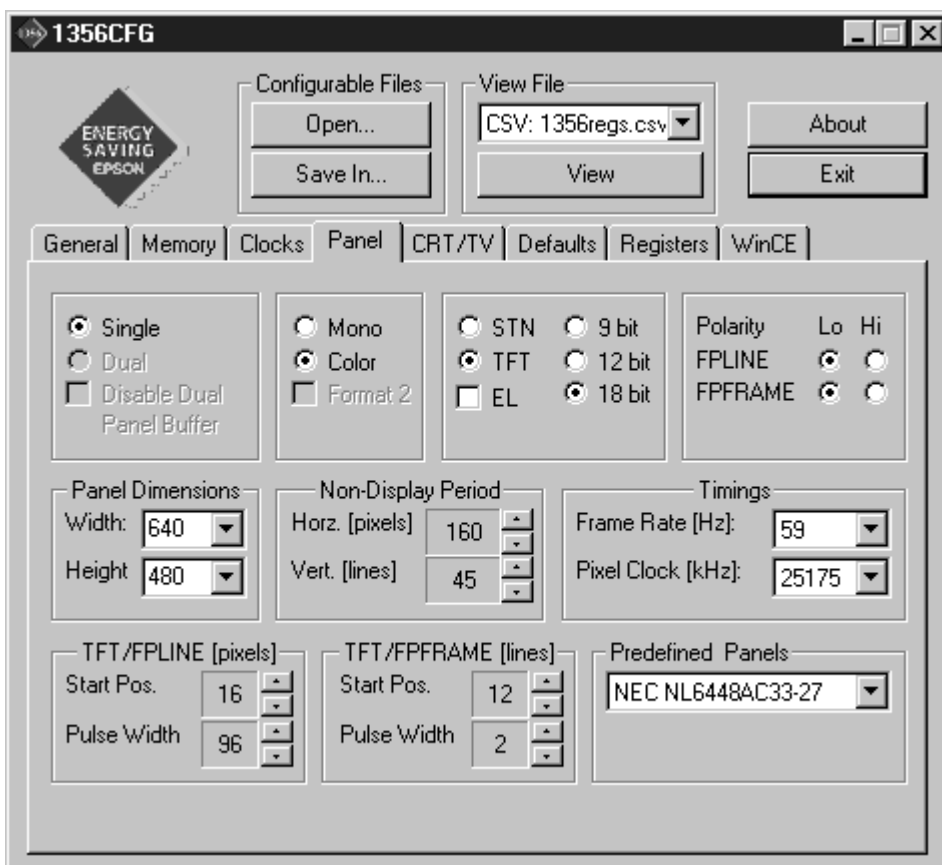
Figure 2-4 Panel Page

This page allows configuration of panel dimensions, type and timings. If the file PANELS.CFG is present in the same directory as 13506CFG.EXE, specific panels can be selected from a list of predefined panels.

The Panel Page allows the user to select the following settings.

Panel Page	
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 *"S1D13506 Hardware Functional Specification"* for format 1/format 2 descriptions.
STN/TFT	Selects between a passive LCD and TFT/D-TFD panel.
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.

2.5.5 CRT/TV Page
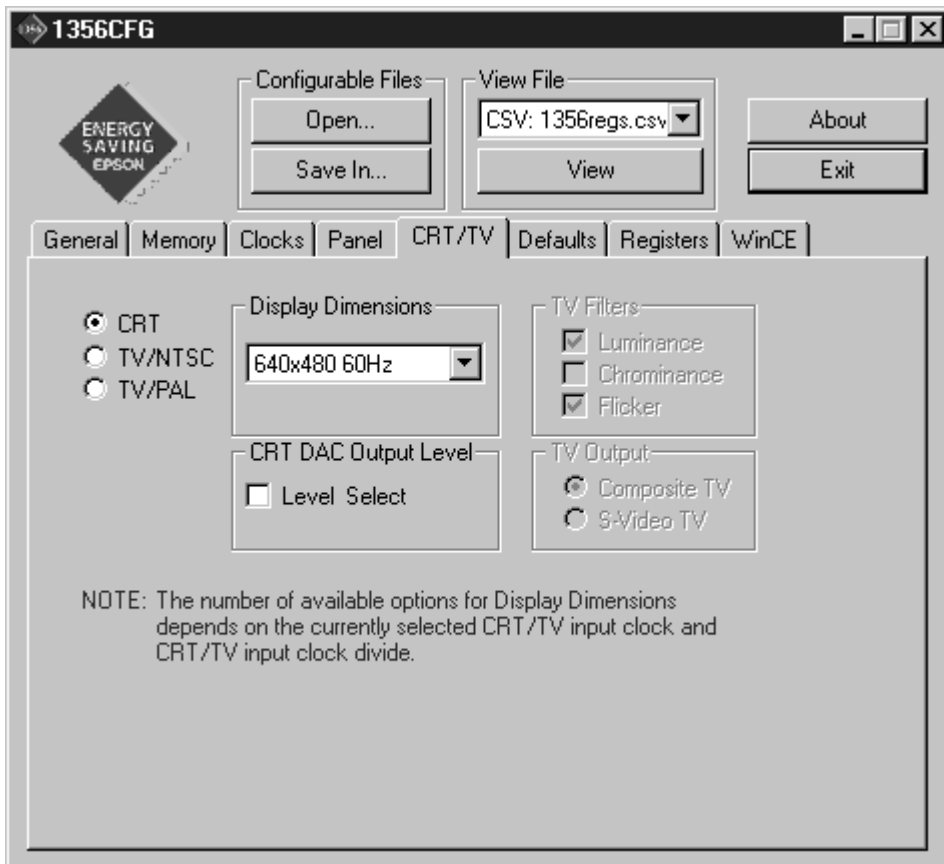
Figure 2-5 CRT/TV Page

The CRT/TV Page allows the user to select the following settings.

CRT/TV Page	
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).

2.5.6 Defaults Page
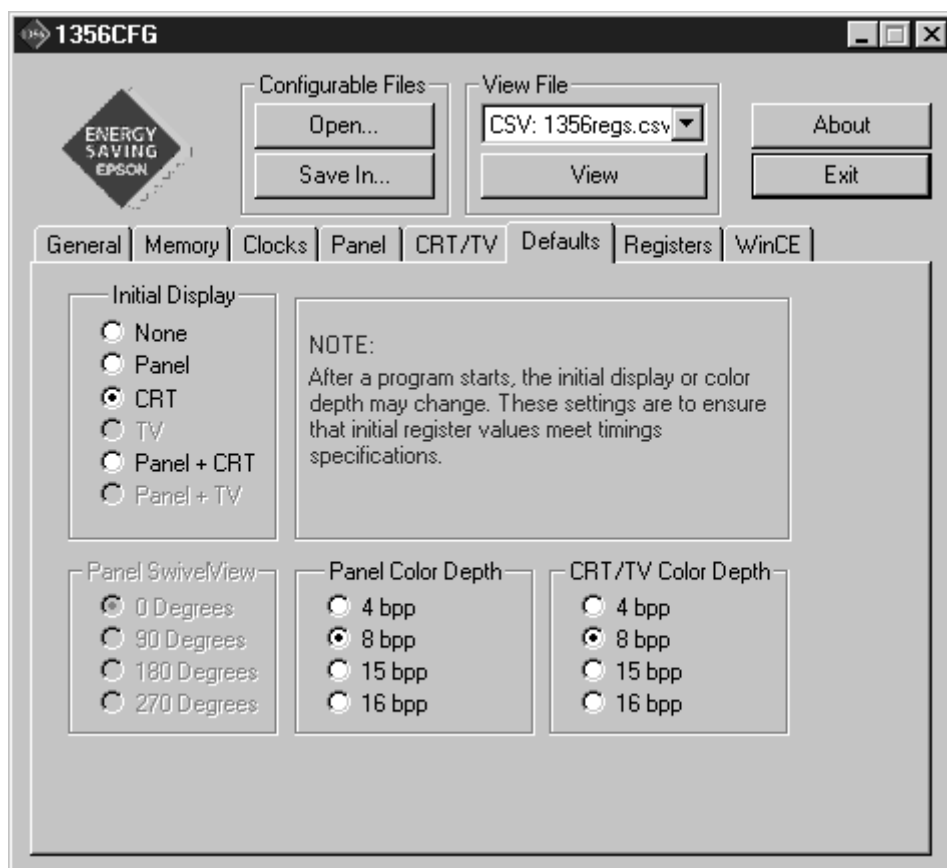
Figure 2-6 Defaults Page

Note: Options that are grayed out are invalid within the current configuration.

The Defaults page allows the user to select the following default settings.

Default Page	
Initial Display	Sets the default display device.
Panel SwivelView™	Sets the SwivelView™ default setting. This option is only available when a LCD panel is selected.
Panel Color Depth	Sets the initial color depth (4/8/15/16 bpp) for the LCD panel.
CRT/TV Color Depth	Sets the initial color depth (4/8/15/16 bpp) for the CRT/TV.

2.5.7 Registers Page
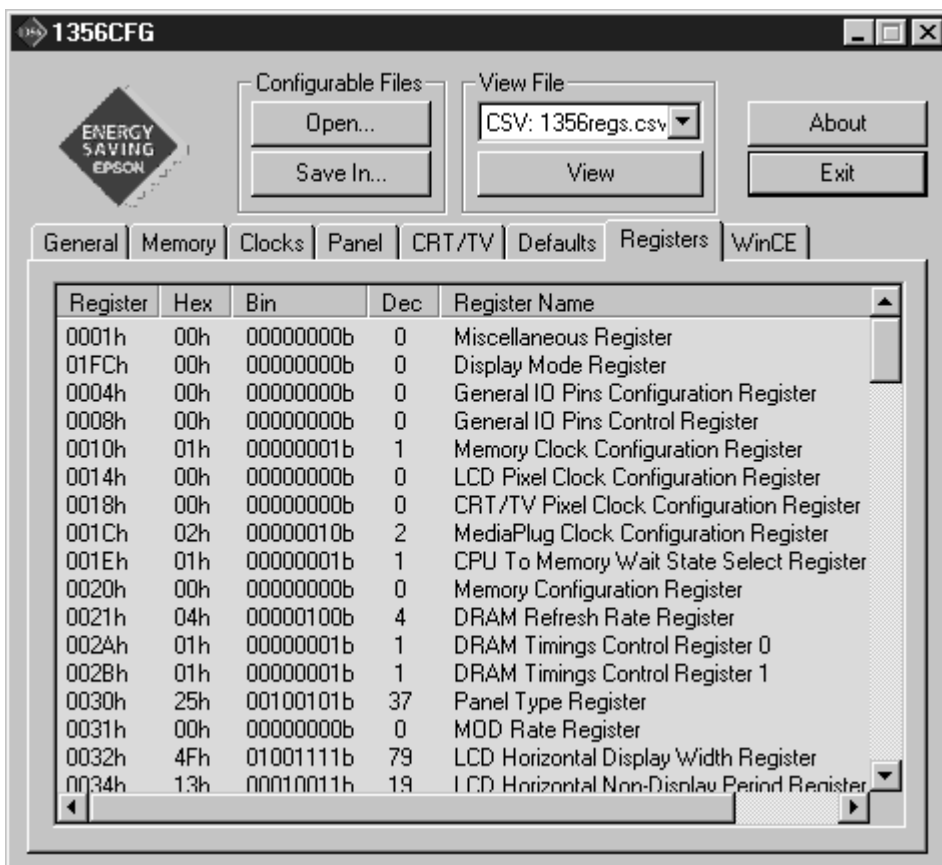
Figure 2-7 Registers Page

The Registers Page lists the register settings that will be 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 further changed by 13506CFG.EXE if further configuration changes are made on the other pages. In this case, the user will be notified.

Note: Manual changes to the registers may have unpredictable results if incorrect values are entered.

2.5.8 WinCE Page

Figure 2-8 WinCE Page

The WinCE Page generates the header files used to write Windows® CE display drivers. Two files are generated: MODE.H and CHIP.H.

WinCE Page	
Cursor	Selects between Hardware Cursor, software cursor and no cursor support.
Hardware Acceleration	Selects whether 2D BitBlt hardware acceleration is used.

2.6 Open File Dialog Box
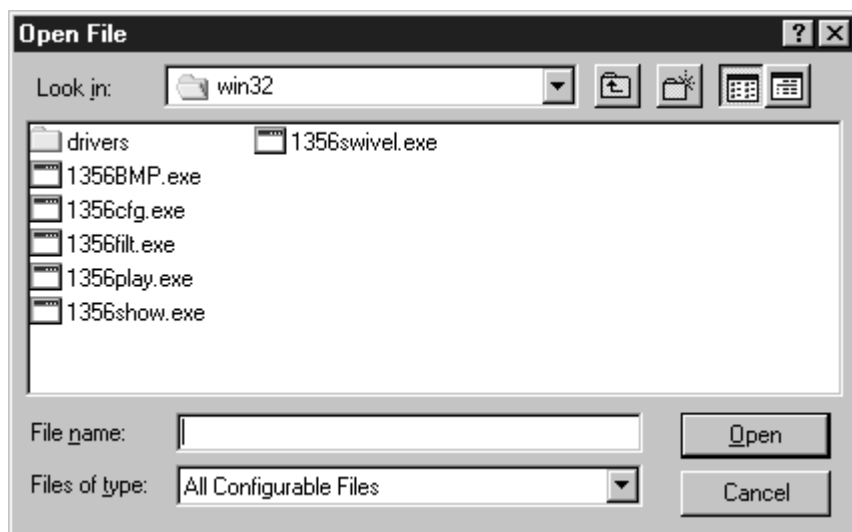
Figure 2-9 Open File Dialog Box

When the "OPEN" button is pressed on the main window, the Open Dialog Box is shown.
13506CFG.EXE will read 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 13506 HAL library.

2.7 Save In Dialog Box
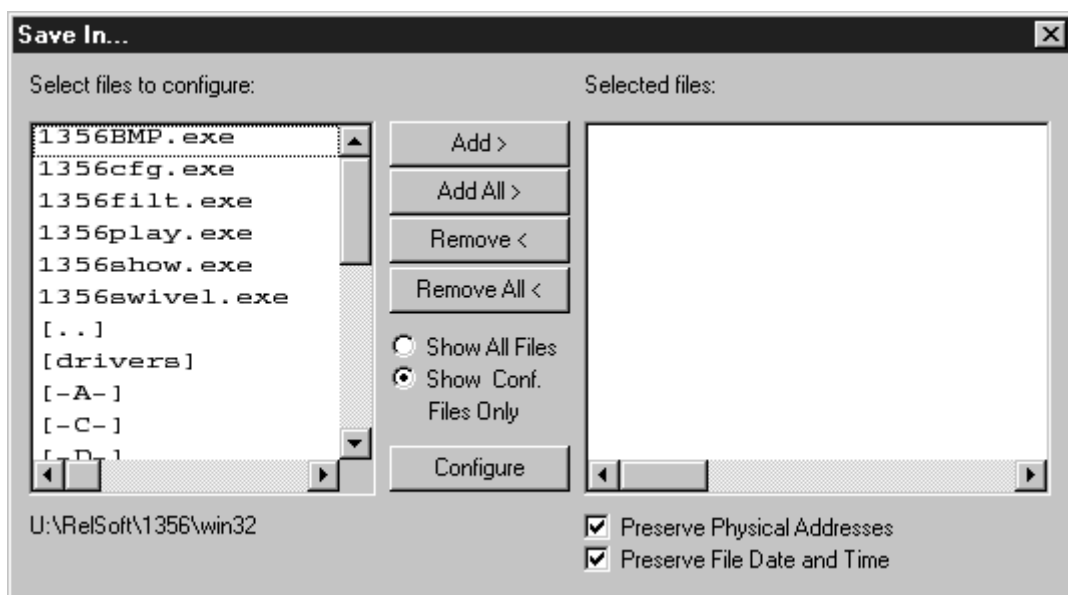
Figure 2-10 Save In Dialog Box

When the "Save In" button is pressed on the main window or a file is dragged & dropped on the 13506CFG.EXE window, the Save In Dialog Box is shown. The left pane shows the files available; the right pane shows the files to be configured. Files are selected to be configured 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 13506 HAL library. The configuration values can also be saved to an ASCII header file (i.e. 13506reg.h) for use by the software/hardware developer. The selected files are configured when the "Configure" button is pressed. 13506CFG.EXE itself can be configured, allowing changes to the default configuration settings. Programs cannot be configured while they are running.

Clicking on "Preserve Physical Addresses" will force the program to retain the Physical Addresses for the display buffer and registers. This means that the addresses specified in the "General" tab will be 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 would be retained.

Clicking on "Preserve File Date and Time" will save the files without changing the date or time stamp of the file.

2.8 Comments

- It is assumed that the user is familiar with the S1D13506 controller and software utilities. For further information on the S1D13506, refer to the "*S1D13506 Functional Hardware Specification"*. For further information on programming the S1D13506, refer to the "*S1D13506 Programming Notes and Examples"*.

- 13506CFG.EXE itself can be configured, allowing changes to the default configuration settings. To configure 13506CFG.EXE, 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 will be provided.

- The file PANELS.CFG is a simple text file containing any supported panels. This file can be edited and will be available to 13506CFG.EXE if stored in the same directory.

- 13506CFG.EXE allows manual entry of values that violate the S1D13506 Hardware Functional Specification 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.

3 13506SHOW.EXE DEMONSTRATION PROGRAM

3.1 13506SHOW.EXE

13506SHOW.EXE is designed to demonstrate and test some of the S1D13506 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. 13506SHOW.EXE supports SwivelView™ (90°, 180°, and 270° hardware rotation of the display image).

The 13506SHOW.EXE demonstration program must be configured and/or compiled to work with your hardware platform. The program 13506CFG.EXE can be used to configure 13506SHOW.EXE. Consult Chapter 3, "2 13506CFG.EXE Configuration Program" on page 3-6, for more information on configuring S1D13506 utilities.

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.

3.2 S1D13506 Supported Evaluation Platforms

13506SHOW.EXE supports the following S1D13506 evaluation platforms:

- PC system with an Intel 80x86 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.

3.3 Installation

PC platform:

Copy the file **13506SHOW.EXE** to a directory specified in the path (e.g. PATH=C:\13506).

Embedded platform:

Download the program **13506SHOW.EXE** to the system.

3.4 Usage

PC Platform

At the prompt, type:

13506SHOW.EXE [/a] [bl=n] [bc=n] [ds=n | ds=?] [/g] [/noinit] [/r90 | /r180 | /r270] [/read] [/s] [/write] [/?]

Embedded platform

Execute **13506SHOW.EXE** 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, 15, 16).
bc=n	shows the CRT/TV display at a user specified color depth (bpp) where n = (4, 8, 15, 16).
ds=n	selects display surfaces (see Section 3.5 "Display Surfaces" on page 1-23).
ds=?	shows the available display surfaces (see Section 3.5 "Display Surfaces" on page 1-23).
/g	shows a grid on the image.
/noinit	skips full register initialization. Only registers used for changing the color depth (bpp), selecting the memory/registers 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	SwivelView™ 90°, clockwise hardware rotation of LCD image by 90 degrees
/r180	SwivelView™ 180°, clockwise hardware rotation of LCD image by 180 degrees
/r270	SwivelView™ 270°, 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.

3.5 Display Surfaces

A surface is a block of memory assigned to one or more physical display devices.
13506SHOW.EXE provides 7 display surfaces (0-6) which cover the possible combinations of
display types. Table 3-1 lists the predefined display surfaces that may be selected.

Table 3-1 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 completely different 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 2MB display buffer.

3.6 13506SHOW.EXE Examples

13506SHOW.EXE is designed to both demonstrate and test some of the features of the S1D13506. The following examples show how to use the program in both instances.

3.6.1 Using 13506SHOW.EXE For Demonstration

1. To show color patterns which must be manually stepped through, type the following:

   13506SHOW.EXE

   The program displays the default color depth and display surface as selected in 13506CFG.EXE. 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 all color depths beginning with the default color depth (as selected in 13506CFG.EXE), type the following:

   13506SHOW.EXE /a

   The program will display the default color depth. 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:

   13506SHOW.EXE bl=[mode]

   where mode = 4, 8, 15, or 16.

   The program displays the requested screen and then exits.

   Note: If a monochrome LCD panel is used, the image is formed using only the green component of the Look-Up Table.

4. To show a color pattern for a specific color depth on the CRT, type the following:

   13506SHOW.EXE bc=[mode]

   where mode = 4, 8, 15, or 16.

   The program display the requested screen and then exits.

5. To show the color patterns in SwivelView™ 90° mode, type the following:

   13506SHOW.EXE /r90

   The program will display the default color depth (as selected in 13506CFG.EXE). Press any key to go to the next screen. Since SwivelView™ 90° is limited to color depths of 8, 15 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.

6. To show solid vertical stripes, type the following:

   13506SHOW.EXE /s

   The program will display the default color depth (as selected in 13506CFG.EXE). 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.

3.6.2 Using 13506SHOW.EXE For Testing

1. To show a test grid over the 8 bpp color pattern on an LCD, type the following:

   13506SHOW.EXE bl=8 /g

   The program will display the 8 bpp color pattern overlaid with a white grid 20 pixels wide and then exit. 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.

2. To test background memory reads to the CRT, type the following:

   13506SHOW.EXE bc=16 /read

   The program will test screen reads. If there is a problem with memory access, the displayed pattern will appear different than when the "/read" switch is not used. If there is a problem, check the configuration parameters of 13506SHOW.EXE using the utility 13506CFG.EXE. See Chapter 3, "2 13506CFG.EXE Configuration Program" on page 3-6, for more information.

   The "/read" switch should be used in combination with the "b=" setting, otherwise the test will always start with the default color depth (as selected in 13506CFG.EXE). 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).

3.7 Comments

- If 13506SHOW.EXE is started without defining the color depth, 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 13506SHOW.EXE using 13506CFG.EXE. This approach avoids showing color depths not supported by a given hardware configuration.

- 13506SHOW.EXE checks if the display(s) selected from the DS= option have been previously configured by 13506CFG.EXE. If these display(s) have not been configured, 13506SHOW.EXE displays an error message. For example, if the system 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.

- 13506SHOW.EXE cannot show a greater color depth than the display allows.

