EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

S1D13706
TECHNICAL MANUAL

Document Number: X31B-Q-001-06

THIS PAGE LEFT BLANK

COMPREHENSIVE SUPPORT TOOLS

EPSON provides the designer and manufacturer a complete set of resources and tools for the development of LCD Graphics Systems.

Documentation

- Technical manuals
- Evaluation/Demonstration board manual

Evaluation/Demonstration Board

- Assembled and fully tested Graphics Evaluation/Demonstration board
- Schematic of Evaluation/Demonstration board
- Parts List
- Installation Guide
- CPU Independent Software Utilities
- Evaluation Software
- Display Drivers

Application Engineering Support

EPSON offers the following services through their Sales and Marketing Network:

- Sales Technical Support
- Customer Training
- Design Assistance

Application Engineering Support

Engineering and Sales Support is provided by:

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

THIS PAGE LEFT BLANK

March 2001

S1D13706 Embedded Memory LCD Controller

The S1D13706 is a color/monochrome LCD graphics controller with an embedded 80K byte SRAM display buffer. While supporting all other panel types, the S1D13706 is the only LCD controller to directly interface to both the Epson D-TFD and the Sharp HR-TFT family of products thus removing the requirement of an external Timing Control IC. This high level of integration provides a low cost, low power, single chip solution to meet the demands of embedded markets such as Mobile Communications devices and Palm-size PCs, where board size and battery life are major concerns.

The S1D13706 utilizes a guaranteed low-latency CPU architecture thus providing support for microprocessors without READY/WAIT# handshaking signals. The 32-bit internal data path provides high performance bandwidth into display memory allowing for fast screen updates.

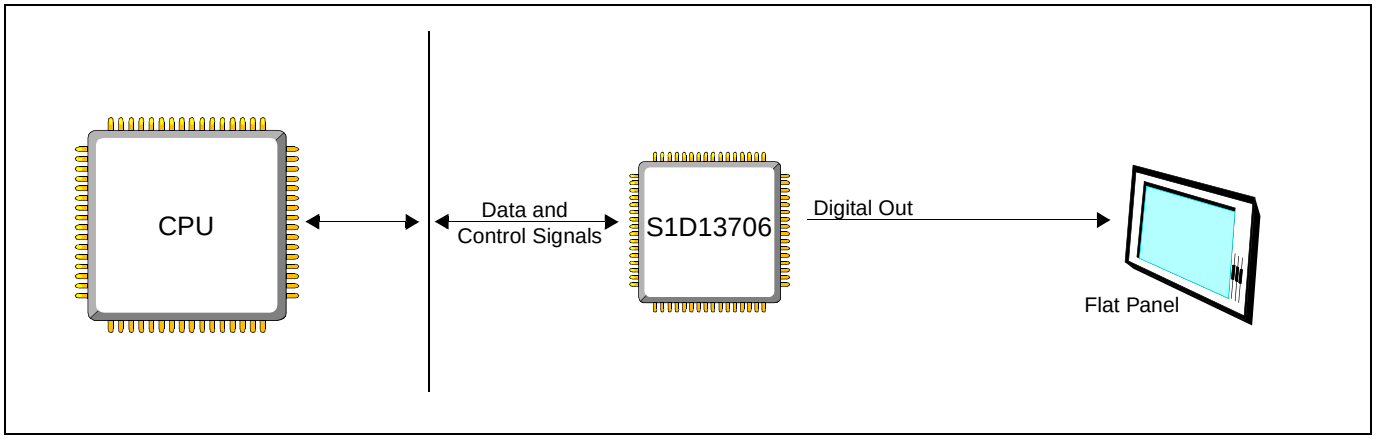
Products requiring a rotated display image can take advantage of the SwivelView™ feature which provides hardware rotation of the display memory transparent to the software application. The S1D13706 also provides support for “Picture-in-Picture Plus” (a variable size Overlay window).

The S1D13706 provides impressive support for Palm OS® handhelds, however its impartiality to CPU type or operating system makes it an ideal display solution for a wide variety of applications.

FEATURES

- Embedded Display Buffer.
- Low Operating Voltage.
- Low-latency CPU interface.
- Direct support for the multiple CPU types.
- Programmable Resolutions and Color depths.
- STN LCD support.
- Active Matrix LCD support.
- Reflective Active Matrix support.
- SwivelView™ (90, 180, 270° hardware rotation of displayed image).
- “Picture-in-Picture Plus”.
- Software Initiated Power Save Mode.
- Hardware or Software Video Invert.
- 100-pin TQFP15 package.
- 104-pin CFLGA package.

SYSTEM BLOCK DIAGRAM



■ DESCRIPTION

Memory Interface

- Embedded 80K byte SRAM display buffer.

CPU Interface

- 'Fixed' low-latency CPU access times.
- Direct support for:
  Hitachi SH-4 / SH-3.
  Motorola M68xxx (REDCAP2, DragonBall, ColdFire).
  MPU bus interface with programmable READY.

Display Support

- 4/8-bit monochrome LCD interface.
- 4/8/16-bit color STN LCD interface.
- Single-panel, single-drive passive displays.
- 9/12/18-bit Active matrix TFT interface.
- 'Direct' support for Epson D-TFD and Sharp HR-TFT (external timing control IC not required).
- Typical resolutions supported:
  - 320x240@8bpp
  - 160x160 @16bpp
  - 160x240 @16bpp

Clock Source

- Two clock inputs (single clock possible).
- Clock source can be internally divided down for a higher frequency clock input.

Display Modes

- 1/2/4/8/16 bit-per-pixel (bpp) support.
- Up to 64 gray shades using FRM and dithering on monochrome passive LCD panels.
- Up to 64K colors on passive STN panels.
- Up to 64K colors on active matrix panels.
- SwivelView: direct hardware rotation of display image by 90°, 180°, 270°.
- "Picture-in-Picture Plus": displays a variable size window overlaid over background image.
- Double Buffering/multi-pages: provides smooth animation and instantaneous screen update.

Power Down Modes

- Software Initiated Power Save Mode.

Operating Voltage

- $CORE_{VDD}$ 1.8 to 2.2 volts and 3.0 to 3.6 volts.
- HIO_{VDD} 1.8 to 2.2 volts and 3.0 to 3.6 volts.
- NIO_{VDD} 3.0 to 3.6 volts.

Package

- 100-pin TQFP15.
- 104-pin CFLGA.

CONTACT YOUR SALES REPRESENTATIVE FOR THESE COMPREHENSIVE DESIGN TOOLS

- S1D13706 Technical Manual
- S5U13706 Evaluation Boards
- CPU Independent Software Utilities
- Palm OS® Hardware Abstraction Layer
- Windows® CE Display Driver
- VXWorks® Tornado™ Display Driver

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Taiwan
Epson Taiwan Technology & Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

VDC

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Hardware Functional Specification

Document Number: X31B-A-001-06

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

1 Introduction

1.1 Scope

This is the Hardware Functional Specification for the S1D13706 Embedded Memory LCD 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.

For additional documentation related to the S1D13706 see Section 17, “References” on page 150.

This document is updated as appropriate. Please check the Epson Electronics America website at **www.eea.epson.com** or the Epson Research and Development Website at **www.erd.epson.com** for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

1.2 Overview Description

The S1D13706 is a color/monochrome LCD graphics controller with an embedded 80K byte SRAM display buffer. While supporting all other panel types, the S1D13706 is the only LCD controller to directly interface to both the Epson D-TFD and the Sharp HR-TFT family of products thus removing the requirement of an external Timing Control IC. This high level of integration provides a low cost, low power, single chip solution to meet the demands of embedded markets such as Mobile Communications devices and Palm-size PCs, where board size and battery life are major concerns.

The S1D13706 utilizes a guaranteed low-latency CPU architecture providing support for microprocessors without READY/WAIT# handshaking signals. The 32-bit internal data path provides high performance bandwidth into display memory allowing for fast screen updates.

Products requiring a rotated display image can take advantage of the SwivelView ™ feature which provides hardware rotation of the display memory transparent to the software application. The S1D13706 also provides support for “Picture-in-Picture Plus” (a variable size Overlay window).

The S1D13706 provides impressive support for Palm OS® handhelds, however its impartiality to CPU type or operating system makes it an ideal display solution for a wide variety of applications.

2 Features

2.1 Integrated Frame Buffer

- Embedded 80K byte SRAM display buffer.

2.2 CPU Interface

- Direct support of the following interfaces:
  Generic MPU bus interface using WAIT# signal.
  Hitachi SH-3.
  Hitachi SH-4.
  Motorola M68K.
  Motorola MC68EZ328/MC68VZ328 DragonBall.
  Motorola “REDCAP2” - no WAIT# signal.
- 8-bit processor support with “glue logic”.
- “Fixed” low-latency CPU access times.
- Registers are memory-mapped - M/R# input selects between memory and register address space.
- The complete 80K byte display buffer is directly and contiguously available through the 17-bit address bus.
- Single level CPU write buffer.

2.3 Display Support

- Single-panel, single-drive passive displays.
  - 4/8-bit monochrome LCD interface.
  - 4/8/16-bit color LCD interface.
- Active Matrix TFT interface.
  - 9/12/18-bit interface.
- ‘Direct’ support for 18-bit Epson D-TFD interface.
- ‘Direct’ support for 18-bit Sharp HR-TFT interface.

2.4 Display Modes

- 1/2/4/8/16 bit-per-pixel (bpp) color depths.
- Up to 64 gray shades using Frame Rate Modulation (FRM) and dithering on monochrome passive LCD panels.
- Up to 64K colors on passive STN panels.
- Up to 64K colors on active matrix LCD panels.
- Example resolutions:
  320x240 at a color depth of 8 bpp
  160x160 at a color depth of 16 bpp
  160x240 at a color depth of 16 bpp

2.5 Display Features

- SwivelView™: 90°, 180°, 270° counter-clockwise hardware rotation of display image.
- "Picture-in-Picture Plus": displays a variable size window overlaid over background image.
- Double Buffering/Multi-pages: provides smooth animation and instantaneous screen updates.

2.6 Clock Source

- Two clock inputs: CLKI and CLKI2. It is possible to use one clock input only.
- Bus clock is derived from CLKI and can be internally divided by 2, 3, or 4.
- Memory clock is derived from bus clock. It can be internally divided by 2, 3, or 4.
- Pixel clock can be derived from CLKI, CLKI2, bus clock, or memory clock. It can be internally divided by 2, 3, 4, or 8.

2.7 Miscellaneous

- Hardware/Software Video Invert.
- Software Power Save mode.
- General Purpose Input/Output pins are available.
- 100-pin TQFP15 package.
- 104-pin CFLGA package.

3 Typical System Implementation Diagrams

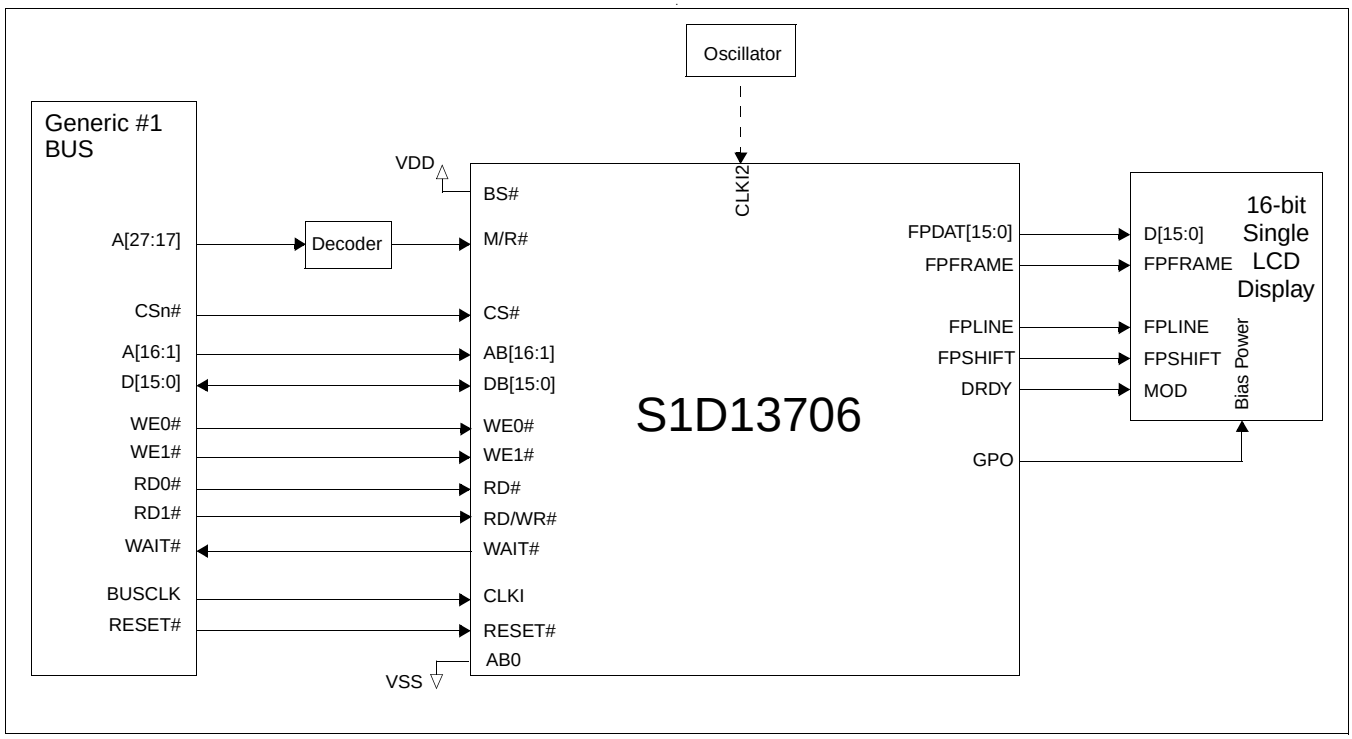


Figure 3-1: Typical System Diagram (Generic #1 Bus)

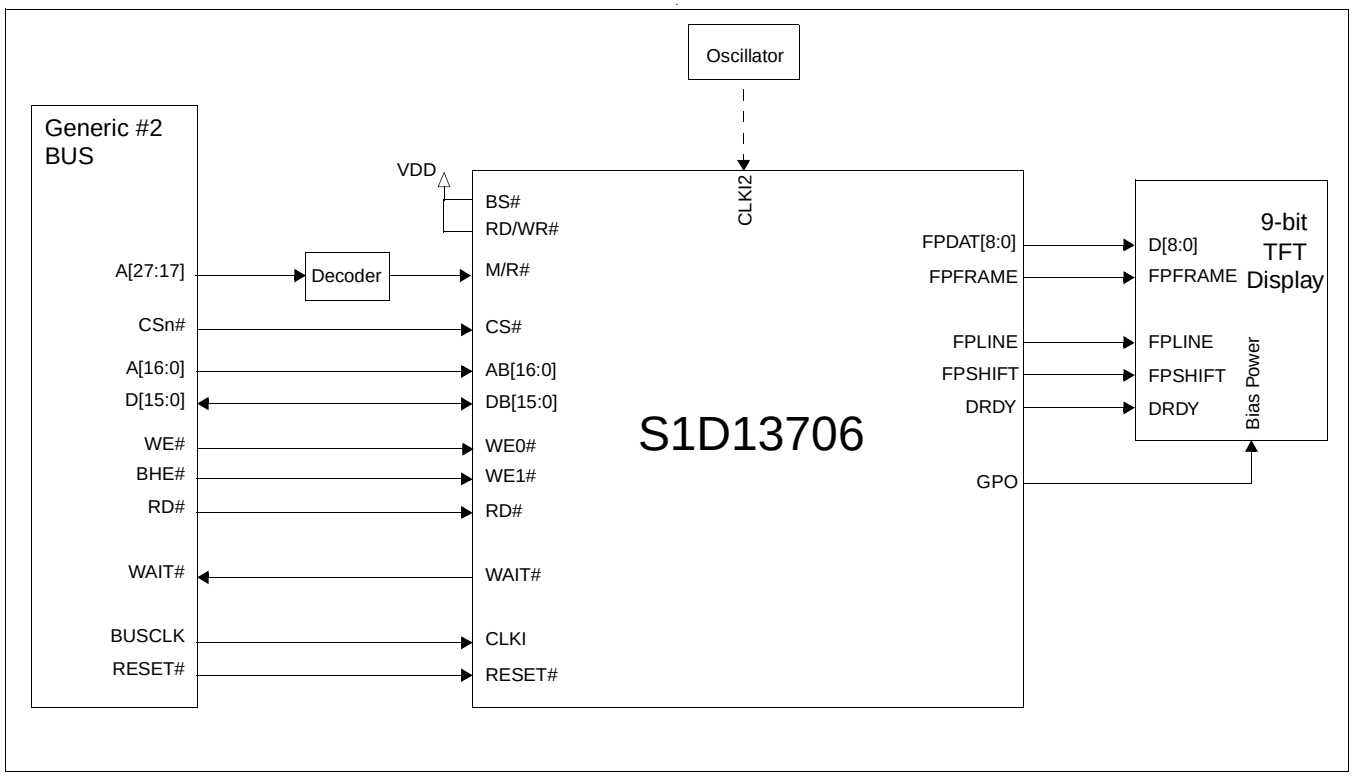


Figure 3-2: Typical System Diagram (Generic #2 Bus)

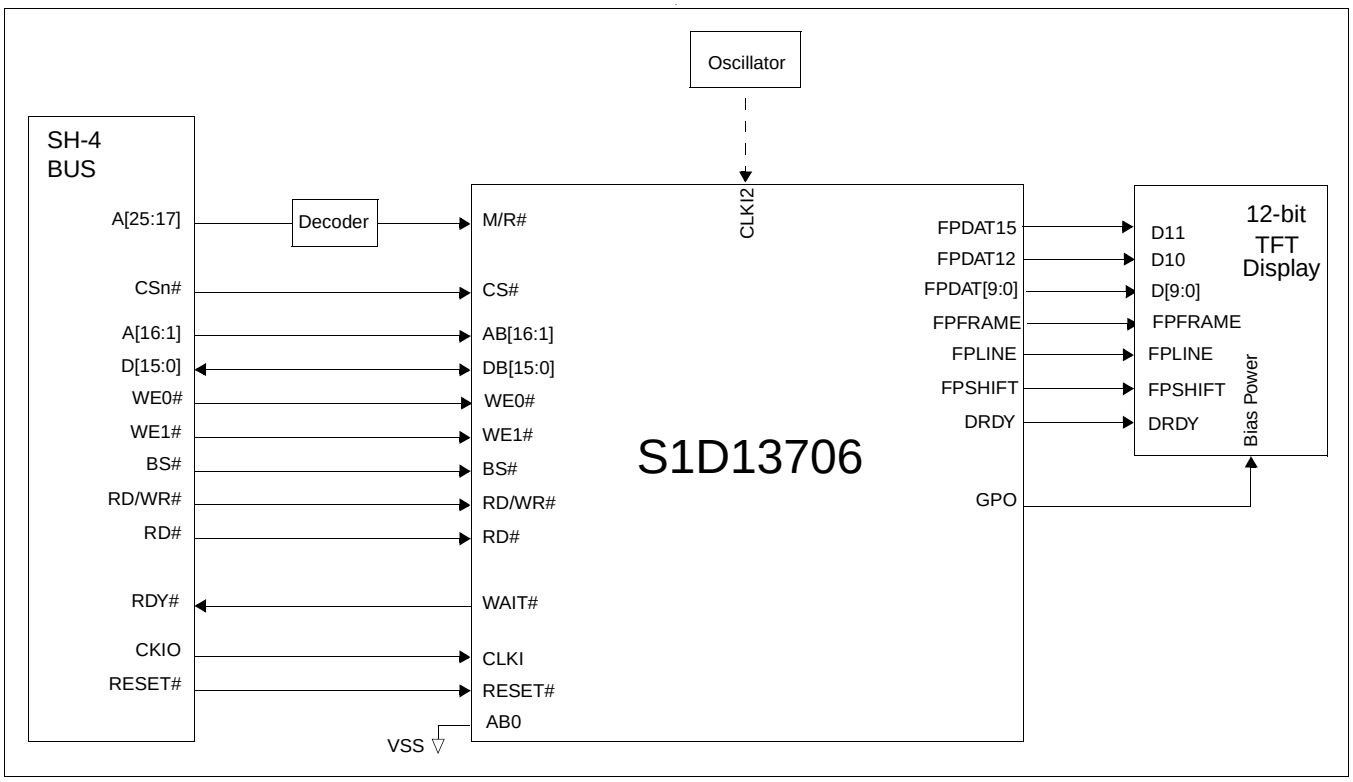


Figure 3-3: Typical System Diagram (Hitachi SH-4 Bus)

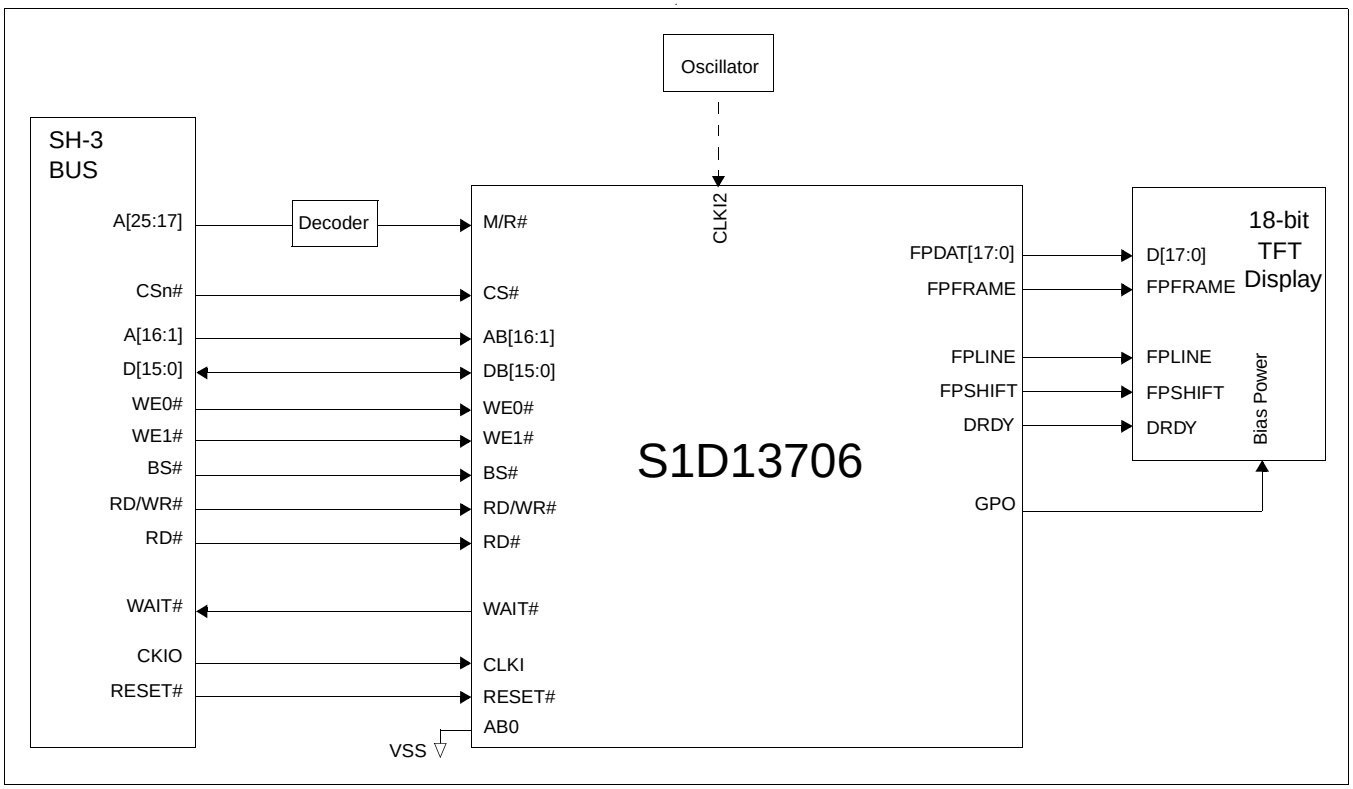


Figure 3-4: Typical System Diagram (Hitachi SH-3 Bus)

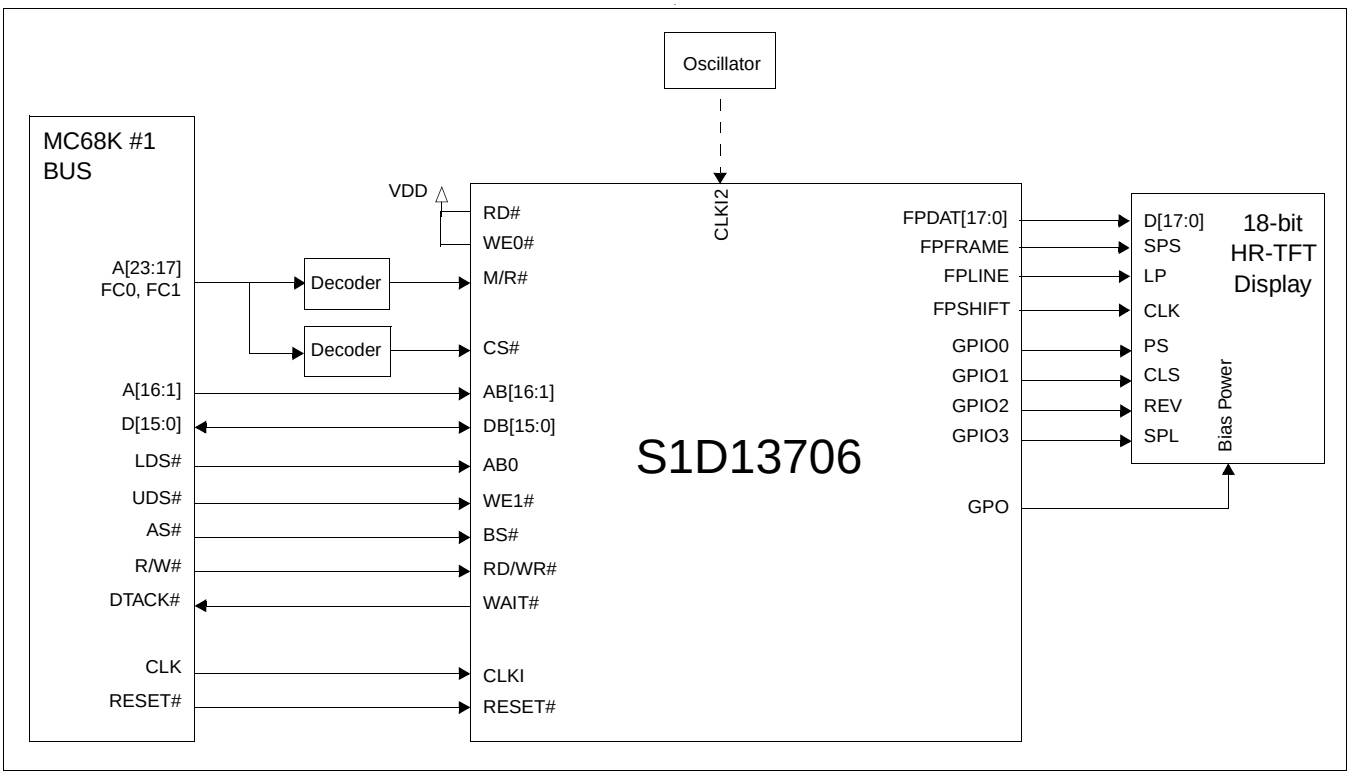


Figure 3-5: Typical System Diagram (MC68K # 1, Motorola 16-Bit 68000)

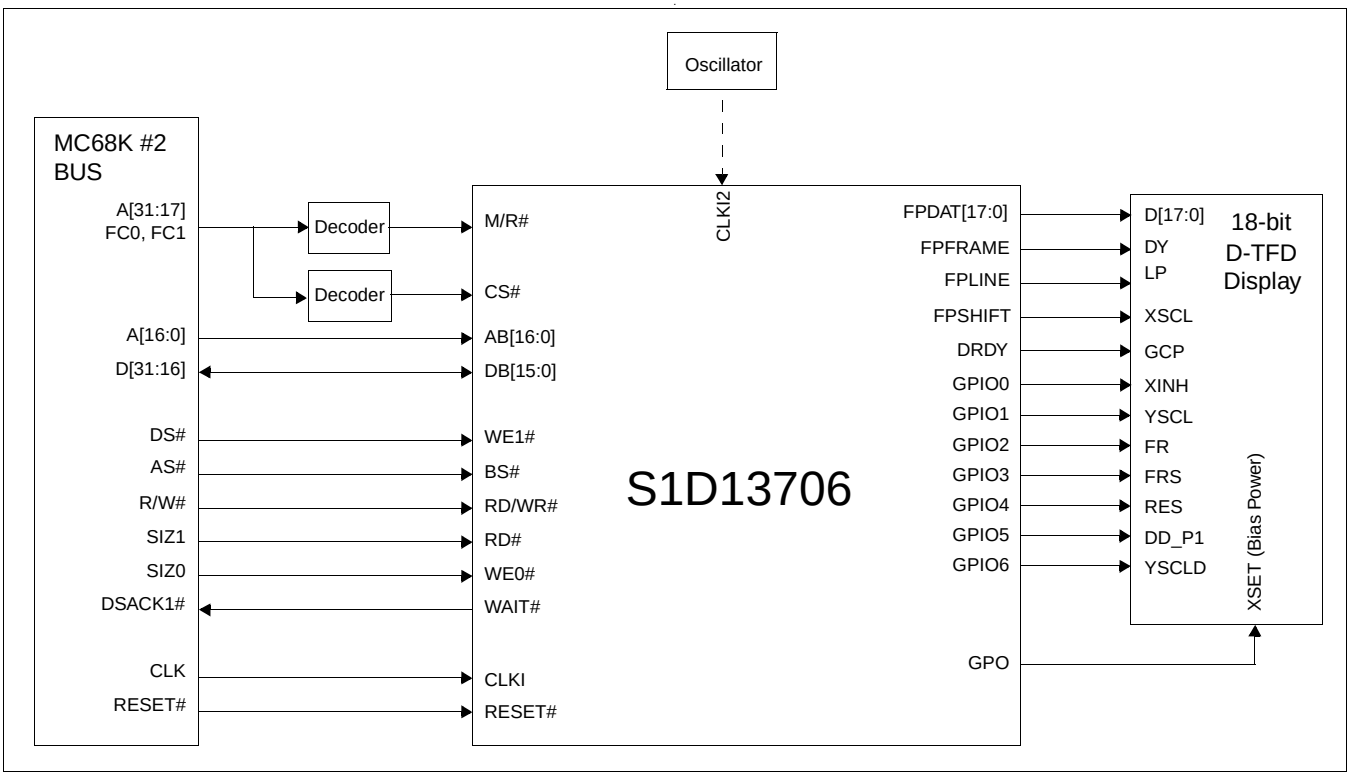


Figure 3-6: Typical System Diagram (MC68K #2, Motorola 32-Bit 68030)

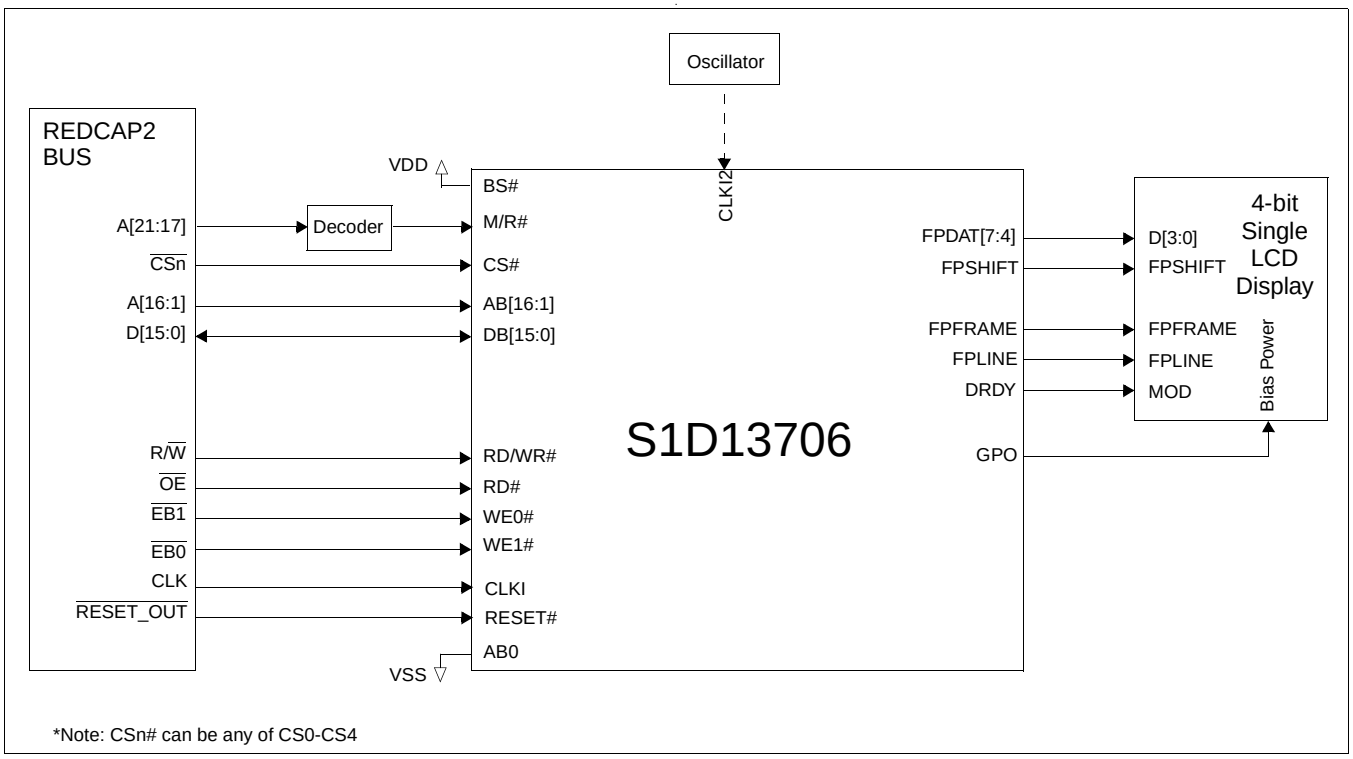


Figure 3-7: Typical System Diagram (Motorola REDCAP2 Bus)

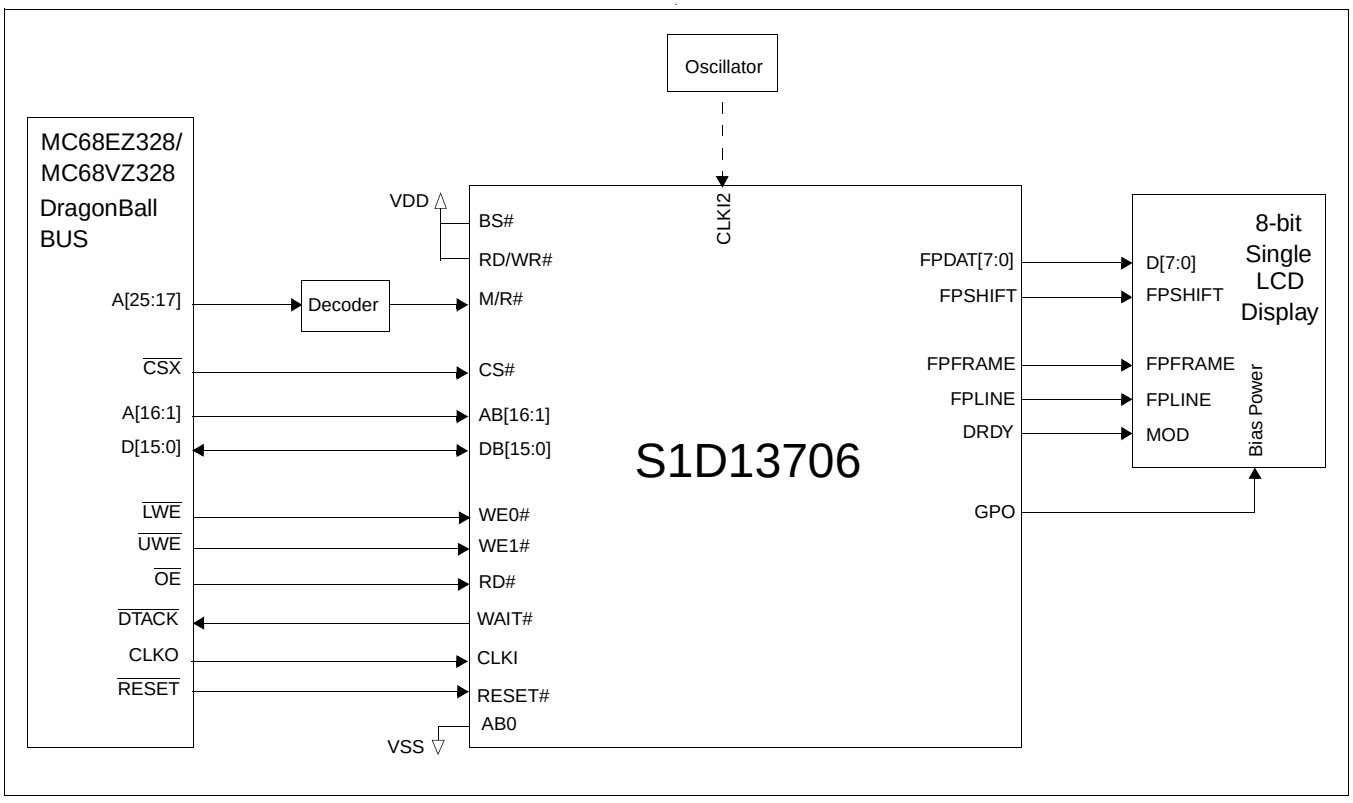


Figure 3-8: Typical System Diagram (Motorola MC68EZ328/MC68VZ328 "DragonBall" Bus)

4 Pins

4.1 Pinout Diagram - TQFP15 - 100pin

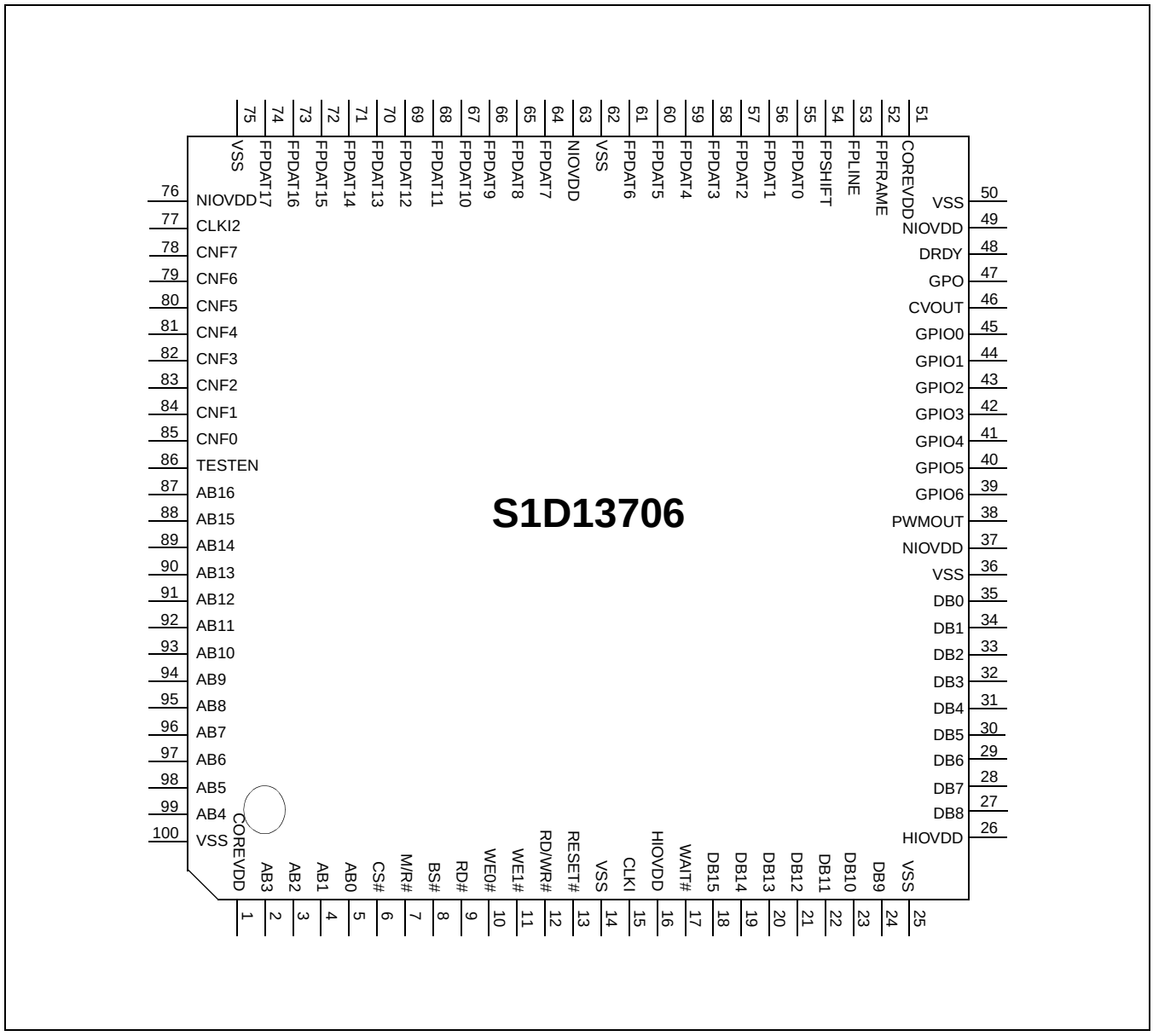


Figure 4-1: Pinout Diagram - TQFP15 - 100pin

Note
Package type: 100 pin surface mount TQFP15

4.2 Pinout Diagram - CFLGA - 104pin

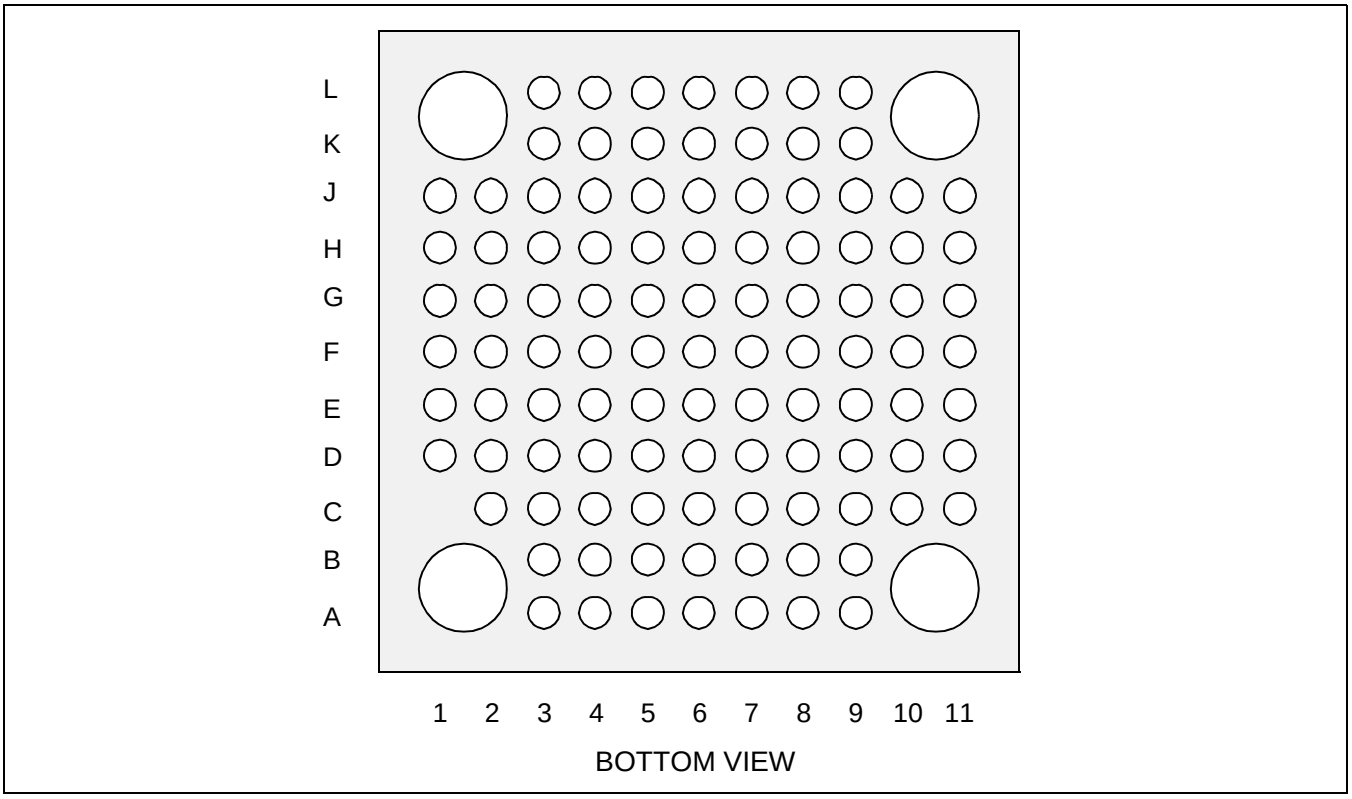


Figure 4-2: Pinout Diagram - CFLGA - 104pin

Table 4-1: CFLGA Pin Mapping

L	NC		NIOVDD	GPIO0	GPIO4	COREVDD	DB0	DB4	DB6	NC	
K			GPO	GPIO2	GPIO6	GPIO5	DB2	DB8	DB9		
J	NIOVDD	FPFRAME	FPLINE	CVOUT	GPIO3	PWMOUT	DB1	DB5	DB7	DB11	HIOVDD
H	FPDAT1	FPDAT0	FPSHIFT	FPDAT2	DRDY	GPIO1	DB3	DB10	DB13	DB14	DB12
G	FPDAT5	FPDAT4	FPDAT3	FPDAT6	VSS	NC	VSS	WE1#	CLKI	DB15	WAIT#
F	FPDAT10	FPDAT7	FPDAT8	VSS	VSS	NC	NC	VSS	BS#	RD/WR#	RESET#
E	FPDAT11	FPDAT9	FPDAT13	FPDAT16	VSS	NC	VSS	AB1	M/R#	WE0#	RD#
D	NIOVDD	FPDAT12	FPDAT14	CNF7	CNF3	AB13	AB11	AB7	AB3	CS#	AB0
C	NC	FPDAT15	FPDAT17	CNF5	CNF1	TESTEN	AB14	AB9	AB5	AB2	HIOVDD
B	NC		CLKI2	CNF6	CNF0	AB15	AB16	AB8	AB4	NC	
A			NIOVDD	CNF4	CNF2	COREVDD	AB12	AB10	AB6		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**

4.3 Pin Descriptions

Key:

I	=	Input
O	=	Output
IO	=	Bi-Directional (Input/Output)
P	=	Power pin
LIS	=	LVTTL[a] Schmitt input
LI	=	LVTTL input
LB2A	=	LVTTL IO buffer (6mA/-6mA@3.3V)
LB3P	=	Low noise LVTTL IO buffer (12mA/-12mA@3.3V)
LO3	=	Low noise LVTTL Output buffer (12mA/-12mA@3.3V)
LB3M	=	Low noise LVTTL IO buffer with input mask (12mA/-12mA@3.3V)
T1	=	Test mode control input with pull-down resistor (typical value of 50Ω at 3.3V)
Hi-Z	=	High Impedance

[a] LVTTL is Low Voltage TTL (see Section 5, “D.C. Characteristics” on page 30).

4.3.1 Host Interface

Table 4-2: Host Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	IO Voltage	RESET# State	Description
AB0	I	5	LIS	HIOVDD	0	This input pin has multiple functions. • For Generic #1, this pin is not used and should be connected to VSS. • For Generic #2, this pin inputs system address bit 0 (A0). • For SH-3/SH-4, this pin is not used and should be connected to VSS. • For MC68K #1, this pin inputs the lower data strobe (LDS#). • For MC68K #2, this pin inputs system address bit 0 (A0). • For REDCAP2, this pin is not used and should be connected to VSS. • For DragonBall, this pin is not used and should be connected to VSS. See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.
AB[16:1]	I	87-99, 2-4	LI	HIOVDD	0	System address bus bits 16-1.

Table 4-2: Host Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	IO Voltage	RESET# State	Description
DB[15:0]	IO	18-24, 27-35	LB2A	HIOVDD	Hi-Z	Input data from the system data bus. • For Generic #1, these pins are connected to D[15:0]. • For Generic #2, these pins are connected to D[15:0]. • For SH-3/SH-4, these pins are connected to D[15:0]. • For MC68K #1, these pins are connected to D[15:0]. • For MC68K #2, these pins are connected to D[31:16] for a 32-bit device (e.g. MC68030) or D[15:0] for a 16-bit device (e.g. MC68340). • For REDCAP2, these pins are connected to D[15:0]. • For DragonBall, these pins are connected to D[15:0]. See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.
WE0#	I	10	LIS	HIOVDD	1	This input pin has multiple functions. • For Generic #1, this pin inputs the write enable signal for the lower data byte (WE0#). • For Generic #2, this pin inputs the write enable signal (WE#) • For SH-3/SH-4, this pin inputs the write enable signal for data byte 0 (WE0#). • For MC68K #1, this pin must be tied to IO V_{DD} • For MC68K #2, this pin inputs the bus size bit 0 (SIZ0). • For REDCAP2, this pin inputs the byte enable signal for the D[7:0] data byte ($\overline{EB1}$). • For DragonBall, this pin inputs the byte enable signal for the D[7:0] data byte ($\overline{LWE}$). See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.
WE1#	I	11	LIS	HIOVDD	1	This input pin has multiple functions. • For Generic #1, this pin inputs the write enable signal for the upper data byte (WE1#). • For Generic #2, this pin inputs the byte enable signal for the high data byte (BHE#). • For SH-3/SH-4, this pin inputs the write enable signal for data byte 1 (WE1#). • For MC68K #1, this pin inputs the upper data strobe (UDS#). • For MC68K #2, this pin inputs the data strobe (DS#). • For REDCAP2, this pin inputs the byte enable signal for the D[15:8] data byte ($\overline{EB0}$). • For DragonBall, this pin inputs the byte enable signal for the D[15:8] data byte ($\overline{UWE}$). See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.
CS#	I	6	LI	HIOVDD	1	Chip select input. See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.
M/R#	I	7	LIS	HIOVDD	0	This input pin is used to select between the display buffer and register address spaces of the S1D13706. M/R# is set high to access the display buffer and low to access the registers. See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.

Table 4-2: Host Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	IO Voltage	RESET# State	Description
BS#	I	8	LIS	HIOVDD	1	This input pin has multiple functions. • For Generic #1, this pin must be tied to IO V_{DD}. • For Generic #2, this pin must be tied to IO V_{DD}. • For SH-3/SH-4, this pin inputs the bus start signal (BS#). • For MC68K #1, this pin inputs the address strobe (AS#). • For MC68K #2, this pin inputs the address strobe (AS#). • For REDCAP2, this pin must be tied to IO V_{DD}. • For DragonBall, this pin must be tied to IO V_{DD}. See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.
RD/WR#	I	12	LIS	HIOVDD	1	This input pin has multiple functions. • For Generic #1, this pin inputs the read command for the upper data byte (RD1#). • For Generic #2, this pin must be tied to IO V_{DD}. • For SH-3/SH-4, this pin inputs the RD/WR# signal. The S1D13706 needs this signal for early decode of the bus cycle. • For MC68K #1, this pin inputs the R/W# signal. • For MC68K #2, this pin inputs the R/W# signal. • For REDCAP2, this pin inputs the R/$\overline{W}$ signal. • For DragonBall, this pin must be tied to IO V_{DD}. See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.
RD#	I	9	LIS	HIOVDD	1	This input pin has multiple functions. • For Generic #1, this pin inputs the read command for the lower data byte (RD0#). • For Generic #2, this pin inputs the read command (RD#). • For SH-3/SH-4, this pin inputs the read signal (RD#). • For MC68K #1, this pin must be tied to IO V_{DD}. • For MC68K #2, this pin inputs the bus size bit 1 (SIZ1). • For REDCAP2, this pin inputs the output enable ($\overline{OE}$). • For DragonBall, this pin inputs the output enable ($\overline{OE}$). See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.

Table 4-2: Host Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	IO Voltage	RESET# State	Description
WAIT#	O	17	LB2A	HIOVDD	Hi-Z	During a data transfer, this output pin is driven active to force the system to insert wait states. It is driven inactive to indicate the completion of a data transfer. WAIT# is released to the high impedance state after the data transfer is complete. Its active polarity is configurable. See Table 4-7: “Summary of Power-On/Reset Options,” on page 27. • For Generic #1, this pin outputs the wait signal (WAIT#). • For Generic #2, this pin outputs the wait signal (WAIT#). • For SH-3 mode, this pin outputs the wait request signal (WAIT#). • For SH-4 mode, this pin outputs the device ready signal (RDY#). • For MC68K #1, this pin outputs the data transfer acknowledge signal (DTACK#). • For MC68K #2, this pin outputs the data transfer and size acknowledge bit 1 (DSACK1#). • For REDCAP2, this pin is unused (Hi-Z). • For DragonBall, this pin outputs the data transfer acknowledge signal (DTACK). See Table 4-8: “Host Bus Interface Pin Mapping,” on page 28 for summary.
RESET#	I	13	LIS	HIOVDD	0	Active low input to set all internal registers to the default state and to force all signals to their inactive states.

4.3.2 LCD Interface

Table 4-3: LCD Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	IO Voltage	RESET# State	Description
FPDAT[17:0]	O	74-64, 61-55	LB3P	NIOVDD	0	Panel Data bits 17-0.
FPFRAME	O	52	LB3P	NIOVDD	0	This output pin has multiple functions. • Frame Pulse • SPS for Sharp HR-TFT • DY for Epson D-TFD See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
FPLINE	O	53	LB3P	NIOVDD	0	This output pin has multiple functions. • Line Pulse • LP for Sharp HR-TFT • LP for Epson D-TFD See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
FPSHIFT	O	54	LB3P	NIOVDD	0	This output pin has multiple functions. • Shift Clock • CLK for Sharp HR-TFT • XSCL for Epson D-TFD See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
DRDY	O	48	LO3	NIOVDD	0	This output pin has multiple functions. • Display enable (DRDY) for TFT panels • 2nd shift clock (FPSHIFT2) for passive LCD with Format 1 interface • GCP for Epson D-TFD • LCD backplane bias signal (MOD) for all other LCD panels See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
GPIO0	IO	45	LB3M	NIOVDD	0	This pin has multiple functions. • PS for Sharp HR-TFT • XINH for Epson D-TFD • General purpose IO pin 0 (GPIO0) • Hardware Video Invert See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
GPIO1	IO	44	LB3M	NIOVDD	0	This pin has multiple functions. • CLS for Sharp HR-TFT • YSCL for Epson D-TFD • General purpose IO pin 1 (GPIO1) See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.

Table 4-3: LCD Interface Pin Descriptions

Pin Name	Type	Pin #	Cell	IO Voltage	RESET# State	Description
GPIO2	IO	43	LB3M	NIOVDD	0	This pin has multiple functions. • REV for Sharp HR-TFT • FR for Epson D-TFD • General purpose IO pin 2 (GPIO2) See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
GPIO3	IO	42	LB3M	NIOVDD	0	This pin has multiple functions. • SPL for Sharp HR-TFT • FRS for Epson D-TFD • General purpose IO pin 3 (GPIO3) See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
GPIO4	IO	41	LB3M	NIOVDD	0	This pin has multiple functions. • RES for Epson D-TFD • General purpose IO pin 4 (GPIO4) See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
GPIO5	IO	40	LB3M	NIOVDD	0	This pin has multiple functions. • DD_P1 for Epson D-TFD • General purpose IO pin 5 (GPIO5) See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
GPIO6	IO	39	LB3M	NIOVDD	0	This pin has multiple functions. • YSCLD for Epson D-TFD • General purpose IO pin 6 (GPIO6) See Table 4-9: “LCD Interface Pin Mapping,” on page 29 for summary.
PWMOUT	O	38	LB3P	NIOVDD	0	This output pin has multiple functions. • PWM Clock output • General purpose output
CVOUT	O	46	LB3P	NIOVDD	0	This output pin has multiple functions. • CV Pulse Output • General purpose output

4.3.3 Clock Input

Table 4-4: Clock Input Pin Descriptions

Pin Name	Type	Pin #	Cell	IO Voltage	RESET# State	Description
CLKI	I	15	LI	NIOVDD	—	Typically used as input clock source for bus clock and memory clock
CLKI2	I	77	LI	NIOVDD	—	Typically used as input clock source for pixel clock

4.3.4 Miscellaneous

Table 4-5: Miscellaneous Pin Descriptions

Pin Name	Type	Pin #	Cell	IO Voltage	RESET# State	Description
CNF[7:0]	I	78-85	LI	NIOVDD	—	These inputs are used to configure the S1D13706 - see Table 4-7: "Summary of Power-On/Reset Options," on page 27. **Note: These pins are used for configuration of the S1D13706 and must be connected directly to IO V_{DD} or V_{SS}.**
GPO	O	47	LO3	NIOVDD	0	General Purpose Output (possibly used for controlling the LCD power). It may also be used for the MOD control signal of the Sharp HR-TFT panel.
TESTEN	I	86	T1	NIOVDD	0	Test Enable input used for production test only (has type 1 pull-down resistor with a typical value of 50Ω at 3.3V).

4.3.5 Power And Ground

Table 4-6: Power And Ground Pin Descriptions

Pin Name	Type	Pin #	Cell	IO Voltage	RESET# State	Description
HIOVDD	P	16, 26	P	—	—	IO V_{DD} pins associated with the host interface pins as described in Section 4.3.1, "Host Interface" on page 20.
NIOVDD	P	37, 49, 63, 76	P	—	—	IO V_{DD} pins associated with the non-host interface pins as described in Section 4.3.2, "LCD Interface" on page 24, Section 4.3.3, "Clock Input" on page 26, and Section 4.3.4, "Miscellaneous" on page 26.
COREVDD	P	1, 51	P	—	—	2 Core $V_{DD.}$ pins.
VSS	P	14, 25, 36, 50, 62, 75, 100	P	—	—	7 V_{SS} pins.

4.4 Summary of Configuration Options

These pins are used for configuration of the S1D13706 and must be connected directly to IOV_{DD} or V_{SS}. The state of CNF[6:0] is latched on the rising edge of RESET# or after the software reset function is activated (REG[A2h] bit 0). Changing state at any other time has no effect.

Table 4-7: Summary of Power-On/Reset Options

S1D13706 Configuration Input	Power-On/Reset State	
	1 (connected to IOV_{DD})	**0 (Connected to V_{SS})**
CNF[2:0]	Select host bus interface as follows: **CNF2 CNF1 CNF0 — Host Bus** 0 0 0 — SH-4/SH-3 interface 0 0 1 — MC68K #1 0 1 0 — MC68K #2 0 1 1 — Generic #1 1 0 0 — Generic #2 1 0 1 — REDCAP2 1 1 0 — DragonBall (MC68EZ328/MC68VZ328) 1 1 1 — Reserved Note: The host bus interface is 16-bit only.	
CNF3	Configure GPIO pins as inputs at power-on	Configure GPIO pins as outputs at power-on (for use by HR-TFT/D-TFD when selected)
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	WAIT# is active high	WAIT# is active low
CNF[7:6]	CLKI to BCLK divide select: **CNF7 CNF6 — CLKI to BCLK Divide Ratio** 0 0 — 1 : 1 0 1 — 2 : 1 1 0 — 3 : 1 1 1 — 4 : 1	

4.5 Host Bus Interface Pin Mapping

Table 4-8: Host Bus Interface Pin Mapping

S1D13706 Pin Name	Generic #1	Generic #2	Hitachi SH-3 /SH-4	Motorola MC68K #1	Motorola MC68K #2	Motorola REDCAP2	Motorola MC68EZ328/ MC68VZ328 DragonBall
AB[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]
AB0	A0[1]	A0	A0[1]	LDS#	A0	A0[1]	A0[1]
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0][2]	D[15:0]	D[15:0]
CS#	External Decode		CSn#	External Decode		$\overline{CSn}$	$\overline{CSX}$
M/R#	External Decode						
CLKI	BUSCLK	BUSCLK	CKIO	CLK	CLK	CLK	CLKO
BS#	Connected to V_{DD}		BS#	AS#	AS#	Connected to V_{DD}	
RD/WR#	RD1#	Connected to V_{DD}	RD/WR#	R/W#	R/W#	R/$\overline{W}$	Connected to V_{DD}
RD#	RD0#	RD#	RD#	Connected to V_{DD}	SIZ1	$\overline{OE}$	$\overline{OE}$
WE0#	WE0#	WE#	WE0#	Connected to V_{DD}	SIZ0	$\overline{EB1}$	$\overline{LWE}$
WE1#	WE1#	BHE#	WE1#	UDS#	DS#	$\overline{EB0}$	$\overline{UWE}$
WAIT#	WAIT#	WAIT#	WAIT#/ RDY#	DTACK#	DSACK1#	N/A	$\overline{DTACK}$
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	$\overline{RESET\_OUT}$	$\overline{RESET}$

Note

[1] A0 for these busses is not used internally by the S1D13706 and should be connected to V_{SS}.

[2] If the target MC68K bus is 32-bit, then these signals should be connected to D[31:16].

4.6 LCD Interface Pin Mapping

Table 4-9: LCD Interface Pin Mapping

Pin Name	Monochrome Passive Panel		Color Passive Panel				Color TFT Panel				
	Single		Single				Others			Sharp HR-TFT[1]	Epson D-TFD[1]
	4-bit	8-bit	4-bit	Format 1 8-bit	Format 2 8-bit	16-Bit	9-bit	12-bit	18-bit	18-bit	18-bit
FPFRAME	FPFRAME									SPS	DY
FPLINE	FPLINE									LP	LP
FPSHIFT	FPSHIFT									DCLK	XSCL
DRDY	MOD			FPSHIFT2	MOD		DRDY			no connect	GCP
FPDAT0	driven 0	D0	driven 0	D0 (B5)[2]	D0 (G3)[2]	D0 (R6)[2]	R2	R3	R5	R5	R5
FPDAT1	driven 0	D1	driven 0	D1 (R5)[2]	D1 (R3)[2]	D1 (G5)[2]	R1	R2	R4	R4	R4
FPDAT2	driven 0	D2	driven 0	D2 (G4)[2]	D2 (B2)[2]	D2 (B4)[2]	R0	R1	R3	R3	R3
FPDAT3	driven 0	D3	driven 0	D3 (B3)[2]	D3 (G2)[2]	D3 (R4)[2]	G2	G3	G5	G5	G5
FPDAT4	D0	D4	D0 (R2)[2]	D4 (R3)[2]	D4 (R2)[2]	D8 (B5)[2]	G1	G2	G4	G4	G4
FPDAT5	D1	D5	D1 (B1)[2]	D5 (G2)[2]	D5 (B1)[2]	D9 (R5)[2]	G0	G1	G3	G3	G3
FPDAT6	D2	D6	D2 (G1)[2]	D6 (B1)[2]	D6 (G1)[2]	D10 (G4)[2]	B2	B3	B5	B5	B5
FPDAT7	D3	D7	D3 (R1)[2]	D7 (R1)[2]	D7 (R1)[2]	D11 (B3)[2]	B1	B2	B4	B4	B4
FPDAT8	driven 0	driven 0	driven 0	driven 0	driven 0	D4 (G3)[2]	B0	B1	B3	B3	B3
FPDAT9	driven 0	driven 0	driven 0	driven 0	driven 0	D5 (B2)[2]	driven 0	R0	R2	R2	R2
FPDAT10	driven 0	driven 0	driven 0	driven 0	driven 0	D6 (R2)[2]	driven 0	driven 0	R1	R1	R1
FPDAT11	driven 0	driven 0	driven 0	driven 0	driven 0	D7 (G1)[2]	driven 0	driven 0	R0	R0	R0
FPDAT12	driven 0	driven 0	driven 0	driven 0	driven 0	D12 (R3)[2]	driven 0	G0	G2	G2	G2
FPDAT13	driven 0	driven 0	driven 0	driven 0	driven 0	D13 (G2)[2]	driven 0	driven 0	G1	G1	G1
FPDAT14	driven 0	driven 0	driven 0	driven 0	driven 0	D14 (B1)[2]	driven 0	driven 0	G0	G0	G0
FPDAT15	driven 0	driven 0	driven 0	driven 0	driven 0	D15 (R1)[2]	driven 0	B0	B2	B2	B2
FPDAT16	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B1	B1	B1
FPDAT17	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B0	B0	B0
GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	PS	XINH
GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	CLS	YSCL
GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	REV	FR
GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	SPL	FRS
GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4 (output only)	RES
GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5 (output only)	DD_P1
GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6 (output only)	YSCLD
GPO	GPO (General Purpose Output)									MOD[3]	GPO
CVOUT	CVOUT										
PWMOUT	PWMOUT										

Note

[1] GPIO pins must be configured as outputs (CNF3 = 0 at RESET#) when the HR-TFT or D-TFD interface is selected.

[2] These pin mappings use signal names commonly used for each panel type, however signal names may differ between panel manufacturers. The values shown in brackets represent the color components as mapped to the corresponding FPDATxx signals at the first valid edge of FPSHIFT. For further FPDATxx to LCD interface mapping, see Section 6.4, “Display Interface” on page 55.

[3] When the HR-TFT interface is selected (REG[10h] bits 1-0 = 10), this GPO can be used to control the HR-TFT MOD signal. Note this is not the same signal as the S1D13706 DRDY(MOD) signal used for passive panels.

5 D.C. Characteristics

Table 5-1: Absolute Maximum Ratings

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

Table 5-2: Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
Core V_{DD}	Supply Voltage	V_{SS} = 0 V	1.8	2.0	2.2	V
			3.0	3.3	3.6	V
HIO V_{DD}	Supply Voltage	V_{SS} = 0 V	1.8	2.0	2.2	V
			3.0	3.3	3.6	V
NIO V_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	3.6	V
V_{IN}	Input Voltage		V_{SS}		IO V_{DD}	V
T_{OPR}	Operating Temperature		-40	25	85	° C

Note
The S1D13706 requires that Core VDD ≤ HIO VDD and Core VDD ≤ NIO VDD.

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

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{DDS}	Quiescent Current	Quiescent Conditions			170	µA
I_{IZ}	Input Leakage Current		-1		1	µA
I_{OZ}	Output Leakage Current		-1		1	µA
V_{OH}	High Level Output Voltage	VDD = min I_{OH} = -6mA (Type 2) -12mA (Type 3)	V_{DD} - 0.4			V
V_{OL}	Low Level Output Voltage	VDD = min I_{OL} = 6mA (Type 2) 12mA (Type 3)			0.4	V
V_{IH}	High Level Input Voltage	LVTTL Level, V_{DD} = max	2.0			V
V_{IL}	Low Level Input Voltage	LVTTL Level, V_{DD} = min			0.8	V
V_{T+}	High Level Input Voltage	LVTTL Schmitt	1.1		2.4	V
V_{T-}	Low Level Input Voltage	LVTTL Schmitt	0.6		1.8	V
V_{H1}	Hysteresis Voltage	LVTTL Schmitt	0.1			V
R_{PD}	Pull Down Resistance	$V_I = V_{DD}$	20	50	120	kΩ
C_I	Input Pin Capacitance				10	pF
C_O	Output Pin Capacitance				10	pF
C_{IO}	Bi-Directional Pin Capacitance				10	pF

6 A.C. Characteristics

Conditions: HIO V_{DD} = 2.0V ± 10% and HIO V_{DD} = 3.3V ± 10%
NIO V_{DD} = 3.3V ± 10%
T_A = -40° C to 85° C
T_{rise} and T_{fall} for all inputs must be ≤ 5 nsec (10% ~ 90%)
C_L = 50pF (Bus/MPU Interface)
C_L = 0pF (LCD Panel Interface)

6.1 Clock Timing

6.1.1 Input Clocks

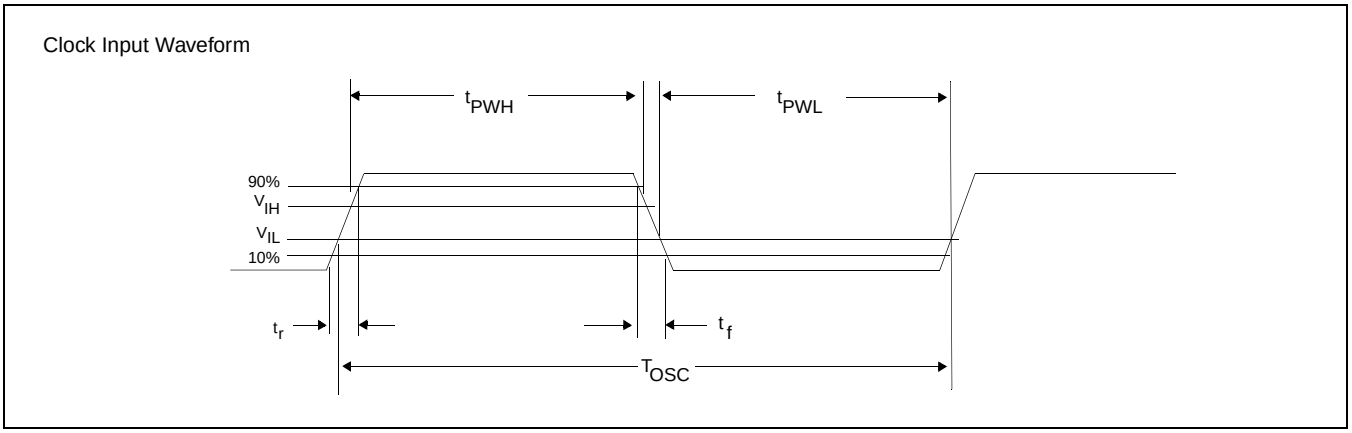


Figure 6-1: Clock Input Requirements

Table 6-1: Clock Input Requirements for CLKI when CLKI to BCLK divide > 1

Symbol	Parameter	2.0V		3.3V		Units
		Min	Max	Min	Max	
f_{OSC}	Input Clock Frequency (CLKI)		40		100	MHz
T_{OSC}	Input Clock period (CLKI)	$1/f_{OSC}$		$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High (CLKI)	4.5		4.5		ns
t_{PWL}	Input Clock Pulse Width Low (CLKI)	4.5		4.5		ns
t_f	Input Clock Fall Time (10% - 90%)		5		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5		5	ns

Note
Maximum internal requirements for clocks derived from CLKI must be considered when determining the frequency of CLKI. See Section 6.1.2, “Internal Clocks” on page 33 for internal clock requirements.

Table 6-2: Clock Input Requirements for CLKI when CLKI to BCLK divide = 1

Symbol	Parameter	2.0V		3.3V		Units
		Min	Max	Min	Max	
f_{OSC}	Input Clock Frequency (CLKI)		20		66	MHz
T_{OSC}	Input Clock period (CLKI)	$1/f_{OSC}$		$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High (CLKI)	3		3		ns
t_{PWL}	Input Clock Pulse Width Low (CLKI)	3		3		ns
t_f	Input Clock Fall Time (10% - 90%)		5		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5		5	ns

Note

Maximum internal requirements for clocks derived from CLKI must be considered when determining the frequency of CLKI. See Section 6.1.2, “Internal Clocks” on page 33 for internal clock requirements.

Table 6-3: Clock Input Requirements for CLKI2

Symbol	Parameter	2.0V		3.3V		Units
		Min	Max	Min	Max	
f_{OSC}	Input Clock Frequency (CLKI2)		20		66	MHz
T_{OSC}	Input Clock period (CLKI2)	$1/f_{OSC}$		$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High (CLKI2)	3		3		ns
t_{PWL}	Input Clock Pulse Width Low (CLKI2)	3		3		ns
t_f	Input Clock Fall Time (10% - 90%)		5		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5		5	ns

Note

Maximum internal requirements for clocks derived from CLKI2 must be considered when determining the frequency of CLKI2. See Section 6.1.2, “Internal Clocks” on page 33 for internal clock requirements.

6.1.2 Internal Clocks

Table 6-4: Internal Clock Requirements

Symbol	Parameter	2.0V		3.3V		Units
		Min	Max	Min	Max	
f_{BCLK}	Bus Clock frequency		20		66	MHz
f_{MCLK}	Memory Clock frequency		20		50	MHz
f_{PCLK}	Pixel Clock frequency		20		50	MHz
f_{PWMCLK}	PWM Clock frequency		20		66	MHz

Note
For further information on internal clocks, refer to Section 7, “Clocks” on page 88.

6.2 CPU Interface Timing

The following section includes CPU interface AC Timing for both 2.0V and 3.3V. The 2.0V timings are based on HIO V_{DD} = Core V_{DD} = 2.0V. The 3.3V timings are based on HIO V_{DD} = Core V_{DD} = 3.3V.

6.2.1 Generic #1 Interface Timing

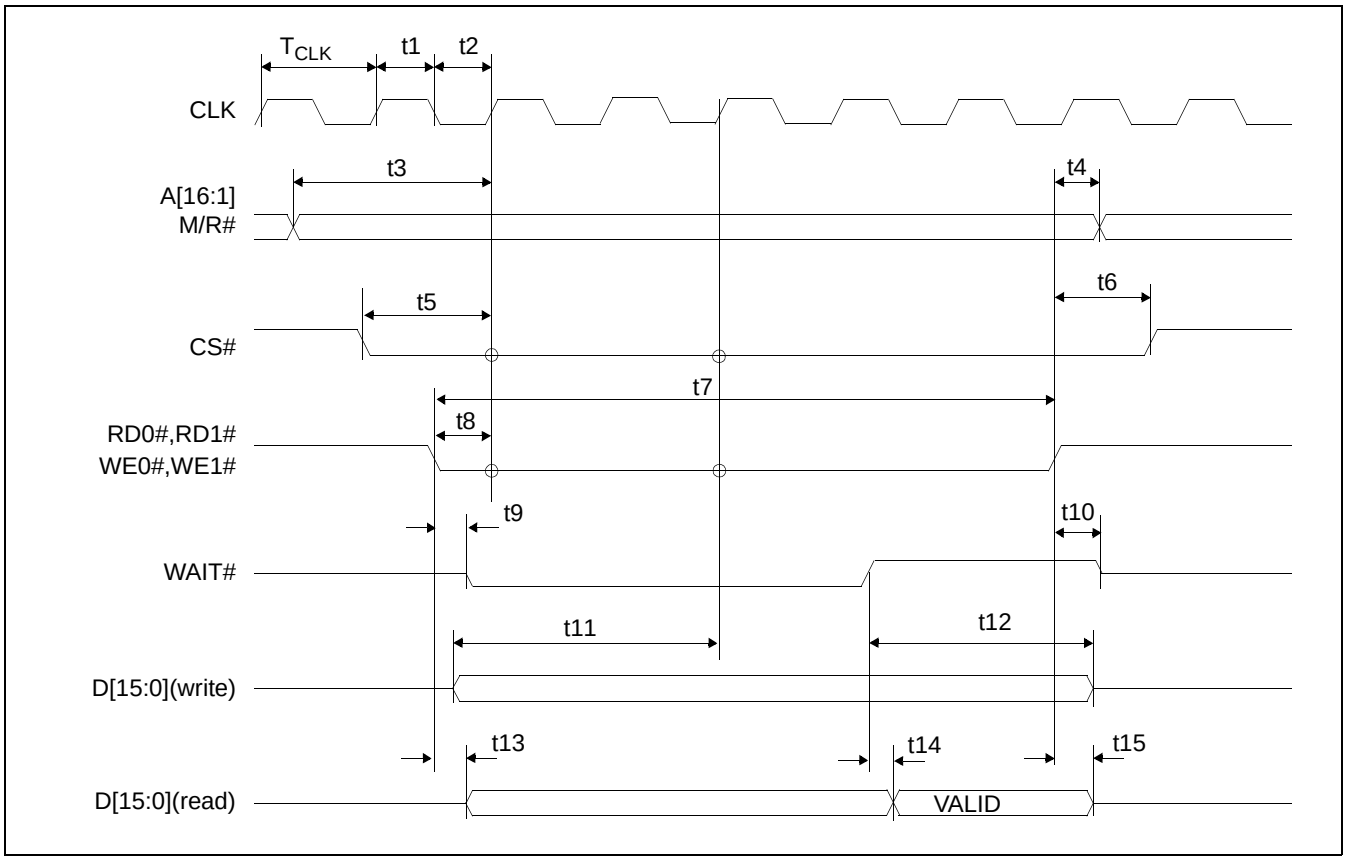


Figure 6-2: Generic #1 Interface Timing

Table 6-5: Generic #1 Interface Timing

Symbol	Parameter	2.0V		3.3V		Unit
		Min	Max	Min	Max	
f_{CLK}	Bus Clock frequency		20		50	MHz
T_{CLK}	Bus Clock period	$1/f_{CLK}$		$1/f_{CLK}$		ns
t1	Clock pulse width high	22.5		9		ns
t2	Clock pulse width low	22.5		9		ns
t3	A[16:1], M/R# setup to first CLK rising edge where CS# = 0 and either RD0#, RD1# = 0 or WE0#, WE1# = 0	1		1		ns
t4	A[16:1], M/R# hold from either RD0#, RD1# or WE0#, WE1# rising edge	0		0		ns
t5	CS# setup to CLK rising edge	0		1		ns
t6	CS# hold from either RD0#, RD1# or WE0#, WE1# rising edge	0		0		ns
t7a	RD0#, RD1#, WE0#, WE1# asserted for MCLK = BCLK		8.5		8.5	T_{CLK}
t7b	RD0#, RD1#, WE0#, WE1# asserted for MCLK = BCLK ÷ 2		11.5		11.5	T_{CLK}
t7c	RD0#, RD1#, WE0#, WE1# asserted for MCLK = BCLK ÷ 3		13.5		13.5	T_{CLK}
t7d	RD0#, RD1#, WE0#, WE1# asserted for MCLK = BCLK ÷ 4		17.5		17.5	T_{CLK}
t8	RD0#, RD1#, WE0#, WE1# setup to CLK rising edge	2		1		ns
t9	Falling edge of either RD0#, RD1# or WE0#, WE1# to WAIT# driven low	5	31	3	15	ns
t10	Rising edge of either RD0#, RD1# or WE0#, WE1# to WAIT# high impedance	5	34	3	13	ns
t11	D[15:0] setup to third CLK rising edge where CS# = 0 and WE0#, WE1# = 0 (write cycle) (see note 1)	1		0		ns
t12	D[15:0] hold from WAIT# rising edge (write cycle)	1		0		ns
t13	RD0#, RD1# falling edge to D[15:0] driven (read cycle)	4	27	3	14	ns
t14	WAIT# rising edge to D[15:0] valid (read cycle)		0		2	ns
t15	RD0#, RD1# rising edge to D[15:0] high impedance (read cycle)	3	29	3	11	ns

1. t11 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.2 Generic #2 Interface Timing (e.g. ISA)

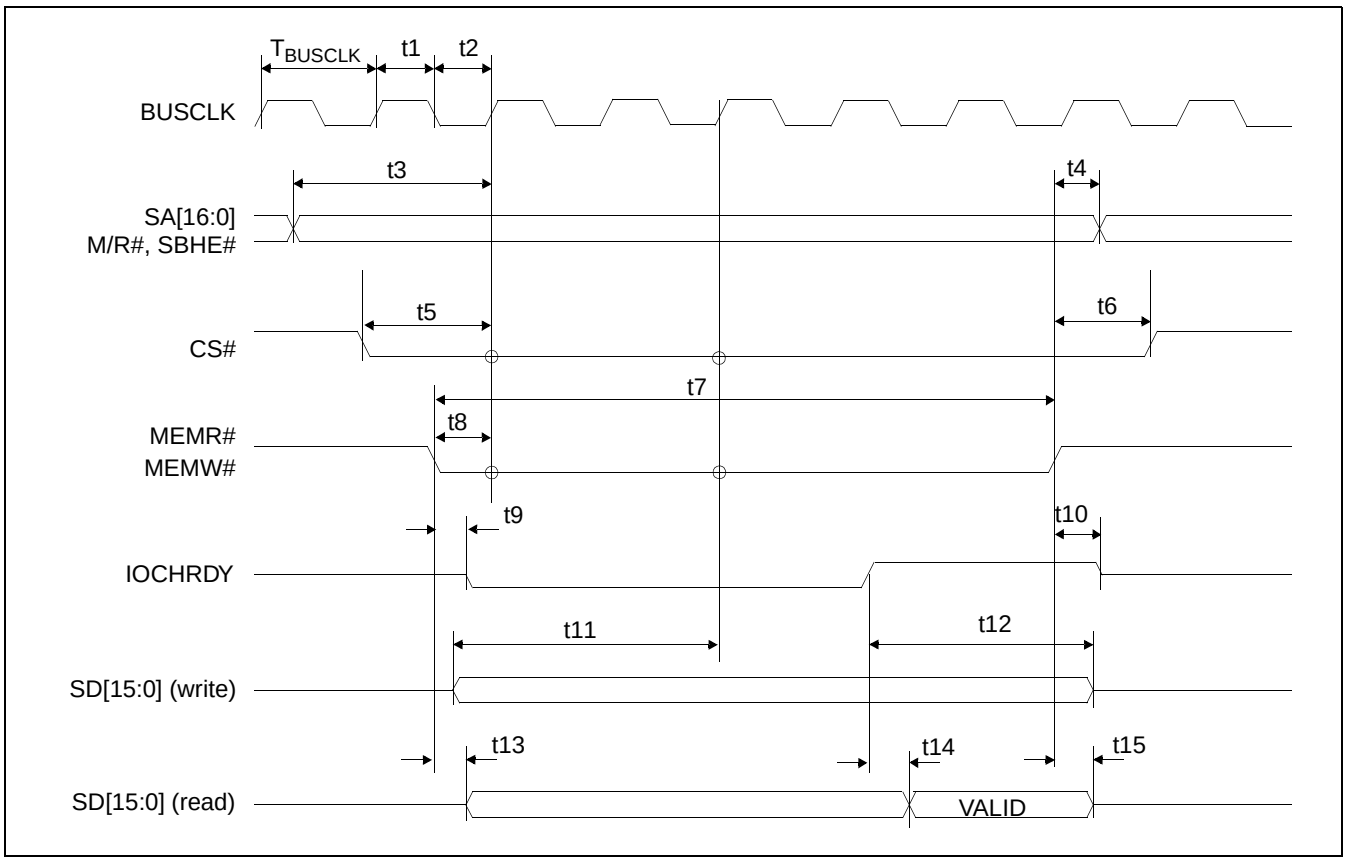


Figure 6-3: Generic #2 Interface Timing

Table 6-6: Generic #2 Interface Timing

Symbol	Parameter	2.0V		3.3V		Unit
		Min	Max	Min	Max	
f_{BUSCLK}	Bus Clock frequency		20		50	MHz
T_{BUSCLK}	Bus Clock period	$1/f_{BUSCLK}$		$1/f_{BUSCLK}$		ns
t1	Clock pulse width high	22.5		9		ns
t2	Clock pulse width low	22.5		9		ns
t3	SA[16:0], M/R#, SBHE# setup to first BUSCLK rising edge where CS# = 0 and either MEMR# = 0 or MEMW# = 0	1		1		ns
t4	SA[16:0], M/R#, SBHE# hold from either MEMR# or MEMW# rising edge	0		0		ns
t5	CS# setup to BUSCLK rising edge	0		1		ns
t6	CS# hold from either MEMR# or MEMW# rising edge	0		0		ns
t7a	MEMR#/MEMW# asserted for MCLK = BCLK		8.5		8	T_{BUSCLK}
t7b	MEMR#/MEMW# asserted for MCLK = BCLK ÷ 2		11.5		11	T_{BUSCLK}
t7c	MEMR#/MEMW# asserted for MCLK = BCLK ÷ 3		13.5		13	T_{BUSCLK}
t7d	MEMR#/MEMW# asserted for MCLK = BCLK ÷ 4		17.5		17	T_{BUSCLK}
t8	MEMR# or MEMW# setup to BUSCLK rising edge	2		1		ns
t9	Falling edge of either MEMR# or MEMW# to IOCHRDY driven low	5		3	15	ns
t10	Rising edge of either MEMR# or MEMW# to IOCHRDY high impedance	5		3	13	ns
t11	SD[15:0] setup to third BUSCLK rising edge where CS# = 0 and MEMW# = 0 (write cycle) (see note 1)	1		0		ns
t12	SD[15:0] hold from IOCHRDY rising edge (write cycle)	1		0		ns
t13	MEMR# falling edge to SD[15:0] driven (read cycle)	4	26	3	13	ns
t14	IOCHRDY rising edge to SD[15:0] valid (read cycle)		0		2	ns
t15	Rising edge of MEMR# to SD[15:0] high impedance (read cycle)	5	33	3	12	ns

1. t11 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.3 Hitachi SH-4 Interface Timing

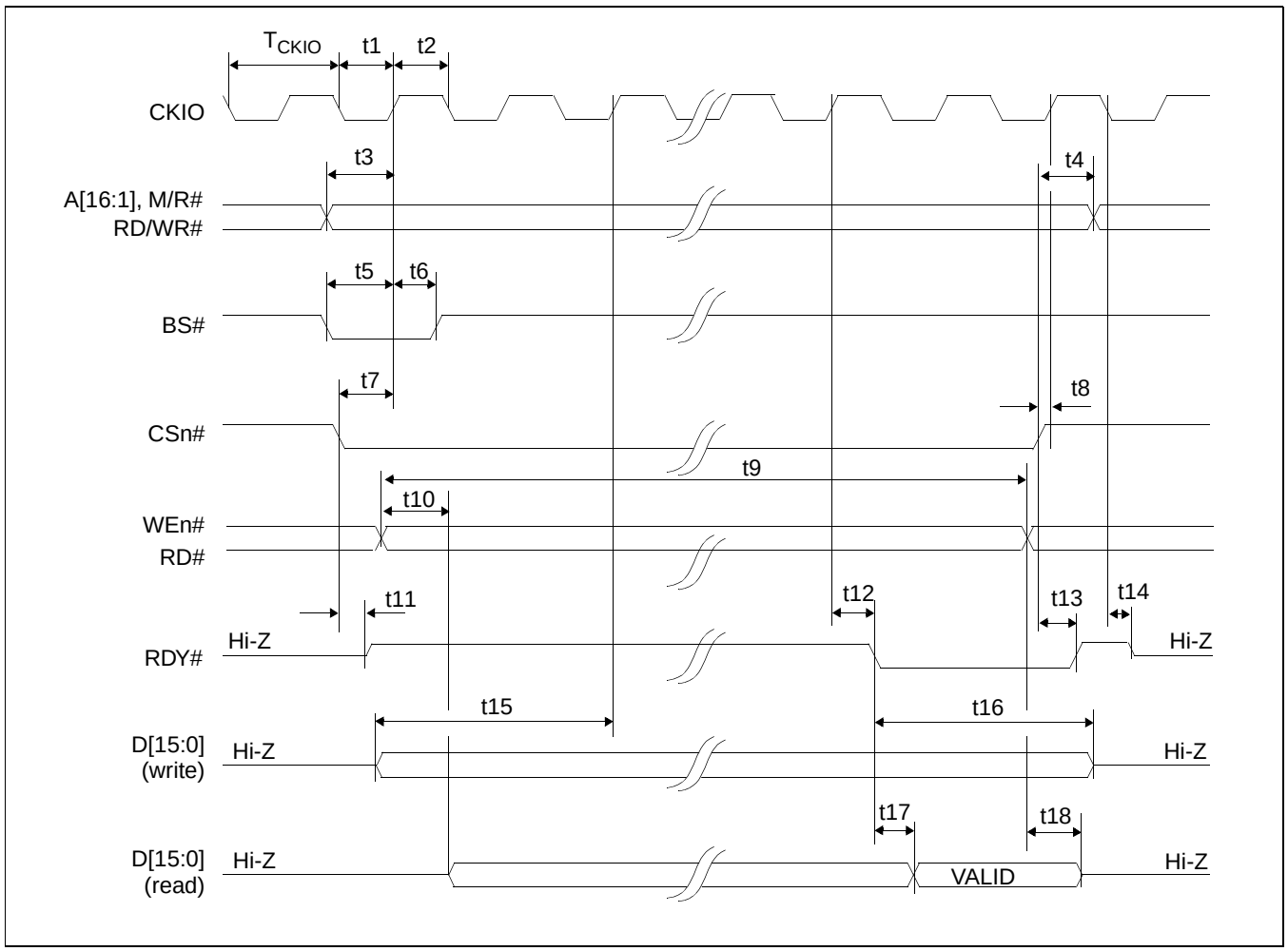


Figure 6-4: Hitachi SH-4 Interface Timing

Table 6-7: Hitachi SH-4 Interface Timing

Symbol	Parameter	2.0V		3.3V		Unit
		Min	Max	Min	Max	
f_{CKIO}	Clock frequency		20		66	MHz
T_{CKIO}	Clock period	$1/f_{CKIO}$		$1/f_{CKIO}$		ns
t1	Clock pulse width low	22.5		6.8		ns
t2	Clock pulse width high	22.5		6.8		ns
t3	A[16:1], M/R#, RD/WR# setup to CKIO	0		1		ns
t4	A[16:1], M/R#, RD/WR# hold from CSn#	0		0		ns
t5	BS# setup	3		1		ns
t6	BS# hold	7		2		ns
t7	CSn# setup	0		1		ns
t8	CSn# high setup to CKIO	0		2		ns
t9a	RD# or WEn# asserted for MCLK = BCLK (max. MCLK = 50MHz)		8.5		8.5	T_{CKIO}
t9b	RD# or WEn# asserted for MCLK = BCLK ÷ 2		11.5		11.5	T_{CKIO}
t9c	RD# or WEn# asserted for MCLK = BCLK ÷ 3		13.5		13.5	T_{CKIO}
t9d	RD# or WEn# asserted for MCLK = BCLK ÷ 4		18.5		18.5	T_{CKIO}
t10	Falling edge RD# to D[15:0] driven (read cycle)	5	24	3	12	ns
t11	Falling edge CSn# to RDY# driven high	3	19	3	12	ns
t12	CKIO to RDY# low	5	42	4	18	ns
t13	CSn# high to RDY# high	5	35	4	14	ns
t14	Falling edge CKIO to RDY# high impedance	5	38	4	14	ns
t15	D[15:0] setup to 2nd CKIO after BS# (write cycle) (see note 1)	1		0		ns
t16	D[15:0] hold (write cycle)	0		0		ns
t17	RDY# falling edge to D[15:0] valid (read cycle)		0		2	ns
t18	Rising edge RD# to D[15:0] high impedance (read cycle)	5	31	3	12	ns

1. t15 is the delay from when data is placed on the bus until the data is latched into the write buffer.

Note
Minimum one software WAIT state is required.

6.2.4 Hitachi SH-3 Interface Timing

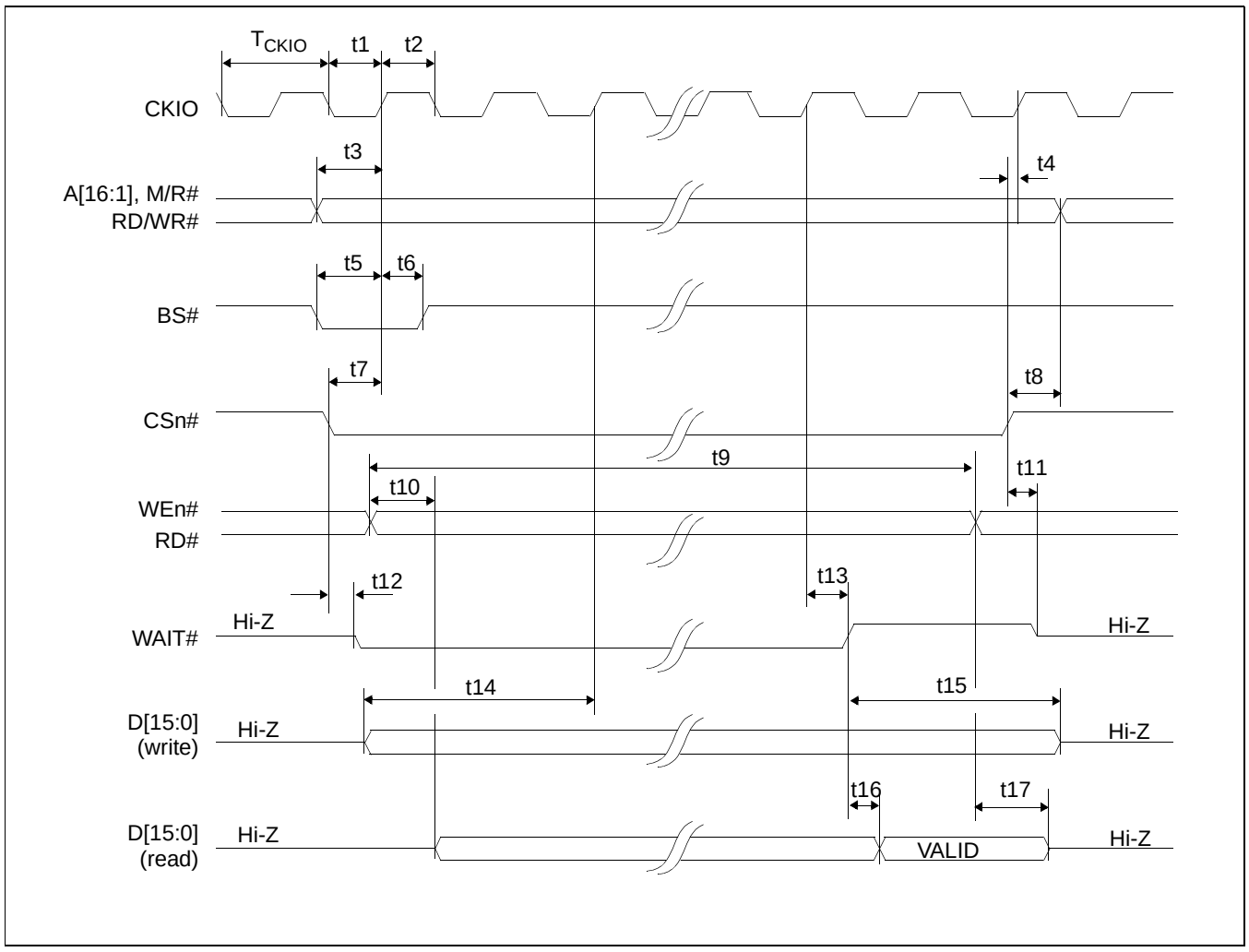


Figure 6-5: Hitachi SH-3 Interface Timing

Table 6-8: Hitachi SH-3 Interface Timing

Symbol	Parameter	2.0V		3.3V		Unit
		Min	Max	Min	Max	
f_{CKIO}	Bus Clock frequency		20		66	MHz
T_{CKIO}	Bus Clock period	$1/f_{CKIO}$		$1/f_{CKIO}$		ns
t1	Bus Clock pulse width low	22.5		6.8		ns
t2	Bus Clock pulse width high	22.5		6.8		ns
t3	A[16:1], M/R#, RD/WR# setup to CKIO	0		1		ns
t4	CSn# high setup to CKIO	0		1		ns
t5	BS# setup	3		1		ns
t6	BS# hold	7		2		ns
t7	CSn# setup	0		1		ns
t8	A[16:1], M/R#, RD/WR# hold from CS#	0		0		ns
t9a	RD# or WEn# asserted for MCLK = BCLK (max. MCLK = 50MHz)		8.5		8.5	T_{CKIO}
t9b	RD# or WEn# asserted for MCLK = BCLK ÷ 2		11.5		11.5	T_{CKIO}
t9c	RD# or WEn# asserted for MCLK = BCLK ÷ 3		13.5		13.5	T_{CKIO}
t9d	RD# or WEn# asserted for MCLK = BCLK ÷ 4		18.5		18.5	T_{CKIO}
t10	Falling edge RD# to D[15:0] driven (read cycle)	5	24	3	12	ns
t11	Rising edge CSn# to WAIT# high impedance	4	24	2	10	ns
t12	Falling edge CSn# to WAIT# driven low	3	24	2	12	ns
t13	CKIO to WAIT# delay	6	45	4	18	ns
t14	D[15:0] setup to 2nd CKIO after BS# (write cycle) (see note 1)	1		0		ns
t15	D[15:0] hold (write cycle)	0		0		ns
t16	WAIT# rising edge to D[15:0] valid (read cycle)		0		2	ns
t17	Rising edge RD# to D[15:0] high impedance (read cycle)	5	31	3	12	ns

1. t14 is the delay from when data is placed on the bus until the data is latched into the write buffer.

Note
Minimum one software WAIT state is required.

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

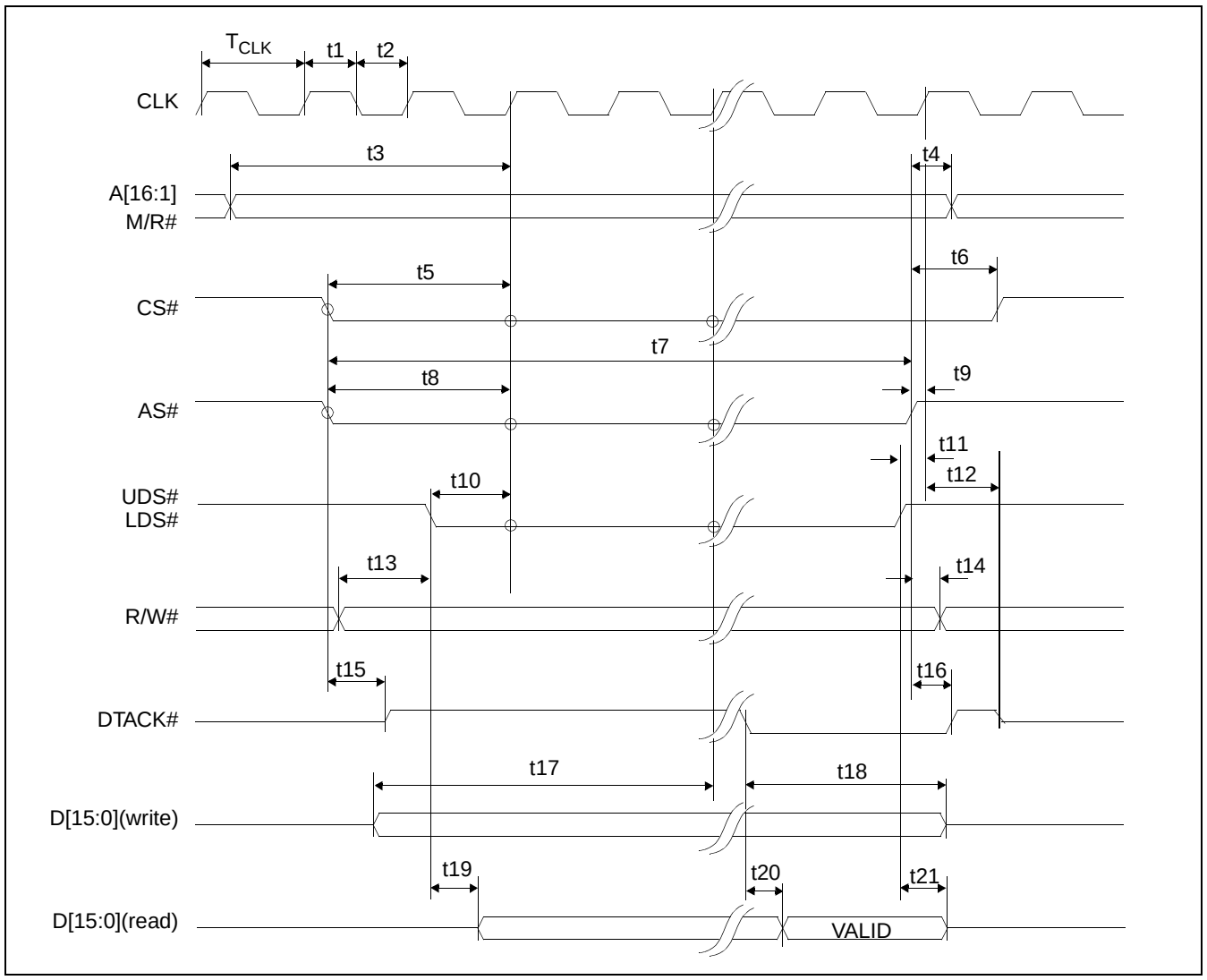


Figure 6-6: Motorola MC68K #1 Interface Timing

Table 6-9: Motorola MC68K #1 Interface Timing

Symbol	Parameter	2.0V		3.3V		Unit
		Min	Max	Min	Max	
f_{CLK}	Bus Clock Frequency		20		50	MHz
T_{CLK}	Bus Clock period	$1/f_{CLK}$		$1/f_{CLK}$		ns
t1	Clock pulse width high	22.5		9		ns
t2	Clock pulse width low	22.5		9		ns
t3	A[16:1], M/R# setup to first CLK rising edge where CS# = 0, AS# = 0, UDS# = 0, and LDS# = 0	1		1		ns
t4	A[16:1], M/R# hold from AS# rising edge	0		0		ns
t5	CS# setup to CLK rising edge while CS#, AS#, UDS#/LDS# = 0	0		1		ns
t6	CS# hold from AS# rising edge	0		0		ns
t7a	AS# asserted for MCLK = BCLK		8		8	T_{CLK}
t7b	AS# asserted for MCLK = BCLK ÷ 2		11		11	T_{CLK}
t7c	AS# asserted for MCLK = BCLK ÷ 3		13		13	T_{CLK}
t7d	AS# asserted for MCLK = BCLK ÷ 4		18		18	T_{CLK}
t8	AS# setup to CLK rising edge while CS#, AS#, UDS#/LDS# = 0	1		1		ns
t9	AS# setup to CLK rising edge	1		2		ns
t10	UDS#/LDS# setup to CLK rising edge while CS#, AS#, UDS#/LDS# = 0	3		1		ns
t11	UDS#/LDS# high setup to CLK rising edge	3		2		ns
t12	First CLK rising edge where AS# = 1 to DTACK# high impedance	5	40	3	14	ns
t13	R/W# setup to CLK rising edge before all CS#, AS#, UDS# and/or LDS# = 0	0		1		ns
t14	R/W# hold from AS# rising edge	0		0		ns
t15	AS# = 0 and CS# = 0 to DTACK# driven high	4	23	3	13	ns
t16	AS# rising edge to DTACK# rising edge	6	39	4	16	ns
t17	D[15:0] valid to third CLK rising edge where CS# = 0, AS# = 0 and either UDS# = 0 or LDS# = 0 (write cycle) (see note 1)	1		0		ns
t18	D[15:0] hold from DTACK# falling edge (write cycle)	0		0		ns
t19	UDS# = 0 and/or LDS# = 0 to D[15:0] driven (read cycle)	4	27	3	13	ns
t20	DTACK# falling edge to D[15:0] valid (read cycle)		0		2	ns
t21	UDS#, LDS# rising edge to D[15:0] high impedance (read cycle)	5	33	3	13	ns

1. t17 is the delay from when data is placed on the bus until the data is latched into the write buffer.

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

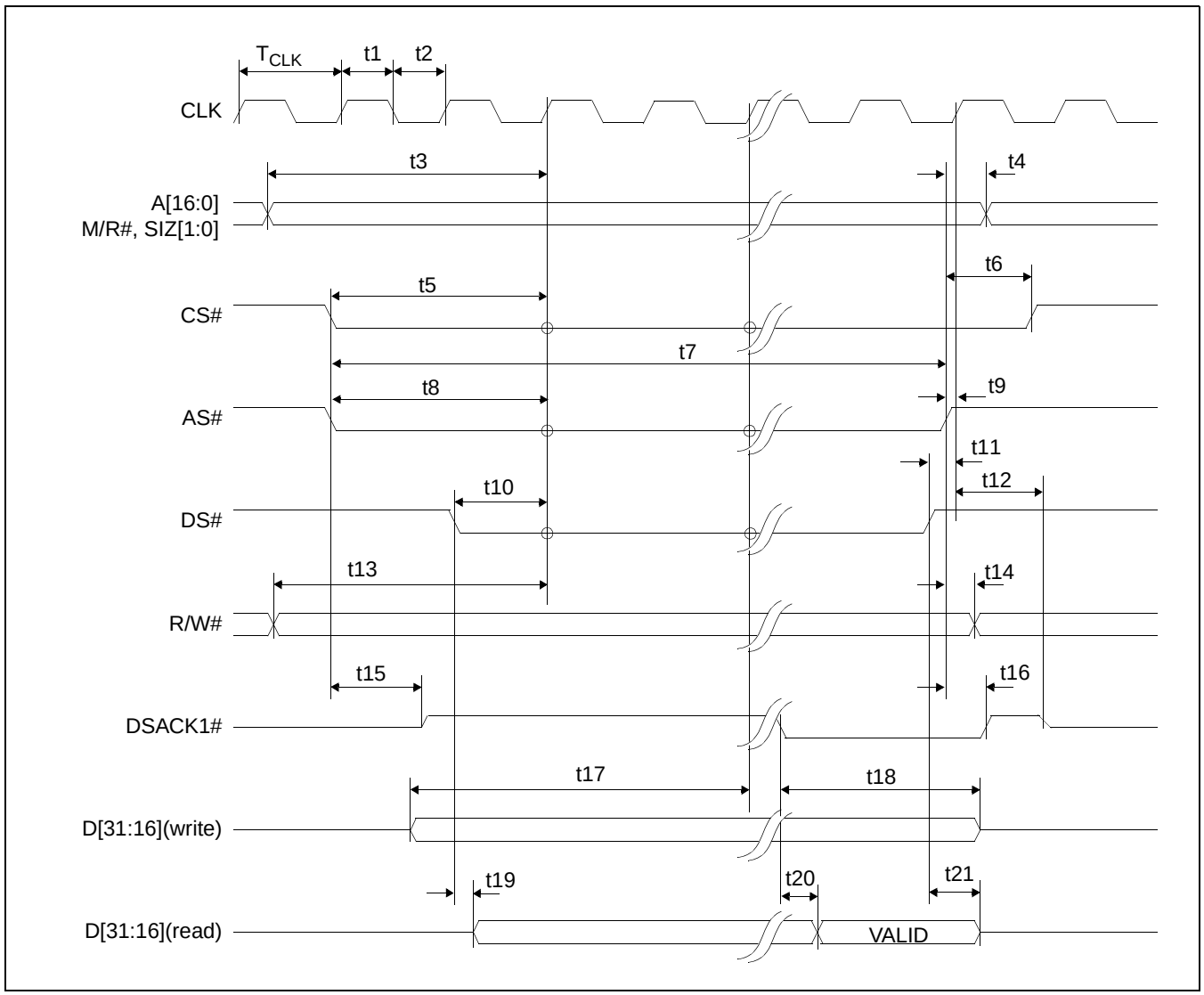


Figure 6-7: Motorola MC68K #2 Interface Timing

Note

For information on the implementation of the Motorola 68K #2 Host Bus Interface, see *Interfacing To The Motorola MC68030 Microprocessor*, document number X31B-G-013-xx.

Table 6-10: Motorola MC68K #2 Interface Timing

Symbol	Parameter	2.0V		3.3V		Unit
		Min	Max	Min	Max	
f_{CLK}	Bus Clock frequency		20		50	MHz
T_{CLK}	Bus Clock period	$1/f_{CLK}$		$1/f_{CLK}$		ns
t1	Clock pulse width high	22.5		9		ns
t2	Clock pulse width low	22.5		9		ns
t3	A[16:0], SIZ[1:0], M/R# setup to first CLK rising edge where CS# = 0, AS# = 0, DS# = 0	1		1		ns
t4	A[16:0], SIZ[1:0], M/R# hold from AS# rising edge	0		0		ns
t5	CS# setup to CLK rising edge	0		1		ns
t6	CS# hold from AS# rising edge	0		0		ns
t7a	AS# asserted for MCLK = BCLK		8		8	T_{CLK}
t7b	AS# asserted for MCLK = BCLK ÷ 2		11		11	T_{CLK}
t7c	AS# asserted for MCLK = BCLK ÷ 3		13		13	T_{CLK}
t7d	AS# asserted for MCLK = BCLK ÷ 4		18		18	T_{CLK}
t8	AS# falling edge to CLK rising edge	1		1		ns
t9	AS# rising edge to CLK rising edge	1		3		ns
t10	DS# falling edge to CLK rising edge	1		1		ns
t11	DS# setup to CLK rising edge	1		3		ns
t12	First CLK where AS# = 1 to DSACK1# high impedance	5	40	3	14	ns
t13	R/W# setup to CLK rising edge before all CS# = 0, AS# = 0, and DS# = 0	1		1		ns
t14	R/W# hold from AS# rising edge	0		0		ns
t15	AS# = 0 and CS# = 0 to DSACK1# rising edge	4	23	3	14	ns
t16	AS# rising edge to DSACK1# rising edge	6	39	4	17	ns
t17	D[31:16] valid to third CLK rising edge where CS# = 0, AS# = 0, and DS# = 0 (write cycle) (see note 1)	1		0		ns
t18	D[31:16] hold from falling edge of DSACK1# (write cycle)	0		0		ns
t19	DS# falling edge to D[31:16] driven (read cycle)	4	32	3	14	ns
t20	DSACK1# falling edge to D[31:16] valid (read cycle)		0		2	ns
t21	DS# rising edge to D[31:16] invalid/high impedance (read cycle)	5	36	3	13	ns

1. t17 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.7 Motorola REDCAP2 Interface Timing

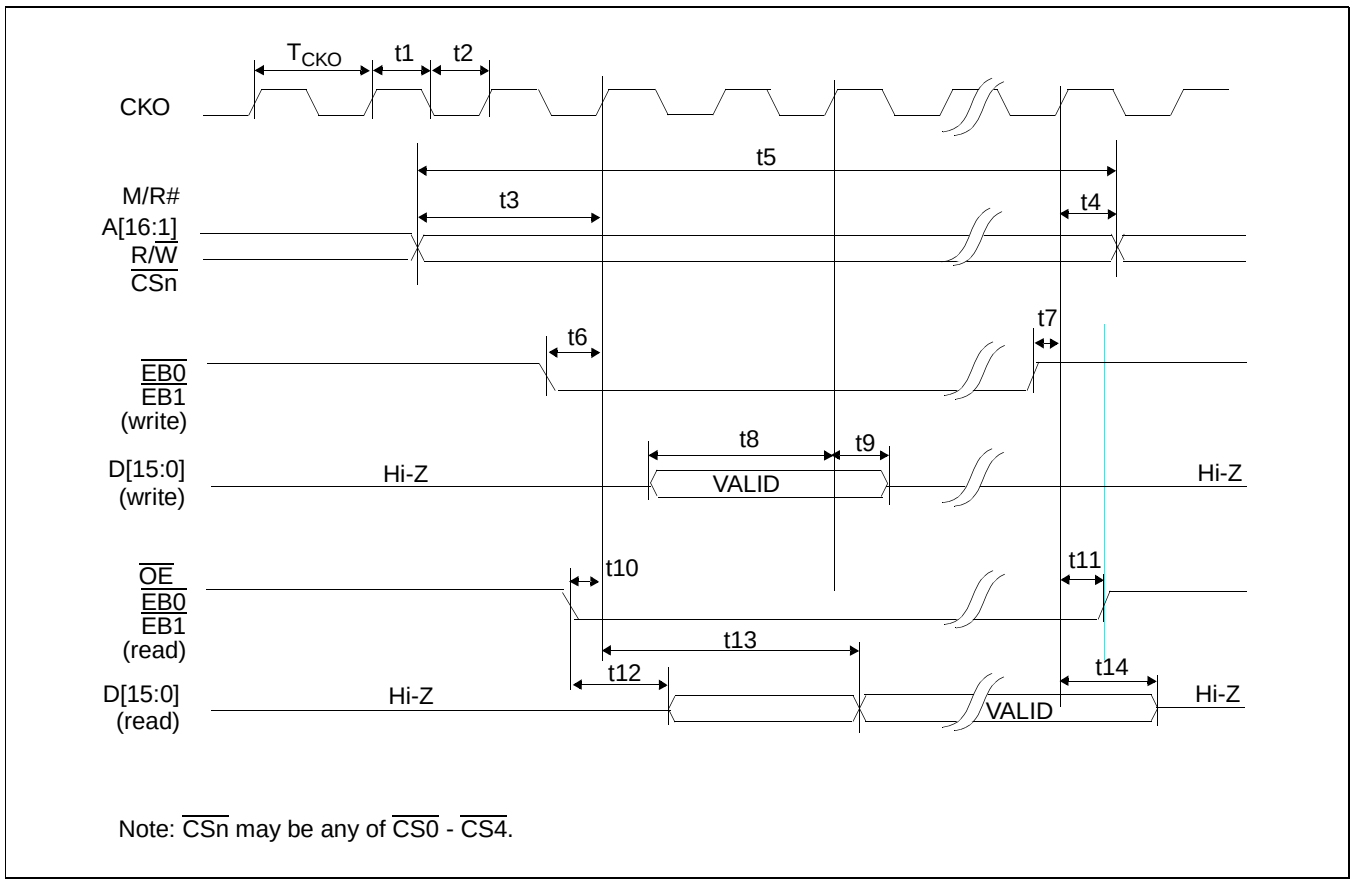


Figure 6-8: Motorola REDCAP2 Interface Timing

Note

For further information on implementing the REDCAP2 microprocessor, see *Interfacing to the Motorola REDCAP2 DSP with Integrated MCU*, document number X31B-G-013-xx.

Table 6-11: Motorola REDCAP2 Interface Timing

Symbol	Parameter	2.0V		3.3V		Units
		Min	Max	Min	Max	
f_{CKO}	Bus Clock frequency		17		17	MHz
T_{CKO}	Bus Clock period	$1/f_{CKO}$		$1/f_{CKO}$		ns
t1	Bus Clock pulse width low	26		26		ns
t2	Bus Clock pulse width high	26		26		ns
t3	A[16:1], M/R#, R/$\overline{W}$, $\overline{CSn}$ setup to CKO rising edge	1		1		ns
t4	A[16:1], M/R#, R/$\overline{W}$, $\overline{CSn}$ hold from CKO rising edge	0		0		ns
t5a	$\overline{CSn}$ asserted for MCLK = BCLK	8		8		T_{CKO}
t5b	$\overline{CSn}$ asserted for MCLK = BCLK ÷ 2	10		10		T_{CKO}
t5c	$\overline{CSn}$ asserted for MCLK = BCLK ÷ 3	13		13		T_{CKO}
t5d	$\overline{CSn}$ asserted for MCLK = BCLK ÷ 4	15		15		T_{CKO}
t6	$\overline{EB0}$, $\overline{EB1}$ asserted to CKO rising edge (write cycle)	1		1		ns
t7	$\overline{EB0}$, $\overline{EB1}$ de-asserted to CKO rising edge (write cycle)	1		4		ns
t8	D[15:0] input setup to 3rd CKO rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low (write cycle) (see note 1)	1		0		ns
t9	D[15:0] input hold from 3rd CKO rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low (write cycle)	23		8		ns
t10	$\overline{OE}$, $\overline{EB0}$, $\overline{EB1}$ setup to CKO rising edge (read cycle)	1		0		ns
t11	$\overline{OE}$, $\overline{EB0}$, $\overline{EB1}$ hold to CKO rising edge (read cycle)	1		0		ns
t12	D[15:0] output delay from $\overline{OE}$, $\overline{EB0}$, $\overline{EB1}$ falling edge (read cycle)	4	29	3	10	ns
t13a	1st CKO rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low to D[15:0] valid for MCLK = BCLK (read cycle)		4.5CKO + 7		4.5CKO + 20	ns
t13b	1st CKO rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 2 (read cycle)		7CKO + 10		6.5CKO + 20	ns
t13c	1st CKO rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 3 (read cycle)		8.5CKO + 8		9.5CKO + 20	ns
t13d	1st CKO rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 4 (read cycle)		9CKO + 11		11.5CKO + 20	ns
t14	CKO rising edge to D[15:0] output in Hi-Z (read cycle)	4	31	1	11	ns

1. t8 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.8 Motorola DragonBall Interface Timing with $\overline{\text{DTACK}}$ (e.g. MC68EZ328/MC68VZ328)

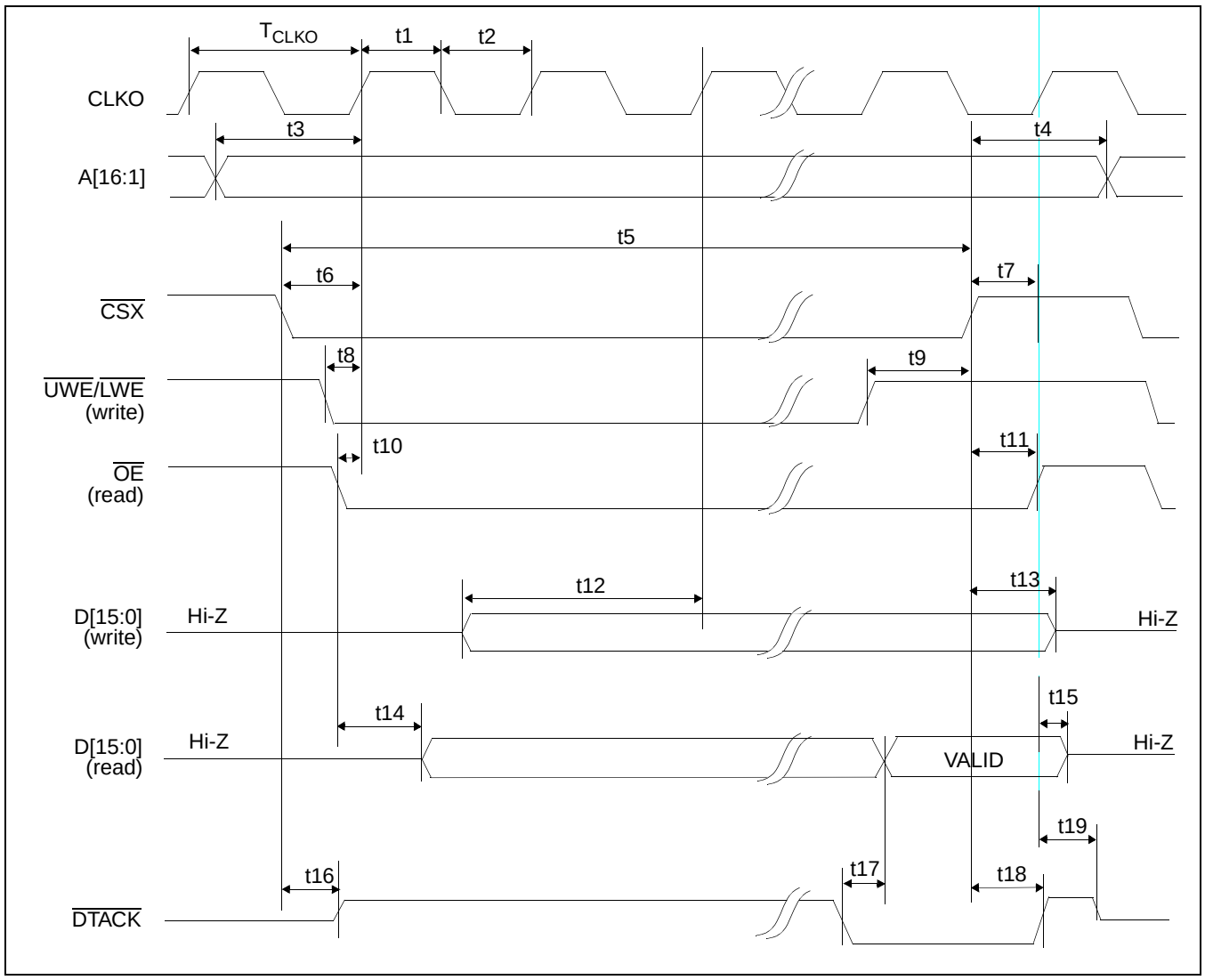


Figure 6-9: Motorola DragonBall Interface with $\overline{\text{DTACK}}$ Timing

Table 6-12: Motorola DragonBall Interface with $\overline{DTACK}$ Timing

Symbol	Parameter	MC68EZ328				MC68VZ328				Unit
		2.0V		3.3V		2.0V		3.3V		
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{CLKO}	Bus Clock frequency		16		16		20		33	MHz
T_{CLKO}	Bus Clock period	$1/f_{CLKO}$		$1/f_{CLKO}$		$1/f_{CLKO}$		$1/f_{CLKO}$		ns
t1	Clock pulse width high	28.1		28.1		22.5		13.5		ns
t2	Clock pulse width low	28.1		28.1		22.5		13.5		ns
t3	A[16:1] setup 1st CLKO when $\overline{CSX}$ = 0 and either $\overline{UWE}$/$\overline{LWE}$ or $\overline{OE}$ = 0	0		0		0		0		ns
t4	A[16:1] hold from $\overline{CSX}$ rising edge	0		0		0		0		ns
t5a	$\overline{CSX}$ asserted for MCLK = BCLK		8		8		8		8	T_{CLKO}
t5b	$\overline{CSX}$ asserted for MCLK = BCLK ÷ 2		11		11		11		11	T_{CLKO}
t5c	$\overline{CSX}$ asserted for MCLK = BCLK ÷ 3		13		13		13		13	T_{CLKO}
t5d	$\overline{CSX}$ asserted for MCLK = BCLK ÷ 4		17		17		17		17	T_{CLKO}
t6	$\overline{CSX}$ setup to CLKO rising edge	0		0		0		0		ns
t7	$\overline{CSX}$ rising edge to CLKO rising edge	0		0		0		0		ns
t8	$\overline{UWE}$/$\overline{LWE}$ falling edge to CLKO rising edge	1		0		1		0		ns
t9	$\overline{UWE}$/$\overline{LWE}$ rising edge to $\overline{CSX}$ rising edge	0		0		0		0		ns
t10	$\overline{OE}$ falling edge to CLKO rising edge	1		1		1		1		ns
t11	$\overline{OE}$ hold from $\overline{CSX}$ rising edge	0		0		0		0		ns
t12	D[15:0] setup to 3rd CLKO when $\overline{CSX}$, $\overline{UWE}$/$\overline{LWE}$ asserted (write cycle) (see note 1)	1		0		1		0		ns
t13	D[15:0] in hold from $\overline{CSX}$ rising edge (write cycle)	0		0		0		0		ns
t14	Falling edge of $\overline{OE}$ to D[15:0] driven (read cycle)	4	30	3	15	4	30	3	15	ns
t15	CLKO rising edge to D[15:0] output Hi-Z (read cycle)	4	21	2	12	4	21	2	12	ns
t16	$\overline{CSX}$ falling edge to $\overline{DTACK}$ driven high	3	20	3	13	3	20	3	13	ns
t17	$\overline{DTACK}$ falling edge to D[15:0] valid (read cycle)		0		2		0		2	ns
t18	$\overline{CSX}$ high to $\overline{DTACK}$ high	5	34	3	16	5	34	3	16	ns
t19	CLKO rising edge to $\overline{DTACK}$ Hi-Z	5	40	1	6	5	40	1	6	ns

1. t12 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.9 Motorola DragonBall Interface Timing w/o $\overline{\text{DTACK}}$ (e.g. MC68EZ328/MC68VZ328)

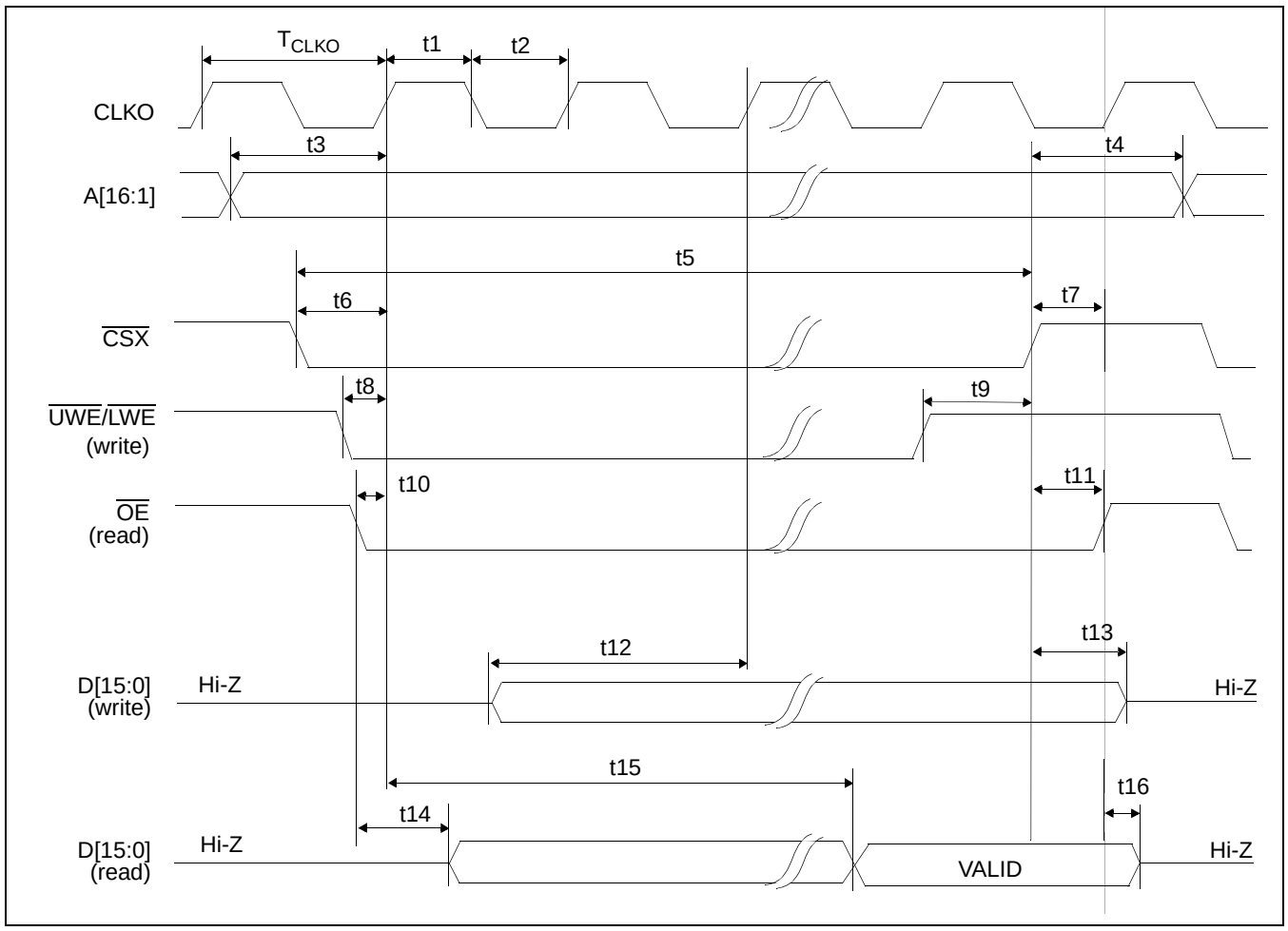


Figure 6-10: Motorola DragonBall Interface without DTACK# Timing

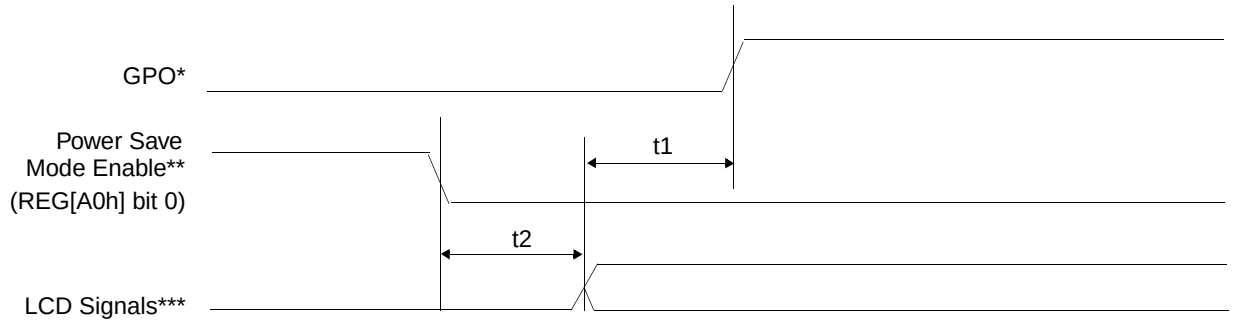
Table 6-13: Motorola DragonBall Interface without $\overline{DTACK}$ Timing

Symbol	Parameter	MC68EZ328				MC68VZ328				Unit
		2.0V		3.3V		2.0V		3.3V		
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{CLKO}	Bus Clock frequency		16		16		20		33	MHz
T_{CLKO}	Bus Clock period	$1/f_{CLKO}$		$1/f_{CLKO}$		$1/f_{CLKO}$		$1/f_{CLKO}$		ns
t1	Clock pulse width high	28.1		28.1		22.5		13.6		ns
t2	Clock pulse width low	28.1		28.1		22.5		13.6		ns
t3	A[16:1] setup 1st CLKO when $\overline{CSX}$ = 0 and either $\overline{UWE}$/$\overline{LWE}$ or $\overline{OE}$ = 0	0		0		0		0		ns
t4	A[16:1] hold from $\overline{CSX}$ rising edge	0		0		0		0		ns
t5a	$\overline{CSX}$ asserted for MCLK = BCLK (CPU wait state register should be programmed to 4 wait states)		8		8		8		8	T_{CLKO}
t5b	$\overline{CSX}$ asserted for MCLK = BCLK ÷ 2 (CPU wait state register should be programmed to 6 wait states)		11		11		11		11	T_{CLKO}
t5c	$\overline{CSX}$ asserted for MCLK = BCLK ÷ 3 (CPU wait state register should be programmed to 10 wait states)	—	Note 1	—	Note 1		13		13	T_{CLKO}
t5d	$\overline{CSX}$ asserted for MCLK = BCLK ÷ 4 (CPU wait state register should be programmed to 12 wait states)	—	Note 1	—	Note 1		17		17	T_{CLKO}
t6	$\overline{CSX}$ setup to CLKO rising edge	0		0		0		0		ns
t7	$\overline{CSX}$ rising edge setup to CLKO rising edge	0		0		0		0		ns
t8	$\overline{UWE}$/$\overline{LWE}$ setup to CLKO rising edge	1		0		1		0		ns
t9	$\overline{UWE}$/$\overline{LWE}$ rising edge to $\overline{CSX}$ rising edge	0		0		0		0		ns
t10	$\overline{OE}$ setup to CLKO rising edge	1		1		1		1		ns
t11	$\overline{OE}$ hold from $\overline{CSX}$ rising edge	0		0		0		0		ns
t12	D[15:0] setup to 3rd CLKO after $\overline{CSX}$, $\overline{UWE}$/$\overline{LWE}$ asserted (write cycle) (see note 2)	1		0		1		0		ns
t13	$\overline{CSX}$ rising edge to D[15:0] output Hi-Z (write cycle)	0		0		0		0		ns
t14	Falling edge of $\overline{OE}$ to D[15:0] driven (read cycle)	4	30	3	15	4	30	3	15	ns
t15a	1st CLKO rising edge after $\overline{OE}$ and $\overline{CSX}$ asserted low to D[15:0] valid for MCLK = BCLK (read cycle)		$5.5T_{CLKO}$ + 4		$5.5T_{CLKO}$ + 20		$5.5T_{CLKO}$ + 4		$5.5T_{CLKO}$ + 20	ns
t15b	1st CLKO rising edge after $\overline{OE}$ and $\overline{CSX}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 2 (read cycle)		$8T_{CLKO}$ + 19		$8.5T_{CLKO}$ + 20		$8T_{CLKO}$ + 19		$8.5T_{CLKO}$ + 20	ns
t15c	1st CLKO rising edge after $\overline{OE}$ and $\overline{CSX}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 3 (read cycle)		$9.5T_{CLKO}$ + 17		$10.5T_{CLKO}$ + 20		$9.5T_{CLKO}$ + 17		$10.5T_{CLKO}$ + 20	ns
t15d	1st CLKO rising edge after $\overline{OE}$ and $\overline{CSX}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 4 (read cycle)		$13T_{CLKO}$ + 9		$14.5T_{CLKO}$ + 20		$13T_{CLKO}$ + 9		$14.5T_{CLKO}$ + 20	ns
t16	CLKO rising edge to D[15:0] output Hi-Z (read cycle)	4	21	2	12	4	21	2	12	ns

1. The MC68EZ328 cannot support the MCLK = BCLK ÷ 3 and MCLK = BCLK ÷ 4 settings without $\overline{DTACK}$.
2. t12 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.3 LCD Power Sequencing

6.3.1 Passive/TFT Power-On Sequence



*It is recommended to use the general purpose output pin GPO to control the LCD bias power.
**The LCD power-on sequence is activated by programming the Power Save Mode Enable bit (REG[A0h] bit 0) to 0.
***LCD Signals include: FPDAT[17:0], FPSHIFT, FPLINE, FPFRAME, and DRDY.

Figure 6-11: Passive/TFT Power-On Sequence Timing

Table 6-14: Passive/TFT Power-On Sequence Timing

Symbol	Parameter	Min	Max	Units
t1	LCD signals active to LCD bias active	Note 1	Note 1	
t2	Power Save Mode disabled to LCD signals active	0	20	ns

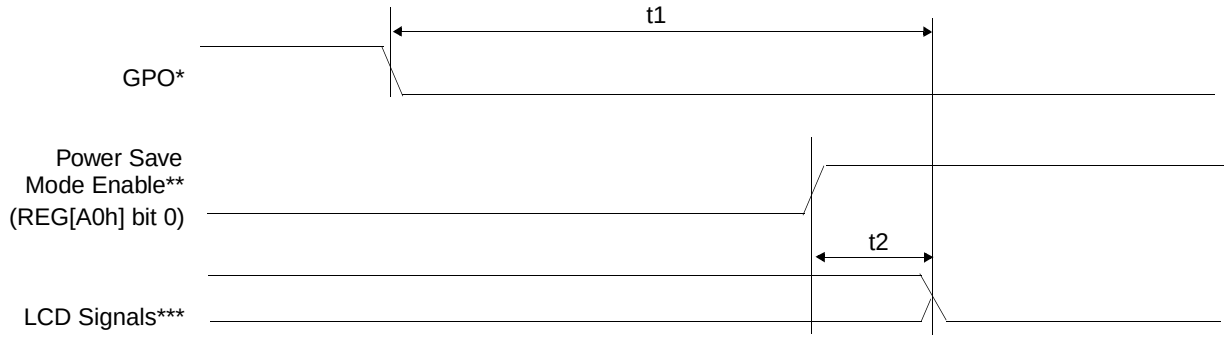
1. t1 is controlled by software and must be determined from the bias power supply delay requirements of the panel connected.

Note

For HR-TFT Power-On/Off sequence information, see *Connecting to the Sharp HR-TFT Panels*, document number X31B-G-011-xx.

For D-TFD Power-On/Off sequence information, see *Connecting to the Epson D-TFD Panels*, document number X31B-G-012-xx.

6.3.2 Passive/TFT Power-Off Sequence



*It is recommended to use the general purpose output pin GPO to control the LCD bias power.
**The LCD power-off sequence is activated by programming the Power Save Mode Enable bit (REG[A0h] bit 0) to 1.
***LCD Signals include: FPDAT[17:0], FPSHIFT, FPLINE, FPFRAME, and DRDY.

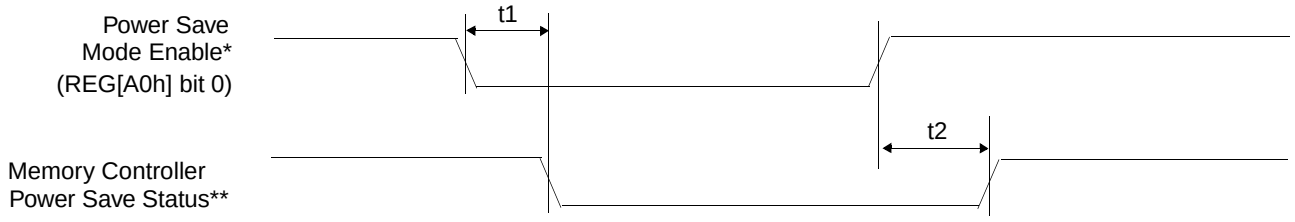
Figure 6-12: Passive/TFT Power-Off Sequence Timing

Table 6-15: Passive/TFT Power-Off Sequence Timing

Symbol	Parameter	Min	Max	Units
t1	LCD bias deactivated to LCD signals inactive	Note 1	Note 1	
t2	Power Save Mode enabled to LCD signals low	0	20	ns

1. t1 is controlled by software and must be determined from the bias power supply delay requirements of the panel connected.

6.3.3 Power Save Status



* Power Save Mode is controlled by the Power Save Mode Enable bit (REG[A0h] bit 0).

** Memory Controller Power Save Status is controlled by the Memory Controller Power Save Status bit (REG[A0h] bit 3).

Figure 6-13: Power Save Status Timing

Table 6-16: Power Save Status Timing

Symbol	Parameter	Min	Max	Units
t1	Power Save Mode disabled to Memory Controller Power Save Status low	0	2	ns
t2	Power Save Mode enabled to Memory Controller Power Save Status high	0	7	MCLK (note 1)

1. For further information on the internal clock MCLK, see Section 7.1.2, “MCLK” on page 88.

6.4 Display Interface

The timing parameters required to drive a flat panel display are shown below. Timing details for each supported panel type are provided in the remainder of this section.

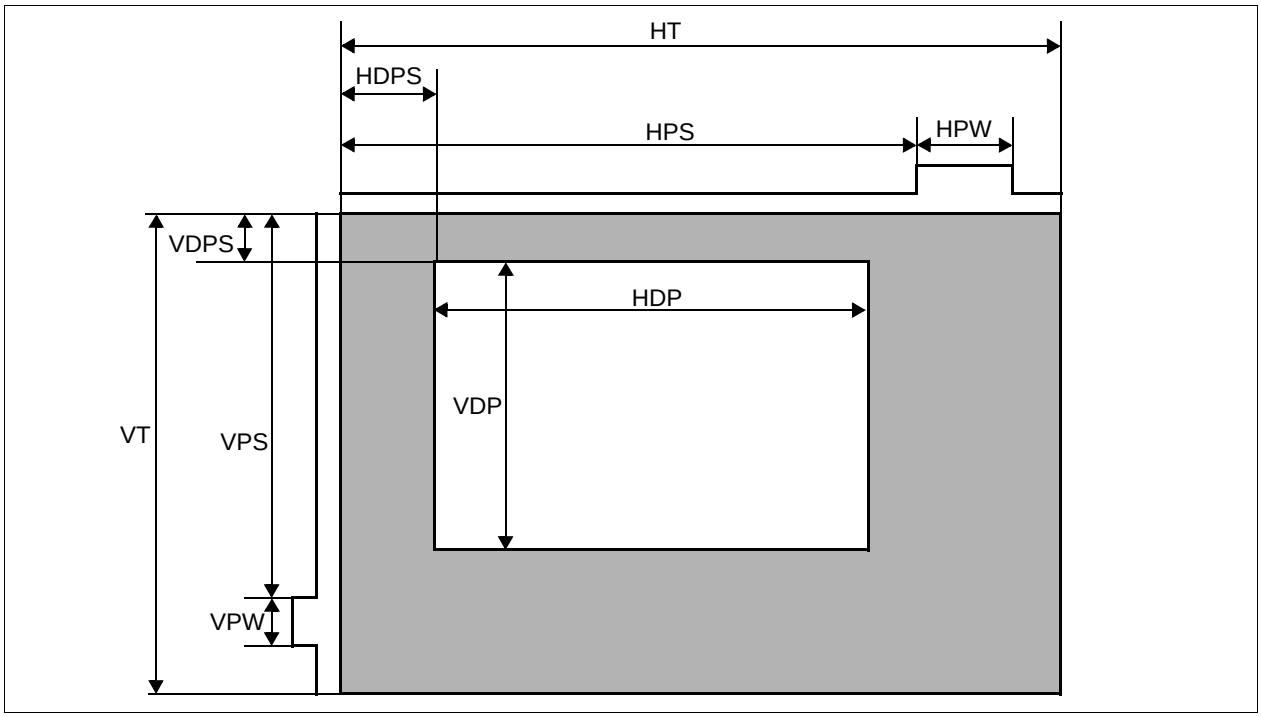


Figure 6-14: Panel Timing Parameters

Table 6-17: Panel Timing Parameter Definition and Register Summary

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

1. For passive panels, the HDP must be a minimum of 32 pixels and can be increased by multiples of 16. For TFT panels, the HDP must be a minimum of 16 pixels and can be increased by multiples of 8.
2. The HDPS parameter contains an offset that depends on the panel type. This offset is the constant in the equation which describes parameter t_{14min} in the AC Timing tables for the various panel types. Note that for passive panels the value of REG[16h], REG[17h] is always 00h so only the offset has any effect.
3. For passive panels the value of VDPS is always 00h.
4. The following formulas must be valid for all panel timings:
   HDPS + HDP < HT
   VDPS + VDP < VT

6.4.1 Generic STN Panel Timing

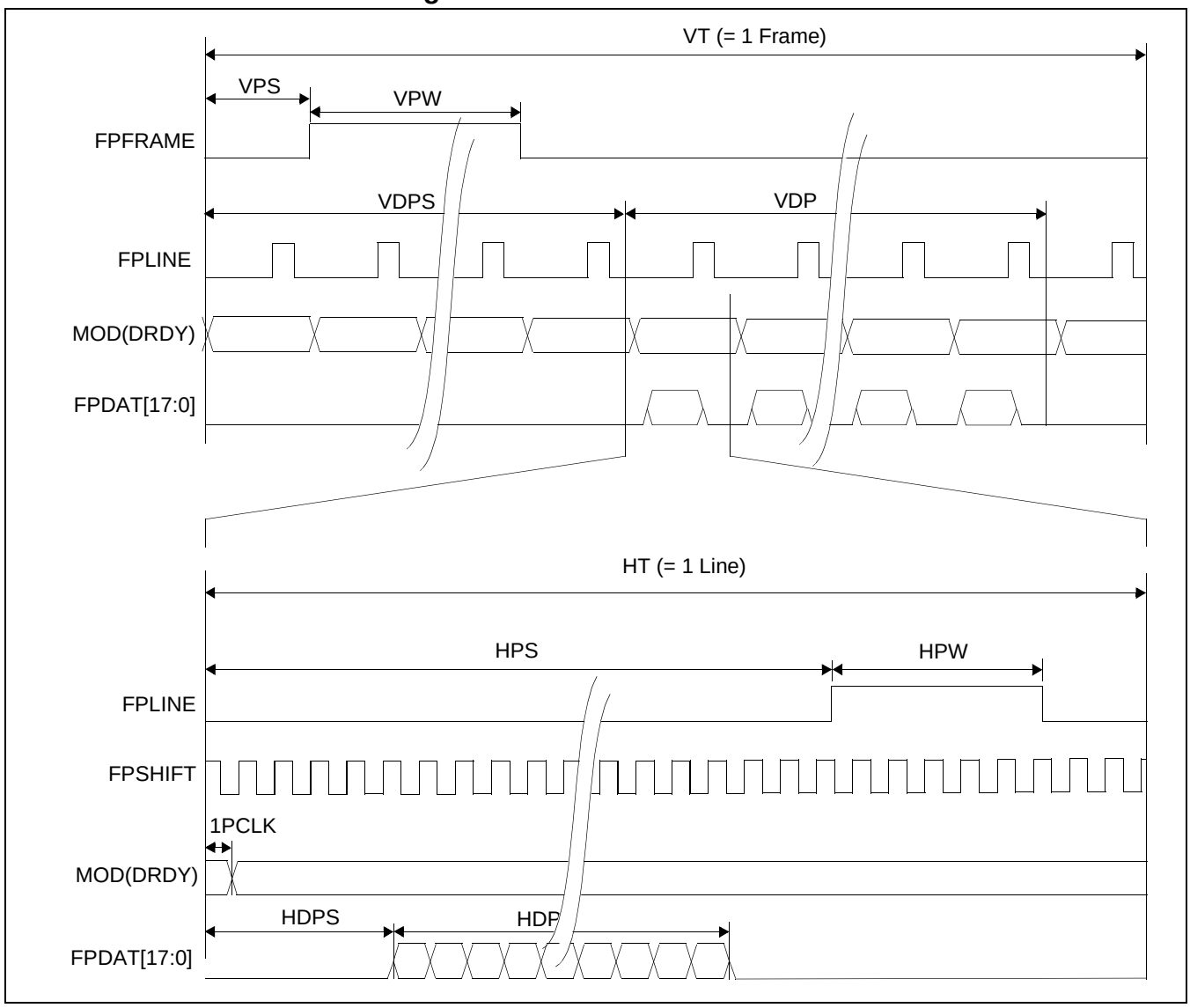


Figure 6-15: Generic STN Panel Timing

VT = Vertical Total
= [(REG[19h] bits 1-0, REG[18h] bits 7-0) + 1] lines
VPS = FPFRAME Pulse Start Position
= [(REG[27h] bits 1-0, REG[26h] bits 7-0)] lines
VPW = FPFRAME Pulse Width
= [(REG[24h] bits 2-0) + 1] lines
VDPS = Vertical Display Period Start Position
= [(REG[1Fh] bits 1-0, REG[1Eh] bits 7-0)] lines
VDP = Vertical Display Period
= [(REG[1Dh] bits 1-0, REG[1Ch] bits 7-0) + 1] lines
HT = Horizontal Total
= [((REG[12h] bits 6-0) + 1) x 8] pixels
HPS = FPLINE Pulse Start Position
= [(REG[23h] bits 1-0, REG[22h] bits 7-0) + 1] pixels
HPW = FPLINE Pulse Width
= [(REG[20h] bits 6-0) + 1] pixels
HDPS = Horizontal Display Period Start Position
= [(REG[17h] bits 1-0, REG[16h] bits 7-0) + 22] pixels
HDP = Horizontal Display Period
= [((REG[14h] bits 6-0) + 1) x 8] pixels
*For passive panels, the HDP must be a minimum of 32 pixels and can be increased by multiples of 16.

*Panel Type Bits (REG[10h] bits 1-0) = 00b (STN)
*FPFRAME Pulse Polarity Bit (REG[24h] bit 7) = 1 (active high)
*FPLINE Polarity Bit (REG[20h] bit 7) = 1 (active high)

6.4.2 Single Monochrome 4-Bit Panel Timing

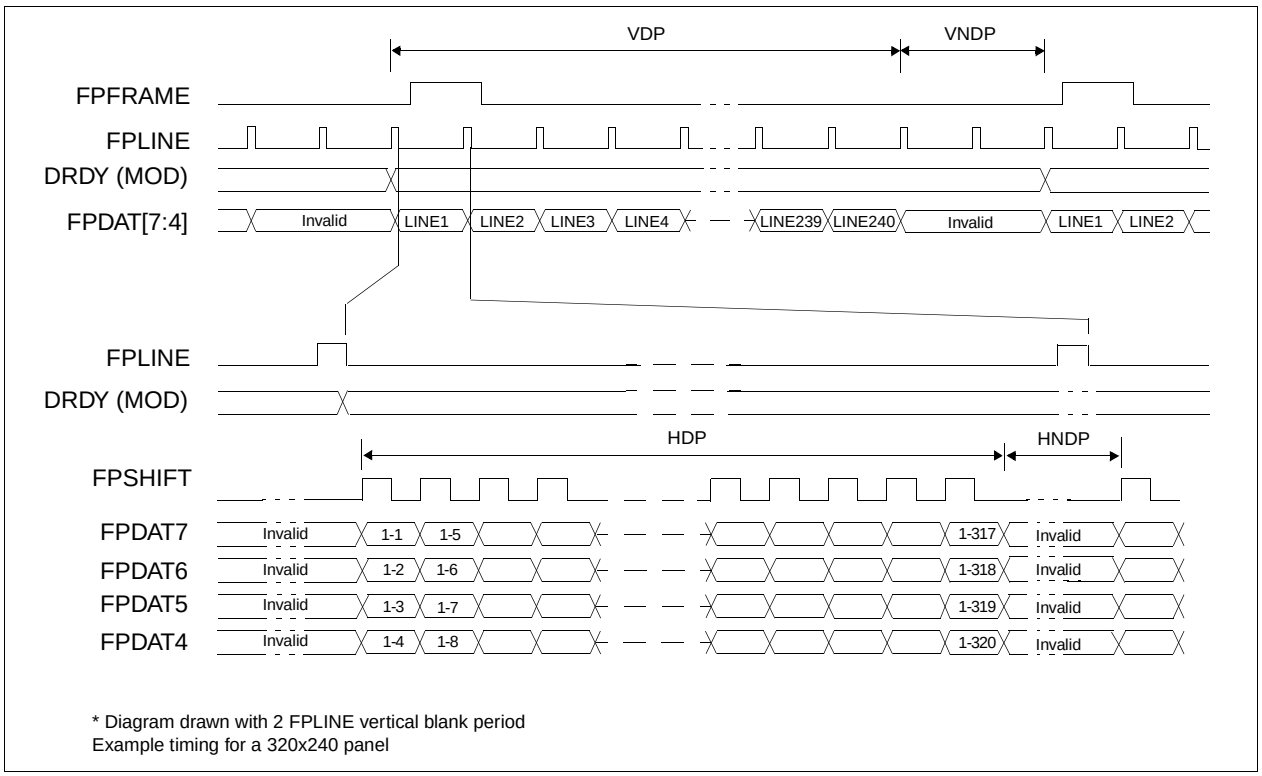


Figure 6-16: Single Monochrome 4-Bit Panel Timing

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

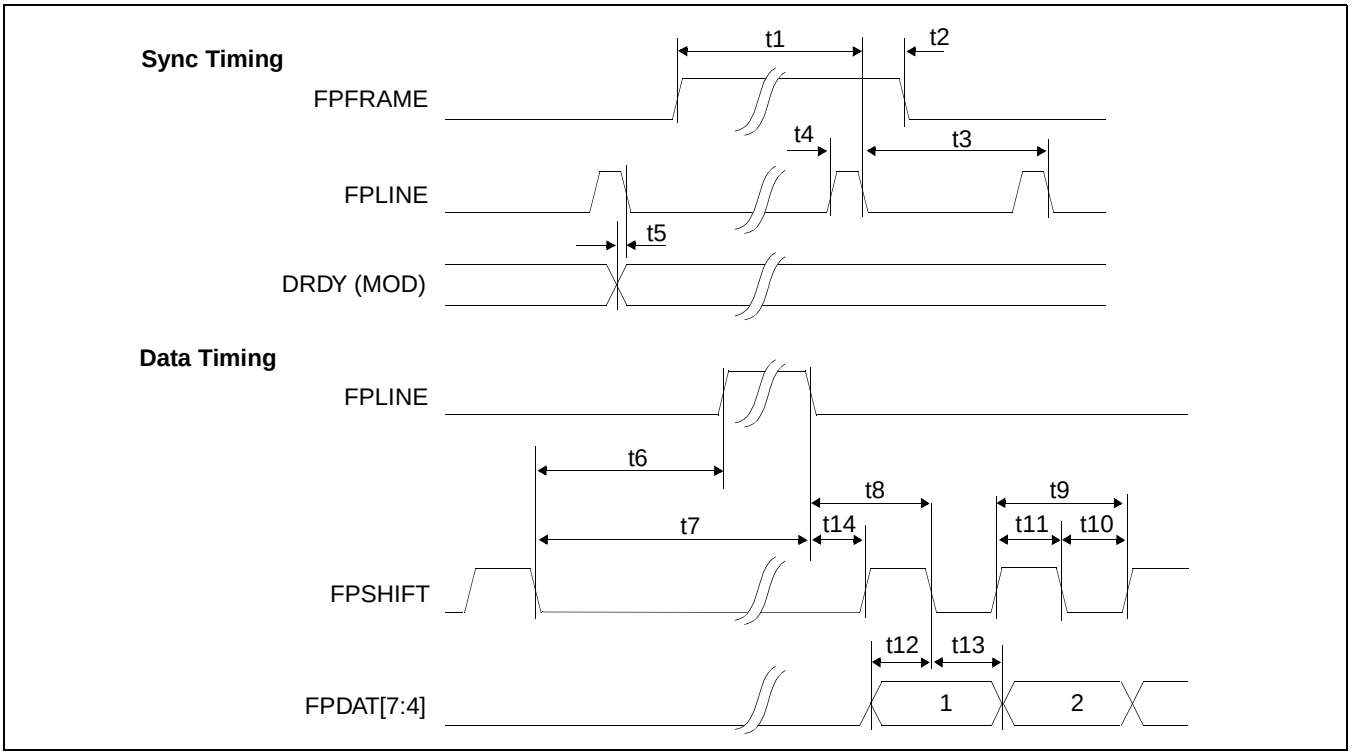


Figure 6-17: Single Monochrome 4-Bit Panel A.C. Timing

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

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

1. Ts = pixel clock period
2. $t1_{min}$ = (HPS + $t4_{min}$ + 1) - (VPS x $t3_{min}$)
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$ + 1) + (VPW - 1 + VPS) x $t3_{min}$
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = $t3_{min}$ - HPS
7. $t6_{min}$ = (HPS + 1) - (HDP + HDPS + 20) if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$ + 1) + 22 if negative add $t3_{min}$

6.4.3 Single Monochrome 8-Bit Panel Timing

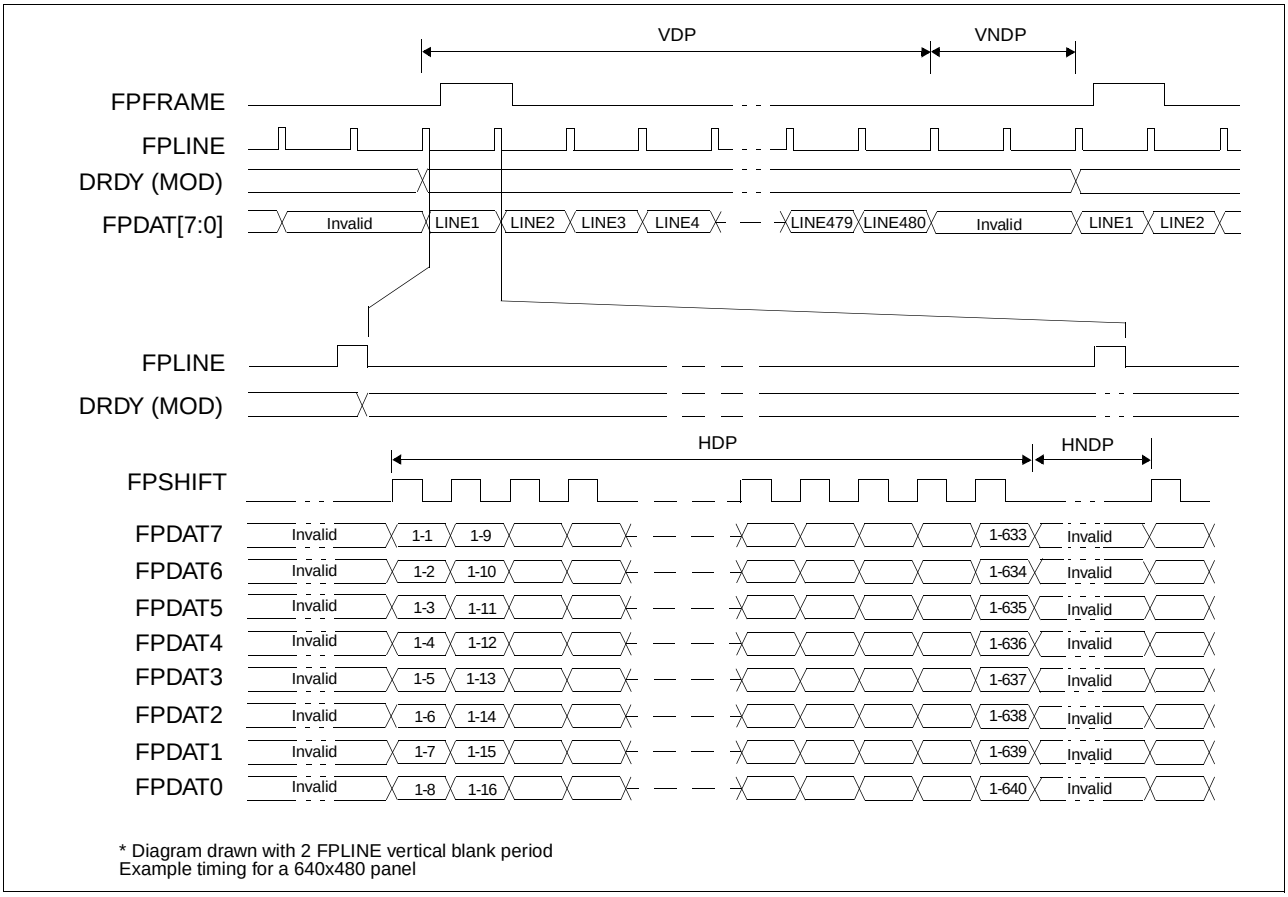


Figure 6-18: Single Monochrome 8-Bit Panel Timing

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

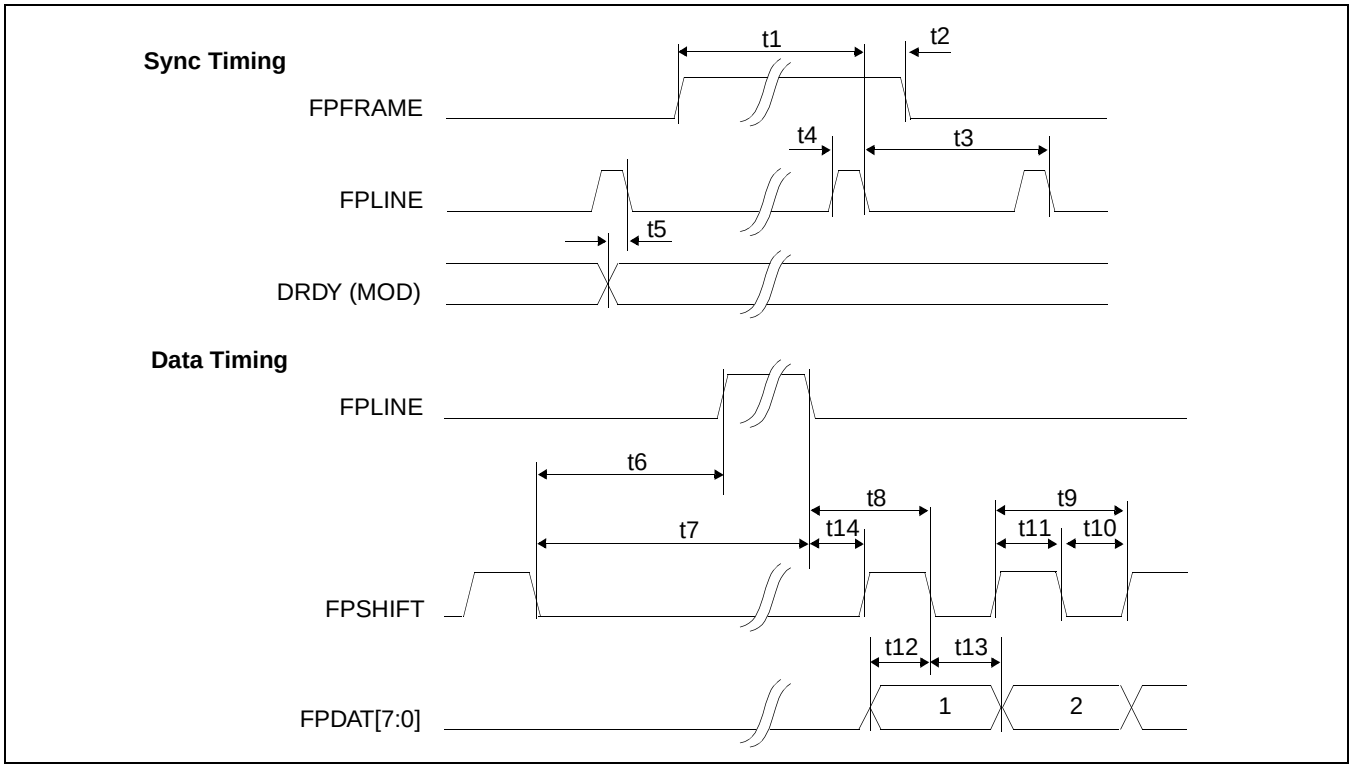


Figure 6-19: Single Monochrome 8-Bit Panel A.C. Timing

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

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

1. Ts = pixel clock period
2. $t1_{min}$ = (HPS + $t4_{min}$ + 1) - (VPS x $t3_{min}$)
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$ + 1) + (VPW - 1 + VPS) x $t3_{min}$
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = $t3_{min}$ - HPS
7. $t6_{min}$ = (HPS + 1) - (HDP + HDPS + 18) if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$ + 1) + 22 if negative add $t3_{min}$

6.4.4 Single Color 4-Bit Panel Timing

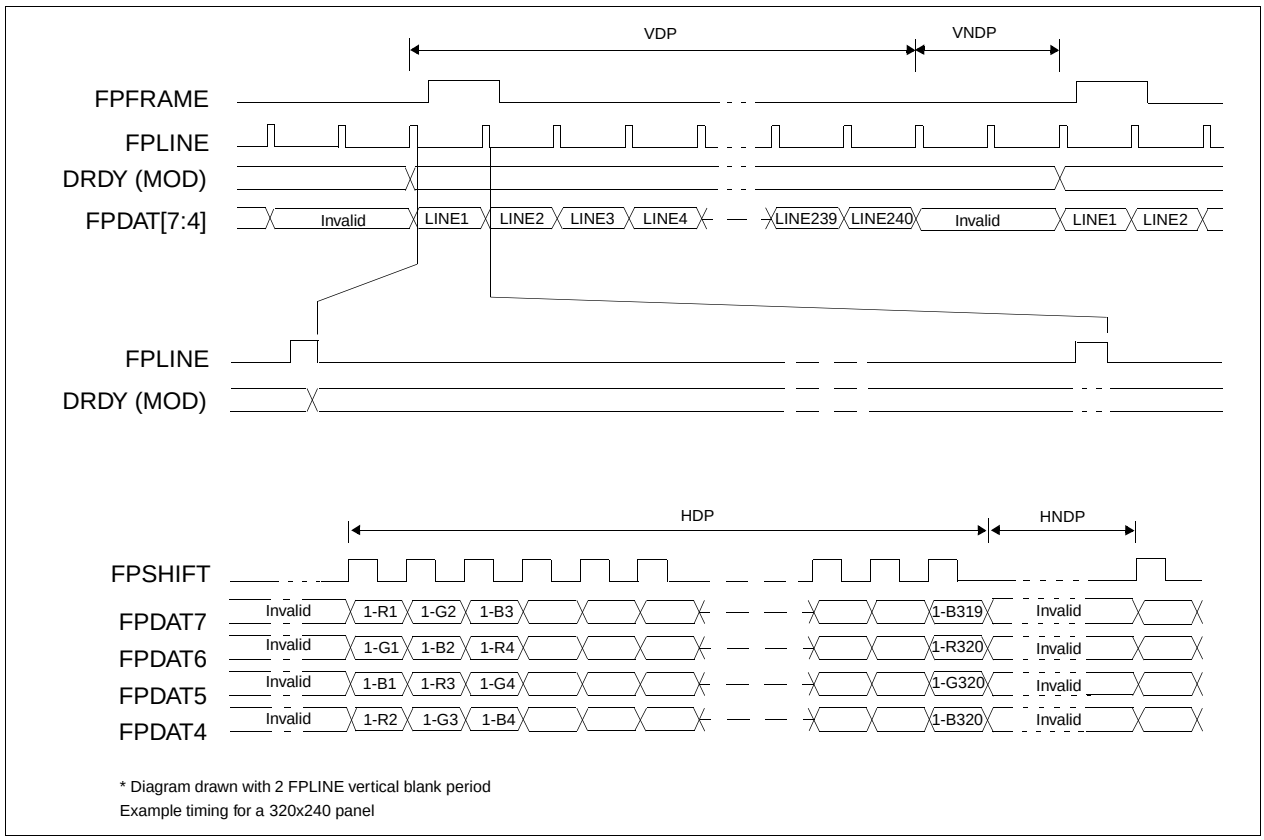


Figure 6-20: Single Color 4-Bit Panel Timing

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

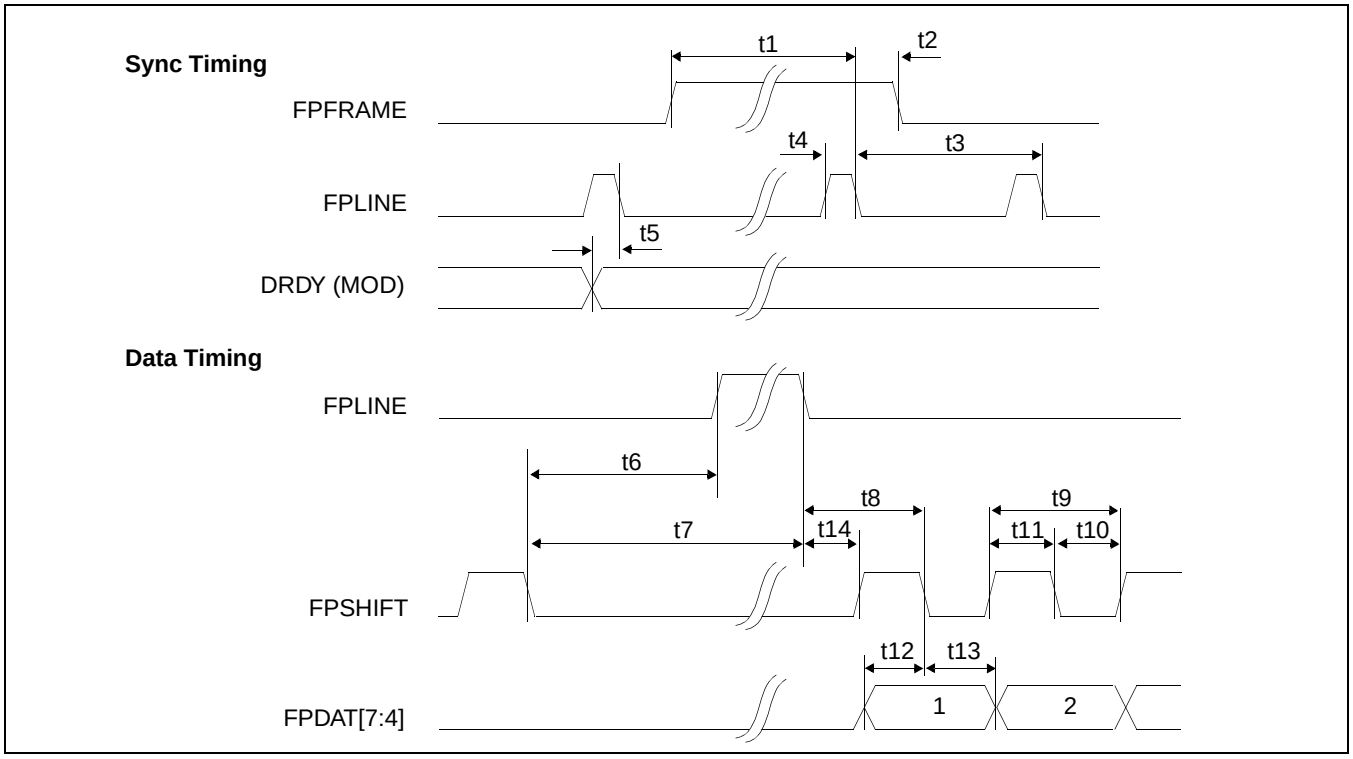


Figure 6-21: Single Color 4-Bit Panel A.C. Timing

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

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	note 3			Ts
t3	FPLINE period	note 4			Ts
t4	FPLINE pulse width	note 5			Ts
t5	MOD transition to FPLINE falling edge	note 6			Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 7			Ts
t7	FPSHIFT falling edge to FPLINE falling edge	t6 + t4			Ts
t8	FPLINE falling edge to FPSHIFT falling edge	t14 + 0.5			Ts
t9	FPSHIFT period	1			Ts
t10	FPSHIFT pulse width low	0.5			Ts
t11	FPSHIFT pulse width high	0.5			Ts
t12	FPDAT[7:4] setup to FPSHIFT falling edge	0.5			Ts
t13	FPDAT[7:4] hold to FPSHIFT falling edge	0.5			Ts
t14	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = (HPS + $t4_{min}$ + 1) - (VPS x $t3_{min}$)
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$ + 1) + (VPW - 1 + VPS) x $t3_{min}$
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = $t3_{min}$ - HPS
7. $t6_{min}$ = (HPS + 1) - (HDP + HDPS + 20.5) if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$ + 1) + 23 if negative add $t3_{min}$

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

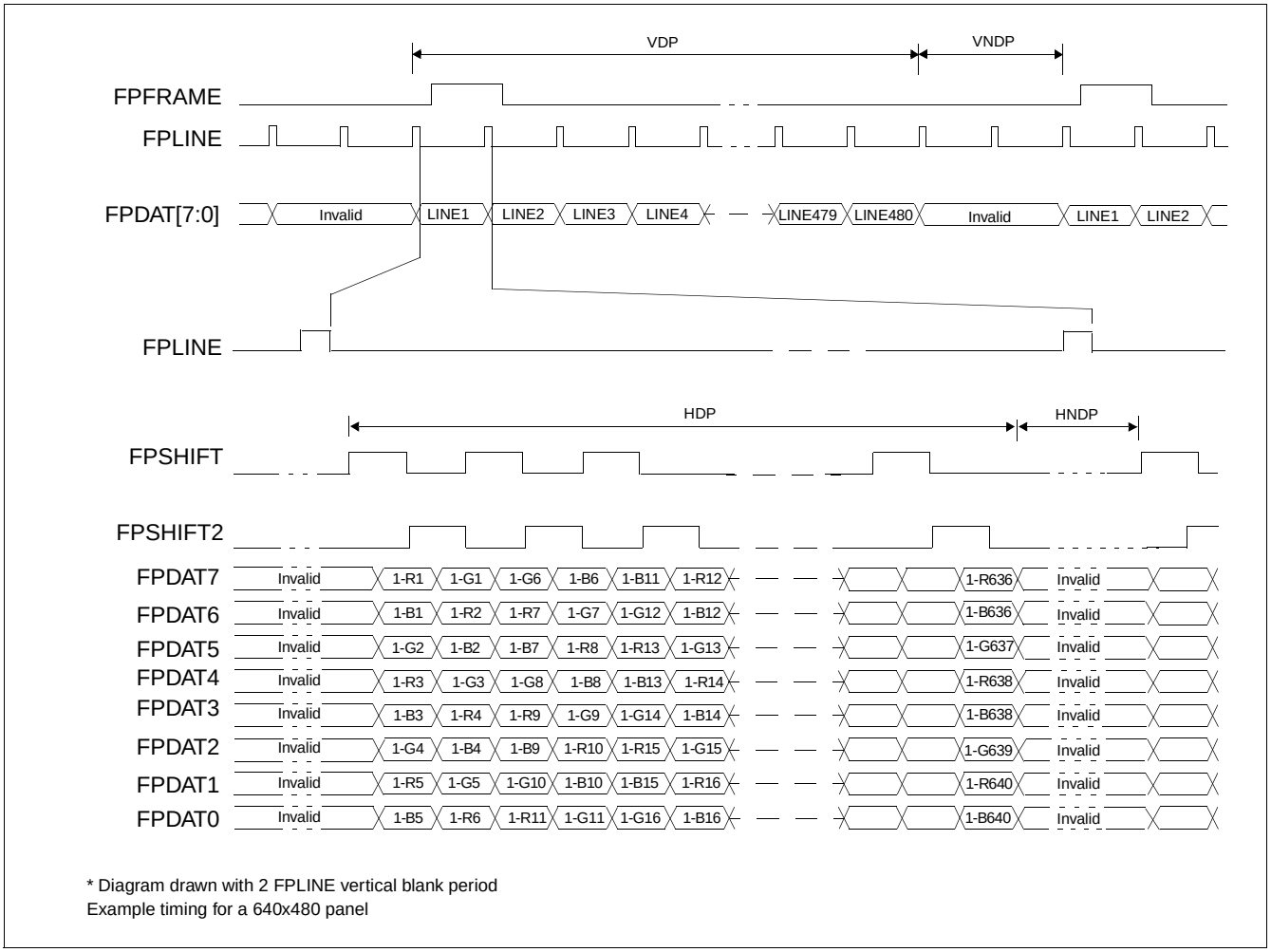


Figure 6-22: Single Color 8-Bit Panel Timing (Format 1)

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

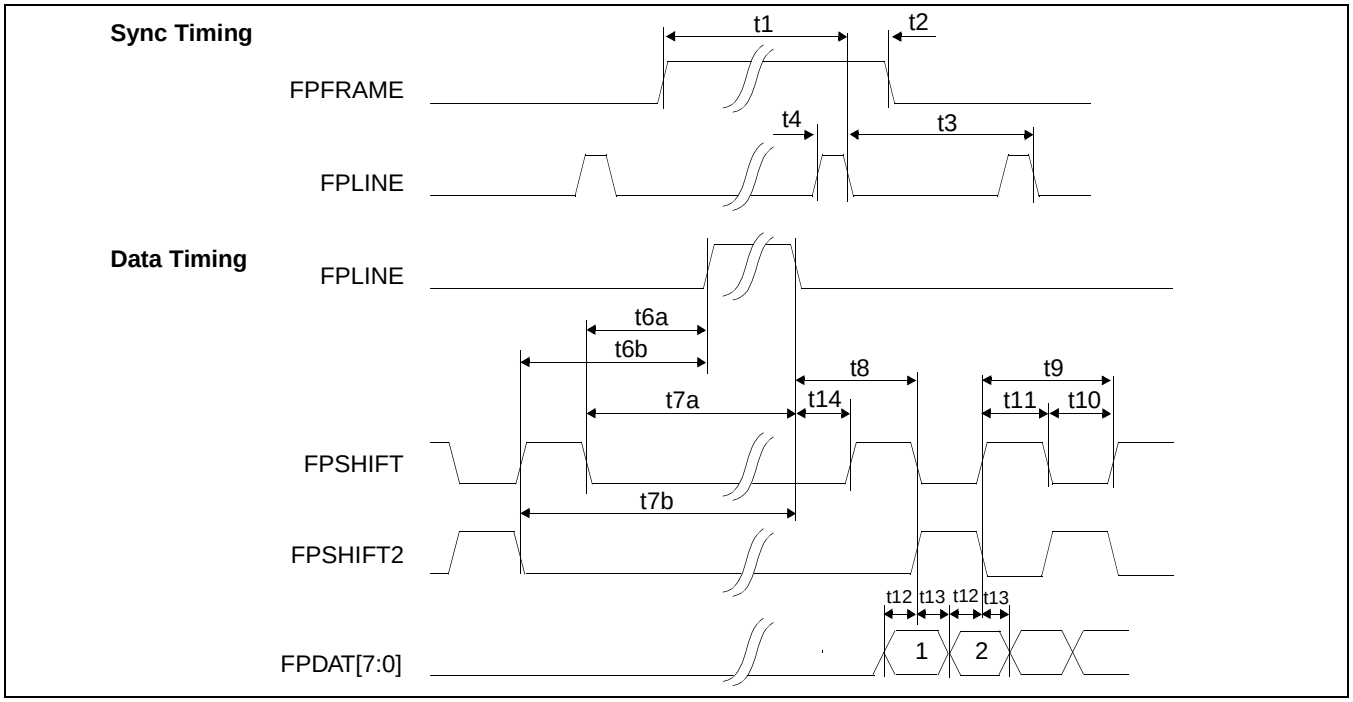


Figure 6-23: Single Color 8-Bit Panel A.C. Timing (Format 1)

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

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	note 3			Ts
t3	FPLINE period	note 4			Ts
t4	FPLINE pulse width	note 5			Ts
t6a	FPSHIFT falling edge to FPLINE rising edge	note 6			Ts
t6b	FPSHIFT2 falling edge to FPLINE rising edge	note 7			Ts
t7a	FPSHIFT falling edge to FPLINE falling edge	t6a + t4			Ts
t7b	FPSHIFT2 falling edge to FPLINE falling edge	t6b + t4			Ts
t8	FPLINE falling edge to FPSHIFT rising, FPSHIFT2 falling edge	t14 + 2			Ts
t9	FPSHIFT2, FPSHIFT period	4			Ts
t10	FPSHIFT2, FPSHIFT pulse width low	2			Ts
t11	FPSHIFT2, FPSHIFT pulse width high	2			Ts
t12	FPDAT[7:0] setup to FPSHIFT2, FPSHIFT falling edge	1			Ts
t13	FPDAT[7:0] hold from FPSHIFT2, FPSHIFT falling edge	1			Ts
t14	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = (HPS + $t4_{min}$ + 1) - (VPS x $t3_{min}$)
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$ + 1) + (VPW - 1 + VPS) x $t3_{min}$
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t6a_{min}$ = (HPS + 1) - (HDP + HDPS + 22) if negative add $t3_{min}$
7. $t6b_{min}$ = (HPS + 1) - (HDP + HDPS + 20) if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$ + 1) + 22 if negative add $t3_{min}$

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

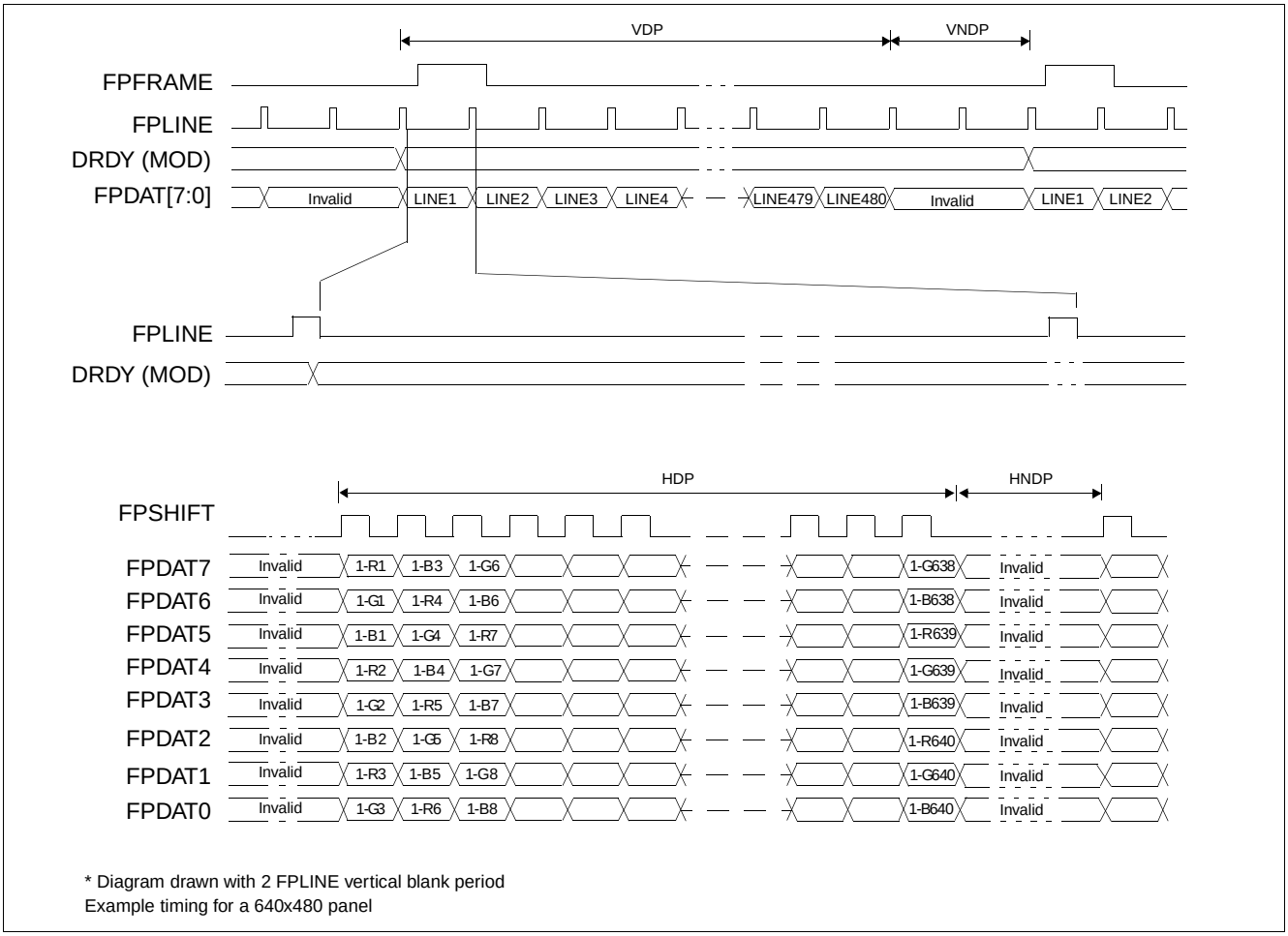


Figure 6-24: Single Color 8-Bit Panel Timing (Format 2)

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

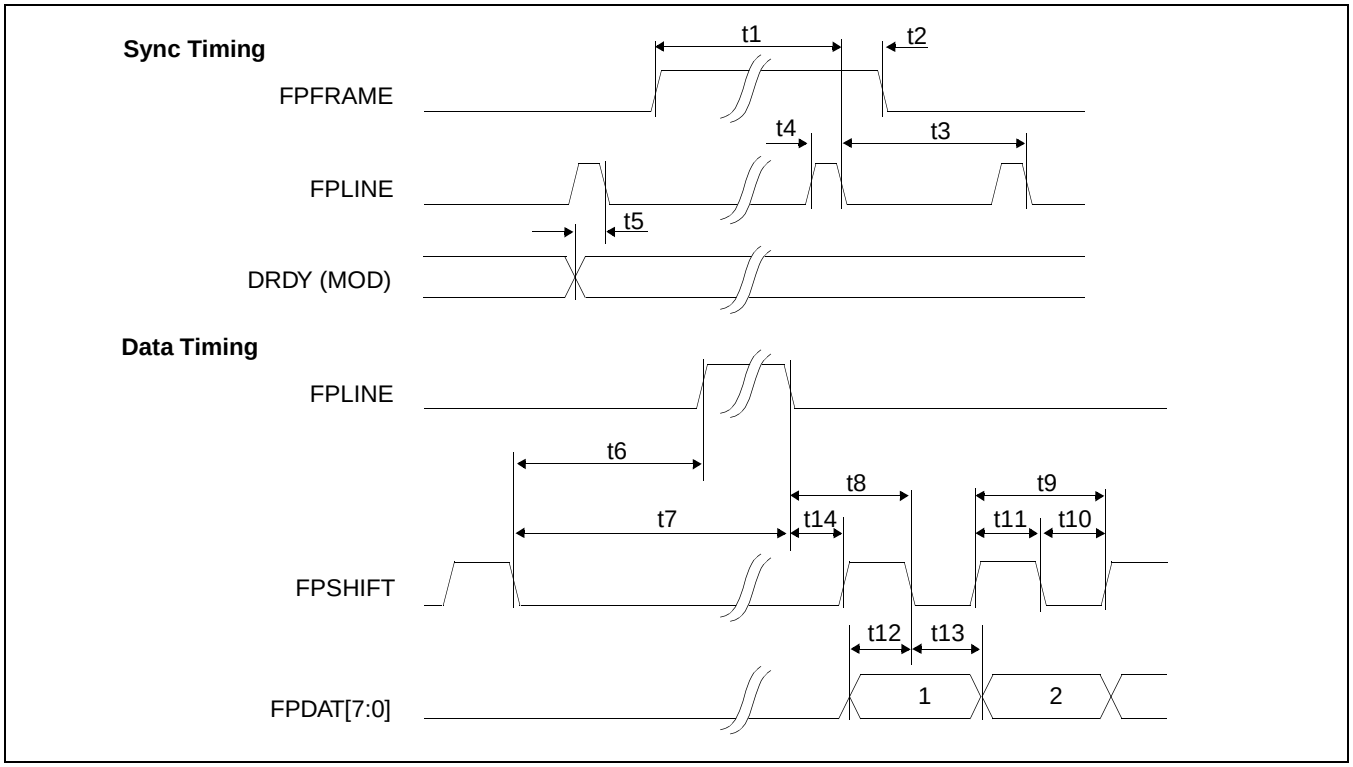


Figure 6-25: Single Color 8-Bit Panel A.C. Timing (Format 2)