- SwivelView™ (/r90, /r270) is available only for color depths of 8, 15, and 16 bpp.

- SwivelView™ (/r180) is available for color depths of 4, 8, 15, and 16 bpp.

- If the "bl=" or "bc=" options are not used, 13506SHOW.EXE will cycle through all available color depths.

3.8 Program Messages

ERROR: Could not detect S1D13506.
The ID register did not indicate the presence of the 13506.

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 memory.

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 memory for the given configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a half-frame 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.
13506SHOW.EXE is unable to change the color depth due to insufficient display buffer memory. Memory requirements depend on:

- the display resolution(s).

- the bit-per-pixel depth(s).

- whether a half-frame 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 13506CFG.EXE and configure for LCD/CRT/TV.
The program was configured by 13506CFG.EXE 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.

ERROR: b= option cannot be used with /noinit.
The command line options b= and /noinit are contradictory, since b= instructs 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.

4 13506PLAY.EXE DIAGNOSTIC UTILITY

4.1 13506PLAY.EXE

13506PLAY.EXE is a diagnostic utility which allows the user to read/write to all the S1D13506 Registers, Look-Up Tables and Display Buffer. 13506PLAY.EXE 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).

13506PLAY.EXE 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 13506PLAY.EXE diagnostic utility must be configured and/or compiled to work with your hardware platform. The program 13506CFG.EXE can be used to configure 13506PLAY.EXE. Consult Chapter 3, "2 13506CFG.EXE Configuration Program" on page 3-6, for more information on configuring S1D13506 utilities.

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.

4.2 S1D13506 Supported Evaluation Platforms

13506PLAY.EXE supports the following S1D13506 evaluation platforms:

• PC with an Intel 80x86 processor running Windows® '95/'98/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.

4.3 Installation

PC platform

Copy the file **13506PLAY.EXE** to a directory in the path (e.g. PATH=C:\S1D13506).

Embedded platform

Download the program **13506PLAY.EXE** to the system.

4.4 Usage

PC platform

At the prompt, type:

13506PLAY.EXE [/?]

Where:

/?   displays program version information.

Embedded platform

Execute **13506PLAY.EXE** and at the prompt, type the command line argument.

Where:

/?   displays program version information.

4.5 Commands

The following commands are designed to be used from within the 13506PLAY.EXE 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. 13506PLAY.EXE "f 0 1FFFFF AB" q).

CLKI [?] iFreq
Selects a preset clock frequency (MHz) for CLKI. If the "?" option is used, the current CLKI frequency is displayed.
Where:
  ?   Displays a list of available frequencies for CLKI (MHz).
  iFreq   Sets CLKI to a preset frequency (MHz) specified by iFreq.

CLKI2 [?] iFreq
Selects a preset clock frequency (MHz) for CLKI2. If the "?" option is used, the current CLKI2 frequency is displayed.
Where:
  ?   Displays a list of available frequencies for CLKI2 (MHz).
  iFreq   Sets CLKI2 to a preset frequency (MHz) specified by iFreq.

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	Address 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	Address 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	Address 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 S1D13506 registers for 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 4-1 iCrtTv Selection

iCrtTv	Resolution	Display Type
0	640x480	CRT
1	800x600	CRT
2	752x484	TV (NTSC)
3	696x436	TV (NTSC)
4	640x480	TV (NTSC)
5	920x572	TV (PAL)
6	856x518	TV (PAL)
7	800x572	TV (PAL)
8	640x480	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 basic information on the current mode for a given display.
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).

Where:

?	Displays further information on the current mode for a given display.
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).

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).

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).

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).

XA

Reads all the S1D13506 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).

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).

?

Displays the help screen.

4.6 13506PLAY.EXE Example

1. Type **13506PLAY.EXE** 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.

4.7 Scripting

13506PLAY.EXE 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 13506PLAY.EXE command per line. All scripts must end with a "q" (quit) command.

On a PC platform, a typical script command line might be:
 "13506PLAY.EXE < dumpregs.scr > results."

This causes the file "dumpregs.scr" to be interpreted as commands by 13506PLAY.EXE and the results to be sent to the file "results."

Example: Create an ASCII text file that contains the commands i, xa, and q.

; This file initializes the S1D13506 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

4.8 Comments

• All displayed numeric values are considered to be hexadecimal unless identified otherwise. For example, 10 = 10h = 16 decimal; 10t = 10 decimal; 010b = 2 decimal.

• Redirecting commands from a script file (PC platform) allows those commands to be executed as though they were typed.

4.9 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 memory.

ERROR: Not enough display buffer memory.
There was insufficient display buffer memory for the given configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a half-frame 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.
13506PLAY.EXE is unable to change the color depth due to insufficient display buffer memory. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a half-frame 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 13506CFG.EXE and configure for LCD/CRT/TV.
The program was configured by 13506CFG.EXE 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 13506PLAY.EXE 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 memory for the given Hardware Cursor/Ink Layer configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a half-frame 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 13506PLAY.EXE, 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.

5 13506BMP.EXE DEMONSTRATION PROGRAM

5.1 13506BMP.EXE

13506BMP.EXE is a demonstration utility used to show the S1D13506 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. 13506BMP.EXE supports SviwelView™ (90˚, 180˚, and 270˚ hardware rotation of the display image).

13506BMP.EXE is designed to operate on a personal computer (PC) within a 32-bit environment only (Windows '95/'98/NT). Other embedded platforms are not supported due to the possible lack of system memory or structured file system.

The 13506BMP.EXE demonstration utility must be configured and/or compiled to work with your hardware configuration. The program 13506CFG.EXE can be used to configure 13506BMP.EXE. Consult Chapter 4, "2 13506CFG.EXE Configuration Program" on page 3-6, for more information on configuring S1D13506 utilities.

5.2 S1D13506 Supported Evaluation Platforms

13506BMP.EXE supports the following S1D13506 evaluation platforms:

• PC with an Intel 80x86 processor running Windows® '95/'98/NT.

> **Note:** The 13506BMP.EXE source code may be modified by the OEM to support other evaluation platforms.

5.3 Installation

Copy the file **13506BMP.EXE** to a directory in the path (e.g. PATH=C:\S1D13506).

5.4 Usage

At the prompt, type:

13506BMP.EXE bmpfile [ds=n | ds=?] [/noinit] [/r90 | /r180 | /r270] [/v] [/?]

Where:

bmpfile	specifies filename of a windows format bmp image
ds=n	selects display surfaces (see Section 5.5 "Display Surfaces" on page 1-41)
ds=?	shows available display surfaces (see Section 5.5 "Display Surfaces" on page 1-41)
/noinit	skips full register initialization. Only registers used for changing the color depth (bpp), selecting the memory/registers 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	SwivelView™ 90˚, clockwise hardware rotation of LCD image by 90 degrees
/r180	SwivelView™ 180˚, clockwise hardware rotation of LCD image by 180 degrees
/r270	SwivelView™ 270˚, clockwise hardware rotation of LCD image by 270 degrees
/v	verbose mode (provides information about the displayed image)
/?	displays the help message

Note: 13506BMP.EXE will automatically finish execution and return to the prompt.

5.5 *Display Surfaces*

A surface is a block of memory assigned to one or more physical display devices. 13506BMP.EXE provides 7 display surfaces (0-6) which cover the possible combinations of display types. Table 3-1 lists the predefined display surfaces that may be selected.

Table 5-1 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 seperate 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 seperate devices from different memory blocks. This allows two completely different 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 2MB display buffer.

5.6 13506BMP.EXE Examples

To display a bmp image on an LCD, type the following:
13506BMP.EXE bmpfile.bmp ds=0

To display a bmp image on a CRT, type the following:
13506BMP.EXE bmpfile.bmp ds=1

To display a bmp image on an LCD with 90˚ SwivelView™ enabled, type the following:
13506BMP.EXE bmpfile.bmp ds=0 /r90

To display a bmp image on both the LCD and CRT, type the following:
13506BMP.EXE bmpfile.bmp ds=3

To display independent bmp images on the LCD and TV, type the following:
13506BMP.EXE bmpfile.bmp ds=6

5.7 Comments

- 13506BMP.EXE displays only Windows BMP format images.

- A 24-bit true color bitmap will be displayed at a color depth of 16 bit-per-pixel.

- Only the green component of the image will be seen on a monochrome display.

5.8 Program Messages

ERROR: Could not detect S1D13506.
The ID register did not indicate the presence of the S1D13506.

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 memory.

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 landscape mode if both the LCD display and CRT/TV are active.

ERROR: Not enough display buffer memory.
There was insufficient display buffer memory for the given configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a half-frame buffer is required.
- the number of displays active (LCD or LCD and CRT/TV).

ERROR: Not enough memory for LCD/CRT/TV in 4/8/15/16 bits-per-pixel.
13506BMP.EXE is unable to change the color depth due to insufficient display buffer memory. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a half-frame 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 memory for the given configuration. Memory requirements depend on:

- the display resolution(s).
- the bit-per-pixel depth(s).
- whether a half-frame 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 13506CFG.EXE and configure for LCD/CRT/TV.
The program was configured by 13506CFG.EXE 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.

S1D13506F00A
Color LCD/CRT/TV Controller

S5U13506P00C
ISA Bus
Evaluation Board
User's Manual

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

This manual describes the setup and operation of the S5U13506P00C Evaluation Board. The S5U13506P00C is designed as an evaluation platform for the S1D13506 Color LCD/CRT/TV Controller chip.

2 FEATURES

The S5U13506P00C features the following:

- S1D13506 Color LCD/CRT/TV controller chip.
- Headers for connecting to a 3.3V or 5V host bus interface.
- 1Mx16 EDO DRAM.
- Configuration options.
- Adjustable positive LCD bias power supplies from +24V to +40V.
- Adjustable negative LCD bias power supplies from -23V 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-bit 3.3V or 5V TFT/D-TFD LCD panel support.
- Embedded RAMDAC for CRT and TV support.
- WINNOV VideumCam digital camera support at 320×240×256 colors at 30 frames per second.
- Clock synthesizer for maximum clock flexibility.
- Software initiated Power Save Mode.
- Selectable clock source for BUSCLK and CLKI.

3 INSTALLATION AND CONFIGURATION

The S5U13506P00C is designed to support as many platforms as possible. The S5U13506P00C incorporates a DIP switch and several jumpers which allow both evaluation board and S1D13506 LCD controller settings to be configured for a specified evaluation platform.

3.1 Configuration DIP Switches

The S1D13506 LCD controller has 16 configuration inputs (MD[15:0]) which are read on the rising edge of RESET#. Where appropriate, the S5U13506P00C hard-wires some of these configuration inputs, but in order to configure the S1D13506 for multiple host bus interfaces a ten-position DIP switch is required. The following DIP switch settings configure the S1D13506.

Table 3-1 Configuration DIP Switch Settings

Switch	Signal	value of this pin at rising edge of RESET# is used to configure: (1/0)	
		Closed/On=1 (or high)	Open/Off=0 (or low)
S1-1	MD1		
S1-2	MD2	See Table 3-2	
S1-3	MD3		
S1-4	MD4	Little Endian	Big Endian
S1-5	MD5	WAIT# is active high	WAIT# is active low
S1-6	MD10	Reserved	
S1-7	MD11	Alternate Host Bus Interface Selected	Primary Host Bus Interface Selected
S1-8	MD12	BUSCLK input divided by 2	BUSCLK input not divided
S1-9	MD13, MD14	MD13: FPDAT[15:8] is MediaPlug interface; external latches required for 16-bit STN panels. MD14: MA11 is VMPEPWR.	MD13: support 16-bit STN panels directly. MD14: MA11 is GPIO2.
S1-10	MD15	WAIT# is always driven.	WAIT# is tristated when S1D13506 is not selected.

[ ] = Required configuration when used in a PCI environment

Note: MD13 and MD14 are configured using the same switch, for further information see Section 6 "Schematic Diagrams" on page 4-18.

The following table shows the Host Bus Interface options available. The Host Bus Interface chosen will depend on the evaluation platform to be used.

Table 3-2 Host Bus Selection

MD11	MD3	MD2	MD1	Host Bus Interface
0	0	0	0	SH-4/SH-3
0	0	0	1	MC68K Bus 1
0	0	1	0	MC68K Bus 2
0	0	1	1	Generic
0	1	0	0	Reserved
0	1	0	1	MIPS/ISA
0	1	1	0	PowerPC
0	1	1	1	PC Card
1	1	1	1	Philips PR31500/PR31700 / Toshiba TX3912

	= Required configuration when used in a PCI environment

3.2 Configuration Jumpers

The S5U13506P00C has seven jumper blocks which configure various board settings. The jumper positions for each function are shown below.

Table 3-3 Jumper Settings

Jumper	Function	Position 1-2	Position 2-3	Jumper Off
JP1	S1D13506 V_{DD} Selection	3.3V	5V	n/a
JP2	LCD panel signalling	5V	3.3V	n/a
JP3	FPDAT[15:8] function	MediaPlug interface (eight jumpers at 1-2, 3-4, 5-6, 7-8, 9-10, 11-12, 13-14 and 15-16)		16-bit LCD panel MSBs (all jumpers disconnected)
JP4	BUSCLK	Buffered 33MHz from PCI bus	From header	n/a
JP5	GPIO2 to VMPEPWR	Always use this position	n/a	Do not disconnect
JP6	CLKI	From clock synthesizer	From header	n/a
JP7	IREF for CRT/TV DAC	4.6mA for CRT	9.2mA for TV	n/a
JP8	FPDAT[15:8] output	Always use this position	Do not use this position	n/a
JP9	PCI bridge FPGA	Disabled for non-PCI host	n/a	Enabled for PCI host

	= Default configuration

4 TECHNICAL DESCRIPTION

The S5U13506P00C 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-bit), CRT and TV (NTSC and PAL). Additionally, it supports a variety of clock options.

4.1 PCI Bus Support

As a PCI device, the S5U13506P00C has the following characteristics.

- 33MHz bus clock.
- Target with no interrupts.
- Non-cacheable memory read and write.
- 3.3V or 5V PCI signalling.

> **Note:** In a 3.3V PCI system, the S1D13506 **must** be powered at 3.3V by setting jumper JP1. In a 5V PCI system, the S1D13506 may be powered at either 3.3V or 5V.

Although the S1D13506 does not support the PCI bus directly, the S5U13506P00C supports the PCI bus using a Bridge Adapter FPGA. The FPGA translates PCI accesses into PC Card accesses which are then decoded by the S1D13506.

A 4M byte PCI address range is allocated to the S5U13506P00C by the system BIOS. The S1D13506 uses this address range to map the internal registers and the 2M byte display buffer. The following table shows the memory mapping of the PCI address block.

Table 4-1 S1D13506 Memory Mapping onto 4M byte PCI Address Block

PCI Memory Offset	Description	S1D13506 M/R#	S1D13506 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 3F FFFFh	Display Buffer (2M byte)	1	00 0000h to 1F FFFFh

4.1.1 On-Board PCI Configuration Registers

Read-Only Registers

The PCI Bridge Adapter board 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	Meaning
Vendor ID	0h	16 bits	10F4h	S-MOS
Device ID	2h	16 bits	1300h	S1D13XX series
Status	6h	16 bits	400h	Slow DEVSEL# response
Revision ID	8h	8 bits	1	FPGA revision 1
Class Code	9h	24 bits	FF 0000h	Non-standard device
Subsystem Vendor ID	2Ch	16 bits	10F4h	S-MOS
Subsystem ID	2Dh	16 bits	8000h	S5U13506P00C Evaluation Board
Header Type	Eh	8 bits	0	Standard PCI register layout
n/a	Fh-FFh	32 bits	0	Other registers are not supported.

Read/Write Registers

The PCI Bridge Adapter board 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 S5U13506P00C evaluation board is fully PCI compliant and handles the PCI configuration registers automatically.

S5U13506P00C ISA BUS EVALUATION BOARD
USER'S MANUAL (X25B-G-004-01)

4.2 Non-PCI Host Interface Support

The S5U13506P00C is specifically designed to support a standard PCI bus environment (using the PCI Bridge Adapter board). However, the S5U13506P00C can directly support many other Host Bus Interfaces. When the FPGA is disabled (using jumper JP9), headers H1 and H2 provide the necessary IO pins to interface to the Host Bus Interfaces listed in Table 4-4. The S1D13506 power supply must be set to 3.3V or 5V (using jumper JP1) according to the host CPU signalling voltage.

4.2.1 CPU Interface Pin Mapping

The functions of the S1D13506 host interface pins are mapped to each host bus interface according to the following table.

Table 4-4 CPU Interface Pin Mapping

S1D13506 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]	V_{DD}	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*[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#	BI	-CE2	/CARDxCSH	CARDxCSH*
M/R#	External Decode							V_{DD}	V_{DD}
CS#	External Decode							V_{DD}	V_{DD}
BUSCLK	BCLK	CKIO	CLK	CLK	CLK	CLKOUT	CLK	DCLKOUT	DCLKOUT
BS#	V_{DD}	BS#	V_{DD}	AS#	AS#	TS	V_{DD}	V_{DD}	V_{DD}
RD/WR#	RD1#	RD/WR#	V_{DD}	R/W#	R/W#	RD/WR	-CE1	/CARDxCSL	CARDxCSL*
RD#	RD0#	RD#	MEMR#	V_{DD}	SIZ1	TSIZ0	-OE	/RD	RD*
WE0#	WE0#	WE0#	MEMW#	V_{DD}	SIZ0	TSIZ1	-WE	/WE	WE*
WAIT#	WAIT#	RDY# /WAIT#	IOCHRDY	DTACK#	DSACK1#	TA	-WAIT	/ CARDxWAIT	CARDxWAIT*
RESET#	RESET#	RESET#	inverted RESET	RESET#	RESET#	RESET#	inverted RESET	RESET#	PON*

Note: *[1] A0 for these busses is not used internally by the S1D13506.

4.2.2 CPU Bus Connector Pin Mapping

The pinouts for Connector H1 are listed in the following table.

Table 4-5 CPU/BUS Connector (H1) Pinout

Pin No.	Function
1	Connected to DB0 of the S1D13506
2	Connected to DB1 of the S1D13506
3	Connected to DB2 of the S1D13506
4	Connected to DB3 of the S1D13506
5	Ground
6	Ground
7	Connected to DB4 of the S1D13506
8	Connected to DB5 of the S1D13506
9	Connected to DB6 of the S1D13506
10	Connected to DB7 of the S1D13506
11	Ground
12	Ground
13	Connected to DB8 of the S1D13506
14	Connected to DB9 of the S1D13506
15	Connected to DB10 of the S1D13506
16	Connected to DB11 of the S1D13506
17	Ground
18	Ground
19	Connected to DB12 of the S1D13506
20	Connected to DB13 of the S1D13506
21	Connected to DB14 of the S1D13506
22	Connected to DB15 of the S1D13506
23	Connected to RESET# of the S1D13506
24	Ground
25	Ground
26	Ground
27	+12 volt supply, required in non-PCI applications
28	+12 volt supply, required in non-PCI applications
29	Connected to WE0# of the S1D13506
30	Connected to WAIT# of the S1D13506
31	Connected to CS# of the S1D13506
32	Connected to MR# of the S1D13506
33	Connected to WE1# of the S1D13506
34	S1D13506 supply, provided by the S5U13506P00C

EPSON   **S5U13506P00C ISA BUS EVALUATION BOARD**
USER'S MANUAL (X25B-G-004-01)

The pinouts for Connector H2 are listed in the following table.

Table 4-6 CPU/BUS Connector (H2) Pinout

Pin No.	Function
1	Connected to AB0 of the S1D13506
2	Connected to AB1 of the S1D13506
3	Connected to AB2 of the S1D13506
4	Connected to AB3 of the S1D13506
5	Connected to AB4 of the S1D13506
6	Connected to AB5 of the S1D13506
7	Connected to AB6 of the S1D13506
8	Connected to AB7 of the S1D13506
9	Ground
10	Ground
11	Connected to AB8 of the S1D13506
12	Connected to AB9 of the S1D13506
13	Connected to AB10 of the S1D13506
14	Connected to AB11 of the S1D13506
15	Connected to AB12 of the S1D13506
16	Connected to AB13 of the S1D13506
17	Ground
18	Ground
19	Connected to AB14 of the S1D13506
20	Connected to AB15 of the S1D13506
21	Connected to AB16 of the S1D13506
22	Connected to AB17 of the S1D13506
23	Connected to AB18 of the S1D13506
24	Connected to AB19 of the S1D13506
25	Ground
26	Ground
27	+5 volt supply, required in non-PCI applications
28	+5 volt supply, required in non-PCI applications
29	Connected to RD/WR# of the S1D13506
30	Connected to BS# of the S1D13506
31	Connected to S1D13506 BUSCLK if JP4 is in position 2-3
32	Connected to RD# of the S1D13506
33	Connected to AB20 of the S1D13506
34	Connected to S1D13506 CLKI if JP6 is in position 2-3

4.3 LCD Support

The S1D13506 supports 4/8-bit dual and single passive monochrome panels, 4/8/16-bit dual and single passive color panels, and 9/12/18-bit active matrix color TFT/D-TFD panels. All necessary signals are provided on the 40-pin LCD connector (J1). The interface signals are alternated with grounds on the cable to reduce cross-talk and noise. When supporting an 18-bit TFT/D-TFD panel, the S1D13506 can display 64K of a possible 256K colors because only 16 of the18 bits of LCD data are available from the S1D13506. For details, refer to the *"S1D13506 Hardware Functional Specification"*.

For S1D13506 FPDAT[15:0] pin mapping for various types of panel see Table 4-7, "LCD Signal Connector (J1 / J2)" on page 4-11.

4.3.1 LCD Interface Pin Mapping

Table 4-7 LCD Signal Connector (J1 / J2)

S1D13506 Pin Names	Connector Pin No.	Monochrome Passive Panels			Color Passive Panels						Color TFT/D-TFD Panels		
		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
FPDAT0	1		D0	LD0		D0	D0	D0	LD0	LD0	R2	R3	R5
FPDAT1	3		D1	LD1		D1	D1	D1	LD1	LD1	R1	R2	R4
FPDAT2	5		D2	LD2		D2	D2	D2	LD2	LD2	R0	R1	R3
FPDAT3	7		D3	LD3		D3	D3	D3	LD3	LD3	G2	G3	G5
FPDAT4	9	D0	D4	UD0	D0	D4	D4	D4	UD0	UD0	G1	G2	G4
FPDAT5	11	D1	D5	UD1	D1	D5	D5	D5	UD1	UD1	G0	G1	G3
FPDAT6	13	D2	D6	UD2	D2	D6	D6	D6	UD2	UD2	B2	B3	B5
FPDAT7	15	D3	D7	UD3	D3	D7	D7	D7	UD3	UD3	B1	B2	B4
FPDAT8	17							D8		LD4	B0	B1	B3
FPDAT9	19							D9		LD5		R0	R2
FPDAT10	21							D10		LD6			R1
FPDAT11	23							D11		LD7		G0	G2
FPDAT12	25							D12		UD4			G1
FPDAT13	27							D13		UD5			G0
FPDAT14	29							D14		UD6		B0	B2
FPDAT15	31							D15		UD7			B1
FPSHIFT	33	FPSHIFT											
DRDY	35 and 38	MOD				FPSHIFT2	MOD				DRDY		
FPLINE	37	FPSHIFT											
FPFRAME	39	FPFRAME											
GND	2-26 (Even Pins)	GND											
N/C	28	N/C											
VLCD	30	Adjustable -23 to -14V negative LCD bias											
LCDVCC	32	+5V or +3.3V according to JP2											
+12V	34	+12V											
VDDH	36	Adjustable +24 to +40V positive LCD bias											
NC	40	not connected											

☐ = Driven low

4.3.2 Buffered LCD Connector

J1 provides the same LCD panel signals as those directly from S1D13506, but with voltage-adapting buffers which can be set to 3.3V or 5V. Pin 32 on this connector provides power for the LCD panel logic at the same voltage as the buffer power supply.

4.3.3 16-bit Passive Color Panel Support

When the MediaPlug option is enabled, (MD13 and MD14 set to "On", see Table 3-1, "Configuration DIP Switch Settings" on page 4-3) S1D13506 pins FPDAT[15:8] are used for the MediaPlug interface and are not available for panel connection. Instead, S1D13506 pins FPDAT[7:0] are multiplexed for 16-bit panel operation. If the MediaPlug option is selected and 16-bit panel operation is desired, demultiplexing circuitry must be built externally according to the schematic below. Refer to Table 4-7, "LCD Signal Connector (J1 / J2)" on page 4-11 for connector pin mapping.
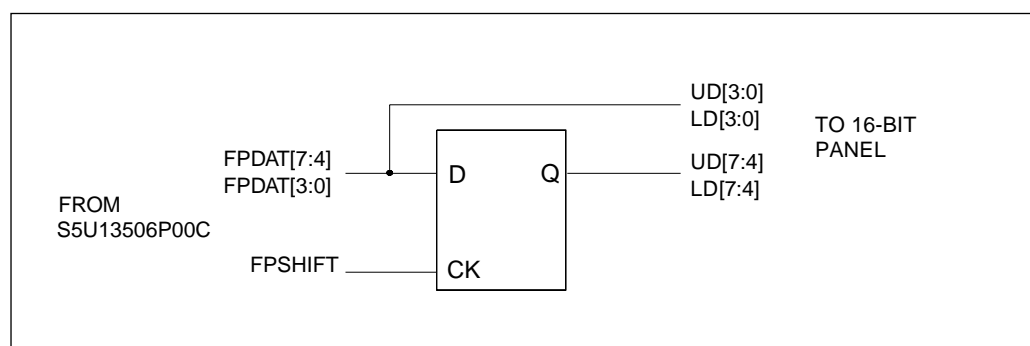

Figure 4-1 External Circuit for 16-bit Panel (MD13/MD14 = 1)

Note: When the MediaPlug option is enabled, the S1D13506 should be powered at 3.3V.

4.3.4 Adjustable LCD Panel Positive Power Supply

For those LCD panels requiring a positive power supply to provide between +24V and +40V (Iout=45mA) a power supply has been provided as an integral part of this design. The VDDH power supply can be adjusted by R15 to provide an output voltage from +24V to +40V and can be enabled and disabled by a GPIO pin.

Note: Determine the panel's specific power requirements and set the potentiometer accordingly before connecting the panel.

4.3.5 Adjustable LCD Panel Negative Power Supply

For those LCD panels requiring a negative power supply to provide between -23V and -14V (Iout=25mA) a power supply has been provided as an integral part of this design. The VLCD power supply can be adjusted by R21 to give an output voltage from -23V to -14V, and can be enabled and disabled by a GPIO pin.

Note: Determine the panel's specific power requirements and set the potentiometer accordingly before connecting the panel.

4.4 CRT/TV Support

4.4.1 CRT/TV Interface Pin Mapping

CRT/TV signals are supplied on a standard CRT connector (J3), Composite Video connector (J2), and S-Video connector (J4):

Table 4-8 CRT/TV Interface Pin Mapping

S1D13506 Pin Name	CRT	Composite Video	S-Video
HRTC	Horizontal retrace	N/A	N/A
VRTC	Vertical retrace	N/A	N/A
RED	Red	N/A	Luminance
GREEN	Green	Composite	N/A
BLUE	Blue	N/A	Chrominance

4.4.2 CRT Support

CRT support is provided on connector J3 via the S1D13506 embedded RAMDAC. An external current reference is implemented to provide the necessary RAMDAC output gain. The reference current should be set to 4.6mA using jumper JP7.

CRT output is not available when TV output is enabled.

4.4.3 TV Support

The S1D13506 supports PAL or NTSC TV output. Composite Video is available on connector J2 and S-Video is available on connector J4. An external current reference is implemented to provide the necessary RAMDAC output gain. The reference current should be set to 9.2mA using jumper JP7.

TV output is not available when CRT output is enabled. PAL and NTSC modes cannot be enabled at the same time.

4.5 MediaPlug Interface

The S5U13506P00C supports the Winnov Videum®Cam digital camera through the S1D13506 built-in MediaPlug interface. The Winnov Videum®Cam digital camera inputs are TTL compatible and can be driven by the S1D13506 powered at 3.3V or 5V. Therefore, the power supply to the camera is 5V while the S1D13506 can powered at 3.3V or 5V. However, if the S1D13506 is powered at 5V, then 150nH inductors must be added at locations L8, L10, L11, L12 and L13.

Jumper JP5 selects whether MA11/GPIO2 is used to enable the video camera power. For more information, see the note from Section 3.2, "Configuration Jumpers". Eight jumpers identified globally as JP3 must be set for MediaPlug operation.

The table below describes the S1D13506 pin mapping for the MediaPlug interface.

Table 4-9 MediaPlug Connector (J5) Pin Mapping

S1D13506 Pin Names	Connector Pin No.	IO Type	MediaPlug I/F
FPDAT8	1	O	VMPLCTL
FPDAT9	2	I	VMPRCTL
FPDAT10	3	IO	VMPD0
FPDAT11	4	IO	VMPD1
FPDAT12	6	IO	VMPD2
FPDAT13	5	IO	VMPD3
FPDAT14	8	O	VMPCLK
FPDAT15	7	O	VMPCLKN
MA11/GPIO2	9	O	VMPEPWR
GND	Shield	-	Ground

4.6 Clock Synthesizer and Clock Options

For maximum flexibility, the S5U13506P00C implements a Cypress ICD2061A programmable clock chip. MCLKOUT from the clock chip is connected to CLKI of the S1D13506 and VCLKOUT from the clock chip is connected to CLKI2 of the S1D13506. A 14.31818MHz crystal (Y1) is connected to XTALIN of the clock chip and a 17.734475MHz oscillator (U14) is connected to the FEATCLK input of the clock chip. The diagram below shows a simplified representation of the clock synthesizer connections.
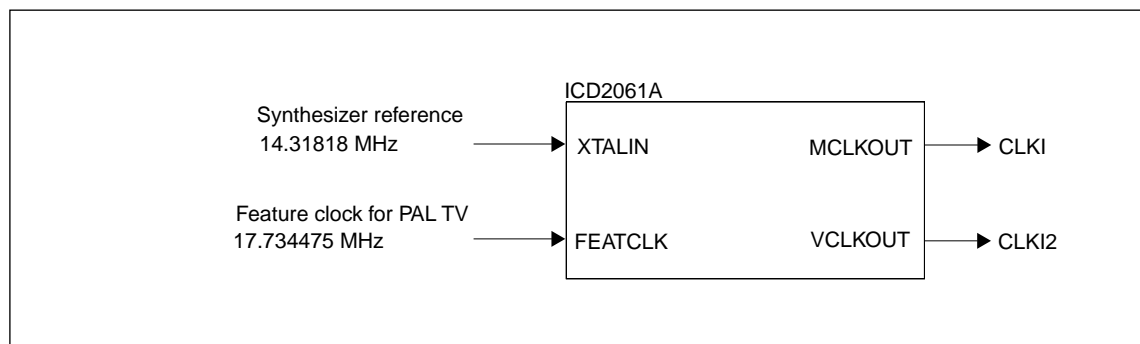

Figure 4-2 Symbolic Clock Synthesizer Connections

Upon power-up, CLKI (MCLKOUT) is 40MHz and CLKI2 (VCLKOUT) is configured to 25.175MHz.

The utility 13506PLAY.EXE (see *"13506PLAY.EXE Diagnostic Utility"*) can be used to program the clock chip to selected frequencies. When 13506PLAY.EXE is required to set CLKI2 for PAL TV output (to 17.734475MHz), the ICD2061A is set to propagate the FEATCLK input instead of relying on the clock synthesizer to obtain the correct frequency. This ensures the greatest possible frequency accuracy for TV applications.

For manual ICD2061A programming, refer to the Cypress ICD2061A specification and to the S5U13506P00C schematics for details.

5 PARTS LIST

Table 5-1 Parts List

Item	Quantity	Reference	Part	Description
1	22	C1-C6,C10,C13-C16,C32-C34, C46-C49,C52,C54,C55,C57	0.1uF	1206 capacitor +/-20% 50V
2	4	C7,C8,C9,C38	0.01uF	1206 capacitor +/-20% 50V
3	6	C11,C12,C40,C45,C53,C56	10uF/16V	Tantalum size C, 10uF 16V +/-10%
4	2	C17,C21	47uF/10V	Tantalum size C, 47uF 10V +/-10%
5	3	C18,C19,C20	4.7uF/50V	Tantalum size D, 4.7uF 50V +/-10%
6	1	C22	56uF/35V	Low-ESR radial electrolytic capacitor
7	9	C23,C24,C25,C26,C27,C28,C29,C30,C31	0.22uF	1206 capacitor +/-20% 50V
8	6	C35,C36,C37,C39,C41,C42	33pF	1206 capacitor +/-20% 50V
9	2	C43,C44	220pF	1206 capacitor +/-20% 50V
10	2	C50,C51	tbd	1206 capacitor, not populated
11	4	D1,D2,D3,D5	BAV99L	Diode BAV99L
12	1	D4	BAT54	Diode BAT54
13	2	H1,H2	Header 17X2	0.1" x 0.1" 2 rows by 17 header
14	2	JP5,JP9	Header 2	0.1" 1 row by 2 header
15	6	JP1,JP2,JP4,JP6,JP7,JP8	Header 3	0.1" 1 row by 3 header
16	1	JP3	Header 8X2	Not populated
17	1	J1	Header 20x2	2x20, .025" sq. shrouded connector, ctr-key, t/h
18	1	J2	C-Video	Keystone 901 or equivalent
19	1	J3	CRT	AMP 749264 or equivalent
20	1	J4	S-Video	Assman A-HDF 15 A KG/T or equivalent
21	1	J5	MediaPlug Connector	CUI Stack P/N:MD-90S or Digi-Key P/N:CP-2490-ND
22	8	L1,L2,L3,L5,L6,L7,L14,L15	Ferrite	Philips BDS3/3/8.9-4S2
23	1	L4	1uH	RCD MCI-1812 1uH MT or MSI-1812 1uH MT
24	1	L9	150nH	Panasonic ELJNCR15JF or Delevan 1008-151K
25	3	Q1,Q5,Q6	MMBT2222A	Transistor MMBT2222A
26	1	Q2	MMBT3906	Transistor MMBT3906
27	1	Q3	MMBT3904	Transistor MMBT3904
28	1	Q4	NDS9400A	National Semi NDS9400A
29	16	R1-R11,R17,R26,R29,R61,R62	15K	1206 resistor +/-5%
30	1	R13	470K	1206 resistor +/-5%
31	6	R14,R16,R18,R19,R47,R50	10K	1206 resistor +/-5%
32	1	R15	200K potentiometer	Spectrol 63S204T607
33	4	R20,R21,R23,R56	100K	1206 resistor +/-5%
34	1	R22	100K potentiometer	Spectrol 63S104T607
35	3	R24,R25,R30	1K	1206 resistor +/-5%
36	3	R31,R32,R33	150 1%	1206 resistor +/-1%
37	1	R34	6.04K 1%	1206 resistor +/-1%

EPSON   S5U13506P00C ISA BUS EVALUATION BOARD
USER'S MANUAL (X25B-G-004-01)

Table 5-1 Parts List (Continued)

| Item	Quantity	Reference	Part	Description
38	6	R35,R38,R41,R44,R48,R53	68 Ohms	1206 resistor +/-5%
39	1	R36	1.5K 1%	1206 resistor +/-1%
40	3	R37,R43,R46	316 1%	1206 resistor +/-1%
41	3	R39,R45,R49	357 1%	1206 resistor +/-1%
42	2	R40,R42	137 1%	1206 resistor +/-1%
43	2	R51,R54	22 Ohms	1206 resistor +/-5%
44	2	R52,R55	33 Ohms	1206 resistor +/-5%
45	1	R57	1.5K 1%	1206 resistor +/-1%
46	1	R58	1K 1%	1206 resistor +/-1%
47	1	R59	140 Ohms 1%	1206 resistor +/-1%
48	1	R60	69.8 Ohms 1%	1206 resistor +/-1%
49	1	R63	4.7K	1206 resistor +/-5%
50	1	S1	SW DIP-10	10-position DIP switch
51	1	S2	SW DIP-4	Not populated
52	1	U1	S1D13506F00A	Epson S1D13506F00A
53	1	U2	DRAM 1Mx16-SOJ	Alliance AS4LC1M16E5-50JC or ISSI IS41LV16100K-50 or OKI MSM51V18165D50-JS
54	1	U3	RD-0412	Xentek RD-0412
55	1	U4	74AHC04	TI 74AHC04 or National 74VHC04 SO-14 package
56	1	U5	EPN001	Xentek EPN001
57	1	U6	EPF6016TC144-2	Altera EPF6016TC144-2
58	1	U7	EPC1441PC8	Altera EPC1441PC8
59	3	U8,U9,U10	74AHC244	TI 74AHC244 or National 74VHC244 SO-20 package
60	1	U11	LT1117CST-5	Linear Technology LT1117CST-5 or SGS-Thomson LD1117S50C
61	1	U12	ICD2061A	Cypress ICD2061A
62	1	U13	LT1117CST-3.3	Linear Technology LT1117CST-3.3 or SGS-Thomson LD1117S33C
63	1	U15	LT1117CM-3.3	Linear Technology LT1117CM-3.3 or SGS-Thomson LD1117DT33C
64	1	U14	17.734475MHz 14-DIP oscillator	DIP-14 oscillator
65	1	U16	74AHC374	TI 74AHC374 or National 74VHC374, SO-20 package
66	1	Y1	14.31818MHz crystal	14.31818MHz crystal HC-49 Fox FoxS/143-20

6 SCHEMATIC DIAGRAMS
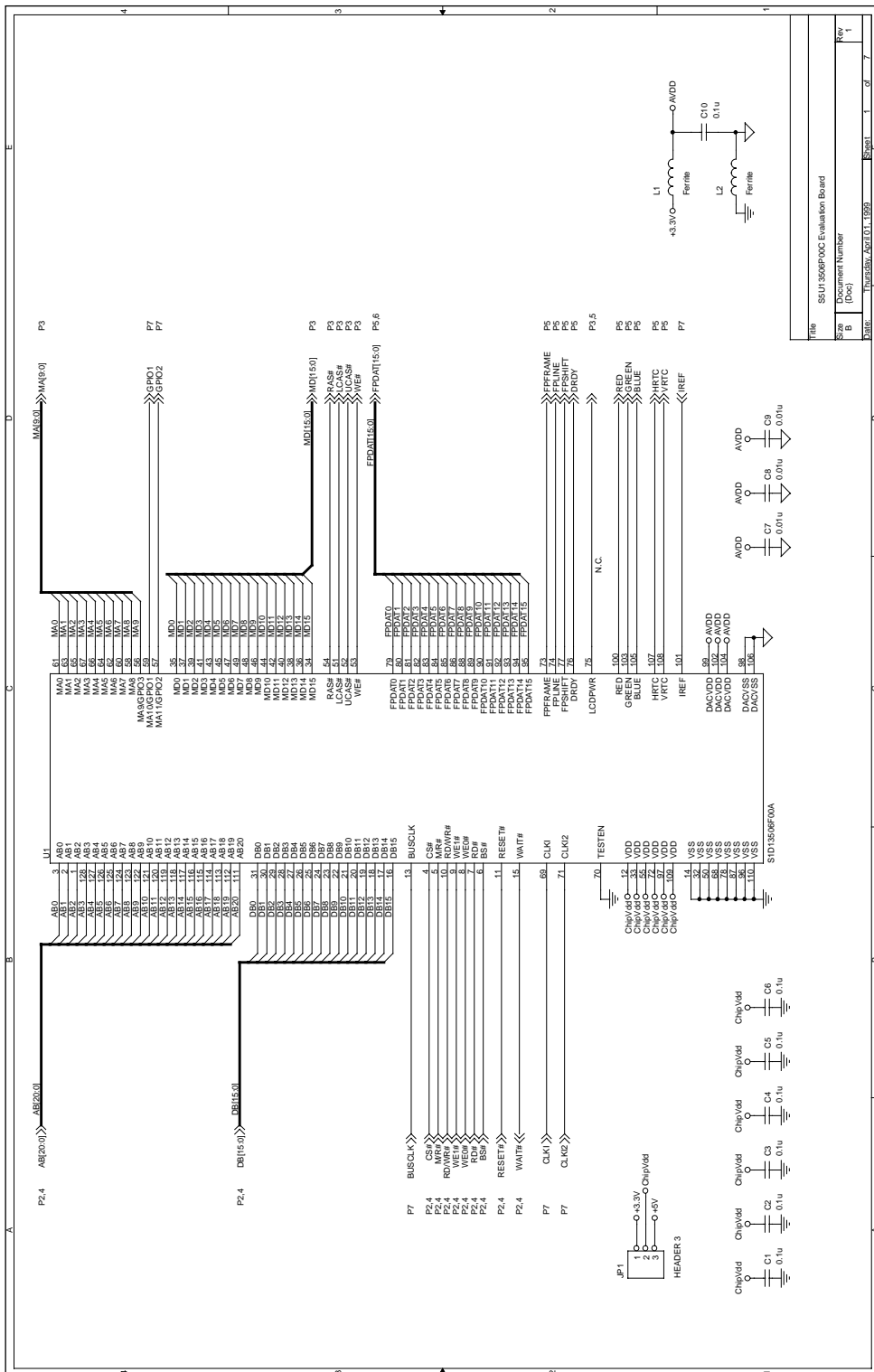
Figure 6-1 S5U13506P00C Schematic Diagram (1 of 7)

EPSON

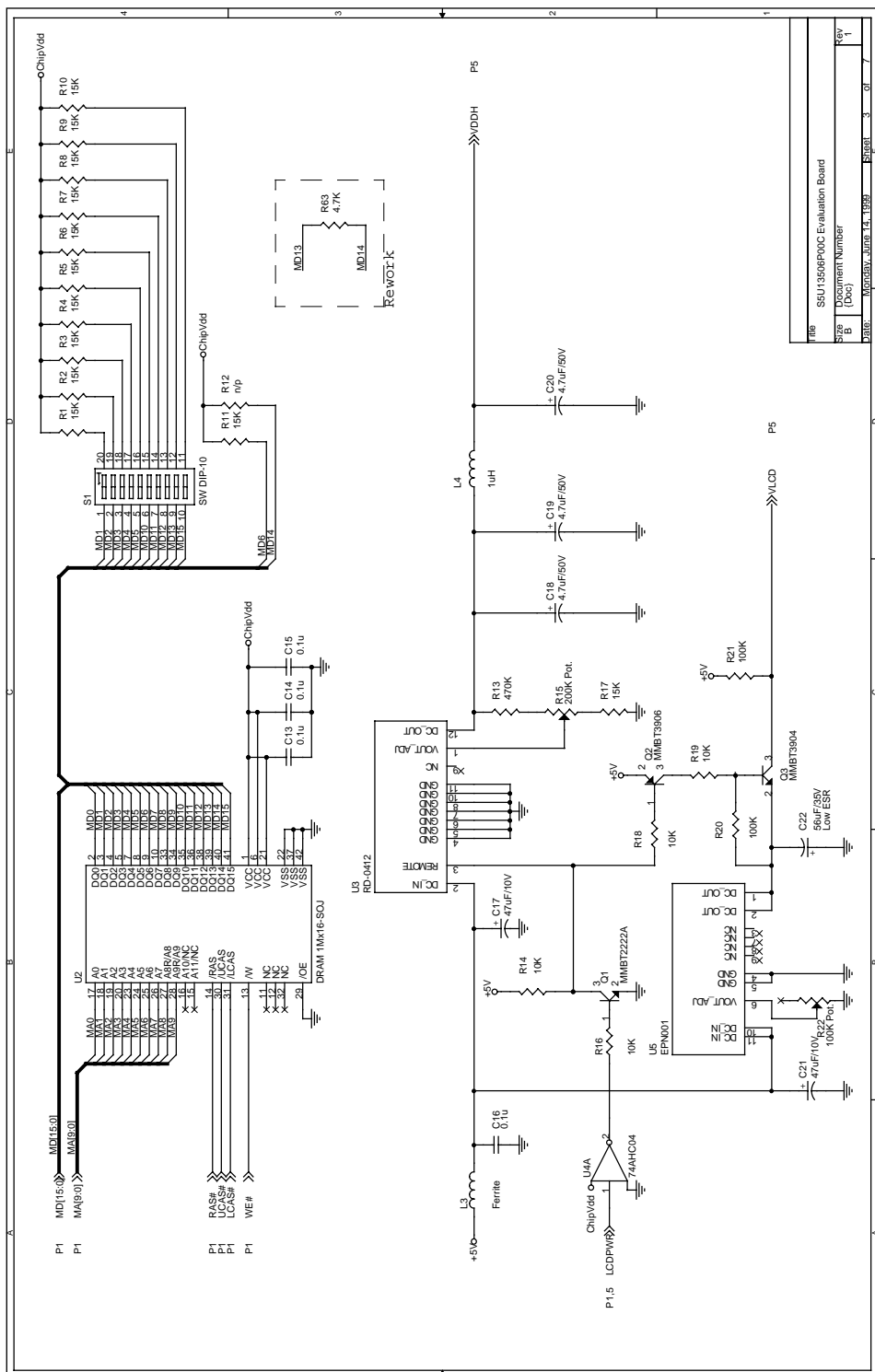Figure 6-2 S5U13506P00C Schematic Diagram (2 of 7)