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

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

1. Ts = pixel clock period
2. $t1_{min}$ = (HPS + $t4_{min}$ + 1) - (VPS x $t3_{min}$)
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$ + 1) + (VPW - 1 + VPS) x $t3_{min}$
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = $t3_{min}$ - HPS
7. $t6_{min}$ = (HPS + 1) - (HDP + HDPS + 21) if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$ +1) + 22 if negative add $t3_{min}$

6.4.7 Single Color 16-Bit Panel Timing

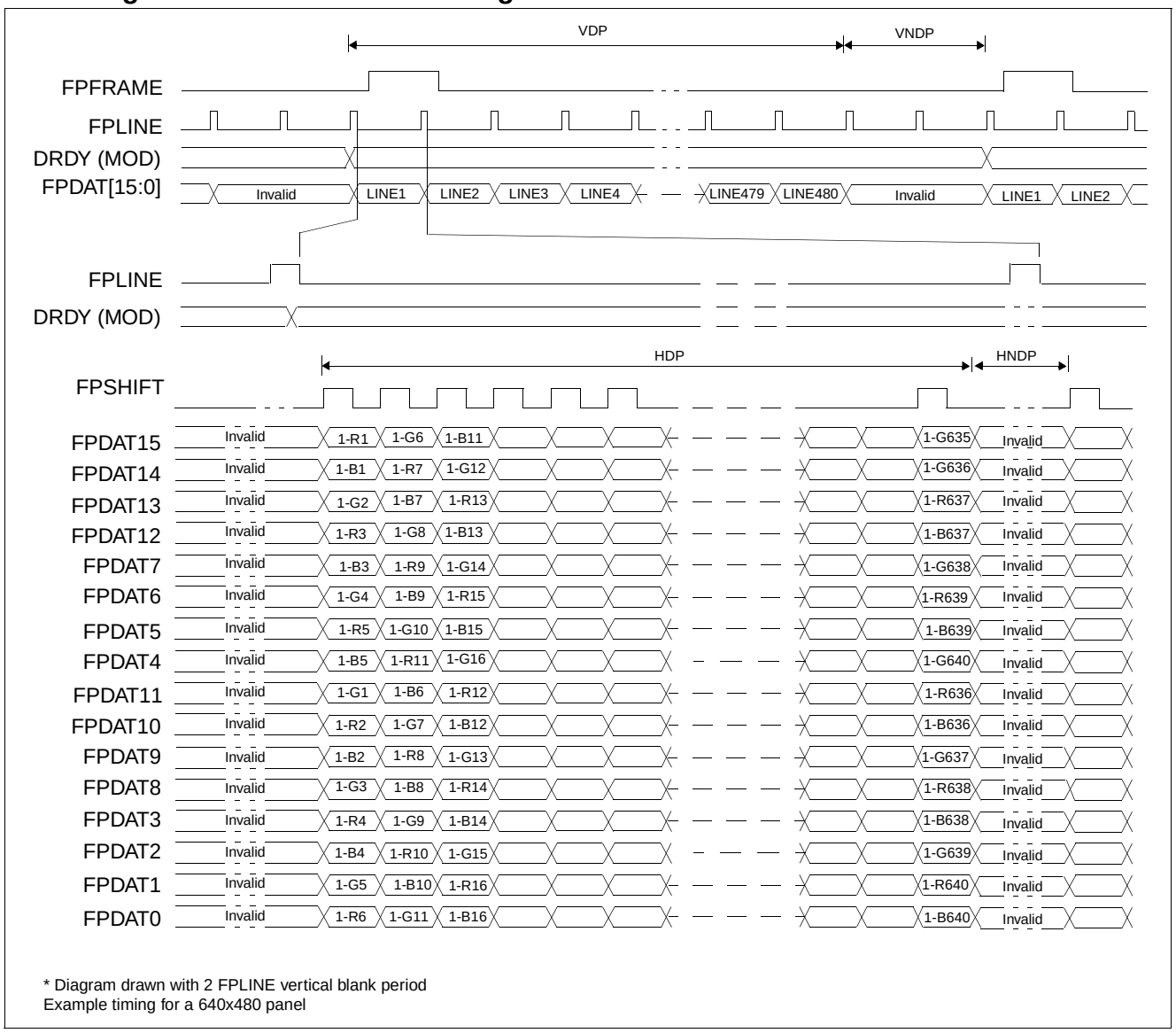


Figure 6-26: Single Color 16-Bit Panel Timing

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

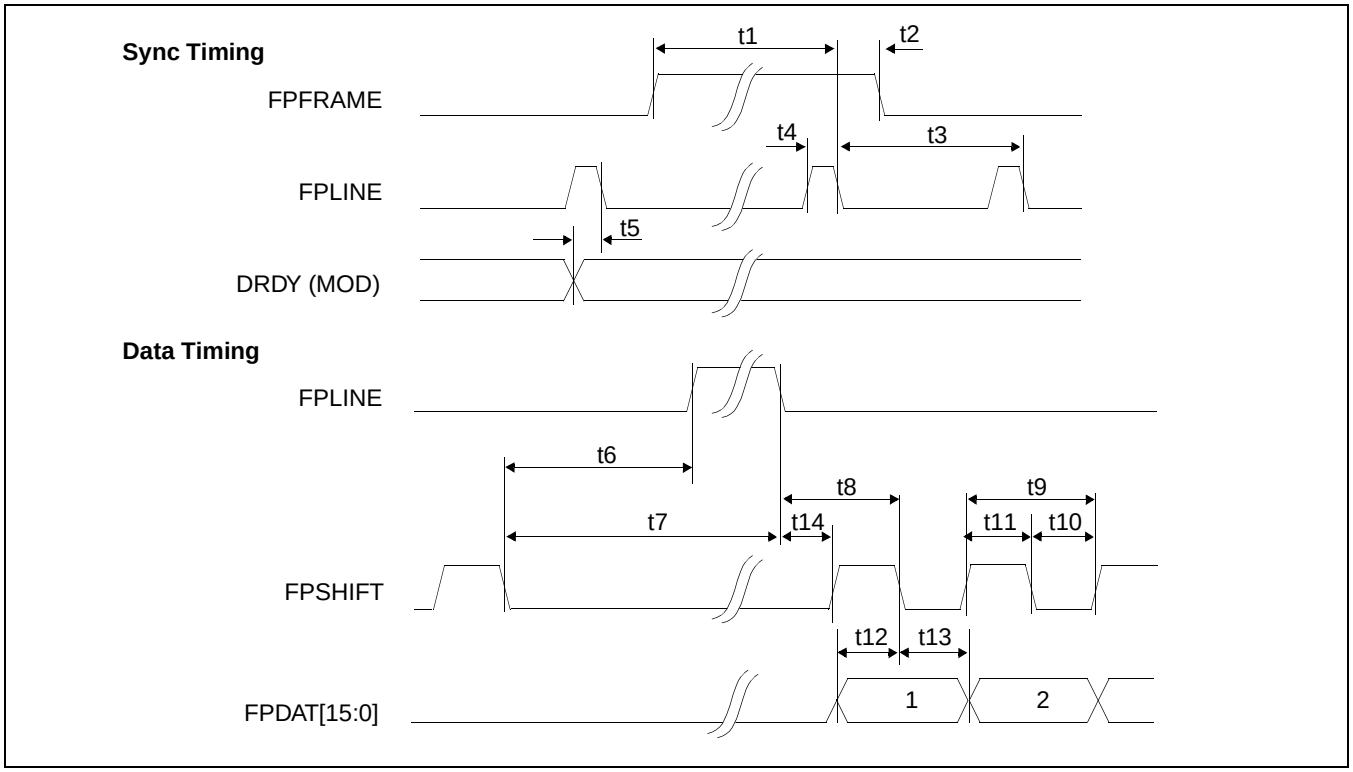


Figure 6-27: Single Color 16-Bit Panel A.C. Timing

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

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	note 3			Ts
t3	FPLINE period	note 4			Ts
t4	FPLINE pulse width	note 5			Ts
t5	MOD transition to FPLINE falling edge	note 6			Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 7			Ts
t7	FPSHIFT falling edge to FPLINE falling edge	t6 + t4			Ts
t8	FPLINE falling edge to FPSHIFT falling edge	t14 + 3			Ts
t9	FPSHIFT period	5			Ts
t10	FPSHIFT pulse width low	2			Ts
t11	FPSHIFT pulse width high	2			Ts
t12	FPDAT[15:0] setup to FPSHIFT rising edge	2			Ts
t13	FPDAT[15:0] hold to FPSHIFT rising edge	2			Ts
t14	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = (HPS + $t4_{min}$ + 1) - (VPS x $t3_{min}$)
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$ + 1) + (VPW - 1 + VPS) x $t3_{min}$
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = $t3_{min}$ - HPS
7. $t6_{min}$ = (HPS + 1) - (HDP + HDPS + 20) if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$ + 1) + 22 if negative add $t3_{min}$

6.4.8 Generic TFT Panel Timing

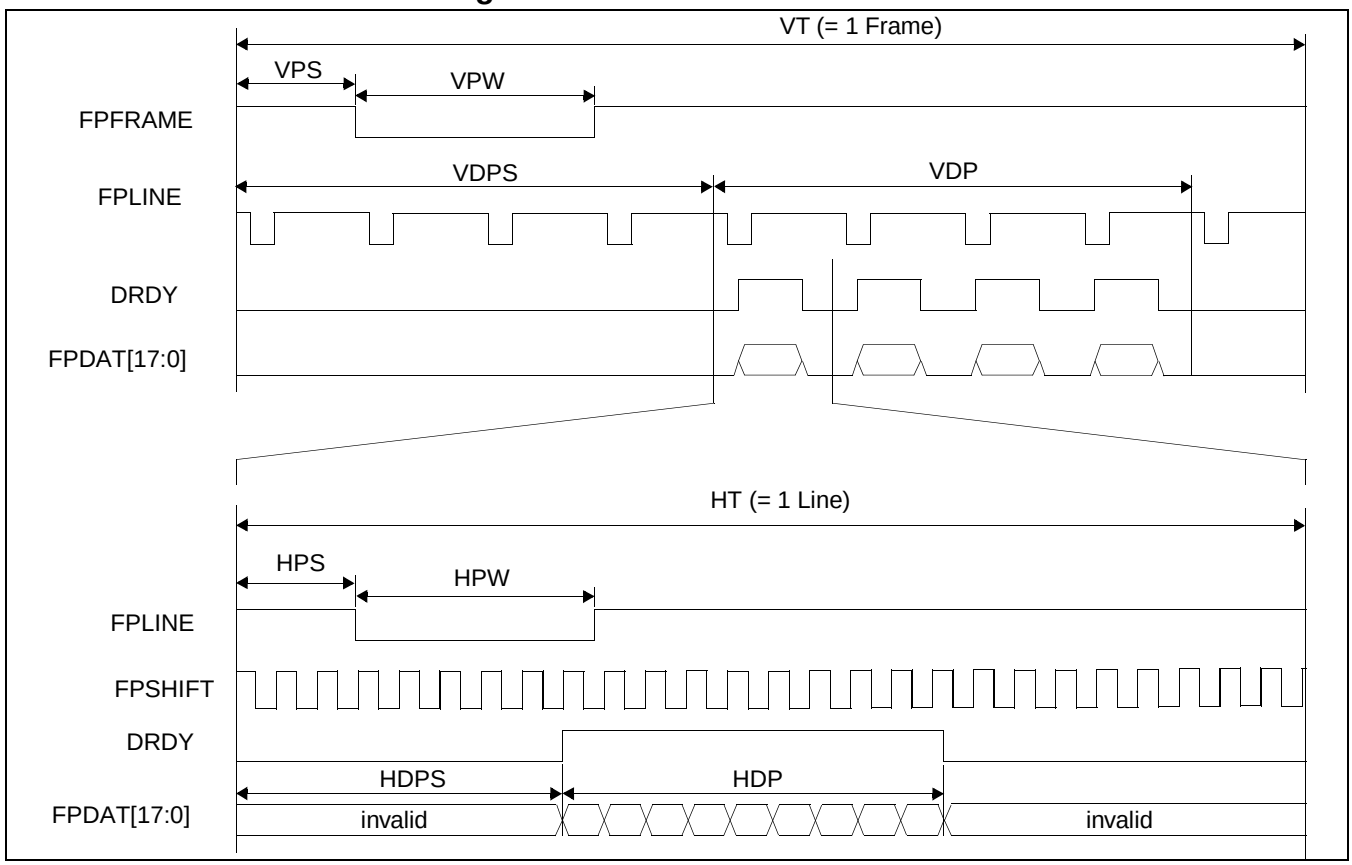


Figure 6-28: Generic TFT Panel Timing

VT = Vertical Total
= [(REG[19h] bits 1-0, REG[18h] bits 7-0) + 1] lines
VPS = FPFRAME Pulse Start Position
= [(REG[27h] bits 1-0, REG[26h] bits 7-0)] lines
VPW = FPFRAME Pulse Width
= [(REG[24h] bits 2-0) + 1] lines
VDPS = Vertical Display Period Start Position
= [(REG[1Fh] bits 1-0, REG[1Eh] bits 7-0)] lines
VDP = Vertical Display Period
= [(REG[1Dh] bits 1-0, REG[1Ch] bits 7-0) + 1] lines
HT = Horizontal Total
= [((REG[12h] bits 6-0) + 1) x 8] pixels
HPS = FPLINE Pulse Start Position
= [(REG[23h] bits 1-0, REG[22h] bits 7-0) + 1] pixels
HPW = FPLINE Pulse Width
= [(REG[20h] bits 6-0) + 1] pixels
HDPS = Horizontal Display Period Start Position
= [(REG[17h] bits 1-0, REG[16h] bits 7-0) + 5] pixels
HDP = Horizontal Display Period
= [((REG[14h] bits 6-0) + 1) x 8] pixels
*For TFT panels, the HDP must be a minimum of 16 pixels and can be increased by multiples of 8.

*Panel Type Bits (REG[10h] bits 1-0) = 01 (TFT)
*FPLINE Pulse Polarity Bit (REG[24h] bit 7) = 0 (active low)
*FPFRAME Polarity Bit (REG[20h] bit 7) = 0 (active low)

6.4.9 9/12/18-Bit TFT Panel Timing

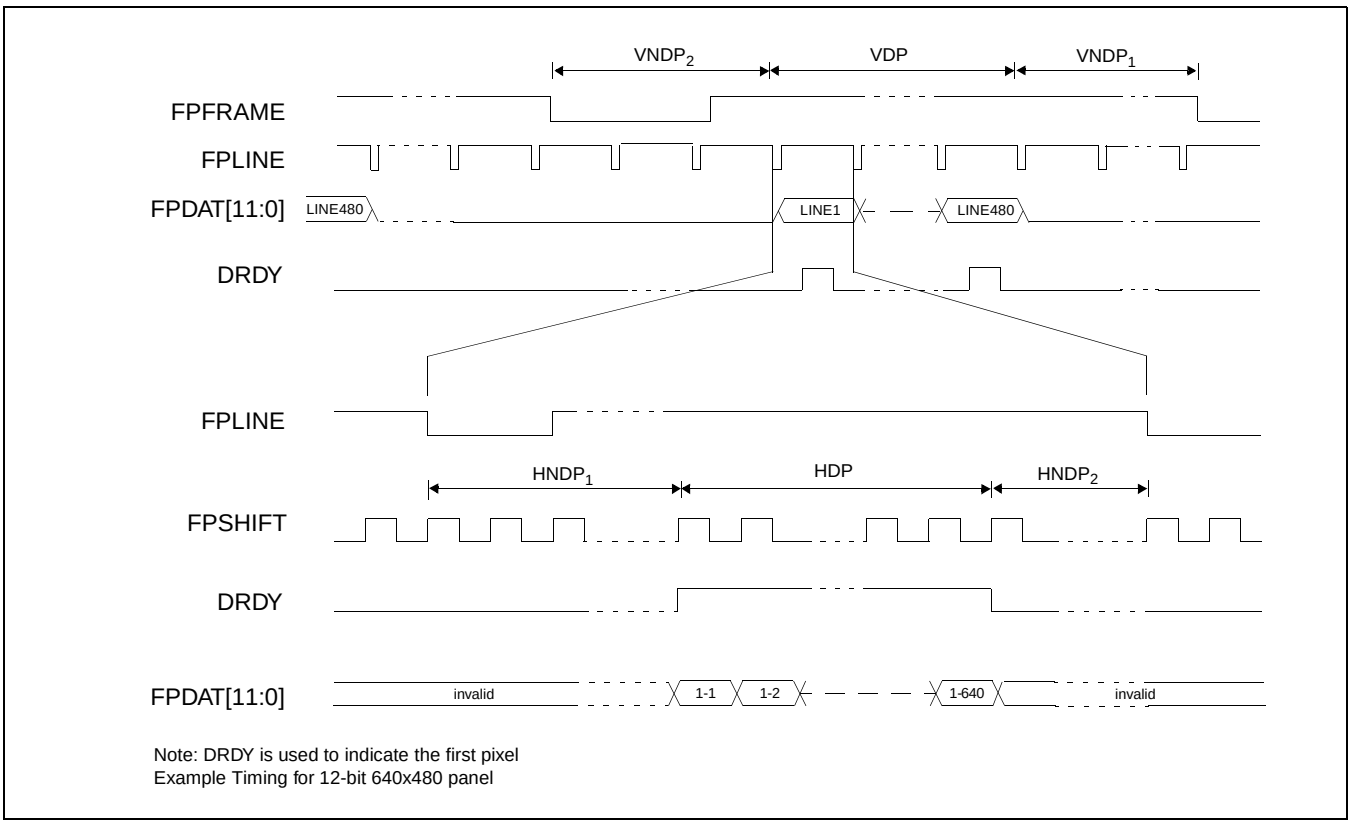


Figure 6-29: 12-Bit TFT Panel Timing

VDP = Vertical Display Period
= VDP Lines
VNDP = Vertical Non-Display Period
= VNDP1 + VNDP2
= VT - VDP Lines
VNDP1 = Vertical Non-Display Period 1
= VNDP - VNDP2 Lines
VNDP2 = Vertical Non-Display Period 2
= VDPS - VPS Lines if negative add VT
HDP = Horizontal Display Period
= HDP Ts
HNDP = Horizontal Non-Display Period
= HNDP1 + HNDP2
= HT - HDP Ts
HNDP1 = Horizontal Non-Display Period 1
= HDPS - (HPS + 1) + 5 Ts if negative add HT
HNDP2 = Horizontal Non-Display Period 2
= (HPS + 1) - (HDP + HDPS + 5) Ts if negative add HT

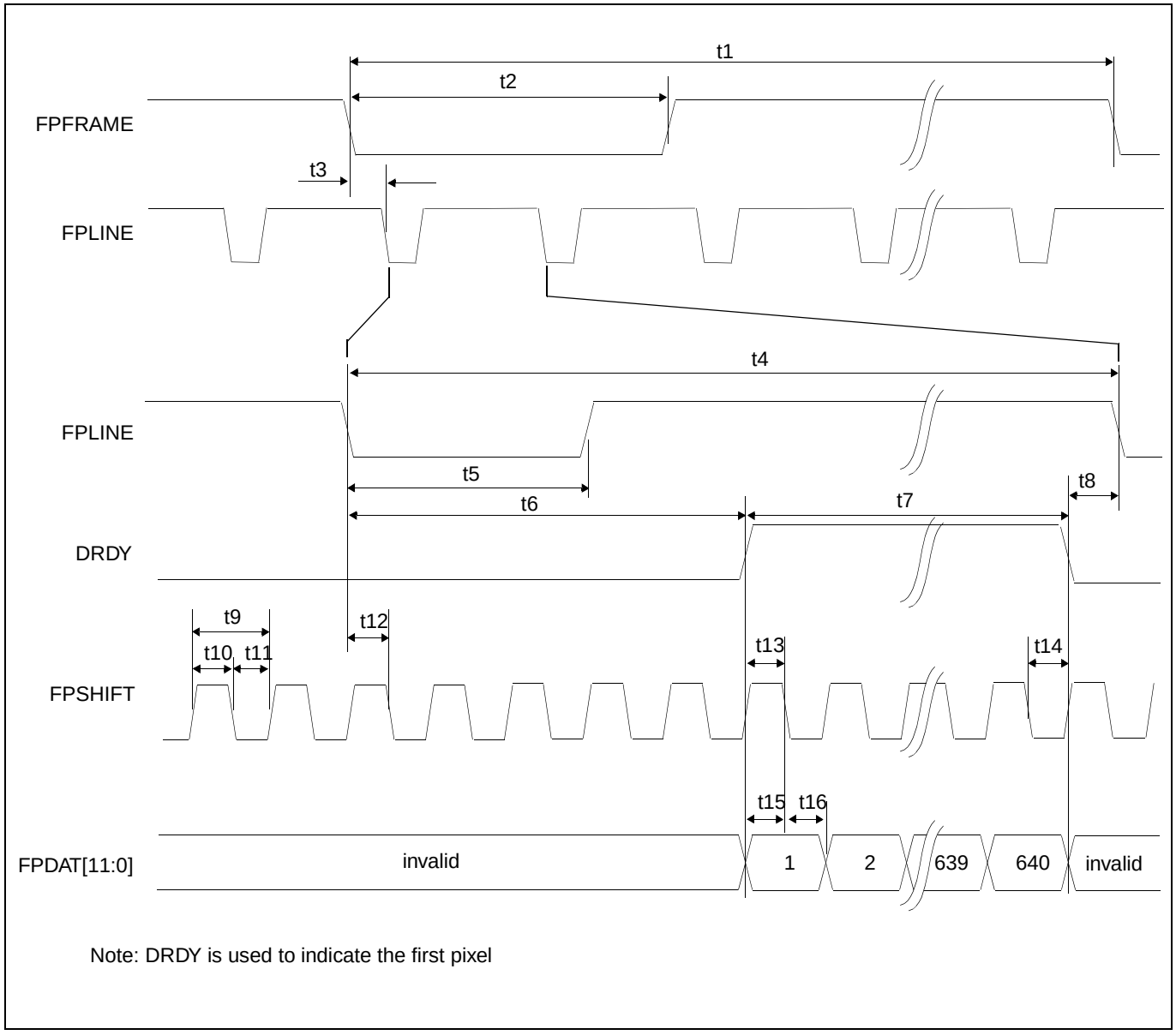


Figure 6-30: TFT A.C. Timing

Table 6-24: TFT A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME cycle time	VT			Lines
t2	FPFRAME pulse width low	VPW			Lines
t3	FPFRAME falling edge to FPLINE falling edge phase difference	HPS + 1			Ts (note 1)
t4	FPLINE cycle time	HT			Ts
t5	FPLINE pulse width low	HPW			Ts
t6	FPLINE Falling edge to DRDY active	note 2		250	Ts
t7	DRDY pulse width	HDP			Ts
t8	DRDY falling edge to FPLINE falling edge	note 3			Ts
t9	FPSHIFT period	1			Ts
t10	FPSHIFT pulse width high	0.5			Ts
t11	FPSHIFT pulse width low	0.5			Ts
t12	FPLINE setup to FPSHIFT falling edge	0.5			Ts
t13	DRDY to FPSHIFT falling edge setup time	0.5			Ts
t14	DRDY hold from FPSHIFT falling edge	0.5			Ts
t15	Data setup to FPSHIFT falling edge	0.5			Ts
t16	Data hold from FPSHIFT falling edge	0.5			Ts

1. Ts = pixel clock period
2. t6min = HDPS - (HPS + 1) + 5 if negative add HT
3. t8min = (HPS + 1) - (HDP + HDPS + 5) if negative add HT

6.4.10 160x160 Sharp HR-TFT Panel Timing (e.g. LQ031B1DDxx)

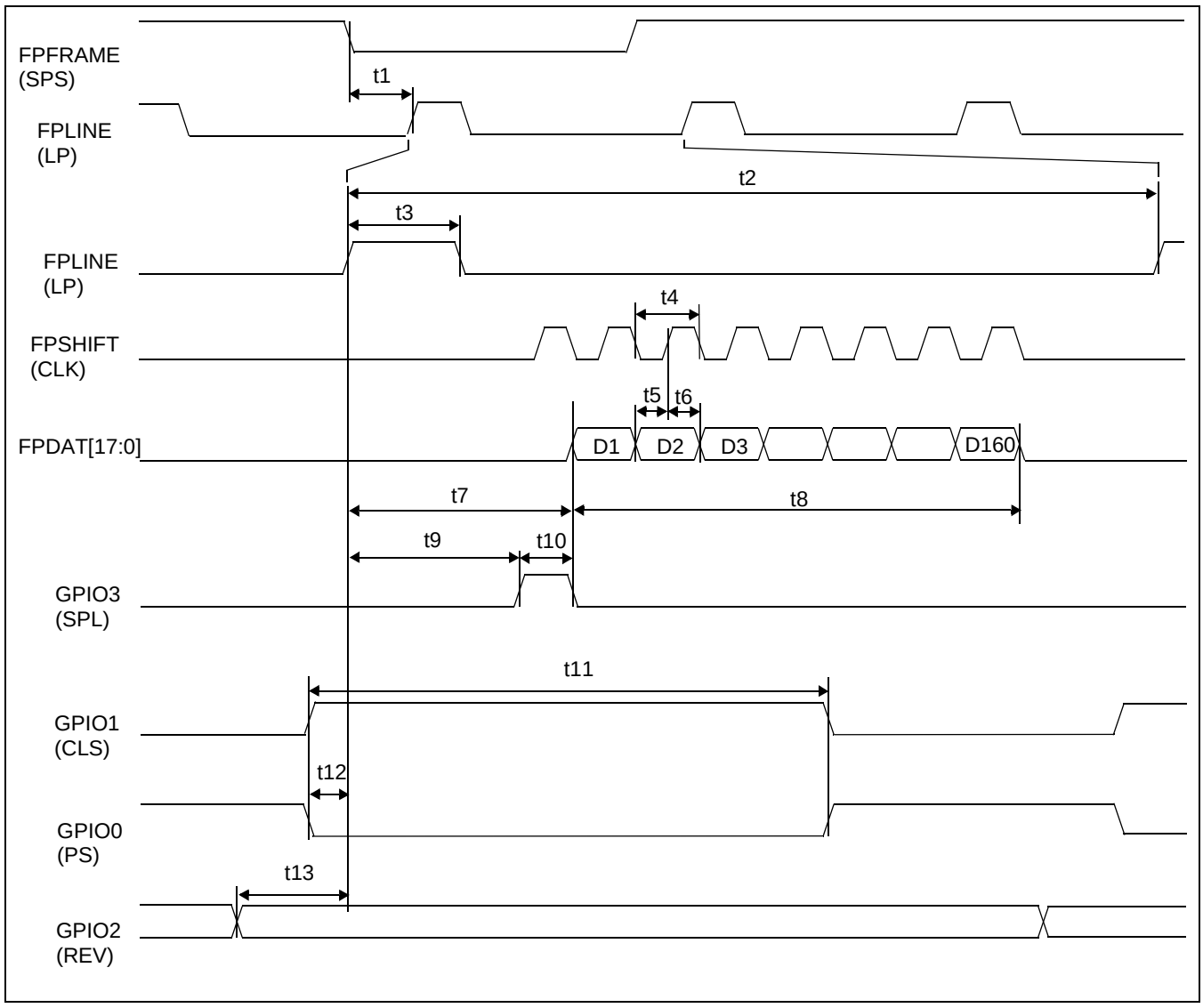


Figure 6-31: 160x160 Sharp HR-TFT Panel Horizontal Timing

Table 6-25: 160x160 Sharp HR-TFT Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPLINE start position		13		Ts (note 1)
t2	Horizontal total period	180		220	Ts
t3	FPLINE width		2		Ts
t4	FPSHIFT period		1		Ts
t5	Data setup to FPSHIFT rising edge	0.5			Ts
t6	Data hold from FPSHIFT rising edge	0.5			Ts
t7	Horizontal display start position		5		Ts
t8	Horizontal display period		160		Ts
t9	FPLINE rising edge to GPIO3 rising edge		4		Ts
t10	GPIO3 pulse width		1		Ts
t11	GPIO1(GPIO0) pulse width		136		Ts
t12	GPIO1 rising edge (GPIO0 falling edge) to FPLINE rise edge		4		Ts
t13	GPIO2 toggle edge to FPLINE rise edge		10		Ts

1. Ts = pixel clock period
2. t1typ = (REG[22h] bits 7-0) + 1
3. t2typ = ((REG[12h] bits 6-0) + 1) x 8
4. t3typ = (REG[20h] bits 6-0) + 1
5. t5typ = ((REG[16h] bits 7-0) + 1) - ((REG[22h] bits 7-0) + 1)
6. t6typ = ((REG[14h] bits 6-0) + 1) x 8

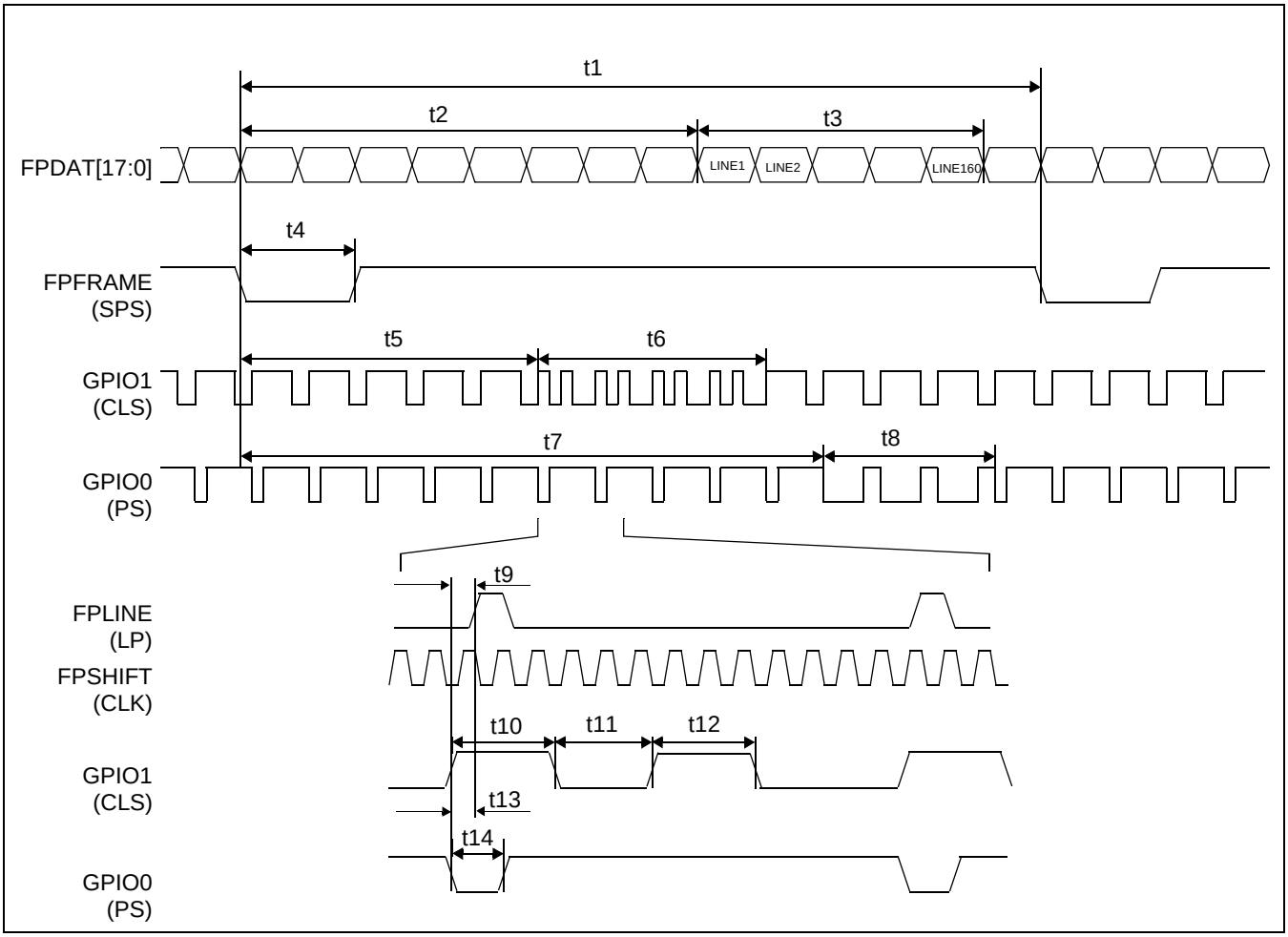


Figure 6-32: 160x160 Sharp HR-TFT Panel Vertical Timing

Table 6-26: 160x160 Sharp HR-TFT Panel Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Vertical total period	203		264	Lines
t2	Vertical display start position		40		Lines
t3	Vertical display period		160		Lines
t4	Vertical sync pulse width		2		Lines
t5	FPFRAME falling edge to GPIO1 alternate timing start		5		Lines
t6	GPIO1 alternate timing period		4		Lines
t7	FPFRAME falling edge to GPIO0 alternate timing start		40		Lines
t8	GPIO0 alternate timing period		162		Lines
t9	GPIO1 first pulse rising edge to FPLINE rising edge		4		Ts (note 1)
t10	GPIO1 first pulse width		48		Ts
t11	GPIO1 first pulse falling edge to second pulse rising edge		40		Ts
t12	GPIO1 second pulse width		48		Ts
t13	GPIO0 falling edge to FPLINE rising edge		4		Ts
t14	GPIO0 low pulse width		24		Ts

1. Ts = pixel clock period

6.4.11 320x240 Sharp HR-TFT Panel Timing (e.g. LQ039Q2DS01)

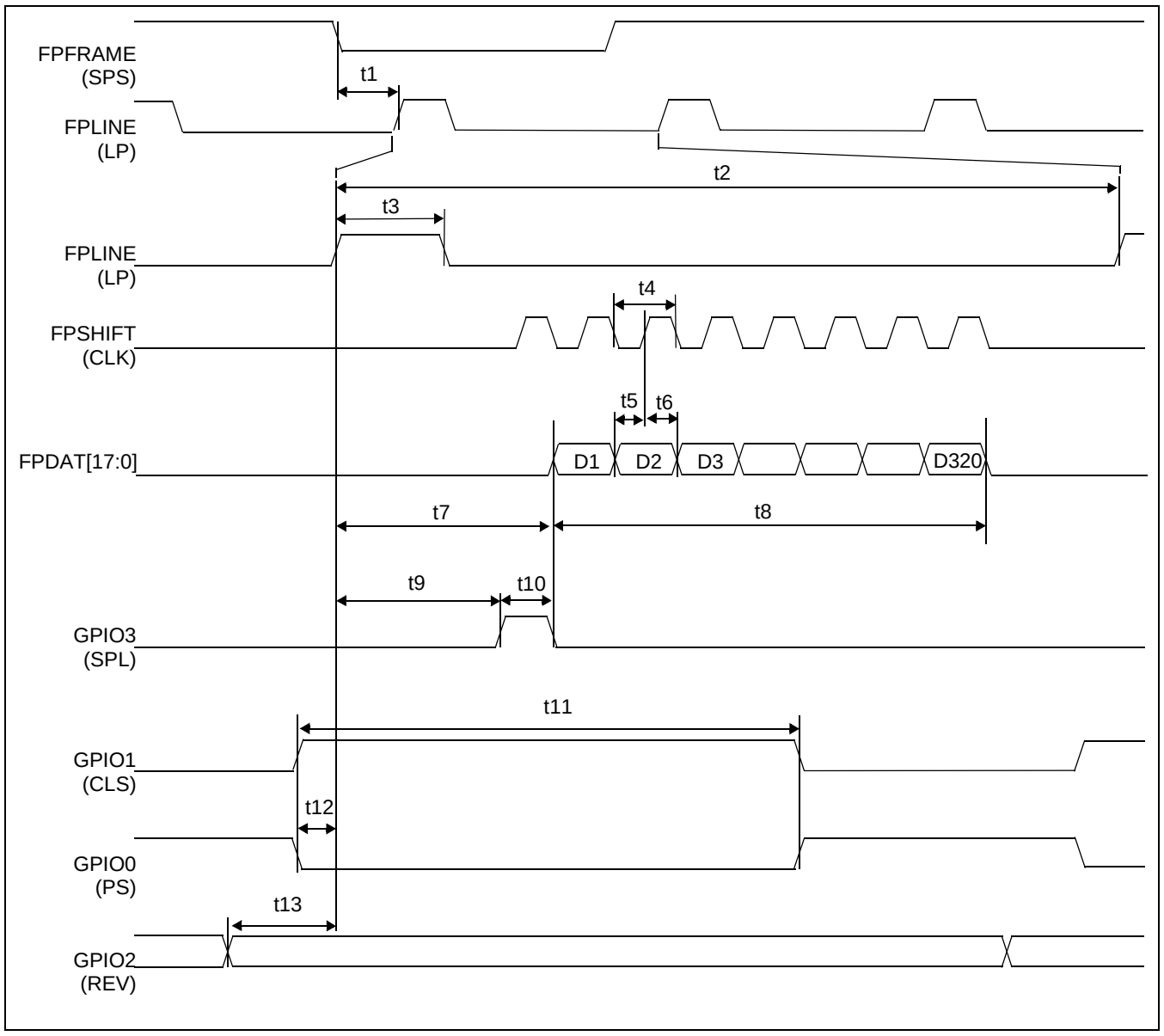


Figure 6-33: 320x240 Sharp 'Direct' HR-TFT Panel Horizontal Timing

Table 6-27: 320x240 Sharp 'Direct' HR-TFT Panel Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPLINE start position		14		Ts (note 1)
t2	Horizontal total period	400		440	Ts
t3	FPLINE width		1		Ts
t4	FPSHIFT period		1		Ts
t5	Data setup to FPSHIFT rising edge	0.5			Ts
t6	Data hold from FPSHIFT rising edge	0.5			Ts
t7	Horizontal display start position		60		Ts
t8	Horizontal display period		320		Ts
t9	FPLINE rising edge to GPIO3 rising edge		59		Ts
t10	GPIO3 pulse width		1		Ts
t11	GPIO1(GPIO0) pulse width		353		Ts
t12	GPIO1 rising edge (GPIO0 falling edge) to FPLINE rise edge		5		Ts
t13	GPIO2 toggle edge to FPLINE rise edge		11		Ts

1. Ts = pixel clock period
2. t1typ = (REG[22h] bits 7-0) + 1
3. t2typ = ((REG[12h] bits 6-0) + 1) x 8
4. t3typ = (REG[20h] bits 6-0) + 1
5. t5typ = ((REG[16h] bits 7-0) + 1) - ((REG[22h] bits 7-0) + 1)
6. t6typ = ((REG[14h] bits 6-0) + 1) x 8

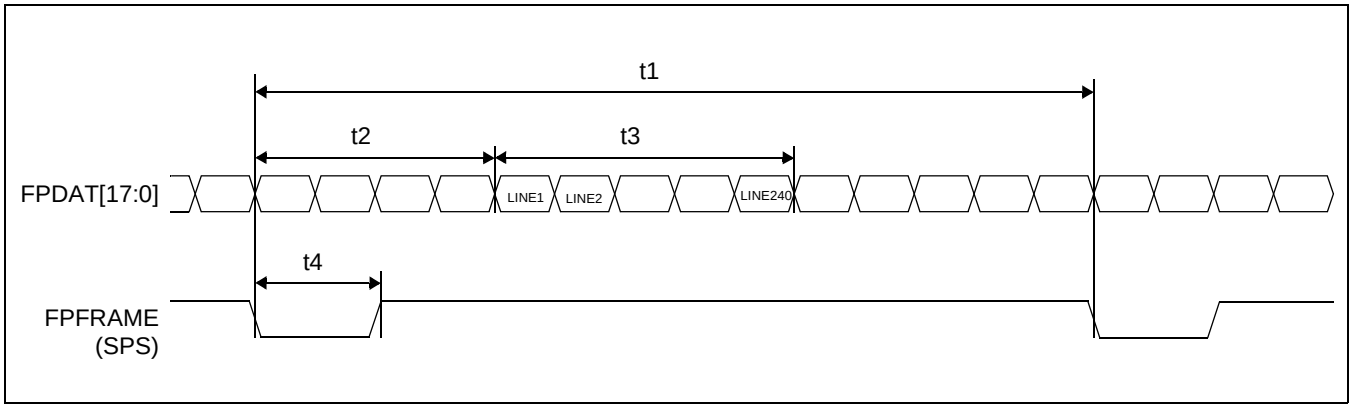


Figure 6-34: 320x240 Sharp HR-TFT Panel Vertical Timing

Table 6-28: 320x240 Sharp HR-TFT Panel Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Vertical total period	245		330	Lines
t2	Vertical display start position		4		Lines
t3	Vertical display period		240		Lines
t4	Vertical sync pulse width		2		Lines

6.4.12 160x240 Epson D-TFD Panel Timing (e.g. LF26SCR)

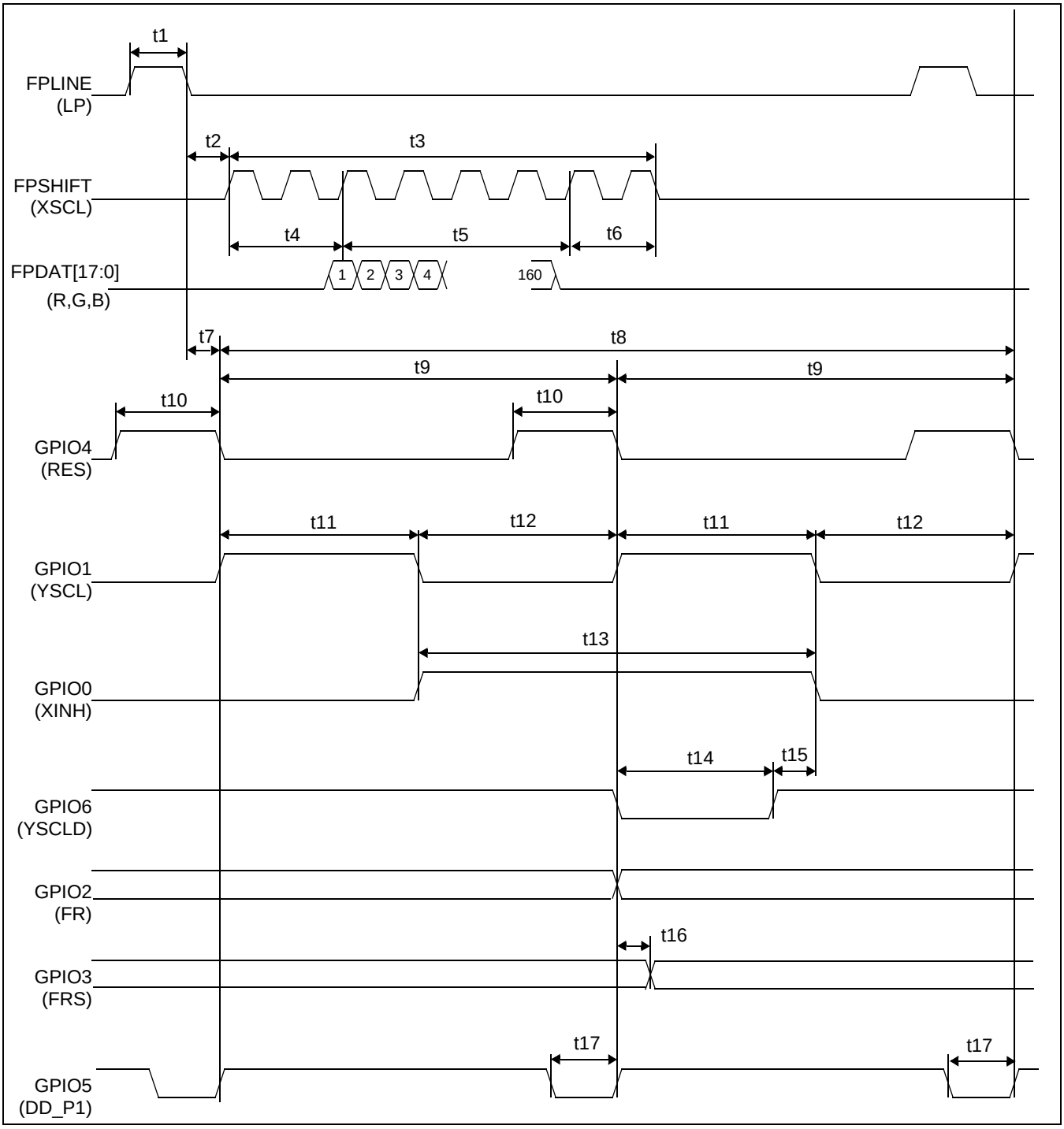


Figure 6-35: 160x240 Epson D-TFD Panel Horizontal Timing

Table 6-29: 160x240 Epson D-TFD Panel Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPLINE pulse width		9		Ts (note 1)
t2	FPLINE falling edge to FPSHIFT start position		8.5		Ts
t3	FPSHIFT active period		167		Ts
t4	FPSHIFT start to first data		4		Ts
t5	Horizontal display period		160		Ts
t6	Last data to FPSHIFT inactive		3		Ts
t7	FPLINE falling edge to GPIO4 first pulse falling edge		1		Ts
t8	Horizontal total period		400		Ts
t9	GPIO4 first pulse falling edge to second pulse falling edge		200		Ts
t10	GPIO4 pulse width		11		Ts
t11	GPIO1 pulse width		100		Ts
t12	GPIO1 low period		100)		Ts
t13	GPIO0 pulse width		200		Ts
t14	GPIO6 low pulse width		90		Ts
t15	GPIO6 rising edge to GPIO0 falling edge		10		Ts
t16	GPIO2 toggle to GPIO3 toggle		1		Ts
t17	GPIO5 low pulse width		7		Ts

1. Ts = pixel clock period

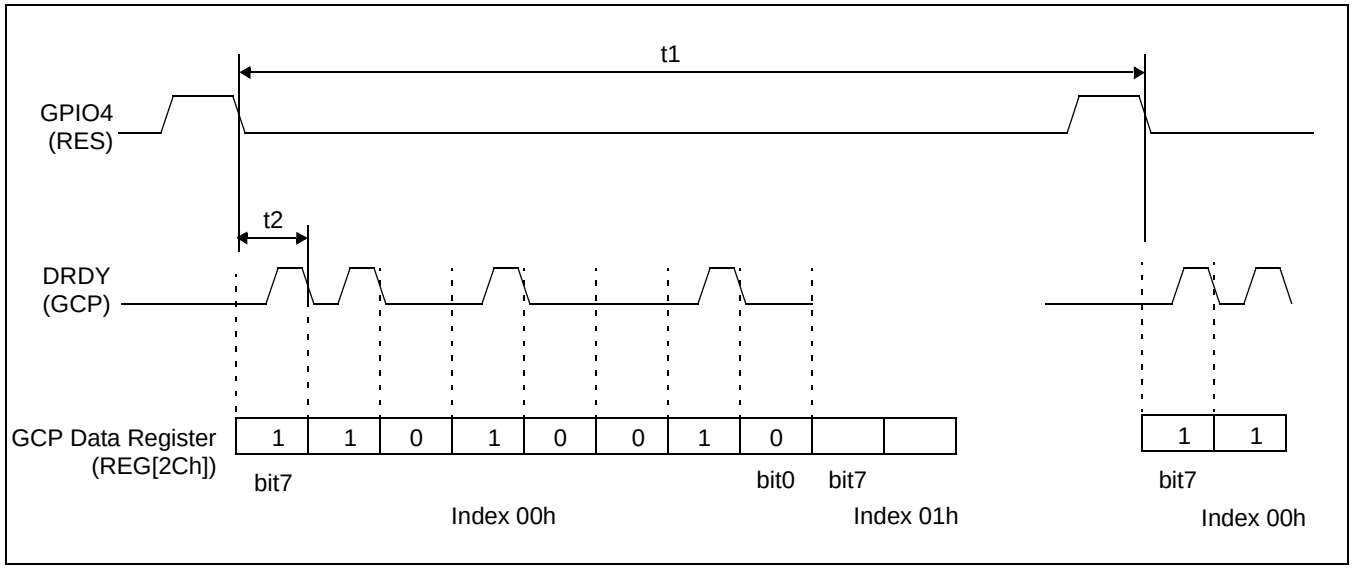


Figure 6-36: 160x240 Epson D-TFD Panel GCP Horizontal Timing

Table 6-30: 160x240 Epson D-TFD Panel GCP Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Half of the horizontal total period		200		Ts (note 1)
t2	GCP clock period		1		Ts

1. Ts = pixel clock period

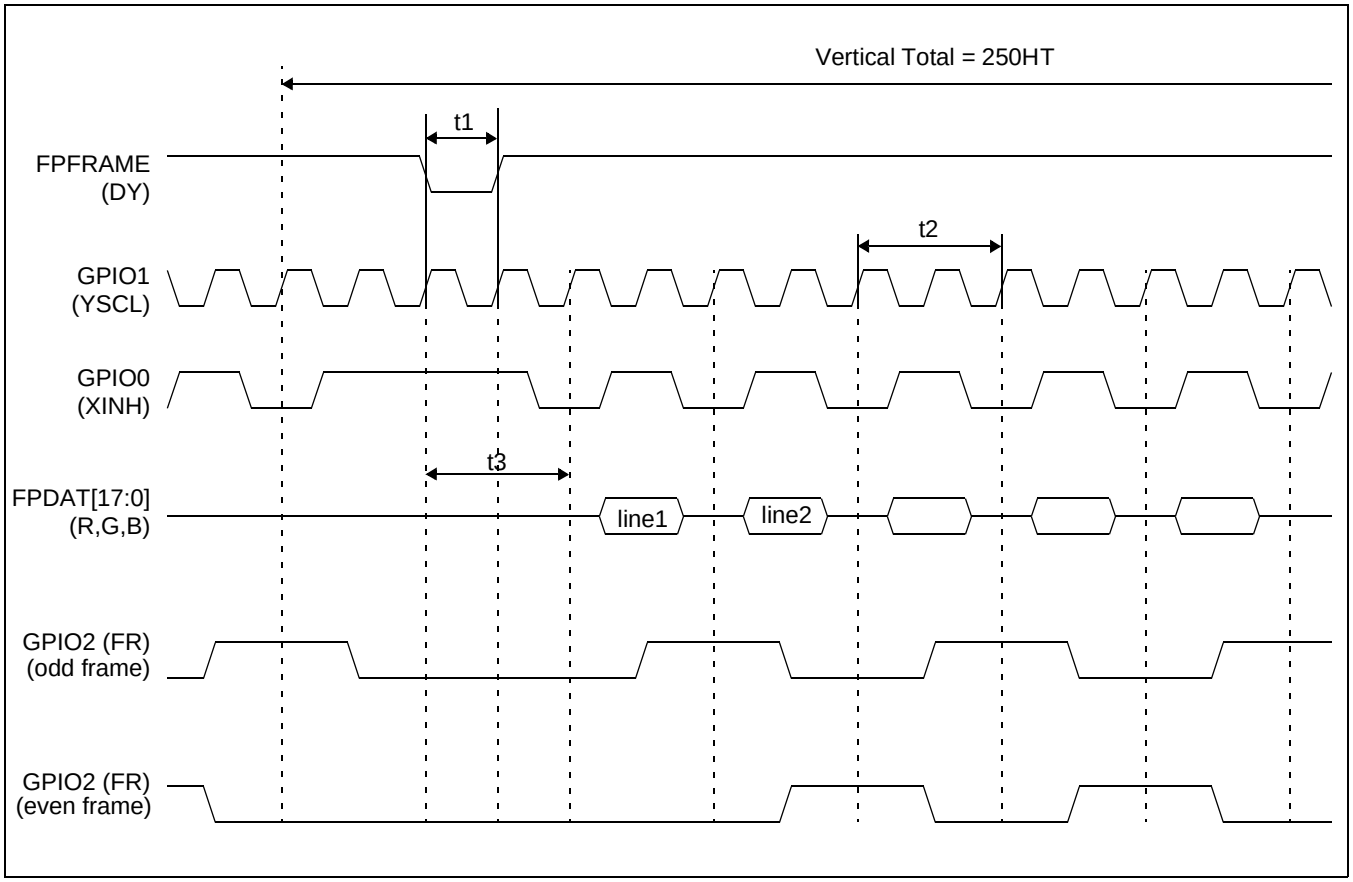


Figure 6-37: 160x240 Epson D-TFD Panel Vertical Timing

Table 6-31: 160x240 Epson D-TFD Panel Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME pulse width		200		Ts (note 1)
t2	Horizontal total period		400		Ts
t3	Vertical display start		400		Ts

1. Ts = pixel clock period

6.4.13 320x240 Epson D-TFD Panel Timing (e.g. LF37SQR)

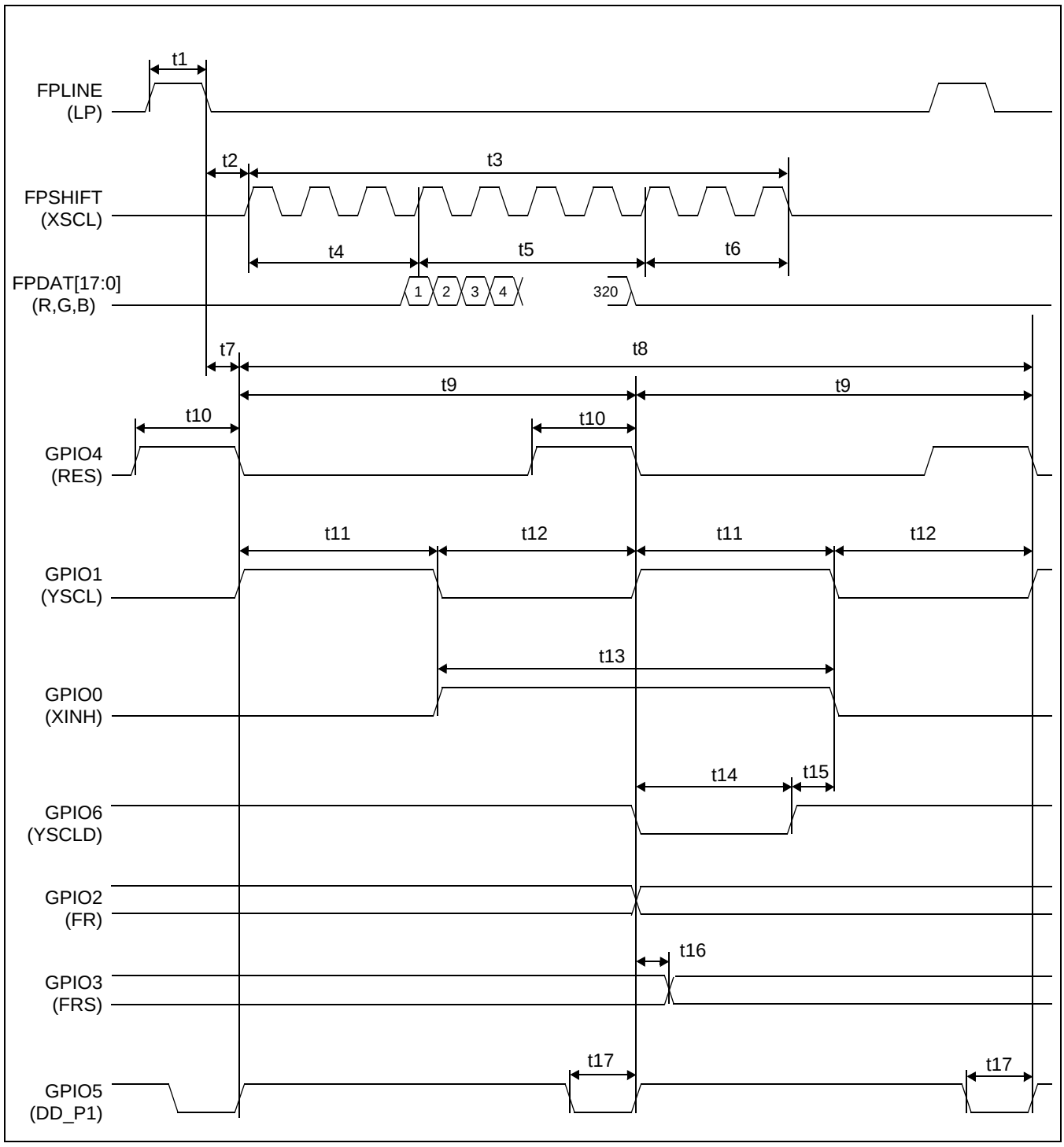


Figure 6-38: 320x240 Epson D-TFD Panel Horizontal Timing

Table 6-32: 320x240 Epson D-TFD Panel Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPLINE pulse width		9		Ts (note 1)
t2	FPLINE falling edge to FPSHIFT start position		8.5		Ts
t3	FPSHIFT active period		331		Ts
t4	FPSHIFT start to first data		6		Ts
t5	Horizontal display period		320		Ts
t6	Last data to FPSHIFT inactive		5		Ts
t7	FPLINE falling edge to GPIO4 first pulse falling edge		1		Ts
t8	Horizontal total period		400		Ts
t9	GPIO4 first pulse falling edge to second pulse falling edge		200		Ts
t10	GPIO4 pulse width		11		Ts
t11	GPIO1 pulse width		100		Ts
t12	GPIO1 low period		100		Ts
t13	GPIO0 pulse width		200		Ts
t14	GPIO6 low pulse width		90		Ts
t15	GPIO6 rising edge to GPIO0 falling edge		10		Ts
t16	GPIO2 toggle to GPIO3 toggle		1		Ts
t17	GPIO5 low pulse width		7		Ts

1. Ts = pixel clock period

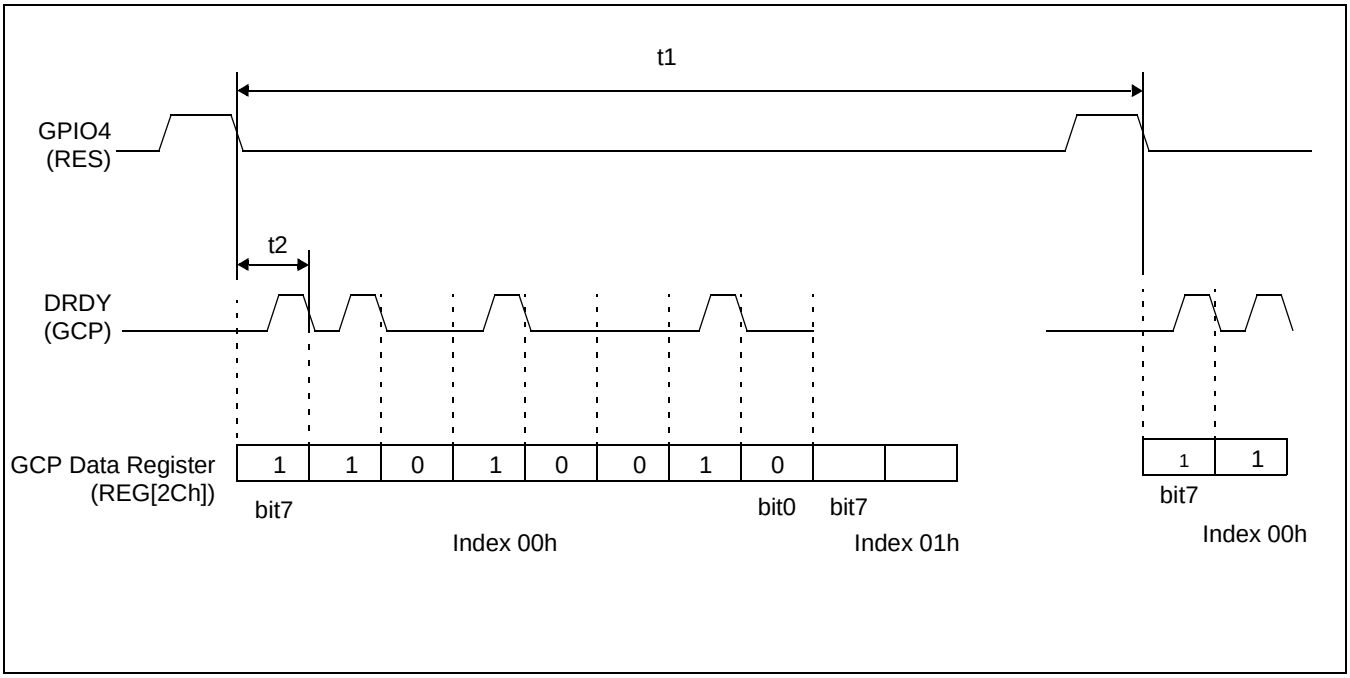


Figure 6-39: 320x240 Epson D-TFD Panel GCP Horizontal Timing

Table 6-33: 320x240 Epson D-TFD Panel GCP Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Half of the horizontal total period		200		Ts (note 1)
t2	GCP clock period		1		Ts

1. Ts = pixel clock period

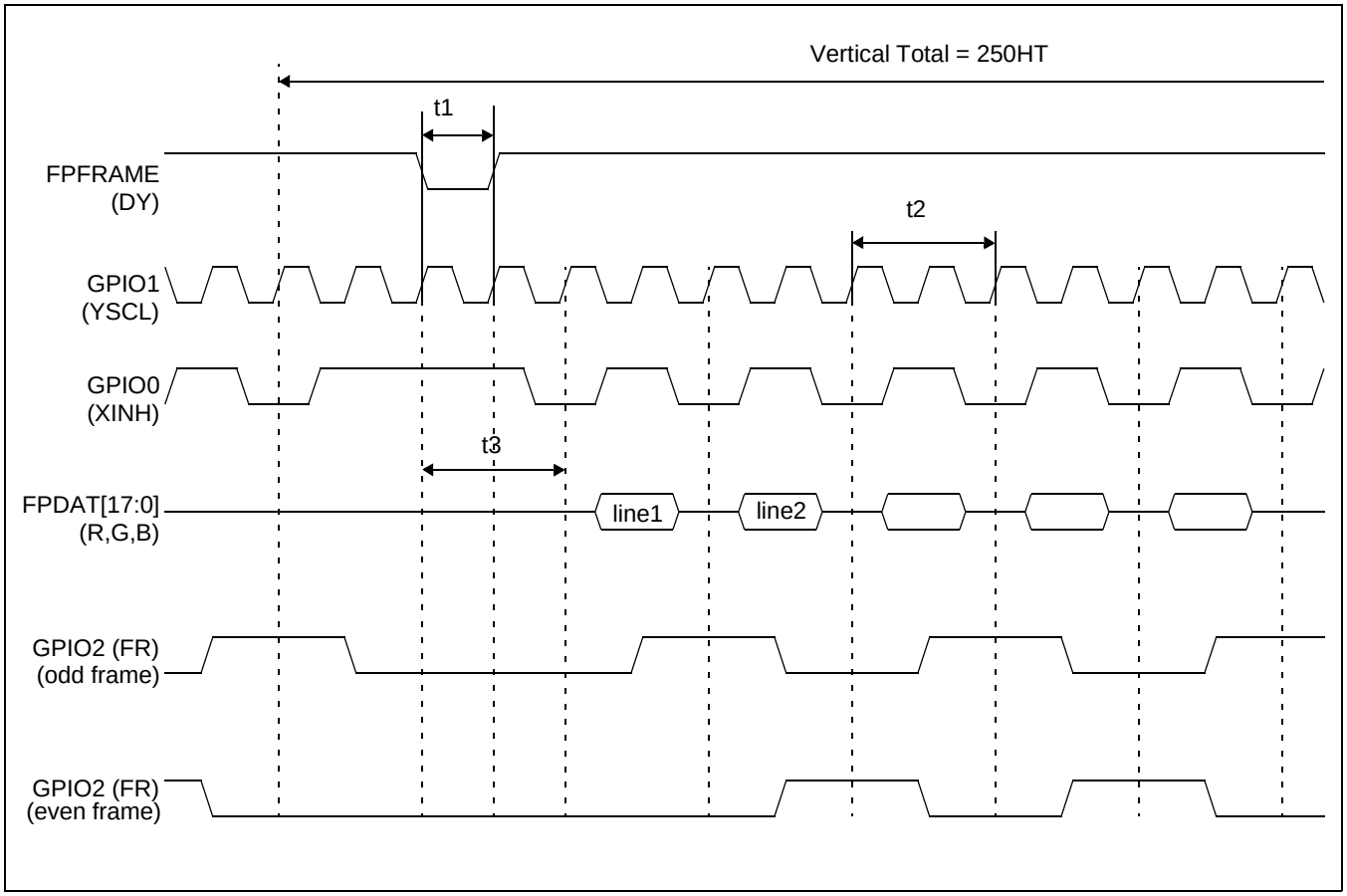


Figure 6-40: 320x240 Epson D-TFD Panel Vertical Timing

Table 6-34: 320x240 Epson D-TFD Panel Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME pulse width		200		Ts (note 1)
t2	Horizontal total period		400		Ts
t3	Vertical display start		400		Ts

1. Ts = pixel clock period

7 Clocks

7.1 Clock Descriptions

7.1.1 BCLK

BCLK is an internal clock derived from CLKI. BCLK can be a divided version (÷1, ÷2, ÷3, ÷4) of CLKI. CLKI is typically derived from the host CPU bus clock.

The source clock options for BCLK may be selected as in the following table.

Table 7-1: BCLK Clock Selection

Source Clock Options	BCLK Selection
CLKI	CNF[7:6] = 00
CLKI ÷2	CNF[7:6] = 01
CLKI ÷3	CNF[7:6] = 10
CLKI ÷4	CNF[7:6] = 11

Note
For synchronous bus interfaces, it is recommended that BCLK be set the same as the CPU bus clock (not a divided version of CLKI) e.g. SH-3, SH-4.

Note
The CLKI ÷ 3 and CLKI ÷ 4 options may not work properly with bus interfaces with short back-to-back cycle timing.

7.1.2 MCLK

MCLK provides the internal clock required to access the embedded SRAM. The S1D13706 is designed with efficient power saving control for clocks (clocks are turned off when not used); reducing the frequency of MCLK does not necessarily save more power. Furthermore, reducing the MCLK frequency relative to the BCLK frequency increases the CPU cycle latency and so reduces screen update performance. For a balance of power saving and performance, the MCLK should be configured to have a high enough frequency setting to provide sufficient screen refresh as well as acceptable CPU cycle latency.

The source clock options for MCLK may be selected as in the following table.

Table 7-2: MCLK Clock Selection

Source Clock Options	MCLK Selection
BCLK	REG[04h] bit 5,4 = 00
BCLK ÷2	REG[04h] bit 5,4 = 01
BCLK ÷3	REG[04h] bit 5,4 = 10
BCLK ÷4	REG[04h] bit 5,4 = 11

7.1.3 PCLK

PCLK is the internal clock used to control the LCD panel. PCLK should be chosen to match the optimum frame rate of the LCD panel. See Section 9, “Frame Rate Calculation” on page 128 for details on the relationship between PCLK and frame rate.

Some flexibility is possible in the selection of PCLK. Firstly, LCD panels typically have a range of permissible frame rates. Secondly, it may be possible to choose a higher PCLK frequency and tailor the horizontal and vertical non-display periods to lower the frame-rate to its optimal value.

The source clock options for PCLK may be selected as in the following table.

Table 7-3: PCLK Clock Selection

Source Clock Options	PCLK Selection
MCLK	REG[05h] = 00h
MCLK ÷2	REG[05h] = 10h
MCLK ÷3	REG[05h] = 20h
MCLK ÷4	REG[05h] = 30h
MCLK ÷8	REG[05h] = 40h
BCLK	REG[05h] = 01h
BCLK ÷2	REG[05h] = 11h
BCLK ÷3	REG[05h] = 21h
BCLK ÷4	REG[05h] = 31h
BCLK ÷8	REG[05h] = 41h
CLKI	REG[05h] = 02h
CLKI ÷2	REG[05h] = 12h
CLKI ÷3	REG[05h] = 22h
CLKI ÷4	REG[05h] = 32h
CLKI ÷8	REG[05h] = 42h
CLKI2	REG[05h] = 03h
CLKI2 ÷2	REG[05h] = 13h
CLKI2 ÷3	REG[05h] = 23h
CLKI2 ÷4	REG[05h] = 33h
CLKI2 ÷8	REG[05h] = 43h

There is a relationship between the frequency of MCLK and PCLK that must be maintained.

Table 7-4: Relationship between MCLK and PCLK

SwivelView Orientation	Color Depth (bpp)	MCLK to PCLK Relationship
SwivelView 0° and 180°	16	$f_{MCLK} \geq f_{PCLK}$
	8	$f_{MCLK} \geq f_{PCLK} \div 2$
	4	$f_{MCLK} \geq f_{PCLK} \div 4$
	2	$f_{MCLK} \geq f_{PCLK} \div 8$
	1	$f_{MCLK} \geq f_{PCLK} \div 16$
SwivelView 90° and 270°	16/8/4/2/1	$f_{MCLK} \geq 1.25 f_{PCLK}$

7.1.4 PWMCLK

PWMCLK is the internal clock used by the Pulse Width Modulator for output to the panel.

The source clock options for PWMCLK may be selected as in the following table.

Table 7-5: PWMCLK Clock Selection

Source Clock Options	PWMCLK Selection
CLKI	REG[B1h] bit 0 = 0
CLKI2	REG[B1h] bit 0 = 1

For further information on controlling PWMCLK, see Section 8.3.9, “Pulse Width Modulation (PWM) Clock and Contrast Voltage (CV) Pulse Configuration Registers” on page 124.

Note

The S1D13706 provides Pulse Width Modulation output on the pin PWMOUT. PWMOUT can be used to control LCD panels which support PWM control of the back-light inverter.

7.2 Clock Selection

The following diagram provides a logical representation of the S1D13706 internal clocks.

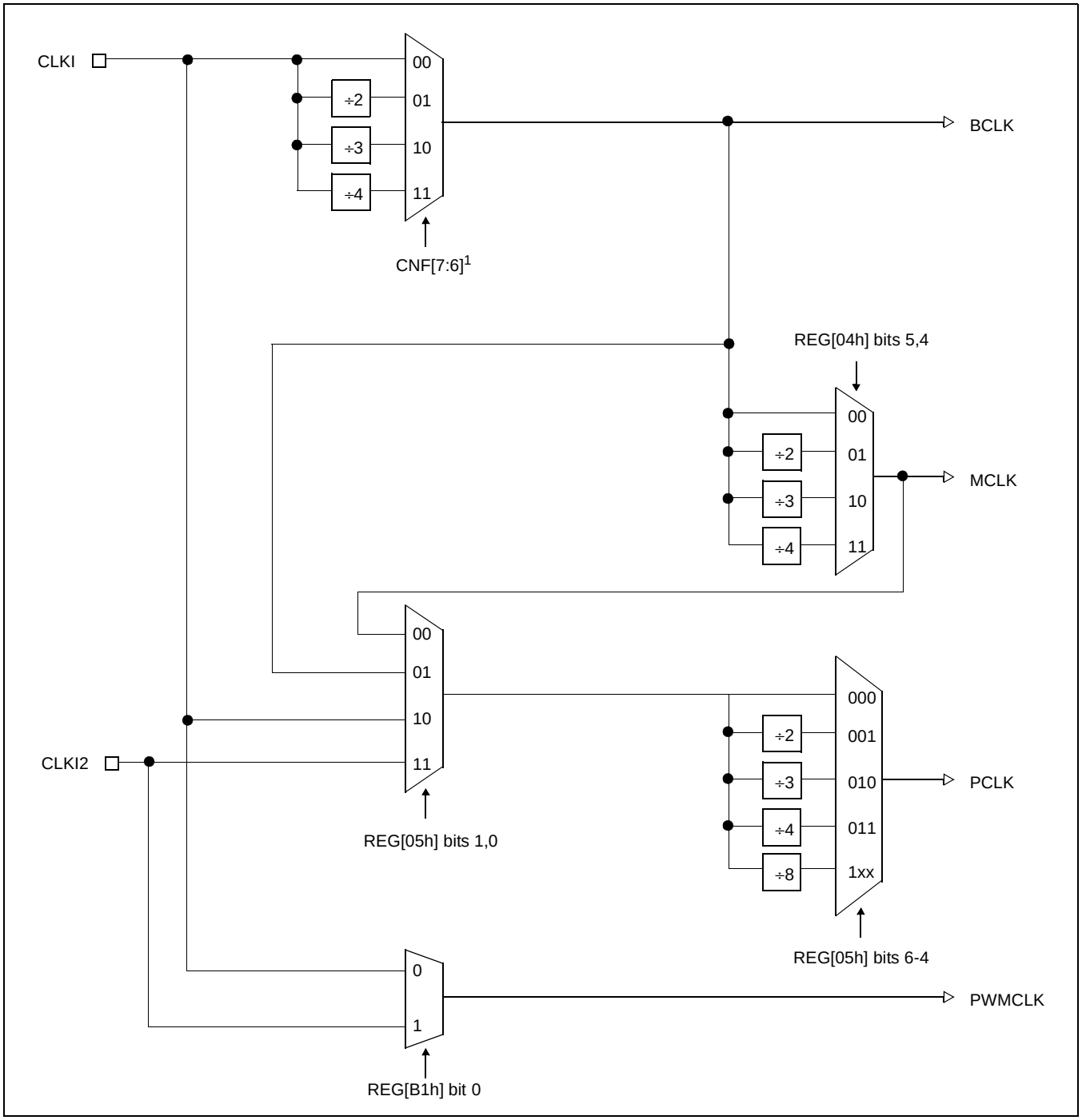


Figure 7-1: Clock Selection

Note

[1] CNF[7:6] must be set at RESET#.

7.3 Clocks versus Functions

Table 7-6: “S1D13706 Internal Clock Requirements”, lists the internal clocks required for the following SED1376 functions.

Table 7-6: S1D13706 Internal Clock Requirements

Function	Bus Clock (BCLK)	Memory Clock (MCLK)	Pixel Clock (PCLK)	PWM Clock (PWMCLK)
Register Read/Write	Required	Not Required	Not Required	Not Required[1]
Memory Read/Write	Required	Required	Not Required	Not Required[1]
Look-Up Table Register Read/Write	Required	Required	Not Required	Not Required[1]
Software Power Save	Required	Not Required	Not Required	Not Required[1]
LCD Output	Required	Required	Required	Not Required[1]

Note
[1]PWMCLK is an optional clock (see Section 7.1.4, “PWMCLK” on page 90).

8 Registers

This section discusses how and where to access the S1D13706 registers. It also provides detailed information about the layout and usage of each register.

8.1 Register Mapping

The S1D13706 registers are memory-mapped. When the system decodes the input pins as CS# = 0 and M/R# = 0, the registers may be accessed. The register space is decoded by A[16:0].

8.2 Register Set

The S1D13706 register set is as follows.

Table 8-1: S1D13706 Register Set

Table 8-1: S1D13706 Register Set

Register	Pg	Register	Pg
Picture-in-Picture Plus (PIP+) Registers			
REG[7Ch] PIP+ Window Display Start Address Register 0	113	REG[7Dh] PIP+ Window Display Start Address Register 1	113
REG[7Eh] PIP+ Window Display Start Address Register 2	113	REG[80h] PIP+ Window Line Address Offset Register 0	113
REG[81h] PIP+ Window Line Address Offset Register 1	113	REG[84h] PIP+ Window X Start Position Register 0	114
REG[85h] PIP+ Window X Start Position Register 1	114	REG[88h] PIP+ Window Y Start Position Register 0	115
REG[89h] PIP+ Window Y Start Position Register 1	115	REG[8Ch] PIP+ Window X End Position Register 0	116
REG[8Dh] PIP+ Window X End Position Register 1	116	REG[90h] PIP+ Window Y End Position Register 0	117
REG[91h] PIP+ Window Y End Position Register 1	117		
Miscellaneous Registers			
REG[A0h] Power Save Configuration Register	118	REG[A1h] Reserved	118
REG[A2h] Software Reset Register	118	REG[A3h] Reserved	119
REG[A4h] Scratch Pad Register 0	119	REG[A5h] Scratch Pad Register 1	119
General Purpose IO Pins Registers			
REG[A8h] General Purpose IO Pins Configuration Register 0	119	REG[A9h] General Purpose IO Pins Configuration Register 1	120
REG[ACh] General Purpose IO Pins Status/Control Register 0	121	REG[ADh] General Purpose IO Pins Status/Control Register 1	123
PWM Clock and CV Pulse Configuration Registers			
REG[B0h] PWM Clock / CV Pulse Control Register	124	REG[B1h] PWM Clock / CV Pulse Configuration Register	126
REG[B2h] CV Pulse Burst Length Register	127	REG[B3h] PWMOUT Duty Cycle Register	127

8.3 Register Descriptions

Unless specified otherwise, all register bits are set to 0 during power-on or software reset (REG[A2h] bit 0).

8.3.1 Read-Only Configuration Registers

Revision Code Register
REG[00h] Read Only

7	6	5	4	3	2	1	0
Product Code Bits 5-0						Revision Code Bits 1-0	

Note
The S1D13706 returns a value of 28h.

bits 7-2 Product Code
These are read-only bits that indicates the product code. The product code is 001010.

bits 1-0 Revision Code
These are read-only bits that indicates the revision code. The revision code is 00.

Display Buffer Size Register
REG[01h] Read Only

Display Buffer Size Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 Display Buffer Size Bits [7:0]
This is a read-only register that indicates the size of the SRAM display buffer measured in 4K byte increments. The S1D13706 display buffer is 80K bytes and therefore this register returns a value of 20 (14h).

Value of this register = display buffer size ÷ 4K bytes
= 80K bytes ÷ 4K bytes
= 20 (14h)

Configuration Readback Register
REG[02h] Read Only

CNF7 Status	CNF6 Status	CNF5 Status	CNF4 Status	CNF3 Status	CNF2 Status	CNF1 Status	CNF0 Status
7	6	5	4	3	2	1	0

bits 7-0 CNF[7:0] Status
These read-only status bits return the status of the configuration pins CNF[7:0]. CNF[7:0] are latched at the rising edge of RESET# or when a 1 is written to the Software Reset bit (REG[A2h] bit 0).

8.3.2 Clock Configuration Registers

Memory Clock Configuration Register
REG[04h] Read/Write

n/a		MCLK Divide Select Bits 1-0		n/a			Reserved
7	6	5	4	3	2	1	0

bits 5-4 MCLK Divide Select Bits [1:0]
These bits determine the divide used to generate the Memory Clock (MCLK) from the Bus Clock (BCLK).

Table 8-2: MCLK Divide Selection

MCLK Divide Select Bits	BCLK to MCLK Frequency Ratio
00	1:1
01	2:1
10	3:1
11	4:1

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

Pixel Clock Configuration Register
REG[05h] Read/Write

7	6	5	4	3	2	1	0
n/a	PCLK Divide Select Bits 2-0			n/a		PCLK Source Select Bits 1-0	

bits 6-4 PCLK Divide Select Bits [1:0]
These bits determine the divide used to generate the Pixel Clock (PCLK) from the Pixel Clock Source.

Table 8-3: PCLK Divide Selection

PCLK Divide Select Bits	PCLK Source to PCLK Frequency Ratio
000	1:1
001	2:1
010	3:1
011	4:1
1XX	8:1

bits 1-0 PCLK Source Select Bits [1:0]
These bits determine the source of the Pixel Clock (PCLK).

Table 8-4: PCLK Source Selection

PCLK Source Select Bits	PCLK Source
00	MCLK
01	BCLK
10	CLKI
11	CLKI2

8.3.3 Look-Up Table Registers

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

7	6	5	4	3	2	1	0
LUT Blue Write Data Bits 5-0						n/a	

bits 7-2 LUT Blue Write Data Bits [5:0]
This register contains the data to be written to the blue component of the Look-Up Table. The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

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

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

LUT Green Write Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Green Write Data Bits [5:0]
This register contains the data to be written to the green component of the Look-Up Table. The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

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

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

LUT Red Write Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Red Write Data Bits [5:0]
This register contains the data to be written to the red component of the Look-Up Table. The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

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

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

LUT Write Address Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 LUT Write Address Bits [7:0]
This register forms a pointer into the Look-Up Table (LUT) which is used to write LUT data stored in REG[08h], REG[09h], and REG[0Ah]. **The data is updated to the LUT only with the completion of a write to this register**. This is a write-only register and returns 00h if read.

Note
The S1D13706 has three 256-position, 6-bit wide LUTs, one for each of red, green, and blue (see Section 11, “Look-Up Table Architecture” on page 130).

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

LUT Blue Read Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Blue Read Data Bits [5:0]
This register contains the data from the blue component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]). This is a read-only register.

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

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

LUT Green Read Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Green Read Data Bits [5:0]
This register contains the data from the green component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]). This is a read-only register.

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

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

LUT Red Read Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Red Read Data Bits [5:0]
This register contains the data from the red component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]). This is a read-only register.

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

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

LUT Read Address Bits 7-0							
7	6	5	4	3	2	1	0

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

Note
The S1D13706 has three 256-position, 6-bit wide LUTs, one for each of red, green, and blue (see Section 11, “Look-Up Table Architecture” on page 130).

8.3.4 Panel Configuration Registers

Panel Type Register
REG[10h] Read/Write

Panel Data Format Select	Color/Mono. Panel Select	Panel Data Width Bits 1-0		Active Panel Resolution Select	n/a	Panel Type Bits 1-0	
7	6	5	4	3	2	1	0

bit 7 Panel Data Format Select
When this bit = 0, 8-bit single color passive LCD panel data format 1 is selected. For AC timing see Section 6.4.5, “Single Color 8-Bit Panel Timing (Format 1)” on page 64.
When this bit = 1, 8-bit single color passive LCD panel data format 2 is selected. For AC timing see Section 6.4.6, “Single Color 8-Bit Panel Timing (Format 2)” on page 66.

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

bits 5-4 Panel Data Width Bits [1:0]
These bits select the data width size of the LCD panel.

Table 8-5: Panel Data Width Selection

Panel Data Width Bits [1:0]	Passive Panel Data Width Size	Active Panel Data Width Size
00	4-bit	9-bit
01	8-bit	12-bit
10	16-bit	18-bit
11	Reserved	Reserved

bit 3 Active Panel Resolution Select

This bit selects one of two panel resolutions when an HR-TFT or D-TFD panel is selected. This bit has no effect for other panel types.

Table 8-6: Active Panel Resolution Selection

Active Panel Resolution Select Bit	HR-TFT Resolution	D-TFD Resolution
0	160x160	160x240
1	320x240	320x240

Note

This bit sets some internal non-configurable timing values for the selected panel. However, all panel configuration registers (REG[12h] - REG[27h]) still require programming with the appropriate values for the selected panel. For panel AC timing, see Section 6.4, “Display Interface” on page 55.

bits 1-0 Panel Type Bits[1:0]

These bits select the panel type.

Table 8-7: LCD Panel Type Selection

REG[10h] Bits[1:0]	Panel Type
00	STN
01	TFT
10	HR-TFT
11	D-TFD

MOD Rate Register
REG[11h] Read/Write

7	6	5	4	3	2	1	0
n/a		MOD Rate Bits 5-0					

bits 5-0 MOD Rate Bits [5:0]

These bits are for passive LCD panels only.

When these bits are all 0, the MOD output signal (DRDY) toggles every FPFRAME. For a non-zero value *n*, the MOD output signal (DRDY) toggles every n FPLINE.

Horizontal Total Register
REG[12h] Read/Write

n/a	Horizontal Total Bits 6-0						
7	6	5	4	3	2	1	0

bits 6-0 Horizontal Total Bits [6:0]
These bits specify the LCD panel Horizontal Total period, in 8 pixel resolution. The Horizontal Total is the sum of the Horizontal Display period and the Horizontal Non-Display period. Since the maximum Horizontal Total is 1024 pixels, the maximum panel resolution supported is 800x600.

Horizontal Total in number of pixels = ((REG[12h] bits 6:0) + 1) × 8

Note
[1] This register must be programmed such that the following formulas are valid.
HDPS + HDP < HT
[2] For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 55.

Horizontal Display Period Register
REG[14h] Read/Write

n/a	Horizontal Display Period Bits 6-0						
7	6	5	4	3	2	1	0

bits 6-0 Horizontal Display Period Bits [6:0]
These bits specify the LCD panel Horizontal Display Period (HDP), in 8 pixel resolution. The Horizontal Display Period should be less than the Horizontal Total to allow for a sufficient Horizontal Non-Display Period.

Horizontal Display Period in number of pixels = ((REG[14h] bits 6:0) + 1) × 8

Note
For passive panels, HDP must be a minimum of 32 pixels and can be increased by multiples of 16. For TFT panels, HDP must be a minimum of 16 pixels and can be increased by multiples of 8.

For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 55.

Horizontal Display Period Start Position Register 0
REG[16h] Read/Write

7	6	5	4	3	2	1	0
Horizontal Display Period Start Position Bits 7-0							

Horizontal Display Period Start Position Register 1
REG[17h] Read/Write

7	6	5	4	3	2	1	0
n/a						Horizontal Display Period Start Position Bits 9-8	

bits 9-0 Horizontal Display Period Start Position Bits [9:0]
These bits specify a value used in the calculation of the Horizontal Display Period Start Position (in 1 pixel resolution) for TFT, HR-TFT and D-TFD panels. For passive LCD panels these bits must be set to 00h.

To calculate the Horizontal Display Period Start Position (HDPS) an offset which depends on the panel type is required. For further information on calculating the HDPS, see the specific panel AC Timing in Section 6.4, “Display Interface” on page 55.

Note
This register must be programmed such that the following formula is valid.
HDPS + HDP < HT

Vertical Total Register 0
REG[18h] Read/Write

7	6	5	4	3	2	1	0
Vertical Total Bits 7-0							

Vertical Total Register 1
REG[19h] Read/Write

7	6	5	4	3	2	1	0
n/a						Vertical Total Bits 9-8	

bits 9-0 Vertical Total Bits [9:0]
These bits specify the LCD panel Vertical Total period, in 1 line resolution. The Vertical Total is the sum of the Vertical Display Period and the Vertical Non-Display Period. The maximum Vertical Total is 1024 lines.

Vertical Total in number of lines = (REG[18h] bits 7:0, REG[19h] bits 1:0) + 1

Note
[1] This register must be programmed such that the following formula is valid.
VDPS + VDP < VT
[2] For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 55.

Vertical Display Period Register 0
REG[1Ch] Read/Write

7	6	5	4	3	2	1	0
Vertical Display Period Bits 7-0							

Vertical Display Period Register 1
REG[1Dh] Read/Write

7	6	5	4	3	2	1	0
n/a						Vertical Display Period Bits 9-8	

bits 9-0 Vertical Display Period Bits [9:0]
These bits specify the LCD panel Vertical Display period, in 1 line resolution. The Vertical Display period should be less than the Vertical Total to allow for a sufficient Vertical Non-Display period.

Vertical Display Period in number of lines = (REG[1Ch] bits 7:0, REG[1Dh] bits 1:0) + 1

Note
For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 55.

Vertical Display Period Start Position Register 0
REG[1Eh] Read/Write

7	6	5	4	3	2	1	0
Vertical Display Period Start Position Bits 7-0							

Vertical Display Period Start Position Register 1
REG[1Fh] Read/Write

7	6	5	4	3	2	1	0
n/a						Vertical Display Period Start Position Bits 9-8	

bits 9-0 Vertical Display Period Start Position Bits [9:0]
These bits specify the Vertical Display Period Start Position for TFT, HR-TFT and D-TFD panels in 1 line resolution. For passive LCD panels these bits must be set to 00h.

Note
[1] This register must be programmed such that the following formula is valid.
VDPS + VDP < VT
[2] For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 55.

FPLINE Pulse Width Register
REG[20h] Read/Write

7	6	5	4	3	2	1	0
FPLINE Pulse Polarity	FPLINE Pulse Width Bits 6-0						

bit 7 FPLINE Pulse Polarity

For active panels only (i.e. TFT/HR-TFT/D-TFD), this bit selects the polarity of the horizontal sync signal. The horizontal sync signal is typically FPLINE or LP, depending on the panel type.
When this bit = 0, the horizontal sync signal is active low.
When this bit = 1, the horizontal sync signal is active high.

bits 6-0 FPLINE Pulse Width Bits [6:0]

These bits specify the width of the panel horizontal sync signal, in 1 pixel resolution. The horizontal sync signal is typically FPLINE or LP, depending on the panel type.

FPLINE Pulse Width in number of pixels = (REG[20h] bits 6:0) + 1

Note

For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 55.

FPLINE Pulse Start Position Register 0
REG[22h] Read/Write

7	6	5	4	3	2	1	0
FPLINE Pulse Start Position Bits 7-0							

FPLINE Pulse Start Position Register 1
REG[23h] Read/Write

7	6	5	4	3	2	1	0
n/a						FPLINE Pulse Start Position Bits 9-8	

bits 9-0 FPLINE Pulse Start Position Bits [9:0]

These bits specify the start position of the horizontal sync signal, in 1 pixel resolution.

FPLINE Pulse Start Position in pixels = (REG[23h] bits 1-0, REG[22h] bits 7-0) + 1

Note

For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 55.

FPFRAME Pulse Width Register
REG[24h] Read/Write

7	6	5	4	3	2	1	0
FPFRAME Pulse Polarity	n/a				FPFRAME Pulse Width Bits 2-0		

bit 7 FPFRAME Pulse Polarity

For active panels only (i.e. TFT/HR-TFT/D-TFD), this bit selects the polarity of the vertical sync signal. The vertical sync signal is typically FPFRAME, SPS or DY, depending on the panel type.
When this bit = 0, the vertical sync signal is active low.
When this bit = 1, the vertical sync signal is active high.

bits 2-0 FPFRAME Pulse Width Bits [2:0]

These bits specify the width of the panel vertical sync signal, in 1 line resolution. The vertical sync signal is typically FPFRAME, SPS or DY, depending on the panel type.

FPFRAME Pulse Width in number of lines = (REG[24h] bits 2:0) + 1

Note

For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 55.

FPFRAME Pulse Start Position Register 0
REG[26h] Read/Write

7	6	5	4	3	2	1	0
FPFRAME Pulse Start Position Bits 7-0							

FPFRAME Pulse Start Position Register 1
REG[27h] Read/Write

7	6	5	4	3	2	1	0
n/a						FPFRAME Pulse Start Position Bits 9-8	

bits 9-0 FPFRAME Pulse Start Position Bits [9:0]

These bits specify the start position of the vertical sync signal, in 1 line resolution.

Note

For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 55.

D-TFD GCP Index Register
REG[28h] Read/Write

n/a			D-TFD GCP Index Bits 4-0				
7	6	5	4	3	2	1	0

bits 4-0 D-TFD GCP Index Bits [4:0]
For D-TFD panels only. These bits form the index that points to 32 8-bit GCP data registers.

D-TFD GCP Data Register
REG[2Ch] Read/Write

D-TFD GCP Data Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 D-TFD GCP Data Bits [7:0]
For D-TFD panel only. This register stores the data to be written to the GCP data bits and is controlled by the D-TFD GCP Index register (REG[28h]). For further information on the use of this register, see *Connecting to the Epson D-TFD Panels*, document number X31B-G-012-xx.

Note
The Panel Type bits (REG[10h] bits 1:0) must be set to 11 (D-TFD) for the GCP Data bits to have any hardware effect.

8.3.5 Display Mode Registers

Display Mode Register REG[70h]							Read/Write
Display Blank	Dithering Disable	Hardware Video Invert Enable	Software Video Invert	n/a	Bit-per-pixel Select Bits 2-0		
7	6	5	4	3	2	1	0

bit 7 Display Blank

When this bit = 0, the LCD display pipeline is enabled.
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).

bit 6 Dithering Disable

Dithering allows 64 intensity levels for each color component (RGB). In monochrome modes where only the Green color component of the Look-Up-Table is used, 64 shades of gray are available for each position used in the LUT. In color modes, 64 shades of color are available for each color component resulting in 256K possible color combinations.
When this bit = 0, dithering is enabled for passive LCD panels.
When this bit = 1, dithering is disabled for passive LCD panels.

Note

This bit does not refer to the number of simultaneously displayed colors but rather the maximum available colors (refer to Table 8-9: “LCD Bit-per-pixel Selection,” on page 109 for the maximum number of simultaneously displayed colors).

bit 5 Hardware Video Invert Enable

This bit allows the Video Invert feature to be controlled using the General Purpose IO pin GPIO0. **This option is not available if configured for a HR-TFT or D-TFD as GPIO0 is used as an LCD control signal by both panels.**

When this bit = 0, GPIO0 has no effect on the video data.

When this bit = 1, video data may be inverted via GPIO0.

Note

The S1D13706 requires some configuration before the hardware video invert feature can be enabled.

- CNF3 must be set to 1 at RESET#
- GPIO Pin Input Enable (REG[A9h] bit 7) must be set to 1
- GPIO0 Pin IO Configuration (REG[A8h] bit 0) must be set to 0

If Hardware Video Invert is not available (i.e. HR-TFT panel is used), the video invert function can be controlled by software using REG[70h] bit 4. The following table summarizes the video invert options available.

Table 8-8: Inverse Video Mode Select Options

Hardware Video Invert Enable	Software Video Invert	GPIO0	Video Data
0	0	X	Normal
0	1	X	Inverse
1	X	0	Normal
1	X	1	Inverse

Note

Video data is inverted after the Look-Up Table.

bit 4 Software Video Invert

When this bit = 0, video data is normal.

When this bit = 1, video data is inverted.

See Table 8-8: “Inverse Video Mode Select Options”.

Note

Video data is inverted after the Look-Up Table

bits 2-0 Bit-per-pixel Select Bits [2:0]

These bits select the color depth (bit-per-pixel) for the displayed data for both the main window and the PIP+ window (if active).

Note

1, 2, 4 and 8 bpp color depths use the 18-bit LUT, allowing a maximum number of 256K available colors on TFT panels. 16 bpp mode bypasses the LUT, allowing a maximum of only 64K available colors.

Table 8-9: LCD Bit-per-pixel Selection

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

Special Effects Register
REG[71h] Read/Write

Display Data Word Swap	Display Data Byte Swap	n/a	PIP+ Window Enable	n/a		SwivelView Mode Select Bits 1-0	
7	6	5	4	3	2	1	0

bit 7 Display Data Word Swap

The display pipe fetches 32-bits of data from the display buffer. This bit enables the lower 16-bit word and the upper 16-bit word to be swapped before sending them to the LCD display. If the Display Data Byte Swap bit is also enabled, then the byte order of the fetched 32-bit data is reversed.

Note

For further information on byte swapping for Big Endian mode, see Section 14, “Big-Endian Bus Interface” on page 144.

bit 6 Display Data Byte Swap

The display pipe fetches 32-bits of data from the display buffer. This bit enables byte 0 and byte 1 to be swapped, and byte 2 and byte 3 to be swapped, before sending them to the LCD display. If the Display Data Word Swap bit is also enabled, then the byte order of the fetched 32-bit data is reversed.

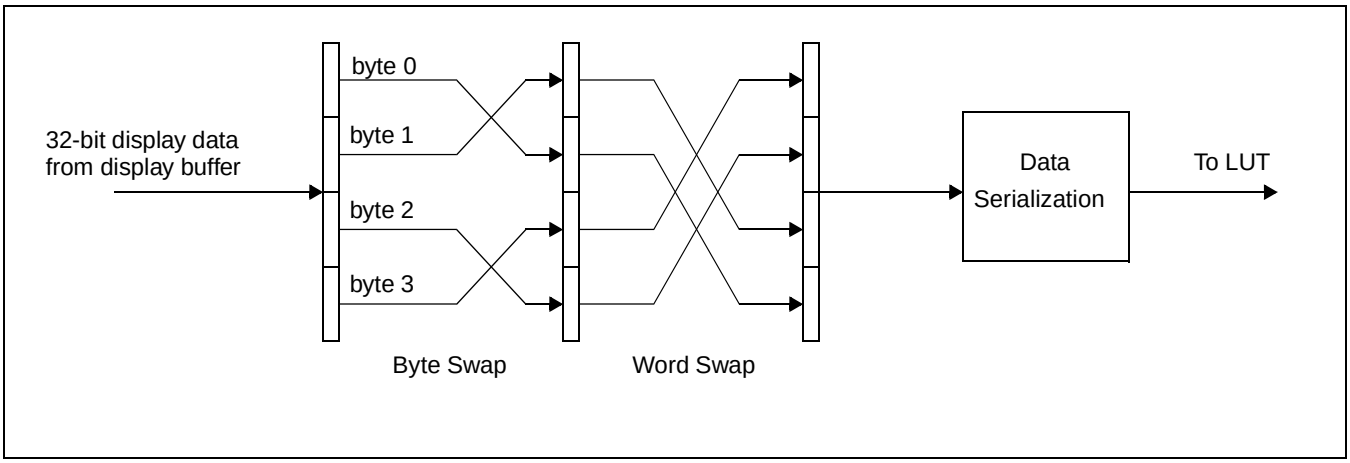


Figure 8-1: Display Data Byte/Word Swap

Note

For further information on byte swapping for Big Endian mode, see Section 14, “Big-Endian Bus Interface” on page 144.

bit 4 Picture-in-Picture Plus (PIP+) Window Enable

This bit enables the PIP+ window within the main window used for the Picture-in-Picture Plus feature. The location of the PIP+ window within the landscape window is determined by the PIP+ Window X Position registers (REG[84h], REG[85h], REG[8Ch], REG[8Dh]) and PIP+ Window Y Position registers (REG[88h], REG[89h], REG[90h], REG[91h]). The PIP+ window has its own Display Start Address register (REG[7Ch], REG[7Dh], REG[7Eh]) and Memory Address Offset register (REG[80h], REG[81h]). The PIP+ window shares the same color depth and SwivelView™ orientation as the main window.

bit 1-0 SwivelView Mode Select Bits [1:0]

These bits select different SwivelView™ orientations:

Table 8-10: SwivelView™ Mode Select Options

SwivelView Mode Select Bits	SwivelView Orientation
00	0° (Normal)
01	90°
10	180°
11	270°

Main Window Display Start Address Register 0
REG[74h] Read/Write

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

Main Window Display Start Address Register 1
REG[75h] Read/Write

7	6	5	4	3	2	1	0
Main window Display Start Address Bits 15-8							

Main Window Display Start Address Register 2
REG[76h] Read/Write

7	6	5	4	3	2	1	0
n/a							Main window Display Start Address Bit 16

bits 16-0 Main Window Display Start Address Bits [16:0]
This register specifies the starting address, in DWORDS, for the LCD image in the display buffer for the main window.

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

Main Window Display Start Address bits 16:0
= image address ÷ 4 (valid only for SwivelView 0°)

Note
For information on setting this register for other SwivelView orientations, see Section 12, “SwivelView™” on page 136.

Main Window Line Address Offset Register 0
REG[78h] Read/Write

7	6	5	4	3	2	1	0
Main window Line Address Offset Bits 7-0							

Main Window Line Address Offset Register 1
REG[79h] Read/Write

7	6	5	4	3	2	1	0
n/a						Main window Line Address Offset Bits 9-8	

bits 9-0 Main Window Line Address Offset Bits [9:0]

This register specifies the offset, in DWORDS, from the beginning of one display line to the beginning of the next display line in the main window. **Note that this is a 32-bit address increment.** Calculate the Line Address Offset as follows:

Main Window Line Address Offset bits 9:0
= display width in pixels ÷ (32 ÷ bpp)

Note

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

8.3.6 Picture-in-Picture Plus (PIP+) Registers

PIP+ Window Display Start Address Register 0
REG[7C] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Display Start Address Bits 7-0							

PIP+ Window Display Start Address Register 1
REG[7Dh] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Display Start Address Bits 15-8							

PIP+ Window Display Start Address Register 2
REG[7Eh] Read/Write

7	6	5	4	3	2	1	0
n/a							PIP+ Window Display Start Address Bit 16

bits 16-0 PIP+ Window Display Start Address Bits [16:0]
These bits form the 17-bit address for the starting double-word of the PIP+ window.

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

Note
These bits have no effect unless the PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).

PIP+ Window Line Address Offset Register 0
REG[80h] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Line Address Offset Bits 7-0							

PIP+ Window Line Address Offset Register 1
REG[81h] Read/Write

7	6	5	4	3	2	1	0
n/a						PIP+ Window Line Address Offset Bits 9-8	

bits 9-0 PIP+ Window Line Address Offset Bits [9:0]
These bits are the LCD display's 10-bit address offset from the starting double-word of line "n" to the starting double-word of line "n + 1" for the PIP+ window. **Note that this is a 32-bit address increment.**

Note
These bits have no effect unless the PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).

PIP+ Window X Start Position Register 0
REG[84h] Read/Write

PIP+ Window X Start Position Bits 7-0							
7	6	5	4	3	2	1	0

PIP+ Window X Start Position Register 1
REG[85h] Read/Write

n/a						PIP+ Window X Start Position Bits 9-8	
7	6	5	4	3	2	1	0

bits 9-0 PIP+ Window X Start Position Bits [9:0]
These bits determine the X start position of the PIP+ window in relation to the origin of the panel. Due to the S1D13706 SwivelView feature, the X start position may not be a horizontal position value (only true in 0° and 180° SwivelView). For further information on defining the value of the X Start Position register, see Section 13, “Picture-in-Picture Plus (PIP+)” on page 141.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the X start position is incremented by *x* pixels where *x* is relative to the current color depth.

Table 8-11: 32-bit Address Increments for Color Depth

Color Depth	Pixel Increment (x)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

For 90° and 270° SwivelView the X start position is incremented in 1 line increments.

Depending on the color depth, some of the higher bits in this register are unused because the maximum horizontal display width is 1024 pixels.

Note
[1] These bits have no effect unless the PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).
[2] The effect of REG[84h] through REG[91h] takes place only after REG[91h] is written and at the next vertical non-display period.

PIP+ Window Y Start Position Register 0
REG[88h] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Y Start Position Bits 7-0							

PIP+ Window Y Start Position Register 1
REG[89h] Read/Write

7	6	5	4	3	2	1	0
n/a						PIP+ Window Y Start Position Bits 9-8	

bits 9-0 PIP+ Window Y Start Position Bits [9:0]
These bits determine the Y start position of the PIP+ window in relation to the origin of the panel. Due to the S1D13706 SwivelView feature, the Y start position may not be a vertical position value (only true in 0° and 180° SwivelView). For further information on defining the value of the Y Start Position register, see Section 13, “Picture-in-Picture Plus (PIP+)” on page 141.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the Y start position is incremented in 1 line increments. For 90° and 270° SwivelView the Y start position is incremented by *y* pixels where *y* is relative to the current color depth.

Table 8-12: 32-bit Address Increments for Color Depth

Color Depth	Pixel Increment (y)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

Depending on the color depth, some of the higher bits in this register are unused because the maximum vertical display height is 1024 pixels.

Note
[1] These bits have no effect unless the PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).
[2] The effect of REG[84h] through REG[91h] takes place only after REG[91h] is written and at the next vertical non-display period.

PIP+ Window X End Position Register 0
REG[8Ch] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window X End Position Bits 7-0							

PIP+ Window X End Position Register 1
REG[8Dh] Read/Write

7	6	5	4	3	2	1	0
n/a						PIP+ Window X End Position Bits 9-8	

bits 9-0 PIP+ Window X End Position Bits [9:0]

These bits determine the X end position of the PIP+ window in relation to the origin of the panel. Due to the S1D13706 SwivelView feature, the X end position may not be a horizontal position value (only true in 0° and 180° SwivelView). For further information on defining the value of the X End Position register, see Section 13, “Picture-in-Picture Plus (PIP+)” on page 141.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the X end position is incremented by *x* pixels where *x* is relative to the current color depth.

Table 8-13: 32-bit Address Increments for Color Depth

Color Depth	Pixel Increment (x)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

For 90° and 270° SwivelView the X end position is incremented in 1 line increments.

Depending on the color depth, some of the higher bits in this register are unused because the maximum horizontal display width is 1024 pixels.

Note

[1] These bits have no effect unless the PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).

[2] The effect of REG[84h] through REG[91h] takes place only after REG[91h] is written and at the next vertical non-display period.

PIP+ Window Y End Position Register 0
REG[90h] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Y End Position Bits 7-0							

PIP+ Window Y End Position Register 1
REG[91h] Read/Write

7	6	5	4	3	2	1	0
n/a						PIP+ Window Y End Position Bits 9-8	

bits 9-0 PIP+ Window Y End Position Bits [9:0]

These bits determine the Y end position of the PIP+ window in relation to the origin of the panel. Due to the S1D13706 SwivelView feature, the Y end position may not be a vertical position value (only true in 0° and 180° SwivelView). For further information on defining the value of the Y End Position register, see Section 13, “Picture-in-Picture Plus (PIP+)” on page 141.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the Y end position is incremented in 1 line increments. For 90° and 270° SwivelView the Y end position is incremented by *y* pixels where *y* is relative to the current color depth.

Table 8-14: 32-bit Address Increments for Color Depth

Color Depth	Pixel Increment (y)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

Depending on the color depth, some of the higher bits in this register are unused because the maximum vertical display height is 1024 pixels.

Note
[1] These bits have no effect unless the PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).
[2] The effect of REG[84h] through REG[91h] takes place only after REG[91h] is written and at the next vertical non-display period.

8.3.7 Miscellaneous Registers

Power Save Configuration Register
REG[A0h] Read/Write

7	6	5	4	3	2	1	0
Vertical Non-Display Period Status (RO)	n/a			Memory Controller Power Save Status (RO)	n/a		Power Save Mode Enable

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

bit 3 Memory Controller Power Save Status
This read-only status bit indicates the power save state of the memory controller.
When this bit = 0, the memory controller is powered up.
When this bit = 1, the memory controller is powered down.

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.
At reset, this bit is set to 1.

Reserved
REG[A1h] Read/Write

7	6	5	4	3	2	1	0
n/a							Reserved

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

Software Reset Register
REG[A2h] Read/Write

7	6	5	4	3	2	1	0
Reserved	n/a						Software Reset (WO)

bit 7 Reserved.
This bit must be set to 0.

bit 0 Software Reset
This bit is write only.
When a one is written to this bit, **the S1D13706 registers are reset**. This bit has no effect on the contents of the display buffer.

Reserved
REG[A3h] Read/Write

Reserved	n/a						
7	6	5	4	3	2	1	0

bit 7 Reserved.
This bit must be set to 0.

Scratch Pad Register 0
REG[A4h] Read/Write

Scratch Pad Bits 7-0							
7	6	5	4	3	2	1	0

Scratch Pad Register 1
REG[A5h] Read/Write

Scratch Pad Bits 15-8							
7	6	5	4	3	2	1	0

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

8.3.8 General IO Pins Registers

General Purpose IO Pins Configuration Register 0
REG[A8h] Read/Write

n/a	GPIO6 Pin IO Configuration	GPIO5 Pin IO Configuration	GPIO4 Pin IO Configuration	GPIO3 Pin IO Configuration	GPIO2 Pin IO Configuration	GPIO1 Pin IO Configuration	GPIO0 Pin IO Configuration
7	6	5	4	3	2	1	0

Note
[1] If CNF3 = 0 at RESET#, then all GPIO pins are configured as outputs only and this register has no effect. This case allows the GPIO pins to be used by the HR-TFT/D-TFD panel interfaces. For a summary of GPIO usage for HR-TFT/D-TFD, see Table 4-9: "LCD Interface Pin Mapping," on page 29.
[2] The input functions of the GPIO pins are not enabled until REG[A9h] bit 7 is set to 1.

bit 6 GPIO6 Pin IO Configuration
When this bit = 0 (default), GPIO6 is configured as an input pin.
When this bit = 1, GPIO6 is configured as an output pin.

bit 5 GPIO5 Pin IO Configuration
When this bit = 0 (default), GPIO5 is configured as an input pin.
When this bit = 1, GPIO5 is configured as an output pin.

bit 4 GPIO4 Pin IO Configuration
When this bit = 0 (default), GPIO4 is configured as an input pin.
When this bit = 1, GPIO4 is configured as an output pin.

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

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

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

bit 0 GPIO0 Pin IO Configuration
When this bit = 0 (default), GPIO0 is configured as an input pin.
When this bit = 1, GPIO0 is configured as an output pin.

General Purpose IO Pins Configuration Register 1
REG[A9h] Read/Write

7	6	5	4	3	2	1	0
GPIO Pin Input Enable	n/a						

bit 7 GPIO Pin Input Enable
This bit is used to enable the input function of the GPIO pins. It must be changed to a 1 after power-on reset to enable the input function of the GPIO pins (default is 0).

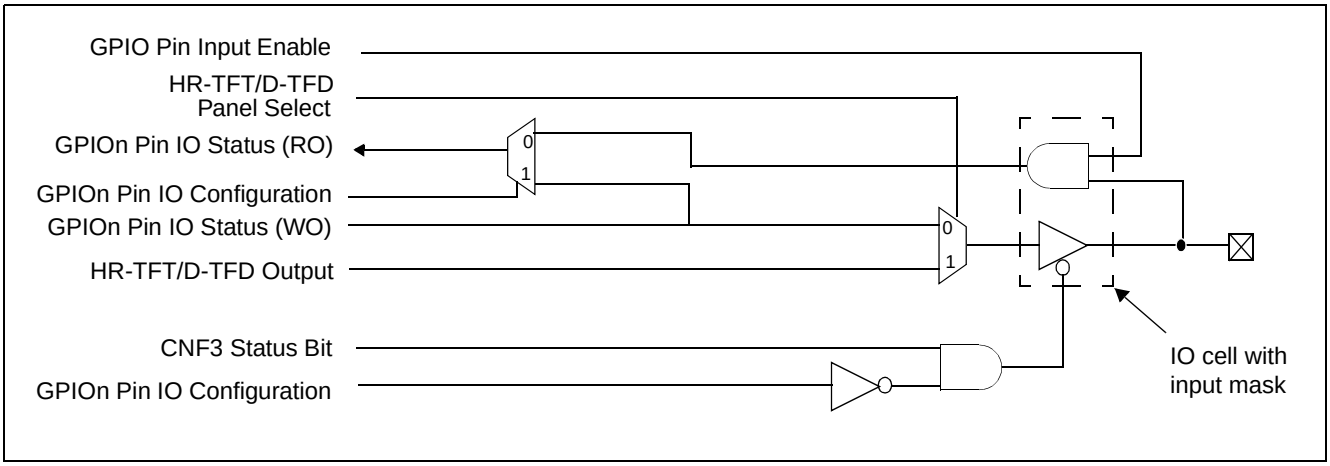


Figure 8-2: Example IO Cell

General Purpose IO Pins Status/Control Register 0 REG[ACh]							Read/Write
n/a	GPIO6 Pin IO Status	GPIO5 Pin IO Status	GPIO4 Pin IO Status	GPIO3 Pin IO Status	GPIO2 Pin IO Status	GPIO1 Pin IO Status	GPIO0 Pin IO Status
7	6	5	4	3	2	1	0

Note
For information on GPIO pin mapping when HR-TFT/D-TFD panels are selected, see Table 4-9: “LCD Interface Pin Mapping,” on page 29.

bit 6 GPIO6 Pin IO Status
When a D-TFD panel is not selected (REG[10h] bits 1:0) and GPIO6 is configured as an output, writing a 1 to this bit drives GPIO6 high and writing a 0 to this bit drives GPIO6 low.
When a D-TFD panel is not selected (REG[10h] bits 1:0) and GPIO6 is configured as an input, a read from this bit returns the status of GPIO6.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal YSCLD signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal YSCLD signal is forced low.

bit 5 GPIO5 Pin IO Status
When a D-TFD panel is not selected (REG[10h] bits 1:0) and GPIO5 is configured as an output, writing a 1 to this bit drives GPIO5 high and writing a 0 to this bit drives GPIO5 low.
When a D-TFD panel is not selected (REG[10h] bits 1:0) and GPIO5 is configured as an input, a read from this bit returns the status of GPIO5.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal DD_P1 signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal DD_P1 signal is forced low.

bit 4 GPIO4 Pin IO Status
When a D-TFD panel is not selected (REG[10h] bits 1:0) and GPIO4 is configured as an output, writing a 1 to this bit drives GPIO4 high and writing a 0 to this bit drives GPIO4 low.
When a D-TFD panel is not selected (REG[10h] bits 1:0) and GPIO4 is configured as an input, a read from this bit returns the status of GPIO4.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal RES signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal RES signal is forced low.

bit 3 GPIO3 Pin IO Status
When neither a D-TFD panel or a HR-TFT are selected (REG[10h] bits 1:0) and 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 neither a D-TFD panel or a HR-TFT are selected (REG[10h] bits 1:0) and GPIO3 is configured as an input, a read from this bit returns the status of GPIO3.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal FRS signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal FRS signal is forced low.

When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 1 is written to this bit, the HR-TFT signal SPL signal is enabled.
When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 0 is written to this bit, the HR-TFT signal SPL signal is forced low.

bit 2 GPIO2 Pin IO Status
When neither a D-TFD panel or a HR-TFT are selected (REG[10h] bits 1:0) and 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 neither a D-TFD panel or a HR-TFT are selected (REG[10h] bits 1:0) and GPIO2 is configured as an input, a read from this bit returns the status of GPIO2.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal FR signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal FR signal is forced low.

When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 1 is written to this bit, the HR-TFT signal REV signal is enabled.
When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 0 is written to this bit, the HR-TFT signal REV signal is forced low.

bit 1 GPIO1 Pin IO Status
When neither a D-TFD panel or a HR-TFT are selected (REG[10h] bits 1:0) and 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 neither a D-TFD panel or a HR-TFT are selected (REG[10h] bits 1:0) and GPIO1 is configured as an input, a read from this bit returns the status of GPIO1.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal YSCL signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal YSCL signal is forced low.

When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 1 is written to this bit, the HR-TFT signal CLS signal is enabled.
When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 0 is written to this bit, the HR-TFT signal CLS signal is forced low.

bit 0 GPIO0 Pin IO Status

When neither a D-TFD panel or a HR-TFT are selected (REG[10h] bits 1:0) and GPIO0 is configured as an output, writing a 1 to this bit drives GPIO0 high and writing a 0 to this bit drives GPIO0 low.
When neither a D-TFD panel or a HR-TFT are selected (REG[10h] bits 1:0) and GPIO0 is configured as an input, a read from this bit returns the status of GPIO0.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal XINH signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal XINH signal is forced low.

When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 1 is written to this bit, the HR-TFT signal PS signal is enabled.
When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 0 is written to this bit, the HR-TFT signal PS signal is forced low.

General Purpose IO Pins Status/Control Register 1
REG[ADh] Read/Write

7	6	5	4	3	2	1	0
GPO Control	n/a						

bit 7 GPO Control

This bit controls the General Purpose Output pin.
Writing a 0 to this bit drives GPO to low.
Writing a 1 to this bit drives GPO to high.

Note

Many implementations use the GPO pin to control the LCD bias power (see Section 6.3, “LCD Power Sequencing” on page 52).

8.3.9 Pulse Width Modulation (PWM) Clock and Contrast Voltage (CV) Pulse Configuration Registers

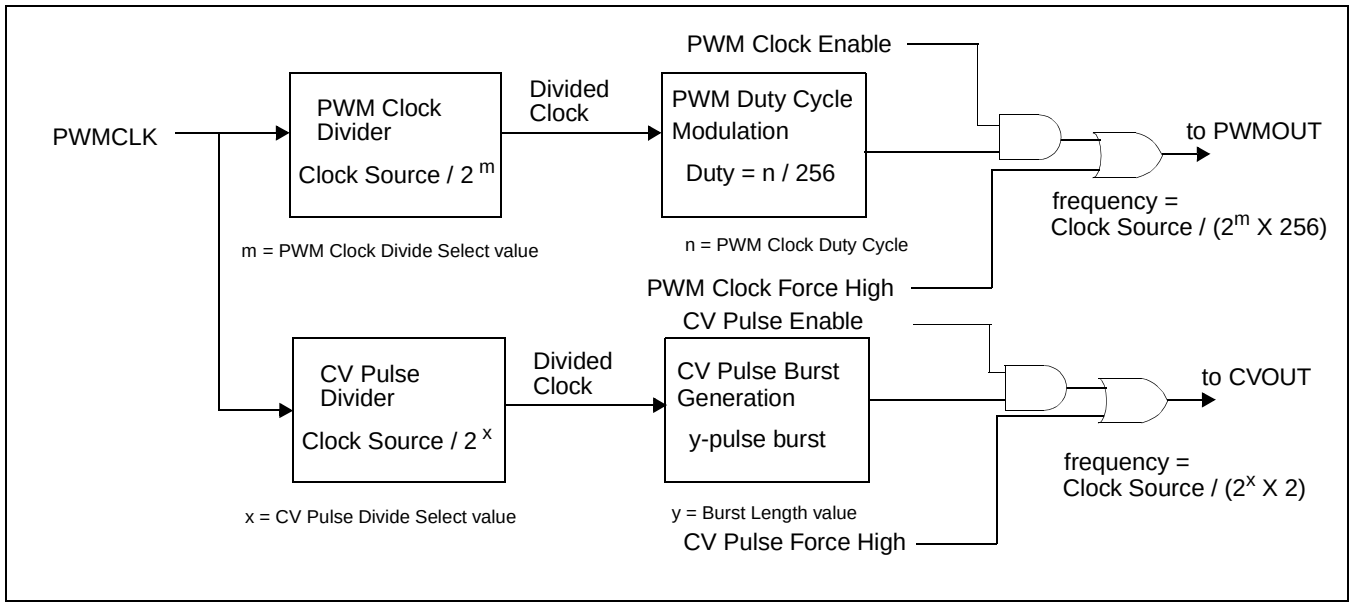


Figure 8-3: PWM Clock/CV Pulse Block Diagram

Note
For further information on PWMCLK, see Section 7.1.4, "PWMCLK" on page 90.

PWM Clock / CV Pulse Control Register
REG[B0h] Read/Write

PWM Clock Force High	n/a		PWM Clock Enable	CV Pulse Force High	CV Pulse Burst Status (RO)	CV Pulse Burst Start	CV Pulse Enable
7	6	5	4	3	2	1	0

bit 7 and bit 4 PWM Clock Force High (bit 7) and PWM Clock Enable (bit 4)
These bits control the PWMOUT pin and PWM Clock circuitry as follows.

Table 8-15: PWM Clock Control

Bit 7	Bit 4	Result
0	1	PWM Clock circuitry enabled (controlled by REG[B1h] and REG[B3h])
0	0	PWMOUT forced low
1	x	PWMOUT forced high

x = don't care

When PWMOUT is forced low or forced high it can be used as a general purpose output.

Note
The PWM Clock circuitry is disabled when Power Save Mode is enabled.

bit 3 and bit 0 CV Pulse Force High (bit 3) and CV Pulse Enable (bit 0)
These bits control the CVOUT pin and CV Pulse circuitry as follows.

Table 8-16: CV Pulse Control

Bit 3	Bit 0	Result
0	1	CV Pulse circuitry enabled (controlled by REG[B1h] and REG[B2h])
0	0	CVOUT forced low
1	x	CVOUT forced high

x = don't care

When CVOUT is forced low or forced high it can be used as a general purpose output.

Note
[1] Bit 3 must be set to 0 and bit 0 must be set to 1 before initiating a new burst using the CV Pulse Burst Start bit.
[2] The CV Pulse circuitry is disabled when Power Save Mode is enabled.

bit 2 CV Pulse Burst Status
This is a read-only bit. A "1" indicates a CV pulse burst is occurring. A "0" indicates no CV pulse burst is occurring. Software should wait for this bit to clear before starting another burst.

bit 1 CV Pulse Burst Start
A 1 in this bit initiates a single CVOUT pulse burst. The number of clock pulses generated is programmable from 1 to 256. The frequency of the pulses is the divided CV Pulse source divided by 2, with 50/50 duty cycle. This bit should be cleared to 0 by software before initiating a new burst.

Note
This bit has effect only if the CV Pulse Enable bit is 1.

bit 0 CV Pulse Enable
See description for bit 3.

PWM Clock / CV Pulse Configuration Register
REG[B1h] Read/Write

PWM Clock Divide Select Bits 3-0				CV Pulse Divide Select Bits 2-0			PWMCLK Source Select
7	6	5	4	3	2	1	0

bits 7-4 PWM Clock Divide Select Bits [3:0]
The value of these bits represents the power of 2 by which the selected PWM clock source is divided.

Table 8-17: PWM Clock Divide Select Options

PWM Clock Divide Select Bits [3:0]	PWM Clock Divide Amount
0h	1
1h	2
2h	4
3h	8
...	...
Ch	4096
Dh-Fh	Reserved

Note
This divided clock is further divided by 256 before it is output at PWMOUT.

bits 3-1 CV Pulse Divide Select Bits [2:0]
The value of these bits represents the power of 2 by which the selected CV Pulse source is divided.

Table 8-18: CV Pulse Divide Select Options

CV Pulse Divide Select Bits [2:0]	CV Pulse Divide Amount
0h	1
1h	2
2h	4
3h	8
...	...
7h	128

Note
This divided clock is further divided by 2 before it is output at the CVOUT.

bit 0 PWMCLK Source Select
When this bit = 0, the clock source for PWMCLK is CLKI.
When this bit = 1, the clock source for PWMCLK is CLKI2.

Note
For further information on the PWMCLK source select, see Section 7.2, “Clock Selection” on page 91.

CV Pulse Burst Length Register
REG[B2h] Read/Write

CV Pulse Burst Length Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 CV Pulse Burst Length Bits [7:0]
The value of this register determines the number of pulses generated in a single CV Pulse burst:

Number of pulses in a burst = (ContentsOfThisRegister) + 1

PWMOUT Duty Cycle Register
REG[B3h] Read/Write

PWMOUT Duty Cycle Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 PWMOUT Duty Cycle Bits [7:0]
This register determines the duty cycle of the PWMOUT output.

Table 8-19: PWMOUT Duty Cycle Select Options

PWMOUT Duty Cycle [7:0]	PWMOUT Duty Cycle
00h	Always Low
01h	High for 1 out of 256 clock periods
02h	High for 2 out of 256 clock periods
...	...
FFh	High for 255 out of 256 clock periods

9 Frame Rate Calculation

The following formula is used to calculate the display frame rate.

$$\text{FrameRate} = \frac{f_{PCLK}}{(HT) \times (VT)}$$

Where:

f_{PCLK} = PClk frequency (Hz)

HT = Horizontal Total
= ((REG[12h] bits 6-0) + 1) x 8 Pixels

VT = Vertical Total
= ((REG[19h] bits 1-0, REG[18h] bits 7-0) + 1) Lines

10 Display Data Formats

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

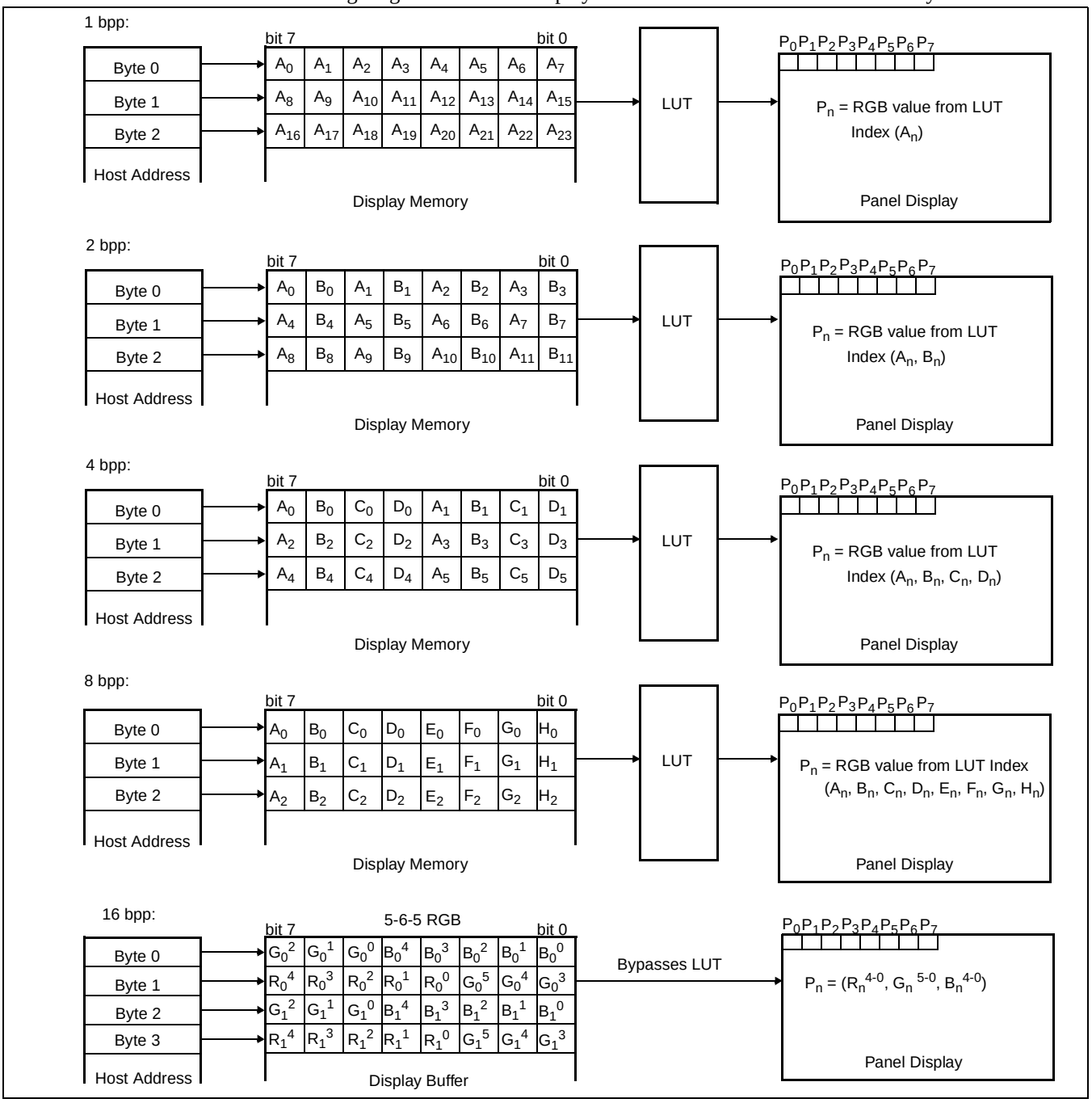


Figure 10-1: 4/8/16 Bit-Per-Pixel Display Data Memory Organization

Note

1. The Host-to-Display mapping shown here is for a little endian system.
2. For 16 bpp format, R_n, G_n, B_n represent the red, green, and blue color components.

11 Look-Up Table Architecture

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

Note
When Video Data Invert is enabled the video data is inverted after the Look-Up Table.

11.1 Monochrome Modes

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

1 Bit-per-pixel Monochrome Mode

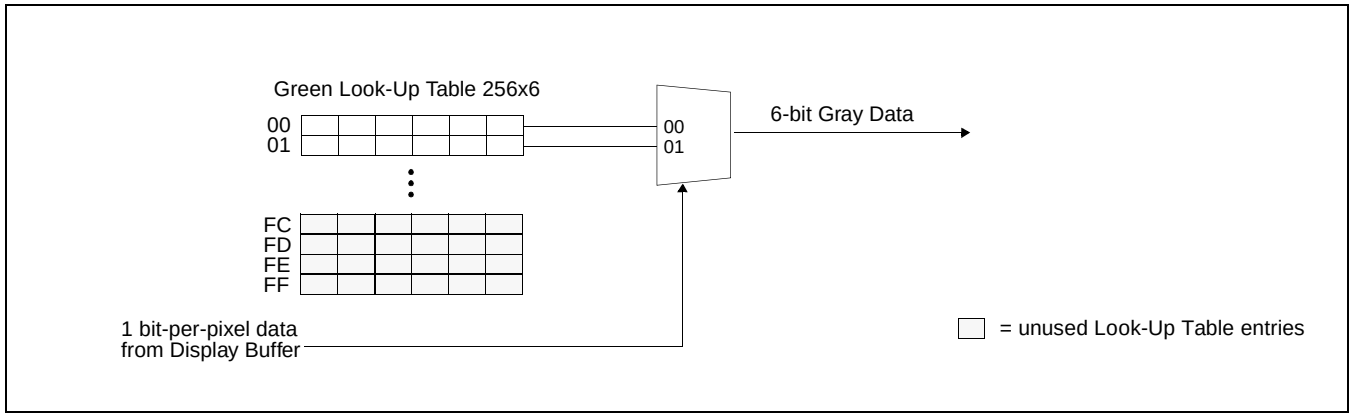


Figure 11-1: 1 Bit-per-pixel Monochrome Mode Data Output Path

2 Bit-per-pixel Monochrome Mode

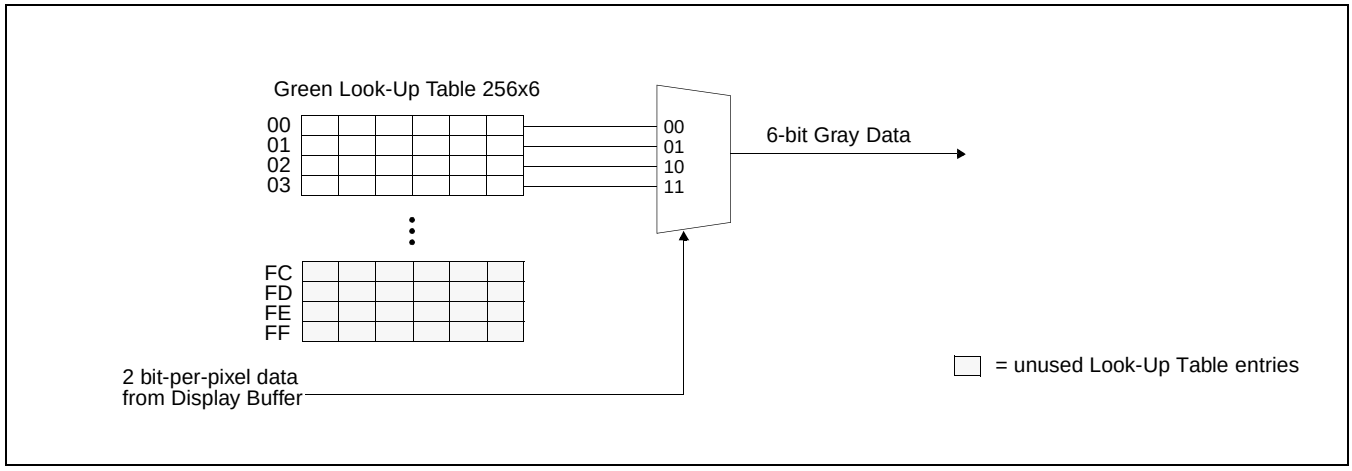


Figure 11-2: 2 Bit-per-pixel Monochrome Mode Data Output Path

4 Bit-per-pixel Monochrome Mode

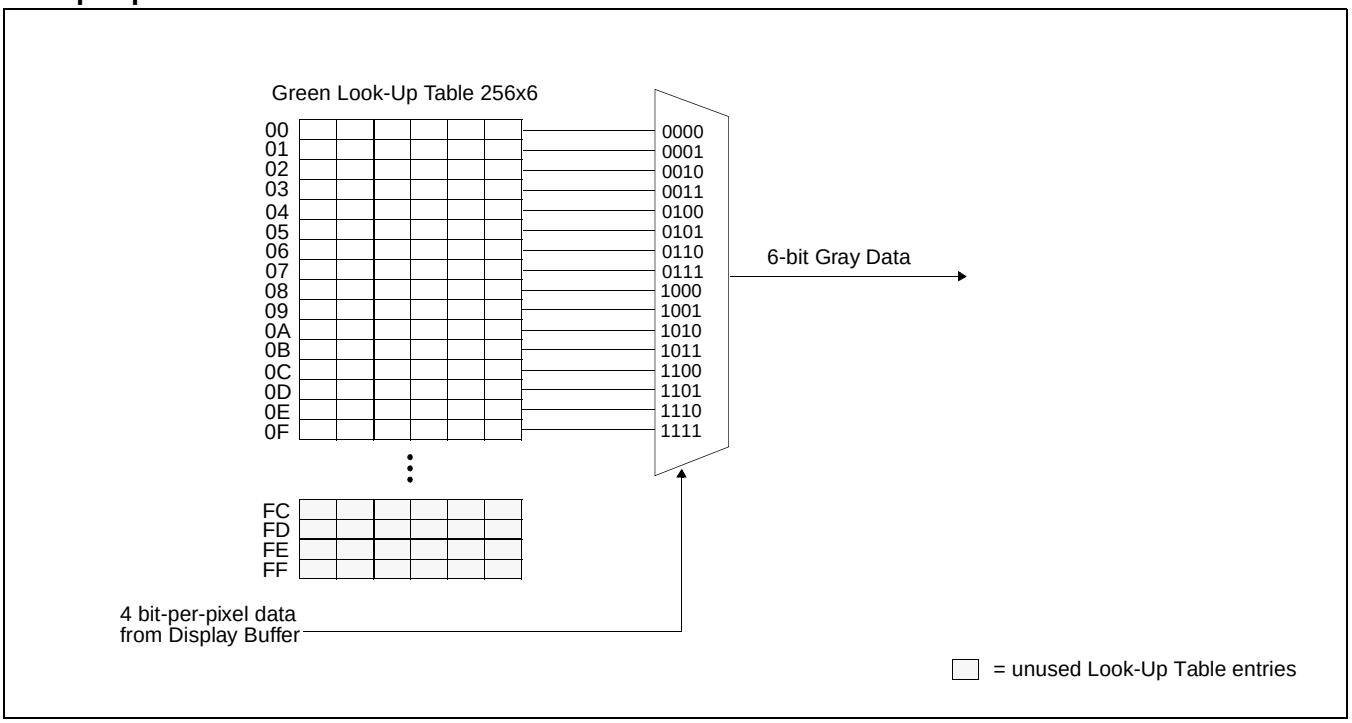


Figure 11-3: 4 Bit-per-pixel Monochrome Mode Data Output Path

8 Bit-per-pixel Monochrome Mode

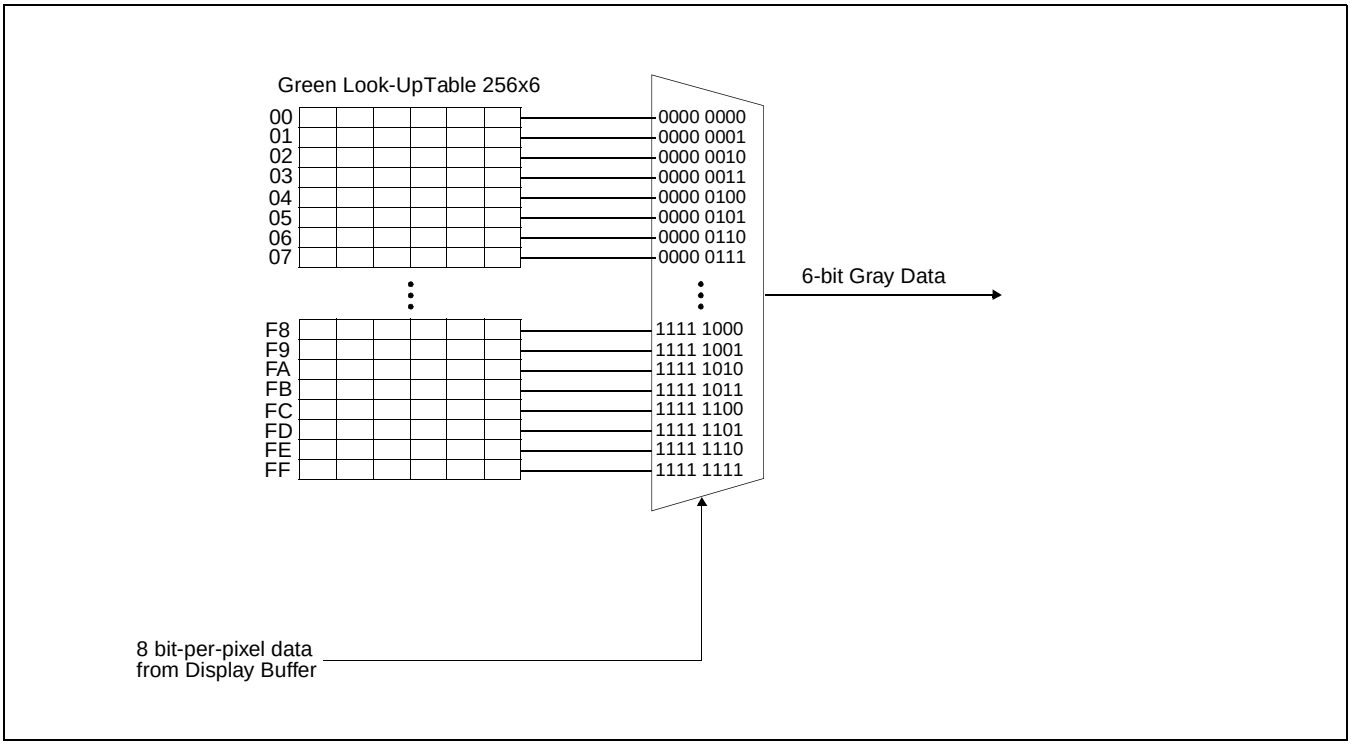


Figure 11-4: 8 Bit-per-pixel Monochrome Mode Data Output Path

16 Bit-Per-Pixel Monochrome Mode

The LUT is bypassed and the green data is directly mapped for this color depth– See “Display Data Formats” on page129..

11.2 Color Modes

1 Bit-Per-Pixel Color

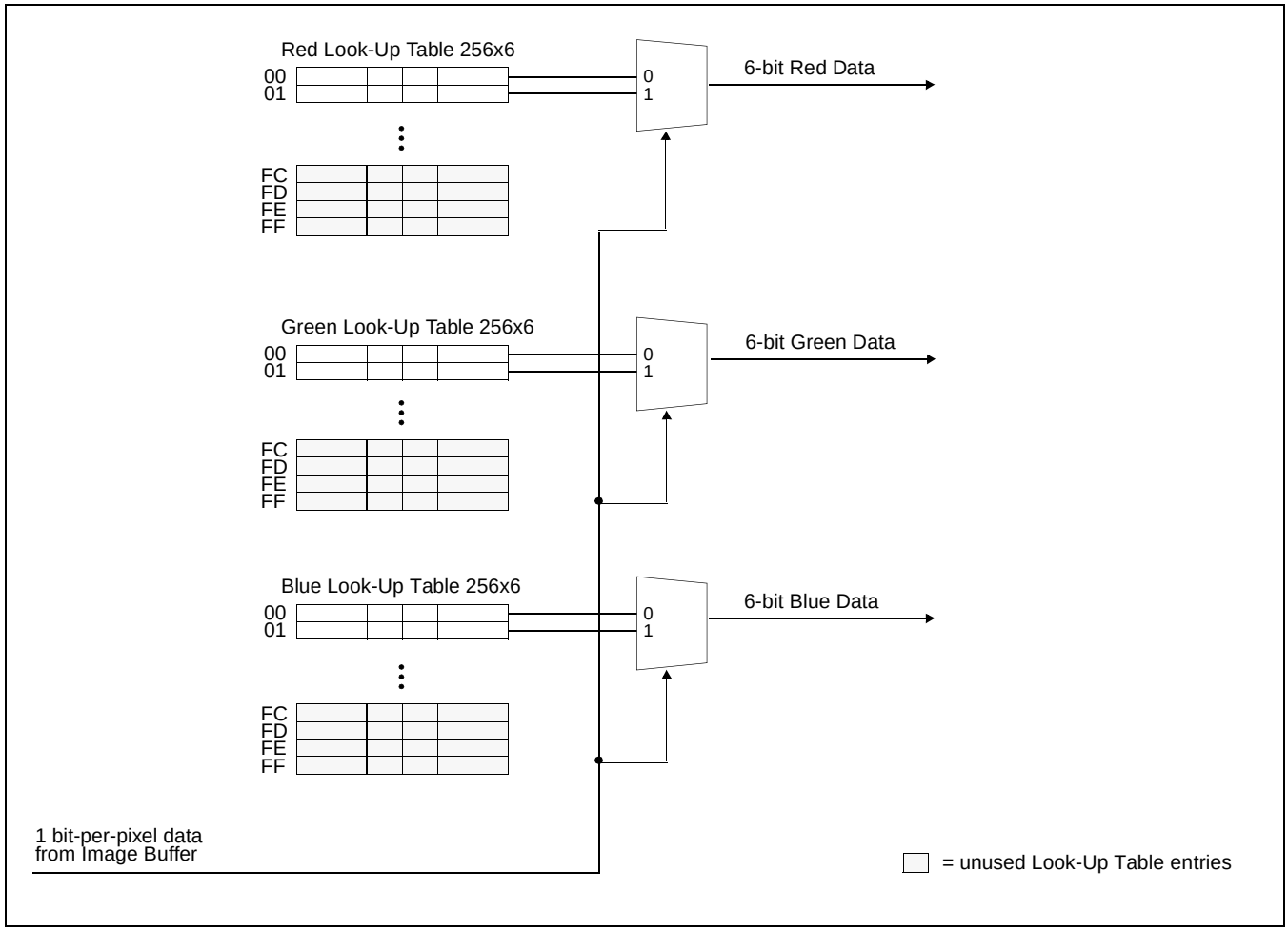


Figure 11-5: 1 Bit-Per-Pixel Color Mode Data Output Path

2 Bit-Per-Pixel Color

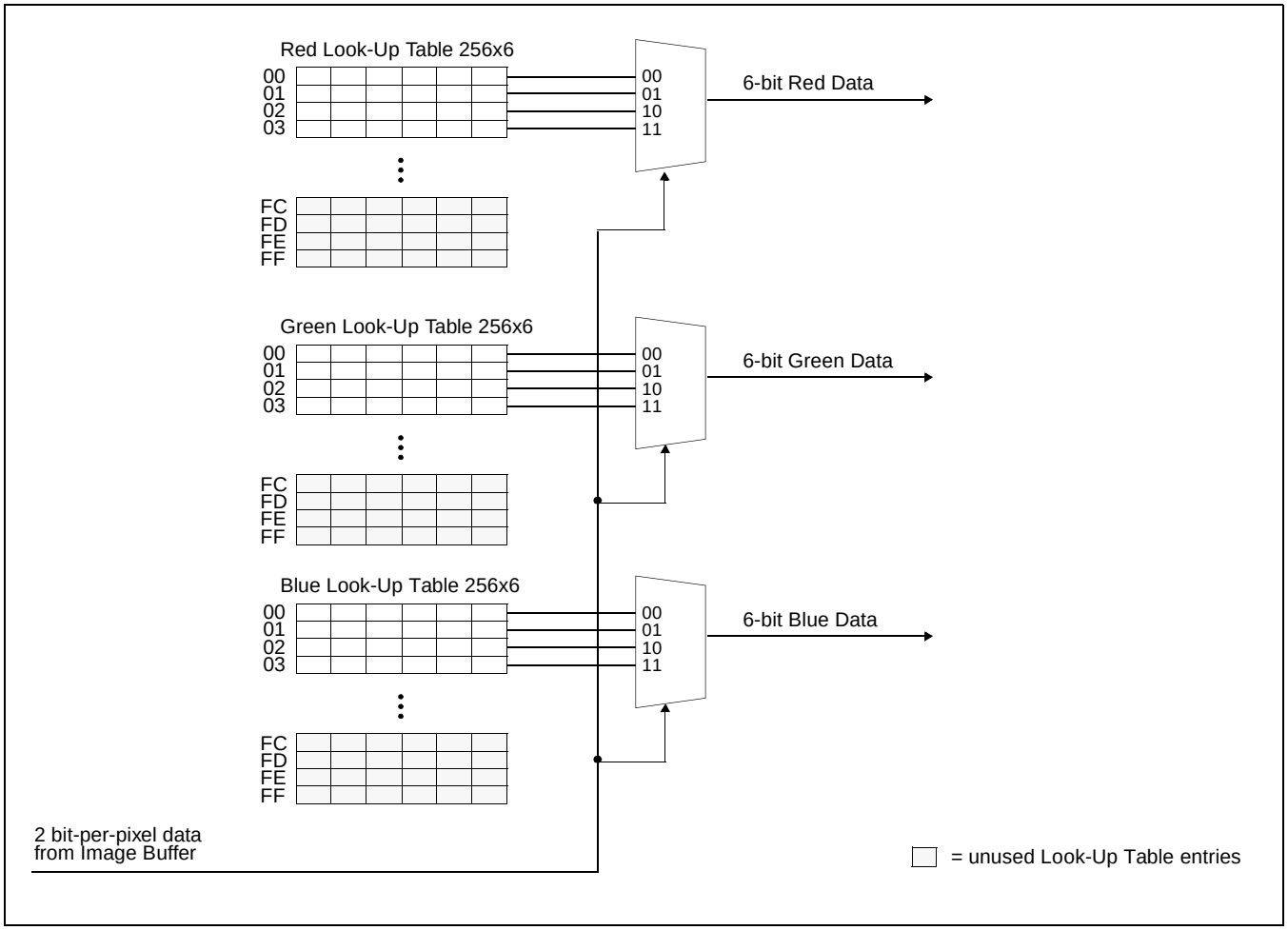


Figure 11-6: 2 Bit-Per-Pixel Color Mode Data Output Path

4 Bit-Per-Pixel Color

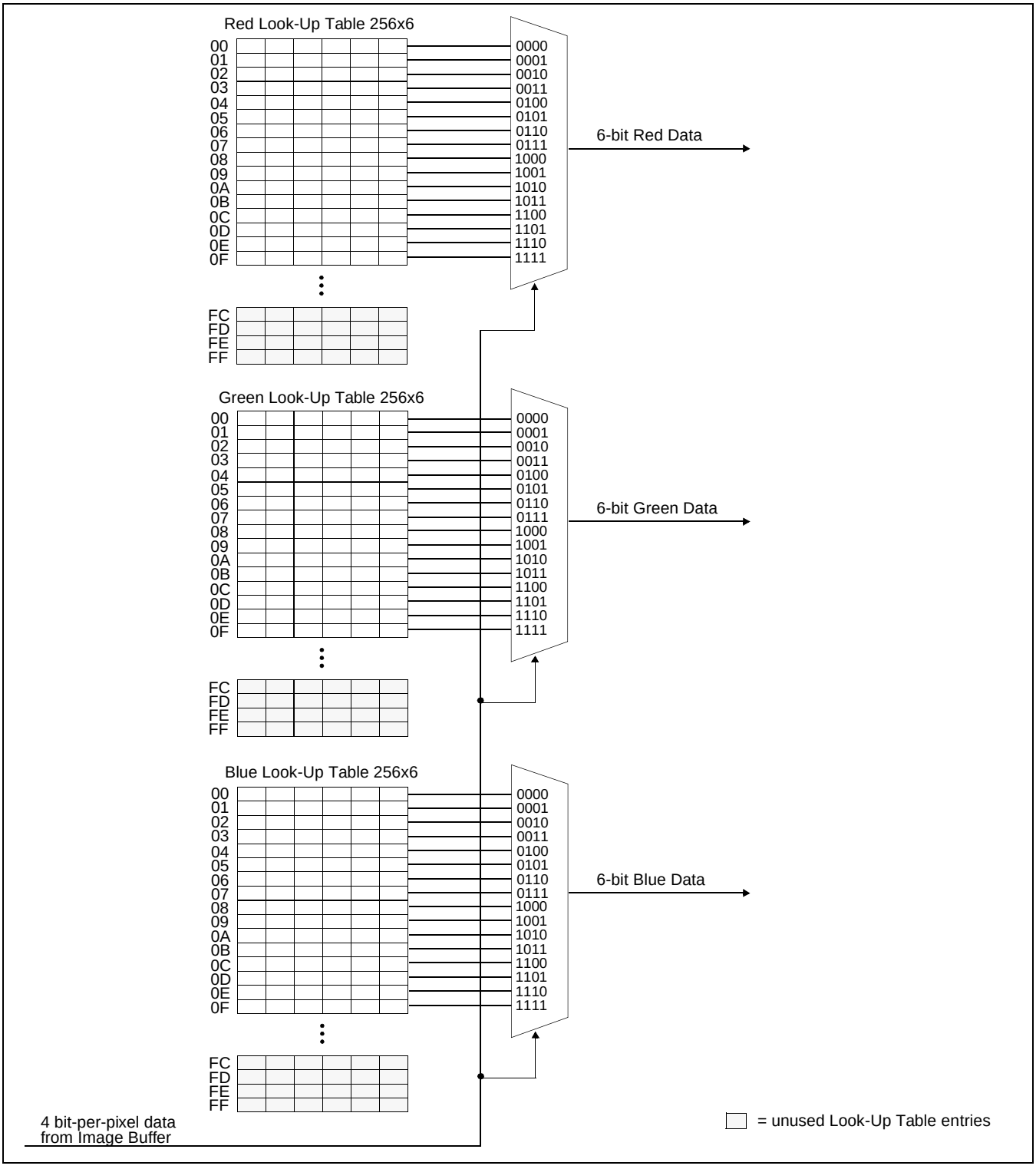


Figure 11-7: 4 Bit-Per-Pixel Color Mode Data Output Path

8 Bit-per-pixel Color Mode

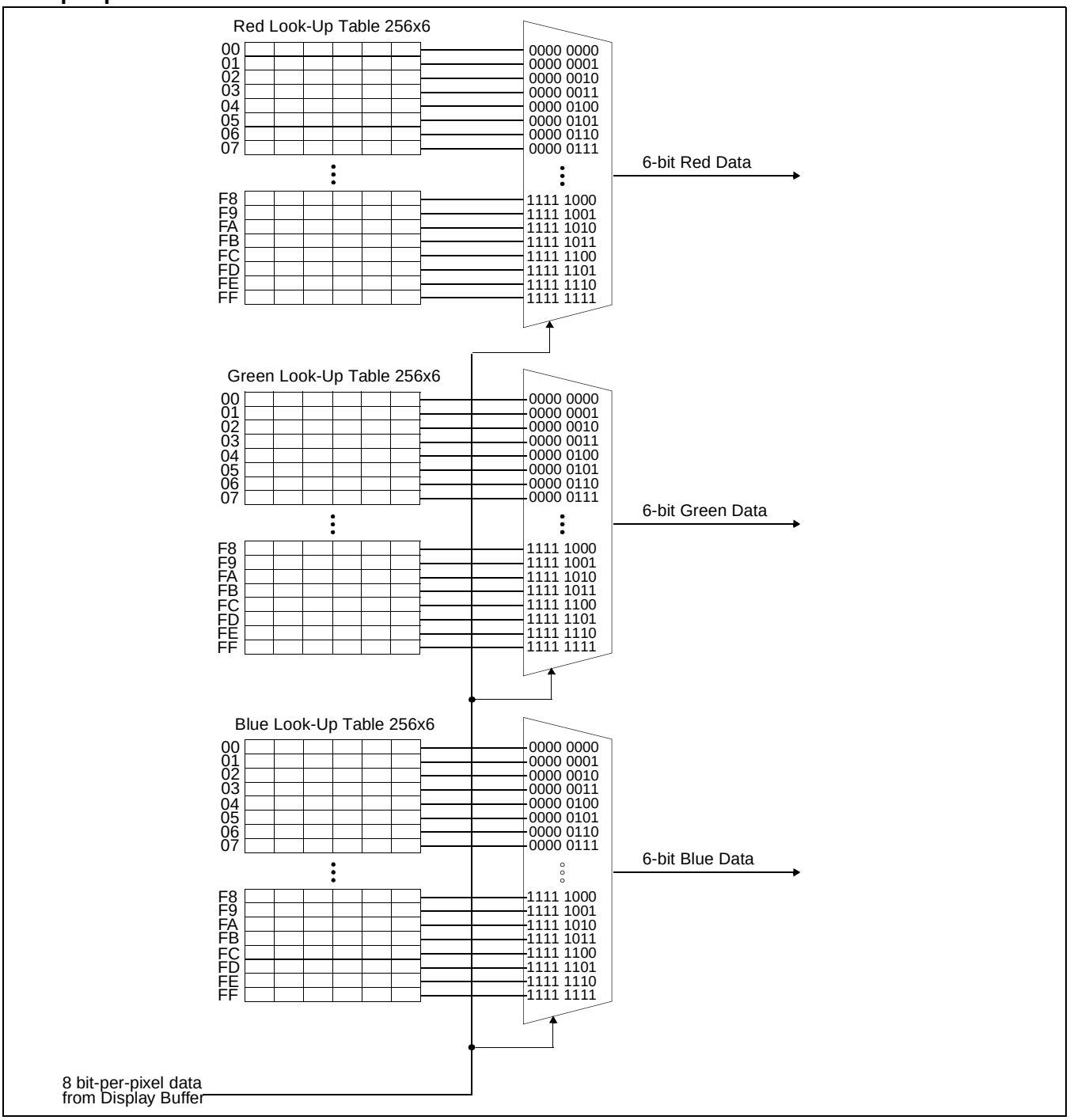


Figure 11-8: 8 Bit-per-pixel Color Mode Data Output Path

16 Bit-Per-Pixel Color Mode

The LUT is bypassed and the color data is directly mapped for this color depth– See "Display Data Formats" on page 129.

12 SwivelView™

12.1 Concept

Most computer displays are refreshed in landscape orientation – 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 an counter-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.

The image is not actually rotated in the display buffer since there is no address translation during CPU read/write. The image is rotated during display refresh.

12.2 90° SwivelView™

90° SwivelView™ requires the Memory Clock (MCLK) to be at least 1.25 times the frequency of the Pixel Clock (PCLK), i.e. MCLK ≥ 1.25PCLK.

The following figure shows how the programmer sees a 320x480 portrait image and how the image is being displayed. The application image is written to the S1D13706 in the following sense: A–B–C–D. The display is refreshed by the S1D13706 in the following sense: B-D-A-C.

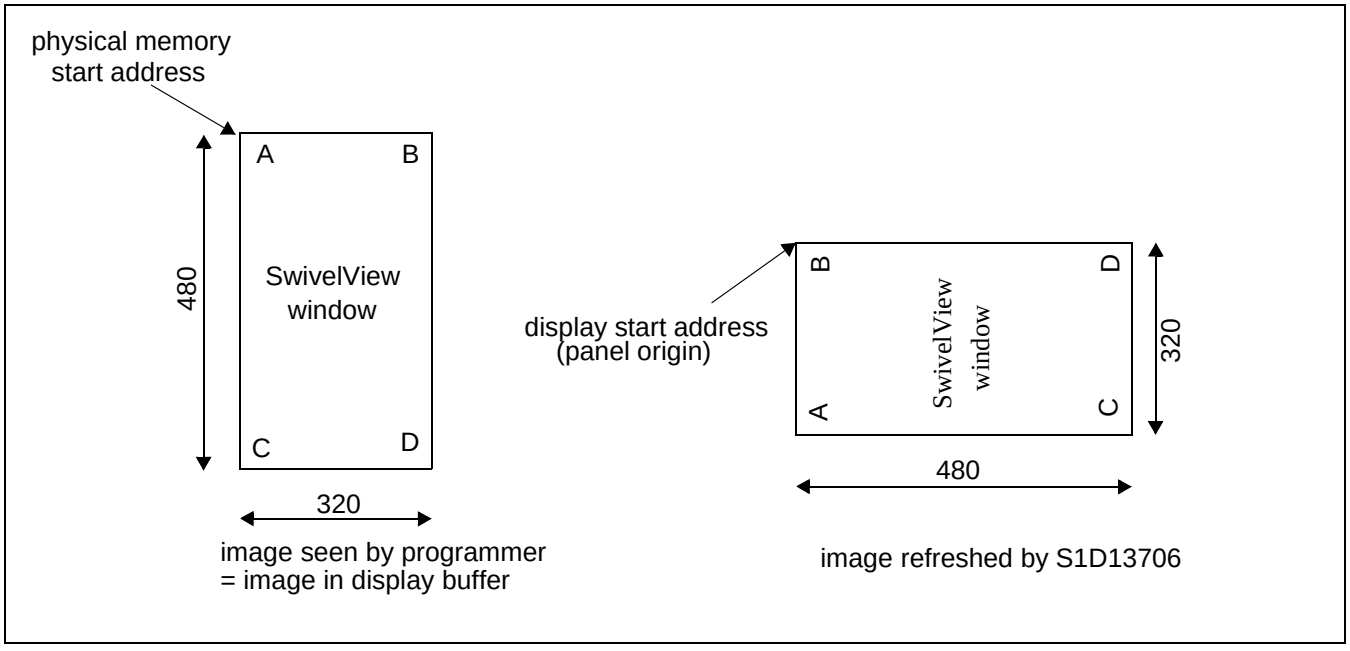


Figure 12-1: Relationship Between The Screen Image and the Image Refreshed in 90° SwivelView.

12.2.1 Register Programming

Enable 90° SwivelView™ Mode

Set SwivelView™ Mode Select bits (REG[71h] bits 1:0) to 01.

Display Start Address

The display refresh circuitry starts at pixel “B”, therefore the Main Window Display Start Address registers (REG[74h], REG[75h], REG[76h]) must be programmed with the address of pixel “B”. To calculate the value of the address of pixel “B” use the following formula (assumes 8 bpp color depth).

Main Window Display Start Address bits 16:0
= ((image address + (panel height x bpp ÷ 8)) ÷ 4) - 1
= ((0 + (320 pixels x 8 bpp ÷ 8)) ÷ 4) -1
= 79 (4Fh)

Line Address Offset

The Main Window Line Address Offset registers (REG[78h], REG[79h]) is based on the display width and programmed using the following formula.

Main Window Line Address Offset bits 9:0
= display width in pixels ÷ (32 ÷ bpp)
= 320 pixels ÷ 32 ÷ 8 bpp
= 80 (50h)

12.3 180° SwivelView™

The following figure shows how the programmer sees a 480x320 landscape image and how the image is being displayed. The application image is written to the S1D13706 in the following sense: A–B–C–D. The display is refreshed by the S1D13706 in the following sense: D-C-B-A.

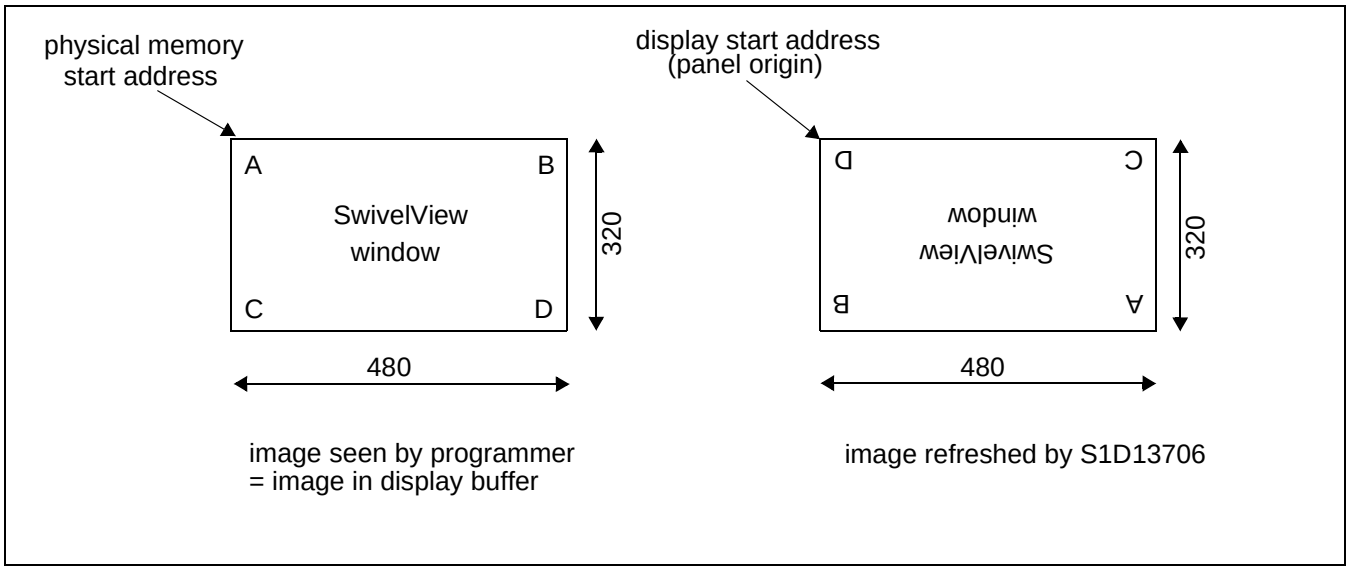


Figure 12-2: Relationship Between The Screen Image and the Image Refreshed in 180° SwivelView.

12.3.1 Register Programming

Enable 180° SwivelView™ Mode

Set SwivelView™ Mode Select bits (REG[71h] bits 1:0) to 10.

Display Start Address

The display refresh circuitry starts at pixel “D”, therefore the Main Window Display Start Address registers (REG[74h], REG[75h], REG[76h]) must be programmed with the address of pixel “D”. To calculate the value of the address of pixel “D” use the following formula (assumes 8 bpp color depth).

Main Window Display Start Address bits 16:0
= ((image address + (offset x (panel height - 1) + panel width) x bpp ÷ 8) ÷ 4) - 1
= ((0 + (480 pixels x 319 pixels + 480 pixels) x 8 bpp ÷ 8) ÷ 4) - 1
= 38399 (95FFh)

Line Address Offset

The Main Window Line Address Offset registers (REG[78h], REG[79h]) is based on the display width and programmed using the following formula.

Main Window Line Address Offset bits 9:0
= display width in pixels ÷ (32 ÷ bpp)
= 480 pixels ÷ 32 ÷ 8 bpp
= 120 (78h)

12.4 270° SwivelView™

270° SwivelView™ requires the Memory Clock (MCLK) to be at least 1.25 times the frequency of the Pixel Clock (PCLK), i.e. MCLK ≥ 1.25PCLK.

The following figure shows how the programmer sees a 320x480 portrait image and how the image is being displayed. The application image is written to the S1D13706 in the following sense: A–B–C–D. The display is refreshed by the S1D13706 in the following sense: C-A-D-B.

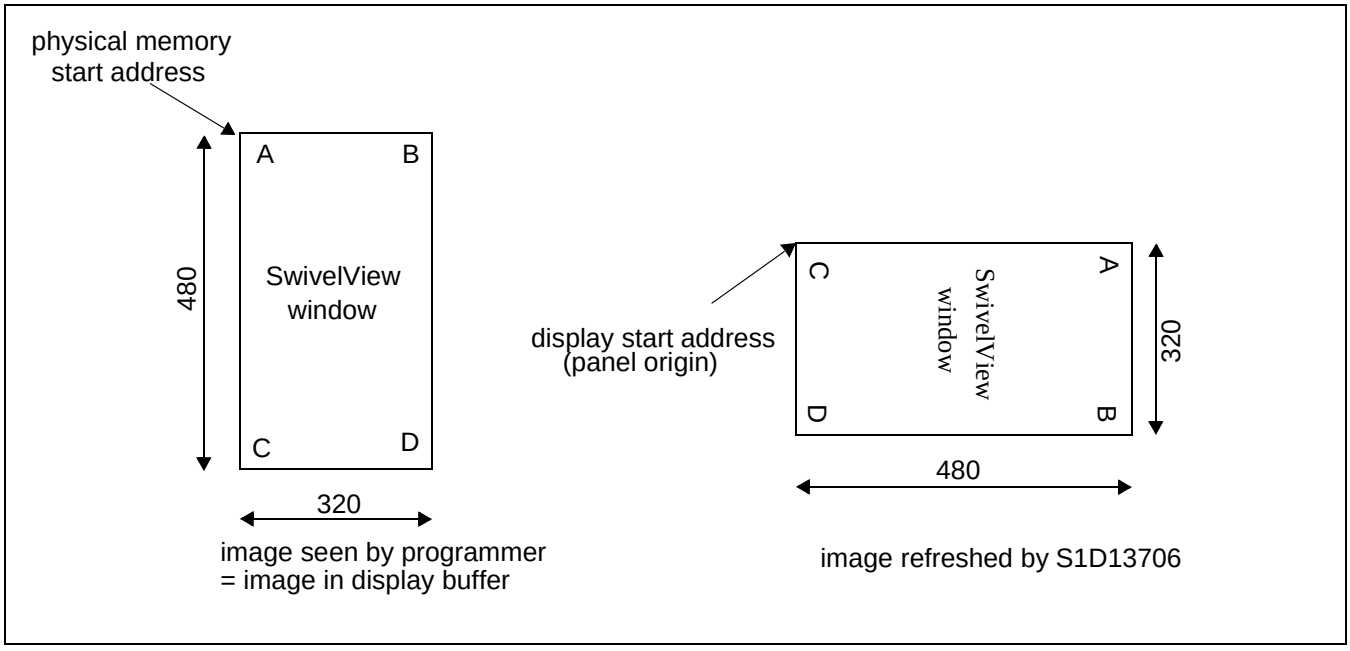


Figure 12-3: Relationship Between The Screen Image and the Image Refreshed in 270° SwivelView.

12.4.1 Register Programming

Enable 270° SwivelView™ Mode

Set SwivelView™ Mode Select bits (REG[71h] bits 1:0) to 11.

The display refresh circuitry starts at pixel “C”, therefore the Main Window Display Start Address registers (REG[74h], REG[75h], REG[76h]) must be programmed with the address of pixel “C”. To calculate the value of the address of pixel “C” use the following formula (assumes 8 bpp color depth).

Main Window Display Start Address bits 16:0
= (image address + ((panel width - 1) x offset x bpp ÷ 8) ÷ 4)
= (0 + ((480 pixels - 1) x 320 pixels x 8 bpp ÷ 8) ÷ 4)
= 38320 (95B0h)

Line Address Offset

The Main Window Line Address Offset registers (REG[78h], REG[79h]) is based on the display width and programmed using the following formula.

Main Window Line Address Offset bits 9:0
= display width in pixels ÷ (32 ÷ bpp)
= 320 pixels ÷ 32 ÷ 8 bpp
= 80 (50h)

13 Picture-in-Picture Plus (PIP+)

13.1 Concept

Picture-in-Picture Plus enables a secondary window (or PIP+ window) within the main display window. The PIP+ window may be positioned anywhere within the virtual display and is controlled through the PIP+ window control registers (REG[7Ch] through REG[91h]). The PIP+ window retains the same color depth and SwivelView orientation as the main window.

The following diagram shows an example of a PIP+ window within a main window and the registers used to position it.

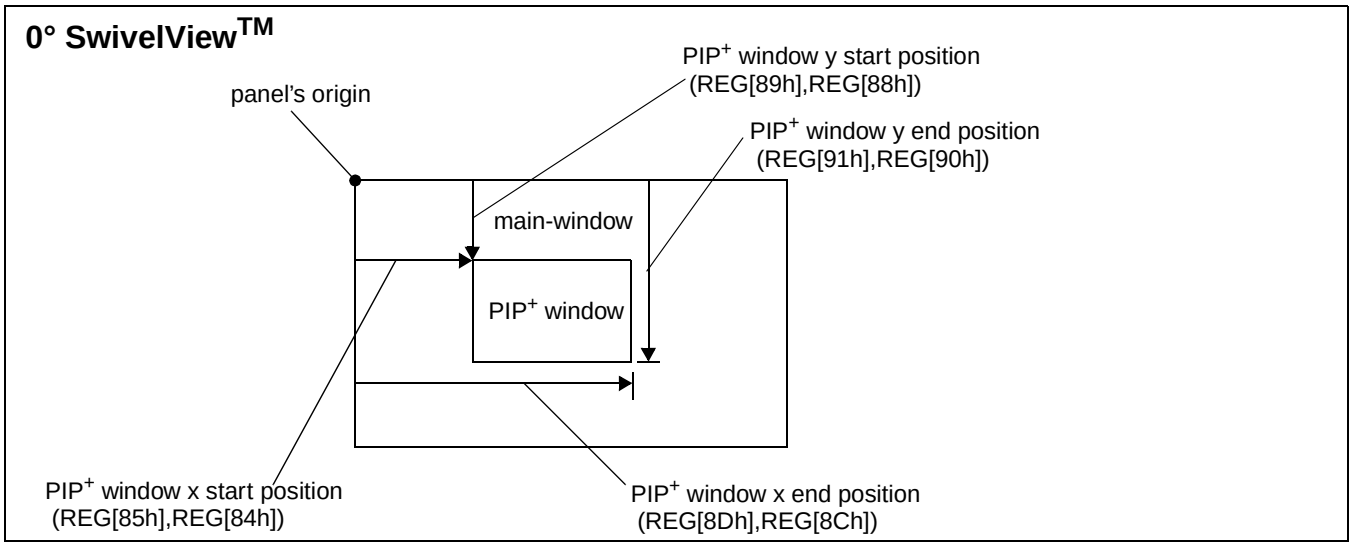


Figure 13-1: Picture-in-Picture Plus with SwivelView disabled

13.2 With SwivelView Enabled

13.2.1 SwivelView 90°

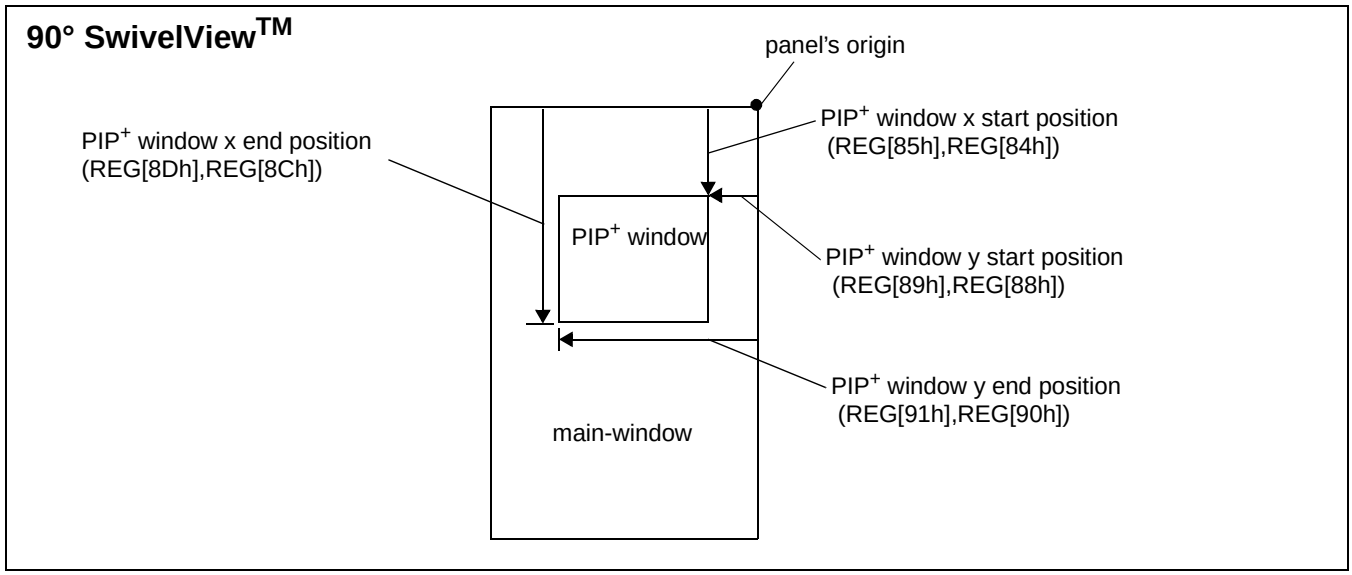


Figure 13-2: Picture-in-Picture Plus with SwivelView 90° enabled

13.2.2 SwivelView 180°

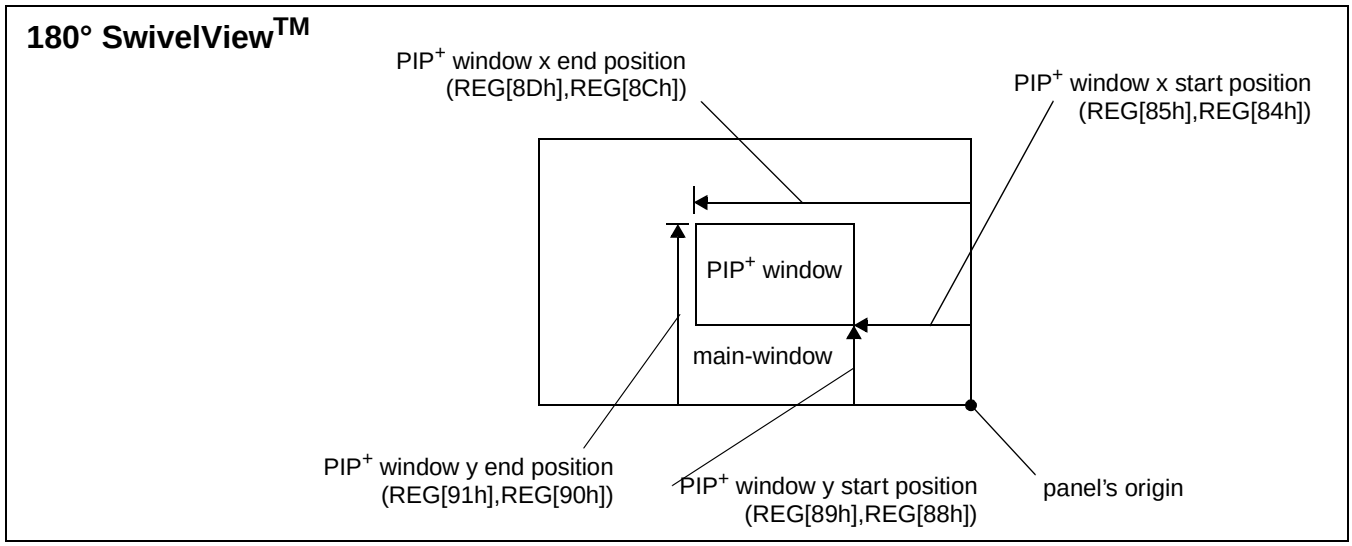


Figure 13-3: Picture-in-Picture Plus with SwivelView 180° enabled

13.2.3 SwivelView 270°

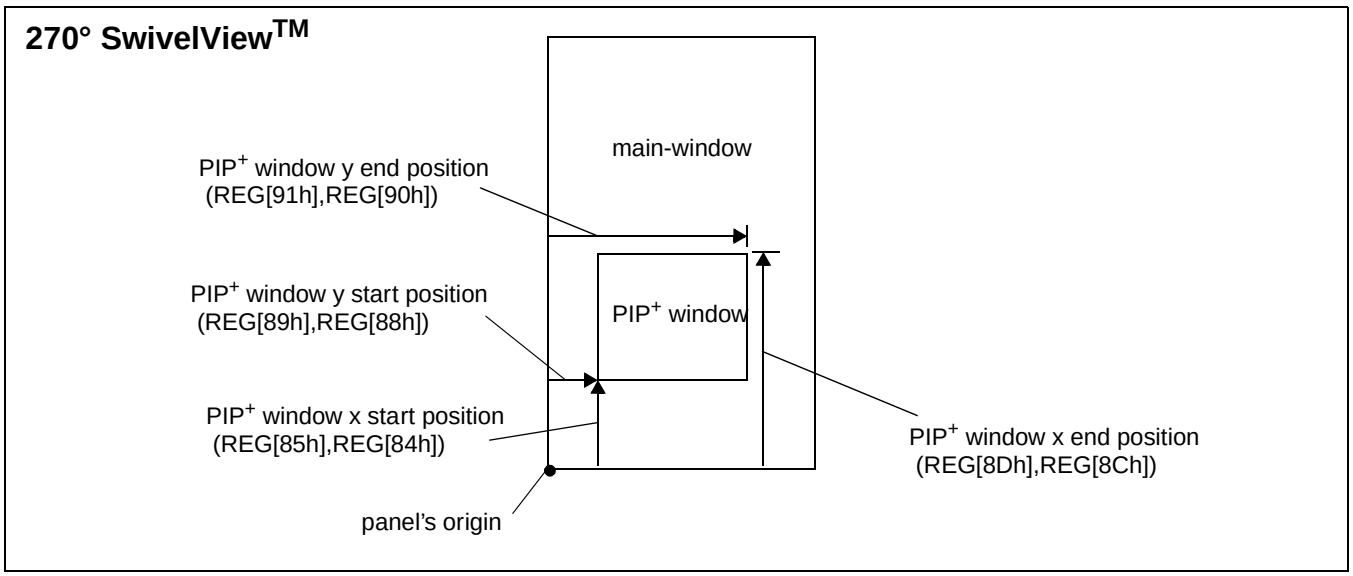


Figure 13-4: Picture-in-Picture Plus with SwivelView 270° enabled

14 Big-Endian Bus Interface

14.1 Byte Swapping Bus Data

The display buffer and register architecture of the S1D13706 is inherently little-endian. If configured as big-endian (CNF4 = 1 at reset), bus accesses are automatically handled by byte swapping all read/write data to/from the internal display buffer and registers.

Bus data byte swapping translates all byte accesses correctly to the S1D13706 register and display buffer locations. To maintain the correct translation for 16-bit word access, even address bytes must be mapped to the MSB of the 16-bit word, and odd address bytes to the LSB of the 16-bit word. For example:

Byte write 11h to register address 1Eh -> REG[1Eh] <= 11h
Byte write 22h to register address 1Fh -> REG[1Fh] <= 22h

Word write 1122h to register address 1Eh-> REG[1Eh] <= 11h
REG[1Fh] <= 22h

14.1.1 16 Bpp Color Depth

For 16 bpp color depth, the Display Data Byte Swap bit (REG[71h] bit 6) must be set to 1.

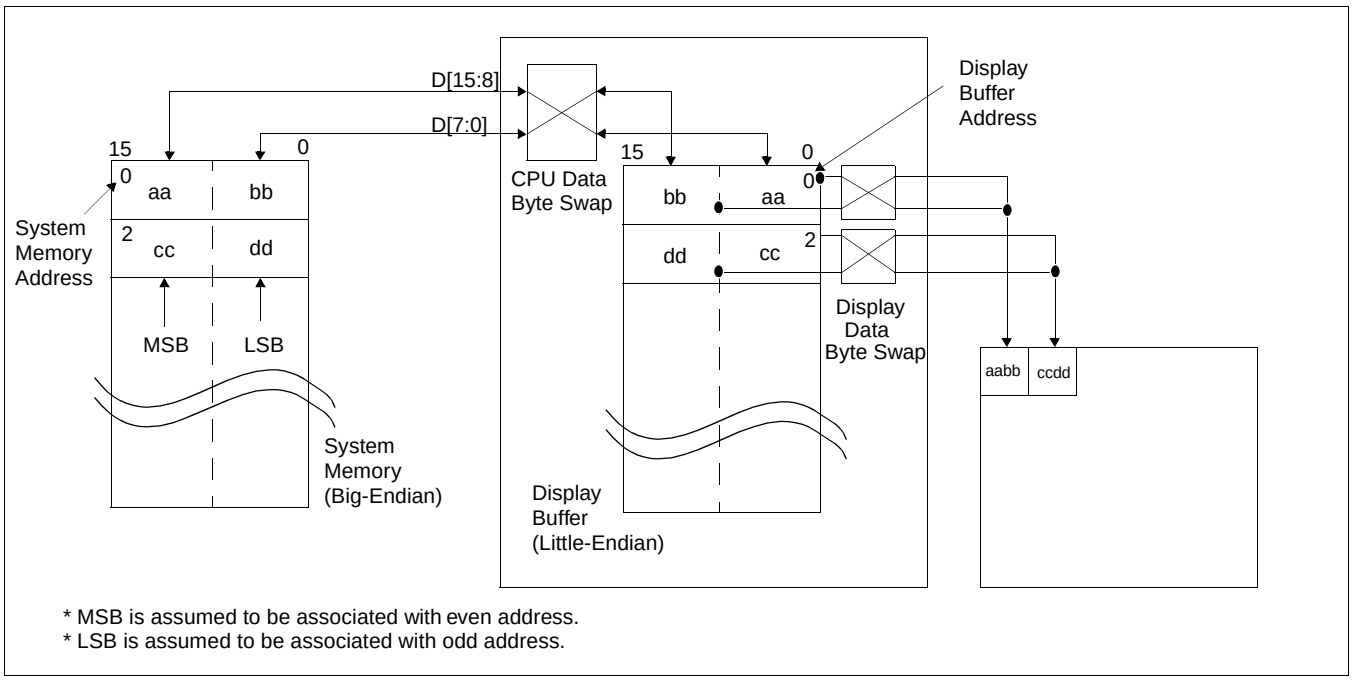


Figure 14-1: Byte-swapping for 16 Bpp

For 16 bpp color depth, the MSB of the 16-bit pixel data is stored at the even system memory address location and the LSB of the 16-bit pixel data is stored at the odd system memory address location. Bus data byte swapping (automatic when the S1D13706 is configured for Big-Endian) causes the 16-bit pixel data to be stored byte-swapped in the S1D13706 display buffer. During display refresh this stored data must be byte-swapped again before it is sent to the display.

14.1.2 1/2/4/8 Bpp Color Depth

For 1/2/4/8 bpp color depth, byte swapping must be performed on the bus data but not the display data.

For 1/2/4/8 bpp color depth, the Display Data Byte Swap bit (REG[71h] bit 6) must be set to 0.