Figure 6-3 S5U13506P00C Schematic Diagram (3 of 7)
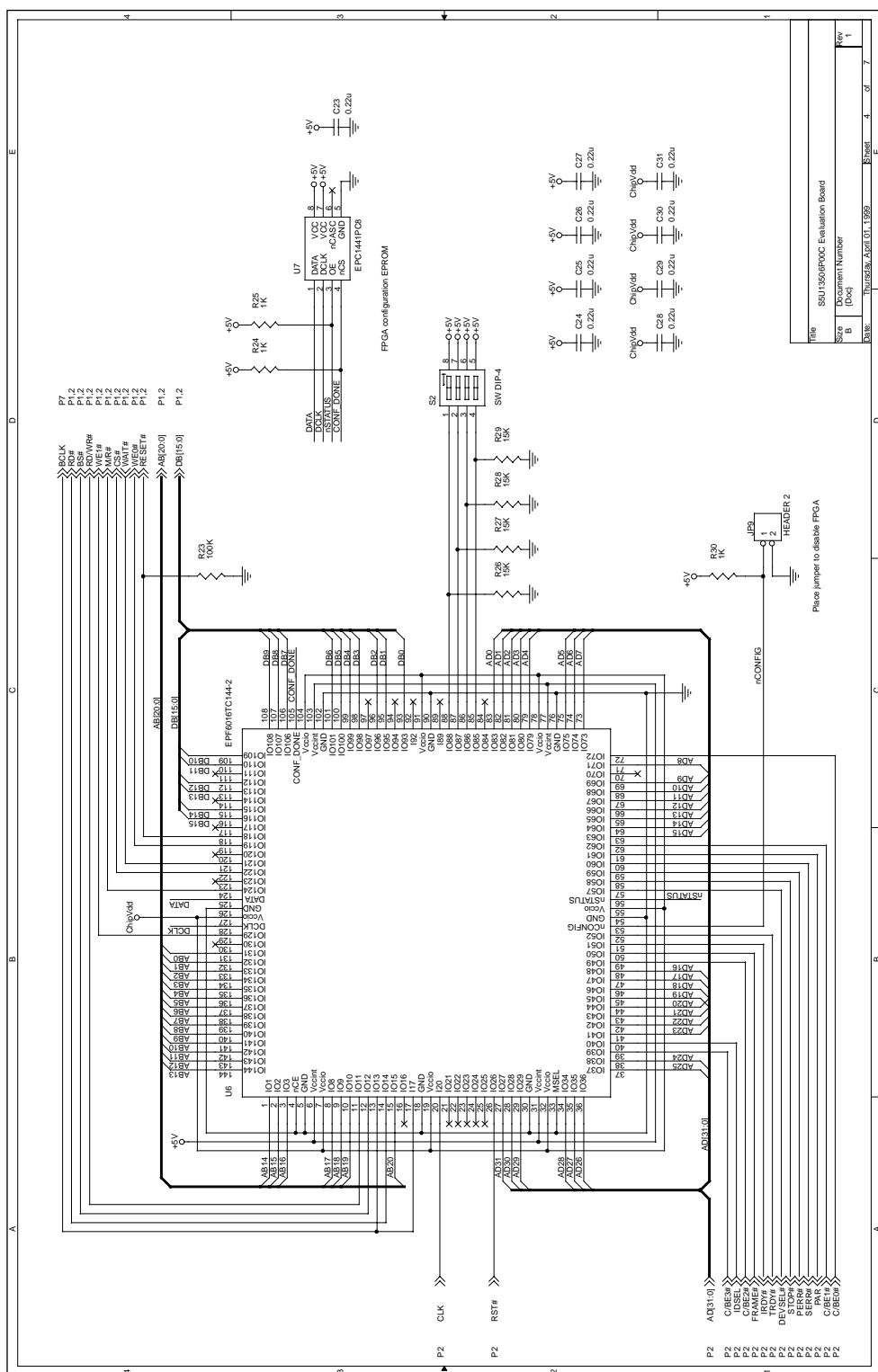
Figure 6-4 S5U13506P00C Schematic Diagram (4 of 7)

Figure 6-5 S5U13506P00C Schematic Diagram (5 of 7)
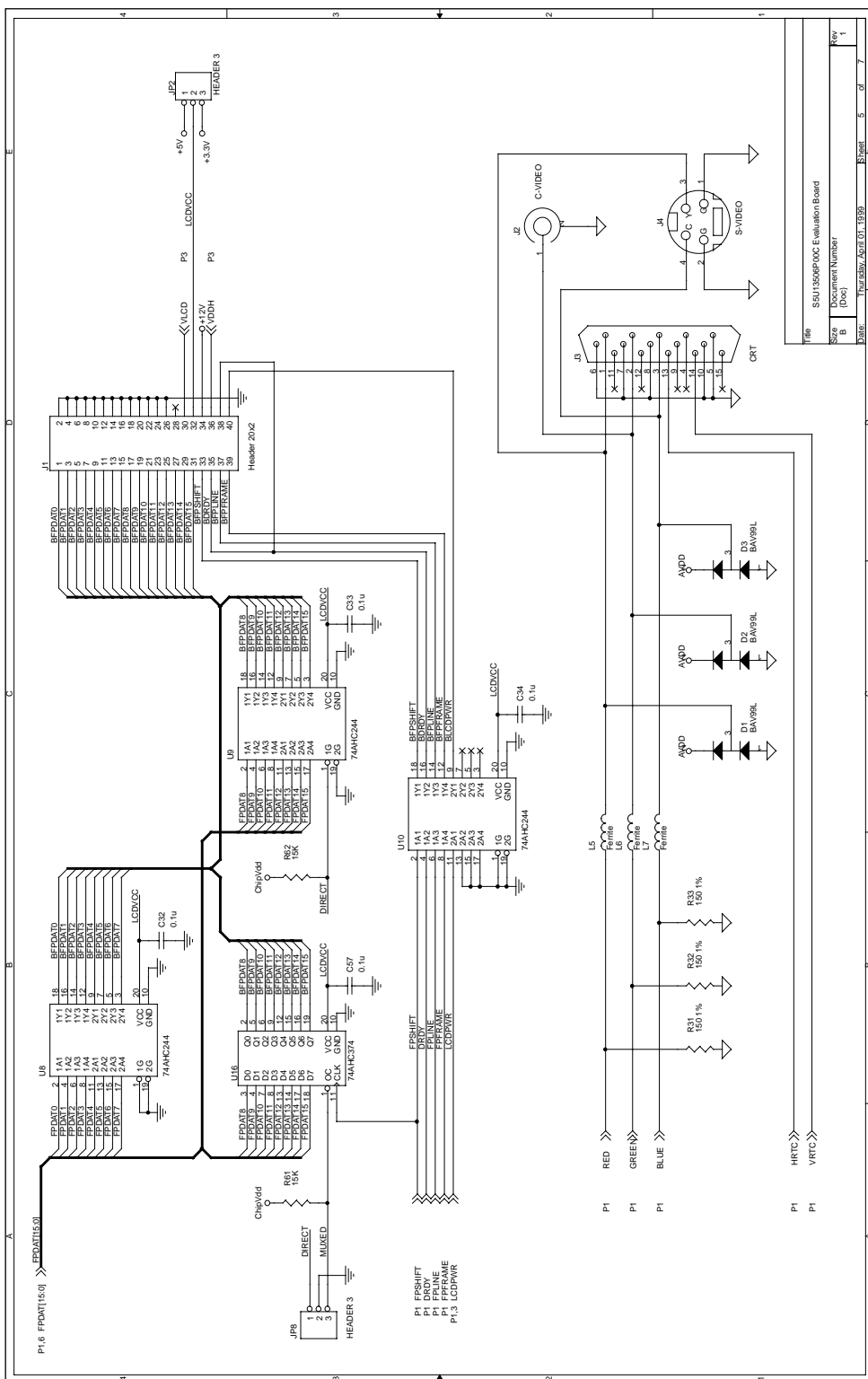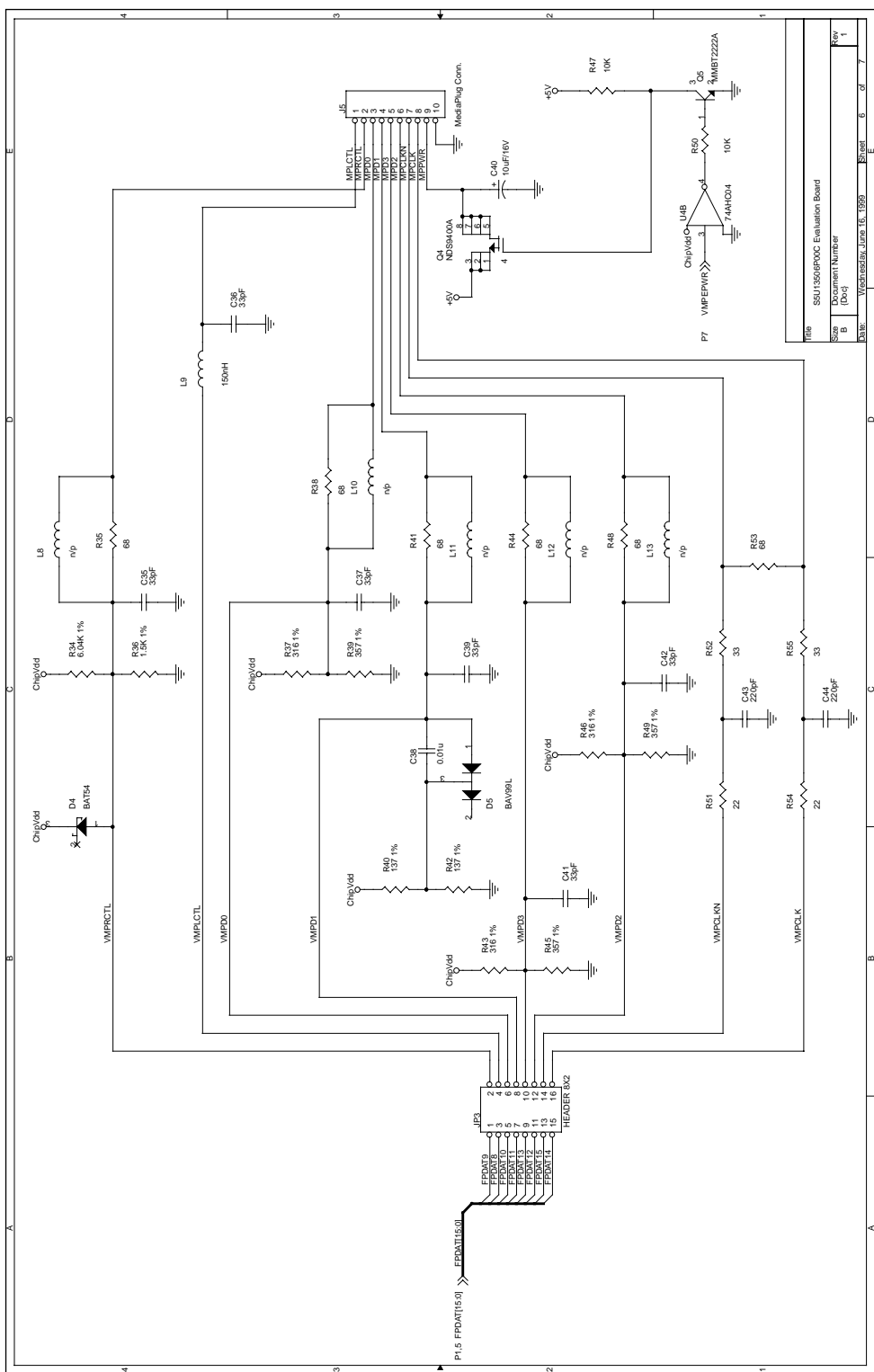
Figure 6-6 S5U13506P00C Schematic Diagram (6 of 7)

Figure 6-7 S5U13506P00C Schematic Diagram (7 of 7)

S1D13506F00A
Color LCD/CRT/TV Controller

Application Notes

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 S1D13506 Color LCD/CRT/TV Controller and the PC Card (PCMCIA) bus.

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 (and later) PC Card Standard support 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.
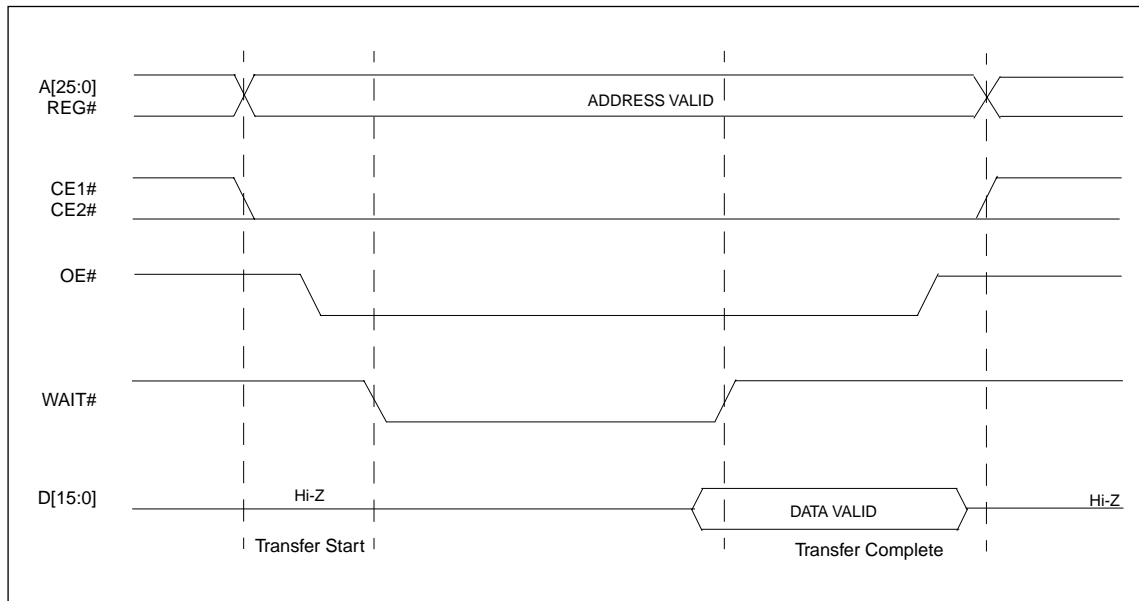
Figure 1-1 PC Card Read Cycle

Figure 1-2 illustrates a typical memory access write cycle on the PC Card bus.
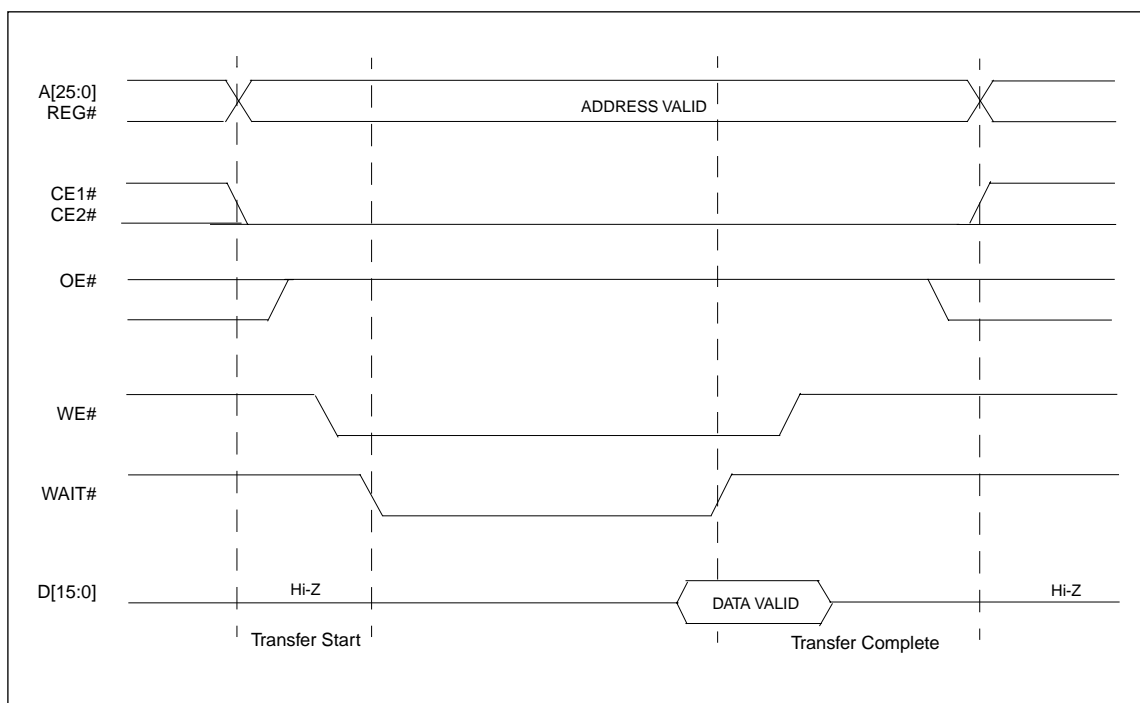
Figure 1-2 PC Card Write Cycle

1.3 S1D13506 Host Bus Interface

The S1D13506 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 S1D13506 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13506 configuration, see Section 1.4.2 "S1D13506 Hardware Configuration" on page 5-7.

> **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

S1D13506 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 S1D13506 internally.
>
> [*2] Although a clock is not directly supplied by the PC Card interface, one is required by the S1D13506 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 5-5.

1.3.2 PC Card Host Bus Interface Signals

The S1D13506 PC Card Host Bus Interface is designed to support processors which interface the S1D13506 through the PC Card bus.

The S1D13506 PC Card Host Bus Interface requires the following signals from the PC Card bus.

- BUSCLK is a clock input which is required by the S1D13506 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 S1D13506 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. MD4 must be set to select little endian mode upon reset.

- 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 whenever the S1D13506 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 S1D13506.

- 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 S1D13506 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 S1D13506 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13506 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete. For PC Card applications, this signal should be set active low using the MD5 configuration input.

- The Bus Start (BS#) signal is not used for the PC Card Host Bus Interface and should be tied high (connected to VDD).

- The RESET# (active low) input of the S1D13506 may be connected to the PC Card RESET (active high) using an inverter.

1.4 PC Card to S1D13506 Interface

1.4.1 Hardware Description

The S1D13506 is designed to directly support a variety of CPUs, providing an interface to each processor's unique "local bus". However, in order to provide support for processors not having an appropriate local bus, the S1D13506 supports a specific PC Card interface.

The S1D13506 provides a "glueless" interface to the PC Card bus except for the following.

- The RESET# signal on the S1D13506 is active low and must be inverted to support the active high RESET provided by the PC Card interface.

- Although the S1D13506 supports an asynchronous bus interface, a clock source is required on the BUSCLK input pin.

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.

The PC Card interface does not provide a bus clock, so one must be supplied for the S1D13506. 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 S1D13506 interface.
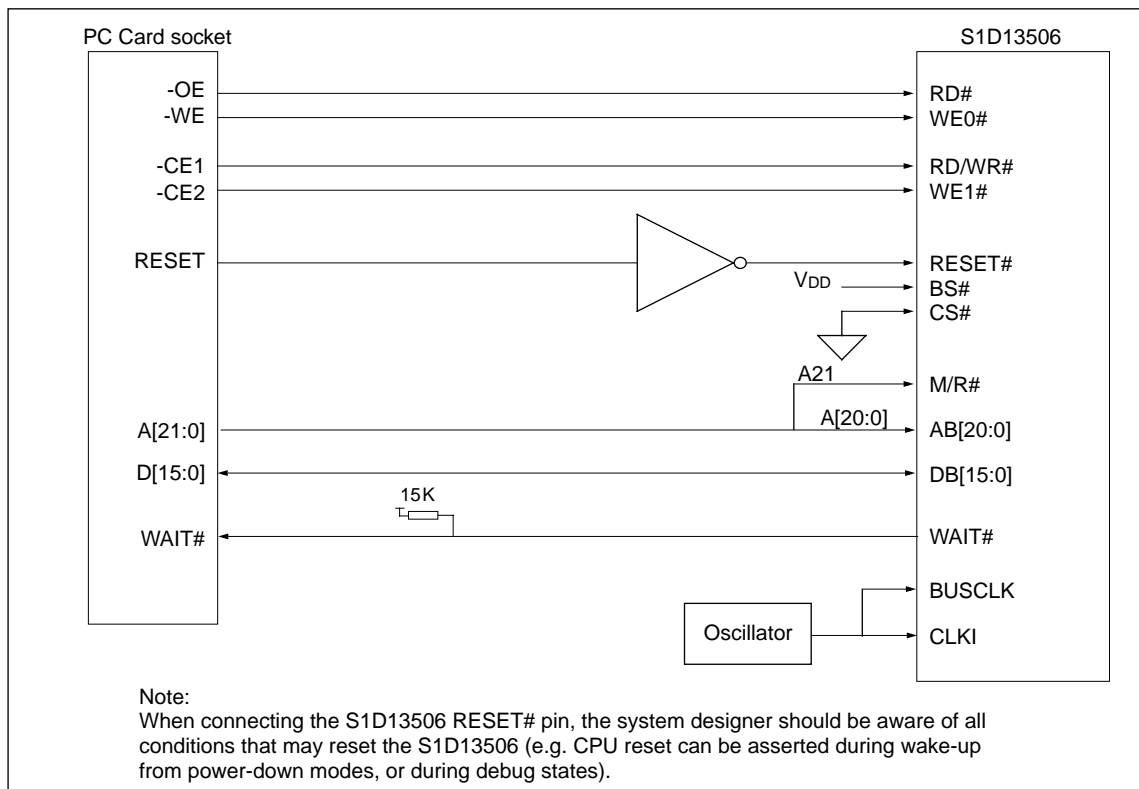
Figure 1-3 Typical Implementation of PC Card to S1D13506 Interface

1.4.2 S1D13506 Hardware Configuration

The S1D13506 latches MD15 through MD0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *"S1D13506 Hardware Functional Specification"*.

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

S1D13506 Pin Name	value on this pin at rising edge of RESET# is used to configure:(1/0)	
	1	0
MD[3:1]	111 = PC Card Host Bus Interface selected	
MD4	Little Endian	Big Endian
MD5	WAIT# is active high (1 = insert wait state)	WAIT# is active low (0 = insert wait state)
MD11	Alternate Host Bus Interface Selected	Primary Host Bus Interface Selected
MD12	BUSCLK input divided by two	BUSCLK input not divided by two
MD15	WAIT# is always driven	WAIT# is floating if S1D13506 is not selected

[shaded box] = configuration for PC Card Host Bus Interface

1.4.3 Performance

The S1D13506 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 S1D13506 is a memory mapped device. The internal registers require 47 bytes and are mapped in the lower PC Card memory address space starting at zero.The display buffer requires 2M bytes and is mapped in the third and fourth megabytes of the PC Card address space (ranging from 200000h to 3FFFFFh).

A typical implementation as shown in Figure 1-3, "Typical Implementation of PC Card to S1D13506 Interface" on page 5-6 has Chip Select (CS#) connected to ground (always enabled) and the Memory/Register select pin (M/R#) connected to address bit A21. This provides the following decoding:

Table 1-3 Register/Memory Mapping for Typical Implementation

CS#	M/R# (A21)	Address Range	Function
0	0	0 - 1F FFFFh	Internal Register Set decoded
0	1	20 0000h - 3F FFFFh	Display Buffer decode

The PC Card socket provides 64M byte of address space. Without further resolution on the decoding logic (M/R# connected to A21), the entire register set is aliased for every 64 byte boundary within the specified address range above. Since address bits A[25:22] are ignored, the S1D13506 registers and display buffer are aliased 16 times.

Note: If aliasing is not desirable, the upper addresses must be fully decoded.

1.5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13506. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13506CFG.EXE, or by directly modifying the source. The Windows® CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

2 INTERFACING TO THE NEC VR4102™/ VR4111™ MICROPROCESSOR

2.1 Introduction

This application note describes the hardware and software environment necessary to provide an interface between the S1D13506 Color LCD/CRT/TV Controller and the NEC VR4102™ (μPD30102) or VR4111™ (μPD30111) Microprocessors.

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.
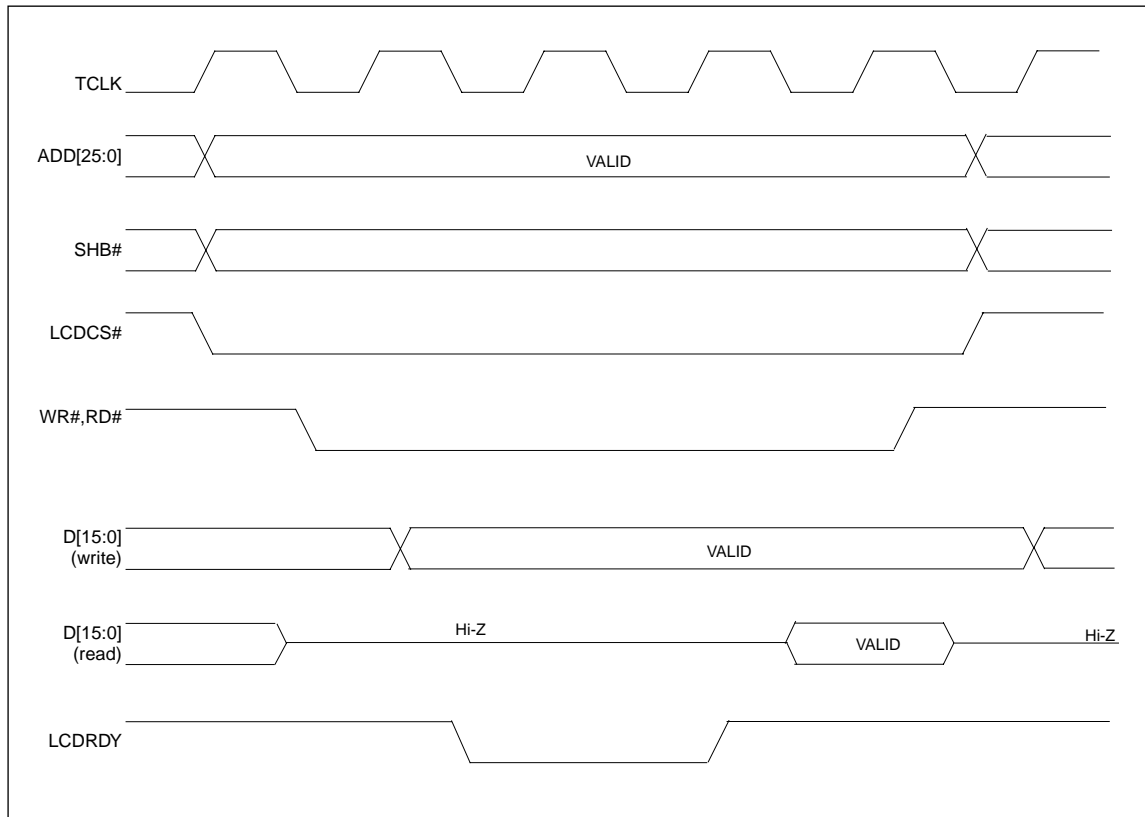
Figure 2-1 NEC V$_R$4102™/V$_R$4111™ Read/Write Cycles

2.3 S1D13506 Host Bus Interface

The S1D13506 directly supports multiple processors. The S1D13506 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 S1D13506 on the rising edge of RESET#. After
releasing reset the bus interface signals assume their selected configuration. For details on
S1D13506 configuration, see Section 2.4.2 "S1D13506 Hardware Configuration" on page 5-14.

> **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

S1D13506 Pin Name	NEC VR4102™/VR4111™ Pin Name
AB20	ADD20
AB[19:0]	ADD[19:0]
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 S1D13506 MIPS/ISA Host Bus Interface requires the following signals.

- BUSCLK is a clock input which is required by the S1D13506 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. MD4 must be set to select the proper endian mode upon reset.

- 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 S1D13506 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 S1D13506.

- 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 S1D13506.

- WAIT# connects to LCDRDY and is a signal output from the S1D13506 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 S1D13506 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13506 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 VDD).

2.4 Vʀ4102™/Vʀ4111™ to S1D13506 Interface

2.4.1 Hardware Description

The NEC Vʀ4102™/Vʀ4111™ 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 S1D13506.

Figure 2-2 NEC Vʀ4102™/Vʀ4111™ to S1D13506 Configuration Schematic

Note: For pin mapping see Table 1-1, "PC Card Host Bus Interface Pin Mapping" on page 1-4.

2.4.2 S1D13506 Hardware Configuration

The S1D13506 latches MD15 through MD0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *"S1D13506 Hardware Functional Specification"*.

The table below shows those configuration settings important to the NEC VR4102™/VR4111™ CPU interface.

Table 2-2 Summary of Power-On/Reset Options

S1D13506 Pin Name	value on this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
MD[3:1]	101 = MIPS/ISA Host Bus Interface	
MD4	Little Endian	Big Endian
MD5	WAIT# is active high (1 = insert wait state)	WAIT# is active low (0 = insert wait state)
MD11	Alternate Host Bus Interface Selected	Primary Host Bus Interface Selected
MD12	BUSCLK input divided by two	BUSCLK input not divided
MD15	WAIT# is always driven	WAIT# is floating if S1D13506 is not selected

☐ = configuration for NEC VR4102™/VR4111™ microprocessor

2.4.3 NEC VR4102™/VR4111™ Configuration

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.

The S1D13506 supports up to 2M bytes of display buffer. The NEC VR4102™/VR4111™ address line A21 is used to select between the S1D13506 display buffer (A21=1) and internal registers (A21=0).

The NEC VR4102™/VR4111™ has a 16-bit internal register named BCUCNTREG2 located at address 0B00 0002h. It must be set to the value of 0001h to indicate that LCD controller accesses using a non-inverting data bus.

2.5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13506. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13506CFG.EXE, or by directly modifying the source. The Windows® CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

3 INTERFACING TO THE NEC VR4121™ MICROPROCESSOR

3.1 Introduction

This application note describes the hardware and software environment necessary to provide an interface between the S1D13506 Color LCD/CRT/TV Controller and the NEC VR4121™ (µPD30121) microprocessor.

3.2 Interfacing to the NEC VR4121™

3.2.1 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.
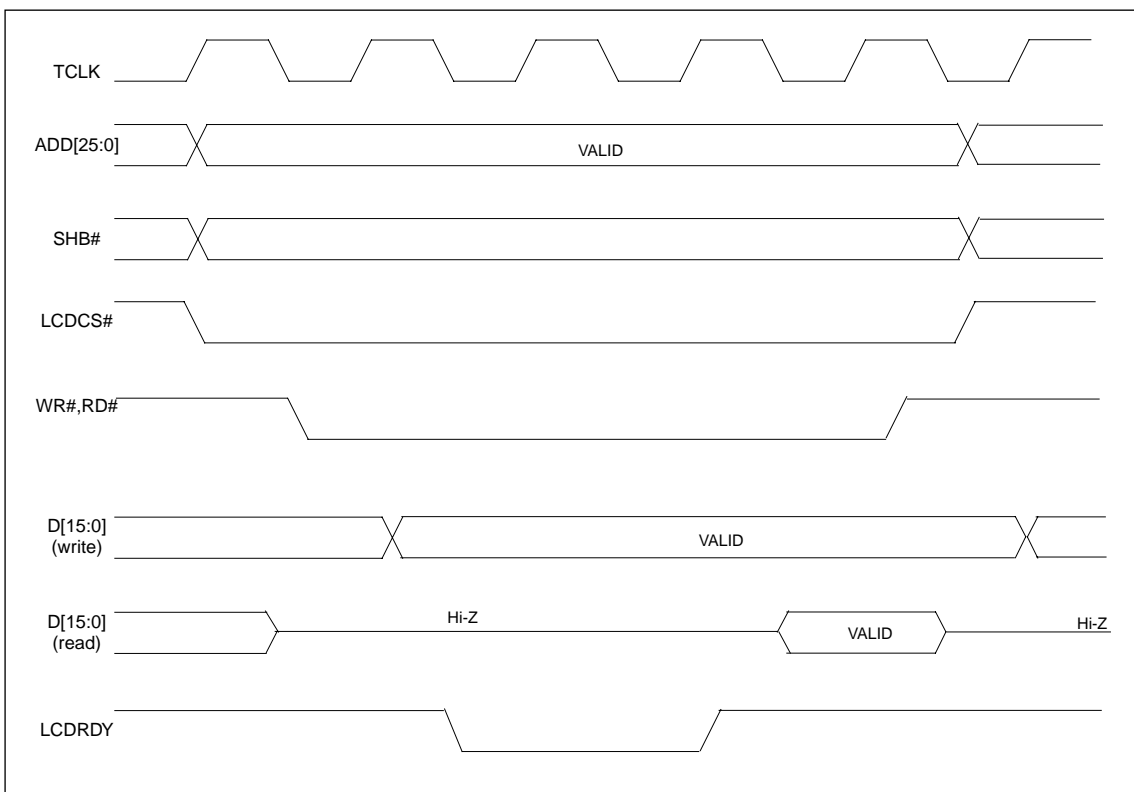
Figure 3-1 NEC VR4121™ Read/Write Cycles

3.3 S1D13506 Host Bus Interface

The S1D13506 directly supports multiple processors. The S1D13506 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 S1D13506 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13506 configuration, see Section 3.4.2 "S1D13506 Configuration" on page 5-20.

> **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 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 3-1 Host Bus Interface Pin Mapping

S1D13506 Pin Name	NEC VR4121™ Pin Name
AB20	ADD20
AB[19:0]	ADD[19:0]
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

3.3.2 Host Bus Interface Signal Descriptions

The S1D13506 MIPS/ISA Host Bus Interface requires the following signals.

- BUSCLK is a clock input which is required by the S1D13506 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. MD4 must be set to select the proper endian mode upon reset.

- 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 S1D13506 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 S1D13506.

- 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 S1D13506.

- WAIT# connects to LCDRDY and is a signal output from the S1D13506 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 S1D13506 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13506 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 VDD).

3.4 VR4121™ to S1D13506 Interface

3.4.1 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 S1D13506.
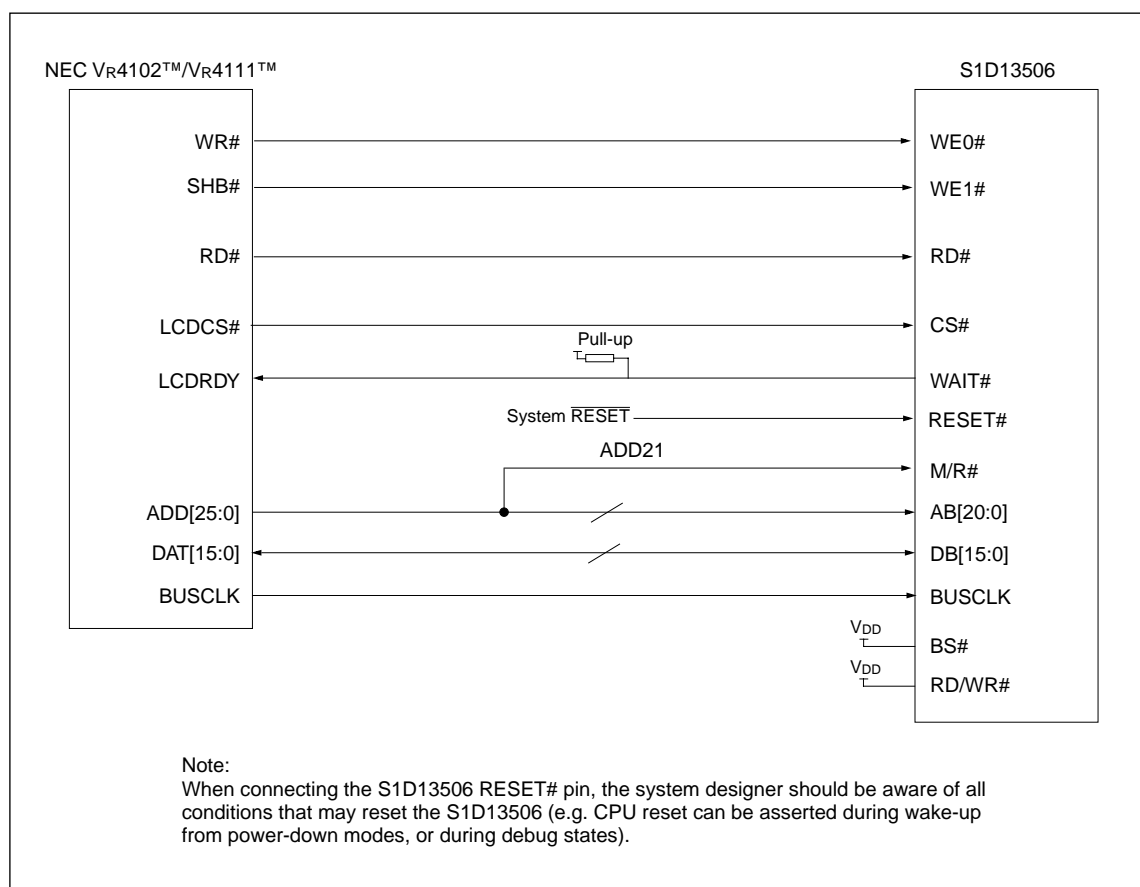
The diagram below shows a typical implementation utilizing the S1D13506.
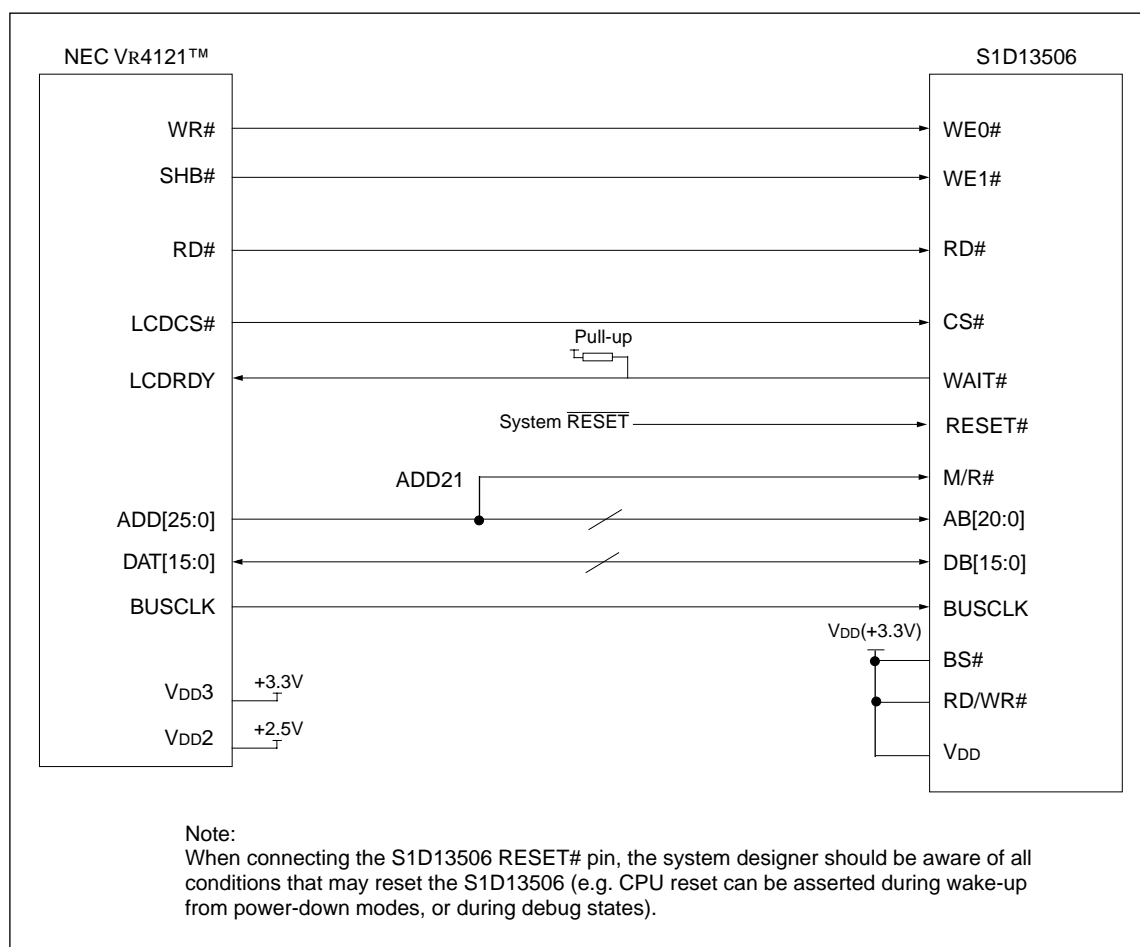
Figure 3-2 NEC VR4121™ to S1D13506 Configuration Schematic

Note: For pin mapping see Table 1-1, "PC Card Host Bus Interface Pin Mapping" on page 1-4.

3.4.2 S1D13506 Configuration

The S1D13506 latches MD15 through MD0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *"S1D13506 Hardware Functional Specification"*.

The table below shows those configuration settings relevant to the MIPS/ISA Host Bus Interface used by the NEC VR4121™ microprocessor.

Table 3-2 Summary of Power-On/Reset Options

S1D13506 Pin Name	value on this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
MD[3:1]	101 = MIPS/ISA Host Bus Interface	
MD4	Little Endian	Big Endian
MD5	WAIT# is active high (1 = insert wait state)	WAIT# is active low (0 = insert wait state)
MD11	Alternate Host Bus Interface Selected	Primary Host Bus Interface Selected
MD12	BUSCLK input divided by 2	BUSCLK input not divided
MD15	WAIT# is always driven	WAIT# is floating if S1D13506 is not selected

☐ = configuration for NEC VR4121™ microprocessor

3.4.3 NEC VR4121™ Configuration

The NEC VR4121™ register BCUCNTREG1 bit ISAM/LCD must be set to 0. A 0 indicates that the reserved address space is for the LCD controller, and not for the high-speed ISA memory. The register BCUCNTREG2 bit GMODE must be set to 1 to indicate that a non-inverting data bus is used for LCD controller accesses.

The LCD interface must be set to operate using a 16-bit data bus. This is accomplished by setting the NEC VR4121™ register BCUCNTREG3 bit LCD32/ISA32 to 0.

Note: Setting the register BCUCNTREG3 bit LCD32/ISA32 to 0 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 S1D13506 works at any of the frequencies provided by the NEC VR4121™.

3.4.4 Memory Mapping and Aliasing

The NEC VR4121™ provides the internal address decoding required by an external LCD controller. The physical address range from 0A00 0000h to 0AFF FFFFh (16M bytes) is reserved for use by an external LCD controller (e.g. S1D13506).

The S1D13506 supports up to 2M bytes of display buffer. The NEC VR4121™ address line ADD21 (connected to M/R#) is used to select between the S1D13506 display buffer (ADD21=1) and the S1D13506 internal registers (ADD21=0). NEC VR4121™ address lines ADD[23:22] are ignored, thus the S1D13506 is aliased four times at 4M byte intervals over the LCD controller address range. Address lines ADD[25:24] are set at 10b and never change while the LCD controller is being addressed.