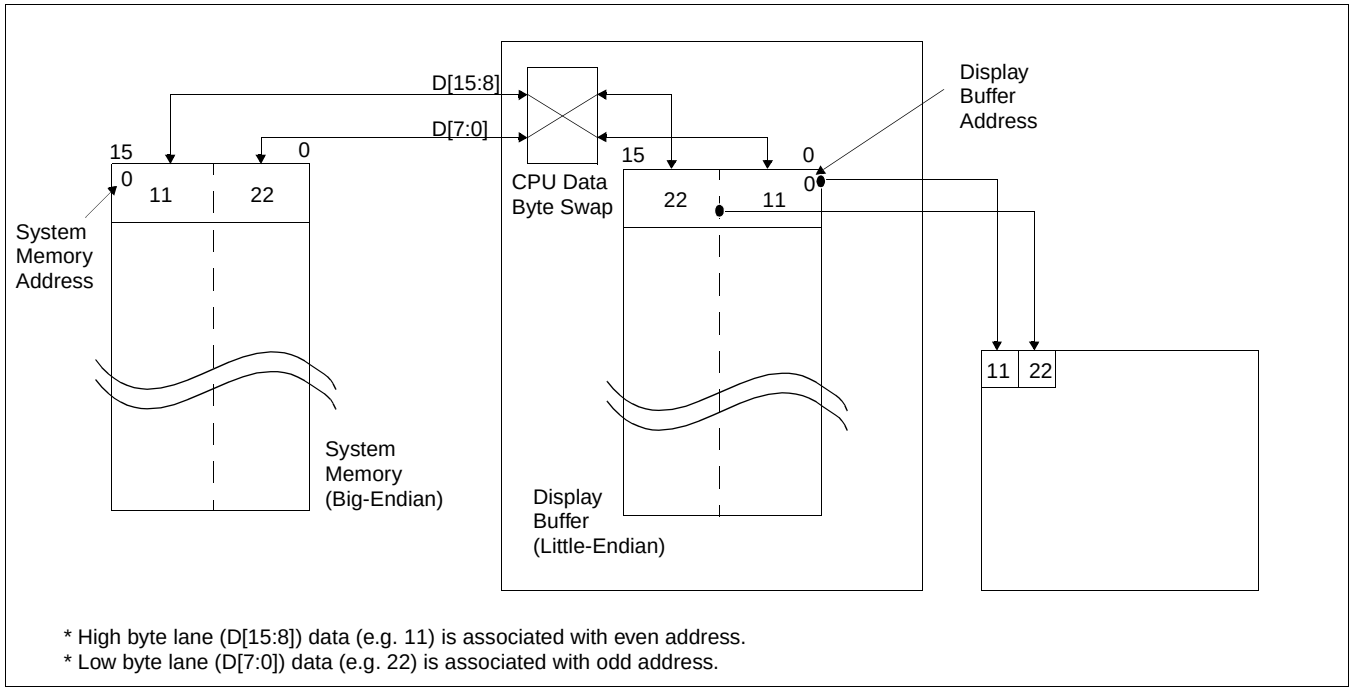


Figure 14-2: Byte-swapping for 1/2/4/8 Bpp

15 Power Save Mode

A software initiated Power Save Mode is incorporated into the S1D13706 to accommodate the need for power reduction in the hand-held devices market. This mode is enabled via the Power Save Mode Enable bit (REG[A0h] bit 0).

Software Power Save Mode saves power by powering down the panel and stopping display refresh accesses to the display buffer.

Table 15-1: Power Save Mode Function Summary

	Software Power Save	Normal
IO Access Possible?	Yes	Yes
Memory Access Possible?	No[1]	Yes
Look-Up Table Registers Access Possible?	Yes	Yes
Sequence Controller Running?	No	Yes
Display Active?	No	Yes
LCD I/F Outputs	Forced Low	Active
PWMCLK	Stopped	Active
GPIO Pins configured for HR-TFT/D-TFD[2]	Forced Low	Active
GPIO Pins configured as GPIOs Access Possible?[2]	Yes[3]	Yes

Note

[1] When power save mode is enabled, the memory controlled is powered down. The status of the memory controlled is indicated by the Memory Controller Power Save Status bit (REG[A0h] bit 3). For Power Save Status AC timing, see Section 6.3.3, “Power Save Status” on page 54.

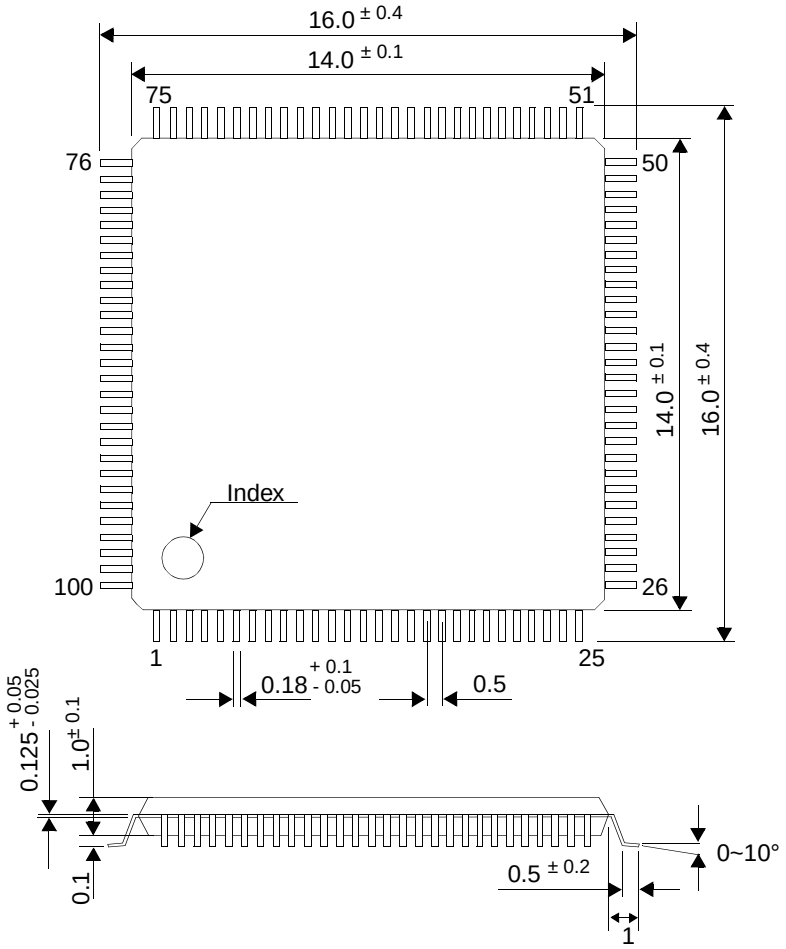
[2] GPIO Pins are configured using the configuration pin CNF3 which is latched on the rising edge of RESET#. For information on CNF3, see Table 4-7: “Summary of Power-On/Reset Options,” on page 27.

[3] GPIOs can be accessed and if configured as outputs can be changed.

After reset, the S1D13706 is always in Power Save Mode. Software must initialize the chip (i.e. programs all registers) and then clear the Power Save Mode Enable bit.

16 Mechanical Data

100-pin TQFP15 surface mount package



All dimensions in mm

Figure 16-1: Mechanical Data 100pin TQFP15 (S1D13706F00A)

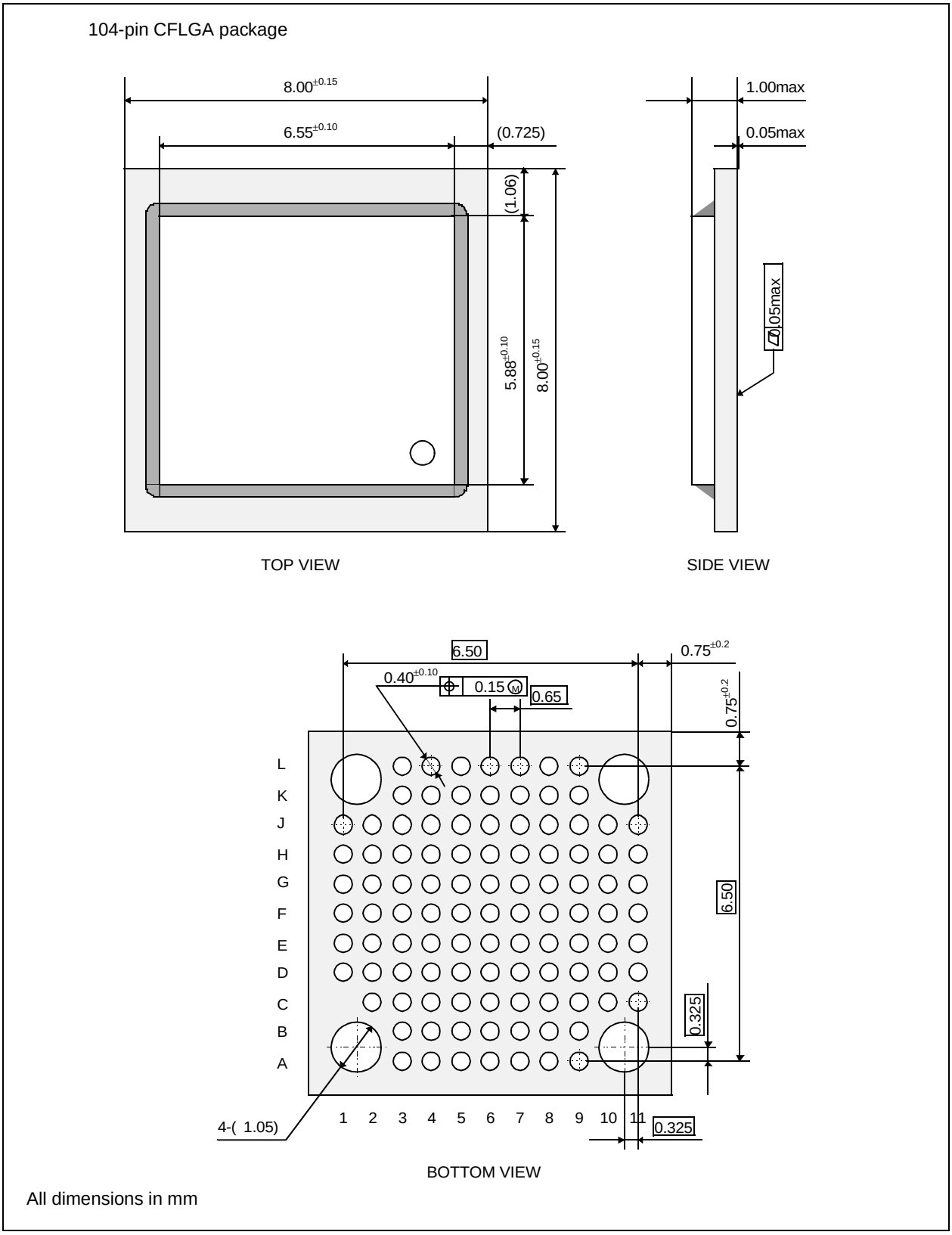


Figure 16-2: Mechanical Data 104pin CFLGA (S1D13706B00A)

17 References

The following documents contain additional information related to the S1D13706. Document numbers are listed in parenthesis after the document name. All documents can be found at the Epson Electronics America website at **www.eea.epson.com** or the Epson Research and Development Website at **www.erd.epson.com**.

- 13706CFG Configuration Utility Users Manual (X31B-B-001-xx)
- 13706SHOW Demonstration Program Users Manual (X31B-B-002-xx)
- 13706PLAY Diagnostic Utility Users Manual (X31B-B-003-xx)
- 13706BMP Demonstration Program Users Manual (X31B-B-004-xx)
- S1D13706 Product Brief (X31B-C-001-xx)
- S1D13706 Windows CE Display Drivers (X31B-E-001-xx)
- Interfacing to the Toshiba TMPR3905/3912 Microprocessor (X31B-G-002-xx)
- S1D13706 Programming Notes And Examples (X31B-G-003-xx)
- S5U13706B00C Rev. 1.0 Evaluation Board User Manual (X31B-G-004-xx)
- Interfacing to the PC Card Bus (X31B-G-005-xx)
- S1D13706 Power Consumption (X31B-G-006-xx)
- Interfacing to the NEC VR4102/VR4111 Microprocessors (X31B-G-007-xx)
- Interfacing to the NEC VR4181 Microprocessor (X31B-G-008-xx)
- Interfacing to the Motorola MPC821 Microprocessor (X31B-G-009-xx)
- Interfacing to the Motorola MCF5307 "Coldfire" Microprocessors (X31B-G-010-xx)
- Connecting to the Sharp HR-TFT Panels (X31B-G-011-xx)
- Connecting to the Epson D-TFD Panels (X31B-G-012-xx)
- Interfacing to the Motorola MC68030 Microprocessor (X31B-G-013-xx)
- Interfacing to the Motorola RedCap2 DSP with Integrated MCU (X31B-G-014-xx)
- Interfacing to 8-Bit Processors (X31B-G-015-xx)
- Interfacing to the Motorola MC68VZ328 Dragonball Microprocessor (X31B-G-016-xx)
- S1D13706 Register Summary (X31B-R-001-xx)

18 Technical Support

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Programming Notes and Examples

Document Number: X31B-G-003-03

THIS PAGE LEFT BLANK

Table of Contents

List of Tables

THIS PAGE LEFT BLANK

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This guide provides information on programming the S1D13706 Embedded Memory LCD Controller. Included are algorithms which demonstrate how to program the S1D13706. This guide discusses Power-on Initialization, Panning and Scrolling, LUT initialization, LCD Power Sequencing, SwivelView™, Picture-In-Picture Plus, etc. The example source code referenced in this guide is available on the web at www.eea.epson.com or www.erd.epson.com.

This guide also introduces the Hardware Abstraction Layer (HAL), which is designed to simplify the programming of the S1D13706. Most SED135x and SED137x products have HAL support, thus allowing OEMs to do multiple designs with a common code base.

This document will be updated as appropriate. Please check the Epson Electronics America Website at www.eea.epson.com for the latest revision of this document and source before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Initialization

This section describes how to initialize the S1D13706. Sample code for performing initialization of the S1D13706 is provided in the file **init13706.c** which is available on the internet at www.eea.epson.com or www.erd.epson.com.

S1D13706 initialization can be broken into the following steps.

1. Disable the display using the Display Blank bit (set REG[70h] bit 7 = 1).
2. If the system implementation uses a clock chip instead of a fixed oscillator, program the clock chip. For example, the S5U13706 Evaluation Board uses a Cypress clock chip.
3. Set all registers to initial values. Table 2-1:, “Example Register Values” contains the correct values for an example panel discussed below.
4. Program the Look-Up Table (LUT) with color values. For details on programming the LUT, see Section 4, “Look-Up Table (LUT)” on page 17.
5. Power-up the LCD panel. For details on powering-up the LCD panel, see Section 5.4, “Disabling Power Save Mode” on page 28.
6. Enable the display using the Display Blank bit (set REG[70h] bit 7 = 0).
7. Clear the display buffer (if required).

Note

The simplest way to generate initialization tables for the S1D13706 is to use the utility program 13706CFG.EXE which generates a header file that can be used by the operating system or the HAL. Otherwise modify the **init13706.c** file directly.

The following table represents the sequence and values written to the S1D13706 registers to control a configuration with these specifications.

- 320x240 color single passive LCD @ 70Hz.
- 8-bit data interface, format 2.
- 8 bit-per-pixel (bpp) color depth - 256 colors.
- 50MHz input clock for CLKI.
- MCLK = BCLK = CLKI = 50MHz.
- PCLK = CLKI ÷ 8 = 6.25MHz.

Note
On the S5U13706B00C evaluation board, CNF[7:6] must be set to 00.

Table 2-1: Example Register Values

Register	Value (Hex)	Value (Binary)	Description	Notes
Clock Configuration (MCLK, BCLK, PCLK)				
04h	00	0000 0000	Sets BCLK to MCLK divide to 1:1	
05h	43	0100 0011	Sets PCLK = (PCLK source ÷ 8) and the PCLK source = CLKI2	
Panel Setting Configuration				
10h	D0	1101 0000	Selects the following: • panel data format = 2 • color/mono panel = color • panel data width = 8-bit • active panel resolution = don't care • panel type = STN	
11h	00	0000 0000	MOD rate = don't care	
12h	2B	0010 1011	Sets the horizontal total	
14h	27	0010 0111	Sets the horizontal display period	
16h 17h	00 00	0000 0000 0000 0000	Sets the horizontal display period start position	
18h 19h	FA 00	1111 1010 0000 0000	Sets the vertical total	
1Ch 1Dh	EF 00	1110 1111 0000 0000	Sets the vertical display period	
1Eh 1Fh	00 00	0000 0000 0000 0000	Sets the vertical display period start position	
20h	87	1000 0111	Sets the FPLINE pulse polarity and FPLINE pulse width	
22h 23h	00 00	0000 0000 0000 0000	Sets the FPLINE pulse start position	
24h	80	1000 0000	Sets the FPFRAME pulse polarity and FPFRAME pulse width	
26h 27h	01 00	0000 0001 0000 0000	Sets the FPFRAME pulse start position	

Table 2-1: Example Register Values (Continued)

Register	Value (Hex)	Value (Binary)	Description	Notes
Display Mode Setting Configuration				
70h	83	1000 0011	Selects the following: • display blank = screen is blanked • dithering = enabled • hardware video invert = disabled • software video invert = video data is not inverted • color depth = 8 bpp	
71h	00	0000 0000	Selects the following: • display data word swap = disabled • display data byte swap = disabled • sub-window enable = disabled • SwivelView Mode = not rotated	
74h 75h 76h	00 00 00	0000 0000 0000 0000 0000 0000	Sets the main window display start address	
78h 79h	50 00	0101 0000 0000 0000	Sets the main window line address offset	
7Ch 7Dh 7Eh	00 00 00	0000 0000 0000 0000 0000 0000	Sets the sub-window display start address	
80h 81h	50 00	0101 0000 0000 0000	Sets the sub-window line address offset	
84h 85h	00 00	0000 0000 0000 0000	Sets the sub-window X start position	
88h 89h	00 00	0000 0000 0000 0000	Sets the sub-window Y start position	
8Ch 8Dh	4F 00	0100 1111 0000 0000	Sets the sub-window X end position	
90h 91h	EF 00	1110 1111 0000 0000	Sets the sub-window Y end position	
Miscellaneous Register Configuration				
A0h	00	0000 0000	Disables power save mode	
A1h	00	0000 0000	Reserved register. Must be written 00h.	
A2h	00	0000 0000	Set reserved bit 7 to 0	
A3h	00	0000 0000	Reserved register. Must be written 00h.	
A4h A5h	00 00	0000 0000 0000 0000	Clears the scratch pad registers	
GPIO Pin Configuration				
A8h	00	0000 0000	GPIO[6:0] pins are configured as input pins	
A9h	80	1000 0000	Bit 7 set to 1 to enable GPIO pin inputs.	

Table 2-1: Example Register Values (Continued)

Register	Value (Hex)	Value (Binary)	Description	Notes
ACh	00	0000 0000	GPIO[6:0] pins are driven low	
ADh	00	0000 0000	Set the GPO control bit to low	Bit 7 controls the LCD bias power for the panel on the S5U13706B00C.
PWM Clock and CV Pulse Configuration				
B0h	00	0000 0000	Selects the following: • PWMOUT pin is software controlled • PWM Clock circuitry is disabled • CVOUT pin is software controlled • CV Pulse circuitry is disabled	
B1h	00	0000 0000	Sets the PWM Clock and CV Pulse divides	For this example the divides are not required.
B2h	00	0000 0000	Sets the CV Pulse Burst Length	For this example, the burst length is not required.
B3h	00	0000 0000	Sets the PWMOUT signal to always low	

3 Memory Models

The S1D13706 contains a display buffer of 80K bytes and supports color depths of 1, 2, 4, 8, and 16 bit-per-pixel. For each color depth, the data format is packed pixel.

Packed pixel data may be envisioned as a stream of pixels. In this stream, pixels are packed adjacent to each other. If a pixel requires four bits, then it is located in the four most significant bits of a byte. The pixel to the immediate right on the display occupies the lower four bits of the same byte. The next two pixels to the immediate right are located in the following byte, etc.

3.1 Display Buffer Location

The S1D13706 display buffer is 80K bytes of embedded SRAM. The display buffer is memory mapped and is accessible directly by software. The memory block location assigned to the S1D13706 display buffer varies with each individual hardware platform.

For further information on the display buffer, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

For further information on the S1D13706 Evaluation Board, see the *S5U13706B00C Evaluation Board Rev. 1.0 User Manual*, document number X31B-G-004-xx.

3.2 Memory Organization for One Bit-per-pixel (2 Colors/Gray Shades)

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

Figure 3-1: Pixel Storage for 1 Bpp in One Byte of Display Buffer

At a color depth of 1 bpp, each byte of display buffer contains eight adjacent pixels. Setting or resetting any pixel requires reading the entire byte, masking out the unchanged bits and setting the appropriate bits to 1.

One bit pixels provide 2 gray shades/color possibilities. For monochrome panels the gray shades are generated by indexing into the first two elements of the green component of the Look-Up Table (LUT). For color panels the 2 colors are derived by indexing into the first 2 positions of the LUT.

3.3 Memory Organization for Two Bit-per-pixel (4 Colors/Gray Shades)

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

Figure 3-2: Pixel Storage for 2 Bpp in One Byte of Display Buffer

At a color depth of 2 bpp, each byte of display buffer contains four adjacent pixels. Setting or resetting any pixel requires reading the entire byte, masking out the unchanged bits and setting the appropriate bits to 1.

Two bit pixels provide 4 gray shades/color possibilities. For monochrome panels the gray shades are generated by indexing into the first 4 elements of the green component of the Look-Up Table (LUT). For color panels the 4 colors are derived by indexing into the first 4 positions of the LUT.

3.4 Memory Organization for Four Bit-per-pixel (16 Colors/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 3-3: Pixel Storage for 4 Bpp in One Byte of Display Buffer

At a color depth of 4 bpp, each byte of display buffer contains two adjacent pixels. Setting or resetting any pixel requires 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.

3.5 Memory Organization for 8 Bpp (256 Colors/64 Gray Shades)

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

Figure 3-4: Pixel Storage for 8 Bpp in One Byte of Display Buffer

At a color depth of 8 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 S1D13706 LUT supports six bits per primary color. This translates into 256K possible colors when color mode is selected. Therefore the displayed mode has 256 colors available out of a possible 256K colors.

When a monochrome panel is selected, the green component of the LUT is used to determine the gray shade intensity. The green indices, with six bits, can resolve 64 gray shades.

3.6 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 3-5: Pixel Storage for 16 Bpp in Two Bytes of Display Buffer

At a color depth of 16 bpp the S1D13706 is capable of displaying 64K (65536) colors. The 64K 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.

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. Note that 8 bpp also provides 64 gray shades using less memory.

4 Look-Up Table (LUT)

This section discusses programming the S1D13706 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 *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The S1D13706 is designed with a LUT consisting of 256 indexed red/green/blue entries. Each LUT entry is six bits wide. The color depth (bpp) determines how many indices are used to output the image to the display. For example, 1 bpp uses the first 2 indices, 2 bpp uses the first 4 indices, 4 bpp uses the first 16 indices and 8 bpp uses all 256 indices. Note that 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[10h] bit 6).

4.1 Registers

4.1.1 Look-Up Table Write Registers

REG[08h] Look-Up Table Blue Write Data Register							
LUT Blue Write Data Bit 5	LUT Blue Write Data Bit 4	LUT Blue Write Data Bit 3	LUT Blue Write Data Bit 2	LUT Blue Write Data Bit 1	LUT Blue Write Data Bit 0	n/a	n/a

REG[09h] Look-Up Table Green Write Data Register							
LUT Green Write Data Bit 5	LUT Green Write Data Bit 4	LUT Green Write Data Bit 3	LUT Green Write Data Bit 2	LUT Green Write Data Bit 1	LUT Green Write Data Bit 0	n/a	n/a

REG[0Ah] Look-Up Table Red Write Data Register							
LUT Red Write Data Bit 5	LUT Red Write Data Bit 4	LUT Red Write Data Bit 3	LUT Red Write Data Bit 2	LUT Red Write Data Bit 1	LUT Red Write Data Bit 0	n/a	n/a

These registers contain the data to be written to the blue/green/red components of the Look-Up Table. The data is stored in these registers until a write to the LUT Write Address Register (REG[0Bh]) moves the data to the Look-Up Table.

Note

The LUT entries are updated only when the LUT Write Address Register (REG[0Bh]) is written to.

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

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

Note

For further information on the S1D13706 LUT architecture, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

4.1.2 Look-Up Table Read Registers

REG[0Ch] Look-Up Table Blue Read Data Register							
LUT Blue Read Data Bit 5	LUT Blue Read Data Bit 4	LUT Blue Read Data Bit 3	LUT Blue Read Data Bit 2	LUT Blue Read Data Bit 1	LUT Blue Read Data Bit 0	n/a	n/a

REG[0Dh] Look-Up Table Green Read Data Register							
LUT Green Read Data Bit 5	LUT Green Read Data Bit 4	LUT Green Read Data Bit 3	LUT Green Read Data Bit 2	LUT Green Read Data Bit 1	LUT Green Read Data Bit 0	n/a	n/a

REG[0Eh] Look-Up Table Red Read Data Register							
LUT Red Read Data Bit 5	LUT Red Read Data Bit 4	LUT Red Read Data Bit 3	LUT Red Read Data Bit 2	LUT Red Read Data Bit 1	LUT Red Read Data Bit 0	n/a	n/a

These registers contains the data returned from the blue/green/red components of the Look-Up Table. The data is read and placed in these registers only when a write to the LUT Write Address Register (REG[0Fh]) copies the data from the Look-Up Table.

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

This register forms a pointer into the Look-Up Table (LUT) which is used to read LUT data to REG[0Ch], REG[0Dh], and REG[0Eh]. The data is placed in REG[0Ch], REG[0Dh], and REG[0Eh] only with the completion of a write to this register. This is a write-only register and returns 00h if read.

Note

For further information on the S1D13706 LUT architecture, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

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 S1D13706 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

Color Depth	Look-Up Table Indices Used			Effective Gray Shades/Colors
	RED	GREEN	BLUE	
1 bpp gray		2		2 gray shades
2 bpp gray		4		4 gray shades
4 bpp gray		16		16 gray shades
8 bpp gray		16		64 gray shades
16 bpp gray				64 gray shades
1 bpp color	2	2	2	2 colors
2 bpp color	4	4	4	4 colors
4 bpp color	16	16	16	16 colors
8 bpp color	256	256	256	256 colors
16 bpp color				65536 colors

= Indicates the Look-Up Table is not used for that display mode

4.2.1 Gray Shade Modes

Gray shade (monochrome) modes are defined by the Color/Mono Panel Select bit (REG[10h] bit 6). When this bit is set to 0, the value output to the panel is derived solely from the green component of the LUT.

1 bpp gray shade

The 1 bpp gray shade mode uses the green component of the first 2 LUT entries. The remaining indices of the LUT are unused.

Table 4-2: Suggested LUT Values for 1 Bpp Gray Shade

Index	Red	Green	Blue
00	00	00	00
01	00	FC	00
02	00	00	00
...	00	00	00
FF	00	00	00

Unused entries

2 bpp gray shade

The 2 bpp gray shade mode uses the green component of the first 4 LUT entries. The remaining indices of the LUT are unused.

Table 4-3: Suggested LUT Values for 4 Bpp Gray Shade

Index	Red	Green	Blue
00	00	00	00
01	00	54	00
02	00	A8	00
03	00	FC	00
04	00	00	00
...	00	00	00
FF	00	00	00

Unused entries

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	00	10	00
02	00	20	00
03	00	30	00
04	00	44	00
05	00	54	00
06	00	64	00
07	00	74	00
08	00	88	00
09	00	98	00
0A	00	A8	00
0B	00	B8	00
0C	00	CC	00
0D	00	DC	00
0E	00	EC	00
0F	00	FC	00
10	00	00	00
...	00	00	00
FF	00	00	00

Unused entries

8 bpp gray shade

When configured for 8 bpp gray shade mode, the green component of all 256 LUT entries may be used. However, the green component alone only provides 64 intensities (6 bits).

16 bpp gray shade

The Look-Up Table is bypassed at this color depth, therefore 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.

4.2.2 Color Modes

In color display modes, the number of LUT entries used is automatically selected depending on the color depth.

1 bpp color

When the S1D13706 is configured for 1 bpp color mode the first 2 entries in the LUT are used. Each byte in the display buffer contains eight adjacent pixels.

Table 4-5: Suggested LUT Values for 1 bpp Color

Index	Red	Green	Blue
00	00	00	00
01	FC	FC	FC
02	00	00	00
...	00	00	00
FF	00	00	00

= Indicates unused entries in the LUT

2 bpp color

When the S1D13706 is configured for 2 bpp color mode the first 4 entries in the LUT are used. Each byte in the display buffer contains four adjacent pixels.

Table 4-6: Suggested LUT Values for 2 bpp Color

Index	Red	Green	Blue
00	00	00	00
01	00	00	FF
02	FF	00	00
03	FC	FC	FC
04	00	00	00
...	00	00	00
FF	00	00	00

= Indicates unused entries in the LUT

4 bpp color

When the S1D13706 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 simulate those of a VGA operating in 16 color mode.

Table 4-7: Suggested LUT Values to Simulate VGA Default 16 Color Palette

Index	Red	Green	Blue
00	00	00	00
01	80	00	00
02	00	80	00
03	80	80	00
04	00	00	80
05	80	00	80
06	00	80	80
07	C0	C0	C0
08	80	80	80
09	FC	00	00
0A	00	FC	00
0B	FC	FC	00
0C	00	00	FC
0D	FC	00	FC
0E	00	FC	FC
0F	FC	FC	FC
10	00	00	00
...	00	00	00
FF	00	00	00

= Indicates unused entries in the LUT

8 bpp color

When the S1D13706 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 S1D13706 LUT has six bits (64 intensities) of intensity control per primary color which is the same as a standard VGA RAMDAC.

The following table shows LUT values that simulate the VGA default color palette.

Table 4-8: Suggested LUT Values to Simulate VGA Default 256 Color Palette

Index	R	G	B	Index	R	G	B	Index	R	G	B	Index	R	G	B
00	00	00	00	40	F0	70	70	80	30	30	70	C0	00	40	00
01	00	00	A0	41	F0	90	70	81	40	30	70	C1	00	40	10
02	00	A0	00	42	F0	B0	70	82	50	30	70	C2	00	40	20
03	00	A0	A0	43	F0	D0	70	83	60	30	70	C3	00	40	30
04	A0	00	00	44	F0	F0	70	84	70	30	70	C4	00	40	40
05	A0	00	A0	45	D0	F0	70	85	70	30	60	C5	00	30	40
06	A0	50	00	46	B0	F0	70	86	70	30	50	C6	00	20	40
07	A0	A0	A0	47	90	F0	70	87	70	30	40	C7	00	10	40
08	50	50	50	48	70	F0	70	88	70	30	30	C8	20	20	40
09	50	50	F0	49	70	F0	90	89	70	40	30	C9	20	20	40
0A	50	F0	50	4A	70	F0	B0	8A	70	50	30	CA	30	20	40
0B	50	F0	F0	4B	70	F0	D0	8B	70	60	30	CB	30	20	40
0C	F0	50	50	4C	70	F0	F0	8C	70	70	30	CC	40	20	40
0D	F0	50	F0	4D	70	D0	F0	8D	60	70	30	CD	40	20	30
0E	F0	F0	50	4E	70	B0	F0	8E	50	70	30	CE	40	20	30
0F	F0	F0	F0	4F	70	90	F0	8F	40	70	30	CF	40	20	20
10	00	00	00	50	B0	B0	F0	90	30	70	30	D0	40	20	20
11	10	10	10	51	C0	B0	F0	91	30	70	40	D1	40	20	20
12	20	20	20	52	D0	B0	F0	92	30	70	50	D2	40	30	20
13	20	20	20	53	E0	B0	F0	93	30	70	60	D3	40	30	20
14	30	30	30	54	F0	B0	F0	94	30	70	70	D4	40	40	20
15	40	40	40	55	F0	B0	E0	95	30	60	70	D5	30	40	20
16	50	50	50	56	F0	B0	D0	96	30	50	70	D6	30	40	20
17	60	60	60	57	F0	B0	C0	97	30	40	70	D7	20	40	20
18	70	70	70	58	F0	B0	B0	98	50	50	70	D8	20	40	20
19	80	80	80	59	F0	C0	B0	99	50	50	70	D9	20	40	20
1A	90	90	90	5A	F0	D0	B0	9A	60	50	70	DA	20	40	30
1B	A0	A0	A0	5B	F0	E0	B0	9B	60	50	70	DB	20	40	30
1C	B0	B0	B0	5C	F0	F0	B0	9C	70	50	70	DC	20	40	40
1D	C0	C0	C0	5D	E0	F0	B0	9D	70	50	60	DD	20	30	40
1E	E0	E0	E0	5E	D0	F0	B0	9E	70	50	60	DE	20	30	40
1F	F0	F0	F0	5F	C0	F0	B0	9F	70	50	50	DF	20	20	40
20	00	00	F0	60	B0	F0	B0	A0	70	50	50	E0	20	20	40
21	40	00	F0	61	B0	F0	C0	A1	70	50	50	E1	30	20	40
22	70	00	F0	62	B0	F0	D0	A2	70	60	50	E2	30	20	40

Table 4-8: 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
23	B0	00	F0	63	B0	F0	E0	A3	70	60	50	E3	30	20	40
24	F0	00	F0	64	B0	F0	F0	A4	70	70	50	E4	40	20	40
25	F0	00	B0	65	B0	E0	F0	A5	60	70	50	E5	40	20	30
26	F0	00	70	66	B0	D0	F0	A6	60	70	50	E6	40	20	30
27	F0	00	40	67	B0	C0	F0	A7	50	70	50	E7	40	20	30
28	F0	00	00	68	00	00	70	A8	50	70	50	E8	40	20	20
29	F0	40	00	69	10	00	70	A9	50	70	50	E9	40	30	20
2A	F0	70	00	6A	30	00	70	AA	50	70	60	EA	40	30	20
2B	F0	B0	00	6B	50	00	70	AB	50	70	60	EB	40	30	20
2C	F0	F0	00	6C	70	00	70	AC	50	70	70	EC	40	40	20
2D	B0	F0	00	6D	70	00	50	AD	50	60	70	ED	30	40	20
2E	70	F0	00	6E	70	00	30	AE	50	60	70	EE	30	40	20
2F	40	F0	00	6F	70	00	10	AF	50	50	70	EF	30	40	20
30	00	F0	00	70	70	00	00	B0	00	00	40	F0	20	40	20
31	00	F0	40	71	70	10	00	B1	10	00	40	F1	20	40	30
32	00	F0	70	72	70	30	00	B2	20	00	40	F2	20	40	30
33	00	F0	B0	73	70	50	00	B3	30	00	40	F3	20	40	30
34	00	F0	F0	74	70	70	00	B4	40	00	40	F4	20	40	40
35	00	B0	F0	75	50	70	00	B5	40	00	30	F5	20	30	40
36	00	70	F0	76	30	70	00	B6	40	00	20	F6	20	30	40
37	00	40	F0	77	10	70	00	B7	40	00	10	F7	20	30	40
38	70	70	F0	78	00	70	00	B8	40	00	00	F8	00	00	00
39	90	70	F0	79	00	70	10	B9	40	10	00	F9	00	00	00
3A	B0	70	F0	7A	00	70	30	BA	40	20	00	FA	00	00	00
3B	D0	70	F0	7B	00	70	50	BB	40	30	00	FB	00	00	00
3C	F0	70	F0	7C	00	70	70	BC	40	40	00	FC	00	00	00
3D	F0	70	D0	7D	00	50	70	BD	30	40	00	FD	00	00	00
3E	F0	70	B0	7E	00	30	70	BE	20	40	00	FE	00	00	00
3F	F0	70	90	7F	00	10	70	BF	10	40	00	FF	00	00	00

16 bpp color

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

5 Power Save Mode

The S1D13706 is designed for very low-power applications. During normal operation, the internal clocks are dynamically disabled when not required. The S1D13706 design also includes a Power Save Mode to further save power. When Power Save Mode is initiated, LCD power sequencing is required to ensure the LCD bias power supply is disabled properly. For further information on LCD power sequencing, see Section 6, “LCD Power Sequencing” on page 29.

For Power Save Mode AC Timing, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

5.1 Overview

The S1D13706 includes a software initiated Power Save Mode. Enabling/disabling Power Save Mode is controlled using the Power Save Mode Enable bit (REG[A0h] bit 0).

While Power Save Mode is enabled the following conditions apply.

- LCD display is inactive.
- LCD interface outputs are forced low.
- Memory is in-accessible.
- Registers are accessible.
- Look-Up Table registers are accessible.

5.2 Registers

5.2.1 Power Save Mode Enable

REG[A0h] Power Save Configuration Register							Read/Write
VNDP Status (RO)	n/a	n/a	n/a	Memory Controller Power Save Status (RO)	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 S1D13706 back to normal mode.

Note

Enabling/disabling Power Save Mode requires proper LCD Power Sequencing. See Section 6, “LCD Power Sequencing” on page 29.

5.2.2 Memory Controller Power Save Status

REG[A0h] Power Save Configuration Register							Read/Write
VNDP Status (RO)	n/a	n/a	n/a	Memory Controller Power Save Status (RO)	n/a	n/a	Power Save Mode Enable

The Memory Controller Power Save Status bit is a read-only status bit which indicates the power save state of the S1D13706 SRAM interface. When this bit returns a 1, the SRAM interface is powered down. When this bit returns a 0, the SRAM interface is active. This bit returns a 0 after a chip reset.

Note

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

5.3 Enabling Power Save Mode

Power Save Mode must be enabled using the following steps.

1. Disable the LCD bias power using GPO.

Note
The S5U13706B00C uses GPO to control the LCD bias power supplies. Your system design may vary.

2. Wait for the LCD bias power supply to discharge. The discharge time must be based on the time specified in the LCD panel specification.
3. Enable Power Save Mode - set REG[A0h] bit 0 to 1.
4. At this time, the LCD pixel clock source may be disabled (Optional).
5. Optionally, when the Memory Controller Power Save Status bit (REG[A0h] bit 3) returns a 1, the Memory Clock source may be safely shut down.

5.4 Disabling Power Save Mode

Power Save Mode must be disabled using the following steps.

1. If the Memory Clock source is shut down, it must be started and the Memory Controller Power Save Status bit must return a 0. **Note if the pixel clock source is disabled, it must be started before step 2.**
2. Disable Power Save Mode - set REG[A0h] bit 0 to 0.
3. Wait for the LCD bias power supply to charge. The charge time must be based on the time specified in the LCD panel specification.
4. Enable the LCD bias power using GPO.

Note
The S5U13706B00C uses GPO to control the LCD bias power supplies. Your system design may vary.

6 LCD Power Sequencing

The S1D13706 requires LCD power sequencing (the process of powering-on and powering-off the LCD panel). LCD power sequencing allows the LCD bias voltage to discharge prior to shutting down the LCD signals, preventing long term damage to the panel and avoiding unsightly “lines” at power-on/power-off.

Proper LCD power sequencing for power-off requires a delay from the time the LCD power is disabled to the time the LCD signals are shut down. Power-on requires the LCD signals to be active prior to applying power to the LCD. This time interval depends on the LCD bias power supply design. For example, the LCD bias power supply on the S5U13706 Evaluation board requires 0.5 seconds to fully discharge. Other power supply designs may vary.

This section assumes the LCD bias power is controlled through GPO. The S1D13706 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 *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

Note

This section discusses LCD power sequencing for passive and TFT (non-HR-TFT/D-TFD) panels only. For further information on LCD power sequencing the HR-TFT, see *Connecting to the Sharp HR-TFT Panels*, document number X31B-G-011-xx. For further information on LCD power sequencing the D-TFD, see *Connecting to the Epson D-TFD Panels*, document number X31B-G-012-xx.

6.1 Enabling the LCD Panel

The HAL function seDisplayEnable(TRUE) can be used to enable the LCD panel. The function enables the LCD panel using the following steps.

1. Enable the LCD signals - Set Display Blank bit (REG[70h] bit 7) to 0.
2. Wait the required delay time as specified in the LCD panel specification (must be set using 13706CFG). For further information on 13706CFG, see the *13706CFG User Manual*, document number X31B-B-001-xx.
3. Enable GPO to activate the LCD bias power.

Note
seLcdDisplayEnable is included in the C source file **hal_misc.c** available on the internet at www.eea.epson.com.

6.2 Disabling the LCD Panel

The HAL function seDisplayEnable(FALSE) can be used to disable the LCD panel. The function disables the LCD panel using the following steps.

1. Disable the LCD power using GPO.
2. Wait for the LCD bias power supply to discharge (based on the delay time as specified in the LCD panel specification).
3. Disable the LCD signals - Set Display Blank bit (REG[70h] bit 7) to 1.
4. At this time, the LCD pixel clock source may be disabled (Optional). Note the LUT must not be accessed if the pixel clock is not active.

Note
seLcdDisplayEnable is included in the C source file **hal_misc.c** available on the internet at www.eea.epson.com.

7 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 320x240 is 320 pixels wide and 240 pixels wide.

SwivelView rotates the display image counter-clockwise in ninety degree increments, possibly resulting in a display that is higher than it is wide. Rotating the image on a 320x240 display by 90 or 270 degrees yields a display that is now 240 pixels wide and 320 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.

The S1D13706 provides hardware support for SwivelView in all color depths (1, 2, 4, 8 and 16 bpp).

For further details on the SwivelView feature, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

7.1 Registers

These are the registers which control the SwivelView feature.

REG[71h] Special Effects Register							
Display Data Word Swap	Display Data Byte Swap	n/a	Sub-Window Enable	n/a	n/a	SwivelView Mode Select Bit 1	SwivelView Mode Select Bit 0

The SwivelView modes are selected using the SwivelView Mode Select Bits [1:0]. The combinations of these bits provide the following rotations.

Table 7-1: SwivelView Enable Bits

SwivelView Enable Bit 1	SwivelView Enable Bit 0	SwivelView Orientation
0	0	0° (normal)
0	1	90°
1	0	180°
1	1	270°

REG[74h] Main Window Display Start Address Register0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[75h] Main Window Display Start Address Register 1							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

REG[76h] Main Window Display Start Address Register 2							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	Bit 16

These registers represent a dword address which points to the start of the main window image in the display buffer. An address of 0 is the start of the display buffer. For the following SwivelView mode descriptions, the *desired byte address* is the starting display address for the main window image, and *panel width* and *panel height* refer to the physical panel dimensions.

Note
Truncate all fractional values before writing to the address registers.

In SwivelView 0°, program the start address
= desired byte address ÷ 4.

In SwivelView 90°, program the start address
= ((desired byte address + (panel height × bpp ÷ 8)) ÷ 4) - 1.

In SwivelView 180°, program the start address
= ((desired byte address + (panel width × panel height × bpp ÷ 8)) ÷ 4) - 1.

In SwivelView 270°, program the start address
= (desired byte address + ((panel width - 1) × panel height × bpp ÷ 8)) ÷ 4.

Note
SwivelView 0° and 180° require the panel width to be a multiple of 32 ÷ bits-per-pixel. SwivelView 90° and 270° require the panel height to be a multiple of 32 ÷ bits-per-pixel. If this is not possible, a virtual display (one larger than the physical panel size) is required which does satisfy the above requirements. To create a virtual display, program the main window line address offset to values which are greater than that required for the given display width.

REG[78h] Main Window Line Address Offset Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[79h] Main Window Line Address Offset Register 1							
n/a	n/a	n/a	n/a	n/a	n/a	Bit 9	Bit 8

These registers indicate the number of dwords per line in the main window image (typically the panel width).

number of dwords per line = image width ÷ (32 ÷ bpp)

Note
The image width must be a multiple of 32 ÷ bpp. If the panel width is not such a multiple, a slightly larger width is chosen.

Note
Round up to the nearest integer all line address values that have fractional parts.

7.2 Examples

Example 1: In SwivelView 0° (normal) mode, program the main window registers for a 320x240 panel at color depth of 4 bpp.

1. Confirm the main window coordinates are valid.
   The horizontal coordinates must be a multiple of 32 ÷ bpp.

   320 ÷ (32 ÷ 4) = 40

   Main window horizontal coordinate is valid.

2. Determine the main window display start address.
   The main window is typically placed at the start of display memory which is at display address 0.

   main window display start address register = desired byte address ÷ 4
   = 0

   Program the Main Window Display Start Address registers. REG[74h] is set to 00h, REG[75h] is set to 00h, and REG[76h] is set to 00h.

3. Determine the main window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 320 ÷ (32 ÷ 4)
   = 40
   = 28h

   Program the Main Window Line Address Offset registers. REG[78h] is set to 28h, and REG[79h] is set to 00h.

Example 2: In SwivelView 90° mode, program the main window registers for a 320x240 panel at a color depth of 4 bpp.

1. Confirm the main window coordinates are valid.
   The vertical coordinates must be a multiple of 32 ÷ bpp.

   240 ÷ (32 ÷ 4) = 30

   Main window vertical coordinate is valid.

2. Determine the main window display start address.
   The main window is typically placed at the start of display memory, which is at display address 0.

   main window display start address register
   = ((desired byte address + (panel height × bpp ÷ 8)) ÷ 4) - 1
   = ((0 + (240 × 4 ÷ 8) ÷ 4) - 1
   = 29
   = 1Dh

   Program the Main Window Display Start Address registers. REG[74h] is set to 1Dh, REG[75h] is set to 00h, and REG[76h] is set to 00h.

3. Determine the main window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 240 ÷ (32 ÷ 4)
   = 30
   = 1Eh

Program the Main Window Line Address Offset register. REG[78h] is set to 1Eh, and REG[79h] is set to 00h.

Example 3: In SwivelView 180° mode, program the main window registers for a 320x240 panel at a color depth of 4 bpp.

1. Confirm the main window coordinates are valid.
   The horizontal coordinates must be a multiple of 32 ÷ bpp.

   320 ÷ (32 ÷ 4) = 40

   Main window horizontal coordinate is valid.

2. Determine the main window display start address.
   The main window is typically placed at the start of display memory which is at display address 0.

   main window display start address register
   = ((desired byte address + (panel width × panel height × bpp ÷ 8)) ÷ 4) - 1
   = ((0 + (320 × 240 × 4 ÷ 8)) ÷ 4) - 1
   = 9599
   = 257Fh.

   Program the Main Window Display Start Address registers. REG[74h] is set to 7Fh, REG[75h] is set to 25h, and REG[76h] is set to 00h.

3. Determine the main window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 320 ÷ (32 ÷ 4)
   = 40
   = 28h

   Program the Main Window Line Address Offset registers. REG[78h] is set to 28h, and REG[79h] is set to 00h.

Example 4: In SwivelView 270° mode, program the main window registers for a 320x240 panel at a color depth of 4 bpp.

1. Confirm the main window coordinates are valid.
   The vertical coordinates must be a multiple of 32 ÷ bpp.

   240 ÷ (32 ÷ 4) = 30

   Main window coordinates are valid.

2. Determine the main window display start address.
   The main window is typically placed at the start of display memory, which is at display address 0.

   main window display start address register
   = (desired byte address + ((panel width - 1) × panel height × bpp ÷ 8) ÷ 4)
   = (0 + ((320 - 1) × 240 × 4 ÷ 8) ÷ 4)
   = 9570
   = 2562h

   Program the Main Window Display Start Address registers. REG[74h] is set to 62h, REG[75h] is set to 25h, and REG[76h] is set to 00h.

3. Determine the main window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 240 ÷ (32 ÷ 4)
   = 30
   = 1Eh

   Program the Main Window Line Address Offset registers. REG[78h] is set to 1Eh, and REG[79h] is set to 00h.

7.3 Limitations

7.3.1 SwivelView 0° and 180°

In SwivelView 0° and 180°, the main window line address offset register requires the *panel width* to be a multiple of 32 ÷ bits-per-pixel. If this is not the case, then the main window line address offset register must be programmed to a longer line which is a multiple of 32 ÷ bits-per-pixel. This longer line creates a virtual image where the width is *main window line address offset register* × *32* ÷ *bits-per-pixel* and the main window image must be drawn right-justified to this virtual width.

7.3.2 SwivelView 90° and 270°

In SwivelView 90° and 270°, the main window line address offset register requires the *panel height* to be a multiple of 32 ÷ bits-per-pixel. If this is not the case, then the main window line address offset register must be programmed to a longer line which is a multiple of 32 ÷ bits-per-pixel. This longer line creates a virtual image whose width is *main window line address offset register* × *32* ÷ *bits-per-pixel* and the main window image must be drawn right-justified to this virtual width.

8 Picture-In-Picture Plus

8.1 Concept

Picture-in-Picture Plus enables a sub-window within the main display window. The sub-window may be positioned anywhere within the main window and is controlled through the Sub-Window control registers (see Section 8.2, “Registers”). The sub-window retains the same color depth and SwivelView orientation as the main window.

The following diagram shows an example of a sub-window within a main window.

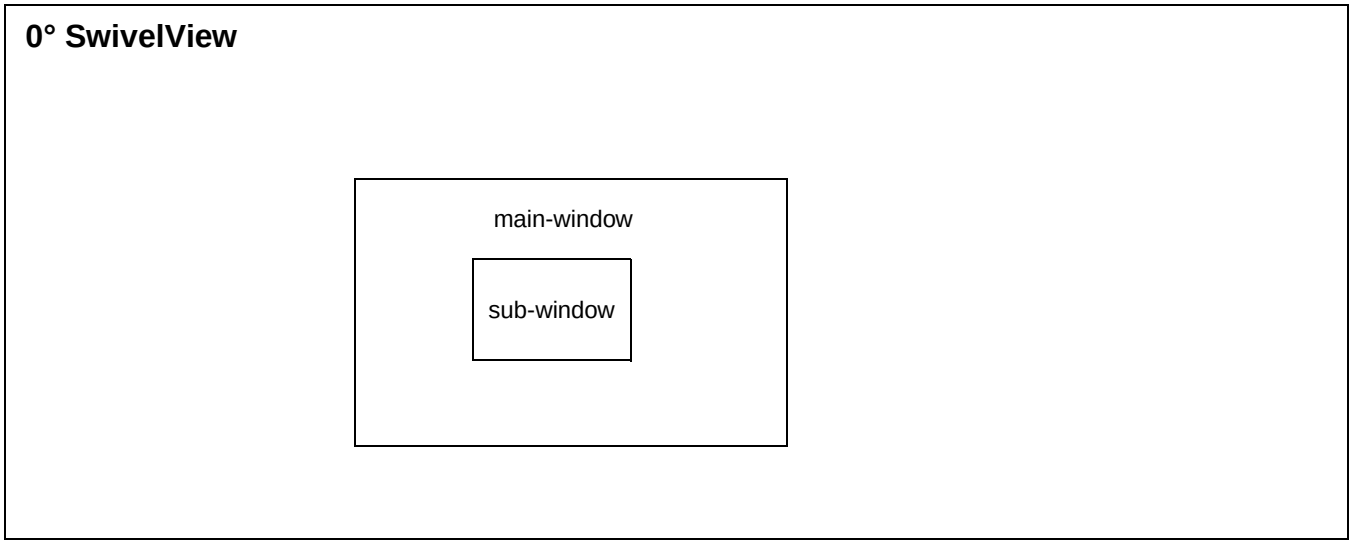


Figure 8-1: Picture-in-Picture Plus with SwivelView disabled

8.2 Registers

These are registers which control the Picture-In-Picture Plus feature.

REG[71h] Special Effects Register							
Display Data Word Swap	Display Data Byte Swap	n/a	Sub-Window Enable	n/a	n/a	SwivelView Mode Select Bit 1	SwivelView Mode Select Bit 0

This bit enables a sub-window within the main window. The location of the sub-window within the landscape window is determined by the Sub-Window X Position registers (REG[84h], REG[85h], REG[8Ch], REG[8Dh]) and Sub-Window Y Position registers (REG[88h], REG[89h], REG[90h], REG[91h]). The sub-window has its own Display Start Address register (REG[7Ch, REG[7Dh], REG[7Eh]) and Memory Address Offset register (REG[80h], REG[81h]). The sub-window shares the same color depth and SwivelView orientation as the main window.

REG[74h] Main Window Display Start Address Register0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[75h] Main Window Display Start Address Register 1							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

REG[76h] Main Window Display Start Address Register 2							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	Bit 16

These registers represent a dword address which points to the start of the main window image in the display buffer. An address of 0 is the start of the display buffer. For the following SwivelView mode descriptions, the *desired byte address* is the starting display address for the main window image, and *panel width* and *panel height* refer to the physical panel dimensions.

Note
Truncate all fractional values before writing to the address registers.

In SwivelView 0°, program the start address
= desired byte address ÷ 4.

In SwivelView 90°, program the start address
= ((desired byte address + (panel height × bpp ÷ 8)) ÷ 4) - 1.

In SwivelView 180°, program the start address
= ((desired byte address + (panel width × panel height × bpp ÷ 8)) ÷ 4) - 1.

In SwivelView 270°, program the start address
= (desired byte address + ((panel width - 1) × panel height × bpp ÷ 8)) ÷ 4.

Note
SwivelView 0° and 180° require the panel width to be a multiple of 32 ÷ bits-per-pixel. SwivelView 90° and 270° require the panel height to be a multiple of 32 ÷ bits-per-pixel. If this is not possible, a virtual display (one larger than the physical panel size) is required which does satisfy the above requirements. To create a virtual display, program the main window line address offset to values which are greater than that required for the given display width.

REG[78h] Main Window Line Address Offset Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[79h] Main Window Line Address Offset Register 1							
n/a	n/a	n/a	n/a	n/a	n/a	Bit 9	Bit 8

These registers indicate the number of dwords per line in the main window image (typically the panel width).

number of dwords per line = image width ÷ (32 ÷ bpp)

Note
The image width must be a multiple of 32 ÷ bpp. If the panel width is not such a multiple, a slightly larger width is chosen.

Note
Round up to the nearest integer all line address values that have fractional parts.

REG[7Ch] Sub-Window Display Start Address Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[7Dh] Sub-Window Display Start Address Register 1							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

REG[7Eh] Sub-Window Display Start Address Register 2							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	Bit 16

These registers represent a dword address which points to the start of the sub-window image in the display buffer. An address of 0 is the start of the display buffer. For the following SwivelView mode descriptions, the *desired byte address* is the starting display address for the sub-window image, and *panel width* and *panel height* refer to the physical panel dimensions. Width and height are used respective to the given SwivelView mode. For example, the sub-window height in SwivelView 90° is the sub-window width in SwivelView 180°.

In SwivelView 0°, program the start address
= desired byte address ÷ 4.

In SwivelView 90°, program the start address
= ((desired byte address + (sub-window width × bpp ÷ 8)) ÷ 4) - 1

In SwivelView 180°, program the start address
= ((desired byte address + (sub-window width × sub-window height × bpp ÷ 8)) ÷ 4) - 1

In SwivelView 270°, program the start address
= (desired byte address + ((sub-window height - 1) × sub-window width × bpp ÷ 8)) ÷ 4

Note
SwivelView 0° and 180° require the panel width to be a multiple of 32 ÷ bpp. SwivelView 90° and 270° require the panel height to be a multiple of 32 ÷ bpp. If this is not possible, a virtual display (one larger than the physical panel size) is required which does satisfy the above requirements. To create a virtual display, program the sub-window line address offset to values which are greater than that required for the given display width.

REG[80h] Sub-Window Line Address Offset Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[81h] Sub-Window Line Address Offset Register 1							
n/a	n/a	n/a	n/a	n/a	n/a	Bit 9	Bit 8

These registers indicate the number of dwords per line in the sub-window image.

number of dwords per line = image width ÷ (32 ÷ bpp)

Note
The image width must be a multiple of 32 ÷ bpp.

REG[84h] Sub-Window X Start Position Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[85h] Sub-Window X Start Position Register 1							
n/a	n/a	n/a	n/a	n/a	n/a	Bit 9	Bit 8

These bits determine the X start position of the sub-window in relation to the origin of the panel. Due to the S1D13706 SwivelView feature, the X start position may not be a horizontal position value (only true in 0° and 180° SwivelView). For further information on defining the value of the X Start Position registers, see Section 8.3, “Picture-In-Picture-Plus Examples” on page 48.

The registers are also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the X start position is incremented by *X* pixels where *X* is relative to the current color depth.

Table 8-1: 32-bit Address Increments for Color Depth

Bits-per-pixel (Color Depth)	Pixel Increment (X)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

For 90° and 270° SwivelView the X start position is incremented in 1 line increments.

In SwivelView 0°, these registers set the horizontal coordinates (x) of the sub-windows’s top left corner. Increasing values of x move the top left corner towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-1:).

Program the Sub-Window X Start Position registers so that
sub-window X start position registers = x ÷ (32 ÷ bits-per-pixel)

Note
x must be a multiple of 32 ÷ bits-per-pixel.

In SwivelView 90°, these registers set the vertical coordinates (y) of the sub-window’s top right corner. Increasing values of y move the top right corner downward in steps of 1 line.

Program the Sub-Window X Start Position registers so that
sub-window X start position registers = y

In SwivelView 180°, these registers set the horizontal coordinates (x) of the sub-window's bottom right corner. Increasing values of x move the bottom right corner towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-1:)

Program the Sub-Window X Start Position registers so that
sub-window X start position registers = (panel width - x) ÷ (32 ÷ bits-per-pixel)

Note
panel width - x must be a multiple of 32 ÷ bits-per-pixel.

In SwivelView 270°, these registers set the vertical coordinates (y) of the sub-window's bottom left corner. Increasing values of y move the bottom left corner downwards in steps of 1 line.

Program the Sub-Window X Start Position registers so that
sub-window X start position registers = panel width - y

REG[88h] Sub-Window Y Start Position Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[89h] Sub-Window Y Start Position Register 1							
n/a	n/a	n/a	n/a	n/a	n/a	Bit 9	Bit 8

These bits determine the Y start position of the sub-window in relation to the origin of the panel. Due to the S1D13706 SwivelView feature, the Y start position may not be a vertical position value (only true in 0° and 180° SwivelView). For further information on defining the value of the Y Start Position registers, see Section 8.3, "Picture-In-Picture-Plus Examples" on page 48.

The registers is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the Y start position is incremented in 1 line increments. For 90° and 270° SwivelView the Y start position is incremented by *Y* pixels where *Y* is relative to the current color depth.

Table 8-2: 32-bit Address Increments for Color Depth

Bits-Per-Pixel (Color Depth)	Pixel Increment (Y)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

In SwivelView 0°, these registers set the vertical coordinates (y) of the sub-windows's top left corner. Increasing values of y move the top left corner downwards in steps of 1 line.

Program the Sub-Window Y Start Position registers so that
sub-window Y start position registers = y

In SwivelView 90°, these registers set the horizontal coordinates (x) of the sub-window's top right corner. Increasing values of x move the top right corner towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-2:)

Program the Sub-Window Y Start Position registers so that
sub-window Y start position registers = (panel height - x) ÷ (32 ÷ bits-per-pixel)

Note
panel height - x must be a multiple of 32 ÷ bits-per-pixel.

In SwivelView 180°, these registers set the vertical coordinates (y) of the sub-window's bottom right corner. Increasing values of y move the bottom right corner downwards in steps of 1 line.

Program the Sub-Window Y Start Position registers so that
sub-window Y start position registers = panel height - y

In SwivelView 270°, these registers set the horizontal coordinates (x) of the sub-window's bottom left corner. Increasing values of x move the bottom left corner towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-2:).

Program the Sub-Window Y Start Position registers so that
sub-window Y start position registers = x ÷ (32 ÷ bits-per-pixel)

Note
x must be a multiple of 32 ÷ bits-per-pixel.

REG[8Ch] Sub-Window X End Position Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[8Dh] Sub-Window X End Position Register 1							
n/a	n/a	n/a	n/a	n/a	n/a	Bit 9	Bit 8

These bits determine the X end position of the sub-window in relation to the origin of the panel. Due to the S1D13706 SwivelView feature, the X end position may not be a horizontal position value (only true in 0° and 180° SwivelView). For further information on defining the value of the X End Position register, see Section 8.3, “Picture-In-Picture-Plus Examples” on page 48.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the X end position is incremented by *X* pixels where *X* is relative to the current color depth.

Table 8-3: 32-bit Address Increments for Color Depth

Bits-Per-Pixel (Color Depth)	Pixel Increment (X)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

For 90° and 270° SwivelView the X end position is incremented in 1 line increments.

In SwivelView 0°, these registers set the horizontal coordinates (x) of the sub-windows’s bottom right corner. Increasing values of x move the bottom right corner towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-3:).

Program the Sub-Window X End Position registers so that
sub-window X end position registers = x ÷ (32 ÷ bits-per-pixel) - 1

Note
x must be a multiple of 32 ÷ bits-per-pixel.

In SwivelView 90°, these registers set the vertical coordinates (y) of the sub-window’s bottom left corner. Increasing values of y move the bottom left corner downward in steps of 1 line.

Program the Sub-Window X End Position registers so that
sub-window X end position registers = y - 1

In SwivelView 180°, these registers set the horizontal coordinates (x) of the sub-window's top left corner. Increasing values of x move the top left corner towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-3:)

Program the Sub-Window X End Position registers so that
sub-window X end position registers = (panel width - x) ÷ (32 ÷ bits-per-pixel) - 1

Note
panel width - x must be a multiple of 32 ÷ bits-per-pixel.

In SwivelView 270°, these registers set the vertical coordinates (y) of the sub-window's top right corner. Increasing values of y move the top right corner downwards in steps of 1 line.

Program the Sub-Window X End Position registers so that
sub-window X end position registers = panel width - y - 1

REG[90h] Sub-Window Y End Position Register 0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[91h] Sub-Window Y End Position Register 1							
n/a	n/a	n/a	n/a	n/a	n/a	Bit 9	Bit 8

These bits determine the Y end position of the sub-window in relation to the origin of the panel. Due to the S1D13706 SwivelView feature, the Y end position may not be a vertical position value (only true in 0° and 180° SwivelView). For further information on defining the value of the Y End Position register, see Section 8.3, “Picture-In-Picture-Plus Examples” on page 48.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the Y end position is incremented in 1 line increments. For 90° and 270° SwivelView the Y end position is incremented by *Y* pixels where *Y* is relative to the current color depth.

Table 8-4: 32-bit Address Increments for Color Depth

Bits-Per-Pixel (Color Depth)	Pixel Increment (Y)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

In SwivelView 0°, these registers set the vertical coordinates (y) of the sub-windows’s bottom right corner. Increasing values of y move the bottom right corner downwards in steps of 1 line.

Program the Sub-Window Y End Position registers so that
sub-window Y end position registers = y - 1

In SwivelView 90°, these registers set the horizontal coordinates (x) of the sub-window’s bottom left corner. Increasing values of x move the top right corner towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-4:)

Program the Sub-Window Y End Position registers so that
sub-window Y end position registers = (panel height - x) ÷ (32 ÷ bits-per-pixel) - 1

Note
panel height - x must be a multiple of 32 ÷ bits-per-pixel.

In SwivelView 180°, these registers set the vertical coordinates (y) of the sub-window’s top left corner. Increasing values of y move the top left corner downwards in steps of 1 line.

Program the Sub-Window Y End Position registers so that
sub-window Y end position registers = panel height - y - 1

In SwivelView 270°, these registers set the horizontal coordinates (x) of the sub-window's top right corner. Increasing values of x move the top right corner towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-4:).

Program the Sub-Window Y End Position registers so that
sub-window Y end position registers = x ÷ (32 ÷ bits-per-pixel) - 1

Note
x must be a multiple of 32 ÷ bits-per-pixel.

8.3 Picture-In-Picture-Plus Examples

8.3.1 SwivelView 0° (Landscape Mode)

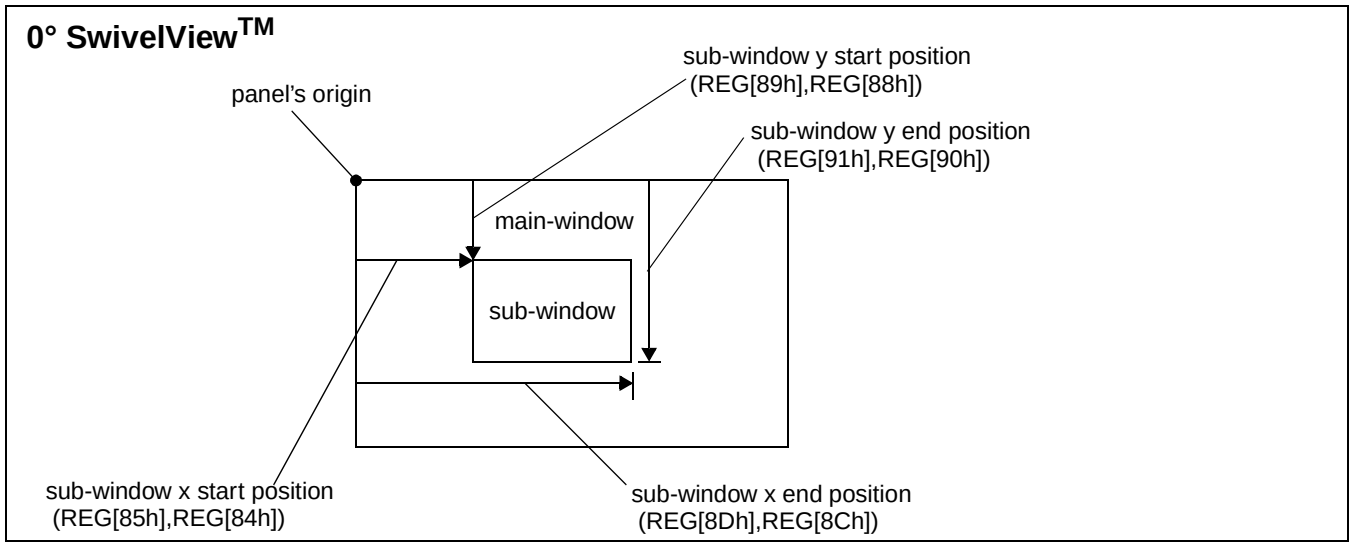


Figure 8-2: Picture-in-Picture Plus with SwivelView disabled

SwivelView 0° (or landscape) is a mode in which both the main and sub-window are non-rotated. The images for each window are typically placed consecutively, with the main window image starting at address 0 and followed by the sub-window image. In addition, both images must start at addresses which are dword-aligned (the last two bits of the starting address must be 0).

Note
It is possible to use the same image for both the main window and sub-window. To do so, set the sub-window line address offset registers to the same value as the main window line address offset registers.

Example 5: Program the main window and sub-window registers for a 320x240 panel at 4 bpp, with the sub-window positioned at (80, 60) with a width of 160 and a height of 120.

1. Confirm the main window coordinates are valid.
   The horizontal coordinates must be a multiple of 32 ÷ bpp.

   320 ÷ (32 ÷ 4) = 40

   Main window horizontal coordinate is valid.

2. Confirm the sub-window coordinates are valid.
   The horizontal coordinates and horizontal width must be a multiple of 32 ÷ bpp.

   80 ÷ (32 ÷ 4) = 10

160 ÷ (32 ÷ 4) = 20

Sub-window horizontal coordinates and horizontal width are valid.

3. Determine the main window display start address.
   The main window is typically placed at the start of display memory which is at display address 0.

   main window display start address register = desired byte address ÷ 4
   = 0

   Program the Main Window Display Start Address registers. REG[74h] is set to 00h, REG[75h] is set to 00h, and REG[76h] is set to 00h.

4. Determine the main window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 320 ÷ (32 ÷ 4)
   = 40
   = 28h

   Program the Main Window Line Address Offset registers. REG[78h] is set to 28h, and REG[79h] is set to 00h.

5. Determine the sub-window display start address.
   The main window image must take up 320 x 240 pixels ÷ 2 pixels per byte= 9600h bytes. If the main window starts at address 0h, the sub-window can start at 9600h.

   sub-window display start address = desired byte address ÷ 4
   = 9600h ÷ 4
   = 2580h.

   Program the Sub-window Display Start Address register. REG[7Ch] is set to 80h, REG[7Dh] is set to 25h, and REG[7Eh] is set to 00h.

6. Determine the sub-window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 160 ÷ (32 ÷ 4)
   = 20
   = 14h

   Program the Sub-window Line Address Offset register. REG[80h] is set to 14h, and REG[81h] is set to 00h.

7. Determine the value for the sub-window X and Y start and end position registers. Let the top left corner of the sub-window be (x1, y1), and let x2 = x1 + width, y2 = y1 + height.

   The X position registers set the horizontal coordinates of the sub-window top left and bottom right corners. Program the X Start Position registers = x1 ÷ (32 ÷ bpp). Program the X End Position registers = x2 ÷ (32 ÷ bpp) - 1.

   The Y position registers, in landscape mode, set the vertical coordinates of the sub-window's top left and bottom right corners. Program the Y Start Position registers = y1. Program the Y End Position registers = y2 - 1.

X Start Position registers	= 80 ÷ (32 ÷ 4)
	= 10
	= 0Ah
Y Start Position registers	= 60
	= 3Ch
X End Position registers	= (80 + 160) ÷ (32 ÷ 4) - 1
	= 29
	= 1Dh
Y End Position registers	= 60 + 120 - 1
	= 179
	= B3h

   Program the Sub-window X Start Position register. REG[84h] is set to 0Ah, and REG[85h] is set to 00h.
   Program the Sub-window Y Start Position register. REG[88h] is set to 3Ch, and REG[89h] is set to 00h.
   Program the Sub-window X End Position register. REG[8Ch] is set to 1Dh, and REG[8Dh] is set to 00h.
   Program the Sub-window Y End Position register. REG[90h] is set to B3h, and REG[91h] is set to 00h.

8. Enable the sub-window.

   Program the Sub-window Enable bit. REG[71h] bit 4 is set to 1.

8.3.2 SwivelView 90°

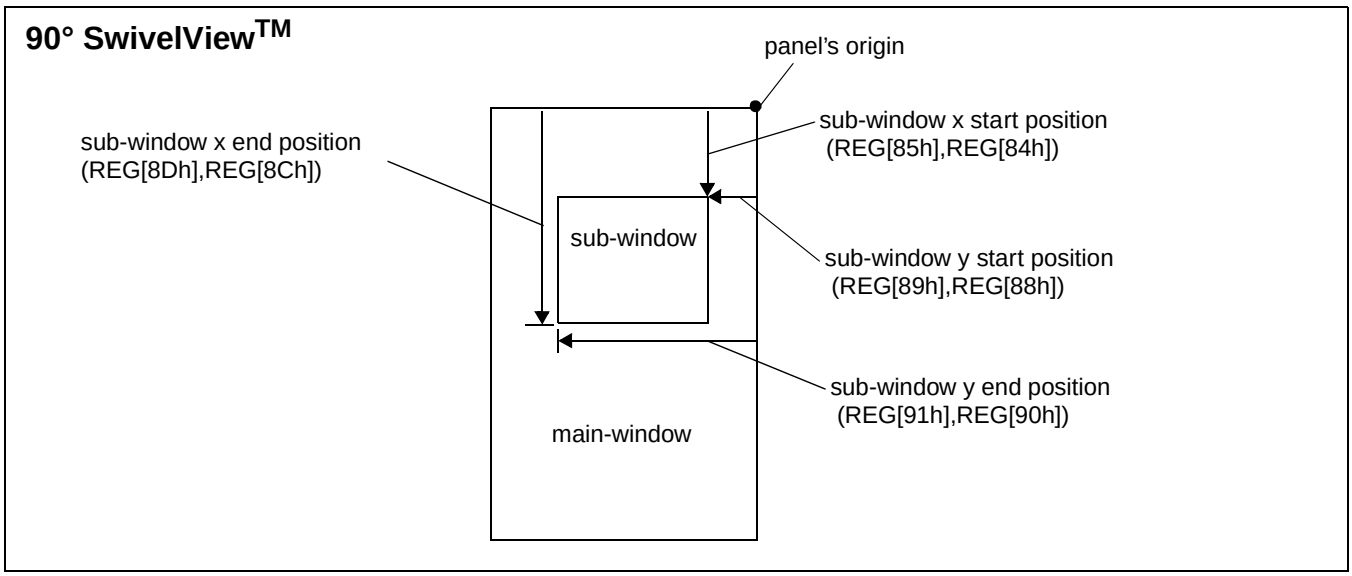


Figure 8-3: Picture-in-Picture Plus with SwivelView 90° enabled

SwivelView 90° is a mode in which both the main and sub-windows are rotated 90° counter-clockwise when shown on the panel. The images for each window are typically placed consecutively, with the main window image starting at address 0 and followed by the sub-window image. In addition, both images must start at addresses which are dword-aligned (the last two bits of the starting address must be 0).

Note
It is possible to use the same image for both the main window and sub-window. To do so, set the sub-window line address offset registers to the same value as the main window line address offset registers.

Note
The Sub-Window X Start Position registers, Sub-Window Y Start Position registers, Sub-Window X End Position registers, and Sub-Window Y End Position registers are named according to the SwivelView 0° orientation. In SwivelView 90°, these registers switch their functionality as described in Section 8.2, "Registers" .

Example 6: In SwivelView 90°, program the main window and sub-window registers for a 320x240 panel at 4 bpp, with the sub-window positioned at SwivelView 90° coordinates (60, 80) with a width of 120 and a height of 160.

1. Confirm the main window coordinates are valid.
   The vertical coordinates must be a multiple of 32 ÷ bpp.

   240 ÷ (32 ÷ 4) = 30

   Main window vertical coordinate is valid.

2. Confirm the sub-window coordinates are valid.
   The horizontal coordinates and horizontal width must be a multiple of 32 ÷ bpp.

   60 ÷ (32 ³ 4) = 7.5 (invalid)
   120 ÷ (32 ÷ 4) = 15

   The sub-window horizontal start coordinate is invalid. Therefore, a valid coordinate close to 60 must be chosen. For example, 8 × (32 ÷ 4) = 64. **Consequently the new sub-window coordinates are (64, 80)**.

3. Determine the main window display start address.
   The main window is typically placed at the start of display memory, which is at display address 0.

   main window display start address register
   = ((desired byte address + (panel height × bpp ÷ 8)) ÷ 4) - 1
   = ((0 + (240 × 4 ÷ 8) ÷ 4) - 1
   = 29
   = 1Dh

   Program the Main Window Display Start Address registers. REG[74h] is set to 1Dh, REG[75h] is set to 00h, and REG[76h] is set to 00h.

4. Determine the main window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 240 ÷ (32 ÷ 4)
   = 30
   = 1Eh

   Program the Main Window Line Address Offset register. REG[78h] is set to 1Eh, and REG[79h] is set to 00h.

5. Determine the sub-window display start address.
   The main window image must take up 320 x 240 pixels ÷ 2 pixels per byte= 9600h bytes. If the main window starts at address 0h, then the sub-window can start at 9600h.

   sub-window display start address register
   = ((desired byte address + (sub-window width × bpp ÷ 8)) ÷ 4) - 1
   = ((9600h + (120 × 4 ÷ 8)) ÷ 4) - 1
   = 9614
   = 258Eh

   Program the Sub-window Display Start Address register. REG[7Ch] is set to 8Eh, REG[7Dh] is set to 25h, and REG[7Eh] is set to 00h.

6. Determine the sub-window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)

=120 ÷ (32 ÷ 4)
= 15
= 0Fh

Program the Sub-window Line Address Offset register. REG[80h] is set to 0Fh, and REG[81h] is set to 00h,

7. Determine the value for the sub-window X and Y start and end position registers. Let the top left corner of the sub-window be (x1, y1), and let x2 = x1 + width, y2 = y1 + height.

   The X position registers set the vertical coordinates of the sub-window top right and bottom left corner. Program the X Start Position registers = y1. Program the X End Position registers = y2 - 1.

   The Y position registers set the horizontal coordinates of the sub-window top right and bottom left corner. Program the Y Start Position registers = (panel height - x2) ÷ (32 ÷ bpp). Program the Y End Position registers = (panel height - x1) ÷ (32 ÷ bpp) - 1.

X Start Position registers	= 80
	= 50h
Y Start Position registers	= (240 - (64 + 120)) ÷ (32 ÷ 4)
	= 07h
X End Position registers	= (80 + 160) - 1
	= 239
	= EFh
Y End Position registers	= (240 - 64) ÷ (32 ÷ 4) - 1
	= 21
	= 15h

   Program the Sub-window X Start Position register. REG[84h] is set to 50h, and REG[85h] is set to 00h.
   Program the Sub-window Y Start Position register. REG[88h] is set to 07h, and REG[89h] is set to 00h.
   Program the Sub-window X End Position register. REG[8Ch] is set to EFh, and REG[8Dh] is set to 00h.
   Program the Sub-window Y End Position register. REG[90h] is set to 15h, and REG[91h] is set to 00h.

8. Enable the sub-window.

   Program the Sub-window Enable bit. REG[71h] bit 4 is set to 1.

8.3.3 SwivelView 180°

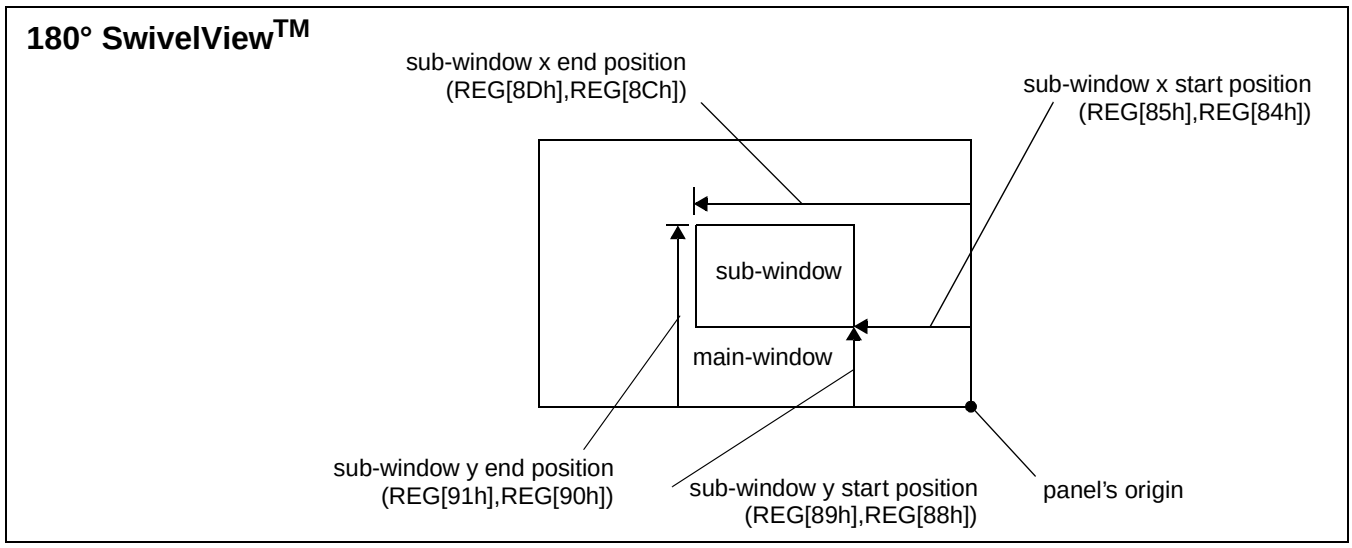


Figure 8-4: Picture-in-Picture Plus with SwivelView 180° enabled

SwivelView 180° is a mode in which both the main and sub-windows are rotated 180° counter-clockwise when shown on the panel. The images for each window are typically placed consecutively, with the main window image starting at address 0 and followed by the sub-window image. In addition, both images must start at addresses which are dword-aligned (the last two bits of the starting address must be 0).

Note

It is possible to use the same image for both the main window and sub-window. To do so, set the sub-window line address offset registers to the same value as the main window line address offset registers.

Note

The Sub-Window X Start Position registers, Sub-Window Y Start Position registers, Sub-Window X End Position registers, and Sub-Window Y End Position registers are named according to the SwivelView 0° orientation. In SwivelView 180°, these registers switch their functionality as described in Section 8.2, "Registers" .

Example 7: In SwivelView 180°, program the main window and sub-window registers for a 320x240 panel at 4 bpp, with the sub-window positioned at SwivelView 180° coordinates (80, 60) with a width of 160 and a height of 120.

1. Confirm the main window coordinates are valid.
   The horizontal coordinates must be a multiple of 32 ÷ bpp.

   320 ÷ (32 ÷ 4) = 40

   Main window horizontal coordinate is valid.

2. Confirm the sub-window coordinates are valid.
   The horizontal coordinates and horizontal width must be a multiple of 32 ÷ bpp.

   80 ÷ (32 ÷ 4) = 10
   160 ÷ (32 ÷ 4) = 20

   Sub-window horizontal coordinates and horizontal width are valid.

3. Determine the main window display start address.
   The main window is typically placed at the start of display memory which is at display address 0.

   main window display start address register
   = ((desired byte address + (panel width × panel height × bpp ÷ 8)) ÷ 4) - 1
   = ((0 + (320 × 240 × 4 ÷ 8)) ÷ 4) - 1
   = 9599
   = 257Fh.

   Program the Main Window Display Start Address registers. REG[74h] is set to 7Fh, REG[75h] is set to 25h, and REG[76h] is set to 00h.

4. Determine the main window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 320 ÷ (32 ÷ 4)
   = 40
   = 28h

   Program the Main Window Line Address Offset registers. REG[78h] is set to 28h, and REG[79h] is set to 00h.

5. Determine the sub-window display start address.
   The main window image must take up 320 x 240 pixels ÷ 2 pixels per byte= 9600h bytes. If the main window starts at address 0h, then the sub-window can start at 9600h.

   sub-window display start address
   = ((desired byte address + (sub-window width × sub-window height × bpp ÷ 8)) ÷ 4) - 1 = ((9600h + (160 × 120 × 4 ÷ 8)) ÷ 4) - 1
   = 11999
   = 2EDFh

   Program the Sub-window Display Start Address registers. REG[7Ch] is set to DFh, REG[7Dh] is set to 2Eh, and REG[7Eh] is set to 00h.

6. Determine the sub-window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 160 ÷ (32 ÷ 4)
   = 20
   = 14h

   Program the Sub-window Line Address Offset registers. REG[80h] is set to 14h, and REG[81h] is set to 00h.

7. Determine the value for the sub-window X and Y start and end position registers. Let the top left corner of the sub-window be (x1, y1), and let x2 = x1 + width, y2 = y1 + height.

   The X position registers set the horizontal coordinates of the sub-window bottom right and top left corner. Program the X Start Position registers = (panel width - x2) ÷ (32 ÷ bpp). Program the X End Position registers = (panel width - x1) ÷ (32 ÷ bpp) - 1.

   The Y position registers set the horizontal coordinates of the sub-window bottom right and top left corner. Program the Y Start Position registers = panel height - y2. Program the Y End Position registers = panel height - y1 - 1.

   X start position registers = (320 - (80 + 160)) ÷ (32 ÷ 4)
   = 10
   = 0Ah
   Y start position registers = 240 - (60 + 120)
   = 60
   = 3Ch
   X end position registers = (320 - 80) ÷ (32 ÷ 4) - 1
   = 29
   = 1Dh
   Y end position registers = 240 - 60 - 1
   = 179
   = B3h

   Program the Sub-window X Start Position registers. REG[84h] is set to 0Ah, and REG[85h] is set to 00h.
   Program the Sub-window Y Start Position registers. REG[88h] is set to 3Ch, and REG[89h] is set to 00h.
   Program the Sub-window X End Position registers. REG[8Ch] is set to 1Dh, and REG[8Dh] is set to 00h.
   Program the Sub-window Y End Position registers. REG[90h] is set to B3h, and REG[91h] is set to 00h.

8. Enable the sub-window.

   Program the Sub-window Enable bit. REG[71h] bit 4 is set to 1.

8.3.4 SwivelView 270°

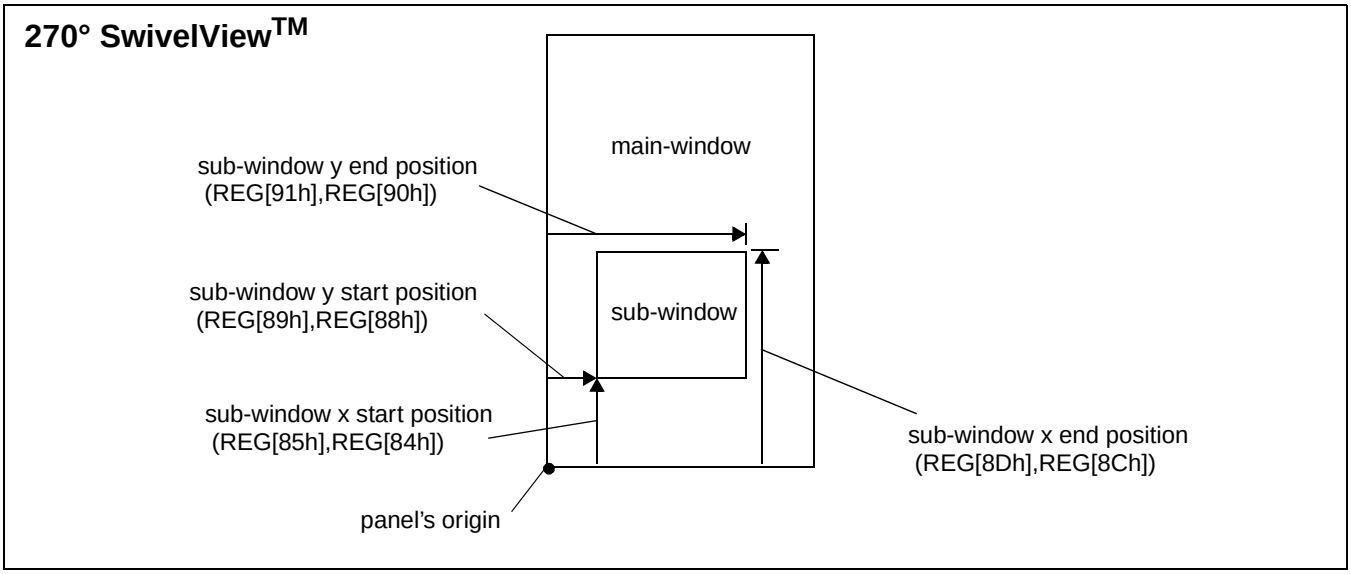


Figure 8-5: Picture-in-Picture Plus with SwivelView 270° enabled

SwivelView 270° is a mode in which both the main and sub-windows are rotated 270° counter-clockwise when shown on the panel. The images for each window are typically placed consecutively, with the main window image starting at address 0 and followed by the sub-window image. In addition, both images must start at addresses which are dword-aligned (the last two bits of the starting address must be 0).

Note
It is possible to use the same image for both the main window and sub-window. To do so, set the sub-window line address offset registers to the same value as the main window line address offset registers.

Note
The Sub-Window X Start Position registers, Sub-Window Y Start Position registers, Sub-Window X End Position registers, and Sub-Window Y End Position registers are named according to the SwivelView 0° orientation. In SwivelView 270°, these registers switch their functionality as described in Section 8.2, "Registers" .

Example 8: In SwivelView 270°, program the main window and sub-window registers for a 320x240 panel at 4 bpp, with the sub-window positioned at SwivelView 270° coordinates (60, 80) with a width of 120 and a height of 160.

1. Confirm the main window coordinates are valid.
   The vertical coordinates must be a multiple of 32 ÷ bpp.

   240 ÷ (32 ÷ 4) = 30

   Main window coordinates are valid.

2. Confirm the sub-window coordinates are valid.
   The horizontal coordinates and horizontal width must be a multiple of 32 ÷ bpp.

   60 ÷ (32 ÷ 4) = 7.5 (invalid)
   120 ÷ (32 ÷ 4) = 15

   The sub-window horizontal start coordinate is invalid. Therefore, a valid coordinate close to 60 must be chosen. For example, 8 × (32 ÷ 4) = 64. **Consequently the new sub-window coordinates are (64, 80).**

3. Determine the main window display start address.
   The main window is typically placed at the start of display memory, which is at display address 0.

   main window display start address register
   = (desired byte address + ((panel width - 1) × panel height × bpp ÷ 8) ÷ 4)
   = (0 + ((320 - 1) × 240 × 4 ÷ 8) ÷ 4)
   = 9570
   = 2562h

   Program the Main Window Display Start Address registers. REG[74h] is set to 62h, REG[75h] is set to 25h, and REG[76h] is set to 00h.

4. Determine the main window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 240 ÷ (32 ÷ 4)
   = 30
   = 1Eh

   Program the Main Window Line Address Offset registers. REG[78h] is set to 1Eh, and REG[79h] is set to 00h.

5. Determine the sub-window display start address.
   The main window image must take up 320 x 240 pixels ÷ 2 pixels per byte= 9600h bytes. If the main window starts at address 0h, then the sub-window can start at 9600h.

   sub-window display start address register
   = (desired byte address + ((sub-window height - 1) × sub-window width × bpp ÷ 8)) ÷ 4
   = (9600h + ((160 - 1) × 120 × 4 ÷ 8)) ÷ 4
   = 11985
   = 2ED1h

   Program the Sub-window Display Start Address registers. REG[7Ch] is set to D1h, REG[7Dh] is set to 2Eh, and REG[7Eh] is set to 00h.

6. Determine the sub-window line address offset.

   number of dwords per line = image width ÷ (32 ÷ bpp)
   = 120 ÷ (32 ÷ 4)
   = 15
   = 0Fh

   Program the Sub-window Line Address Offset. REG[80h] is set to 0Fh, and REG[81h] is set to 00h.

7. Determine the value for the sub-window X and Y start and end position registers. Let the top left corner of the sub-window be (x1, y1), and let x2 = x1 + width, y2 = y1 + height.

   The X position registers sets the vertical coordinates of the sub-window top right and bottom left corner. Program the X Start Position registers = panel width - y2. Program the X End Position registers = panel width - y1 - 1.

   The Y position registers sets the horizontal coordinates of the sub-window top right and bottom left corner. Program the Y Start Position registers = x1 ÷ (32 ÷ bpp). Program the Y End Position registers = x2 ÷ (32 ÷ bpp) - 1.

   X start position registers = 320 - (80 + 160)
   = 80
   = 50h
   Y start position registers = 64 ÷ (32 ÷ 4)
   = 08h
   X end position registers = 320 - 80 - 1
   = 239
   = EFh
   Y end position registers = (64 + 120) ÷ (32 ÷ 4) - 1
   = 22
   = 16h

   Program the Sub-window X Start Position registers. REG[84h] is set to 50h, and REG[85h] is set to 00h.
   Program the Sub-window Y Start Position registers. REG[88h] is set to 08h, and REG[89h] is set to 00h.
   Program the Sub-window X End Position registers. REG[8Ch] is set to EFh, and REG[8Dh] is set to 00h.
   Program the Sub-window Y End Position registers. REG[90h] is set to 16h, and REG[91h] is set to 00h.

8. Enable the sub-window.

   Program the Sub-window Enable bit. REG[71h] bit 4 is set to 1.

8.4 Limitations

8.4.1 SwivelView 0° and 180°

In SwivelView 0° and 180°, the main window line address offset register requires the *panel width* to be a multiple of 32 ÷ bits-per-pixel. If this is not the case, then the main window line address offset register must be programmed to a longer line which is a multiple of 32 ÷ bits-per-pixel. This longer line creates a virtual image where the width is *main window line address offset register* × *32* ÷ *bits-per-pixel* and the main window image must be drawn right-justified to this virtual width.

Similarly, the sub-window line address offset register requires the sub-window image *width* to be a multiple of 32 ÷ bits-per-pixel. If this is not the case, then the sub-window line address offset register must be programmed to a longer line which is a multiple of 32 ÷ bits-per-pixel. This longer line creates a virtual image whose width is *sub-window line address offset register* × *32* ÷ *bits-per-pixel* and the sub-window image must be drawn right-justified to this virtual width.

8.4.2 SwivelView 90° and 270°

In SwivelView 90° and 270°, the main window line address offset register requires the *panel height* to be a multiple of 32 ÷ bits-per-pixel. If this is not the case, then the main window line address offset register must be programmed to a longer line which is a multiple of 32 ÷ bits-per-pixel. This longer line creates a virtual image whose width is *main window line address offset register* × *32* ÷ *bits-per-pixel* and the main window image must be drawn right-justified to this virtual width.