3.5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13506. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13506CFG.EXE, or by directly modifying the source. The Windows® CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

4 INTERFACING TO THE NEC V832™ MICROPROCESSOR

4.1 Introduction

This application note describes the hardware and software environment required to provide an interface between the S1D13506 Color LCD/CRT/TV Controller and the NEC V832™ microprocessor (µPD705102).

4.2 Interfacing to the NEC V832™

4.2.1 The NEC V832™ System Bus

This section provides an overview of the operation of the CPU bus in order to establish interface requirements.

Overview

The NEC V832™ is designed around the RISC architecture developed by MIPS. This microprocessor is based on the 32-bit V830 CPU core. The CPU communicates with external devices via the Bus Control Unit (BCU). The BCU in turn communicates using its ADD and DATA buses which can be dynamically sized to 16 or 32-bit operation.

The NEC V832™ features dedicated chip select pins which allow memory-mapped IO operations. A 16M byte block of addressing space can be assigned for the LCD controller and its own chip select and ready signals are available. Word or byte accesses are controlled by system byte enable signals ($\overline{\text{LLBEN}}$ and $\overline{\text{LUBEN}}$).

Access Cycles

Once an address in the appropriate range is placed on the external address bus (A[23:1]), the corresponding chip select ($\overline{\text{CSn}}$) is driven low. The read or write enable signals ($\overline{\text{IORD}}$ or $\overline{\text{IOWR}}$) are driven low and $\overline{\text{READY}}$ is driven low by the S1D13506 to insert wait states into the cycle. The byte enable signals ($\overline{\text{LLBEN}}$ and $\overline{\text{LUBEN}}$) allow byte steering.

The following figure illustrates typical NEC V832™ memory-mapped IO access cycles.
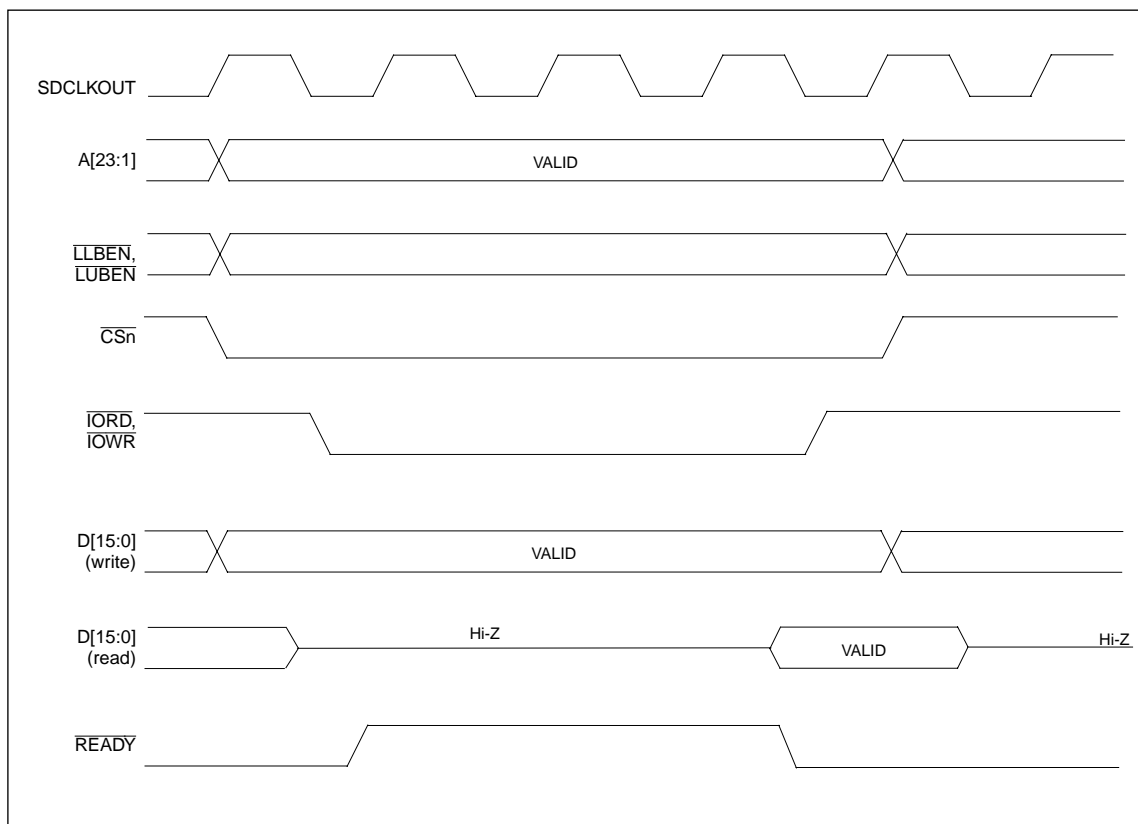
Figure 4-1 NEC V832™ Read/Write Cycles

4.3 S1D13506 Host Bus Interface

The S1D13506 directly supports multiple processors. The S1D13506 implements a 16-bit PC Card (PCMCIA) Host Bus Interface which is most suitable for direct connection to the V832™ microprocessor.

The PC Card Host Bus Interface is selected by the S1D13506 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13506 configuration, see Section 4.4.2 "S1D13506 Hardware Configuration" on page 5-27.

> **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 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 4-1 Host Bus Interface Pin Mapping

S1D13506 Pin Name	NEC V832™ Pin Name
AB[20:1]	A[20:1]
A0	GND[*1]
DB[15:0]	D[15:0]
WE1#	LUBEN
M/R#	A21
CS#	$\overline{CS3}$, $\overline{CS4}$, $\overline{CS5}$ or $\overline{CS6}$
BUSCLK	SDCLKOUT
BS#	VDD (+3.3V)
RD/WR#	LLBEN
RD#	IORD
WE0#	IOWR
WAIT#	READY
RESET#	connected to system reset

> **Note:** [*1] The bus signal A0 is not used by the S1D13506 internally.

4.3.2 Host Bus Interface Signal Descriptions

The S1D13506 PC Card Host Bus Interface requires the following signals.

• BUSCLK is a clock input which is required by the S1D13506 Host Bus Interface. It is driven by the V832™ signal SDCLKOUT.

• The address inputs AB[20:0], and the data bus DB[15:0], connect directly to the V832™ address (A[20:0]) and data bus (D[15:0]), respectively. MD4 must be set to select little endian mode upon reset.

• 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 $\overline{CSx}$ (where x is the V832™ chip select used) whenever the S1D13506 is accessed by the V832™.

• WE1# and RD/WR# connect to $\overline{LUBEN}$ and $\overline{LLBEN}$ (the byte enables for the high-order and low-order bytes). They are driven low when the V832™ is accessing the S1D13506.

• RD# connects to $\overline{IORD}$ (the read enable signal from the V832™).

• WE0# connects to $\overline{IOWR}$ (the write enable signal from the V832™).

• WAIT# is a signal output from the S1D13506 that indicates the V832™ must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since V832™ accesses to the S1D13506 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13506 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete. For V832™ applications, this signal should be set active low using the MD5 configuration input.

• 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 S1D13506 may be connected to the system RESET.

4.4 V832™ to S1D13506 Interface

4.4.1 Hardware Description

The NEC V832™ microprocessor features configurable chip select lines which can easily be used for an external LCD controller. It provides all the necessary internal address decoding and control signals required by the S1D13506.

The diagram below shows a typical implementation utilizing the S1D13506.
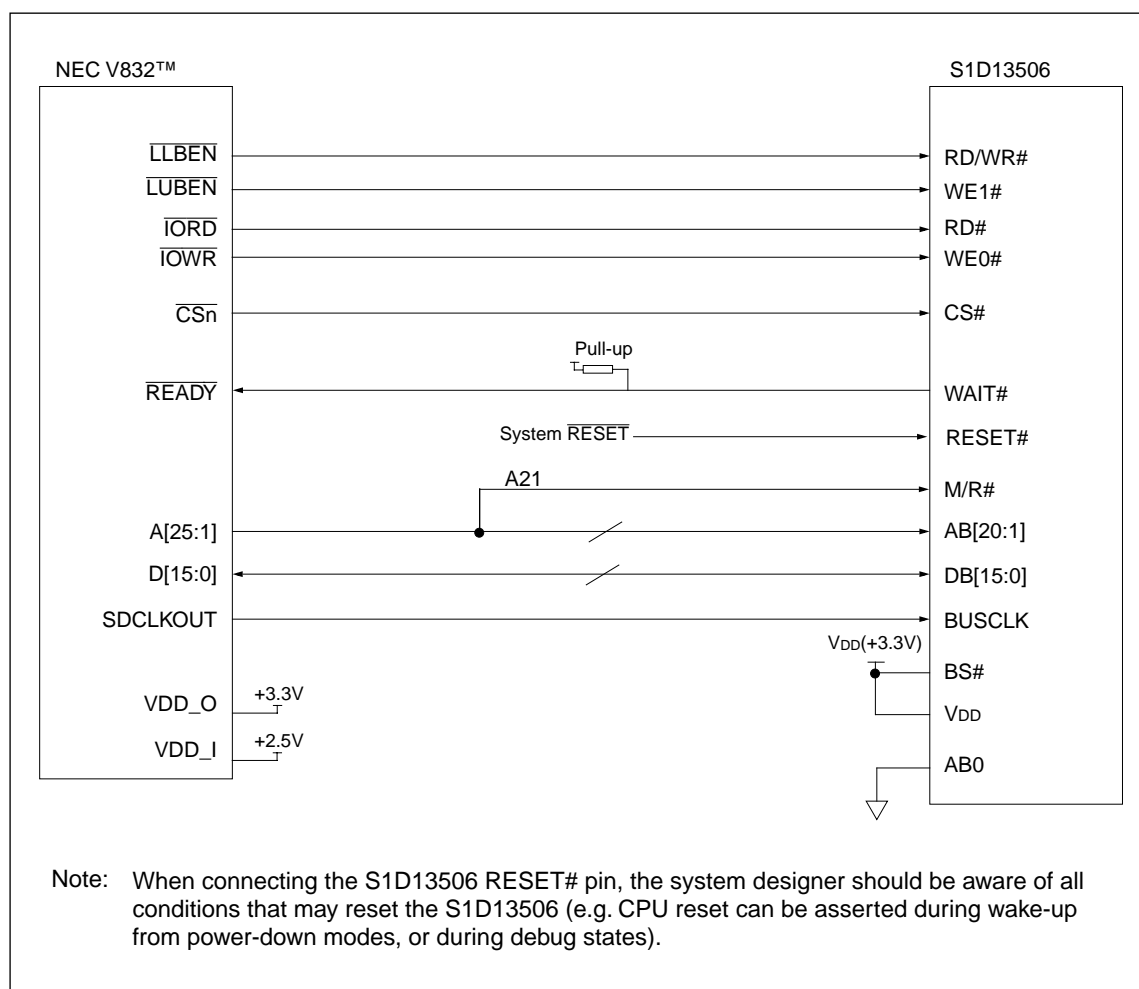
Figure 4-2 NEC V832™ to S1D13506 Configuration Schematic

Note: For pin mapping see Table 4-1, "Host Bus Interface Pin Mapping" on page 1-24.

4.4.2 S1D13506 Hardware Configuration

The S1D13506 latches MD15 through MD0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *"S1D13506 Hardware Functional Specification"*.

The table below shows those configuration settings relevant to the PC Card Host Bus Interface used by the NEC V832™ microprocessor.

Table 4-2 Summary of Power-On/Reset Options

S1D13506 Pin Name	Value on this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
MD[3:1]	111 = PC Card Host Bus Interface	
MD4	Little Endian	Big Endian
MD5	WAIT# is active high (1 = insert wait state)	WAIT# is active low (0 = insert wait state)
MD11	Alternate Host Bus Interface Selected	Primary Host Bus Interface Selected
MD12	BUSCLK input divided by two	BUSCLK input not divided by two
MD15	WAIT# is always driven	WAIT# is floating if S1D13506 is not selected

	= configuration for NEC V832™ microprocessor

4.5 NEC V832™ Configuration

The NEC V832™ should access the S1D13506 in non-burst mode only. This is ensured by using any one of the $\overline{\text{CS3}}$ to $\overline{\text{CS6}}$ lines to control the S1D13506 and setting that line to respond to IO operations using the NEC V832™ BCTC register. For example, if line $\overline{\text{CS5}}$ is used then bit 5 (CT5) of the BCTC register should be set to 1 (IO cycle).

The NEC V832™ data bus should be programmed to use 16 bits as the maximum width for S1D13506 bus transactions. This does not affect the width of other NEC V832™ data bus transactions. Data bus width is set in the NEC V832™ DBC register. For example, if line $\overline{\text{CS4}}$ is used then bit 4 (BW4) of the DBC register should be set to 1 (16-bit bus width).

Depending on bus clock frequencies, a different number of wait states may be required. These need to be programmed into the NEC V832™ PWC0 and PWC1 registers in the bit field corresponding to the $\overline{\text{CSn}}$ line chosen for the S1D13506. For example, if $\overline{\text{CS3}}$ is used and one wait state is required, then bits 14-12 of the NEC V832™ PWC0 register (WS3) must be set to 001b (one wait state). If $\overline{\text{CS6}}$ is used and no wait state is needed, then bits 11-8 of the NEC V832™ PWC1 register (WS6) must be set to 0000b (zero wait state).

The table below shows the recommended wait states depending on the bus clock frequency.

Table 4-3 NEC V832™ Wait States vs. Bus Clock Frequency

Wait States	Maximum Frequency (SDCLKOUT)
0	10.8MHz
1	32.6MHz
2	No limit

No idle state needs to be added. The NEC V832™ PIC0 and PIC1 register bit field corresponding to the $\overline{\text{CSn}}$ line chosen for the S1D13506 must be set to zero. For example, if $\overline{\text{CS3}}$ is used then bits 14-12 of the NEC V832™ PIC0 register (IS3) must be set to 000b (no idle state).

4.5.1 Memory Mapping and Aliasing

The $\overline{\text{CSn}}$ line selected determines the address range to be reserved for the S1D13506. The table below summarizes the S1D13506 address mapping.

Table 4-4 NEC V832™ IO Address Range For Each $\overline{\text{CSn}}$ Line

$\overline{\text{CSn}}$ Line	NEC V832™ IO Address		S1D13506 Function
CS3	0300 0000h to 03FF FFFFh	0300 0000h	Registers
		0320 0000h	Display buffer (2M bytes)
CS4	0400 0000h to 04FF FFFFh	0400 0000h	Registers
		0420 0000h	Display buffer (2M bytes)
CS5	0500 0000h to 05FF FFFFh	0500 0000h	Registers
		0520 0000h	Display buffer (2M bytes)
CS6	0600 0000h to 06FF FFFFh	0600 0000h	Registers
		0620 0000h	Display buffer (2M bytes)

Each address range is 16M bytes, therefore, the S1D13506 is aliased four times over the address range.

4.6 *Software*

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13506. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13506CFG.EXE, or by directly modifying the source. The Windows® CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

5 INTERFACING TO THE MOTOROLA MPC821 MICROPROCESSOR

5.1 Introduction

This application note describes the hardware and software environment required to provide an interface between the S1D13506 Color LCD/CRT/TV Controller and the Motorola MPC821 processor.

5.2 Interfacing to the MPC821

5.2.1 The MPC8xx System Bus

The MPC8xx family of processors feature a high-speed synchronous system bus typical of modern RISC microprocessors. This section provides an overview of the operation of the CPU bus in order to establish interface requirements.

5.2.2 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 MCLK (Master Clock). This clock runs at the machine cycle speed of the CPU core (typically 25 to 50 MHz). 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 run at one-half the CPU core speed using the clock control register. This is typically used when the CPU core is operated above 50 MHz.

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.

It should be noted that all Power PC microprocessors, including the MPC8xx family, use bit notation 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. All IO accesses are memory-mapped meaning there is no separate IO space in the Power PC architecture.

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

A data transfer is initiated by the bus master by placing the memory address on address lines A0 through A31 and driving $\overline{TS}$ (Transfer Start) low for one clock cycle. Several control signals are also provided with the memory address:

- TSIZ[0:1] (Transfer Size) -- indicates whether the bus cycle is 8, 16, or 32-bit.
- RD/$\overline{WR}$ -- set high for read cycles and low for write cycles.
- AT[0:3] (Address Type Signals) -- provides more detail on the type of transfer being attempted.

When the peripheral device being accessed has completed the bus transfer, it asserts $\overline{TA}$ (Transfer Acknowledge) for one clock cycle to complete the bus transaction. Once $\overline{TA}$ has been asserted, the MPC821 will not start another bus cycle until $\overline{TA}$ has been de-asserted. The minimum length of a bus transaction is two bus clocks.

Figure 5-1 "Power PC Memory Read Cycle" on page 31 illustrates a typical memory read cycle on the Power PC system bus.
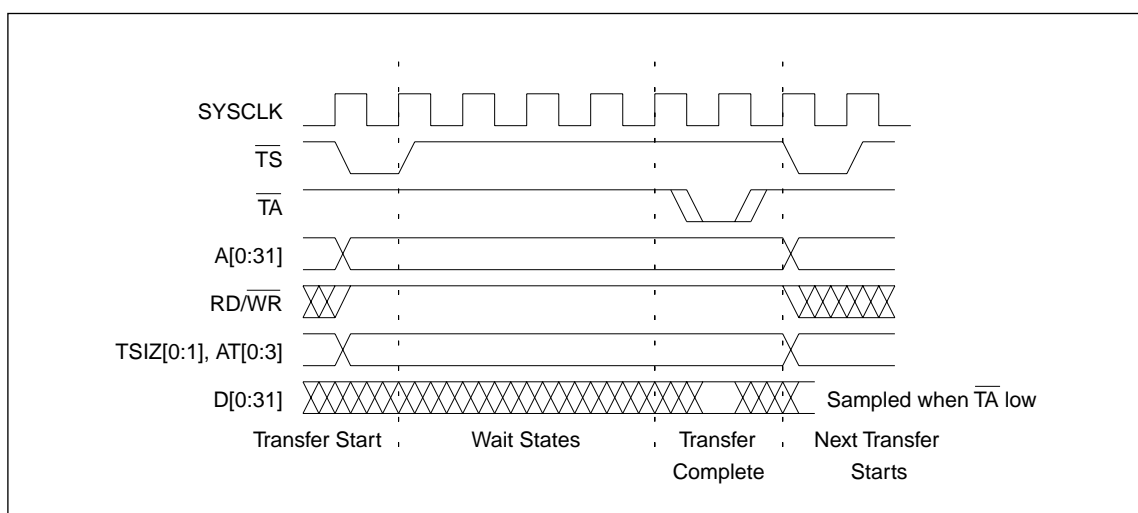
Figure 5-1 Power PC Memory Read Cycle

Figure 5-2 "Power PC Memory Write Cycle" on page 32 illustrates a typical memory write cycle on the Power PC system bus.
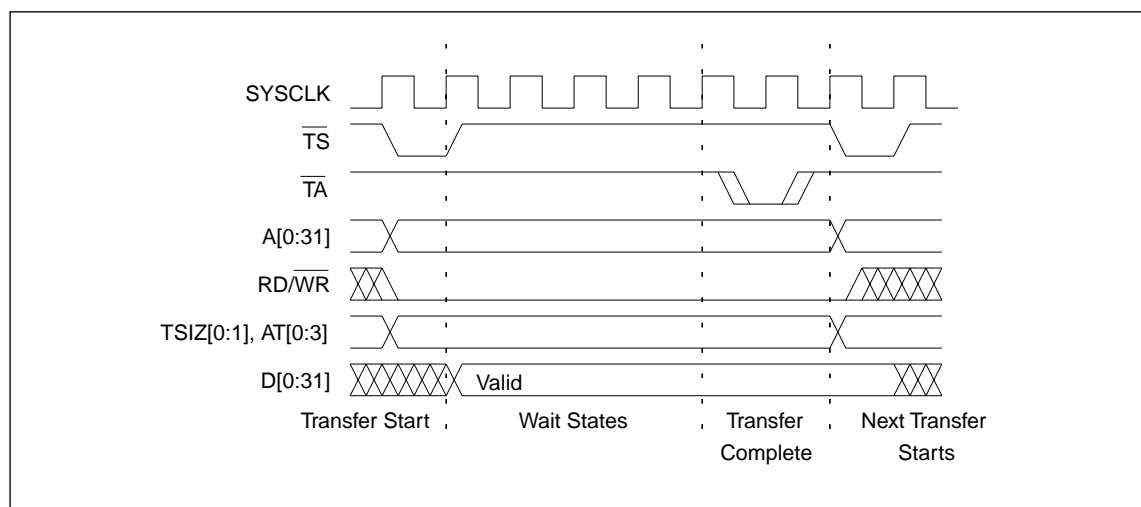
Figure 5-2 Power PC Memory Write Cycle

If an error occurs, $\overline{TEA}$ (Transfer Error Acknowledge) is asserted and the bus cycle is aborted. For example, a peripheral device may assert $\overline{TEA}$ if a parity error is detected, or the MPC821 bus controller may assert $\overline{TEA}$ 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 ($\overline{BI}$) simultaneously with $\overline{TA}$, and the processor will revert 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 S1D13506, 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 $\overline{BDIP}$ and respond with $\overline{BI}$ if caching is accidently enabled for the S1D13506 address space.

5.2.3 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 ($\overline{OE}$) and Write Enable ($\overline{WE}$) signals compatible with most memory and x86-style peripherals. The MPC821 bus controller also provides a Read/ Write (RD/$\overline{WR}$) 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, π, or $\int$ clock cycle with respect to the address bus valid.

- The CSNT bit causes chip select and $\overline{WE}$ to be negated $\int$ 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 $\overline{TA}$ (Transfer Acknowledge).

- Any chip select may be programmed to assert $\overline{BI}$ (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 S1D13506 and it is desirable to leave the UPM free to handle other interfacing duties, such as EDO DRAM.

5.3 S1D13506 Host Bus Interface

The S1D13506 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 S1D13506 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13506 configuration, see Section 5.4.3 "S1D13506 Hardware Configuration" on page 5-38.

> **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 PowerPC Host Bus Interface Pin Mapping

The following table shows the functions of each host bus interface signal.