Similarly, the sub-window line address offset register requires the sub-window image *width* to be a multiple of 32 ÷ bits-per-pixel. If this is not the case, then the sub-window line address offset register must be programmed to a longer line which is a multiple of 32 ÷ bits-per-pixel. This longer line creates a virtual image whose width is *sub-window line address offset register* × *32* ÷ *bits-per-pixel* and the sub-window image must be drawn right-justified to this virtual width.

9 Identifying the S1D13706

The S1D13706 can be identified by reading the value contained in the Revision Code Register (REG[00h]). To identify the S1D13706 follow the steps below.

1. Read REG[00h].
2. The production version of the S1D13706 returns a value of 28h (00101000b).
3. The product code is Ah (001010b based on bits 7-2).
4. The revision code is 0h (00b based on bits 1-0).

10 Hardware Abstraction Layer (HAL)

The HAL is a processor independent programming library designed to help port applications and utilities from one SED13xx product to another. Epson has provided this library as a result of developing test utilities for the SED13xx LCD controller products.

The HAL contains functions which are designed to be consistent between SED13xx products, but as the semiconductor products evolve, so must the HAL; consequently there are some differences between HAL functions for different SED13xx products.

Note

As the SED13xx 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.

10.1 API for 13706HAL

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 10-1: HAL Functions

Function	Description
Initialization	
seRegisterDevice	Registers the S1D13706 parameters with the HAL. seRegisterDevice MUST be the first HAL function called by an application.
seInitReg	Initializes the registers, LUT, and allocates memory for default surfaces.
seGetHalVersion	Returns HAL library version information.
seHalTerminate	Frees up memory allocated by the HAL before the application exits.
seGetId	Identifies the controller by interpreting the revision code register.
General HAL Support:	
seGetInstalledMemorySize	Returns the total size of the display buffer in bytes.
seGetAvailableMemorySize	Determines the last byte of display buffer available to an application.
seEnableHardwareDisplaySwapping	Enables hardware data swapping for Big-Endian systems.
seGetResolution seGetMainWinResolution seGetSubWinResolution	Returns the width and height of the active display surface.
seSetSubWinCoordinates	Sets the sub-window coordinates.
seGetSubWinCoordinates	Returns the sub-window coordinates.
seGetBytesPerScanline seGetMainWinBytesPerScanline seGetSubWinBytesPerScanline	Returns the number of bytes in each line of the displayed image. Note that the displayed image may be larger than the physical size of the LCD.
seSetPowerSaveMode	Enables/disables power save mode.
seGetPowerSaveMode	Returns the current state of power save mode.
seSetPowerUpDelay	Sets the power-on delay for power save mode.
seSetPowerDownDelay	Sets the power-down delay for power save mode.
seCheckEndian	Returns the Endian mode of the host CPU platform.

Table 10-1: HAL Functions (Continued)

Function	Description
seSetSwivelViewMode	Sets the SwivelView orientation of the LCD.
seGetSwivelViewMode	Returns the SwivelView orientation of the LCD.
seCheckSwivelViewClocks	Verifies the clocks are set correctly for the requested SwivelView orientation.
seDelay	Delays the given number of seconds before returning.
seDisplayBlank seMainWinDisplayBlank seSubWinDisplayBlank	Blank/unblank the display.
seDisplayEnable seMainWinDisplayEnable seSubWinDisplayEnable	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 buffer for the active surface.
seGetSurfaceOffsetAddress	Returns the offset from the start of display buffer to the start of surface memory.
seAllocMainWinSurface seAllocSubWinSurface	Manually allocates display buffer memory for a surface.
seFreeSurface	Frees any allocated surface memory.
seSetMainWinAsActiveSurface seSetSubWinAsActiveSurface	Changes the active surface.
sePwmEnable	Enables the PWMCLK circuitry.
seCvEnable	Enables the CV Pulse circuitry.
sePwmControl	Configures the PWMCLK registers.
seCvControl	Configures the CV Pulse registers.
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 buffer.
seReadDisplayWord	Reads one word from display buffer.
seReadDisplayDword	Reads one dword from display buffer.
seWriteDisplayBytes	Writes one or more bytes to display buffer.
seWriteDisplayWords	Writes one or more words to display buffer.
seWriteDisplayDwords	Writes one or more dwords to display buffer.
Color Manipulation:	
seWriteLutEntry	Writes one RGB element to the lookup table.

Table 10-1: HAL Functions (Continued)

Function	Description
seReadLutEntry	Reads one RGB element from the lookup table.
seWriteLut	Write the entire lookup table.
seReadLut	Read the entire lookup table.
seSetMode	Sets the color depth of the display and updates the LUT.
seUseMainWinImageForSubWin	Sets the sub-window image to use the same image as the main window.
seGetBitsPerPixel	Gets the current color depth.
Virtual Display	
seVirtInit seMainWinVirtInit seSubWinVirtInit seMainAndSubWinVirtInit	Initialize a surface to hold an image larger than the physical display size. Also required for SwivelView 90° and 270°.
seVirtPanScroll seMainWinVirtPanScroll seSubWinVirtPanScroll seMainAndSubWinVirtPanScroll	Pan (right/left) and Scroll (up/down) the display device over the indicated virtual surface.
Drawing	
seSetPixel seSetMainWinPixel seSetSubWinPixel	Set one pixel at the specified (x,y) co-ordinate and color.
seGetPixel seGetMainWinPixel seGetSubWinPixel	Returns the color of the pixel at the specified (x,y) co-ordinate.
seDrawLine seDrawMainWinLine seDrawSubWinLine	Draws a line between two endpoints in the specified color
seDrawRect seDrawMainWinRect seDrawSubWinRect	Draws a rectangle. The rectangle can be outlined or filled.
seDrawCircle seDrawMainWinCircle seDrawSubWinCircle	Draws a circle of given radius and color at the specified center point.
seDrawEllipse seDrawMainWinEllipse seDrawSubWinEllipse	Draws an ellipse centered on a given point with the specified horizontal and vertical radius.
Register/Display Memory	
seGetLinearDisplayAddress	Returns the linear address of the start of physical display memory.
seGetLinearRegAddress	Returns the linear address of the start of S1D13706 control registers.

10.2 Initialization

Initialization functions are normally the first functions in the HAL library that an application calls. These routines return information about the controller and prepare the HAL library for use.

int seRegisterDevice(const LPHAL_STRUC lpHalInfo)

Description: This function registers the S1D13706 device parameters with the HAL library. The device parameters include such items as address range, register values, desired frame rate, etc. These parameters 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 S1D13706.
ERR_FAILED	The HAL was unable to map S1D13706 display memory to the host platform.

In addition, on Win32 platforms, the following two error values may be returned:

ERR_PCI_DRIVER_-NOT_FOUND	The HAL was unable to locate file SED13XX.VXD
ERR_PCI_BRIDGE_-ADAPTER_NOT_FOUND	The driver file SED13XX.VXD was unable to locate the PCI bridge adapter board attached to the evaluation board.

Note
seRegisterDevice() MUST be called before any other HAL functions.

int seInitReg(unsigned Flags)

Description: This function initializes the S1D13706 registers, the LUT, assigns default surfaces and allocates memory accordingly.

Parameters: 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_NOT_ENOUGH_MEMORY	Insufficient display buffer.
ERR_CLKI_NOT_IN_TABLE	Could not program CLKI in clock synthesizer because selected frequency not in table.
ERR_CLKI2_NOT_IN_TABLE	Could not program CLKI2 in clock synthesizer because selected frequency not in table.

void seGetHalVersion(const char ** pVersion, const char ** pStatus, const char **pRevision)

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 string containing the HAL version code.
pStatus	A pointer to the string containing the HAL status code A “B” designates a beta version of the HAL, a NULL indicates the release version
pRevision	A pointer to the string containing 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)
*pRevision == “5” (fifth update of the beta)

int seHalTerminate(void)

Description: Frees up memory allocated by HAL before application exits.

Parameters: none.

Return Value:

ERR_OK	HAL is now ready for application to exit.
ERR_PCI_DRIVER_NOT_FOUND	Could not find PCI driver (Intel Windows platform only).
ERR_PCI_BRIDGE_ADAPTER_NOT_FOUND	Could not find PCI Bridge Adapter board (Intel Windows platform only).
ERR_FAILED	Could not free memory.

int seGetId(int * pId)

Description: Reads the S1D13706 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 the revision code.

Return Value:

ERR_OK	The operation completed with no problems
ERR_UNKNOWN_DEVICE	The product code was not for the S1D13706.

10.2.1 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 the display buffer in bytes.

For the S1D13706, seGetInstalledMemorySize() and seGetAvailableMemorySize() return the same value.

Parameters: None

Return Value: The return value is the size of the display buffer in bytes (1 4000h for the S1D13706).

DWORD seGetAvailableMemorySize(void)

Description: This function returns an offset to the last byte of memory 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.

For the S1D13706, seGetInstalledMemorySize() and seGetAvailableMemorySize() return the same value.

Parameters: None.

Return Value: The return value is the size of the available amount of display buffer memory directly accessible to an application.

int seEnableHardwareDisplaySwapping(int Enable)

Description: The S1D13706 requires 16 bits-per-pixel data to be in little-endian format. On big-endian systems, the software or hardware needs to swap this data. seEnableHardwareDisplaySwapping() is intended to be used on big-endian systems, where system performance can be improved by utilizing hardware swapping of display memory bytes in 16 bits-per-pixel.

If the system is not big-endian, or if the bits-per-pixel is not 16, this function will not enable hardware display swapping. However, a flag is set in the HAL, and if seSetMode is later called to set the bits-per-pixel to 16 in a big-endian system, hardware display swapping is enabled. Also, if seSetMode is called to set the bits-per-pixel to a value other than 16, then hardware display swapping is disabled.

Parameters:

Enable	Call with Enable set to TRUE to enable hardware display swapping. Call with Enable set to FALSE to disable hardware display swapping.

Return Value:

ERR_OK	Function completed successfully
ERR_FAILED	Returned when caller requested that hardware display swapping be enabled, but system not in 16 bits-per-pixel or system is not big-endian.

int seGetResolution(unsigned *Width, unsigned *Height)
void seGetMainWinResolution(unsigned *Width, unsigned *Height)
void seGetSubWinResolution(unsigned *Width, unsigned *Height)

Description: seGetResolution() returns the width and height of the active surface (main window or sub-window).

seGetMainWinResolution() and seGetSubWinResolution() return the width and height of the respective window.

Virtual dimensions are not accounted for in the return values for width and height. For example, seGetMainWinResolution() always returns the panel dimensions, regardless of the value of the line address offset registers.

The width and height are adjusted for SwivelView orientation.

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.

seGetMainWinResolution() and seGetSubWinResolution() do not return any value.

void seSetSubWinCoordinates(DWORD x1, DWORD y1, DWORD x2, DWORD y2)

Description: seSetSubWinCoordinates sets the upper left and lower right corners of the sub-window display.

(x1, y1) and (x2, y2) are relative to the upper left corner of the panel as defined by the SwivelView mode.

Parameters:

x1	The sub-window x start position (upper left corner).
y1	The sub-window y start position (upper left corner).
x2	The sub-window x end position (lower right corner).
y2	The sub-window y end position (lower right corner).

Return Value: None.

void seGetSubWinCoordinates(DWORD *x1, DWORD *y1, DWORD *x2, DWORD *y2)

Description: seGetSubWinCoordinates return the upper left and lower right corners of the sub-window display.

The coordinates are adjusted for SwivelView orientation.

Parameters:

x1	A pointer to an unsigned long which will receive the sub-window x start position (upper left corner).
y1	A pointer to an unsigned long which will receive the sub-window y start position (upper left corner).
x2	A pointer to an unsigned long which will receive the sub-window x end position (lower right corner).
y2	A pointer to an unsigned long which will receive the sub-window y end position (lower right corner).

Return Value: None.

unsigned seGetBytesPerScanline(void)
unsigned seGetMainWinBytesPerScanline(void)
unsigned seGetSubWinBytesPerScanline(void)

Description: These functions return the number of bytes in each line of the displayed image. Note that the displayed image may be larger than the physical size of the LCD.

seGetBytesPerScanline() returns the number of bytes per scanline for the current active surface.

seGetMainWinBytesPerScanline() and seGetSubWinBytesPerScanline() return the number of bytes per scanline for the surface indicated in the function name.

To work correctly these routines require the S1D13706 registers to be initialized prior to being called.

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.

void seSetPowerSaveMode(BOOL Enable)

Description: This function enables or disables the power save mode.

When power save mode is enabled the S1D13706 reduces power consumption by making the displays inactive and ignoring memory accesses. Disabling power save mode re-enables the video system to full functionality.

When powering down, the following steps are implemented:

1. Disable LCD power
2. Delay for LCD power down time interval [see seSetPowerDownDelay()].
3. Enable power save mode

When powering up, the following steps are implemented:

1. Disable power save mode
2. Delay for LCD power up time interval [see seSetPowerUpDelay()]
3. Enable LCD power

Note
seSetPowerSaveMode() waits on vertical non-display (VNDP) cycles for delays. If there is no VNDP cycle, this function will freeze the system. To ensure VNDP cycles are being generated, ensure that there is a clock available for PCLK. Alternatively, set the power-up and power-down times to 0.

Parameters: Enable — Call with Enable set to TRUE to set power save mode. Call with Enable set to FALSE to disable power save mode.

Return Value: None.

BOOL seGetPowerSaveMode(void)

Description: seGetPowerSaveMode() returns the current state of power save mode.

Parameters: None.

Return Value: The return value is TRUE if power save mode is enabled. The return value is FALSE if power save mode is not enabled.

void seSetPowerUpDelay(WORD PowerupTime)

Description: seSetPowerUpDelay() sets the power-up delay for seSetPowerSaveMode().

Parameters: PowerupTime — Power-up time, in milliseconds.

Return Value: None.

void seSetPowerDownDelay(WORD PowerdownTime)

Description: seSetPowerDownDelay() sets the power-down delay for seSetPowerSaveMode().

Parameters: PowerdownTime — Power-down time, in milliseconds.

Return Value: None.

void 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: None.

int seSetSwivelViewMode(int rotate)

Description: This function sets the SwivelView orientation of the LCD display. Display memory is automatically released and then reallocated as necessary for the display size.

IMPORTANT

When the SwivelView mode is changed, memory allocated for both the main window and sub-window display buffer is freed and the display buffer memory is reassigned. The application must redraw the display and re-initialize the sub-window (if used) and redraw after calling seSetSwivelViewMode().

Parameters:

rotate	The values for rotate are: LANDSCAPE: display not rotated ROTATE90: display rotated 90 degrees counterclockwise ROTATE180: display rotated 180 degrees counterclockwise ROTATE270: display rotated 270 degrees counterclockwise	

Return Value:

ERR_OK	The new rotation was completed with no problems.
ERR_NOT_ENOUGH_MEMORY	Insufficient display buffer.

int seGetSwivelViewMode(void)

Description: This function retrieves the SwivelView orientation of the LCD display.

The SwivelView status is read directly from the S1D13706 registers. Calling this function when the LCD display is not initialized will result in an erroneous return value.

Note
seGetSwivelViewMode() was previously called seGetLcdOrientation(). It is now recommended to call seGetSwivelViewMode() instead of seGetLcdOrientation().

Parameters: None.

Return Value:

LANDSCAPE	Not rotated.
ROTATE90	Display is rotated 90 degrees counterclockwise.
ROTATE180	Display is rotated 180 degrees counterclockwise.
ROTATE270	Display is rotated 270 degrees counterclockwise.

int seCheckSwivelViewClocks(unsigned BitsPerPixel, unsigned Rotate)

Description: This function verifies that the clocks are properly configured for the a SwivelView mode given the bits-per-pixel and rotation (see the section titled SwivelView in the S1D13706 Hardware Functional Specification document).

Parameters:

BitsPerPixel	The given color depth. BitsPerPixel can be one of the following: 1, 2, 4, 8, 16.
Rotate	The values for Rotate are: LANDSCAPE: display not rotated ROTATE90: display rotated 90 degrees counterclockwise ROTATE180: display rotated 180 degrees counterclockwise ROTATE270: display rotated 270 degrees counterclockwise

Return Value:

ERR_OK	The function completed with no problems
ERR_SWIVELVIEW_CLOCK	The clocks are not configured correctly.

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.

For non-Intel 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 S1D13706 control registers must be configured to correct values.
b) The display interface must be enabled (not in power save mode).

For Intel platforms, seDelay() calls the C library time functions to delay the desired amount of time using the system clock.

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 non-Intel platforms in which the power save mode is enabled.

void seDisplayBlank(BOOL Blank)
void seMainWinDisplayBlank(BOOL Blank)
void seSubWinDisplayBlank(BOOL Blank)

Description: These functions blank their respective display. 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, the operation is perceived to be smoother and cleaner.

seDisplayBlank() will blank the display associated with the current active surface.

seDisplayMainWinBlank() and seDisplaySubWinBlank() 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 seMainWinDisplayEnable(BOOL Enable)
void seSubWinDisplayEnable(BOOL Enable)

Description: These functions enable or disable the selected display device.

seDisplayEnable() enables or disables the display for the active surface.

seMainWinDisplayEnable() enables or disables the main window display (for the S1D13706, the display blank feature is used to enable or disable the main window).

seSubWinDisplayEnable() enables or disables the sub-window display.

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.

10.2.2 Advance HAL Functions

The advanced HAL functions include a level of access that most applications will never need to access.

int seBeginHighPriority(void)

Description: Writing and debugging software under the Windows operating system greatly simplifies the development process for the S1D13706 evaluation system. One issue which impedes application programming is that of latency. Time critical operations (i.e. performance measurement) 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: The priority nest count which is the number of times seBeginHighPriority() has been called without a corresponding call to seEndHighPriority().

int seEndHighPriority(void)

Description: This function decreases the priority nest count. When this count reaches zero, the thread priority of the calling application is set 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: The priority nest count which is the number of times seBeginHighPriority() has been called without a corresponding call to seEndHighPriority().

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 for CLKI or CLKI2 (defined in HAL.H).

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
FREQ_100000	100.000 MHz

Return Value:

ERR_OK — The function completed with no problems.

ERR_FAILED — seSetClock failed because of an invalid ClockSelect or an invalid frequency index.

10.2.3 Surface Support

The S1D13706 HAL library depends heavily on the concept of surfaces. Through surfaces the HAL tracks memory requirements of the main window and sub-window.

Surfaces allow the HAL to permit or fail function calls which change the geometry of the S1D13706 display memory. Most HAL functions either allocate surface memory or manipulate a surface that has been allocated.

The functions in this section 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:

MAIN_WIN	The main window is the active surface.
SUB_WIN	The sub-window is the active surface.

DWORD seGetSurfaceSize(void)

Description: This function returns the number of display memory bytes allocated to the current active surface.

Parameters: None.

Return Value: The return value is the number of bytes allocated to the current active surface.

The return value will be 0 if this function is called before initializing the registers.

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.

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 seAllocMainWinSurface(DWORD Size)
DWORD seAllocSubWinSurface(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 seAllocMainWinSurface() or seAllocSubWinSurface() 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 seSetMode() 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 seAllocMainWinSurface() and seAllocSubWinSurface().

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 seSetMainWinAsActiveSurface(void)
void seSetSubWinAsActiveSurface(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 functions.

Parameters: None.

Return Value: None.

void sePwmEnable(int Enable)

Description: This function enables or disables the Pulse Width Modulation (PWM) clock circuitry.

Parameters: Enable — Set to TRUE or FALSE to enable or disable PWM.

Return Value: None.

void seCvEnable(int Enable)

Description: This function enables or disables the Contrast Voltage (CV) pulse circuitry.

Parameters: Enable — Set to TRUE or FALSE to enable or disable CV.

Return Value: None.

void sePwmControl(CLOCKSELECT ClkSource, int ClkDivide, int DutyCycle)

Description: This function sets up the Pulse Width Modulation (PWM) clock configuration registers.

Parameters:

- ClkSource — The clock source for PWM; set to either CLKI or CLKI2.
- ClkDivide — The clock source is divided by 2^ClkDivide. Legal values for ClkDivide are from 0 to 12 (decimal). For example, if ClkDivide is 3, the clock source is divided by 2^3=8.
- DutyCycle — The PWM clock duty cycle; values can be from 0 to 255. A value of 0 makes the PWM output always low, and a value of 255 makes the PWM output high for 255 out of 256 clock periods.

Return Value: None.

void seCvControl(CLOCKSELECT ClkSource, int ClkDivide, int BurstLength)

Description: This function sets up the Contrast Voltage (CV) pulse configuration registers.

Parameters:

- ClkSource — The clock source for CV; set to either CLKI or CLKI2.
- ClkDivide — The clock source is divided by 2^ClkDivide. Legal values for ClkDivide are from 0 to 12 (decimal). For example, if ClkDivide is 3, the clock source is divided by 2^3=8.
- BurstLength — The number of pulses generated in a single CV pulse burst. Legal values are from 1 to 256.

Return Value: None.

10.2.4 Register Access

The Register Access functions provide a convenient method of accessing the control registers of the S1D13706 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.
- The word and dword register functions will swap bytes if the endian of the host CPU differs from the S1D13706 (the S1D13706 is little-endian).

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 least significant byte of the return value is the byte read from the register.

unsigned seReadRegWord(DWORD Index)

Description: This routine reads two consecutive registers as a word and returns the value.

Parameters: Index — Offset to the first register to read.

Return Value: The least significant word of the return value is the word read from the S1D13706 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 S1D13706 registers.

void 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 first of four registers to be written to.
Value	The dword value to be written to the registers.

Return Value: None.

10.2.5 Memory Access

The Memory Access functions provide convenient method of accessing the display memory on an S1D13706 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.
- These functions will not swap bytes if the endian of the host CPU differs from the S1D13706 (the S1D13706 is little-endian).

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.

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.

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.

10.2.6 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 six bits of each byte.

void seWriteLutEntry(int Index, BYTE *pRGB)

Description: seWriteLutEntry() writes one lookup table entry to the specified index of the lookup table.

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 a 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)

Description: seReadLutEntry() reads one lookup table entry and returns the results in the byte array pointed to by pRGB.

Parameter:

Index	Offset to the lookup table entry to be read. (i.e. setting index to 2 returns the value of the third RGB 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)

Description: seWriteLut() writes one or more lookup table entries starting at offset zero.

These routines are intended to allow setting as many lookup table entries as the current color depth allows.

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.
Count	The number of lookup table entries to modify.

Return Value: None.

void seReadLut(BYTE *pRGB, int Count)

Description: seReadLut() 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.

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.

int seSetMode(unsigned BitsPerPixel)

Description: seSetMode() changes the color depth of the display and updates the appropriate LUT. Display memory is automatically released and then reallocated as necessary for the display resolution.

Note

seSetMode() was previously called seSetBitsPerPixel(). It is now recommended to call seSetMode() instead of seSetBitsPerPixel(). In addition, hardware display swapping is enabled or disabled, based on the requirements described in seEnableHardwareDisplaySwapping().

IMPORTANT

When the LCD color depth is changed, memory allocated for both the main window and sub-window display buffer is freed and the display buffer memory is reassigned. The application must redraw the main window display and re-initialize the sub-window (if used) and redraw the sub-window after calling seSetMode().

Parameters:

BitsPerPixel	The new color depth. BitsPerPixel can be one of the following: 1, 2, 4, 8, 16.

Return Value:

ERR_OK	Function completed successfully.
ERR_NOT_ENOUGH_MEMORY	There is insufficient free display memory for the given bits-per-pixel mode and display resolution.
ERR_FAILED	Function failed because of invalid BitsPerPixel.

void seUseMainWinImageForSubWin(void)

Description: This function instructs the HAL to use the image pointed to by the main window registers as the image to be used by the sub-window. The sub-window start address and sub-window line address offset registers are programmed accordingly.

Note

It is the responsibility of the caller to first free any memory used by the sub-window before calling this function.

Parameters: None.

Return Value: None.

unsigned seGetBitsPerPixel(void)

Description: seGetBitsPerPixel() returns the current color depth for the associated display surface.

Parameters: None.

Return Value: The color depth of the surface. This value will be 1, 2, 4, 8, or 16.

10.2.7 Virtual Display

int seVirtInit(DWORD Width, DWORD Height)
int seMainWinVirtInit(DWORD Width, DWORD Height)
int seSubWinVirtInit(DWORD Width, DWORD Height)
int seMainAndSubWinVirtInit(DWORD width, DWORD height)

Description: These functions prepare the S1D13706 to display 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 physical display is used as a window into the larger display memory area (display surface). Panning (right/left) and scrolling (up/down) are used to move the display in order to view the entire image a portion at a time.

seVirtInit() prepares the current active surface for a virtual image display. Memory is allocated based on width, height and the current color depth.

seMainWinVirtInit() initializes and allocates memory for the main window based on width and height and color depth.

seSubWinVirtInit() initializes and allocates memory for the sub-window based on current width and height and color depth.

seMainAndSubWinVirtInit() initializes and allocates one block of memory for both the main window and sub-window based on width and height and color depth.

Memory previously allocated for the given display surface is released then reallocated to the larger size.

Note

The width programmed may be larger than that requested in the respective function argument. This is to ensure that the value programmed into the address offset registers is a multiple of 4 bytes. For example, suppose seVirtInit(240, 320) is called in SwivelView 90° and at 1 bits-per-pixel. Since four bytes corresponds to 32 pixels in 1 bits-per-pixel mode, the width must be a multiple of 32. Since 240 is not a multiple of 32, the width is automatically changed to the next available multiple, which in this case is 256.

Parameters:

Width	The desired virtual width of the display in pixels. Width must be a multiple of the number of pixels contained in one dword of display memory. In other words, Width must be a multiple of 32 ÷ bits-per-pixel.
Height	The desired virtual height of the display in pixels. The HAL performs internal memory management to ensure that all display surfaces 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 seMainWinVirtPanScroll(DWORD x, DWORD y)
void seSubWinVirtPanScroll(DWORD x, DWORD y)
void seMainAndSubWinVirtPanScroll(DWORD x, DWORD y)

Description:

When displaying a virtual image the physical display 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.

seMainWinVirtPanScroll() and seSubWinVirtPanScroll() will pan and scroll the surface indicated in the function name.

seMainAndSubWinVirtPanScroll() will pan and scroll the surface which is used by both the main and sub-windows.

Note

Panning operations are limited to 32-bit boundaries; x must be a multiple of 32 ÷ bits-per-pixel.

Parameters:

x	The new x offset, in pixels, of the upper left corner of the display. x must be a multiple of 32 ÷ bits-per-pixel.
y	The new y offset, in pixels, of the upper left corner of the display.

Return Value: None.

10.2.8 Drawing

Functions in this category perform primitive drawing on the specified display surface. Supported drawing primitive include pixels, lines, rectangles, ellipses and circles.

All drawing functions are in relation to the given SwivelView mode. For example, co-ordinate (0, 0) is always the top left corner of the image, but this is physically in different corners of the panel depending on what SwivelView mode is selected.

void seSetPixel(long x, long y, DWORD Color)
void seSetMainWinPixel(long x, long y, DWORD Color)
void seSetSubWinPixel(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 seSetMainWinAsActiveSurface() and seSetSubWinAsActiveSurface() for information about changing the active surface.

Use seSetMainWinPixel() and seSetSubWinPixel() to set one pixel on the surface indicated in the function name.

If no memory was allocated to the surface, these functions return without writing to display memory.

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 1, 2, 4 and 8 bpp, display colors are derived from the lookup table values. The least significant byte of Color forms an index into the lookup table. At 16 bpp the lookup table is bypassed and each word of display memory forms the color to display. In this 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 seGetMainWinPixel(long x, long y)
DWORD seGetSubWinPixel(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 seSetMainWinAsActiveSurface() and seSetSubWinAsActiveSurface() for information about changing the active surface.

Use seGetMainWinPixel() and seGetSubWinPixel() 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.

If no memory was allocated to the surface, the return value is (DWORD) -1.

At 1, 2, 4 and 8 bpp, 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 16 bpp the lookup table is bypassed and each word of display memory form the color to display. In this 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 seDrawMainWinLine(long x1, long y1, long x2, long y2, DWORD Color)
void seDrawSubWinLine(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 seSetMainWinAsActiveSurface() and seSetSubWinAsActiveSurface() for information about changing the active surface.

Use seDrawMainWinLine() and seDrawSubWinLine() to draw a line on the surface referenced by the function name.

If no memory was allocated to the surface, these functions return without writing to display memory.

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, of 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 1, 2, 4 and 8 bpp, display colors are derived from the lookup table values. The least significant byte of Color is an index into the lookup table. At 16 bpp the lookup table is bypassed and each word of display memory forms the color to display. In this 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 seDrawMainWinRect(long x1, long y1, long x2, long y2, DWORD Color, BOOL SolidFill)
void seDrawSubWinRect(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 of the rectangle or to only draw the border.

Use seDrawRect() to draw a rectangle on the current active display surface. See seSetMainWinAsActiveSurface() and seSetSubWinAsActiveSurface() for information about changing the active surface.

Use seDrawMainWinRect() and seDrawSubWinRect() to draw a rectangle on the display surface indicated by the function name.

If no memory was allocated to the surface, these functions return without writing to display memory.

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 1, 2, 4 and 8 bpp, display colors are derived from the lookup table values. The least significant byte of Color is an index into the lookup table. At 16 bpp the lookup table is bypassed and each word of display memory forms the color to display. In this 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 to draw only the rectangle border. Set to TRUE 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 seDrawMainWinCircle(long xCenter, long yCenter, long Radius, DWORD Color)
void seDrawSubWinCircle(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. These functions only draw the border of the circle; there is no solid fill feature.

Use seDrawCircle() to draw the circle on the current active display surface. See seSetMainWinAsActiveSurface() and seSetSubWinAsActiveSurface() for information about changing the active surface.

Use seDrawMainWinCircle() and seDrawSubWinCircle() draw the circle on the display surface indicated by the function name

If no memory was allocated to the surface, these functions return without writing to display memory.

Parameters:

x	The X co-ordinate, in pixels, of the center of the circle.
y	The Y co-ordinate, in pixels, of 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 1, 2, 4 and 8 bpp display colors are derived from the lookup table values. The least significant byte of Color is an index into the lookup table. At 16 bpp the lookup table is bypassed and each word of display memory forms the color to display. In this 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 seDrawMainWinEllipse(long xc, long yc, long xr, long yr, DWORD Color)
void seDrawSubWinEllipse(long xc, long yc, long xr, long yr, DWORD Color)

Description:

These routines draw an ellipse on the screen in the specified color. The ellipse 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. These functions only draw the border of the ellipse; there is no solid fill feature.

Use seDrawEllipse() to draw the ellipse on the current active display surface. See seSetMainWinAsActiveSurface() and seSetSubWinAsActiveSurface() for information about changing the active surface.

Use seDrawMainWinEllipse() and seDrawSubWinEllipse() to draw the ellipse on the display surface indicated by the function name.

If no memory was allocated to the surface, these functions return without writing to display memory.

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 1, 2, 4 and 8 bpp display colors are derived from the lookup table values. The least significant byte of Color is an index into the lookup table. At 16 bpp the lookup table is bypassed and each word of display memory forms the color to display. In this mode the least significant word describes the color to draw the circle with in 5-6-5 RGB format.

Return Value:

None.

10.2.9 Register/Display Memory

The S5U13706 Evaluation Board utilizes 2M bytes of display memory address space. The S1D13706 contains 80K bytes of embedded SDRAM.

In order for an application to directly access the S1D13706 display memory and registers, the following two functions are provided.

DWORD seGetLinearDisplayAddress(void)

Description: This function returns the linear address for the start of physical display memory.

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 S1D13706 control registers.

Parameters: None.

Return Value: The return value is the linear address of the start of S1D13706 control registers. A linear address is a 32-bit offset, in CPU address space.

10.3 Porting LIBSE to a new target platform

Building Epson applications like a simple HelloApp for a new target platform requires the following:

- HelloApp code.
- 13706HAL library.
- LIBSE library which contains target specific code for embedded platforms.

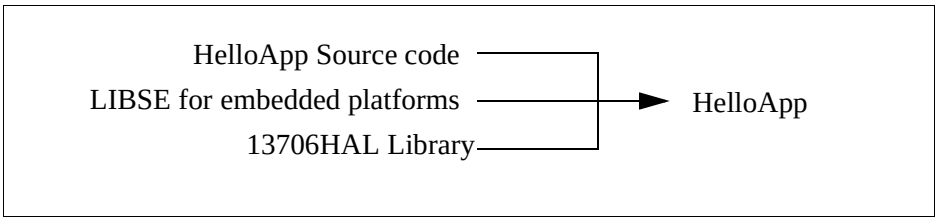


Figure 10-1: Components needed to build 13706 HAL application

For example, when building HELLOAPP.EXE for the x86 windows 32-bit platform, you need the HELLOAPP source files, the 13706HAL 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 32-bit windows .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 windows?

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 target specific code (like putch() and getch()) 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 13706HAL 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 S1D13706 utilities, including the makefiles, as well as a copy of the GNU Compiler v2.8.1 for Hitachi SH3. These are available on the Epson Electronics America Website at www.eea.epson.com.

10.3.1 Building the LIBSE library for SH3 target example

In the LIBSE files, there are two main types of files:

- C and assembler files that contain the target specific code.
- makefiles that describe the build process to construct the library.

The C and assembler files contain some platform setup code (evaluation board communications, chip selects) and jumps into the main entry point of the C code that is contained in the applications main() function. For our example, the startup file, which is sh3entry.c, performs some board configuration (board communications and assigning memory blocks with chip selects) and a jump into the applications main() function.

In the embedded targets, putch (xxxputch.c) and getch (xxxgetch.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.

To build the software for our target example, type the following at the root directory of the software (i.e. C:\13706).

make "TARGETS=SH3" "BUILDS=release"

11 Sample Code

Example source code demonstrating programming the S1D13706 using the HAL library is available on the internet at www.eea.epson.com.

REG[00h] Revision Code Register [1] — RO

Product Code = 001010						Revision Code = 00	
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0

REG[01h] Display Buffer Size Register — RW

Display Buffer Size							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[02h] Configuration Readback Register — RO

CNF7 Status	CNF6 Status	CNF5 Status	CNF4 Status	CNF3 Status	CNF2 Status	CNF1 Status	CNF0 Status

REG[04h] Memory Clock Configuration Register [2] — RW

n/a	n/a	MCLK Divide Select		n/a	n/a	n/a	Reserved
		bit 1	bit 0				

REG[05h] Pixel Clock Configuration Register [3,4] — RW

n/a	PCLK Divide Select			n/a	n/a	PCLK Source Select	
	Bit 2	Bit 1	Bit 0			Bit 1	Bit 0

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

LUT Blue Write Data							
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	n/a	n/a

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

LUT Green Write Data							
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	n/a	n/a

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

LUT Red Write Data							
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	n/a	n/a

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

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

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

LUT Blue Read Data							
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	n/a	n/a

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

LUT Green Read Data							
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	n/a	n/a

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

LUT Red Write Data							
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	n/a	n/a

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

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

REG[10h] Panel Type Register [5,6] — RW

Panel Data Format Select	Color/Mono Panel Select	Panel Data Width		Active Panel Res. Select	n/a	Panel Type	
		Bit 1	Bit 0			Bit 1	Bit 0

REG[11h] MOD Rate Register — RW

n/a	n/a	MOD Rate					
		bit 5	bit 4	bit 3	bit 2	bit1	bit 0

REG[12h] Horizontal Total Register — RW

n/a	Horizontal Total						
	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[14h] Horizontal Display Period Register — RW

n/a	Horizontal Display Period						
	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

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

Horizontal Display Period Start Position							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

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

n/a	n/a	n/a	n/a	n/a	n/a	Horizontal Display Period Start Position	
						bit 9	bit 8

REG[18h] Vertical Total Register 0 — RW

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

REG[19h] Vertical Total Register 1 — RW

n/a	n/a	n/a	n/a	n/a	n/a	Vertical Total	
						Bit 9	Bit 8

REG[1Ch] Vertical Display Period Register 0 — RW

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

REG[1Dh] Vertical Display Period Register 1 — RW

n/a	n/a	n/a	n/a	n/a	n/a	Vertical Display Period	
						Bit 9	Bit 8

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

Vertical Display Period Start Position							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[1Fh] Vertical Display Period Start Position Register 1 — RW

n/a	n/a	n/a	n/a	n/a	n/a	Vertical Display Period Start Position	
						bit 9	bit 8

REG[20h] FPLINE Pulse Width Register — RW

FPLINE Pulse Polarity	FPLINE Pulse Width						
	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[22h] FPLINE Pulse Start Position Register 0 — RW

FPLINE Pulse Start Position							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[23h] FPLINE Pulse Start Position Register 1 — RW

n/a	n/a	n/a	n/a	n/a	n/a	FPLINE Pulse Start Position	
						Bit 9	Bit 8

REG[24h] FPFRAME Pulse Width Register — RW

FPFRAME Pulse Polarity	n/a	n/a	n/a	n/a	FPFRAME Pulse Width		
					Bit 2	Bit 1	Bit 0

REG[26h] FPFRAME Pulse Start Position Register 0 — RW

FPFRAME Pulse Start Position							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[27h] FPFRAME Pulse Start Position Register 1 — RW

n/a	n/a	n/a	n/a	n/a	n/a	FPFRAME Pulse Start Position	
						Bit 9	Bit 8

REG[28h] D-TFD GCP Index Register — RW

n/a	n/a	n/a	D-TFD GCP Index				
			Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[2Ch] D-TFD GCP Data Register — RW

D-TFD GCP Data							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[70h] Display Mode Register [7] — RW

Display Blank	Dithering Disable	Hardware Video Invert Enable	Software Video Invert	n/a	Bit-per-pixel Select		
					Bit 2	Bit 1	Bit 0

REG[71h] Special Effects Register [8] — RW

Display Data Word Swap	Display Data Byte Swap	n/a	Sub-Window Enable	n/a	n/a	SwivelView™ Mode Select	
						Bit 1	Bit 0

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

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

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

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

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

n/a	n/a	n/a	n/a	n/a	n/a	n/a	Main Window Display Start Address Bit 16

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

Main Window Line Address Offset							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

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

n/a	n/a	n/a	n/a	n/a	n/a	Main Window Line Address Offset	
						Bit 9	Bit 8

REG[7Ch] Sub-Window Display Start Address Register 0 — RW

Sub-Window Display Start Address							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[7Dh] Sub-Window Display Start Address Register 1 — RW

Sub-Window Display Start Address							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

REG[7Eh] Sub-Window Display Start Address Register 2 — RW

n/a	n/a	n/a	n/a	n/a	n/a	n/a	Sub-Window Display Start Address Bit 16

REG[80h] Sub-Window Line Address Offset Register 0 — RW

Sub-Window Line Address Offset							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[81h] Sub-Window Line Address Offset Register 1 — RW

n/a	n/a	n/a	n/a	n/a	n/a	Sub-Window Line Address Offset	
						Bit 9	Bit 8

REG[84h] Sub-Window X Start Position Register 0 — RW

Sub-Window X Start Position							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[85h] Sub-Window X Start Position Register 1 — RW

n/a	n/a	n/a	n/a	n/a	n/a	Sub-Window X Start Position	
						Bit 9	Bit 8

REG[88h] Sub-Window Y Start Position Register 0 — RW

Sub-Window Y Start Position							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[89h] Sub-Window Y Start Position Register 1							RW
n/a	n/a	n/a	n/a	n/a	n/a	Sub-Window Y Start Position	
						Bit 9	Bit 8

REG[8Ch] Sub-Window X End Position Register 0							RW
Sub-Window X End Position							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[8Dh] Sub-Window X End Position Register 1							RW
n/a	n/a	n/a	n/a	n/a	n/a	Sub-Window X End Position	
						Bit 9	Bit 8

REG[90h] Sub-Window Y End Position Register 0							RW
Sub-Window Y End Position							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[91h] Sub-Window Y End Position Register 1							RW
n/a	n/a	n/a	n/a	n/a	n/a	Sub-Window Y End Position	
						Bit 9	Bit 8

REG[A0h] Power Save Configuration Register							RW
VNDP Status (RO)	n/a	n/a	n/a	Memory Controller Power Save Status (RO)	n/a	n/a	Power Save Mode Enable

REG[A1h] Reserved							RW
n/a	n/a	n/a	n/a	n/a	n/a	n/a	Reserved

REG[A2h] Software Reset Register							RW
Reserved	n/a	n/a	n/a	n/a	n/a	n/a	Software Reset (WO)

REG[A3h] Reserved							RW
Reserved	n/a	n/a	n/a	n/a	n/a	n/a	n/a

REG[A4h] Scratch Pad Register 0							RW
Scratch Pad							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[A5h] Scratch Pad Register 1							RW
Scratch Pad							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

REG[A8h] General Purpose IO Pins Configuration Register 0							RW
n/a	GPIO6 Pin IO Config	GPIO5 Pin IO Config	GPIO4 Pin IO Config	GPIO3 Pin IO Config	GPIO2 Pin IO Config	GPIO1 Pin IO Config	GPIO0 Pin IO Config

REG[A9h] General Purpose IO Pins Configuration Register 1							RW
GPIO Pin Input Enable	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

REG[ACh] General Purpose IO Pins Status/Control Register 0							RW
n/a	GPIO6 Pin IO Status	GPIO5 Pin IO Status	GPIO4 Pin IO Status	GPIO3 Pin IO Status	GPIO2 Pin IO Status	GPIO1 Pin IO Status	GPIO0 Pin IO Status

REG[ADh] General Purpose IO Pins Status/Control Register 1							RW
GPO Control	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

REG[B0h] PWM Clock / CV Pulse Control Register							RW
PWM Clock Force High	n/a	n/a	PWM Clock Enable	CV Pulse Force High	CV Pulse Burst Status (RO)	CV Pulse Burst Start	CV Pulse Enable

REG[B1h] PWM Clock / CV Pulse Configuration Register [9,10]							RW
PWM Clock Divide Select				CV Pulse Divide Select			PWMCLK Source Select
Bit 3	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	

REG[B2h] CV Pulse Burst Length Register								RW
CV Pulse Burst Length								
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

REG[B3h] PWMOUT Duty Cycle Register [11]								RW
PWMOUT Duty Cycle								
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Notes

1 REG[00h] These bits are used to identify the S1D13706. For the S1D13706, the product code should be 10.

2 REG[04h] Memory Clock Configuration Register

MCLK Divide Select Bits	BCLK to MCLK Frequency Ratio
00	1:1
01	2:1
10	3:1
11	4:1

3 REG[05h] Pixel Clock Configuration Register

PCLK Divide Select Bits	PCLK Source to PCLK Frequency Ratio
000	1:1
001	2:1
010	3:1
011	4:1
1XX	8:1

4 REG[05h] Pixel Clock Configuration Register

PCLK Source Select Bits	PCLK Source
00	MCLK
01	BCLK
10	CLKI
11	CLKI2

5 REG[10h] Panel Type Register

Panel Data Width Bits [1:0]	Passive LCD Panel Data Width Size	Active Panel Data Width Size
00	4-bit	9-bit
01	8-bit	12-bit
10	16-bit	18-bit
11	Reserved	Reserved

6 REG[10h] Panel Type Register

REG[10h] Bits[1:0]	Panel Type
00	STN
01	TFT
10	HR-TFT
11	D-TFD

7 REG[70h] Display Mode Register

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

8 REG[71h] Special Effects Register

SwivelView™ Mode Select Bits	SwivelView™ Orientation
00	Normal
01	90°
10	180°
11	270°

9 REG[B1h] PWM Clock / CV Pulse Configuration Register

PWM Clock Divide Select Bits [3:0]	PWM Clock Divide Amount
0h	1
1h	2
2h	4
3h	8
...	...
Ch	4096
Dh-Fh	Reserved

10 REG[B1h] PWM Clock / CV Pulse Configuration Register

CV Pulse Divide Select Bits [2:0]	CV Pulse Divide Amount
0h	1
1h	2
2h	4
3h	8
...	...
7h	128

11 REG[B3h] PWMOUT Duty Cycle Register

PWMOUT Duty Cycle [7:0]	PWMOUT Duty Cycle
00h	Always Low
01h	High for 1 out of 256 clock periods
02h	High for 2 out of 256 clock periods
...	...
FFh	High for 255 out of 256 clock periods

EPSON®

S1D13706 Embedded Memory LCD Controller

13706CFG Configuration Program

Document Number: X31B-B-001-03

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

13706CFG

13706CFG is an interactive Windows® 9x/ME/NT/2000 program that calculates register values for a user defined S1D13706 configuration. The configuration information can be used to directly alter the operating characteristics of the S1D13706 utilities or any program built with the Hardware Abstraction Layer (HAL) library. Alternatively, the configuration information can be saved in a variety of text file formats for use in other applications.

S1D13706 Supported Evaluation Platforms

13706CFG runs on PC system running Windows 9x/ME/NT/2000 and can modify the executable files for the following 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.

Installation

Create a directory for **13706cfg.exe** and the S1D13706 utilities. Copy the files **13706cfg.exe** and **panels.def** to that directory. **Panels.def** contains configuration information for a number of panels and must reside in the same directory as **13706cfg.exe**.

Usage

13706CFG can be started from the Windows desktop or from a Windows command prompt.

To start 13706CFG from the Windows desktop, double click the program icon or the link icon if one was created during installation.

To start 13706CFG from a Windows command prompt, change to the directory **13706cfg.exe** was installed to and type the command **13706cfg**.

The basic procedure for using 13706CFG is:

1. Start 13706CFG as described above.
2. Open an existing file to serve as a starting reference point (this step is optional).
3. Modify the configuration. For specific information on editing the configuration, see "13706CFG Configuration Tabs" on page 7.
4. Save the new configuration. The configuration information can be saved in two ways; as an ASCII text file or by modifying the executable image on disk.

   Several ASCII text file formats are supported. Most are formatted C header files used to build display drivers or standalone applications.

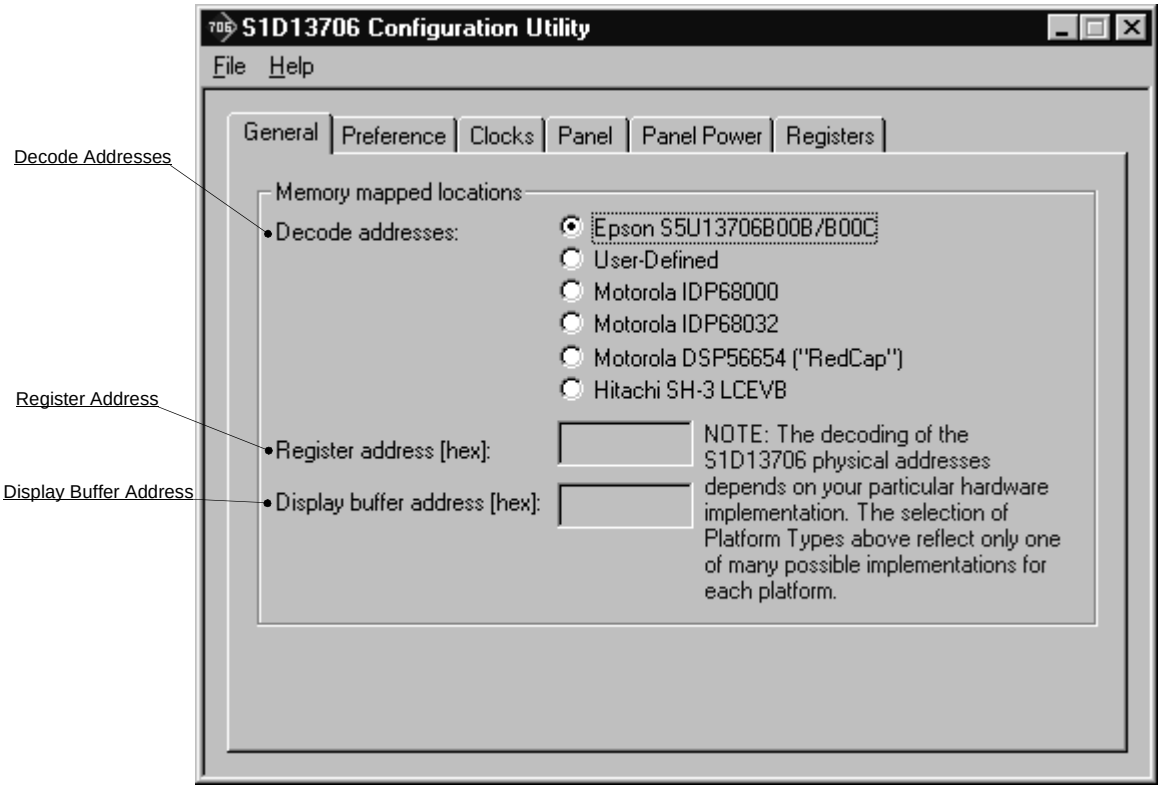
   Utility files based on the Hardware Abstraction Layer (HAL) can be modified directly by 13706CFG.

13706CFG Configuration Tabs

13706CFG provides a series of tabs which can be selected at the top of the main window. Each tab allows the configuration of a specific aspect of S1D13706 operation.

The tabs are labeled “General”, “Preference”, “Clocks”, “Panel”, “Panel Power”, and “Registers”. The following sections describe the purpose and use of each of the tabs.

General Tab



The General tab contains S1D13706 evaluation board specific information. The values presented are used for configuring HAL based executable utilities. The settings on this tab specify where in CPU address space the registers and display buffer are located.

Decode Addresses

Selecting one of the listed evaluation platforms changes the values for the “Register address” and “Display buffer address” fields. The values used for each evaluation platform are examples of possible implementations as used by the Epson S1D13706 evaluation board. If your hardware implementation differs from the addresses used, select the User-Defined option and enter the correct addresses for “Register address” and “Display buffer address”.

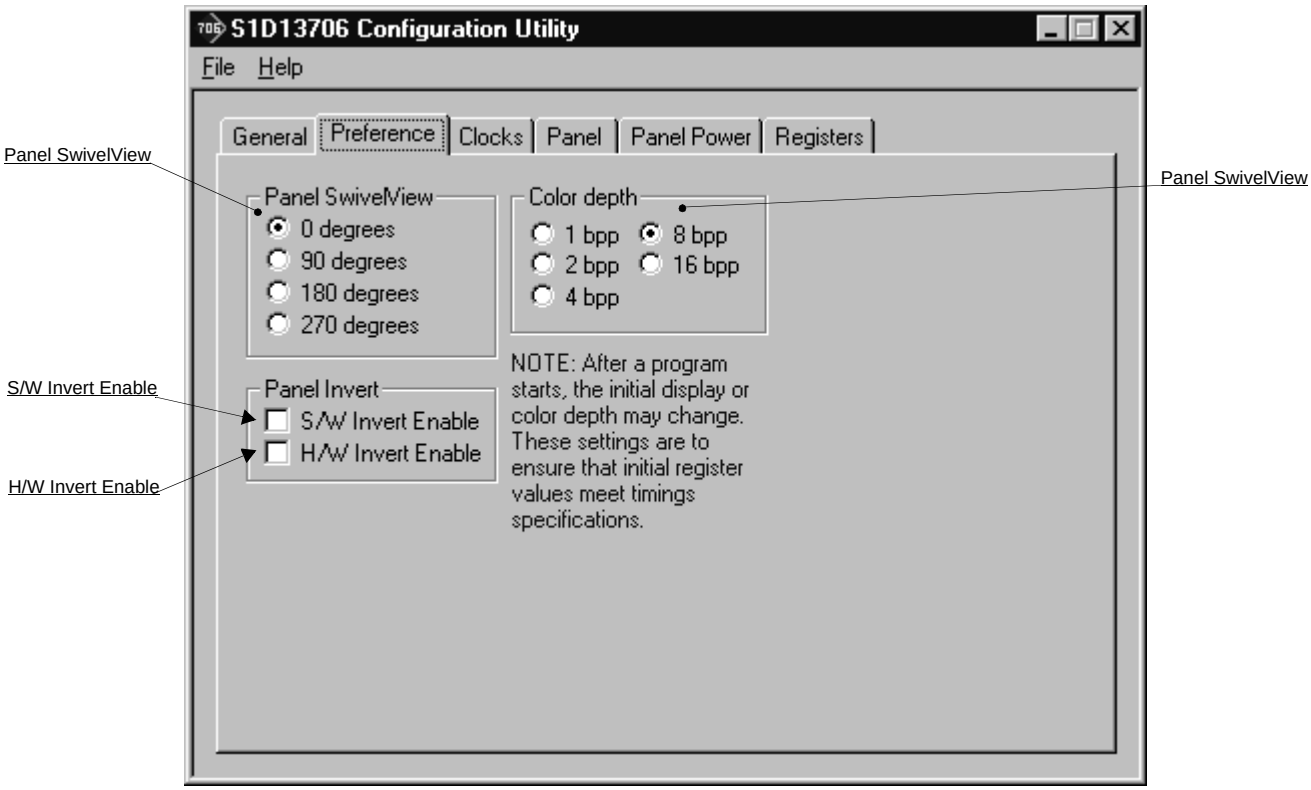
Register Address	The physical address of the start of register decode space (in hexadecimal). This field is automatically set according to the Decode Address unless the “User-Defined” decode address is selected.
Display Buffer Address	The physical address of the start of display buffer decode space (in hexadecimal). This field is automatically set according to the Decode Address unless the “User-Defined” decode address is selected.

Note

When “Epson S5U13706B00B/B00C Evaluation Board” is selected, the register and display buffer addresses are blanked because the evaluation board uses the PCI interface and the decode addresses are determined by the system BIOS during boot-up.

If using the S1D13706 Evaluation Board on a PCI based platform, both Windows and the **S1D13XXX** device driver must be installed. For further information on the S1D13xxx device driver, see the *S1D13XXX 32-bit Windows Device Driver Installation Guide*, document number X00A-E-003-xx.

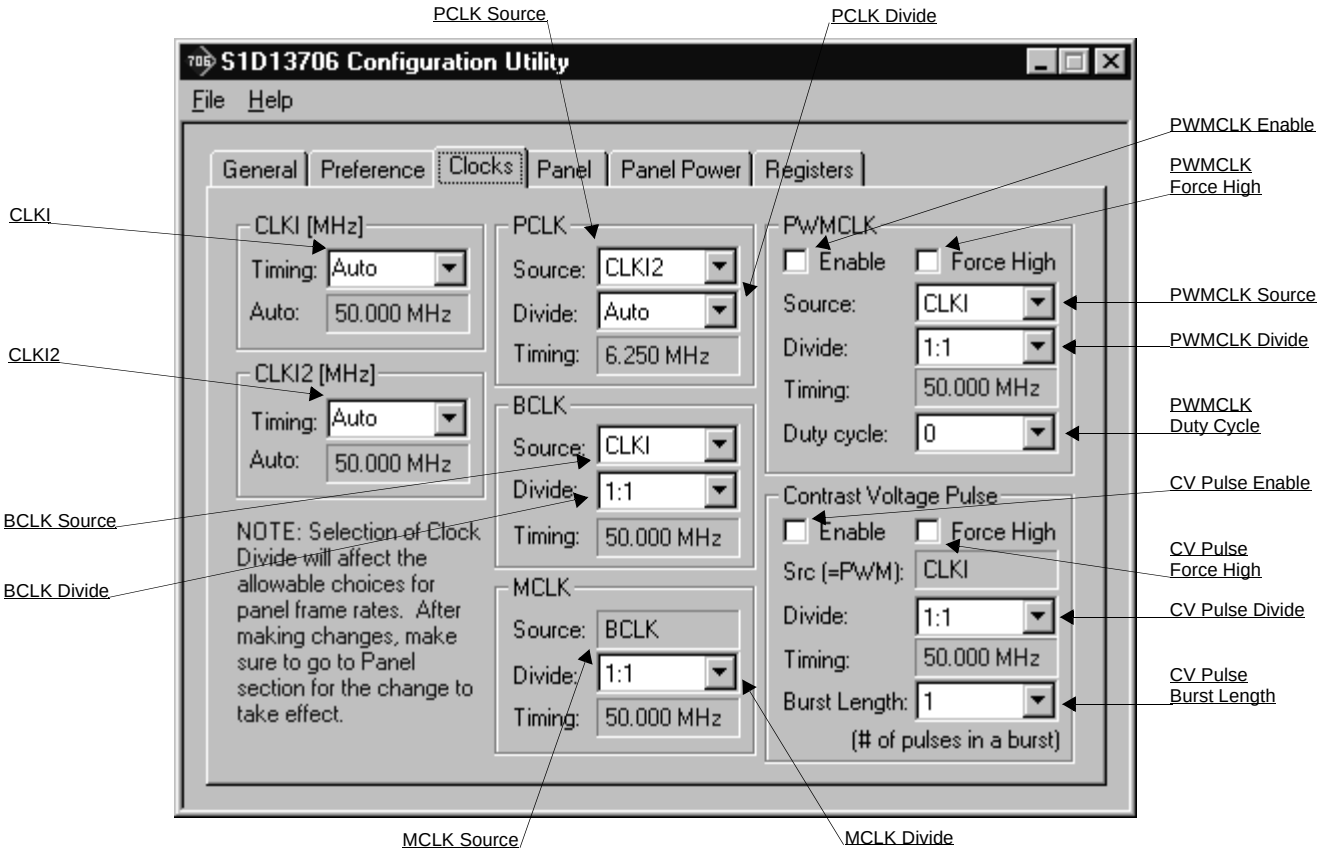
Preferences Tab



The Preference tab contains settings pertaining to the initial display state. During runtime these settings may be changed.

Panel SwivelView	The S1D13706 SwivelView feature is capable of rotating the image displayed on an LCD panel 90°, 180°, or 270° in a counter-clockwise direction. This sets the initial orientation of the panel.
Panel Invert	The S1D13706 can invert the display data going to the LCD panel. The display data is inverted after the Look-Up Table.
S/W Invert Enable	The Video Invert feature can be controlled by software using REG[70h] bit 4. When this box is checked, the Software Video Invert bit is set to one and video data is inverted. If the box is unchecked, the bit is set to zero and video data remains normal.
H/W Invert Enable	The Video Invert feature can be controlled by hardware if the GPIO0 is available. Hardware control is not possible if a HR-TFT or D-TFD panel is used as both panels use GPIO0 as an LCD control signal.
Panel Color Depth	Sets the initial color depth on the LCD panel.

Clocks Tab



The Clocks tab is intended to simplify the selection of input clock frequencies and the source of internal clocking signals. For further information regarding clocking and clock sources, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

In automatic mode the values for CLKI and CLKI2 are calculated based on selections made for LCD timings from the Panel tab. In this mode, the required frequencies for the input clocks are displayed in blue in the “Auto” section of each group. It is the responsibility of the system designer to ensure that the correct CLKI frequencies are supplied to the S1D13706.

Making a selection other than “Auto” indicates that the values for CLKI or CLKI2 are known and are fixed by the system design. Options for LCD frame rates are limited to ranges determined by the clock values.

Note

Changing clock values may modify or invalidate Panel settings. Confirm all settings on the Panel tab after modifying any clock settings.

The S1D13706 may use as many as two input clocks or as few as one. The more clocks used the greater the flexibility of choice in display type and memory speed.

CLKI

This setting determines the frequency of CLKI.

Select "Auto" to have the CLKI frequency determined automatically based on settings made on other configuration tabs. After completing the other configurations, the required CLKI frequency will be displayed in blue in the Auto section.

If the system design requires the CLKI frequency to be fixed at a particular rate, set this value by selecting a preset frequency from the drop down list or entering the desired frequency in MHz.

CLKI2

This setting determines the frequency of CLKI2.

Select "Auto" to have the CLKI2 frequency determined automatically based on settings made on other configuration tabs. After completing the other configurations, the required CLKI2 frequency will be displayed in blue in the Auto section.

If the system design requires the CLKI2 frequency to be fixed at a particular rate, set this value by selecting a preset frequency from the drop down list or entering the desired frequency in MHz.

PCLK

These settings select the clock signal source and divisor for the pixel clock (PCLK).

Source

Selects the PCLK source. Possible sources include CLKI, CLKI2, BCLK or MCLK.

Divide

Specifies the divide ratio for the clock source signal. The divide ratio is applied to the PCLK source to derive PCLK.

Selecting "Auto" for the divisor allows the configuration program to calculate the best clock divisor. Unless a very specific clocking is being specified, it is best to leave this setting on "Auto".

Timing

This field shows the actual PCLK used by the configuration process.

BCLK

These settings select the clock signal source and divisor for the bus interface clock (BCLK).

Source

The BCLK source is CLKI.

Divide

Specifies the divide ratio for the clock source signal. The divide ratio is applied to the BCLK source to derive BCLK.

Timing

This field shows the actual BCLK frequency used by the configuration process.

MCLK

These settings select the clock signal source and input clock divisor for the memory clock (MCLK). MCLK should be set as close to the maximum (50 MHz) as possible.

Source

The MCLK source is BCLK.

Divide

Specifies the divide ratio for the clock source signal. The divide ratio is applied to the MCLK source to derive MCLK.

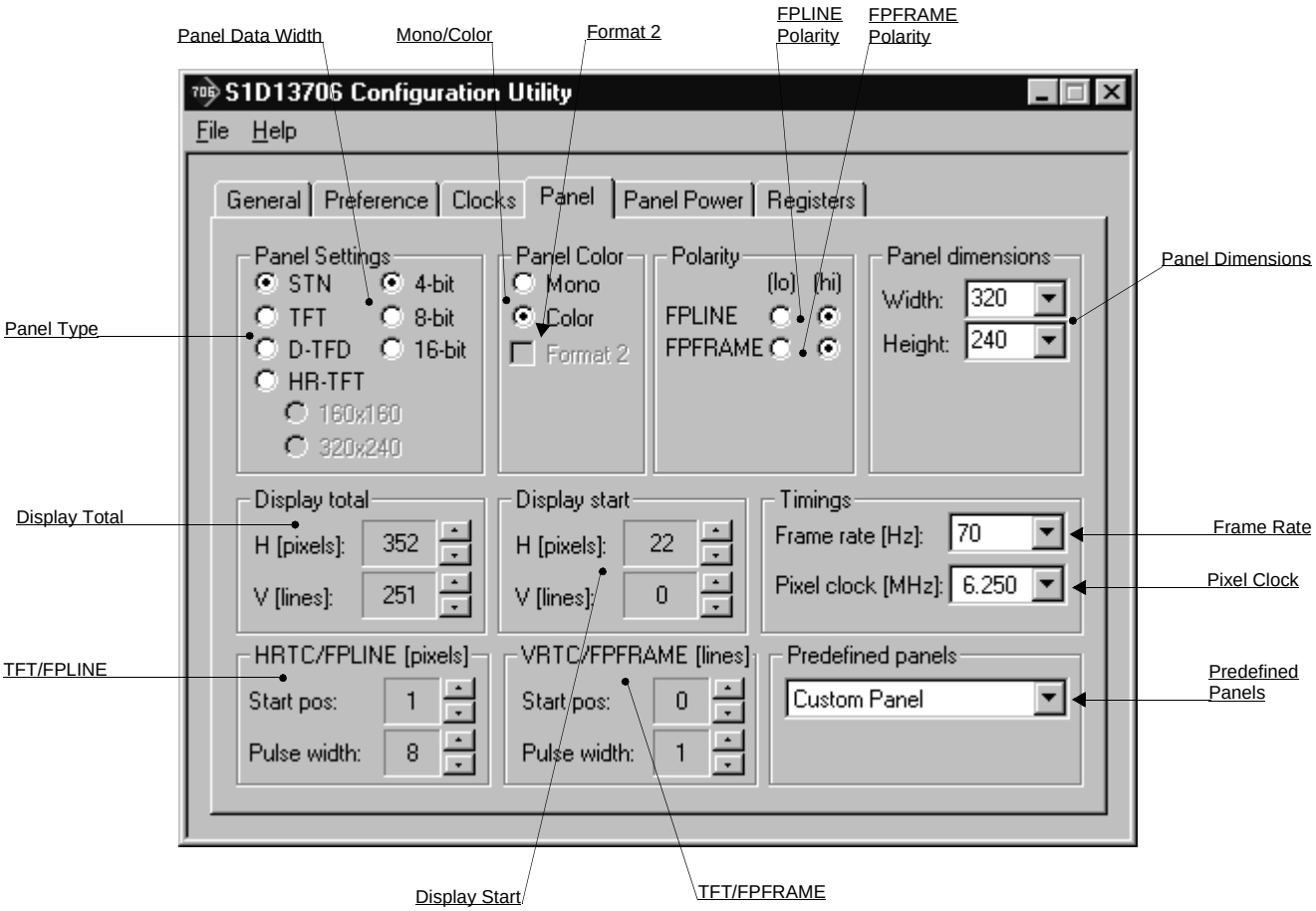
This divide ratio should be left at 1:1 unless the resultant MCLK is greater that 50MHz.

Timing

This field shows the actual MCLK frequency used by the configuration process.

PWMCLK	These controls configure various PWMCLK settings. The PWMCLK is the internal clock used by the Pulse Width Modulator for output to the panel.
Enable	When this box is checked, the PWMCLK circuitry is enabled.
Force High	The signal PWMOUT is forced high when this box is checked. PWMOUT is forced low when this box is not checked and Enable is not checked
Source	Selects the PWMCLK source. Possible sources include CLKI and CLKI2.
Divide	Specifies the divide ratio for the clock source signal. The divide ratio is applied to the PWMCLK source to derive PWMCLK.
Timing	This field shows the actual PWMCLK frequency used by the configuration process.
Duty Cycle	Selects the number of cycles that PWMOUT is high out of 256 clock periods.
Contrast Voltage Pulse	These controls configure various Contrast Voltage (CV) Pulse settings. The CV Pulse is provided for panels which support the contrast voltage function.
Enable	When this box is checked, the CV Pulse circuitry is enabled.
Force High	The signal CVOUT is forced high when this box is checked. CVOUT is forced low when this box is not checked and CVOUT is not enabled.
Source	The CV Pulse uses the same source clock as the PWMCLK.
Divide	Specifies the divide ratio for the clock source signal. The divide ratio is applied to the CVOUT Pulse clock source to derive the CV Pulse clock frequency.
Timing	This field shows the actual CV Pulse frequency used by the configuration process.
Burst Length	The number of pulses generated in a single CV Pulse burst.

Panel Tab



The S1D13706 supports many panel types. This tab allows configuration of most panel settings such as panel dimensions, type and timings.

Panel Type

Selects between passive (STN) and active (TFT/D-TFD/HR-TFT) panel types.

Several options may change or become unavailable when the STN/TFT setting is switched. Therefore, confirm all settings on this tab after the Panel Type is changed.

Panel Data Width

Selects the panel data width. Panel data width is the number of bits of data transferred to the LCD panel on each clock cycle and shouldn't be confused with color depth which determines the number of displayed colors.

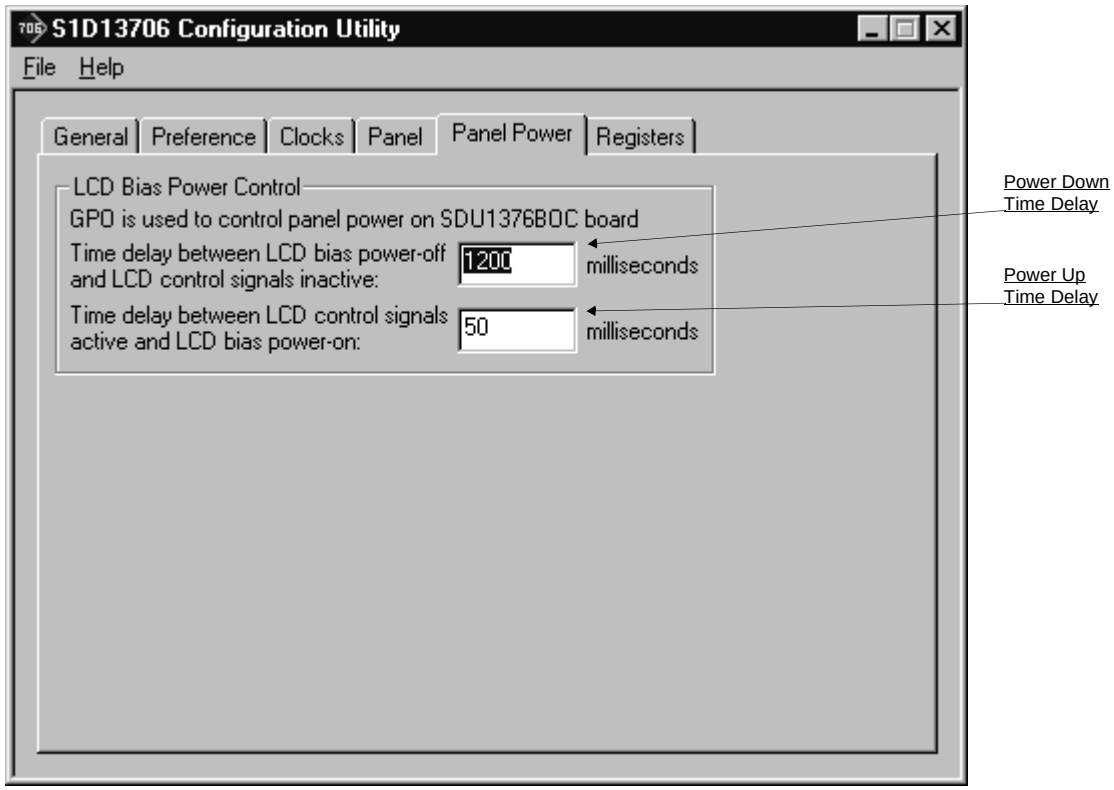
When the panel type is STN, the available options are 4, 8, and 16 bit. When an active panel type is selected the available options are 9, 12, and 18 bit.

Mono / Color	Selects between a monochrome or color panel.
Format 2	Selects color STN panel format 2. This option is specifically for configuring 8-bit color STN panels. See the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx, for description of format 1 / format 2 data formats. Most new panels use the format 2 data format.
FPLINE Polarity	Selects the polarity of the FPLINE pulse. Refer to the panel specification for the correct polarity of the FPLINE pulse.
FPFRAME Polarity	Selects the polarity of the FPFRAME pulse. Refer to the panel specification for the correct polarity of the FPFRAME pulse.
Panel Dimensions	These fields specify the panel width and height. A number of common widths and height are available in the selection boxes. If the width/height of your panel is not listed, enter the actual panel dimensions into the edit field. For passive panels, manually entered pixel widths must be a minimum of 32 pixels and can be increased by multiples of 16. For TFT panels, manually entered pixel widths must be a minimum of 16 pixels and can be increased by multiples of 8. If a value is entered that does not match these requirements, a notification box appears and 13706CFG rounds up the value to the next allowable width.
Display Total	It is recommended that these automatically generated Display Total values be used without adjustment. However, manual adjustment may be useful in fine tuning the horizontal and vertical display totals. The display total equals the display period plus the non-display period.
Display Start	It is recommended that these automatically generated Display Start values be used without adjustment. However, manual adjustment may be useful in fine tuning the horizontal and vertical display start positions. For passive panels these values must always be 0.

Frame Rate	Select the desired frame rate (in Hz) from the drop-down list. The values in the list are the range of possible frame rates using the currently selected pixel clock. To change the range of frame rates, select a different Pixel Clock rate (in MHz). Panel dimensions are fixed therefore frame rate can only be adjusted by changing either PCLK or non-display period values. Higher frame rates correspond to smaller horizontal and vertical non-display values, or higher frequencies.
Pixel Clock	Select the desired Pixel Clock (in MHz) from the drop-down list. The range of frequencies displayed is dependent on settings selected on the Clocks tab. For example: If CLKI is chosen to be Auto and PCLK is sourced from CLKI on the Clocks tab, then the range for Pixel Clock will range from 1.5 MHz to 80 MHz. Selecting a fixed PCLK on the Clocks tab, say 25.175 MHz, will result in only four selections: 6.293, 8.392, 12.587, and 25.175 MHz. (these frequencies represent the four possible frequencies from a fixed 25.175 MHz input clock divided by the PCLK divider).
TFT/FPLINE (pixels)	These settings allow fine tuning of the TFT line pulse parameters and are only available when the selected panel type is TFT/D-TFD/HR-TFT. Refer to *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx for a complete description of the FPLINE pulse settings. Start pos - Specifies the delay (in pixels) from the start of the horizontal non-display period to the leading edge of the FPLINE pulse. Pulse Width - Specifies the delay (in pixels) from the start of the horizontal non-display period to the leading edge of the FPLINE pulse.

TFT/FPFRAME (lines)	These settings allow fine tuning of the TFT frame pulse parameters and are only available when the selected panel type is TFT/D-TFD/HR-TFT. Refer to *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx, for a complete description of the FPFRAME pulse settings. Start pos - Specify the delay (in lines) from the start of the vertical non-display period to the leading edge of the FPFRAME pulse. Pulse width - Specifies the pulse width (in lines) of the FPFRAME output signal.
Predefined Panels	13706CFG uses a file (**panels.def**) which lists various panel manufacturers recommended settings. If the file **panels.def** is present in the same directory as **13706cfg.exe**, the settings for a number of predefined panels are available in the drop-down list. If a panel is selected from the list, 13706CFG loads the predefined settings contained in the file.

Panel Power Tab



The S5U13706B00C evaluation board is designed to use the GPO signal to control the LCD bias power. The following settings allow configuration of the necessary delays.

Power Down Time Delay

This setting controls the time delay between when the LCD panel is powered-off and when the S1D13706 control signals are turned off. This setting must be configured according to the specification for the panel being used.

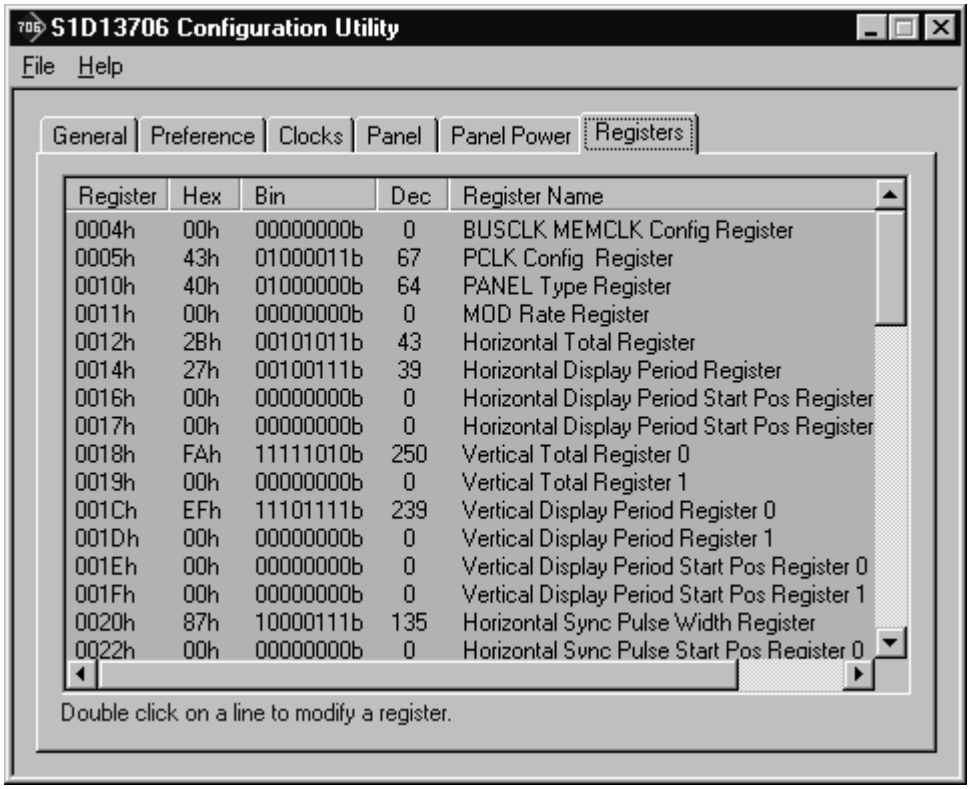
This value is only used by Epson evaluation software designed for the S5U13706B00C evaluation board.

Power Up Time Delay

This setting controls the time delay between when the S1D13706 control signals are turned on and the LCD panel is powered-on. This setting must be configured according to the specification for the panel being used.

This value is only used by Epson evaluation software designed for the S5U13706B00C evaluation board.

Registers Tab



The Registers tab allows viewing and direct editing the S1D13706 register values.

Scroll up and down the list of registers and view their configured value based on the settings the previous tabs. Individual register settings may be changed by double-clicking on the register in the listing. **Manual changes to the registers are not checked for errors, so caution is warranted when directly editing these values.** It is strongly recommended that the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx be referred to before making an manual register settings.

Manually entered values may be changed by 13706CFG if further configuration changes are made on the other tabs. In this case, the user is notified.

Note

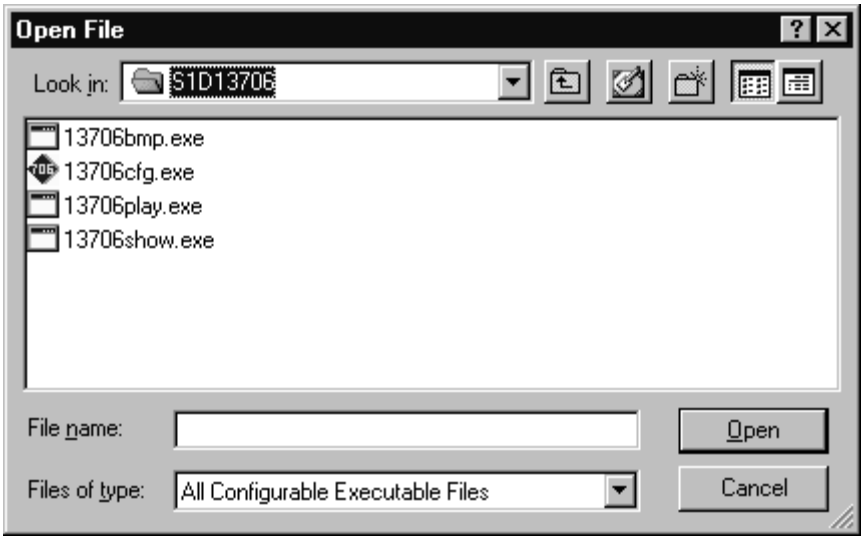
Manual changes to the registers may have unpredictable results if incorrect values are entered.

13706CFG Menus

The following sections describe each of the options in the File and Help menus.

Open...

From the Menu Bar, select “File”, then “Open...” to display the Open File Dialog Box.



The Open option allows 13706CFG to open files containing HAL configuration information. When 13706CFG opens a file it scans the file for an identification string, and if found, reads the configuration information. This may be used to quickly arrive at a starting point for register configuration. The only requirement is that the file being opened must contain a valid S1D13706 HAL library information block.

13706CFG supports a variety of executable file formats. Select the file type(s) 13706CFG should display in the Files of Type drop-down list and then select the filename from the list and click on the Open button.

Note

13706CFG is designed to work with utilities programmed using a given version of the HAL. If the configuration structure contained in the executable file differs from the version 13706CFG expects the Open will fail and an error message is displayed. This may happen if the version of 13706CFG is substantially older, or newer, than the file being opened.

Save

From the Menu Bar, select “File”, then “Save” to initiate the save action. The Save menu option allows a fast save of the configuration information to a file that was opened with the Open menu option.

Note

This option is only available once a file has been opened.

Note

13706cfg.exe can be configured by making a copy of the file 13706cfg.exe and configuring the copy. It is not possible to configure the original while it is running.

Save As...

From the Menu Bar, select “File”, then “Save As...” to display the Save As Dialog Box.

“Save as” is very similar to Save except a dialog box is displayed allowing the user to name the file before saving.

Using this technique a tester can configure a number of files differing only in configuration information and name (e.g. BMP60Hz.EXE, BMP72Hz.EXE, BMP75Hz.EXE where only the frame rate changes in each of these files).

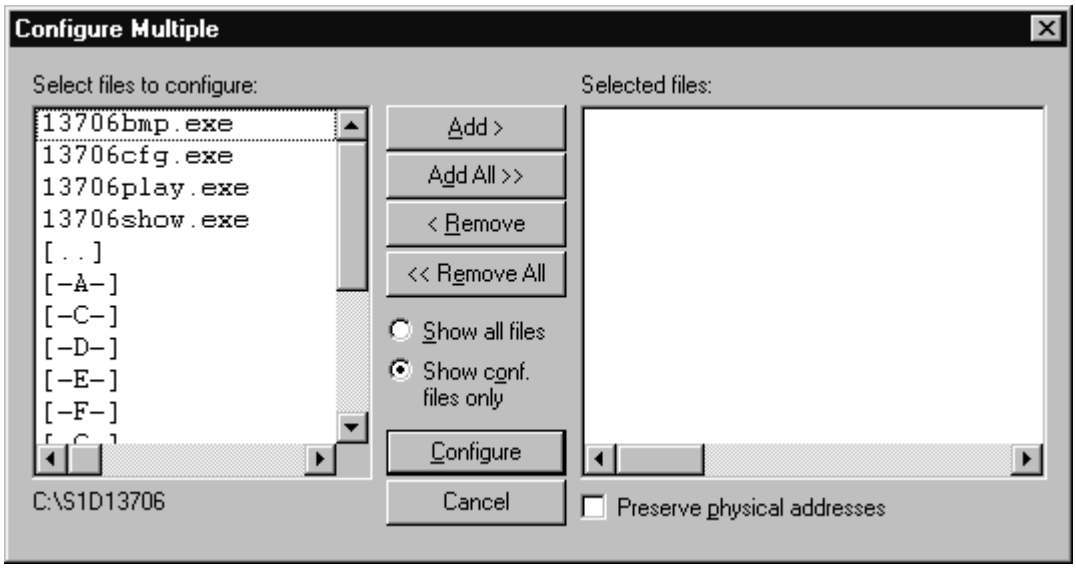
Note

When “Save As” is selected then an exact duplicate of the file as opened by the “Open” option is created containing the new configuration information.

Configure Multiple

After determining the desired configuration, “Configure Multiple” allows the information to be saved into one or more executable files built with the HAL library.

From the Menu Bar, select “File”, then “Configure Multiple” to display the Configure Multiple Dialog Box.This dialog box is also displayed when a file(s) is dragged onto the 13706CFG window.



The left pane lists files available for configuration; the right pane lists files that have been selected for configuration. Files can be selected by clicking the “Add” or “Add All” buttons, double clicking any file in the left pane, or by dragging the file(s) from Windows Explorer.

Selecting “Show all files” displays all files in the selected directory, whereas selecting “Show conf. files only” will display only files that can be configured using 13706CFG (i.e. .exe, .s9, .elf).

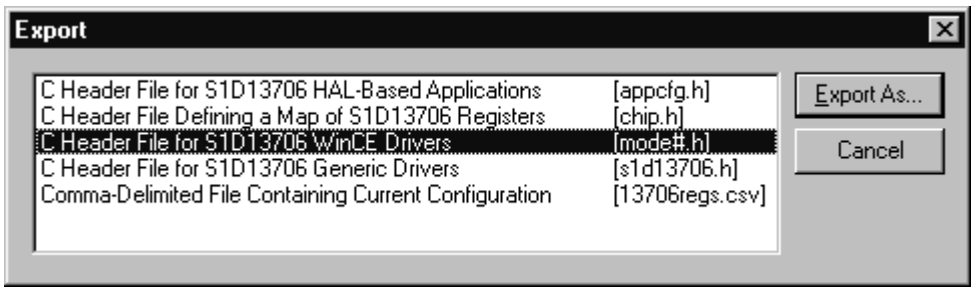
The configuration values can be saved to a specific EXE file for Intel platforms, or to a specific S9 or ELF file for non-Intel platforms. The file must have been compiled using the 13706 HAL library.

Checking “Preserve Physical Addresses” instructs 13706CFG to use the register and display buffer address values the files were previously configured with. Addresses specified in the General Tab are discarded. This is useful when configuring several programs for various hardware platforms at the same time. For example, if configuring PCI, MPC and IDP based programs at the same time for a new panel type, the physical addresses for each are retained. This feature is primarily intended for the test lab where multiple hardware configurations exist and are being tested.

Export

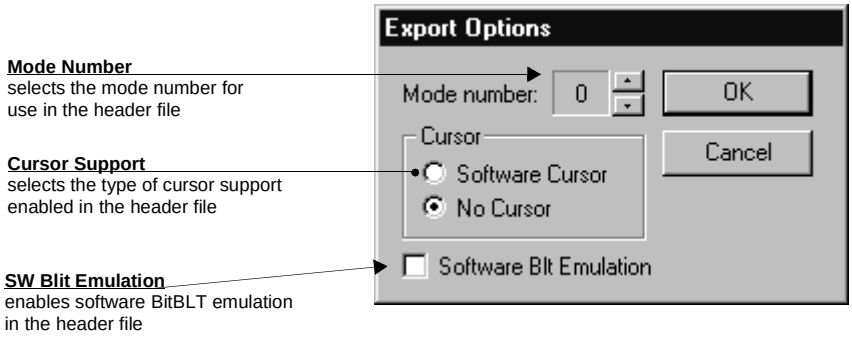
After determining the desired configuration, “Export” permits the user to save the register information as a variety of ASCII text file formats. The following is a list and description of the currently supported output formats:

- a C header file for use in writing HAL library based applications.
- a C header file which lists each register and the value it should be set to.
- a C header file for use in developing Window CE display drivers.
- a C header file for use in developing display drivers for other operating systems such as Linux, QNX, and VxWorks UGL or WindML.
- a comma delimited text file containing an offset, a value, and a description for each S1D13706 register.



After selecting the file format, click the “Export As..." button to display the file dialog box which allows the user to enter a filename before saving. Before saving the configuration file, clicking the “Preview” button starts Notepad with a copy of the configuration file about to be saved.

When the **C Header File for S1D13706 WinCE Drivers** option is selected as the export type, additional options are available and can be selected by clicking on the Options button. The options dialog appears as:



Enable Tooltips

Tooltips provide useful information about many of the items on the configuration tabs. Placing the mouse pointer over nearly any item on any tab generates a popup window containing helpful advice and hints.

To enable/disable tooltips check/uncheck the “Tooltips” option form the “Help” menu.

Note
Tooltips are enabled by default.

ERD on the Web

This “Help” menu item is actually a hotlink to the Epson Research and Development website. Selecting “Help” then “ERD on the Web” starts the default web browser and points it to the ERD product web site.

The latest software, drivers, and documentation for the S1D13706 is available at this website.

About 13706CFG

Selecting the “About 13706CFG” option from the “Help” menu displays the about dialog box for 13706CFG. The about dialog box contains version information and the copyright notice for 13706CFG.

Comments

- On any tab particular options may be grayed out if selecting them would violate the operational specification of the S1D13706 (i.e. Selecting TFT or STN on the Panel tab enables/disables options specific to the panel type).
- The file **panels.def** is a text file containing operational specifications for several supported, and tested, panels. This file can be edited with any text editor.
- 13706CFG allows manually altering register values. The manual changes may violate memory and LCD timings as specified in the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx. If this is done, unpredictable results may occur. Epson Research and Development, Inc. does not assume liability for any damage done to the display device as a result of configuration errors.

EPSON®

S1D13706 Embedded Memory LCD Controller

13706SHOW Demonstration Program

Document Number: X31B-B-002-03

THIS PAGE LEFT BLANK

13706SHOW

13706SHOW is designed to demonstrate and test some of the S1D13706 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. 13706SHOW supports SwivelView™ (90°, 180°, and 270° hardware rotation of the display image).

The 13706SHOW demonstration program must be configured and/or compiled to work with your hardware platform. The utility 13706CFG.EXE can be used to configure 13706SHOW. For further information on 13706CFG, refer to the *13706CFG Users Manual*, document number X31B-B-001-xx.

This software is designed to work in both embedded and personal computer (PC) environments. For the embedded environment, it is assumed that the system has a means of downloading software from the PC to the target platform. Typically this is done by serial communications, where the PC uses a terminal program to send control commands and information to the target processor. Alternatively, the PC can program an EPROM, which is then placed in the target platform. Some target platforms can also communicate with the PC via a parallel port connection, or an Ethernet connection.

S1D13706 Supported Evaluation Platforms

13706SHOW supports the following S1D13706 evaluation platforms:

- PC system with an Intel 80x86 processor running Windows® 9x/NT.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- MC68030IDP (Integrated Development Platform) board, revision 3.0, with a Motorola MC68030 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.
- DSP56654ADS (Applications Development System) board, with a Motorola REDCAP2 processor.

Note
The 13706SHOW source code can be modified or recompiled to allow 13706SHOW to run on other evaluation platforms not listed above.

Installation

PC platform:

Copy the file **13706show.exe** to a directory specified in the path (e.g. PATH=C:\13706).

Embedded platform:

Download the program **13706show** to the system.

Usage

PC Platform

At the prompt, type:

13706SHOW [/a] [/bigmem] [b=n] [/g[=n]] [/noclkerr] [/noinit] [/r90 | /r180 | /r270] [/read] [/s] [/write] [/?]

Embedded platform

Execute **`13706show`** and type the command line argument at the prompt.

Where:

/a	Cycles through all video modes automatically.
b=n	Shows the LCD display at a user specified color depth (bpp) where n = (1, 2, 4, 8, 16).
/g=n	Shows the image overlaid with a 20 pixel wide grid where n = white(0) or black(1). If n is not specified, the grid defaults to white.
/noinit	Skips full register initialization. Only registers used for changing the color depth (bpp) are updated. Additionally, some registers are read to determine information such as display size.
/r90	Enables SwivelView 90° mode, counter-clockwise hardware rotation of LCD image by 90 degrees.
/r180	Enables SwivelView 180° mode, counter-clockwise hardware rotation of LCD image by 180 degrees.
/r270	Enables SwivelView 270° mode, counter-clockwise hardware rotation of LCD image by 270 degrees.
/s	Displays a vertical stripe pattern.

/?	Displays the help screen.

Test Only Switches
(The following switches were added for testing and validation. They are not supported at the customer level.)

/bigmem	Assumes memory size is 2M bytes instead of 80K bytes **(for testing purposes only)**.
/noclkerr	Allows invalid SwivelView clock settings **(for testing purposes only)**.
/read	After drawing the image, continually reads the entire display buffer in dword increments **(for testing purposes only)**.
/write	Continually writes to one word of offscreen memory **(for testing purposes only)**.

Note
Pressing the *Esc* key will exit the program.

13706SHOW Examples

13706SHOW is designed to demonstrate and test some of the features of the S1D13706. The following examples show how to use the program in both instances.

Using 13706SHOW For Demonstration

1. To show color patterns which must be manually stepped through, type the following:

   13706SHOW

   The program displays the default color depth as selected by 13706CFG. Press any key to go to the next screen. Once all screens are shown the program exits. To exit the program immediately press the *Esc* key.

2. To show color patterns which automatically step through, type the following:

   13706SHOW /a

   The program displays the default color depth as selected by 13706CFG. 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, type the following:

   13706SHOW b=[mode]

   where:
   mode = 1, 2, 4, 8, or 16

   The program displays the requested color depth 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 for 1, 2, 4 and 8 bpp color depths. For 16 bpp color depths the green component of the pixel value is used.

4. To show the color patterns in SwivelView 90° mode, type the following:

   13706SHOW /r90

   The program displays the default color depth as selected by 13706CFG. Press any key to go to the next screen. 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.

5. To show solid vertical stripes, type the following:

   13706SHOW /s

   The program displays the default color depth as selected by 13706CFG. 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.

Comments

- If 13706SHOW is started without specifying the color depth (b=), 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 13706SHOW using 13706CFG. This approach avoids showing color depths not supported by a given hardware configuration.
- 13706SHOW cannot show a greater color depth than the display device allows.

THIS PAGE LEFT BLANK

EPSON®

S1D13706 Embedded Memory LCD Controller

13706PLAY Diagnostic Utility

Document Number: X31B-B-003-02

THIS PAGE LEFT BLANK

13706PLAY

13706PLAY is a diagnostic utility allowing a user to read/write to all the S1D13706 registers, Look-Up Tables and display buffer. 13706PLAY 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).

13706PLAY 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 13706PLAY diagnostic utility must be configured and/or compiled to work with your hardware platform. The program 13706CFG.EXE can be used to configure 13706PLAY. For further information on 13706CFG, refer to the *13706CFG Users Manual*, document number X31B-B-001-xx.

This software is designed to work in both embedded and personal computer (PC) environments. For the embedded environment, it is assumed that the system has a means of downloading software from the PC to the target platform. Typically this is done by serial communications, where the PC uses a terminal program to send control commands and information to the target processor. Alternatively, the PC can program an EPROM, which is then placed in the target platform. Some target platforms can also communicate with the PC via a parallel port connection, or an Ethernet connection.

S1D13706 Supported Evaluation Platforms

13706PLAY supports the following S1D13706 evaluation platforms:

- PC with an Intel 80x86 processor running Windows® 9x/NT.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- MC68030IDP (Integrated Development Platform) board, revision 3.0, with a Motorola MC68030 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.
- DSP56654ADS (Applications Development System) board, with a Motorola REDCAP2 processor.

Note
The 13706PLAY source code can be modified or recompiled to allow 13706PLAY to run on other evaluation platforms not listed above.

Installation

PC platform

Copy the file **13706play.exe** to a directory in the path (e.g. PATH=C:\S1D13706).

Embedded platform

Download the program **13706play** to the system.

Usage

PC platform

At the prompt, type:

```
13706play [/?]
```

Where:

/? displays copyright and program version information.

Embedded platform

Execute **`13706play`** and at the prompt, type the command line argument **/?**.

Where:

/? displays copyright and program version information.

Commands

The following commands are designed to be used from within the 13706PLAY 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. **13706play "f 0 14000 AB" q**).

Note

If the endian mode of the host platform is big endian, reading/writing words and dwords to/from the registers and display buffer may be incorrect. It may be necessary for the user to manually swap the bytes in order to perform the IO correctly. For further information on little/big endian and the S1D13706 byte/word swapping capabilities, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

CLKI [?] iFreq

Selects a preset clock frequency (MHz) for CLKI. If the "?" option is used, the list of available frequencies for CLKI is displayed.

Where:

?	Displays a list of available frequencies for CLKI (MHz).
iFreq	Sets CLKI to an index representing a preset frequency (MHz) specified by iFreq. iFreq is based on the table provided with the command **CLKI ?**.

Note

The CLKI command programs preset frequencies available on the S5U13706B00C evaluation board. This function is not designed for use on other evaluation platforms or prototype designs.

CLKI2 [?] iFreq

Selects a preset clock frequency (MHz) for CLKI2. If the "?" option is used, the list of available frequencies for CLKI2 is displayed.

Where:

?	Displays a list of available frequencies for CLKI2 (MHz).
iFreq	Sets CLKI2 to an index representing a preset frequency (MHz) specified by iFreq. iFreq is based on the table provided with the command **CLKI2 ?**.

Note

The CLKI2 command programs preset frequencies available on the S5U13706B00C evaluation board. This function is not designed for use on other evaluation platforms or prototype designs.

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 addr1 addr2 data...

Fills a specified address range with 8-bit data (bytes).

Where:

addr1	Start address of the range to be filled (hex).
addr2	End address of the range to be filled (hex).
data	Data to be written (hex). Data can be a list of bytes to 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 addr1 addr2 data...

Fills a specified address range with 32-bit data (dwords).

Where:

addr1	Start address of the range to be filled (hex).
addr2	End address of the range to be filled (hex).
data	Data to be written (hex). Data can be a list of dwords to 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 addr1 addr2 data...

Fills a specified address range with 16-bit data (words).

Where:

addr1	Start address of the range to be filled (hex).
addr2	End address of the range to be filled (hex).
data	Data to be written (hex). Data can be a list of words to 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 are displayed at a time. The display is halted after the specified number of lines. Setting the number of lines to 0 disables the halt function and allows the data to continue displaying until all data has been shown.

Where:

lines	Number of lines that are shown before halting the displayed data (decimal value).

I

Initializes the S1D13706 registers with the default register settings as configured by the utility 13706CFG. To initialize the S1D13706 with different register values, reconfigure 13706PLAY using 13706CFG. For further information on 13706CFG, see the *13706PLAY User Manual*, document number X31B-B-001-xx.

L 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:

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 7-2 of each color are used in the LUT. For example, 04h is the first color intensity after 00h. Valid LUT colors follow the pattern 00h, 04h, ..., FCh.

LA
Reads all LUT values.

Note
Only bits 7-2 of each color are used in the LUT. For example, 04h is the first color intensity after 00h. Valid LUT colors follow the pattern 00h, 04h, ..., FCh.

M [bpp]
Sets the color depth (bpp). If no color depth is provided, information about the current settings are listed.
Where:

bpp	Color depth to be set (1/2/4/8/16 bpp).

Q
Quits the program.

P [on | off]
Controls the power on/off state of the S1D13706.
Where:

on	Powers on the chip.
off	Powers off the chip.

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) are 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) are 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) are read (hex).
count	Number of words to be read (hex).

T xx
Tests VNDP read for xx seconds. **(This option was developed for testing purposes only and is not supported.)**
Where:

xx	The number of seconds VNDP is tested (decimal).

V
Calculates the current frame rate 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 to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

WD addr data ...
Writes dword(s) of data to specified memory address.
Where:

addr	Address data is written to
data	Data to be written (hex). Data can be a list of dwords to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

WW addr data ...
Writes word(s) of data to specified memory address.
Where:

addr	Address data is written to
data	Data to be written (hex). Data can be a list of words to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

X index [data]
Writes byte data to the register at index. If no data is specified, reads the 8-bit (byte) data from the register at index.
Where:

index	Index into the registers (hex).
data	Data to be written to/read from register (hex). Data can be a list of bytes to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

XA
Reads all the S1D13706 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 to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

XW index [data]
Writes word data to the register at index. If no data is specified, reads the 16-bit (word) data from the register at index.
Where:

index	Index into the registers (hex).
data	Data to be written to/read from register (hex). Data can be a list of words to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

?
Displays the help screen.

13706PLAY Example

1. Configure **13706PLAY** using the utility **13706CFG**. For further information on 13706CFG, see the *13706CFG User Manual*, document number X31B-B-001-xx.
2. Type **13706PLAY** to start the program.
3. Type **?** for help.
4. Type **i** to initialize the registers.
5. Type **xa** to display the contents of the registers.
6. Type **x 34** to read register 34h.
7. Type **x 34 10** to write 10h to register 34h.
8. Type **f 0 ffff aa** to fill the first FFFFh 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.

Scripting

13706PLAY 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 13706PLAY command per line. All scripts must end with a “q” (quit) command.

On a PC platform, a typical script command line might be:

13706PLAY < dumpregs.scr > results

This causes the file **dumpregs.scr** to be interpreted as commands by 13706PLAY and the results to be sent to the file **results.**

Example 1: Create a script file that reads all registers and then exits.

```
; This file initializes the S1D13706 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.

; Initialize the S1D13706
i

; Read all registers
xa

; Exit the program
q
```

Comments

- All displayed numeric values are considered to be hexadecimal unless identified otherwise. For example:
  - 10 = 10h = 16 decimal.
  - 10t = 10 decimal.
  - 010‘b = 2 decimal.
- Redirecting commands from a script file (PC platform) allows those commands to be executed as if entered by a user.

THIS PAGE LEFT BLANK

EPSON®

S1D13706 Embedded Memory LCD Controller

13706BMP Demonstration Program

Document Number: X31B-B-004-02

THIS PAGE LEFT BLANK

13706BMP

13706BMP is a demonstration utility used to show the S1D13706 display capabilities by rendering bitmap images on the display device. The program displays any bitmap stored in Windows BMP file format and then exits. 13706BMP supports SviwelView™ (90°, 180°, and 270° hardware rotation of the display image).

13706BMP is designed to operate on a personal computer (PC) within a 32-bit environment only (Windows® 9x/NT). Other embedded platforms are not supported due to the possible lack of system memory or structured file system.

The 13706BMP demonstration utility must be configured and/or compiled to work with your hardware configuration. The program 13706CFG.EXE can be used to configure 13706BMP. For further information on 13706CFG, refer to the *13706CFG Users Manual*, document number X31B-B-001-xx.

S1D13706 Supported Evaluation Platforms

13706BMP supports the following S1D13706 evaluation platforms:

- PC with an Intel 80x86 processor running Windows 9x/NT.

Note
The 13706BMP source code may be modified by the OEM to support other evaluation platforms.

Installation

Copy the file **13706bmp.exe** to a directory in the path (e.g. PATH=C:\S1D13706).

Usage

At the prompt, type:

13706bmp bmpfile1 [bmpfile2] [ds=n | ds=?] [move=n] [/noinit] [/r90 | /r180 | /r270] [/v] [/?]

Where:

bmpfile1	Specifies filename of the windows format bmp image used for the main window (display surface 0).
bmpfile2	Specifies filename of the windows format bmp image used for the sub-window (display surface 1). If bmpfile2 is not specified, bmpfile1 is also used for the sub-window.
ds=n	Selects display surfaces (see Section , *"Display Surfaces"* on page 5).
ds=?	Shows available display surfaces (see Section , *"Display Surfaces"* on page 5).
move=n	Automatically moves the sub-window for n seconds. To move the sub-window indefinitely set n=-1.
/noinit	Skips full register initialization. Only registers used for changing the color depth (bpp) are updated. Additionally, some registers are read to determine information such as display size.
/r90	Enables SwivelView 90° mode, counter-clockwise hardware rotation of the LCD image by 90 degrees.
/r180	Enables SwivelView 180° mode, counter-clockwise hardware rotation of the LCD image by 180 degrees.
/r270	Enables SwivelView 270° mode, counter-clockwise hardware rotation of the LCD image by 270 degrees.
/v	Verbose mode (provides information about the displayed image).
/?	Displays the help message.

Note

13706BMP displays the bmpfile image(s) and returns to the prompt.

Display Surfaces

A display surface is a block of memory assigned to the main window and/or sub-window of the S1D13706. The sub-window is a feature of the S1D13706 “Picture-In-Picture Plus” feature. For further information on “Picture-In-Picture Plus”, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

13706BMP includes three predefined display surfaces (0-2) which cover the possible combinations of these windows. Table 1: “Display Surfaces” lists the display surfaces that may be selected.

Table 1: Display Surfaces

Display Surface (ds=)	Window(s) using Memory Block 0	Window(s) using Memory Block 1
0	Main Window	—
1	Main & Sub-window	—
2	Main Window	Sub-window

When ds=0, bmpfile1.bmp is displayed in the main window. If ds=n is not specified on the command line, this setting is automatically used when bmpfile2.bmp is not provided. This should be chosen when a sub-window is not required.

When ds=1, bmpfile1.bmp is displayed in the main window and also in the sub-window. Note that only a portion of bmpfile1.bmp is displayed if the sub-window is smaller than the resolution of the bmpfile.

When ds=2, bmpfile1.bmp is displayed in the main window and bmpfile2.bmp is displayed in the sub-window. This is the most useful combination to demonstrate the “Picture-In-Picture Plus” feature.

13706BMP Examples

To display a bmp image in the main window on an LCD, type the following:
13706bmp bmpfile1.bmp ds=0

To display a bmp image in the main window with 90° SwivelView™ enabled, type the following:
13706bmp bmpfile1.bmp ds=0 /r90

To display the same bmp image in both the main window and the sub-window, type the following:
13706bmp bmpfile1.bmp ds=1

To display different bmp images independently in the main and sub-windows and have the sub-window move indefinitely within the main window, type the following:
13706 bmpfile1.bmp bmpfile2.bmp ds=2 move=-1

Comments

- 13706BMP displays only Windows BMP format images.
- A 24-bit true color bitmap is displayed at a color depth of 16 bit-per-pixel.
- Only the green component of the image is seen on a monochrome panel.

S1D13706 Embedded Memory LCD Controller

Windows® CE Display Drivers

Document Number: X31B-E-001-03

THIS PAGE LEFT BLANK

WINDOWS® CE DISPLAY DRIVERS

Windows CE display drivers are available for the S1D13706 Embedded Memory LCD Controller. The drivers support color depths of 2, 4, 8 and 16 bit-per-pixel (bpp) for both landscape (no rotation) and SwivelView modes (90°, 180°, and 270° hardware rotation). Software cursor is supported only for 8 and 16 bpp color depths in landscape mode (no rotation).

This document and the updated source code for the Windows CE drivers are updated as appropriate. Please check the Epson Electronics America Website at www.eea.epson.com for the latest revisions before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

Program Requirements

Video Controller	:	S1D13706
Display Type	:	LCD
Windows Version	:	CE Version 2.11/2.12

Example Driver Builds

Build for CEPC (X86) on Windows CE Platform Builder 2.12

1. Install Microsoft Windows NT v4.0.
2. Install Platform Builder 2.12 by running SETUP.EXE from CD #1.
3. Follow the steps below to create a “Build Epson for x86” shortcut using 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 the “Exploring -- Start Menu” is displayed.

   c. Under the folder \Winnt\Profiles\All Users\Start Menu\Programs\Microsoft Windows CE Platform Builder\x86 Tools, find the icon “x86 MINSHELL”.

   d. Drag the icon “x86 MINSHELL” onto the desktop using the right mouse button.

   e. Select “Copy Here”.

   f. Rename the icon “x86 MINSHELL” to “Build Epson for x86” by right clicking on the icon and selecting “rename”.

   g. Right click on the icon “Build Epson for x86” and click on “Properties” to display the “Build Epson for x86 Properties” window.

   h. Click on “Shortcut” and replace “Minshell” under the entry “Target” with “Epson”.

   i. Click on “OK” to finish.

4. Create an EPSON project.

   a. Make an Epson folder under the folder \WINCE212\PUBLIC.

   b. Copy MAXALL and its sub-folders (\WINCE212\PUBLIC\MAXALL) to the Epson folder.

   xcopy /s /e \wince212\public\maxall*.* \wince212\public\epson

   c. Rename \WINCE212\PUBLIC\EPSON\MAXALL.BAT to EPSON.BAT.

   d. Edit the file EPSON.BAT to append the following lines to the end of the file:

   ```
   @echo on
   set CEPC_DDI_S1D13706=1
   @echo off
   ```

5. Create a new folder called S1D13706 under \WINCE212\PLATFORM\CEPC\DRIVERS\DISPLAY, and copy the S1D13706 driver source code into \WINCE212\PLATFORM\CEPC\DRIVERS\DISPLAY\S1D13706.
6. Add S1D13706 into the directory list in file \WINCE212\PLATFORM\CEPC\DRIVERS\DISPLAY\dirs

7. Edit the file \WINCE212\PLATFORM\CEPC\FILES\platform.bib to add the following after the line "IF ODO_NODISPLAY!":

```
IF CEPC_DDI_S1D13706
    ddi.dll    $(_FLATRELEASEDIR)\S1D13706.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_S1D13706 !
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 13706CFG utility program. The file MODE0.H (located in X:\wince212\platform\cepc\drivers\display\S1D13706) contains the register values required to set the desired screen resolution, color depth (bpp), panel type, rotation, etc.

9. Edit the file PLATFORM.REG to match the screen resolution, color depth (bpp), and rotation information in MODE.H. PLATFORM.REG is located in X:\wince212\platform\cepc\files. For example, the display driver section of PLATFORM.REG should be as follows when using a 320x240 LCD panel with a color depth of 8 bpp in SwivelView 0° (landscape) mode.