Table 5-1 PowerPC Host Bus Interface Pin Mapping

S1D13506 Pin Names	PowerPC
AB[20:0]	A[11:31]
DB[15:0]	D[0:15]
WE1#	BI
M/R#	External Decode
CS#	External Decode
BUSCLK	CLKOUT
BS#	TS
RD/WR#	RD/WR
RD#	TSIZ0
WE0#	TSIZ1
WAIT#	TA
RESET#	RESET#

5.3.2 PowerPC Host Bus Interface Signals

The S1D13506 PowerPC host bus interface is designed to support processors which interface the S1D13506 through the PowerPC bus.

The S1D13506 PowerPC host bus interface requires the following signals:

- BUSCLK is a clock input which is required by the S1D13506 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. MD4 must be set to select the proper endian mode upon reset.

- 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 whenever the S1D13506 is accessed by the PowerPC bus.

- RD/WR# connects to $\overline{\text{RD/WR}}$ which indicates whether a read or a write access is being performed on the S1D13506.

- WE1# connects to $\overline{\text{BI}}$ (burst inhibit signal). WE# is output by the S1D13506 to indicate whether the S1D13506 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 $\overline{\text{TA}}$ and is output from the S1D13506 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 S1D13506 may occur asynchronously to the display update, it is possible that contention may occur while accessing the S1D13506 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 $\overline{\text{TS}}$ (the transfer start signal).

5.4 MPC821 to S1D13506 Interface

5.4.1 Hardware Description

The S1D13506 provides native Power PC bus support, making it very simple to interface the two devices. This application note describes the environment necessary to connect the S1D13506 to the MPC821 native system bus and the connections between the S5U13506P00C Evaluation Board and the Motorola MPC821 Application Development System (ADS).

The S1D13506, by implementing a dedicated display buffer, can reduce system power consumption, improve image quality, and increase system performance as compared to the MPC821's on-chip LCD controller.

The S1D13506, through the use of the MPC821 chip selects, can share the system bus with all other MPC821 peripherals. The following figure demonstrates a typical implementation of the S1D13506 to MPC821 interface.

Note: When connecting the S1D13506 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13506 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).
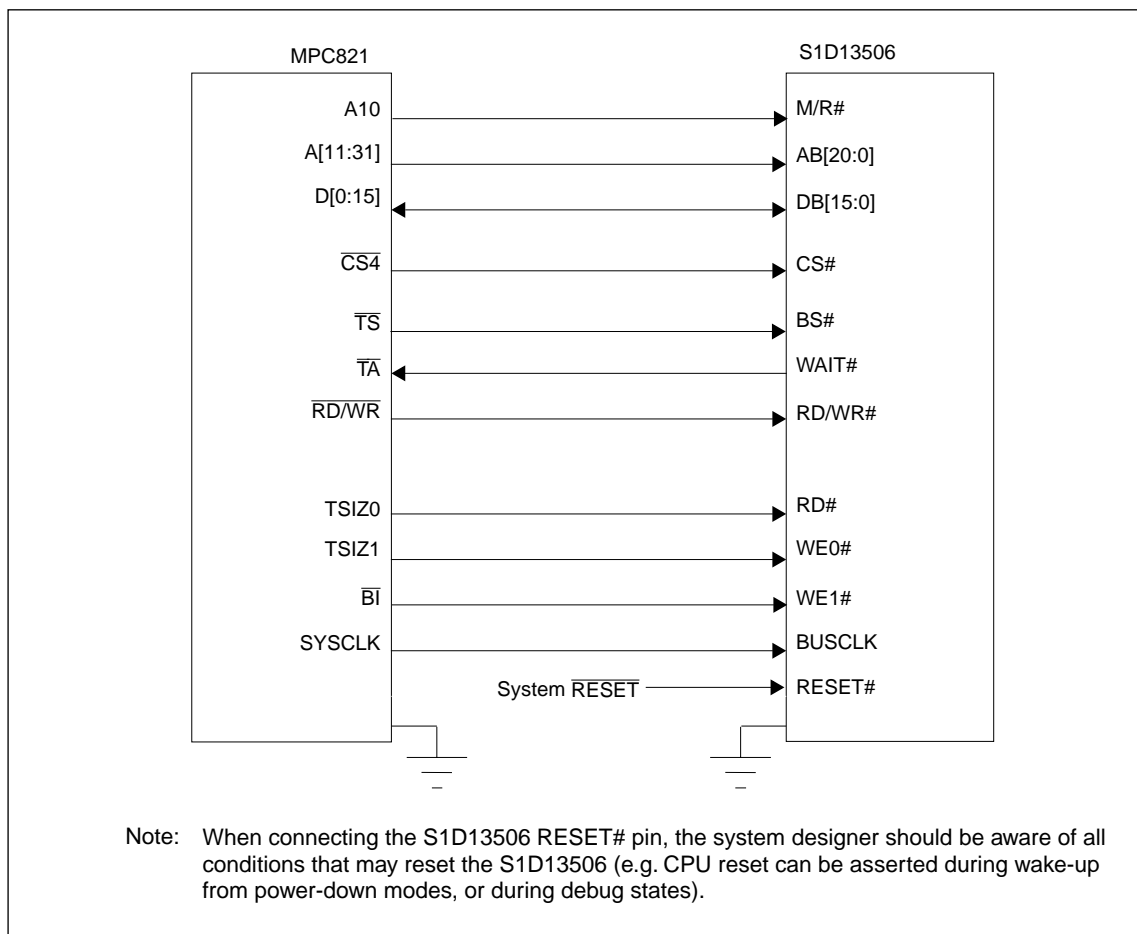
Figure 5-3 Typical Implementation of MPC821 to S1D13506 Interface

Table 5-2, "List of Connections from MPC821ADS to S1D13506" on page 4-37 shows the connections between the pins and signals of the MPC821 and the S1D13506.

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 S1D13506. In a true 3 volt system, no buffering is necessary.

5.4.2 Hardware Connections

The following table details the connections between the pins and signals of the MPC821 and the S1D13506.

Table 5-2 List of Connections from MPC821ADS to S1D13506

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13506 Signal Name
Vcc	P6-A1, P6-B1	Vcc
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
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#

Table 5-2 List of Connections from MPC821ADS to S1D13506 (Continued)

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13506 Signal Name
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.

5.4.3 S1D13506 Hardware Configuration

The S1D13506 latches MD15 through MD0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *"S1D13506 Hardware Functional Specification"*.

The following table shows those configuration settings important to the MPC821 host bus interface.

Table 5-3 Summary of Power-On/Reset Options

S1D13506 Pin Name	value on this pin at rising edge of RESET# is used to configure: (1/0)	
	1	0
MD[3:1]	110 = PowerPC host bus interface selected	
MD4	Little Endian	Big Endian
MD5	Wait# signal is active high	Wait# signal is active low
MD9	Reserved	Configure SUSPEND# pin as Hardware Suspend Enable
MD11	Alternate Host Bus Interface Selected	Primary Host Bus Interface Selected
MD12	BUSCLK input divided by two	BUSCLK input not divided
MD15	WAIT# is always driven	WAIT# is floating if S1D13506 is not selected

[ ] = required settings for MPC821 support.

5.4.4 Register/Memory Mapping

The DRAM on the MPC821 ADS board extends from address 0 through 3F FFFFh, so the S1D13506 is addressed starting at 40 0000h. A total of 4M bytes of address space is used, where the lower 2M bytes is reserved for the S1D13506 on-chip registers and the upper 2M bytes is used to access the S1D13506 display buffer.

5.4.5 MPC821 Chip Select Configuration

Chip select 4 is used to control the S1D13506. The following options are selected in the base address register (BR4):

- BA[0:16] = 0000 0000 0100 0000 0   – set starting address of S1D13506 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; S1D13506 consumes 4M byte of address space.

- ATM[0:2] = 0   – ignore address type bits.

- CSNT = 0   – normal $\overline{CS}/\overline{WE}$ negation.

- ACS[0:1] = 1:1   – delay $\overline{CS}$ assertion by ∫ 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 S1D13506 generates an external transfer acknowledge using the WAIT# line.

- TRLX = 0   – normal timing.

- EHTR = 0   – normal timing.

5.4.6 Test Software

The test software is very simple. It configures chip select 4 (CS4) on the MPC821 to map the S1D13506 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 S1D13506 register REG[001h] to enable full S1D13506 memory/register decoding. Lastly, the software runs a tight loop that reads the S1D13506 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 S1D13506 memory block is tagged as non-cacheable. This ensures the MPC821 does not attempt to cache any data read from, or written to, the S1D13506 or its display buffer.

```
BR4          equ      $120              ; CS4 base register
OR4          equ      $124              ; CS4 option register
MemStart     equ      $40               ; upper word of S1D13506 start address
DisableReg   equ      $1                ; address of S1D13506 Disable Register
RevCodeReg   equ      $0                ; address of Revision Code Register

Start        mfspr    r1,IMMR           ; get base address of internal registers
             andis.   r1,r1,$ffff       ; clear lower 16 bits to 0
             andis.   r2,r0,0           ; clear r2
             oris     r2,r2,MemStart    ; write base address
             ori      r2,r2,$0801       ; port size 16 bits; select GPCM; enable
             stw      r2,BR4(r1)        ; write value to base register
             andis.   r2,r0,0           ; clear r2
             oris     r2,r2,$ffc0       ; address mask – use upper 10 bits
             ori      r2,r2,$0608       ; normal CS negation; delay CS ∫ clock;
                                        ; no burst inhibit (13506 does this)
             stw      r2,OR4(r1)        ; write to option register
             andis.   r1,r0,0           ; clear r1
             oris     r1,r1,MemStart    ; point r1 to start of S1D13506 mem space
             stb      r1,DisableReg(r1) ; write 0 to disable register
Loop         lbz      r0,RevCodeReg(r1) ; read revision code into r1
             b        Loop              ; branch forever

             end
```

Note: MPC8BUG does not support comments or symbolic equates; these have been added for clarity.

5.5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13506. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13506CFG.EXE, or by directly modifying the source. The Windows CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

6 INTERFACING TO THE TOSHIBA MIPS TX3912 PROCESSOR

6.1 Introduction

This application note describes the hardware and software environment necessary to provide an interface between the S1D13506 Color LCD/CRT/TV Controller and the Toshiba MIPS TX3912 Processor.

6.2 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 S1D13506 connects to the TX3912 processor.

The S1D13506 can be successfully interfaced using one of the following configurations:

- Direct connection to the TX3912 (see Section 6.4 "Direct Connection to the Toshiba TX3912" on page 5-45).
- System design using the ITE IT8368E PC Card/GPIO buffer chip (see Section 6.5 "System Design Using the IT8368E PC Card Buffer" on page 5-48).

6.3 S1D13506 Host Bus Interface

The S1D13506 implements a 16-bit Host Bus Interface specifically for interfacing to the TX3912 microprocessor.

The TX3912 Host Bus Interface is selected by the S1D13506 on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. For details on S1D13506 configuration, see Section 6.4.2 "S1D13506 Configuration" on page 5-46.

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 TX3912 Host Bus Interface Pin Mapping

The following table shows the function of each Host Bus Interface signal.

Table 6-1 TX3912 Host Bus Interface Pin Mapping

S1D13506 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*

6.3.2 TX3912 Host Bus Interface Signals

When the S1D13506 is configured to operate with the TX3912, the Host Bus Interface requires the following signals:

- BUSCLK is a clock input required by the S1D13506 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.

- 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.

- 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. MD4 must be set to select the proper endian mode on reset (see Section 6.4.2 "S1D13506 Configuration" on page 5-46). **Because of the TX3912 data bus naming convention and endian mode, S1D13506 DB[15:8] must be connected to TX3912 D[23:16], and S1D13506 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 S1D13506 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 S1D13506 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13506 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.

6.4 Direct Connection to the Toshiba TX3912

The S1D13506 was specifically designed to support the Toshiba MIPS TX3912 processor. When configured, the S1D13506 will utilize one of the PC Card slots supported by the processor.

6.4.1 Hardware Description

In this example implementation, the S1D13506 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 S1D13506 has an internal latch specifically designed for this processor making additional logic unnecessary. To further reduce the need for external components, the S1D13506 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 S1D13506. An optional external oscillator may be used for BUSCLK since the S1D13506 will accept host bus control signals asynchronously with respect to BUSCLK.

The following diagram shows a typical implementation of the interface.

Note: When connecting the S1D13506 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13506 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 6-1 Typical Implementation of Direct Connection

The host interface control signals of the S1D13506 are asynchronous with respect to the S1D13506 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 desired:

• pixel and frame rates.

• power budget.

• part count.

• maximum S1D13506 clock frequencies.

The S1D13506 also has internal CLKI dividers providing additional flexibility.

6.4.2 S1D13506 Configuration

The S1D13506 latches MD15 through MD0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *"S1D13506 Hardware Functional Specification"*.

The table below shows those configuration settings relevant to the Toshiba TX3912 Host Bus Interface.

Table 6-2 S1D13506 Configuration for Direct Connection

S1D13506 Pin Name	Value on this pin at rising edge of RESET# is used to configure:	
	1 (V_DD)	0 (V_SS)
MD[3:1]	111 = Toshiba TX3912 Host Bus Interface if MD11 = 1	
MD4	Little Endian	Big Endian
MD5	WAIT# is active high (1 = insert wait state)	WAIT# is active low (0 = insert wait state)
MD11	Alternate Host Bus Interface selected	Primary Host Bus Interface selected
MD12	BUSCLK input divided by two: use with DCLKOUT	BUSCLK input not divided: use with external oscillator
MD15	WAIT# is always driven	WAIT# is floating if S1D13506 is not selected

| |
|----------| = configuration for Toshiba TX3912 Host Bus Interface

6.4.3 Memory Mapping and Aliasing

The TX3912 uses a portion of the PC Card Attribute and IO space to access the S1D13506. The S1D13506 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 S1D13506 on its PC Card slot as described in the table below.

Table 6-3 TX3912 to PC Card Slots Address Remapping for Direct Connection

S1D13506 Uses PC Card Slot #	Toshiba Address	Size	Function
1	0800 0000h	16M byte	Card 1 IO or Attribute
	0900 0000h	8M byte	S1D13506 registers, aliased 4 times at 2M byte intervals
	0980 0000h	8M byte	S1D13506 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	S1D13506 registers, aliased 4 times at 2M byte intervals
	0D80 0000h	8M byte	S1D13506 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

6.5 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 S1D13506 can be interfaced so as to share one of the PC Card slots.

6.5.1 Hardware Description

The IT8368E can be programmed to allocate the same portion of the PC Card Attribute and IO space to the S1D13506 as in the direct connection implementation described in Section 6.4 "Direct Connection to the Toshiba TX3912" on page 5-45.

Following is a block diagram showing an implementation using the IT8368E PC Card buffer.
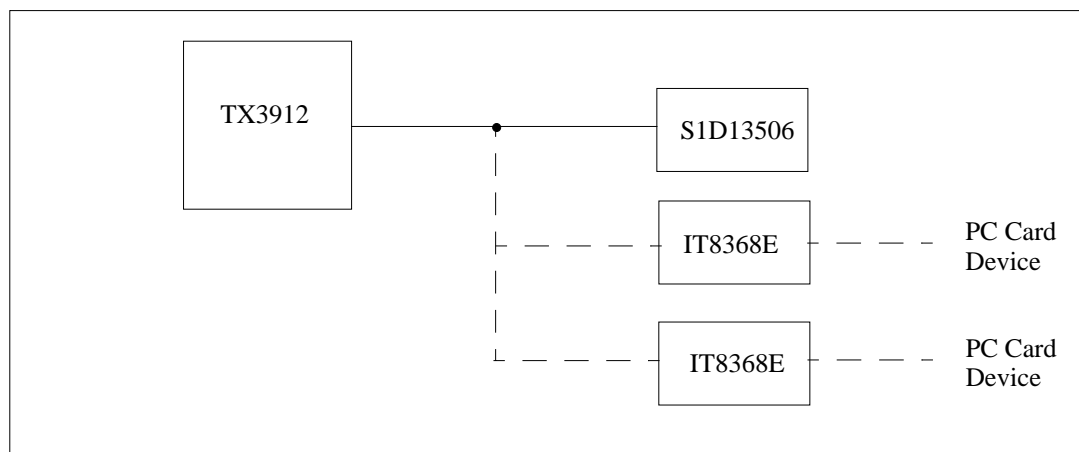
Figure 6-2 IT8368E Implementation Block Diagram

6.5.2 IT8368E Configuration

The ITE IT8368E has been 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 S1D13506 this is not necessary as the Direct Connection described in Section 6.4 "Direct Connection to the Toshiba TX3912" on page 5-45 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 S1D13506.

When the IT8368E senses that the S1D13506 is being accessed, it does not propagate the PC Card signals to its PC Card device. This makes S1D13506 accesses transparent to any PC Card device connected to the same slot.

For mapping details, refer to Section 6.4.3 "Memory Mapping and Aliasing" on page 5-47. For further information on configuring the IT8368E, refer to the *"IT8368E PC Card/GPIO Buffer Chip Specification"*.

6.5.3 S1D13506 Configuration

For S1D13506 configuration, refer to Section 6.4.2 "S1D13506 Configuration" on page 5-46.

6.6 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13506. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13506CFG.EXE, or by directly modifying the source. The Windows® CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

7 INTERFACING TO THE PHILIPS MIPS PR31500/PR31700 PROCESSOR

7.1 Introduction

This application note describes the hardware and software environment necessary to provide an interface between the S1D13506 Color LCD/CRT Controller and the Philips MIPS PR31500/PR31700 Processor.

7.2 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 S1D13506 connects to the PR31500/PR31700 processor.

The S1D13506 can be successfully interfaced using one of the following configurations:

• Direct connection to the PR31500/PR31700 (see Section 6.4 "Direct Connection to the Toshiba TX3912" on page 5-45).

• System design using the ITE IT8368E PC Card/GPIO buffer chip (see Section 6.5 "System Design Using the IT8368E PC Card Buffer" on page 5-48).

7.2.1 S1D13506 Host Bus Interface

The S1D13506 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 S1D13506 on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. For details on S1D13506 configuration, see Section 6.4.2 "S1D13506 Configuration" on page 5-46.

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.

7.2.2 *PR31500/PR31700 Host Bus Interface Pin Mapping*

The following table shows the function of each Host Bus Interface signal.

Table 7-1 PR31500/PR31700 Host Bus Interface Pin Mapping

S1D13506 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#

7.2.3 PR31500/PR31700 Host Bus Interface Signals

When the S1D13506 is configured to operate with the PR31500/PR31700, the Host Bus Interface requires the following signals:

- BUSCLK is a clock input required by the S1D13506 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 PR31500/PR31700 signal /CARDIORD. Either AB18 or the RD# input must be asserted for a read operation to take place.

- Address input AB17 should be connected to the PR31500/PR31700 signal /CARDIOWR. Either AB17 or the WE0# input must be asserted for a write operation to take place.

- 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 6.4.2 "S1D13506 Configuration" on page 5-46). **Because of the PR31500/PR31700 data bus naming convention and endian mode, S1D13506 DB[15:8] must be connected to PR31500/PR31700 D[23:16], and S1D13506 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 S1D13506 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 S1D13506 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13506 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.

7.3 Direct Connection to the Philips PR31500/PR31700

The S1D13506 was specifically designed to support the Philips MIPS PR31500/PR31700 processor. When configured, the S1D13506 will utilize one of the PC Card slots supported by the processor.

7.3.1 Hardware Description

In this example implementation, the S1D13506 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 S1D13506 has an internal latch specifically designed for this processor making additional logic unnecessary. To further reduce the need for external components, the S1D13506 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 S1D13506. An optional external oscillator may be used for BUSCLK since the S1D13506 will accept host bus control signals asynchronously with respect to BUSCLK.

The following diagram shows a typical implementation of the interface.

Note: When connecting the S1D13506 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13506 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 7-1 Typical Implementation of Direct Connection

The host interface control signals of the S1D13506 are asynchronous with respect to the S1D13506 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 desired:

- pixel and frame rates.

- power budget.

- part count.

- maximum S1D13506 clock frequencies.

The S1D13506 also has internal CLKI dividers providing additional flexibility.

7.3.2 S1D13506 Configuration

The S1D13506 latches MD15 through MD0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *"S1D13506 Hardware Functional Specification"*.

The table below shows those configuration settings relevant to the Philips PR31500/PR31700 Host Bus Interface.

Table 7-2 S1D13506 Configuration for Direct Connection

S1D13506 Pin Name	Value on this pin at rising edge of RESET# is used to configure:	
	1 (V$_{DD}$)	0 (V$_{SS}$)
MD[3:1]	111 = Philips PR31500/PR31700 Host Bus Interface when MD11 = 1	
MD4	Little Endian	Big Endian
MD5	WAIT# is active high (1 = insert wait state)	WAIT# is active low (0 = insert wait state)
MD11	Alternate Host Bus Interface selected	Primary Host Bus Interface selected
MD12	BUSCLK input divided by two: use with DCLKOUT	BUSCLK input not divided: use with external oscillator
MD15	WAIT# is always driven	WAIT# is floating if S1D13506 is not selected
	= configuration for Philips PR31500/PR31700 Host Bus Interface	

7.3.3 Memory Mapping and Aliasing

The PR31500/PR31700 uses a portion of the PC Card Attribute and IO space to access the S1D13506. The S1D13506 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 S1D13506 on its PC Card slot as described in the table below.

Table 7-3 PR31500/PR31700 to PC Card Slots Address Remapping for Direct Connection

S1D13506 Uses PC Card Slot #	Philips Address	Size	Function
1	0800 0000h	16M byte	Card 1 IO or Attribute
	0900 0000h	8M byte	S1D13506 registers, aliased 4 times at 2M byte intervals
	0980 0000h	8M byte	S1D13506 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	S1D13506 registers, aliased 4 times at 2M byte intervals
	0D80 0000h	8M byte	S1D13506 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

7.4 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 S1D13506 can be interfaced so as to share one of the PC Card slots.

7.4.1 Hardware Description

The IT8368E can be programmed to allocate the same portion of the PC Card Attribute and IO space to the S1D13506 as in the direct connection implementation described in Section 6.4 "Direct Connection to the Toshiba TX3912" on page 5-45.

Following is a block diagram showing an implementation using the IT8368E PC Card buffer.
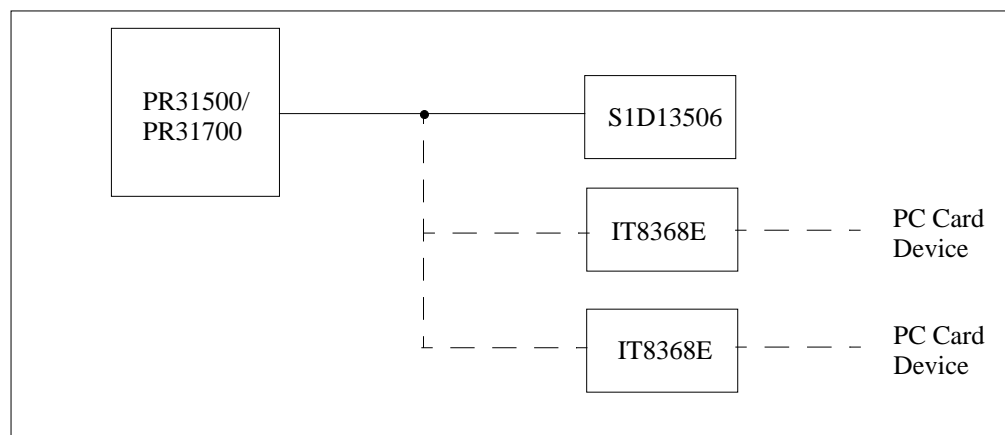
Figure 7-2 IT8368E Implementation Block Diagram

7.4.2 IT8368E Configuration

The ITE IT8368E has been 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 S1D13506 this is not necessary as the Direct Connection described in Section 6.4 "Direct Connection to the Toshiba TX3912" on page 5-45 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 S1D13506.

When the IT8368E senses that the S1D13506 is being accessed, it does not propagate the PC Card signals to its PC Card device. This makes S1D13506 accesses transparent to any PC Card device connected to the same slot.

For mapping details, refer to Section 6.4.3 "Memory Mapping and Aliasing" on page 5-47. For further information on configuring the IT8368E, refer to the *"IT8368E PC Card/GPIO Buffer Chip Specification"*.

7.4.3 S1D13506 Configuration

For S1D13506 configuration, refer to Section 6.4.2 "S1D13506 Configuration" on page 5-46.

7.5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13506. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13506CFG.EXE, or by directly modifying the source. The Windows® CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

8 INTERFACING TO THE STRONGARM SA-1110 PROCESSOR

8.1 Introduction

This application note describes the hardware and software environment required to provide an interface between the S1D13506 Color LCD/CRT/TV Controller and the Intel StrongARM SA-1110.

8.2 Interfacing to the StrongARM SA-1110 Bus

8.2.1 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 S1D13506 LCD controller and provides a high performance, power efficient solution for embedded systems.

8.2.2 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.

8.2.3 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 over 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 will remain 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 8-1 illustrates a typical variable-latency IO access read cycle on the SA-1110 bus.
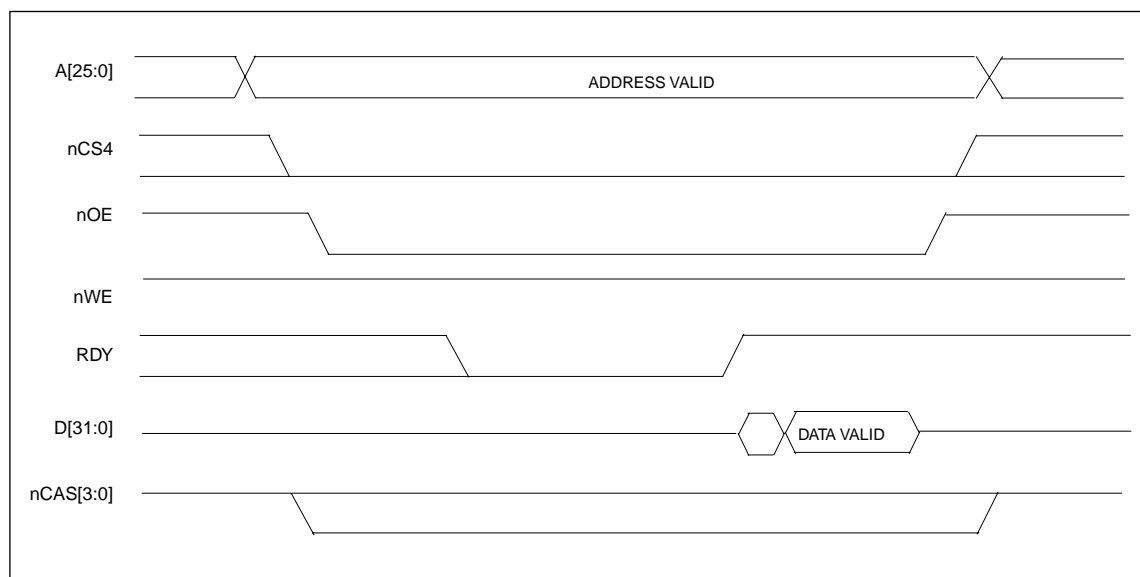
Figure 8-1 SA-1110 Variable-Latency IO Read Cycle

Figure 1-2 illustrates a typical variable-latency IO access write cycle on the SA-1110 bus.
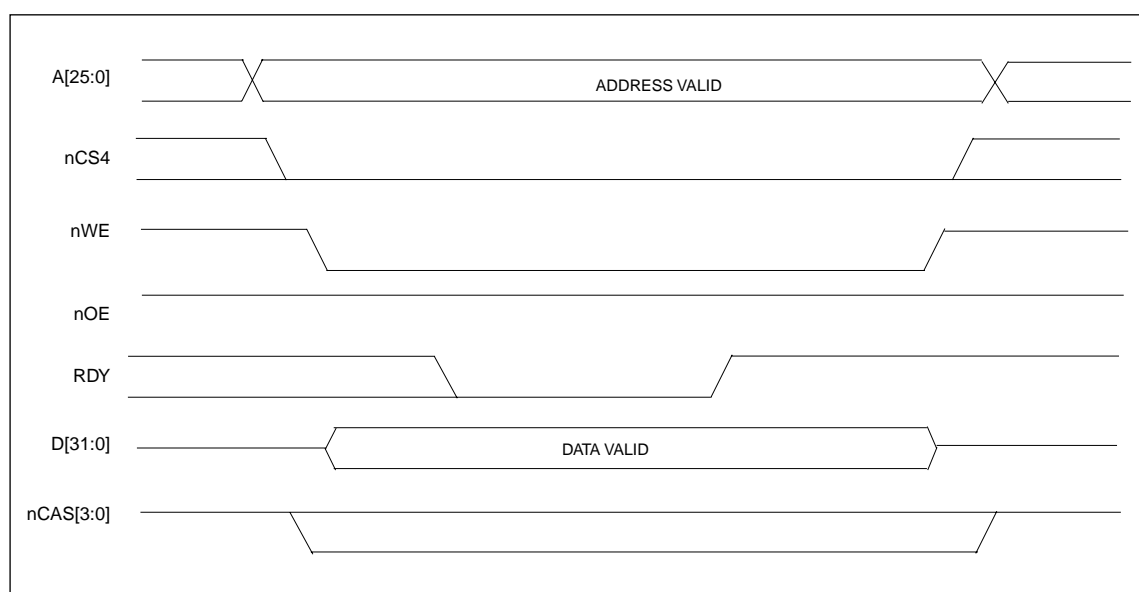
Figure 8-2 SA-1110 Variable-Latency IO Write Cycle

8.3 S1D13506 Host Bus Interface

The S1D13506 directly supports multiple processors. The S1D13506 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 S1D13506 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13506 configuration, see Section 1.4.2 "S1D13506 Hardware Configuration" on page 5-7.

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.

8.3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 8-1 Host Bus Interface Pin Mapping

S1D13506 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 S1D13506 internally.

8.3.2 Host Bus Interface Signal Descriptions

The S1D13506 PC Card Host Bus Interface requires the following signals.

- BUSCLK is a clock input which is required by the S1D13506 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 8.4.3 "Performance" on page 5-64.

- 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. MD4 must be set to select little endian mode upon reset.

- 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 S1D13506 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 S1D13506.

- 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 S1D13506 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 S1D13506 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13506 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete. For the SA-1110, this signal should be set active low using the MD5 configuration input.

- 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 S1D13506 may be connected to the system RESET.

8.4 StrongARM SA-1110 to S1D13506 Interface

8.4.1 Hardware Description

The S1D13506 is designed to directly support a variety of CPUs, providing an interface to each processor's unique "local bus". Using the S1D13506'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 S1D13506 interface.
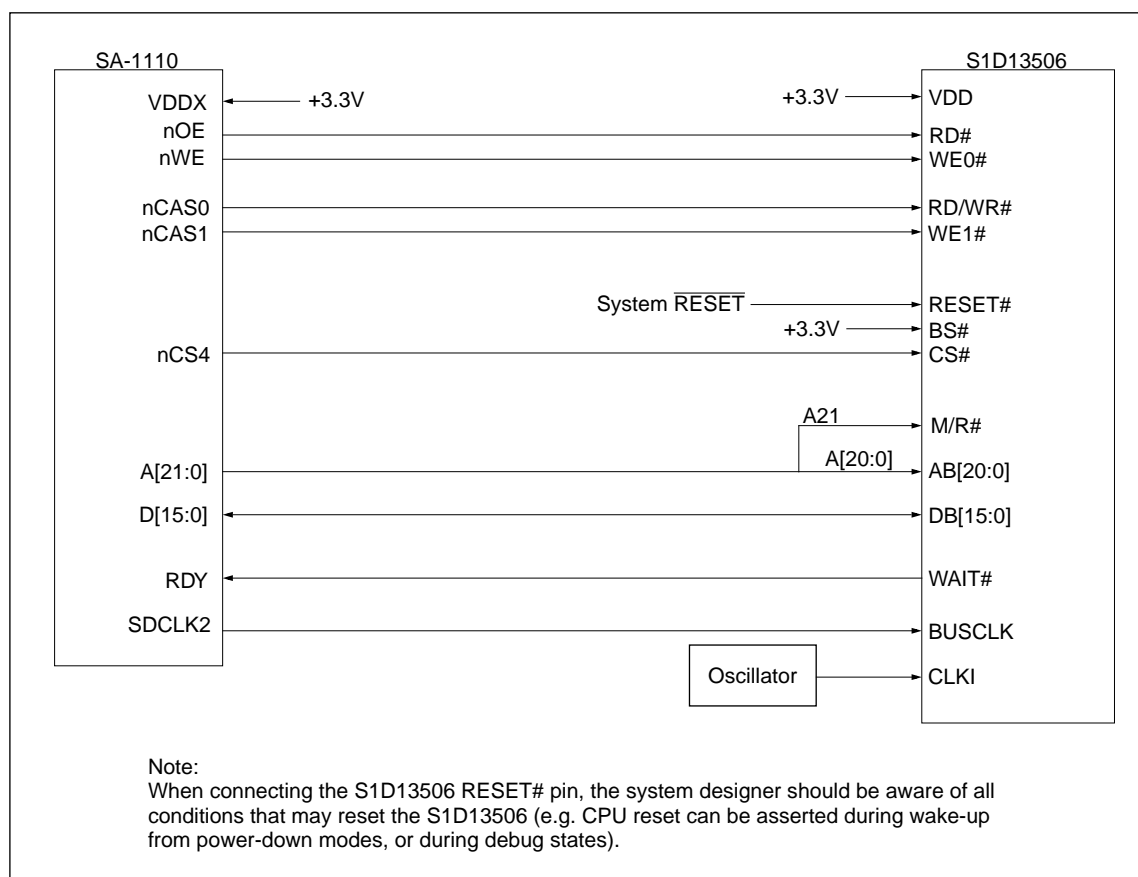
Figure 8-3 Typical Implementation of SA-1110 to S1D13506 Interface

8.4.2 S1D13506 Hardware Configuration

The S1D13506 latches MD15 through MD0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13506 Hardware Functional Specification*.

The table below shows only those configuration settings important to the PC Card Host Bus Interface.

Table 8-2 Summary of Power-On/Reset Options

S1D13506 Pin Name	value on this pin at rising edge of RESET# is used to configure:(1/0)	
	1	0
MD[3:1]	111 = PC Card Host Bus Interface selected	
MD4	Little Endian	Big Endian
MD5	WAIT# is active high (1 = insert wait state)	WAIT# is active low (0 = insert wait state)
MD11	Alternate Host Bus Interface Selected	Primary Host Bus Interface Selected
MD12	BUSCLK input divided by two	BUSCLK input not divided by two
MD15	WAIT# is always driven	WAIT# is tristated when the chip is not accessed by the host

[shaded] = configuration for PC Card Host Bus Interface

8.4.3 Performance

The S1D13506 PC Card Interface specification supports a BUSCLK up to 50MHz, and therefore provides a high performance display solution.

The BUSCLK signal input to the S1D13506 (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 S1D13506 has a maximum BUSCLK of 50MHz, the output clock from the SA-1110 must be a divided down from the processor clock. The DRAM Refresh Control Register (MDREFR) 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.

8.4.4 Register/Memory Mapping

The S1D13506 is a memory mapped device. The SA-1110 will use the memory assigned to a chip select (nCS4 in this example) to map the S1D13506 internal registers and display buffer. The internal registers require 2M bytes of memory and are mapped to the lower memory address space starting at zero. The display buffer also requires 2M bytes and is mapped in the third and fourth megabytes (ranging from 200000h to 3FFFFFh).

A typical implementation as shown in Figure 1-3, "Typical Implementation of PC Card to S1D13506 Interface" on page 5-6 has Chip Select (CS#) connected to nCS4 and the Memory/ Register select pin (M/R#) connected to address bit A21. This provides the following decoding:

Table 8-3 Register/Memory Mapping for Typical Implementation

M/R# (A21)	Address Range	Function
0	0 - 1F FFFFh	Internal Registers
1	20 0000h - 3F FFFFh	Display Buffer

Each chip select on the SA-1110 provides 64M bytes of address space. Without further resolution of the decoding logic (M/R# connected to A21), the entire register set and display buffer are aliased for every 4M byte boundary within the specified address range of the chip select. Since address bits A[25:22] are ignored, the S1D13506 registers and display buffer are aliased 16 times.

Note: If aliasing is not desirable, the upper addresses must be fully decoded.

8.4.5 StrongARM SA-1110 Register Configuration

The SA-1110 requires configuration of several of its internal registers to interface to the S1D13506 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 Max. desired CPU frequency as indicated in the table below.

Table 8-4 RDFx Parameter Value versus CPU Max. Frequency

CPU Frequency	RDFx
147.5MHz	2
206.4MHz	3
Up to SA-1110 Max.	4

Parameter RDNx<4:0> should be set to 0 (Min. command precharge time).

Parameter RRRx<2:0> should be set to 0 (Min. nCSx precharge time).

• The S1D13506 endian mode is set to little endian. To program the SA-1110 for little endian as well use the control register (register 1).

  • Bit 7 must be set to 0.

• The BUSCLK signal input to the S1D13506 (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 S1D13506 has a Max. BUSCLK of 50MHz, the output clock from the SA-1110 must be a divided down from the processor clock. The DRAM Refresh Control Register (MDREFR) 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.

8.5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13506. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13506CFG.EXE, or by directly modifying the source. The Windows® CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

9 POWER CONSUMPTION

9.1 S1D13506 Power Consumption

S1D13506 power consumption is affected by many system design variables.

- Input clock frequency (CLKI/CLKI2): the CLKI/CLKI2 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 S1D13506 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 S1D13506. The power consumption is affected by various system design variables.

- DRAM refresh mode (CBR or self-refresh): self-refresh capable DRAM allows the S1D13506 to disable the internal memory clock thereby saving power.

- Clock states during the power save mode: disabling the clocks during power save mode has substantial power savings.

9.1.1 Conditions

Table 9-1 gives an example of a specific environment and its effects on power consumption.

Table 9-1 S1D13506 Power Consumption

	Test Condition V_{DD} = 3.3V, BUSCLK = 8MHz	Color Depth	S1D13506 Active (mW)	Power Save Mode	
				Clocks Active (mW) [1]	Clocks Removed (mW) [2]
1	CLKI = 6MHz LCD Panel = 60Hz 320×240 4-bit Single Monochrome	4 bpp	21.22	3.14	.13
2	CLKI = 6 MHz LCD Panel = 60Hz 320×240 8-bit Single Color	4 bpp	23.30	3.14	.13
		8 bpp	24.98		
		16 bpp	26.73		
3	CLKI = 25MHz LCD Panel = 60Hz 640×480 8-bit Dual Monochrome	4 bpp	70.75	10.36	.13
4	CLKI = 25MHz LCD Panel = 60Hz 640×480 16-bit Dual Color	4 bpp	91.74	10.40	.13
		8 bpp	96.89		
		16 bpp	99.03		
5	CLKI = 33.333MHz, CLKI2 = 25.175MHz CRT = 60Hz 640x480 Color	4 bpp	224.86	11.72	.13
		8 bpp	233.44		
		16 bpp	242.68		
6	CLKI = 33.333MHz, CLKI2 = 14.31818MHz NTSC TV = 640×480 Color, S-Video output, no filter	4 bpp	359.83	10.00	.13
		8 bpp	365.11		
		16 bpp	372.04		
7	CLKI = 33.333MHz, CLKI2 = 17.734475MHz PAL TV = 640×480 Color, S-Video output, no filter	4 bpp	364.78	10.53	.13
		8 bpp	369.07		
		16 bpp	374.02		

Note: 1. Conditions for software suspend with Clocks active:
- CPU interface inactive
- CLKI, CLKI2, BUSCLK active
- Self-Refresh DRAM

2. Conditions for software suspend with Clocks inactive:
- CPU interface inactive
- CLKI, CLKI2, BUSCLK stopped
- Self-Refresh DRAM

9.2 Summary

The system design variables in Section 9.1, *"S1D13506 Power Consumption"* and in Table 9-1 show that S1D13506 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 S1D13506 can be configured to be an extremely power-efficient LCD/CRT/TV Controller with high performance and flexibility.

S1D13506F00A
Color LCD/CRT/TV Controller

Windows® CE
Display Drivers

Contents

1 Windows® CE DISPLAY DRIVERS

The Windows® CE display drivers are designed to support the S1D13506 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°, 180° and 270° modes.

1.1 Program Requirements

Video Controller	:	S1D13506
Display Type	:	LCD or CRT
Windows® Version	:	CE Version 2.0 and 2.11

1.2 Example Driver Builds

1.2.1 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 S5U13506P00C 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 1.2.1 "Build for CEPC (X86) on Windows® CE 2.0" on page 6-2.

5. Create a sub-directory named S1D13506 under \wince\platform\cepc\drivers\display.

6. Copy the source code to the S1D13506 subdirectory.

7. Add an entry for the S1D13506 in the file \wince\platform\cepc\drivers\display\dirs.

EPSON

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

ENDIFY

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 S1D13506 utility program 13506CFG.EXE. The file MODE0.H (located in X:\wince\platform\odo\drivers\display\S1D13506) 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. The display driver section of PLATFORM.REG should be as follows:

    ; 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\S1D13506]

    "CxScreen"=dword:280

    "CyScreen"=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).

1.2.2 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".

   g. Rename the icon "Build Minshell for x86" to "Build Epson for x86" by right clicking on the icon and choosing "rename".

   h. Right click on the icon "Build Epson for x86" and click on "Properties" to bring up the "Build Epson for x86 Properties" window.

   i. Click on "Shortcut" and replace the string "Minshelll" under the entry "Target" with "Epson".

   j. 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_S1D13506=1

   @echo off

5. Make a S1D13506 directory under \WINCE211\PLATFORM\CEPC\DRIVERS\DISPLAY, and copy the S1D13506 driver source code into \WINCE211\PLATFORM\CEPC\DRIVERS\DISPLAY\S1D13506.

6. Add S1D13506 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_S1D13506

        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_S1D13506 !

    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 S1D13506 utility program 13506CFG.EXE. The file MODE0.H (located in X:\wince\platform\odo\drivers\display\S1D13506) 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. The display driver section of PLATFORM.REG should be:

> ; 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\S1D13506]
>
> "CxScreen"=dword:280
>
> "CyScreen"=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 /D:2 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 /D:2 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 2.11supported 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 color dual passive 16-bit LCD panel at 8 bpp landscape mode. To support other settings, use 13506CFG.EXE to generate the proper MODE0.H file. For further information, refer to Chapter3, "2 13506CFG.EXE Configuration Program" on page 3-6.

- 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 2.11. The drivers are will be updated as appropriate.

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

DÜSSELDORF BRANCH OFFICE

Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone: +49-(0)2171-5045-0   Fax: +49-(0)2171-5045-10

UK & IRELAND 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
Phon:+34-93-544-2490   Fax:+34-93-544-2491

Scotland Design Center
Integration House, The Alba Campus
Livingston West Lothian, EH54 7EG, SCOTLAND
Phone: +44-1506-605040   Fax: +44-1506-605041

ASIA

EPSON (CHINA) CO., LTD.
23F, Beijing Silver Tower 2# North RD DongSanHuan
ChaoYang District, Beijing, CHINA
Phone: 64106655   Fax: 64107319

SHANGHAI BRANCH

7F, High-Tech Bldg., 900, Yishan Road,
Shanghai 200233, CHINA
Phone: 86-21-5423-5577   Fax: 86-21-5423-4677

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.

14F, No. 7, Song Ren Road,
Taipei 110
Phone: 02-8786-6688   Fax: 02-8786-6660

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: +65-6337-7911   Fax: +65-6334-2716

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

GUMI OFFICE

6F, Good Morning Securities Bldg.
56 Songjeong-Dong, Gumi-City, 730-090, KOREA
Phone: 054-454-6027   Fax: 054-454-6093

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

IC 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
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5814   Fax: +81-(0)42-587-5117

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.

S1D13506F00A
Technical Manual

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ EPSON Electronic Devices Website

http://www.epsondevice.com

This manual was made with recycle papaer,
and printed using soy-based inks.