```
; Default for EPSON Display Driver
; 320x240 at 8bits/pixel, LCD display, no rotation
; Useful Hex Values
; 640=0x280 480=0x1E0 320=0x140 240=0xF0
```

[HKEY_LOCAL_MACHINE\Drivers\Display\S1D13706]

“Width”=dword:140

“Height”=dword:F0

“Bpp”=dword:8

“Rotation”=dword:0

10. Remove the \wince212\release directory and delete \wince212\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).

Installation for CEPC Environment

Windows CE v2.1x 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 the CONFIG.SYS file on the floppy disk to contain the following line only.

      device=a:\himem.sys

   c. Edit the AUTOEXEC.BAT file on the floppy disk to contain the following lines.

      mode com1:9600,n,8,1
      loadcepc /B:9600 /C:1 c:\wince\release\nk.bin

   d. Copy LOADCEPC.EXE from c:\wince\public\common\oak\bin\I386 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 the CONFIG.SYS file on the hard drive to contain the following line only.

      device=c:\himem.sys

   c. Edit the AUTOEXEC.BAT file on the hard drive to contain the following lines.

      mode com1:9600,n,8,1

      loadcepc /B:9600 /C:1 c:\wince\release\nk.bin

   d. Confirm that NK.BIN is located in c:\wince\release.

   e. Reboot the system from the hard drive.

Comments

- The display driver is CPU independent, allowing use of the driver for other Windows CE Platform Builder v2.12 supported platforms. The file GPEFLAT.C may require changes to return the correct value for PhysicalPortAddr, PhysicalVmemAddr, etc.
- The sample code is based on the following configuration.
  - Epson 320x240 8-bit color STN LCD panel.
  - SwivelView 0° mode (landscape).
  - 8 bpp color depth.
  - S5U13706B00C evaluation board.
- Other desired display modes may be supported by changing the mode table in the header file mode0.h. Mode0.h is generated using the utility 13706CFG.EXE. For further information on 13706CFG, see the *13706CFG Configuration Utility User Manual*, document number X31B-B-001-xx.
- Compile options such as clock chip support, grayscale palette, etc, are defined in the SOURCES file. The SOURCES file is included in the file 13706_ce.zip. Refer to the comments in the SOURCES file to set the proper definition.
- At the time of printing, the display drivers have been tested using x86 CPUs with Platform Builder v2.12 and v3.0 Beta. The drivers are updated as appropriate. Before beginning any development, check the Epson Electronics America Website at www.eea.epson.com for the latest revision of this document.
- For the latest information and release notes on the S1D13706 Windows CE display driver, see the README.TXT included in the file 13706_CE.ZIP. For the latest release of this file, check the Epson Electronics America Website at www.eea.epson.com.

S1D13706 Embedded Memory LCD Controller

Wind River WindML v2.0 Display Drivers

Document Number: X31B-E-002-03

THIS PAGE LEFT BLANK

Wind River WindML v2.0 DISPLAY DRIVERS

The Wind River WindML v2.0 display drivers for the S1D13706 Embedded Memory LCD Controller are intended as "reference" source code for OEMs developing for Wind River's WindML v2.0. The driver package provides support for both 8 and 16 bit-per-pixel color depths. The source code is written for portability and contains functionality for most features of the S1D13706. Source code modification is required to provide a smaller, more efficient driver for mass production.

The WindML display drivers are designed around a common configuration include file called **mode0.h** which is generated by the configuration utility 13706CFG. This design allows for easy customization of display type, clocks, decode addresses, rotation, etc. by OEMs. For further information on 13706CFG, see the *13706CFG Configuration Program User Manual*, document number X31B-B-001-xx.

Note

The WindML display drivers are provided as "reference" source code only. They are intended to provide a basis for OEMs to develop their own drivers for WindML v2.0. These drivers are not backwards compatible with UGL v1.2. For information on the UGL v1.2 display drivers, see *Wind River UGL v1.2 Display Drivers*, document number X31B-E-003-xx.

This document and the source code for the WindML display drivers is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com or the Epson Research and Development website at http://www.erd.epson.com for the latest revisions before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

Building a WindML v2.0 Display Driver

The following instructions produce a bootable disk that automatically starts the UGL demo program. These instructions assume that Wind River's Tornado platform is already installed.

Note
For the example steps where the drive letter is given as "x:". Substitute "x" with the drive letter that your development environment is on.

1. Create a working directory and unzip the WindML display driver into it.

   From a command prompt or GUI interface create a new directory (e.g. x:\13706).

   Unzip the file **13706windml.zip** to the newly created working directory. The files will be unzipped to the directories "x:\13706\8bpp" and "x:\13706\16bpp".

2. Configure for the target execution model.

   This example build creates a VxWorks image that fits onto and boots from a single floppy diskette. In order for the VxWorks image to fit on the disk certain modifications are required.

   Replace the file "x:\Tornado\target\config\pcPentium\config.h" with the file "x:\13706\8bpp\File\config.h" (or "x:\13706\16bpp\File\config.h"). The new **config.h** file removes networking components and configures the build image for booting from a floppy disk.

Note
Rather than simply replacing the original **config.h** file, rename it so the file can be kept for reference purposes.

3. Build a boot ROM image.

   From the Tornado tool bar, select Build -> Build Boot ROM. Select "pcPentium" as the BSP and "bootrom_uncmp" as the image.

4. Create a bootable disk (in drive A:).

   From a command prompt change to the directory "x:\Tornado\host\x86-win32\bin" and run the batch file **torvars.bat**. Next, change to the directory "x:\Tornado\target\config\pcPentium" and type:

   mkboot a: bootrom_uncmp

5. If necessary, generate a new **mode0.h** configuration file.

   The file **mode0.h** contains the register values required to set the screen resolution, color depth (bpp), display type, rotation, etc. The **mode0.h** file included with the drivers, may not contain applicable values and must be regenerated. The configuration program 13706CFG can be used to build a new **mode0.h** file. If building for 8 bpp, place the new **mode0.h** file in the directory "x:\13706\8bpp\File". If building for 16 bpp, place the new **mode0.h** file in "x:\13706\16bpp\File".

Note

Mode0.h should be created using the configuration utility 13706CFG. For more information on 13706CFG, see the *13706CFG Configuration Program User Manual*, document number X31B-B-001-xx available at www.erd.epson.com.

6. Build the WindML v2.0 library.

   From a command prompt change to the directory "x:\Tornado\host\x86-win32\bin" and run the batch file **torvars.bat**. Next, change to the directory "x:\Tornado\target\src\ugl" and type the command:

   make CPU=PENTIUM ugl

7. Open the S1D13706 workspace.

   From the Tornado tool bar, select File->Open Workspace...->Existing->Browse... and select the file "x:\13706\8bpp\13706.wsp" (or "x:\13706\16bpp\13706.wsp").

8. Add support for single line comments.

   The WindML v2.0 display driver source code uses single line comment notation, "//", rather than the ANSI conventional comments, "/*...*/".

   To add support for single line comments follow these steps:

   a. In the Tornado "Workspace Views" window, click on the "Builds" tab.

   b. Expand the "8bpp Builds" (or "16bpp Builds") view by clicking on the "+" next to it. The expanded view will contain the item "default". Right-click on "default" and select "Properties...". A "Properties:" window will appear.

   c. Select the "C/C++ compiler" tab to display the command switches used in the build. Remove the "-ansi" switch from the line that contains "-g -mpentium -ansi -nostdinc -DRW_MULTI_THREAD".
   (Refer to GNU ToolKit user's guide for details)

9. Compile the VxWorks image.

   Select the "Builds" tab in the Tornado "Workspace Views" window.

   Right-click on "8bpp files" (or "16bpp files") and select "Dependencies...". Click on "OK" to regenerate project file dependencies for "All Project files".

   Right-click on "8bpp files" (or "16bpp files") and select "ReBuild All(vxWorks)" to build VxWorks.

10. Copy the VxWorks file to the diskette.

    From a command prompt or through the Windows interface, copy the file "x:\13706\8bpp\default\vxWorks" (or "x:\13706\16bpp\default\vxWorks") to the bootable disk created in step 4.

11. Start the VxWorks demo.

    Boot the target PC with the VxWorks bootable diskette to run the UGL demo program automatically.

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Wind River UGL v1.2 Display Drivers

Document Number: X31B-E-003-02

THIS PAGE LEFT BLANK

Wind River UGL v1.2 Display Drivers

The Wind River UGL v1.2 display drivers for the S1D13706 Embedded Memory LCD Controller are intended as “reference” source code for OEMs developing for Wind River’s UGL v1.2. The drivers provide support for both 8 and 16 bit-per-pixel color depths. The source code is written for portability and contains functionality for most features of the S1D13706. Source code modification is required to provide a smaller, more efficient driver for mass production.

The UGL display drivers are designed around a common configuration include file called **mode0.h** which is generated by the configuration utility 13706CFG. This design allows for easy customization of display type, clocks, addresses, rotation, etc. by OEMs. For further information on 13706CFG, see the *13706CFG Configuration Program User Manual*, document number X31B-B-001-xx.

This document and the source code for the UGL display drivers are updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com or the Epson Research and Development website at http://www.erd.epson.com for the latest revisions before beginning any development.

We appreciate your comments on our documentation. Please contact us via e-mail at documentation@erd.epson.com.

Building a UGL v1.2 Display Driver

The following instructions produce a bootable disk that automatically starts the UGL demo software. These instructions assume that the Wind River Tornado platform is correctly installed.

Note
For the example steps where the drive letter is given as "x:". Substitute "x" with the drive letter that your development environment is on.

1. Create a working directory and unzip the UGL display driver into it.

   Using a command prompt or GUI interface create a new directory (e.g. x:\13706).

   Unzip the file **13706ugl.zip** to the newly created working directory. The files will be unzipped to the directories "x:\13706\8bpp" and "x:\13706\16bpp".

2. Configure for the target execution model.

   This example build creates a VxWorks image that fits onto and boots from a single floppy diskette. In order for the VxWorks image to fit on the disk certain modifications are required.

   Replace the file "x:\Tornado\target\config\pcPentium\config.h" with the file "x:\13706\8bpp\File\config.h" (or "x:\13706\16bpp\File\config.h"). The new **config.h** file removes networking components and configures the build image for booting from a floppy disk.

Note
Rather than simply replacing the original **config.h** file, rename it so the file can be kept for reference purposes.

3. Build a boot ROM image.

   From the Tornado tool bar, select Build -> Build Boot ROM. Select "pcPentium" as the BSP and "bootrom_uncmp" as the image.

4. Create a bootable disk (in drive A:).

   From a command prompt in the directory "x:\Tornado\target\config\pcPentium" type
   mkboot a: bootrom_uncmp

5. If necessary, generate a new **mode0.h** configuration file.

   The file **mode0.h** contains the register values required to set the screen resolution, color depth (bpp), display type, rotation, etc. The **mode0.h** file included with the drivers, may not contain applicable values and must be regenerated. The configuration program 13706CFG can be used to build a new **mode0.h** file. If building for 8 bpp, place the new **mode0.h** file in the directory "x:\13706\8bpp\File". If building for 16 bpp, place the new **mode0.h** file in "x:\13706\16bpp\File".

Note

Mode0.h should be created using the configuration utility 13706CFG. For more information on 13706CFG, see the *13706CFG Configuration Program User Manual*, document number X31B-B-001-xx available at www.erd.epson.com.

6. Open the S1D13706 workspace.

   From the Tornado tool bar, select File->Open Workspace...->Existing->Browse... and select the file “x:\13706\8bpp\13706.wsp” (or “x:\13706\16bpp\13706.wsp”).

7. Add support for single line comments.

   The UGL v1.2 display driver source code uses single line comment notation, “//”, rather than the ANSI conventional comments, “/* . . . */”.

   To add support for single line comments follow these steps:

   a. In the Tornado “Workspace” window, click on the “Builds” tab.

   b. Expand the “8bpp Builds” (or “16bpp Builds”) view by clicking on the “+” next to it. The expanded view will contain the item “default”. Right-click on “default” and select “Properties...”. A properties window will appear.

   c. Select the “C/C++ compiler” tab to display the command switches used in the build. Remove the “-ansi” switch from the line that contains “-g -mpentium -ansi -nostdinc -DRW_MULTI_THREAD”. (Refer to GNU ToolKit user's guide for details)

8. Compile the VxWorks image.

   Select the “Files” tab in the Tornado “Workspace” window.

   Right-click on “8bpp files” (or “16bpp files”) and select “Dependencies...”. Click on “OK” to regenerate project file dependencies for “All Project files”.

   Right-click on “8bpp files” (or “16bpp files”) and select “ReBuild All(vxWorks)” to build VxWorks.

9. Copy the VxWorks file to the diskette.

   From a command prompt or through the Windows interface, copy the file “x:\13706\8bpp\default\vxWorks” (or “x:\13706\16bpp\default\vxWorks”) to the bootable disk created in step 4.

10. Start the VxWorks demo.

    Boot the target PC with the VxWorks bootable diskette to run the UGL demo program automatically.

THIS PAGE LEFT BLANK

S1D13706 Embedded Memory LCD Controller

Linux Console Driver

Document Number: X31B-E-004-01

THIS PAGE LEFT BLANK

Linux Console Driver

The Linux console driver for the S1D13706 Embedded Memory LCD Controller is intended as “reference” source code for OEMs developing for Linux, and supports 4, 8, and 16 bit-per-pixel color depths.

A Graphical User Interface (GUI) such as Gnome can obtain the frame buffer address from this driver allowing the Linux GUI the ability to update the display.

The console driver is designed around an include file called **s1d13706.h** which is generated by the configuration utility 13706CFG. Using 13706CFG the programmer can easily customize the operation of display type, clocks, addresses, rotation, etc.

This document and the source code for the Linux console drivers are updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com or the Epson Research and Development website at http://www.erd.epson.com for the latest revisions or before beginning any development.

We appreciate your comments on our documentation. Please contact us via e-mail at documentation@erd.epson.com.

Building the S1D13706 Console Driver

Follow the steps below to construct a copy of the Linux operating system using the S1D13706 as the console display device. These instructions assume that the GNU development environment is installed and the user is familiar with GNU and the Linux operating system.

1. Acquire the Linux kernel source code.

   You can obtain the Linux kernel source code from your Linux supplier or download the source from: ftp://ftp.kernel.org.

   The S1D13706 reference driver requires Linux kernel 2.2.0 or greater.

   For information on building the kernel refer to the readme file at:
   ftp://ftp.linuxberg.com/pub/linux/kernel/README

Note
The 13706 reference driver was built using Red Hat Linux 6.1, kernel version 2.2.17.

Before continuing with modifications for the S1D13706, you should ensure that you can build and start the Linux operating system.

2. Unzip the console driver files.

   Using a zip file utility, unzip the S1D13706 archive to a temporary directory. (e.g. /tmp)

   When completed the files:
   s1d13xxxfb.c
   s1d13706.h
   Config.in
   fbmem.c
   fbcon-cfb4.c, and
   Makefile
   should be located in the temporary directory.

3. Copy the console driver files to the build directory.

   Copy the files
   /tmp/s1d13xxxfb.c and
   /tmp/s1d13706.h
   to the directory /usr/src/linux/drivers/video.

   Copy the remaining source files
   /tmp/Config.in
   /tmp/fbmem.c
   /tmp/fbcon-cfb4.c, and
   /tmp/Makefile
   into the directory /usr/src/linux/drivers/video replacing the files of the same name.

If your kernel version is not 2.2.17 or you want to retain greater control of the build process then use a text editor and cut and paste the sections dealing with the Epson driver in the corresponding files of the same names.

4. Modify s1d13706.h

   The file s1d13706.h contains the register values required to set the screen resolution, color depth (bpp), display type, active display (LCD), display rotation, etc.

   Before building the console driver, refer to the descriptions in the file s1d13706.h for the default settings of the console driver. If the default does not match the configuration you are building for then s1d13706.h will have to be regenerated with the correct information.

   Use the program 13706CFG to generate the required header file. For information on how to use 13706CFG, refer to the *13706CFG Configuration Program User Manual*, document number X31B-B-001-xx, available at www.erd.epson.com

   After selecting the desired configuration, choose "File->Export" and select the "C Header File for S1D13706 Generic Drivers" option. Save the new configuration as s1d13706.h in the /usr/src/linux/drivers/video, replacing the original configuration file.

5. Configure the video options.

   From the command prompt in the directory /usr/src/linux run the command:
   make menuconfig

   This command will start a text based interface which allows the selection of build time parameters. From the text interface under "Console drivers" options, select:

   "Support for frame buffer devices"
   "Epson LCD controllers support"
   "S1D13706 support"
   "Advanced low level driver options"
   "xBpp packed pixels support" *

   * where x is the color depth being compiled for.

   If you are using the Epson PCI evaluation board then you must also select:

   "Epson PCI Bridge adapter support"

   Once you have configured the kernel options, save and exit the configuration utility.

6. Compile and install the kernel

   Build the kernel with the following sequence of commands:
   make dep
   make clean
   make bzImage
   /sbin/lilo (if running lilo)

7. Boot to the Linux operating system

   If you are using lilo (Linux Loader), modify the lilo configuration file as discussed in the kernel build README file. If there were no errors during the build, from the command prompt run:

   ```
   lilo
   ```

   and reboot your system.

Note

In order to use the S1D13706 console driver with X server, you need to configure the X server to use the FBDEV device. A good place to look for the necessary files and instructions on this process is on the Internet at www.xfree86.org

EPSON®

S1D13706 Embedded Memory LCD Controller

QNX Photon v2.0 Display Driver

Document Number: X31B-E-005-01

THIS PAGE LEFT BLANK

QNX Photon v2.0 Display Driver

The Photon v2.0 display drivers for the S1D13706 Embedded Memory LCD controller are intended as “reference” source code for OEMs developing for QNX platforms. The driver package provides support for 8 and 16 bit-per-pixel color depths. The source code is written for portability and contains functionality for most features of the S1D13706. Source code modification is required to provide a smaller driver for mass production.

The current revision of the driver is designed for use with either QNX RTP or QNX4 from the latest product CD (Dec. 99).

The Photon v2.0 display driver is designed around a common configuration include file called **S1D13706.h,** which is generated by the configuration utility 13706CFG. This design allows for easy customization of display type, clocks, decode addresses, rotation, etc. by OEMs. For further information on 13706CFG, see the *13706CFG Configuration Program User Manual*, document number X31B-B-001-xx.

Note

The QNX display drivers are provided as “reference” source code only. They are intended to provide a basis for OEMs to develop their own drivers for QNX Photon v2.0.

This document and the source code for the QNX display drivers are updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com or the Epson Research and Development website at http://www.erd.epson.com for the latest revisions before beginning any development.

We appreciate your comments on our documentation. Please contact us via e-mail at documentation@erd.epson.com.

Building the Photon v2.0 Display Driver

The following steps build the Photon v2.0 display driver and integrate it into the QNX operating system. These instructions assume the QNX developer environment is correctly installed and the developer is familiar with building for the QNX operating system.

Unpack the Graphics Driver Development Kit Archive

1. Install the QNX ddk package using the Package Manager utility.

   For information about the Drivers Development Kit contact QNX directly.

2. Once the ddk package is installed, copy the directory tree /usr/scr/gddk_v1.0 into the Project directory.

3. Change directory to Project/gddk_1.0/devg.

4. Unpack the display driver files using the commands:

   #gunzip S1D13706.tar.gz

   #tar –xvf S1D13706.tar

   This unpacks the files into the directory Project/gddk_1.0/devg/S1D13706.

Configure the Driver

The files **s1d13706_16.h** and **s1d13706_8.h** contain register values required to set the screen resolution, color depth (bpp), display type, rotation, etc. The **s1d13706.h** file included with the drivers may not contain applicable values and must be regenerated. The configuration program 13706CFG can be used to build new **s1d13706_16.h** and **s1d13706_8.h** files.

Note

S1d13706.h should be created using the configuration utility 13706CFG. For more information on 13706CFG, see the *13706CFG Configuration Program User Manual*, document number X31B-B-001-xx available at www.erd.epson.com.

Build the Driver

The first time the driver is built, the following command ensures that all drivers and required libraries are built. At the root of the Project source tree, type **make**.

Note

To build drivers for X86 NTO type ‘OSLIST=nto CPULIST=x86 make’.

Further builds do not require all libraries to be re-built. To build only the S1D13706 display driver, change to the directory gddk_1.0/devg/S1D13706 and type **make**.

Installing the Driver

The build step produces two library images:

- lib/disputil/nto/x86/libdisputil.so
- lib/disputil/nto/x86/libffb.so

For the loader to locate them, the files need to be renamed and copied to the lib directory.

1. Rename libdisputil.so to libdisputil.so.1 and libffb.so to libffb.so.1.
2. Copy the files new files libdisputil.so.1 and libffb.so.1 to the directory /usr/lib.
3. Copy the file devg-S1D13706.so to the /lib/dll directory.

Note
To locate the file devg-S1D13706.so, watch the output of the 'true' command during the makefile build.

4. Modify the trap file crt.$NODE in the /etc/config/trap directory by inserting the following lines at the top of the file.

   io-graphics -g640x480x16 -dldevg-S1D13706.so -I0 -d0x0,0x0;#640,480,16 Epson

   io-graphics -g640x480x8 -dldevg-S1D13706.so -I0 -d0x0,0x0;#640,480,8 Epson

Run the Driver

Note
For the remaining steps the S5U13706B00C evaluation board must be installed on the test platform.

It is recommended that the driver be verified **before starting QNX with the S1D13706 as the primary display**. To verify the driver, type the following command at the root of the Project source tree (gddk_1.0 directory).

util/bench/nto/x86/o/devg-bench -dldevg/S1D13706/nto/x86/dll/devg-S1D13706.so -m**W**,**H**,**C**,**F** -d0x0,0x0

Where:

W is the configured width of the display
H is the configured height of the display
C is the color depth in bpp (either 8 or 16)
F is the configured frame rate

This command starts the bench utility which will initialize the driver as the secondary display and exercise the drivers main functions. If the display appears satisfactory, restart QNX Photon and the restart will result in the S1D13706 display driver becoming the primary display device.

Comments

- To restore the display driver to the default, comment out changes made to the trap file crt.$NODE.

S1D13XXX 32-Bit Windows Device Driver Installation Guide

Document No. X00A-E-003-04

THIS PAGE LEFT BLANK

S1D13XXX 32-Bit Windows Device Driver Installation Guide

This manual describes the installation of the Windows 9x/ME/NT 4.0/2000 device drivers for the S5U13xxxB00x series of Epson Evaluation Boards.

The file S1D13XXX.VXD is required for using the Epson supplied Intel32 evaluation and test programs for the S1D13xxx family of LCD controllers with Windows 9x/ME.

The file S1D13XXX.SYS is required for using the Epson supplied Intel32 evaluation and test programs for the S1D13xxx family of LCD controllers with Windows NT 4.0/2000.

The file S1D13XXX.INF is the install script.

For updated drivers, ask your Sales Representative or visit Epson Electronics America on the World Wide Web at www.eea.epson.com.

Driver Requirements

Video Controller	:	S1D13xxx
Display Type	:	N/A
BIOS	:	N/A
DOS Program	:	No
Dos Version	:	N/A
Windows Program	:	Yes, Windows 9x/ME/NT 4.0/2000 device driver
Windows DOS Box	:	N/A
Windows Full Screen	:	N/A
OS/2	:	N/A

Installation

Windows NT Version 4.0

All evaluation boards require the driver to be installed as follows.

1. Install the evaluation board in the computer and boot the computer.
2. Copy the files S1D13XXX.INF and S1D13XXX.SYS to a directory on a local hard drive.
3. Right click your mouse on the file S1D13XXX.INF and select INSTALL from the menu.
4. Windows will install the device driver and ask you to restart.

Windows 2000

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 FOUND NEW HARDWARE dialog box.
3. Click NEXT.
4. The New Hardware Wizard will bring up the dialog box to search for a suitable driver.
5. Click NEXT.
6. 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.
7. Click NEXT.
8. Windows 2000 will open the installation file and show the option EPSON PCI Bridge Card. Select this file and click OPEN.
9. Windows then shows the path to the file. Click OK.
10. Click NEXT.
11. 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/REMOVE HARDWARE, click NEXT.
3. Select ADD/TROUBLESHOOT A DEVICE, and click NEXT. Windows 2000 will attempt to detect any new plug and play device and fail.
4. The CHOOSE HARDWARE DEVICE dialog box appears. Select ADD NEW HARDWARE and click NEXT.
5. Select NO I WANT TO SELECT FROM A LIST and click NEXT.
6. Select OTHER DEVICE from the list and click NEXT.
7. Click HAVE DISK.
8. Specify the location of the driver files, select the S1D13XXX INF file and click OPEN.
9. Click OK.

Windows 98/ME

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 will open the installation file and show the option EPSON 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 double-click on ADD NEW HARDWARE to launch the ADD NEW HARDWARE WIZARD. Click NEXT.
3. Windows will attempt to detect any new plug and play device and fail. Click NEXT.
4. Windows will ask you to let it detect the hardware, or allow you to select from a list. Select NO, I WANT TO SELECT THE HARDWARE FROM A LIST and click NEXT.
5. From the list select OTHER DEVICES and click NEXT.
6. Click HAVE DISK and type the path to the driver files, or select browse to find the driver.
7. Click OK.
8. The driver will be identified as EPSON PCI Bridge Card. Click NEXT.
9. Click FINISH.

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 PCI Bridge 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

3. Click NEXT, Windows will search the floppy drive and fail.
4. Windows will attempt to load the new hardware as a Standard VGA Card.
5. Click CANCEL. The Driver must be loaded from the CONTROL PANEL under ADD/NEW HARDWARE.
6. Select NO for Windows to DETECT NEW HARDWARE.
7. Click NEXT.
8. Select OTHER DEVICES from HARDWARE TYPE and Click NEXT.
9. Click HAVE DISK.
10. Specify the location of the driver and click OK.
11. Click OK.
12. EPSON PCI Bridge Card will appear in the list.
13. Click NEXT.
14. Windows will install the driver.
15. Click FINISH.
16. Windows will ask you to restart the system.
17. 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.

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 S1D13XXX.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 S1D13XXX.INF and click OK.
9. Click OK.
10. The EPSON PCI Bridge Card should be selected in the list window.
11. Click NEXT.
12. Click NEXT.
13. Click Finish.

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

S5U13706B00C Rev. 1.0 Evaluation Board User Manual

Document Number: X31B-G-004-04

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This manual describes the setup and operation of the S5U13706B00C Rev. 1.0 Evaluation Board. The board is designed as an evaluation platform for the S1D13706 Embedded Memory LCD Controller.

This user manual is updated as appropriate. Please check the Epson Electronics America Website at www.eea.epson.com or the Epson Research and Development Website at www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Features

Following are some features of the S5U13706B00C Rev. 1.0 Evaluation Board:

- 100-pin TQFP S1D13706F00A Embedded Memory LCD Controller with 80K bytes of embedded SRAM.
- Headers for connecting to various Host Bus Interfaces.
- Configuration options.
- Manual or software adjustable positive LCD bias power supply from +20V to +40V.
- Manual or software adjustable negative LCD bias power supply from -24V to -8V.
- Software adjustable backlight intensity support.
- 4/8-bit 3.3V or 5V single monochrome passive LCD panel support.
- 4/8/16-bit 3.3V or 5V single color passive LCD panel support.
- 9/12/18-bit 3.3V or 5V active matrix TFT LCD panel support.
- Direct interface for 18-bit Epson D-TFD LCD panel support.
- Direct interface for 18-bit Sharp HR-TFT LCD panel support.
- Programmable clock synthesizer to CLKI and CLKI2 for maximum clock flexibility.
- Software initiated power save mode.
- Hardware or software Video Invert support.
- Selectable clock source for CLKI and CLKI2.
- External oscillator for CLKI and CLKI2.

3 Installation and Configuration

The S5U13706B00C is designed to support as many platforms as possible. The S5U13706B00C incorporates a DIP switch and seven jumpers which allow both evaluation board and S1D13706 LCD controller to be configured for a specified evaluation platform.

3.1 Configuration DIP Switches

The S1D13706 has configuration inputs (CNF[7:0]) which are read on the rising edge of RESET#. In order to configure the S1D13706 for multiple Host Bus Interfaces a ten-position DIP switch (S1) is required. The following figure shows the location of DIP switch SW1 on the S5U13706B00C.

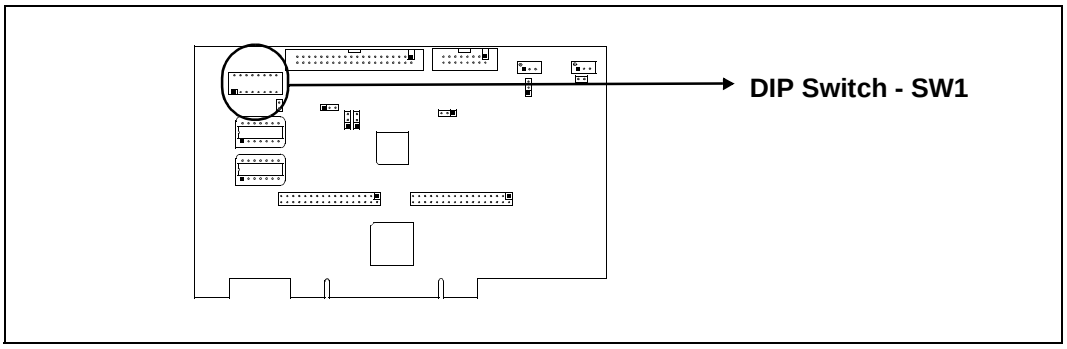


Figure 3-1: Configuration DIP Switch (SW1) Location

The S1D13706 has 8 configuration inputs (CONF[7:0]) which are read on the rising edge of RESET#. All S1D13706 configuration inputs are fully configurable using a ten position DIP switch as described below.

Table 3-1: Configuration DIP Switch Settings

Switch	S1D13706 Signal	Value on this pin at rising edge of RESET# is used to configure: Closed (On/1)	Open (Off/0)
SW1-[3:1]	CNF[2:0]	Select host bus interface as follows: **CNF2 CNF1 CNF0 — Host Bus Interface** 0 0 0 — SH-4/SH-3 0 0 1 — MC68K #1 0 1 0 — MC68K #2 0 1 1 — Generic #1 1 0 0 — Generic #2 1 0 1 — RedCap 2 1 1 0 — DragonBall 1 1 1 — Reserved Note: The host bus interface is 16-bit.	
SW1-4	CNF3	Enable GPIO pins	Enable additional pins for D-TFD/HR-TFT
SW1-5	CNF4	Big Endian bus interface	Little Endian bus interface
SW1-6	CNF5	WAIT# is active high	WAIT# is active low
SW1-[8:7]	CNF[7:6]	CLKI to BClk divide select: **CNF7 CNF6 — CLKI to BClk Divide Ratio** 0 0 — 1 : 1 0 1 — 2 : 1 1 0 — 3 : 1 1 1 — 4 : 1	
SW1-9[1]	-	Hardware Video Invert - invert video data[1]	Hardware Video Invert - normal video data[1]
SW1-10	-	Disable FPGA for non-PCI host	Enable FPGA for PCI host

= Required settings when used with PCI Bridge FPGA

Note

[1] To enable the Hardware Video Invert function the following are required:

- GPIO pins must be enabled (S1-4 closed).
- GPIO0 must be connected to S1-9 (Jumper JP1 set to 1-2).
- GPIO Pin Input Enable (REG[A9h] bit 7) must be set to 1.
- GPIO0 Pin IO Configuration (REG[A8h] bit 0) must be set to 0.
- Hardware Video Invert Enable bit (REG[70h] bit 5) must be set to 1.

3.2 Configuration Jumpers

The S5U13706B00C has seven jumper blocks which configure various setting on the board. The jumper positions for each function are shown below.

Table 3-2: Jumper Summary

Jumper	Function	Position 1-2	Position 2-3	No Jumper
JP1	GPIO0 Connection	GPIO0 connected to SW1-9 for hardware video invert	—	GPIO0 disconnected from SW1-9 for direct HR/TFT/D-TFD or GPIO testing
JP2	CLKI2 Source	MCLKOUT from clock synthesizer	External oscillator (U5)	—
JP3	CLKI Source	VCLKOUT from clock synthesizer	External oscillator (U6)	—
JP4	GP0 Polarity on H1	Normal (Active High)	Inverted (Active Low)	—
JP5	Contrast adjust for +ve LCD bias (VDDH)	Software controlled	Manual controlled	—
JP6	LCD Panel Voltage	+5V LCDVCC	+3.3V LCDVCC	—
JP7	Contrast adjust for -ve LCD bias (VLCD)	Software controlled	—	Manual controlled

= recommended settings

JP1 - GPIO0 Connection

JP1 selects whether GPIO0 is connected to SW1-9. SW1-9 is used to enable hardware video invert on the S1D13706.
When the jumper is on (position 1-2), SW1-9 controls the hardware video invert feature (default setting).
When the jumper is off, the hardware video invert feature is disabled. This setting must be used for HR/TFT and D-TFD panels as GPIO0 is required in each panels LCD interface pin mapping. Refer to the *S1D13706 Hardware Functional Specification*, document number X28B-A-001-xx for details.

Note
When configured for Sharp HR-TFT or Epson D-TFD panels, JP1 must be set to no jumper and JP6 must be set to position 2-3.

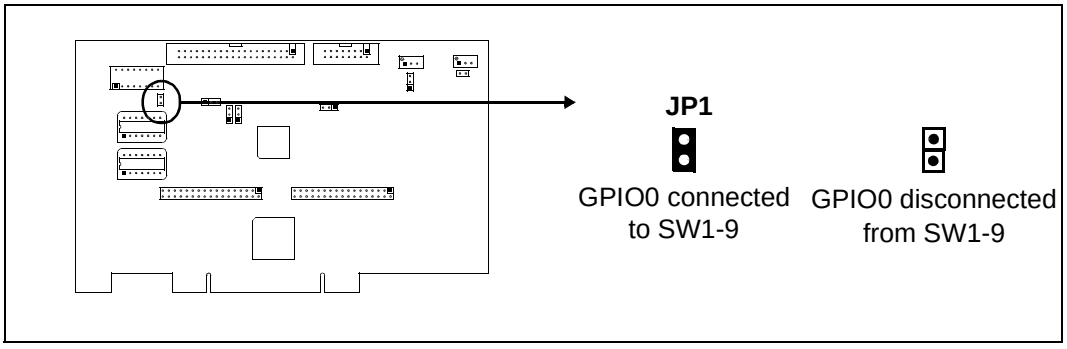


Figure 3-2: Configuration Jumper (JP1) Location

JP2 - CLKI2 Source

JP2 selects the source for the CLKI2.
Position 1-2 sets the CLKI2 source to MCLKOUT from the Cypress clock synthesizer (default setting).
Position 2-3 sets the CLKI2 source to the external oscillator at U5.

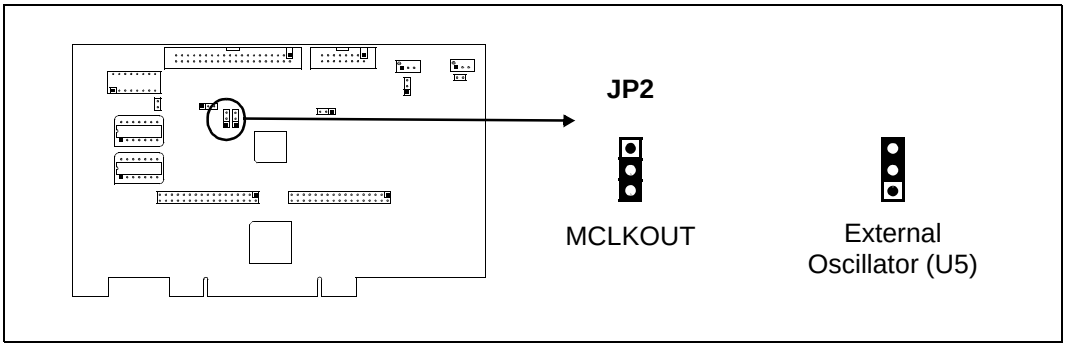


Figure 3-3: Configuration Jumper (JP2) Location

JP3 - CLKI Source

JP2 selects the source for the CLKI.
Position 1-2 sets the CLKI2 source to VCLKOUT from the Cypress clock synthesizer (default setting).
Position 2-3 sets the CLKI2 source to the external oscillator at U6.

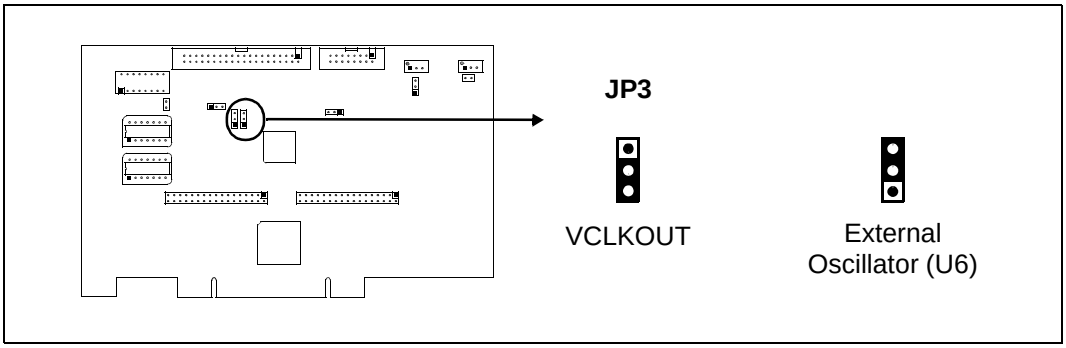


Figure 3-4: Configuration Jumper (JP3) Location

JP4 - GPO Polarity on H1

JP4 selects the polarity of the GPO signal available on the LCD Connector H1.
Position 1-2 sends the GPO signal directly to H1 (default setting).
Position 2-3 inverts the GPO signal before sending it to H1.

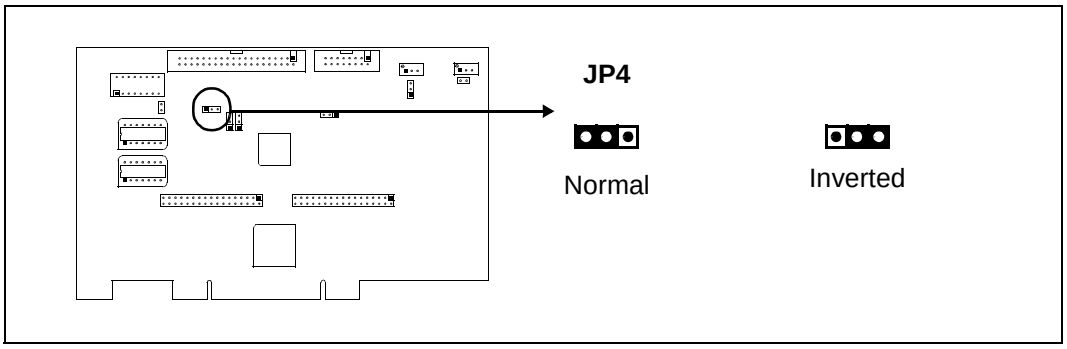


Figure 3-5: Configuration Jumper (JP4) Location

JP5 - Contrast adjust for +ve LCD bias (VDDH)

JP5 selects the type of control used for contrast adjustment of the +ve LCD bias (VDDH).
Position 1-2 selects software control of the contrast adjustment.
Position 2-3 selects manual control of the contrast adjustment using potentiometer R24 (default setting).

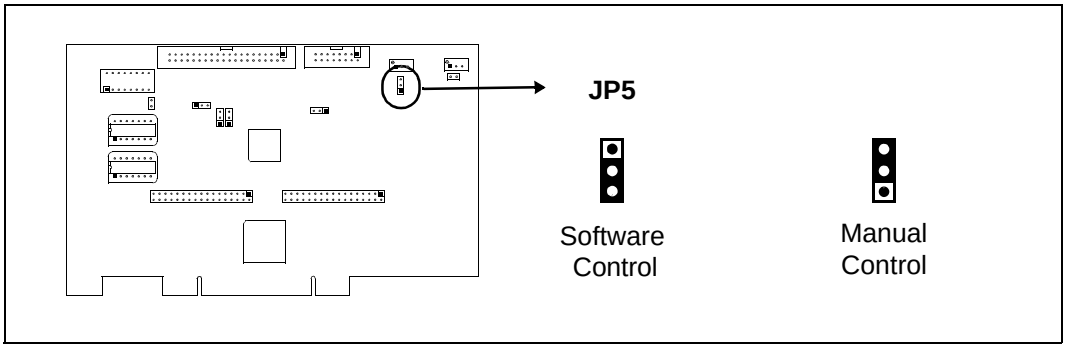


Figure 3-6: Configuration Jumper (JP5) Location

JP6 - LCD Panel Voltage

JP6 selects voltage level to the LCD panel.
Position 1-2 sets the voltage level to 5.0V (default setting).
Position 2-3 sets the voltage level to 3.3V.

Note
When configured for Sharp HR-TFT or Epson D-TFD panels, JP1 must be set to no jumper and JP6 must be set to position 2-3.

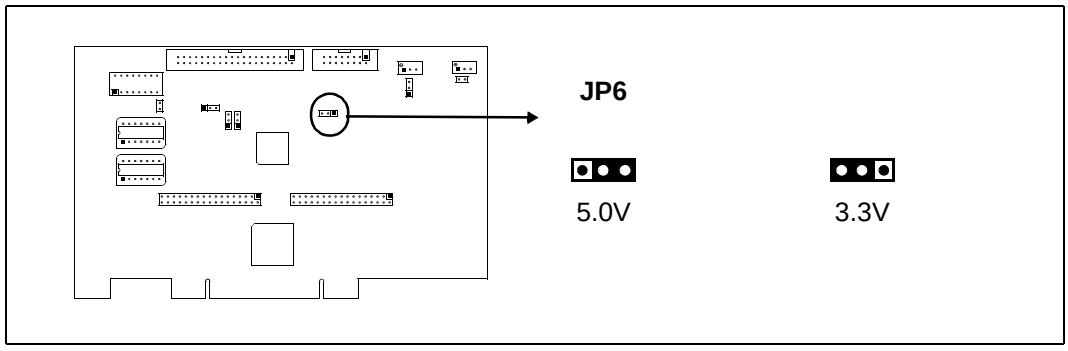


Figure 3-7: Configuration Jumper (JP6) Location

JP7 - Contrast adjust for -ve LCD bias (VLCD)

JP7 selects the type of control used for contrast adjustment of the -ve LCD bias (VLCD).
Position 1-2 selects software control of the contrast adjustment.
Position 2-3 selects manual control of the contrast adjustment using potentiometer R31 (default setting).

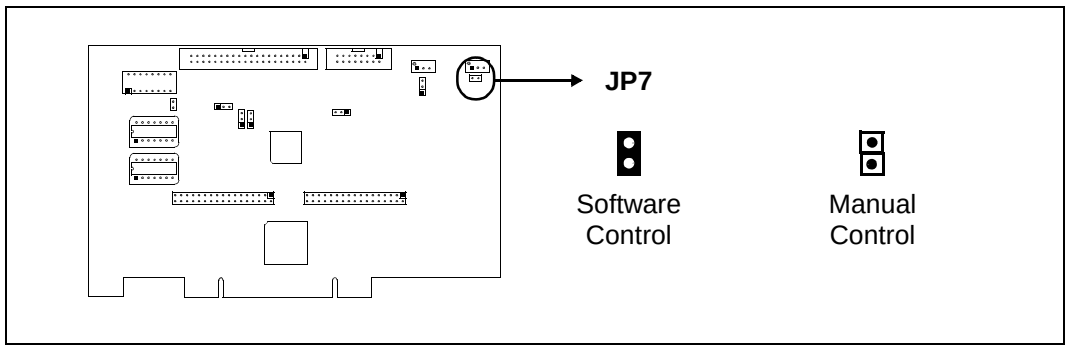


Figure 3-8: Configuration Jumper (JP7) Location

4 CPU Interface

4.1 CPU Interface Pin Mapping

Table 4-1: CPU Interface Pin Mapping

S1D13706 Pin Name	Generic #1	Generic #2	Hitachi SH-3 /SH-4	Motorola MC68K #1	Motorola MC68K #2	Motorola REDCAP2	Motorola MC68EZ328/ MC68VZ328 DragonBall
AB[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]
AB0	A0[1]	A0	A0[1]	LDS#	A0	A0[1]	A0[1]
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0][2]	D[15:0]	D[15:0]
CS#	External Decode		CSn#	External Decode		CSn#	CSA#
M/R#	External Decode						
CLKI	BUSCLK	BUSCLK	CKIO	CLK	CLK	CLK	CLK
BS#	Connected to V_{DD}[3]		BS#	AS#	AS#	Connected to V_{DD}[3]	
RD/WR#	RD1#	Connected to V_{DD}[3]	RD/WR#	R/W#	R/W#	R/W#	Connected to V_{DD}[3]
RD#	RD0#	RD#	RD#	Connected to V_{DD}[3]	SIZ1	OE#	OE#
WE0#	WE0#	WE#	WE0#	Connected to V_{DD}[3]	SIZ0	EB1#	LWE#
WE1#	WE1#	BHE#	WE1#	UDS#	DS#	EB0#	UWE#
WAIT#	WAIT#	WAIT#	WAIT#/ RDY#	DTACK#	DSACK1#	N/A	DTACK#
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	RESET#

Note

[1] A0 for these busses is not used internally by the S1D13706.

[2] If the target MC68K bus is 32-bit, then these signals should be connected to D[31:16].

[3] These pins are not used in their corresponding Host Bus Interface mode. Systems are responsible for externally connecting them to the host interface IO V_{DD}.

4.2 CPU Bus Connector Pin Mapping

Table 4-2: CPU Bus Connector (H3) Pinout

Connector Pin No.	Comments
1	Connected to DB0 of the S1D13706
2	Connected to DB1 of the S1D13706
3	Connected to DB2 of the S1D13706
4	Connected to DB3 of the S1D13706
5	Ground
6	Ground
7	Connected to DB4 of the S1D13706
8	Connected to DB5 of the S1D13706
9	Connected to DB6 of the S1D13706
10	Connected to DB7 of the S1D13706
11	Ground
12	Ground
13	Connected to DB8 of the S1D13706
14	Connected to DB9 of the S1D13706
15	Connected to DB10 of the S1D13706
16	Connected to DB11 of the S1D13706
17	Ground
18	Ground
19	Connected to DB12 of the S1D13706
20	Connected to DB13 of the S1D13706
21	Connected to DB14 of the S1D13706
22	Connected to DB15 of the S1D13706
23	Connected to RESET# of the S1D13706
24	Ground
25	Ground
26	Ground
27	+12 volt supply
28	+12 volt supply
29	Connected to WE0# of the S1D13706
30	Connected to WAIT# of the S1D13706
31	Connected to CS# of the S1D13706
32	Connected to MR# of the S1D13706
33	Connected to WE1# of the S1D13706
34	Connected to TXVDD1

Table 4-3: CPU Bus Connector (H4) Pinout

Connector Pin No.	Comments
1	Connected to A0 of the S1D13706
2	Connected to A1 of the S1D13706
3	Connected to A2 of the S1D13706
4	Connected to A3 of the S1D13706
5	Connected to A4 of the S1D13706
6	Connected to A5 of the S1D13706
7	Connected to A6 of the S1D13706
8	Connected to A7 of the S1D13706
9	Ground
10	Ground
11	Connected to A8 of the S1D13706
12	Connected to A9 of the S1D13706
13	Connected to A10 of the S1D13706
14	Connected to A11 of the S1D13706
15	Connected to A12 of the S1D13706
16	Connected to A13 of the S1D13706
17	Ground
18	Ground
19	Connected to A14 of the S1D13706
20	Connected to A15 of the S1D13706
21	Connected to A16 of the S1D13706
22	Not connected
23	Not connected
24	Not connected
25	Ground
26	Ground
27	+5 volt supply
28	+5 volt supply
29	Connected to RD/WR# of the S1D13706
30	Connected to BS# of the S1D13706
31	Connected to BUSCLK of the S1D13706
32	Connected to RD# of the S1D13706
33	Not connected
34	Not connected

5 LCD Interface Pin Mapping

Table 5-1: LCD Signal Connector (H1)

Pin Name	Connector Pin No.	Monochrome Passive		Color Passive Panel				Color TFT Panel				
		Single		Single				Others			Sharp HR-TFT[1]	Epson D-TFD[1]
					Format 1	Format 2						
		4-bit	8-bit	4-bit	8-bit	8-bit	16-Bit	9-bit	12-bit	18-bit	18-bit	18-bit
FPDAT0	1	driven 0	D0	driven 0	D0 (B5)[1]	D0 (G3)[1]	D0 (R6)[1]	R2	R3	R5	R5	R5
FPDAT1	3	driven 0	D1	driven 0	D1 (R5)[1]	D1 (R3)[1]	D1 (G5)[1]	R1	R2	R4	R4	R4
FPDAT2	5	driven 0	D2	driven 0	D2 (G4)[1]	D2 (B2)[1]	D2 (B4)[1]	R0	R1	R3	R3	R3
FPDAT3	7	driven 0	D3	driven 0	D3 (B3)[1]	D3 (G2)[1]	D3 (R4)[1]	G2	G3	G5	G5	G5
FPDAT4	9	D0	D4	D0 (R2)[1]	D4 (R3)[1]	D4 (R2)[1]	D8 (B5)[1]	G1	G2	G4	G4	G4
FPDAT5	11	D1	D5	D1 (B1)[1]	D5 (G2)[1]	D5 (B1)[1]	D9 (R5)[1]	G0	G1	G3	G3	G3
FPDAT6	13	D2	D6	D2 (G1)[1]	D6 (B1)[1]	D6 (G1)[1]	D10 (G4)[1]	B2	B3	B5	B5	B5
FPDAT7	15	D3	D7	D3 (R1)[1]	D7 (R1)[1]	D7 (R1)[1]	D11 (B3)[1]	B1	B2	B4	B4	B4
FPDAT8	17	driven 0	driven 0	driven 0	driven 0	driven 0	D4 (G3)[1]	B0	B1	B3	B3	B3
FPDAT9	19	driven 0	driven 0	driven 0	driven 0	driven 0	D5 (B2)[1]	driven 0	R0	R2	R2	R2
FPDAT10	21	driven 0	driven 0	driven 0	driven 0	driven 0	D6 (R2)[1]	driven 0	driven 0	R1	R1	R1
FPDAT11	23	driven 0	driven 0	driven 0	driven 0	driven 0	D7 (G1)[1]	driven 0	driven 0	R0	R0	R0
FPDAT12	25	driven 0	driven 0	driven 0	driven 0	driven 0	D12 (R3)[1]	driven 0	G0	G2	G2	G2
FPDAT13	27	driven 0	driven 0	driven 0	driven 0	driven 0	D13 (G2)[1]	driven 0	driven 0	G1	G1	G1
FPDAT14	29	driven 0	driven 0	driven 0	driven 0	driven 0	D14 (B1)[1]	driven 0	driven 0	G0	G0	G0
FPDAT15	31	driven 0	driven 0	driven 0	driven 0	driven 0	D15 (R1)[1]	driven 0	B0	B2	B2	B2
FPDAT16	4	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B1	B1	B1
FPDAT17	6	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B0	B0	B0
FPSHIFT	33	FPSHIFT									CLK	XSCL
DRDY	35 & 38	MOD			FPSHIFT2	MOD		DRDY			no connect	GCP
FPLINE	37	FPLINE									LP	LP
FPFRAME	39	FPFRAME									SPS	DY
GND	2, 8, 14, 20, 26	GND										
PWMOUT	28	PWMOUT										
VLCD	30	Adjustable -24V to -8V negative LCD bias										
VCC	32	LCDVCC (3.3V / 5.0V)										
+12V	34	+12V										
VDDH	36	Adjustable +20V to +40V positive LCD bias										
GPO	40	GPO[2] (for controlling on-board LCD bias power supply on/off)									MOD[3]	GPO[2]

Note

[1]These pin mappings use signal names commonly used for each panel type, however signal names may differ between panel manufacturers. The values shown in brackets represent the color components as mapped to the corresponding FPDATxx signals at the first valid edge of FPSHIFT. For further FPDATxx to LCD interface mapping, see *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

[2]GPO on H1 can be inverted by setting JP4 to 2-3.

[3]The Sharp HR-TFT MOD signal controls the panel power. This must not be confused with the MOD signal used on many passive panels.

Table 5-2: Extended LCD Signal Connector (H2)

Pin Name	Connector Pin No.	Monochrome Passive Panel		Color Passive Panel				Color TFT Panel				
		Single		Single				Others			HR-TFT[1]	D-TFD[1]
					Format 1	Format 2						
		4-bit	8-bit	4-bit	8-bit	8-bit	16-Bit	9-bit	12-bit	18-bit	18-bit	18-bit
GPIO0	1	GPIO0									PS	XINH
GPIO1	3	GPIO1									CLS	YSCL
GPIO2	5	GPIO2									REV	FR
GPIO3	7	GPIO3									SPL	FRS
GPIO4	9	GPIO4									GPIO4	RES
GPIO5	11	GPIO5									GPIO5	DD_P1
GPIO6	13	GPIO6									GPIO6	YSCLD
CVOUT	15	CVOUT										
GND	2, 4, 6, 8, 10, 12, 14, 16	GND										

Note

[1] When dip switch SW1-4 is open (CNF3 = 0 at RESET#), GPIO[6:0] are at low output states after reset. If REG[10h] bits[1:0] are set for either HR-TFT or D-TFD, some of the pins are used for the HR-TFT or D-TFD interfaces and are not available as GPIO pins.

6 Technical Description

6.1 PCI Bus Support

The S1D13706 **does not** have on-chip PCI bus interface support. The S1D13706B00C uses the PCI Bridge FPGA to support the PCI bus.

6.2 Direct Host Bus Interface Support

The S5U13706B00C is specifically designed to work using the PCI Bridge FPGA in a standard PCI bus environment. However, the S1D13706 directly supports many other host bus interfaces. Connectors H3 and H4 provide the necessary IO pins to interface to these host buses. For further information on the host bus interfaces supported, see "CPU Interface" on page 15.

Note
The PCI Bridge FPGA must be disabled using SW1-10 in order for direct host bus interface to operate properly.

6.3 S1D13706 Embedded Memory

The S1D13706 has 80K bytes of embedded SRAM. The 80K byte display buffer address space is directly and contiguously available through the 17-bit address bus.

6.4 Manual/Software Adjustable LCD Panel Positive Power Supply (VDDH)

Most passive LCD color and passive single monochrome LCD panels require a positive bias voltage between +24V and +40V. The S5U13706B00C uses a Maxim MAX754 LCD Contrast Controller to provide this voltage range. The signal VDDH can be adjusted manually (using a potentiometer) or controlled through software.

When JP5 is set to position 1-2, VDDH can be controlled through software to provide an output voltage from +20V to +40V. CVOUT and GPO of the S1D13706 are connected to LADJ and LON of MAX754. The output voltage (VDDH) can be adjusted from +20V to +40V in 64 steps by sending pulses to CVOUT. Each CVOUT pulse decrements VDDH one step towards +20V. When decremented beyond +20V, VDDH resets to +40V again. In other words, 63 pulses equal incrementing 1 step. After the MAX754 is reset (see "Controlling the MAX754" on page 21), VDDH is set at +30V.

The S5U13706B00C uses GPO and CVOUT to control the MAX754 as shown in the following table..

Table 6-1: Controlling the MAX754

Signal	Turn MAX754 On	Turn MAX754 Off	Reset MAX754
GPO	high	low	low
CVOUT	X	low	high

X = don't care

When JP5 is set to position 2-3, VDDH is adjustable using R24 (200Ω potentiometer) to provide an output voltage from +24V to +40V.

Note
When manually adjusting the voltage, set the potentiometer according to the panel's specific power requirements **before connecting the panel.**

6.5 Manual/Software Adjustable LCD Panel Negative Power Supply (VLCD)

Most passive monochrome LCD panels require a negative bias voltage between -14V and -24V. The S5U13706B00C uses a Maxim MAX749 Digitally Adjustable LCD Bias Supply to provide this voltage range. The signal VLCD can be adjusted manually (using a potentiometer) or controlled through software.

When JP7 is set to position 1-2, VLCD can be controlled through software to provide an output voltage from -8V to -24V. CVOUT and GPO of the S1D13706 are connected to ADJ and CTRL of MAX749. The output voltage (VLCD) can be adjusted from -8V to -24V in 64 steps by sending pulses to CVOUT. Each CVOUT pulse increments VLCD one step towards -24V. When decremented beyond -24V, VLCD resets to -8V again. In other words, 63 pulses equal incrementing 1 step. After the MAX749 is reset (see "Controlling the MAX749" on page 21), VLCD is set at -16V.

The S5U13706B00C uses GPO and CVOUT to control the MAX749 as shown in the following table..

Table 6-2: Controlling the MAX749

Signal	Turn MAX749 On	Turn MAX749 Off	Reset MAX749
GPO	high	low	low
CVOUT	X	low	high

X = don't care

When jumper JP7 is set to position 2-3, VLCD can be adjusted by R41 (500K potentiometer) to provide an output voltage from -16V to -23V.

Note
When using manual adjust, set the potentiometer according to the panel's specific power requirements **before connecting the panel.**

6.6 Software Adjustable LCD Backlight Intensity Support Using PWM

The S1D13706 provides Pulse Width Modulation output on PWMOUT. PWMOUT can be used to control LCD panels which support PWM control of the backlight inverter. The PWMOUT signal is provided on the buffered LCD connector (H1).

6.7 Passive/Active LCD Panel Support

The S1D13706 directly supports:

- 4/8-bit, single monochrome passive panels.
- 4/8/16-bit single color passive panels.
- 9/12/18-bit TFT active matrix panels.
- 18-bit Sharp HR-TFT panels.
- 18-bit Epson D-TFD panels.

All the necessary signals are provided on the 40-pin LCD connector H1. For connection information, refer to Table 5-1: “LCD Signal Connector (H1)” on page 18.

6.7.1 Buffered LCD Connector

The buffered LCD connector (H1) provides the same LCD panel signals as those directly from S1D13706, but with voltage-adapting buffers selectable to 3.3V or 5.0V. Pin 32 on this connector provides a voltage level of 3.3V or 5.0V to the LCD panel logic (see “JP6 - LCD Panel Voltage” on page 14 for information on setting the panel voltage).

6.7.2 Extended LCD Connector

The S1D13706 directly supports Sharp 18-bit HR-TFT and Epson 18-bit D-TFD panels. The extended LCD connector (H3) provides the extra signals required to support these panels. The signals on this connector are also buffered from the S1D13706 and adjustable to 3.3V or 5.0V (see “JP6 - LCD Panel Voltage” on page 14 for details on setting the panel voltage).

7 Clock Synthesizer and Clock Options

For maximum flexibility, the S5U13706B00C implements a Cypress ICD2061A Clock Generator. MCLKOUT from the clock synthesizer is connected to CLKI2 of the S1D13706 and VCLKOUT from the clock synthesizer is connected to CLKI of the S1D13706. A 14.31818MHz crystal (Y1) is connected to XTALIN and XTALOUT of the clock synthesizer.

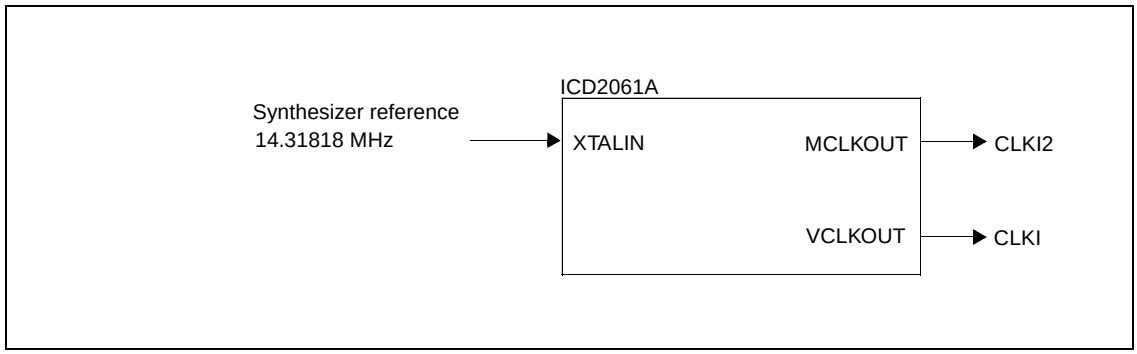


Figure 7-1: Symbolic Clock Synthesizer Connections

At power-on, CLKI2 (MCLKOUT) is configured to be 40MHz and CLKI (VCLKOUT) is configured at 25.175MHz.

Note

If an Epson D-TFD panel is selected, the clock synthesizer cannot be programmed, and external oscillators must provide the clock signals to CLKI and CLKI2. Jumpers JP2 and JP3 allow selection of external oscillators U5 and U6 as the clock source for both CLKI and CLKI2. For further information, see Table 3-2: “Jumper Summary” on page 11.

7.1 Clock Programming

The S1D13706 utilities automatically program the clock generator. If manual programming of the clock generator is required, refer to the source code for the S1D13706 utilities available on the internet at www.eea.epson.com.

For further information on programming the clock generator, refer to the *Cypress ICD2061A specification.*

Note

When CLKI and CLKI2 are programmed to multiples of each other (e.g. CLKI = 20MHz, CLKI2 = 40MHz), the clock output signals from the Cypress clock generator may jitter. Refer to the Cypress ICD2061A specification for details.

To avoid this problem, set CLKI and CLKI2 to different frequencies use the S1D13706 internal clock divides to obtain the lower frequencies.

8 References

8.1 Documents

- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.
- Epson Research and Development, Inc., *S1D13806 Programming Notes and Examples*, document number X31B-G-003-xx.
- Cypress Semiconductor Corporation, *ICD2061A Data Sheet.*

8.2 Document Sources

- Epson Electronics America Website: http://www.eea.epson.com.
- Cypress Semiconductor Corporation Website: http://www.cypress.com.

9 Parts List

Table 9-1: Parts List

Item	Qty	Designation	Part Value	Description	Manufacturer / Part No. / Assembly Instructions
1	21	"C1-C11,C13,C16-C21,C25,C27,C29"	0.1u	"50V X7R +/-5%, 1206 pckg."	
2	2	"C26,C12"	10u 10V	10u 10V	"Tantalum C-Size, 10V +/-10%"
3	2	"C15,C14"	n/p	1206 pckg.	Do not populate
4	2	"C22,C28"	22u 10V	"Tantalum C-Size, 10V +/-10%"	
5	10	"C23,C38,C39,C40,C41,C42,C43,C44,C45,C46"	0.22u	"50V X7R +/-5%, 1206 pckg"	
6	2	"C24,C32"	10u 63V	"Electrolytic, Radial Lead 63V +/-20%"	NIPPON/UNITED CHEMI-CON KMF63VB10RM5X11LL or equivalent
7	4	"C30,C34,C35,C37"	68u 10V	"Tantalum D-Size, 10V +/-10%"	
8	1	C31	1n	"50V X7R +/-5%, 1206 pckg"	
9	2	"C36,C33"	33u 20V	"Tantalum D-Size, 20V +/-10%"	
10	2	"D2,D1"	1N5819	"Schottky Barrier Rectifier, MELF pckg."	Lite-on 1N5819M or equivalent
11	1	H1	HEADER 20X2	"20x2, .025"" sq. shrouded header, keyed"	Thomas&Betts P/N:636-4207 or equivalent
12	1	H2	HEADER 8X2	"8x2, .025"" sq. shrouded header, keyed"	Thomas&Betts P/N:636-1607 or equivalent
13	2	"H4,H3"	HEADER 17X2	"17x2, .025"" sq. unshrouded header"	
14	2	"JP7,JP1"	HEADER 2	"2x1 .1"" pitch unshrouded header"	
15	5	"JP2,JP3,JP4,JP5,JP6"	HEADER 3	"3x1 .1"" pitch unshrouded header"	
16	2	"L2,L1"	47uH	"Shielded SMT power inductor, +/-20%, 1.17A, 0.18 ohm"	J.W.Miller PM105S-470M or Digi-key M1033CT-ND or equivalent
17	1	Q1	MMBT3906	PNP Transistor / SOT-23	Motorola or equivalent
18	1	Q2	MMFT3055VL	"N-channel FET, SOT-223 pckg."	Motorola MMFT3055VL or equivalent
19	1	Q3	FZT792A	"High gain transistor, SOT-223 pckg."	Zetex FZT792A or FZT751
20	2	"Q4,Q5"	MMBT2222A	"NPN transistor, SOT-23 pckg."	Motorola or equivalent
21	14	"R1-R9,R33,R36,R37,R38,R39"	15K	1206 / 5%	
22	9	"R10,R11,R12,R13,R14,R15,R16,R17,R18"	330K	1206 / 5%	
23	1	R19	12.4K 1%	"1206 / 1%, E-96 series"	
24	2	"R20,R21"	80K	1206 / 5%	

Table 9-1: Parts List

Item	Qty	Designation	Part Value	Description	Manufacturer / Part No. / Assembly Instructions
25	1	R22	402 1%	"1206 / 1%, E-96 series"	
26	1	R23	301 1%	"1206 / 1%, E-96 series"	
27	1	R24	200 POT	Trim POT	Spectrol 63S201 or equivalent
28	1	R25	0.22 1/4W	1210 / 5% / 1/4W	Panasonic ERJ-14RQJR22 or equivalent
29	1	R26	470	1206 / 5%	
30	1	R27	22K	1206 / 5%	
31	3	"R28,R29,R32"	100K	1206 / 5%	
32	1	R30	1.2M	1206 / 5%	
33	1	R31	500K POT	Trim POT	Spectrol 63S504 or equivalent
34	4	"R34,R35,R40,R41"	1K	1206 / 5%	
35	1	SW1	SW DIP-10	Dip Switch 10-Position	
36	1	S1	SW DIP-4	"DIP switch, 4-position"	"Do not populate, Do not purchase"
37	1	U1	S1D13706F00A	100-pin TQFP15 surface mount package	"Do not purchase, supplied by EPSON R&D"
38	1	U2	LT1117CST-5	"5V fixed voltage regulator, SOT-223"	Linear Technology LT1117CST-5
39	1	U3	74AHC04	SO-14 package	"NS 74VHC04 or TI 74AHC04, SO-14 package"
40	1	U4	ICD2061A	Wide SO-16 package	Cypress ICD2061A
41	2	"U6,U5"	Test Socket	"14 pin narrow DIP, screw machine socket"	
42	4	"U7,U8,U9,U10"	74HCT244	SO-20 package	
43	1	U11	MAX754	16 pin narrow SO pckg.	Maxim MAX754CSE or MAX754ESE
44	1	U12	LT1117CM-3.3	"3.3V fixed volt reg / M package, Plastic DD"	Linear Technology LT1117CST-3.3
45	1	U13	MAX749	8 pin SO pckg.	Maxim MAX749CSA or MAX749ESA
46	1	U14	EPF6016TC144-2	144-pin QFP	Altera EPF6016TC144-2
47	1	U15	8-pin DIP socket	8-pin DIP socket	"Machined socket, 8-pin"
48	1	(U15)	EPC1441PC8	8-pin DIP pckg	"Altera EPC1441PC8, programmed, socketed"

Table 9-1: Parts List

Item	Qty	Designation	Part Value	Description	Manufacturer / Part No. / Assembly Instructions
49	1	Y1	14.31818MHz	"Fundamental Mode, Parallel Resonant Crystal, HC49 Low Profile pckg."	FOXS/143-20 or equivalent
50	7	(JP1-JP7)	Micro Shunt		
51	1		Bracket	"Computer Bracket, Blank - PCI"	Keystone - Cat. No. 9203
52	2		Screw	"Pan head, #4-40 x 1/4"""	"Screw, pan head, #4-40 x 1/4""--please assemble bracket onto board"

10 Schematics

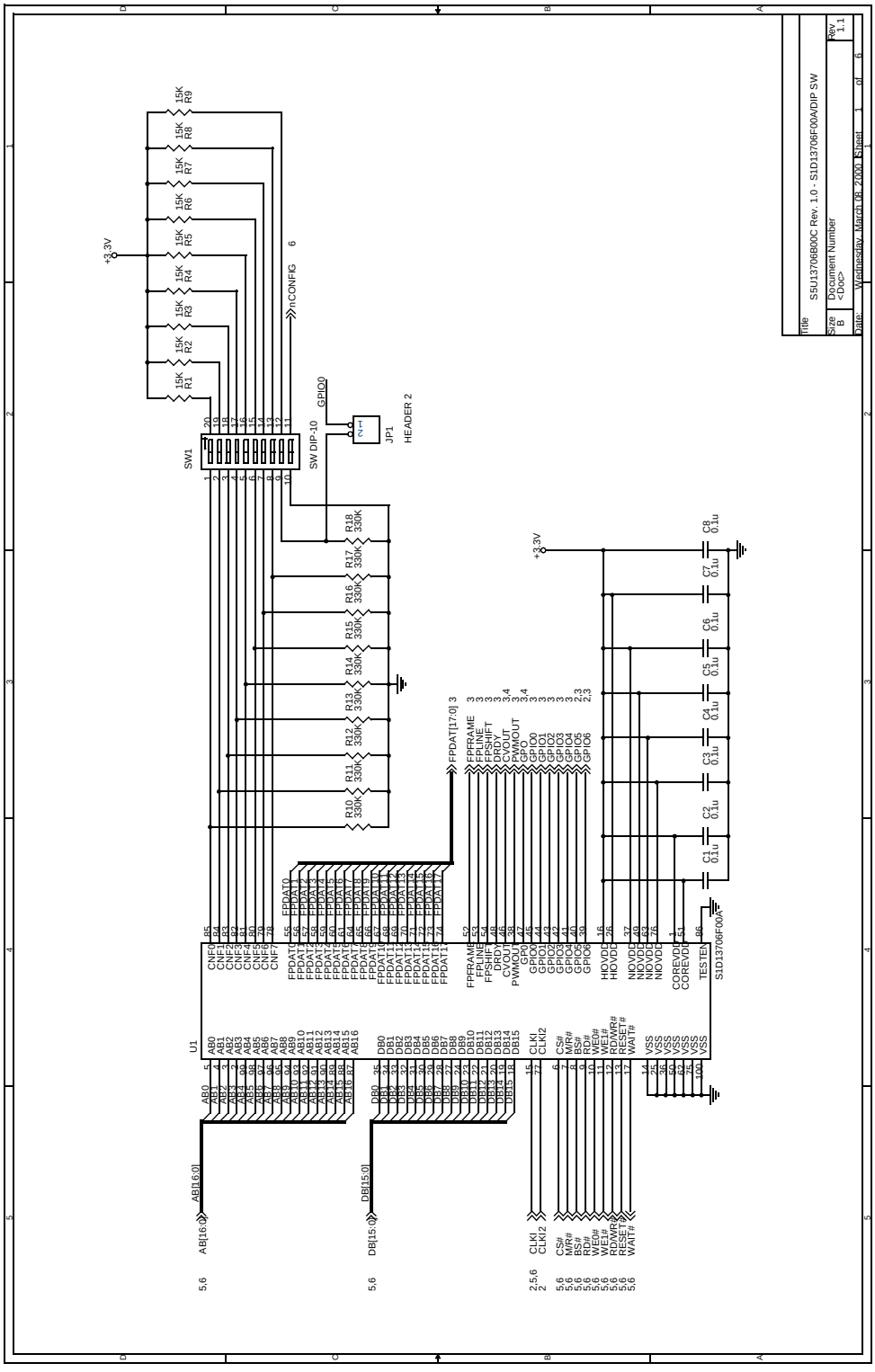


Figure 10-1: S1D13706B00C Schematics (1 of 6)

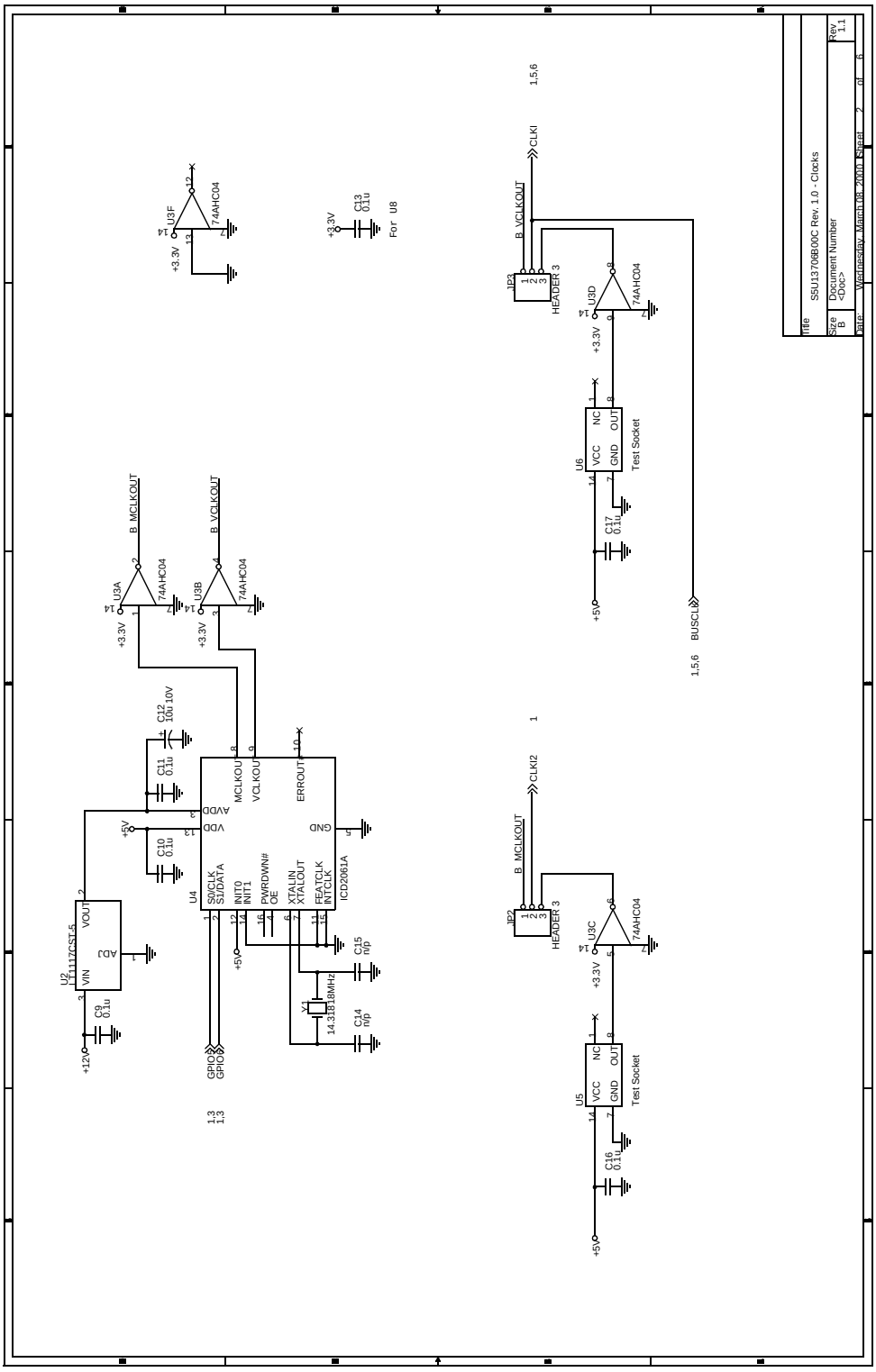


Figure 10-2: S1D13706B00C Schematics (2 of 6)

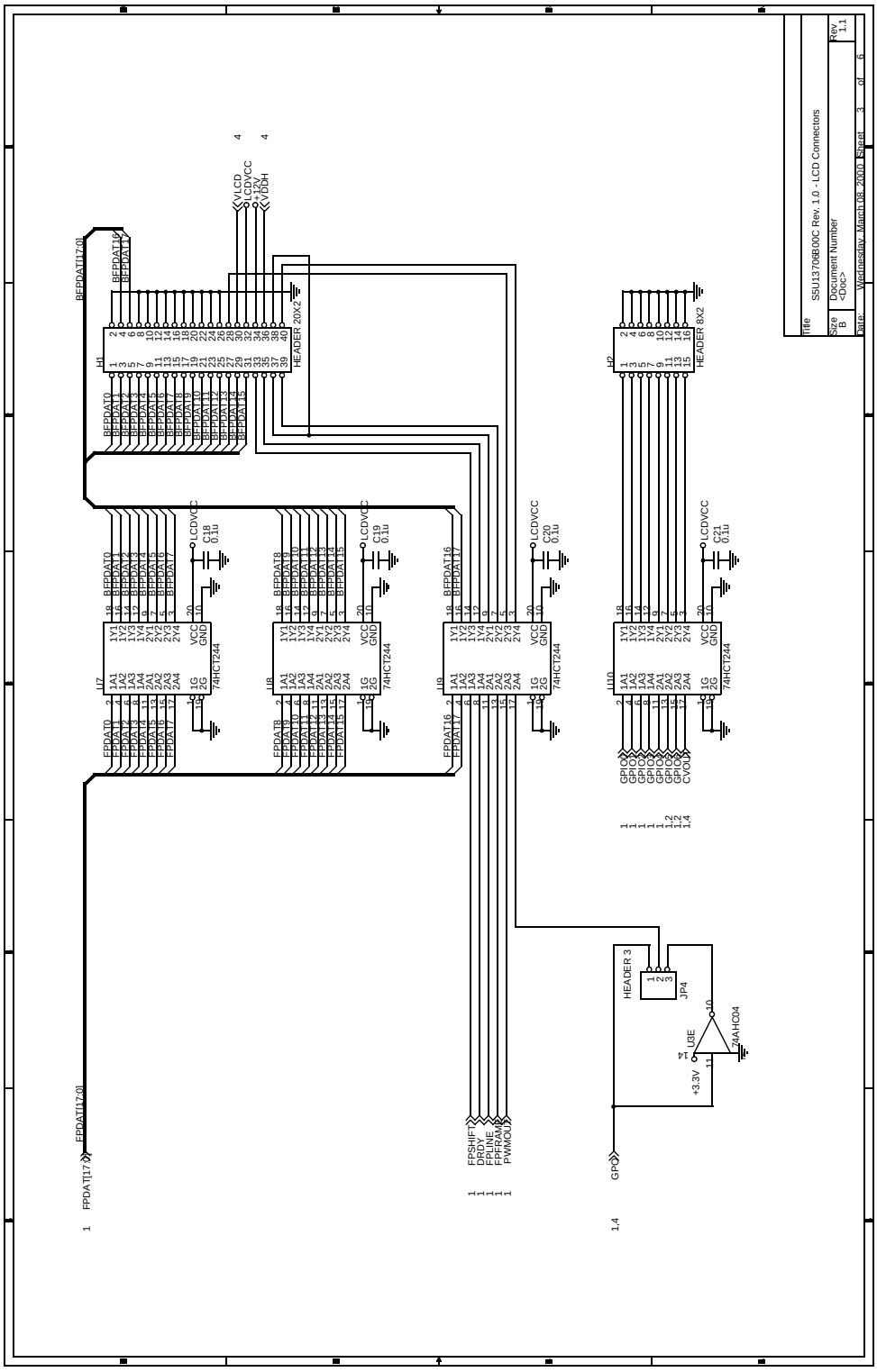


Figure 10-3: S1D13706B00C Schematics (3 of 6)

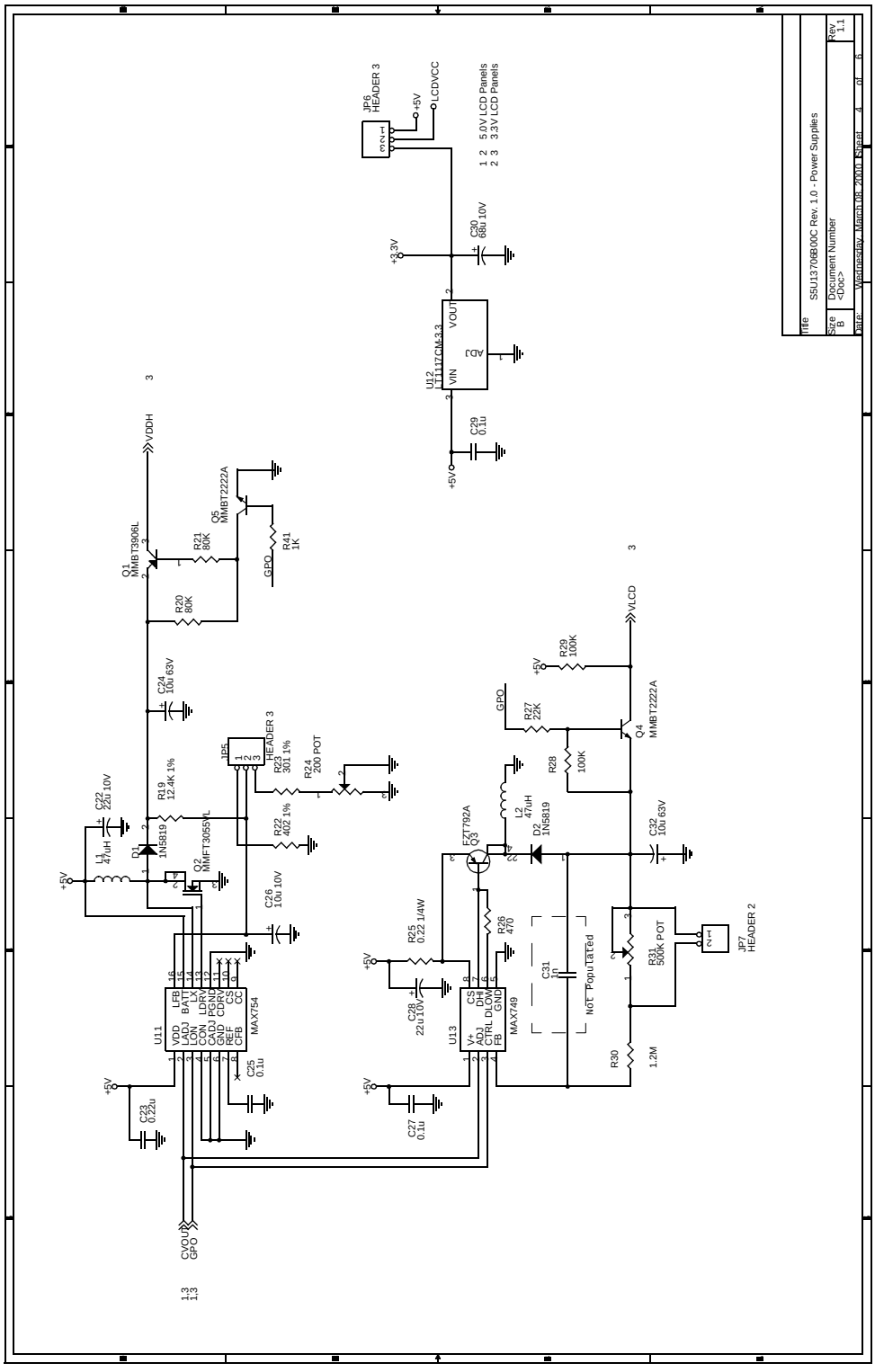


Figure 10-4: S1D13706B00C Schematics (4 of 6)

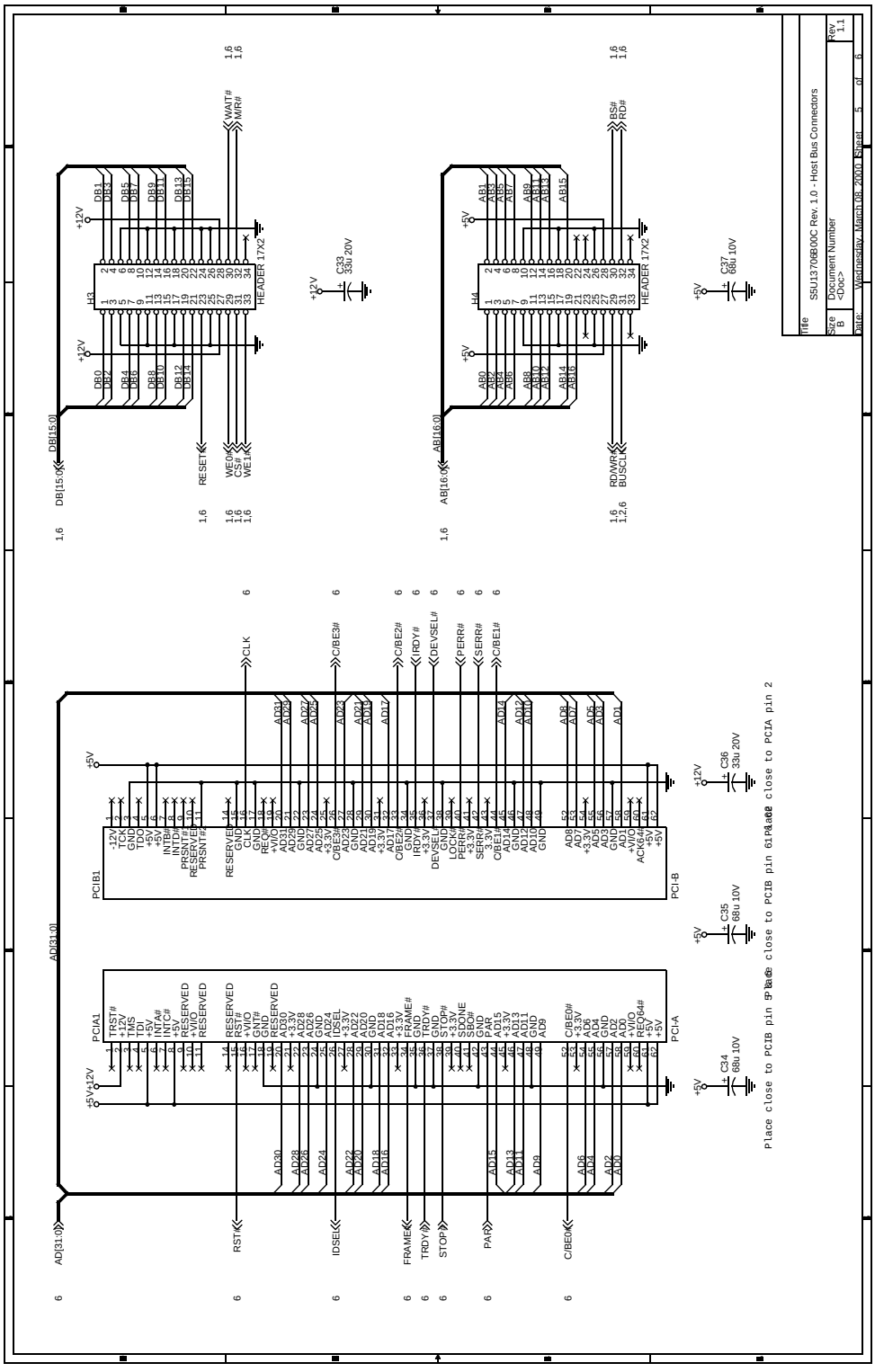


Figure 10-5: S1D13706B00C Schematics (5 of 6)

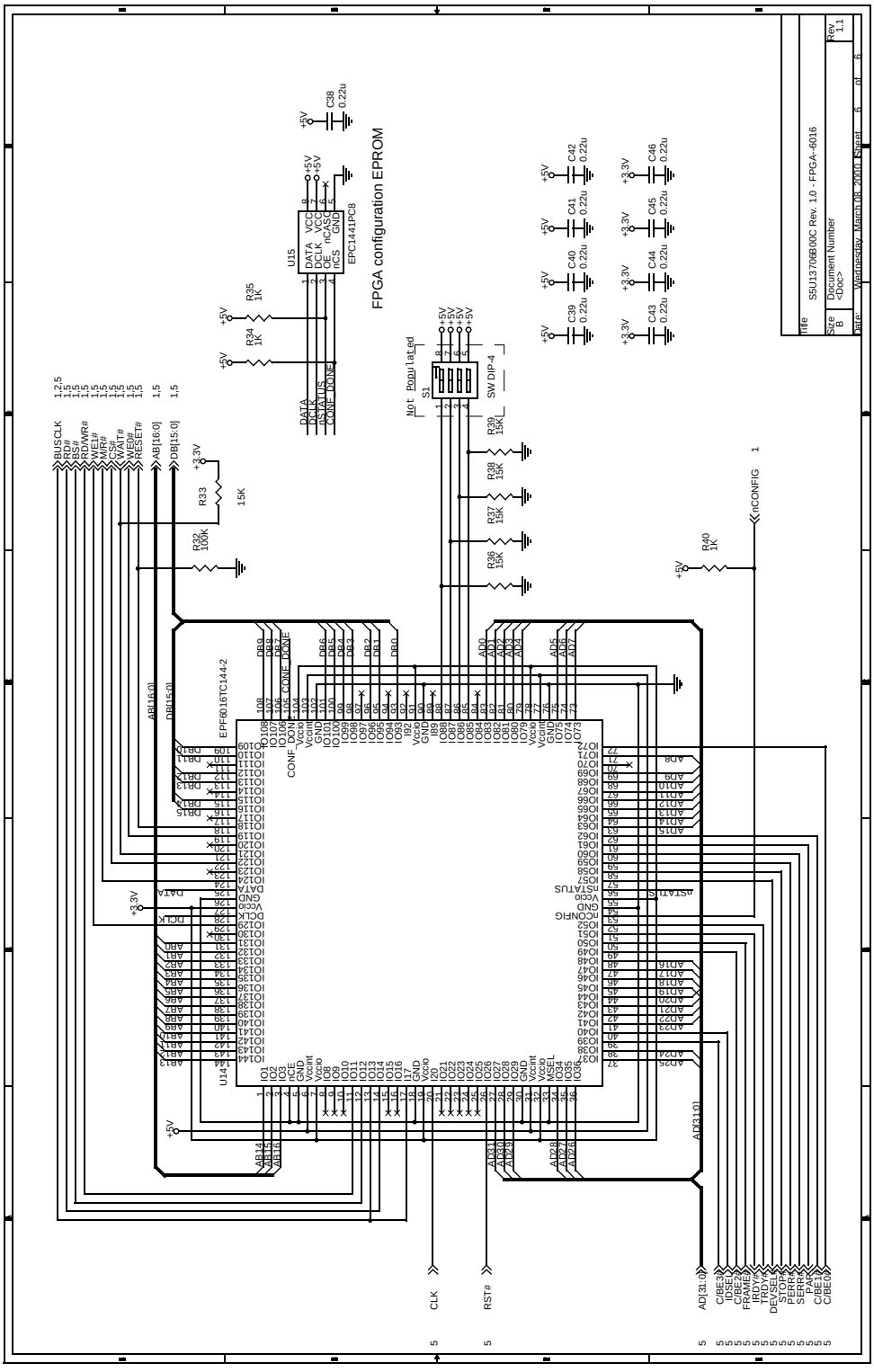


Figure 10-6: S1D13706B00C Schematics (6 of 6)

11 Board Layout

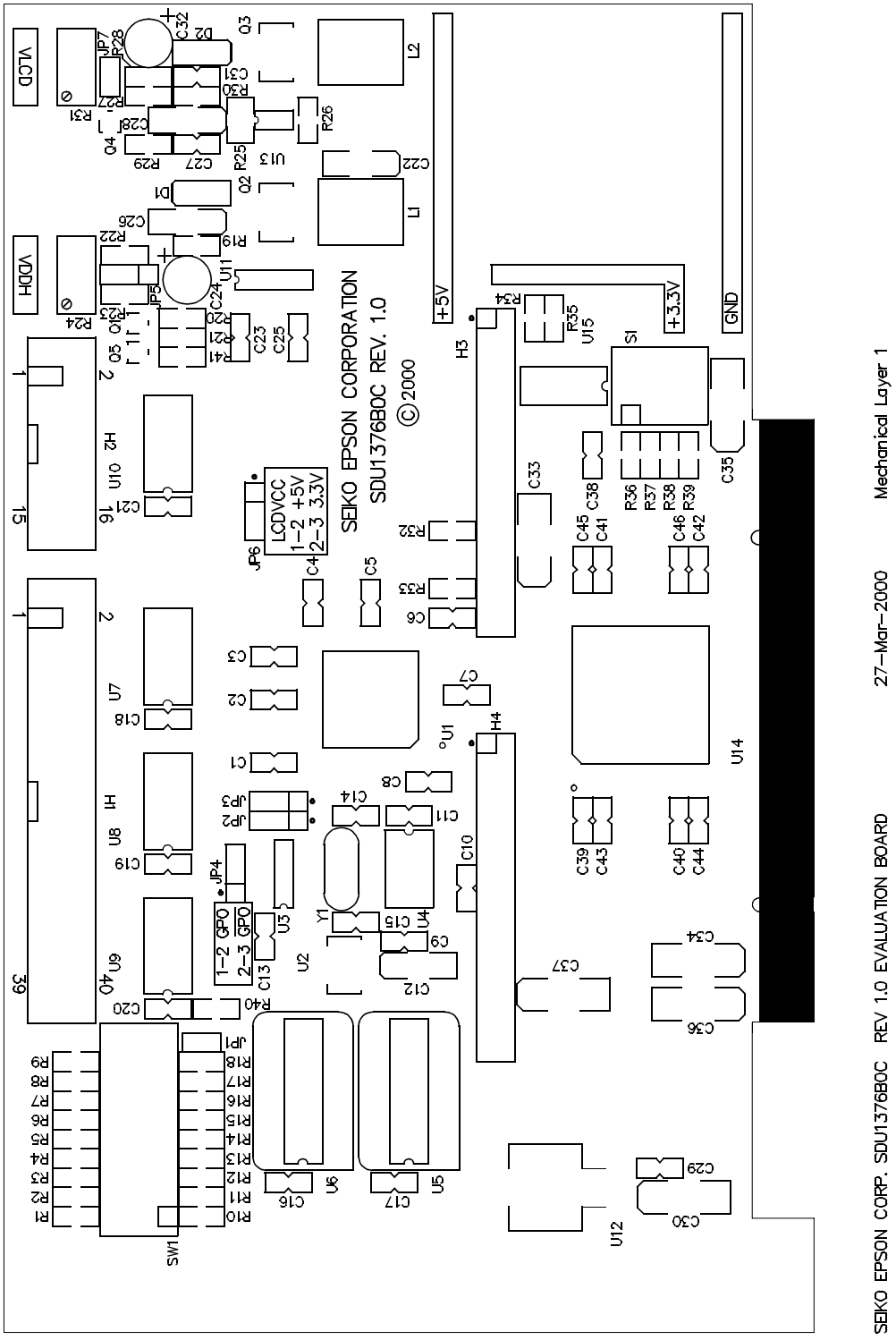


Figure 11-1: S5U13706B00C Board Layout

12 Technical Support

12.1 EPSON LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

THIS PAGE LEFT BLANK

EPSON®

S1D13706 Embedded Memory LCD Controller

Interfacing to the Toshiba MIPS TMPR3905/3912 Microprocessors

Document Number: X31B-G-002-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment necessary to provide an interface between the S1D13706 Embedded Memory LCD Controller and the Toshiba MIPS TMPR3905/3912 processors.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at techpubs@erd.epson.com.

2 Interfacing to the TMPR3905/12

2.1 The Toshiba TMPR3905/12 System Bus

The TMPR39XX family of processors features a high-speed system bus typical of modern MIPS RISC microprocessors. This section provides an overview of the operation of the CPU bus in order to establish interface requirements.

2.1.1 Overview

The TMPR3905/12 is a highly integrated controller developed for handheld products. The microprocessor is based on the R3900 MIPS RISC processor core. The TMPR3905/12 implements an external 26-bit address bus and a 32-bit data bus allowing it to communicate with its many peripheral units. The address bus is multiplexed (A[12:0]) using an address latch signal (ALE) which controls the driving of the address onto the address bus. The full 26-bit address bus (A[25:0]) is generated to devices not capable of receiving a multiplexed address, using external latches (controlled by ALE).

The TMPR3905/12 provides two, revision 2.01 compliant, PC Card slots. The 16-bit PC Card slots provide a 26-bit multiplexed address and additional control signals which allow access to three 64M byte address ranges: IO, memory, and attribute space. The signal CARDREG* selects memory space when high and attribute or IO space when low. Memory and attribute space are accessed using the write and read enable signals (WE* and RD*). When CARDREG* is low, card IO space is accessed using separate write (CARDIOWR*) and read (CARDIORD*) control signals.

2.1.2 Card Access Cycles

A data transfer is initiated when the address is placed on the PC Card bus and one, or both, of the card enable signals (CARD1CSL* and CARD1CSH*) are driven low. CARDREG* is inactive for memory and IO cycles. If only CARD1CSL* is driven low, 8-bit data transfers are enabled and A0 specifies whether the even or odd data byte appears on the PC Card data bus lines D[7:0]. If only CARD1CSH* is driven low, an odd byte transfer occurs on PC Card data lines D[15:8]. If both CARD1CSL* and CARD1CSH* are driven low, a 16-bit word transfer takes place on D[15:0].

During a read cycle, either RD* or CARDIORD* is driven low depending on whether a memory or IO cycle is specified. A write cycle is specified by driving WE* (memory cycle) or CARDIOWR* (IO cycle) low. The cycle can be lengthened by driving CARD1WAIT* low for the time required to complete the cycle.

Figure 2-1: "Toshiba 3905/12 PC Card Memory/Attribute Cycle," illustrates a typical memory/attribute cycle on the Toshiba 3905/12 PC Card bus.

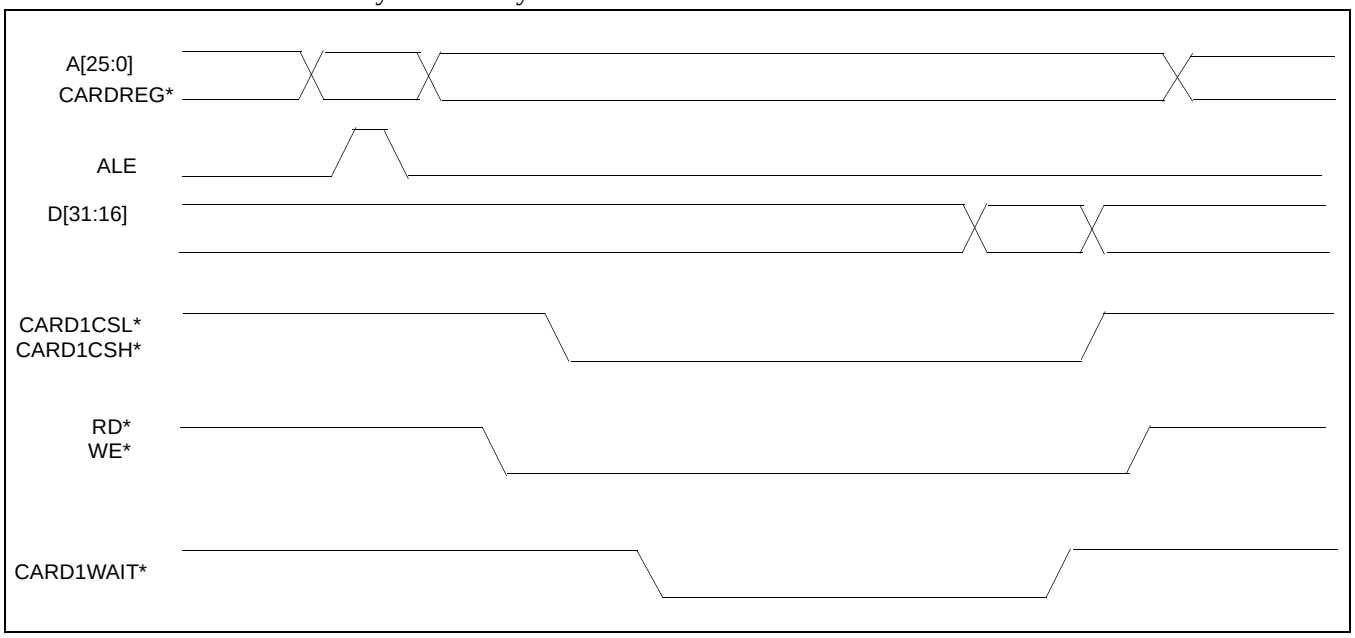


Figure 2-1: Toshiba 3905/12 PC Card Memory/Attribute Cycle

Figure 2-2: "Toshiba 3905/12 PC Card IO Cycle," illustrates a typical IO cycle on the Toshiba 3905/12 PC Card bus.

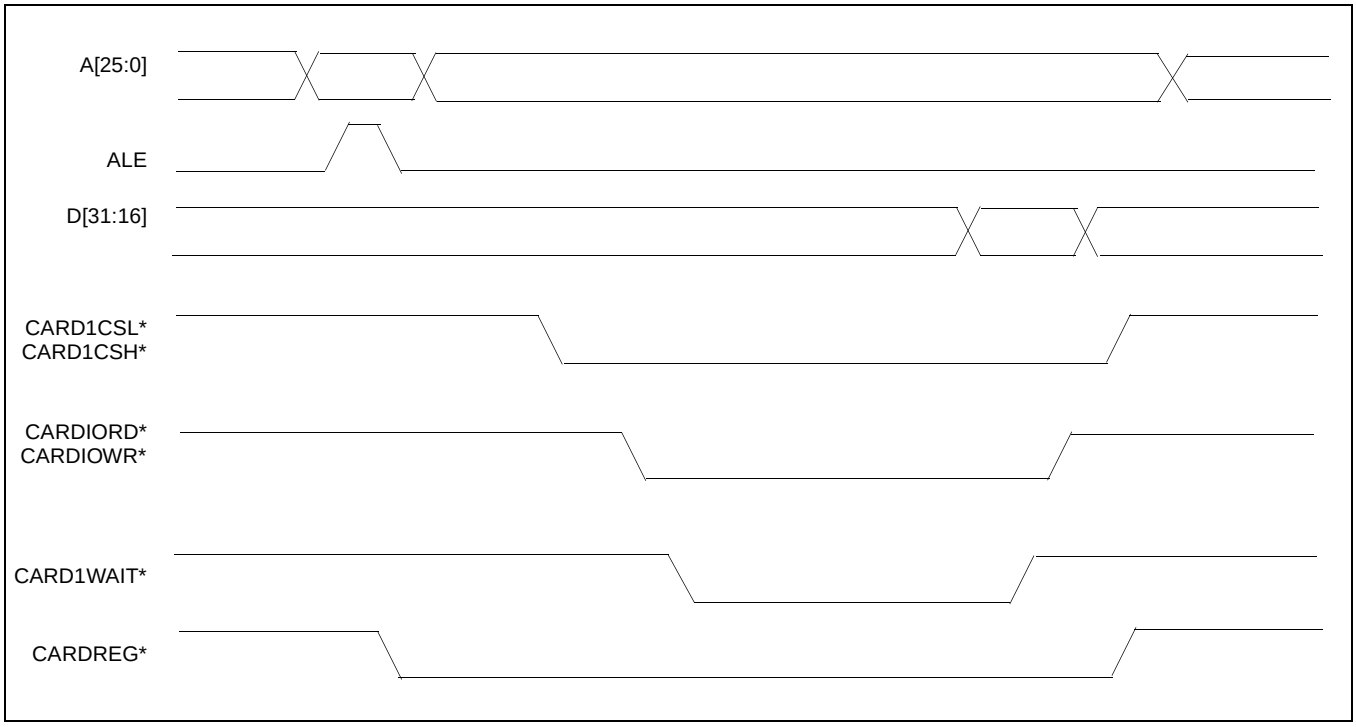


Figure 2-2: Toshiba 3905/12 PC Card IO Cycle

3 S1D13706 Host Bus Interface

The S1D13706 directly supports multiple processors. The S1D13706 implements a 16-bit Generic #2 Host Bus Interface which is most suitable for connection to the Toshiba TMPR3905/12 microprocessor.

The Generic #2 Host Bus Interface is selected by the S1D13706 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on the S1D13706 configuration, see Section 4.2, "S1D13706 Hardware Configuration" on page 14.

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

S1D13706 Pin Names	Toshiba TMPR3905/12
AB[16:0]	External Decode
DB[15:8]	D[23:16]
DB[7:0]	D[31:24]
WE1#	External Decode
CS#	External Decode
M/R#	External Decode
CLKI	DCLKOUT
BS#	connect to HIO V_{DD}
RD/WR#	connect to HIO V_{DD}
RD#	CARDIORD*
WE0#	CARDIOWR*
WAIT#	CARD1WAIT*
RESET#	system $\overline{\text{RESET}}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals.

- CLKI is a clock input required by the S1D13706 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For example, DCLKOUT from the Toshiba TMPR3905/12.
- The address inputs AB[12:0] are connected directly to the TMPR3905/12 address bus. Since the TMPR3905/12 has a multiplexed address bus, the other address inputs A[16:13] must be generated using an external latch controlled by the address latch enable signal (ALE). The low data byte on the TMPR3905/12 data bus for 16-bit ports is D[31:24] and connects to the S1D13706 low data byte, D[7:0]. The high data byte on the TMPR3905/12 data bus for 16-bit ports is D[23:16] and connects to the S1D13706 high data byte, D[15:0]. The hardware engineer must ensure that CNF4 selects the proper endian mode upon reset.
- Chip Select (CS#) is driven by external decoding circuitry to select the S1D13706.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A17 to be connected to the M/R# line. This address line must be generated from the external latch used to provide the upper addresses to the S1D13706.
- WE1# is connected to CARD1CSH* and is the high byte enable for both read and write cycles.
- WE0# is connected to CARDIOWR* (the write enable signal) and must be driven low when the Toshiba TMPR3905/12 is writing data to the S1D13706.
- RD# is connected to CARDIORD* (the read enable signal) and must be driven low when the Toshiba TMPR3905/12 is reading data from the S1D13706.
- WAIT# connects to CARD1WAIT* and is a signal which is output from the S1D13706 to the TMPR3905/12 that indicates when data is ready (read cycle) or accepted (write cycle) on the host bus. Since host CPU accesses to the S1D13706 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13706 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Status (BS#) and Read/Write (RD/WR#) signals are not used in this implementation of the Toshiba TMPR3905/12 using the Generic #2 Host Bus Interface. These pins must be tied high (connected to HIO V_{DD}).

4 Toshiba TMPR3905/12 to S1D13706 Interface

4.1 Hardware Description

In this implementation, the S1D13706 occupies the TMPR3905/12 PC Card slot #1 IO address space. IO address space closely matches the timing parameters for the S1D13706 Generic #2 Host Bus Interface.

The address bus of the TMPR3905/12 PC Card interface is multiplexed and must be demultiplexed using an advanced CMOS latch (e.g., 74AHC373).

BS# (bus start) and RD/WR# are not used in this implementation and should be tied high (connected to HIO V_{DD}).

A pull-up resistor is attached to WAIT# to speed up its rise time when terminating a cycle.

The following diagram demonstrates a typical implementation of the TMPR3905/12 to S1D13706 interface.

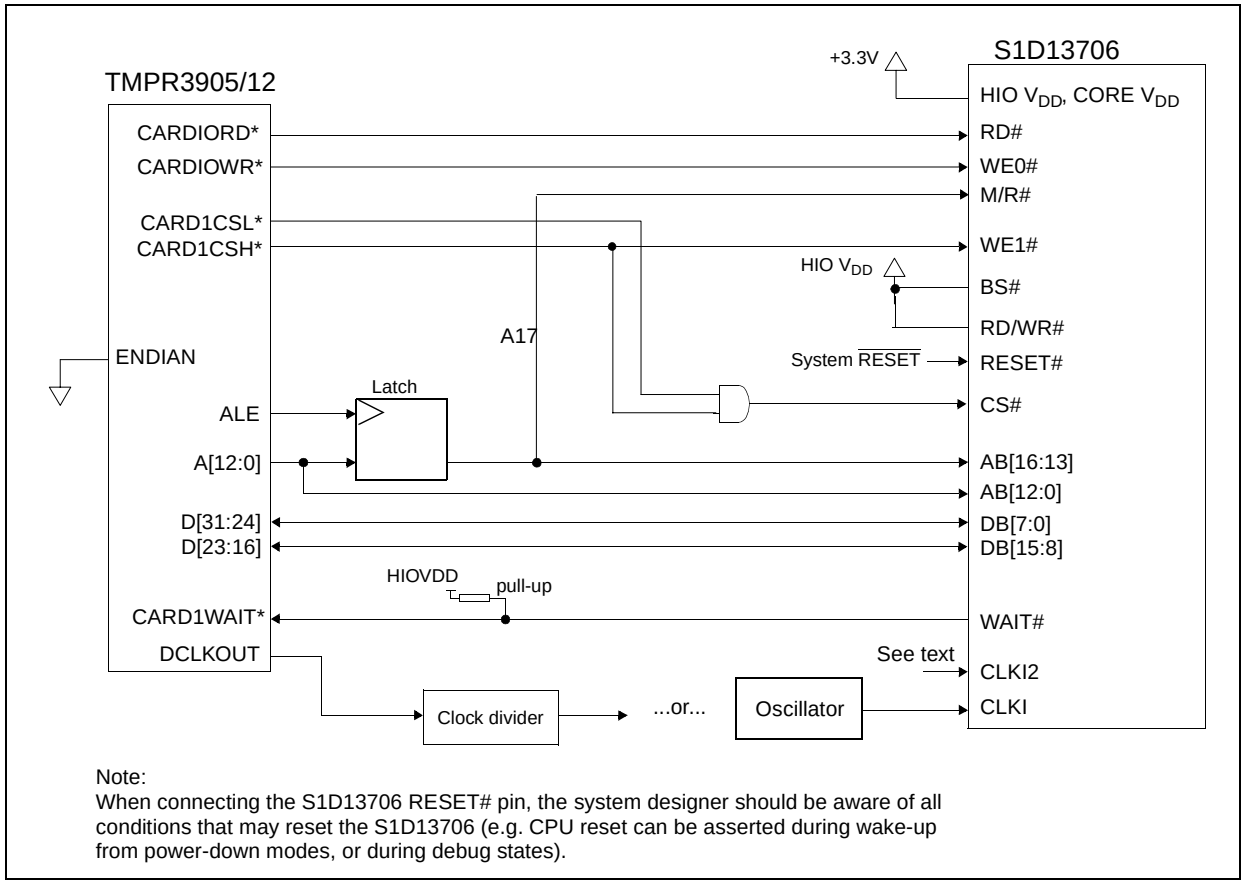


Figure 4-1: S1D13706 to TMPR3905/12 Direct Connection

The Generic #2 Host Bus Interface control signals of the S1D13706 are asynchronous with respect to the S1D13706 bus clock. This gives the system designer full flexibility to choose the appropriate source (or sources) for CLKI and CLKI2. The choice of whether both clocks should be the same, and whether to use DCLKOUT (divided) as clock source, should be based on the desired:

- pixel and frame rates.
- power budget.
- part count.
- maximum S1D13706 clock frequencies.

The S1D13706 also has internal clock dividers providing additional flexibility.

4.2 S1D13706 Hardware Configuration

The S1D13706 latches CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The table below shows the configuration settings important to the Generic #2 host bus interface used by the Toshiba TMPR3905/12.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13706 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[2:0]	100 = Generic #2 Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-2:CLKI to BCLK Divide Selection for recommended setting	

= configuration for Toshiba TMPR3905/12 microprocessor

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for TMPR3905/12 microprocessor

4.3 Memory Mapping and Aliasing

In this implementation the TMPR3905/12 control signal CARDREG* is ignored. This means that the S1D13706 takes up the entire PC Card slot 1.

The S1D13706 is a memory mapped device and uses two 128K byte blocks which are selected using A17 from the MPC821 (A17 is connected to the S1D13706 M/R# pin). The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block.

The registers occupy the range 0h through 1FFFFh while the on-chip display memory occupies the range 20000h through 3FFFFh. Demultiplexed address lines A[25:18] are ignored. Therefore, the S1D13706 is aliased 256 times at 256K byte intervals over the 64M byte PC Card slot #1 memory space.

Note

If aliasing is undesirable, additional decoding circuitry must be added.

5 Software

Test utilities and Windows® CE v2.11/2.12 display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE v2.11/2.12 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE v2.11/2.12 display drivers are available from your sales support contact or www.eea.epson.com.

6 References

6.1 Documents

- Toshiba America Electrical Components, Inc., *TMPR3905/12 Specification*.
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, Document Number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 ISA Bus Evaluation Board User Manual*, Document Number X31B-G-004-xx.
- Epson Research and Development, Inc., *S1D13706 Programming Notes and Examples*, Document Number X31B-G-003-xx.

6.2 Document Sources

- Toshiba America Electrical Components Website: http://www.toshiba.com/taec.
- Epson Electronics America Website: www.eea.epson.com.

7 Technical Support

7.1 EPSON LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

7.2 Toshiba MIPS TMPR3905/12 Processor

http://www.toshiba.com/taec/nonflash/indexproducts.html

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Interfacing to the PC Card Bus

Document Number: X31B-G-005-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13706 Embedded Memory LCD Controller and the PC Card (PCMCIA) bus.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at techpubs@erd.epson.com.

2 Interfacing to the PC Card Bus

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).

2.1.1 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 microprocessors, the high bit is the most significant. Therefore, signals A25 and D15 are the most significant bits for the address and data bus respectively.

Support is provided for on-chip DMA controllers. To find further information on these topics, refer to Section 6, *"References"* on page 15.

PC Card bus signals are asynchronous to the host CPU bus signals. Bus cycles are started with the assertion of either 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.

2.1.2 Memory Access Cycles

A data transfer is initiated when the 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 kept 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 2-1: illustrates a typical memory access read cycle on the PC Card bus.

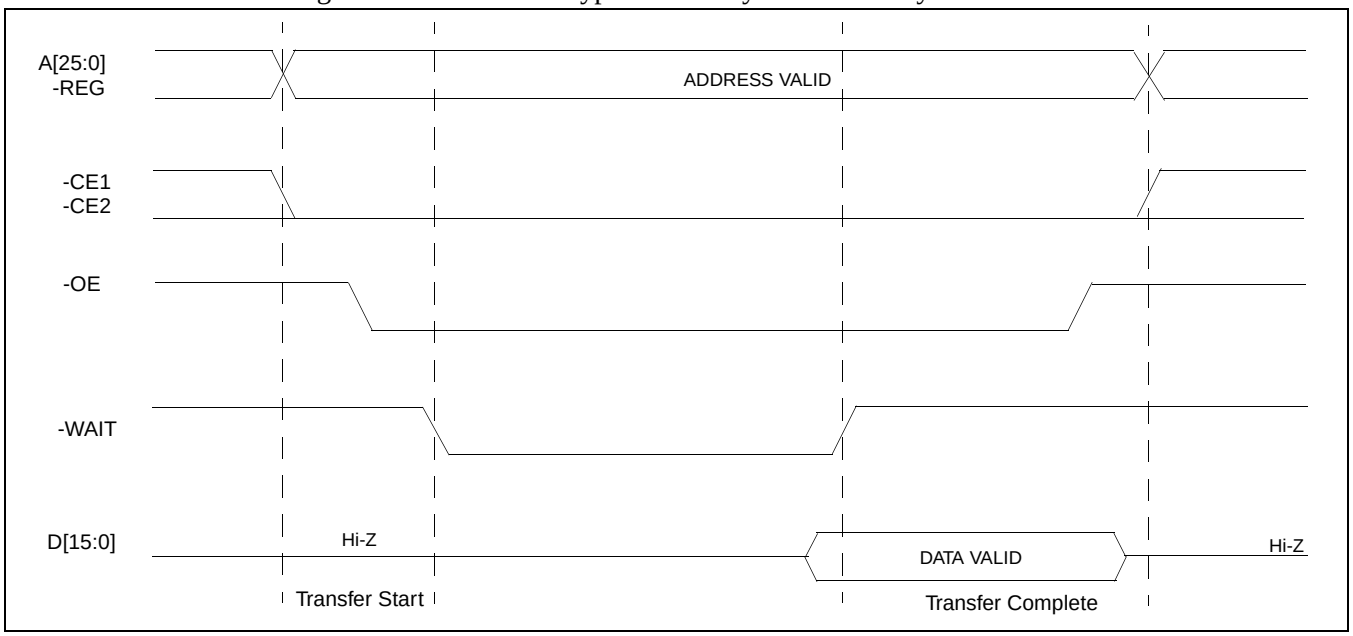


Figure 2-1: PC Card Read Cycle

Figure 2-2: illustrates a typical memory access write cycle on the PC Card bus.

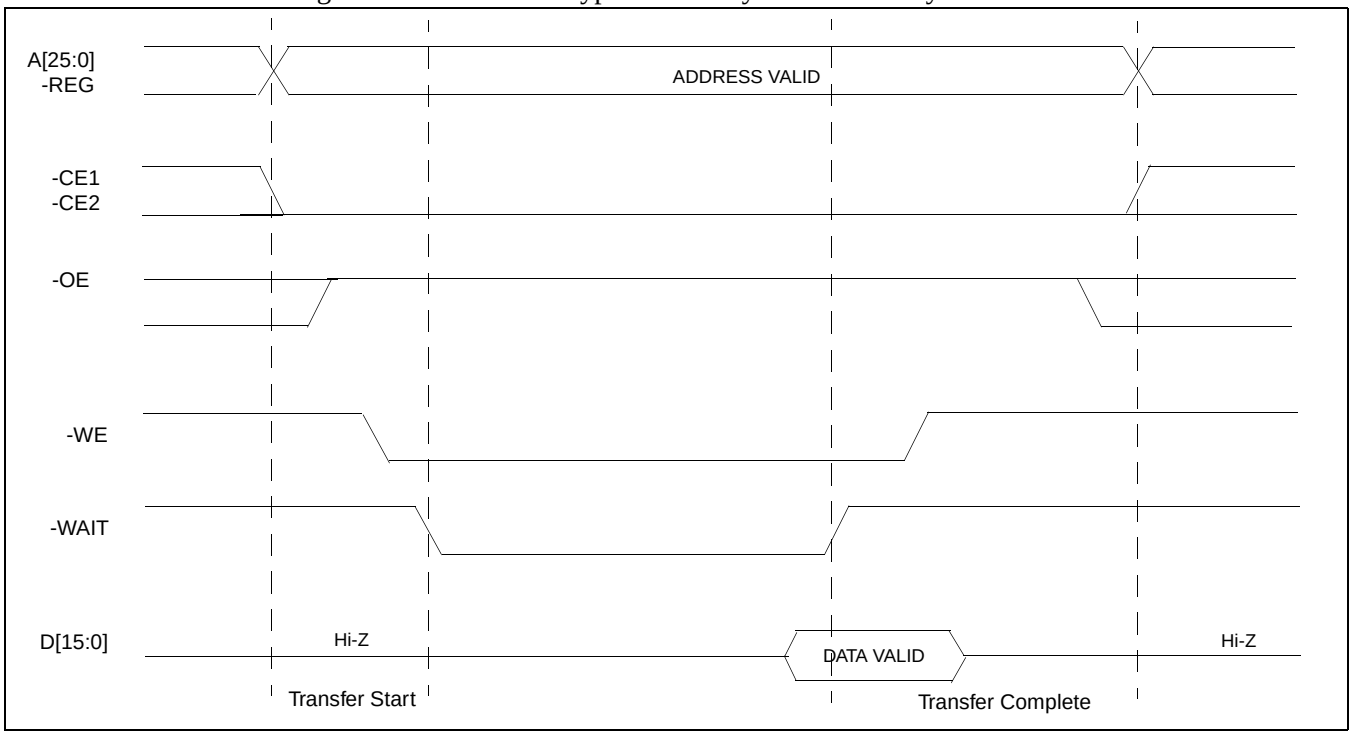


Figure 2-2: PC Card Write Cycle

3 S1D13706 Host Bus Interface

The S1D13706 directly supports multiple processors. The S1D13706 implements a 16-bit Generic #2 Host Bus Interface which is most suitable for direct connection to the PC Card bus. Generic #2 supports an external Chip Select, shared Read/Write Enable for high byte, and individual Read/Write Enable for low byte.

The Generic #2 Host Bus Interface is selected by the S1D13706 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13706 configuration, see Section 4.2, "S1D13706 Hardware Configuration" on page 13.

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

S1D13706 Pin Names	PC Card (PCMCIA)
AB[16:0]	A[16:0]
DB[15:0]	D[15:0]
WE1#	-CE2
CS#	External Decode
M/R#	A17
CLKI	see note
BS#	connect to HIO V_{DD}
RD/WR#	connect to HIO V_{DD}
RD#	-OE
WE0#	-WE
WAIT#	-WAIT
RESET#	Inverted RESET

Note

Although a clock is not directly supplied by the PC Card interface, one is required by the S1D13706 Generic #2 Host Bus Interface. For an example of how this can be accomplished see the discussion on CLKI in Section 3.2, "Host Bus Interface Signals" on page 11.

3.2 Host Bus Interface Signals

The S1D13706 Generic #2 Host Bus Interface requires the following signals from the PC Card bus.

- CLKI is a clock input which is required by the S1D13706 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. Since the PC Card signalling is independent of any clock, CLKI can come from any oscillator already implemented. For example, the source for the CLKI2 input of the S1D13706 may be used.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the PC Card address (A[16:0]) and data bus (D[15:0]), respectively. CNF4 must be set to select little endian mode.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper register and memory address space.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A17 to be connected to the M/R# line.
- WE1# is the high byte enable for both read and write cycles and connects to the PC Card high byte chip select signal (-CE2).
- WE0# connects to -WE (the write enable signal form the PC Card bus) and must be driven low when the PC Card bus is writing data to the S1D13706.
- RD# connects to -OE (the read enable signal from the PC Card bus) and must be driven low when the PC Card bus is reading data from the S1D13706.
- WAIT# is a signal output from the S1D13706 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 S1D13706 may occur asynchronously to the display update, it is possible that contention may occur in accessing the 13706 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Status (BS#) and Read/Write (RD/WR#) signals are not used in this implementation of the PC Card bus using the Generic #2 Host Bus Interface. These pins must be tied high (connected to HIO V_{DD}).
- The RESET# (active low) input of the S1D13706 may be connected to the PC Card RESET (active high) using an inverter.

4 PC Card to S1D13706 Interface

4.1 Hardware Connections

The S1D13706 is interfaced to the PC Card bus with a minimal amount of glue logic. In this implementation, the address inputs (AB[16:0]) and data bus (DB[15:0] connect directly to the CPU address (A[16:0]) and data bus (D[15:0]).

The PC Card interface does not provide a bus clock, so one must be supplied for the S1D13706. 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 CLKI2.

BS# (bus start) and RD/WR# are not used by the Generic #2 Host Bus Interface and should be tied high (connected to HIO V_{DD}).

The following diagram shows a typical implementation of the PC Card to S1D13706 interface.

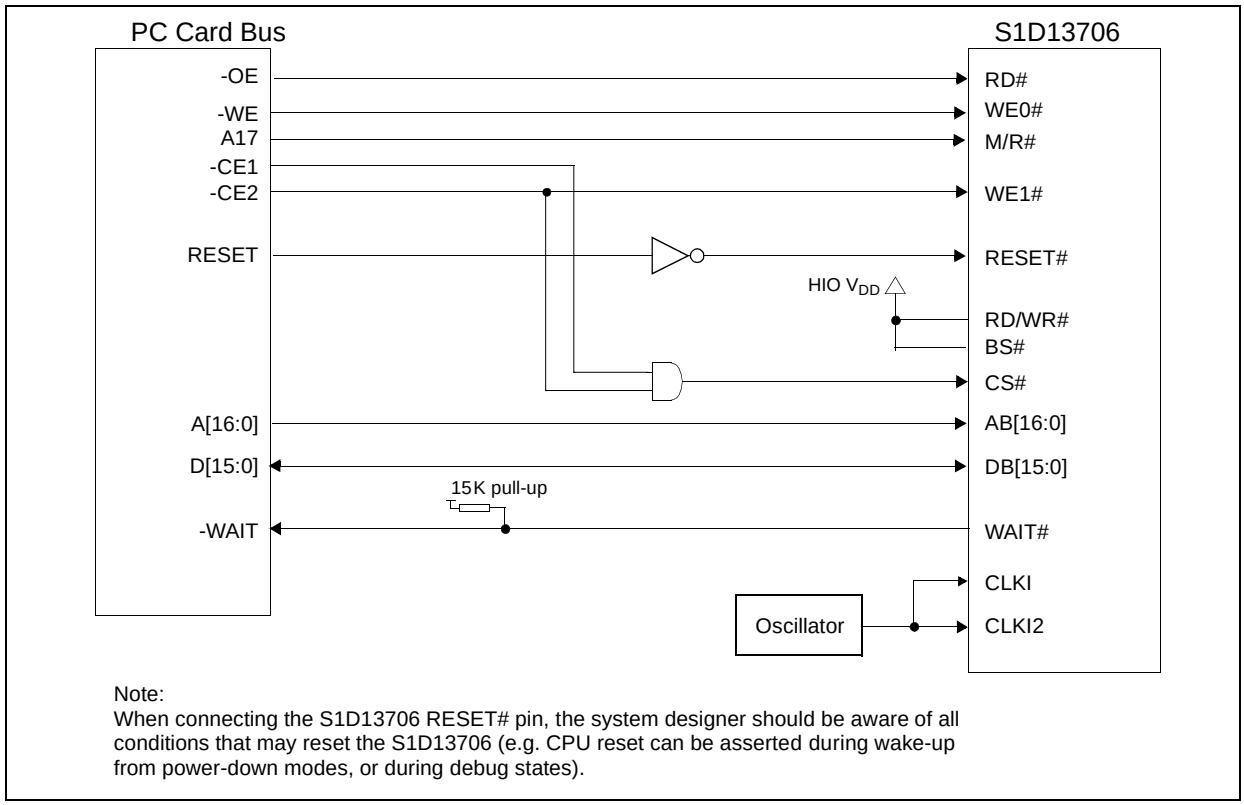


Figure 4-1: Typical Implementation of PC Card to S1D13706 Interface

4.2 S1D13706 Hardware Configuration

The S1D13706 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13706 to PC Card bus interface.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13706 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[2:0]	100 = Generic #2 Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-2: "CLKI to BCLK Divide Selection" for recommended setting	

= configuration for PC Card Bus

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for PC Card Bus

4.3 Register/Memory Mapping

The S1D13706 is a memory mapped device. The S1D13706 uses two 128K byte blocks which are selected using A17 from the PC Card bus (A17 is connected to the S1D13706 M/R# pin). The internal registers occupy the first 128K byte block and the 80K byte display buffer occupies the second 128K byte block.

The PC Card socket provides 64M bytes of memory address space. However, the S1D13706 only needs a 256K byte block of memory to accommodate its 80K byte display buffer and register set. For this reason, only address bits A[17:0] are used while A[25:17] are ignored. The S1D13706's memory and registers are aliased every 256K bytes for a total of 256 times in the 64M byte PC Card memory address space.

Note
If aliasing is not desirable, the upper addresses must be fully decoded.

5 Software

Test utilities and Windows® CE v2.11/2.12 display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE v2.11/2.12 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE v2.11/2.12 display drivers are available from your sales support contact or on the internet at http://www.eea.epson.com.

6 References

6.1 Documents

- PC Card (PCMCIA) Standard March 1997.
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 Evaluation Board User Manual*, document number X31B-G-004-xx.
- Epson Research and Development, Inc., *S1D13706 Programming Notes and Examples*, Document Number X31B-G-003-xx.

6.2 Document Sources

- PC Card website: http://www.pc-card.com.
- Epson Electronics America website: http://www.eea.epson.com

7 Technical Support

7.1 EPSON LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

7.2 PC Card Standard

PCMCIA
(Personal Computer Memory Card International Association)

2635 North First Street, Suite 209
San Jose, CA 95134
Tel: (408) 433-2273
Fax: (408) 433-9558
http://www.pc-card.com

S1D13706 Embedded Memory LCD Controller

Power Consumption

Document Number: X31B-G-006-02

THIS PAGE LEFT BLANK

1 S1D13706 Power Consumption

S1D13706 power consumption is affected by many system design variables.

- Input clock frequency (CLKI/CLKI2): the CLKI/CLKI2 frequency determines the LCD 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 S1D13706 current consumption depends on the BCLK frequency, data width, number of toggling pins, and other factors – the higher the BCLK, 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 S1D13706. The power consumption is affected by various system design variables.

- Clock states during the power save mode: disabling the clocks during power save mode has substantial power savings.

1.1 Conditions

The following table gives an example of a specific environment and its effects on power consumption.

Table 1-1: S1D13706 Total Power Consumption in mW

Test Condition *All V_{DD} = 3.3V*	**MCLK/ PCLK Ratio**	**Color Depth**	**S1D13706 Active (mW)**	**Power Save Mode**	
				Clocks Active (mW)	**Clocks Removed (mW) [1]**
LCD Panel = 60Hz 320x240 8-bit Single Color Format 2 CLKI = 6 MHz, CLKI2 = 6 MHz	1/16	1 bpp	6.58	3.02	0.00
	1/8	2 bpp	7.76	3.02	0.00
	1/4	4 bpp	8.80	3.02	0.00
	1/2	8 bpp	10.61	3.02	0.00
LCD Panel = 60Hz 320x240 4-bit Single Color CLKI = 6 MHz, CLKI2 = 6 MHz	1/2	8 bpp	11.16	3.02	0.00
LCD Panel = 60Hz 320x240 4-bit Single Monochrome CLKI = 6 MHz, CLKI2 = 6 MHz	1/2	8 bpp	9.43	3.02	0.00
LCD Panel = 60Hz 320x240 18-bit TFT CLKI = 6 MHz, CLKI2 = 6 MHz	1/2	8 bpp	8.84	3.02	0.00
LCD Panel = 60Hz 320x240 18-bit HR-TFT CLKI = 6 MHz, CLKI2 = 6 MHz	1/2	8 bpp	9.26	3.02	0.00
LCD Panel = 60Hz 320x240 18-bit D-TFD CLKI = 6 MHz, CLKI2 = 6 MHz	1/2	8 bpp	9.78	3.02	0.00
LCD Panel = 60Hz 160x240 18-bit D-TFD CLKI = 6 MHz, CLKI2 = 6 MHz	1/2	8 bpp	6.45	3.02	0.00
	1/1	16 bpp	8.12	3.02	0.00

Note

[1] CLKI and CLKI2 are stopped for this condition.

2 Summary

The system design variables in Section 1, “S1D13706 Power Consumption” and in Table 1-1: “S1D13706 Total Power Consumption in mW” show that S1D13706 power consumption depends on the specific implementation. Active Mode power consumption depends on the desired CPU performance and LCD frame-rate, whereas power save mode consumption depends on the CPU Interface and Input Clock state.

In a typical design environment, the S1D13706 can be configured to be an extremely power-efficient LCD Controller with high performance and flexibility.

THIS PAGE LEFT BLANK

EPSON®

S1D13706 Embedded Memory LCD Controller

Interfacing to the NEC VR4102 / VR4111 Microprocessors

Document Number: X31B-G-007-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13706 Embedded Memory LCD Controller and the NEC VR4102/4111 microprocessor. The NEC VR4102 and VR4111 microprocessors are specifically designed to support an external LCD controller.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at techpubs@erd.epson.com.

2 Interfacing to the NEC VR4102/VR4111

2.1 The NEC VR41XX 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 is an overview of the operation of the CPU bus to establish interface requirements.

2.1.1 Overview

The NEC VR series microprocessor is designed around the RISC architecture developed by MIPS. The VR4102 microprocessor is designed around the 66MHz VR4100 CPU core and the VR4111 is designed around the 80/100MHz VR4110 core. These microprocessors support 64-bit processing. The CPU communicates with the Bus Control Unit (BCU) through its internal SysAD bus. The BCU in turn communicates with external devices with its ADD and DATA busses which can be dynamically sized for 16 or 32-bit operation.

The NEC VR4102/VR4111 can directly support an external LCD controller through a dedicated bus interface. Specific control signals are assigned for an external LCD controller in order to provide an easy interface to the CPU. A 16M byte block of memory is assigned for the LCD controller with its own chip select and ready signals available. Word or byte accesses are controlled by the system high byte signal (SHB#).

2.1.2 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 enable (RD#) or write enable (WR#) signals are driven low for the appropriate cycle. LCDRDY is driven low by the S1D13706 to insert wait states into the cycle. The system high byte enable is driven low for 16-bit transfers and high for 8-bit transfers.

Figure 2-1: "NEC VR4102/VR4111 Read/Write Cycles," shows the read and write cycles to the LCD Controller Interface.

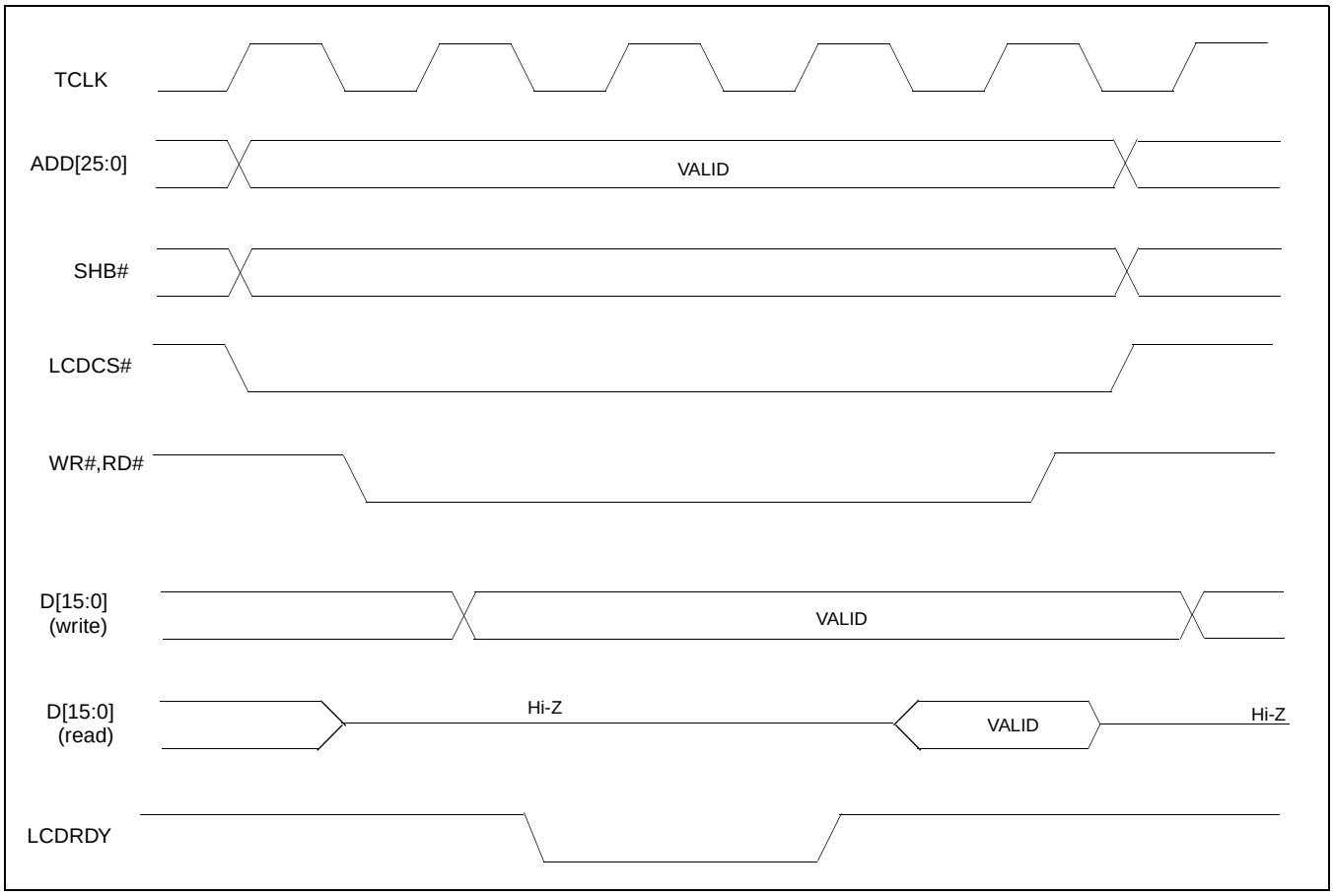


Figure 2-1: NEC VR4102/VR4111 Read/Write Cycles

3 S1D13706 Host Bus Interface

The S1D13706 directly supports multiple processors. The S1D13706 implements a 16-bit Generic #2 Host Bus Interface which is most suitable for direct connection to the NEC VR4102/4111 microprocessor. Generic #2 supports an external Chip Select, shared Read/Write Enable for high byte, and individual Read/Write Enable for low byte.

The Generic #2 Host Bus Interface is selected by the S1D13706 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13706 configuration, see Section 4.2, "S1D13706 Hardware Configuration" on page 13.

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

S1D13706 Pin Names	NEC VR4102/4111
AB[16:0]	ADD[16:0]
DB[15:0]	DAT[15:0]
WE1#	SHB#
CS#	LCDCS#
M/R#	ADD17
CLKI	BUSCLK
BS#	connect to HIO V_{DD}
RD/WR#	connect to HIO V_{DD}
RD#	RD#
WE0#	WR#
LCDRDY	WAIT#
RESET#	system $\overline{\text{RESET}}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals:

- CLKI is a clock input which is required by the S1D13706 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, BUSCLK from the NEC VR4102/4111 is used for CLKI.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the NEC VR4102/4111 address bus (ADD[16:0]) and data bus (DAT[15:0]), respectively. CNF4 must be set to select little endian mode.
- Chip Select (CS#) must be driven low by LCDCS# whenever the S1D13706 is accessed by the VR4102/4111.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address ADD17 to be connected to the M/R# line.
- WE1# connects to SHB# (the high byte enable signal from the NEC VR4102/4111) 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 NEC VR4102/4111) and must be driven low when the VR4102/4111 is writing data to the S1D13706.
- RD# connects to RD# (the read enable signal from the NEC VR4102/4111) and must be driven low when the VR4102/4111 is reading data from the S1D13706.
- WAIT# connects to LCDRDY and is a signal output from the S1D13706 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 S1D13706 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13706 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Status (BS#) and Read/Write (RD/WR#) signals are not used in this implementation of the NEC VR4102/4111 interface using the Generic #2 Host Bus Interface. These pins must be tied high (connected to HIO V_{DD}).

4 VR4102/VR4111 to S1D13706 Interface

4.1 Hardware Description

The NEC VR4102/VR4111 microprocessor is specifically designed to support an external LCD controller by providing the internal address decoding and control signals necessary. By using the Generic # 2 Host Bus Interface, no glue logic is required to interface the S1D13706 and the NEC VR4102/VR4111.

A pull-up resistor is attached to WAIT# to speed up its rise time when terminating a cycle.

BS# (bus start) and RD/WR# are not used by the Generic #2 Host Bus Interface and should be tied high (connected to HIO V_{DD}).

The following diagram shows a typical implementation of the VR4102/VR4111 to S1D13706 interface.

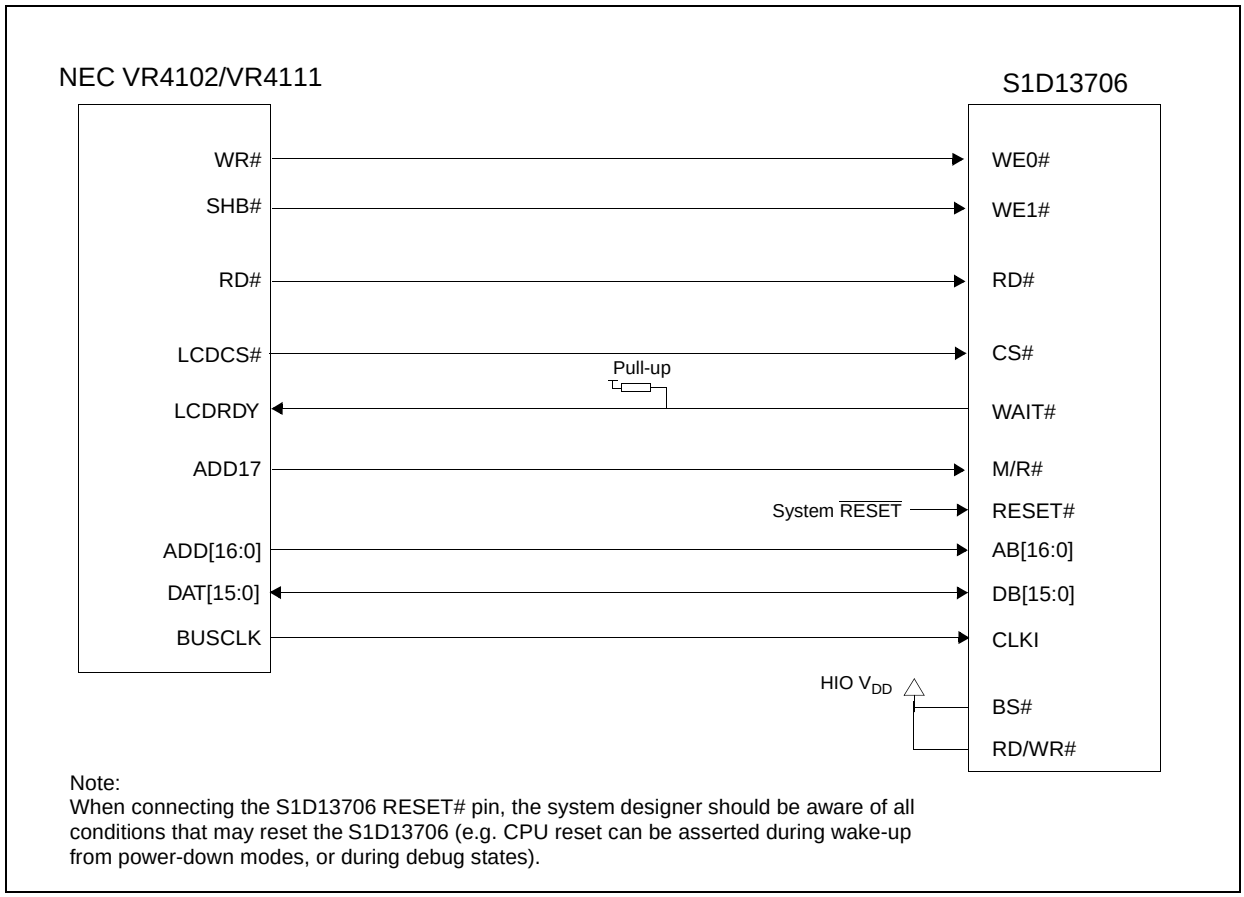


Figure 4-1: Typical Implementation of VR4102/VR4111 to S1D13706 Interface

4.2 S1D13706 Hardware Configuration

The S1D13706 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13706 to NEC VR4102/4111 interface.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13706 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[2:0]	100 = Generic #2 Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-2: “CLKI to BCLK Divide Selection” for recommended setting	

= configuration for NEC VR4102/VR4111

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for NEC VR4102/VR4111

4.3 NEC VR4102/VR4111 Configuration

The NEC VR4102/4111 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 by the NEC VR4102/4111.

The S1D13706 is a memory mapped device. The S1D13706 uses two 128K byte blocks which are selected using ADD17 from the NEC VR4102/4111 (ADD17 is connected to the S1D13706 M/R# pin).The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block.

The starting address of the S1D13706 internal registers is located at 0A00_0000h and the starting address of the display buffer is located at 0A02_0000h. These blocks are aliased over the entire 16M byte address space.

Note
If aliasing is not desirable, the upper addresses must be fully decoded.

The NEC VR4102/VR4111 has a 16-bit internal register named BCUCNTREG2 located at 0B00_0002h. It must be set to the value of 0001h which indicates that LCD controller accesses use a non-inverting data bus.

The 16-bit internal register named BCUCNTREG1 (located at 0B00_0000h) must have bit D[13] (ISA/LCD bit) set to 0. This reserves 16M bytes (from 0A00_0000h to 0AFF_FFFFh) for use by the LCD controller and not as ISA bus memory space.

5 Software

Test utilities and Windows® CE v2.11/2.12 display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE v2.11/2.12 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE v2.11/2.12 display drivers are available from your sales support contact or on the internet at http://www.eea.epson.com.

6 References

6.1 Documents

- NEC Electronics Inc., *VR4102/VR4111 64/32-bit Microprocessor Preliminary User's Manual*.
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 Evaluation Board User Manual*, document number X31B-G-004-xx.
- Epson Research and Development, Inc., *S1D13706 Programming Notes and Examples*, document number X31B-G-003-xx.

6.2 Document Sources

- NEC Electronics Inc. website: http://www.necel.com.
- Epson Electronics America website: http://www.eea.epson.com

7 Technical Support

7.1 Epson LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

7.2 NEC Electronics Inc.

NEC Electronics Inc. (U.S.A.)
Corporate Headquarters
2880 Scott Blvd.
Santa Clara, CA 95050-8062, USA
Tel: (800) 366-9782
Fax: (800) 729-9288
http://www.necel.com

THIS PAGE LEFT BLANK

EPSON®

S1D13706 Embedded Memory LCD Controller

Interfacing to the NEC VR4181A™ Microprocessor

Document Number: X31B-G-008-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13706 Embedded Memory LCD Controller and the NEC VR4181A microprocessor. The NEC VR4181A microprocessor is specifically designed to support an external LCD controller.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at techpubs@erd.epson.com.

2 Interfacing to the NEC VR4181A

2.1 The NEC VR4181A 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 VR4181A offers a highly integrated solution for portable systems. This section is an overview of the operation of the CPU bus to establish interface requirements.

2.1.1 Overview

The NEC VR4181A is designed around the RISC architecture developed by MIPS. This microprocessor is designed around the 100MHz VR4110 CPU core which supports the MIPS III and MIPS16 instruction sets. The CPU communicates with external devices via an ISA interface.

While the VR4181A has an embedded LCD controller, this internal controller can be disabled to provide direct support for an external LCD controller through its external ISA bus. A 64 to 512K byte block of memory is assigned to the external LCD controller with a dedicated chip select signal (LCDCS#). Word or byte accesses are controlled by the system high byte signal (#UBE).

2.1.2 LCD Memory Access Signals

The S1D13706 requires an addressing range of 256K bytes. When the VR4181A external LCD controller chip select signal is programmed to a window of that size, the S1D13706 resides in the VR4181A physical address range of 133C 0000h to 133F FFFFh. This range is part of the external ISA memory space.

The following signals are required to access an external LCD controller. All signals obey ISA signalling rules.

- A[16:0] is the address bus.
- #UBE is the high byte enable (active low).
- #LCDCS is the chip select for the S1D13706 (active low).
- D[15:0] is the data bus.
- #MEMRD is the read command (active low).
- #MEMWR is the write command (active low).
- #MEMCS16 is the acknowledge for 16-bit peripheral capability (active low).
- IORDY is the ready signal from S1D13706.
- SYSCLK is the prescalable bus clock (optional).

Once an address in the LCD block of memory is accessed, the LCD chip select (#LCDCS) is driven low. The read or write enable signals (#MEMRD or #MEMWR) are driven low for the appropriate cycle and IORDY is driven low by the S1D13706 to insert wait states into the cycle. The high byte enable (UBE#) is driven low for 16-bit transfers and high for 8-bit transfers.

3 S1D13706 Host Bus Interface

The S1D13706 directly supports multiple processors. The S1D13706 implements a 16-bit Generic #2 Host Bus Interface which is most suitable for direct connection to the NEC VR4181A microprocessor. Generic #2 supports an external Chip Select, shared Read/Write Enable for high byte, and individual Read/Write Enable for low byte.

The Generic #2 Host Bus Interface is selected by the S1D13706 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13706 configuration, see Section 4.2, "S1D13706 Hardware Configuration" on page 13.

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

S1D13706 Pin Names	NEC VR4181A
AB[16:0]	A[16:0]
DB[15:0]	D[15:0]
WE1#	#UBE
CS#	#LCDCS
M/R#	A17
CLKI	SYSCLK
BS#	Connect to HIO V_{DD}
RD/WR#	Connect to HIO V_{DD}
RD#	#MEMRD
WE0#	#MEMWR
WAIT#	IORDY
RESET#	RESET#

3.2 Host Bus Interface Signals

The interface requires the following signals.

- CLKI is a clock input which is required by the S1D13706 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, SYSCLK from the NEC VR4181A is used for CLKI.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the NEC VR4181A address (A[16:0]) and data bus (D[15:0]), respectively. CNF4 must be set to select little endian mode.
- Chip Select (CS#) must be driven low by #LCDCS whenever the S1D13706 is accessed by the VR4181A.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A17 to be connected to the M/R# line.
- WE1# connects to #UBE (the high byte enable signal from the NEC VR4181A) which in conjunction with address bit 0 allows byte steering of read and write operations.
- WE0# connects to #MEMWR (the write enable signal from the NEC VR4181A) and must be driven low when the NEC VR4181A is writing data to the S1D13706.
- RD# connects to #MEMRD (the read enable signal from the NEC VR4181A) and must be driven low when the NEC VR4181A is reading data from the S1D13706.
- WAIT# connects to IORDY and is a signal which is output from the S1D13706 which indicates the NEC VR4181A must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since VR4181A accesses to the S1D13706 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13706 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Status (BS#) and Read/Write (RD/WR#) signals are not used in this implementation of the NEC VR4181A interface using the Generic #2 Host Bus Interface. These pins must be tied high (connected to HIO V_{DD}).

4 VR4181A to S1D13706 Interface

4.1 Hardware Description

The NEC VR4181A microprocessor is specifically designed to support an external LCD controller by providing the internal address decoding and control signals necessary. By using the Generic # 2 Host Bus Interface, no glue logic is required to interface the S1D13706 to the NEC VR4181A.

A pull-up resistor is attached to WAIT# to speed up its rise time when terminating a cycle.

#MEMCS16 of the NEC VR4181A is connected to #LCDCS to signal that the S1D13706 is capable of 16-bit transfers.

BS# (bus start) and RD/WR# are not used by the Generic #2 Host Bus Interface and should be tied high (connected to HIO V_{DD}).

The diagram below shows a typical implementation of the VR4181A to S1D13706 interface.

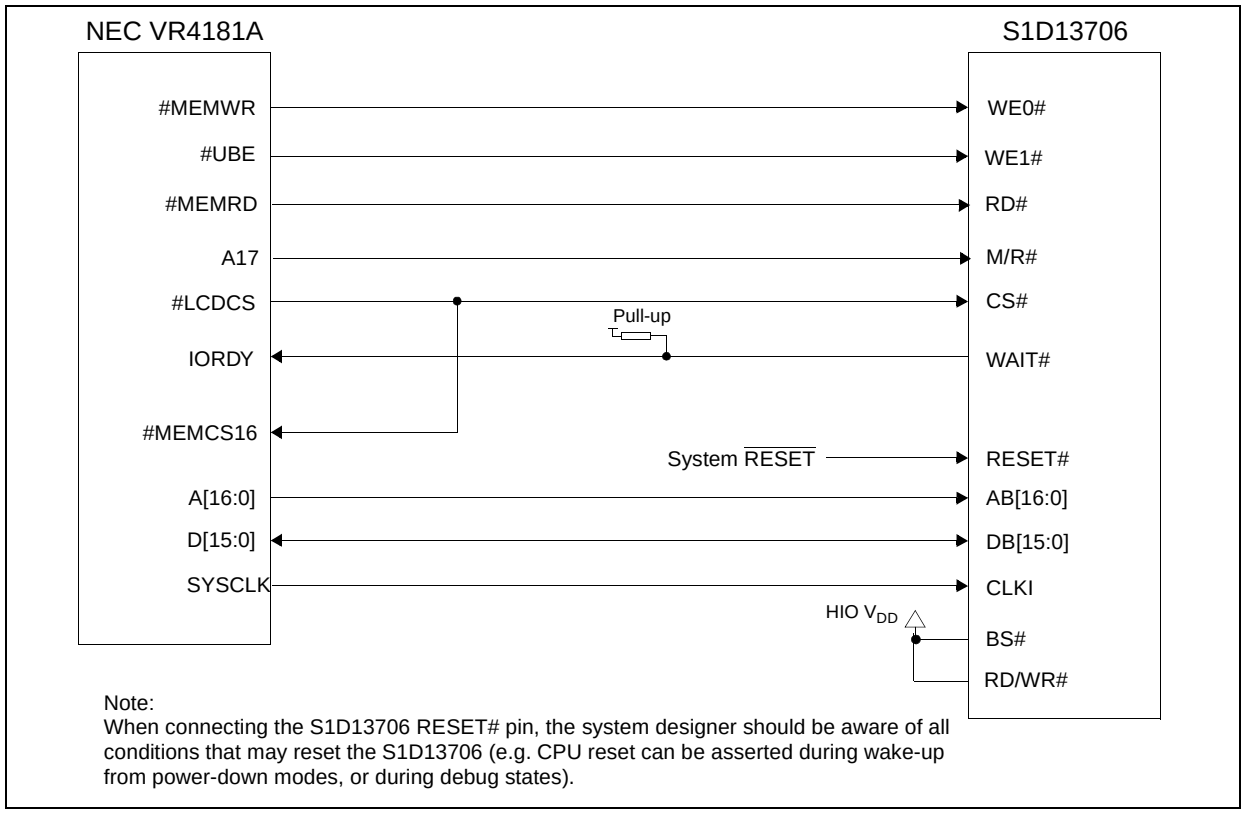


Figure 4-1: Typical Implementation of VR4181A to S1D13706 Interface

4.2 S1D13706 Hardware Configuration

The S1D13706 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13706 to NEC VR181A interface.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13706 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[2:0]	100 = Generic #2 Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table “” for recommended setting	

= configuration for NEC VR4181A

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for NEC VR4181A

4.3 NEC VR4181A Configuration

The S1D13706 is a memory mapped device. The S1D13706 uses two 128K byte blocks which are selected using A17 from the NEC VR181A (A17 is connected to the S1D13706 M/R# pin).The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block.

When the VR4181A embedded LCD controller is disabled, the external LCD controller chip select signal (#LCDCS) decodes either a 64K byte, 128K byte, 256K byte, or 512K byte memory block in the VR4181A external ISA memory. The S1D13706 requires this block of memory to be set to 256K bytes. With this configuration, the S1D13706 internal registers starting address is located at physical memory location 133C_0000h and the display buffer is located at memory location 133E_0000h.

The NEC VR4181A must be configured through its internal registers to map the S1D13706 to the external LCD controller space. The following register values must be set.

- Register LCDGPMD at address 0B00_032Eh must be set as follows.
  - Bit 7 must be set to 1 to disable the internal LCD controller and enable the external LCD controller interface. Disabling the internal LCD controller also maps pin SHCLK to #LCDCS and pin LOCLK to #MEMCS16.
  - Bits [1:0] must be set to 10b to reserve 256Kbytes of memory address range, 133C_0000h to 133F_FFFFh for the external LCD controller.
- Register GPMD2REG at address 0B00_0304h must be set as follows.
  - Bits [9:8] (GP20MD[1:0]) must be set to 11’b to map pin GPIO20 to #UBE.
  - Bits [5:4] (GP18MD[1:0]) must be set to 01’b to map pin GPIO18 to IORDY.

5 Software

Test utilities and Windows® CE v2.11.2.12 display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE v2.11/2.12 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE v2.11/2.12 display drivers are available from your sales support contact or on the internet at http://www.eea.epson.com.S1D13706

6 References

6.1 Documents

- NEC Electronics Inc., *NEC VR4181A Target Specification*, Revision 0.5, 9/11/98
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 Evaluation Board User Manual*, document number X31B-G-004-xx.
- Epson Research and Development, Inc., *S1D13706 Programming Notes and Examples*, document number X31B-G-003-xx.

6.2 Document Sources

- NEC Electronics Inc.website: http://www.necel.com.
- Epson Electronics America website: http://www.eea.epson.com.

7 Technical Support

7.1 Epson LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

7.2 NEC Electronics Inc.

NEC Electronics Inc. (U.S.A.)
Corporate Headquarters
2880 Scott Blvd.
Santa Clara, CA 95050-8062, USA
Tel: (800) 366-9782
Fax: (800) 729-9288
http://www.necel.com

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Interfacing to the Motorola MPC821 Microprocessor

Document Number: X31B-G-009-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13706 Embedded Memory LCD Controller and the Motorola MPC821 microprocessor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at techpubs@erd.epson.com.

2 Interfacing to the MPC821

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.

2.2 MPC8XX 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.

Support is provided for both on-chip (DMA controllers) and off-chip (other processors and peripheral controllers) bus masters. For further information on this topic, refer to Section 6, “References” on page 22.

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.

2.2.1 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 2-1: “Power PC Memory Read Cycle” illustrates a typical memory read cycle on the Power PC system bus.

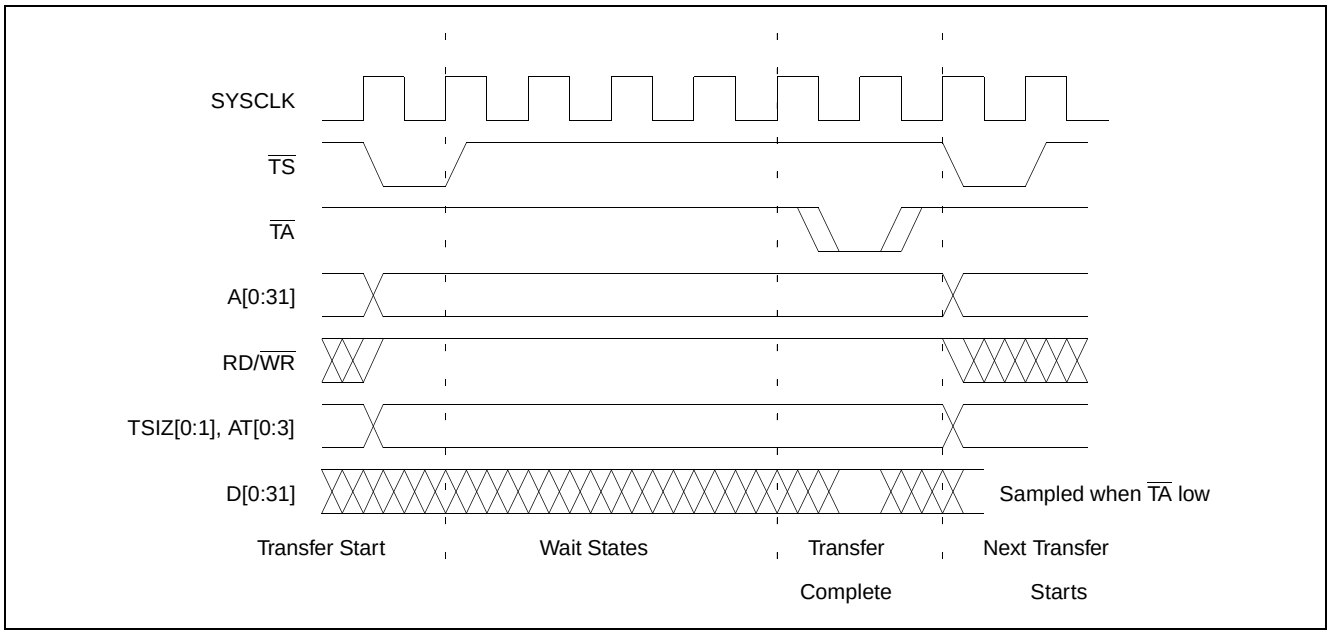


Figure 2-1: Power PC Memory Read Cycle

Figure 2-2: “Power PC Memory Write Cycle” illustrates a typical memory write cycle on the Power PC system bus.

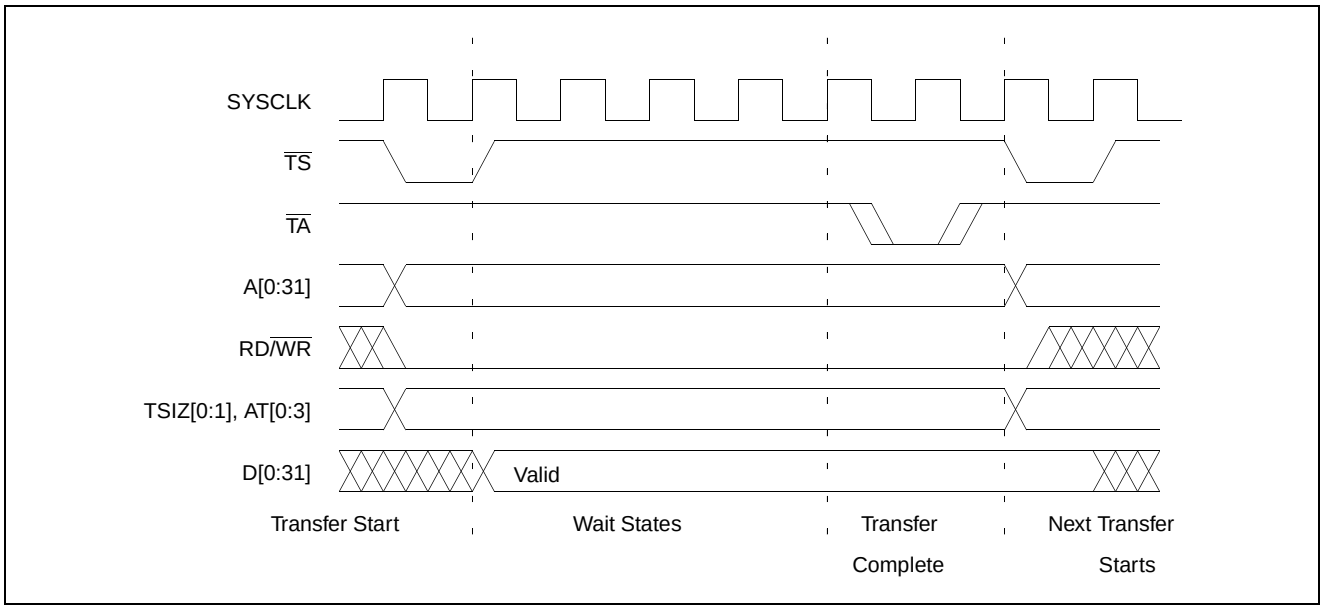


Figure 2-2: Power PC Memory Write Cycle

If an error occurs, TEA (Transfer Error Acknowledge) is asserted and the bus cycle is aborted. For example, a peripheral device may assert TEA if a parity error is detected, or the MPC821 bus controller may assert 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 D0 through D15 are used and address line A31 is ignored. For 8-bit transfers, data lines D0 through D7 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).

2.2.2 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 reverts to normal bus cycles for the remaining data transfers.

Burst cycles are mainly intended to facilitate cache line fills from program or data memory. They are normally not used for transfers to/from IO peripheral devices such as the S1D13706, therefore the interfaces described in this document do not attempt to support burst cycles.

2.3 Memory Controller Module

2.3.1 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 with respect to the address bus valid, by 0, ¼, or ½ clock cycle.
- The CSNT bit causes chip select and $\overline{WE}$ to be negated ½ clock cycle earlier than normal.
- The TRLX (relaxed timing) bit inserts 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 inserts 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.

Figure 2-3: “GPCM Memory Devices Timing” illustrates a typical cycle for a memory mapped device using the GPCM of the Power PC.

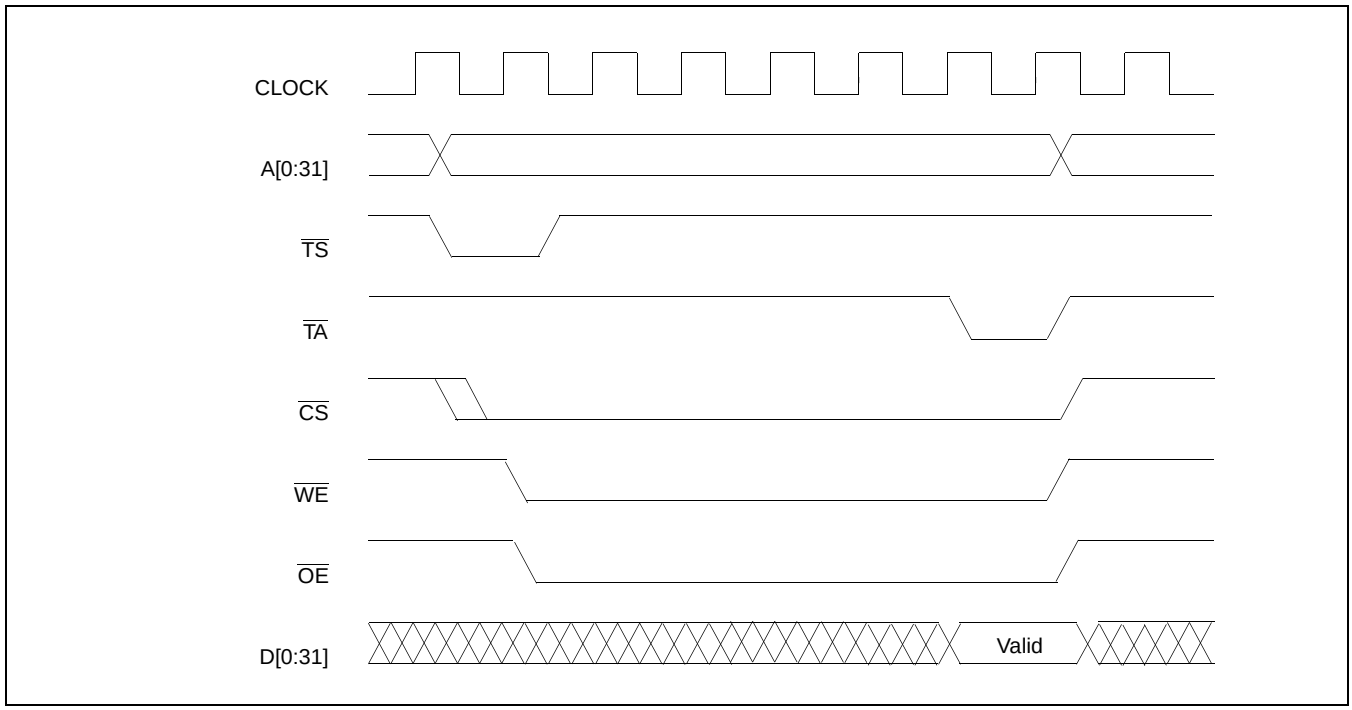


Figure 2-3: GPCM Memory Devices Timing

2.3.2 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 S1D13706 and it is desirable to leave the UPM free to handle other interfacing duties, such as EDO DRAM.

3 S1D13706 Host Bus Interface

The S1D13706 directly supports multiple processors. The S1D13706 implements a 16-bit Generic #1 Host Bus Interface which is most suitable for direct connection to the Motorola MPC821 microprocessor. Generic #1 supports a Chip Select and an individual Read Enable/Write Enable for each byte.

The Generic #1 Host Bus Interface is selected by the S1D13706 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13706 configuration, see Section 4.3, "S1D13706 Hardware Configuration" on page 18.

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

S1D13706 Pin Names	Motorola MPC821
AB[16:0]	A[15:31]
DB[15:0]	D[0:15]
WE1#	$\overline{WE0}$
CS#	$\overline{CS4}$
M/R#	A14
CLKI	SYSCLK
BS#	Connect to HIO V_{DD}
RD/WR#	$\overline{OE}$ (see note)
RD#	$\overline{OE}$ (see note)
WE0#	$\overline{WE1}$
WAIT#	$\overline{TA}$
RESET#	System $\overline{RESET}$

Note

The Motorola MPC821 chip select module only handles 16-bit read cycles. As the S1D13706 uses the chip select module to generate CS#, only 16-bit read cycles are possible and both the high and low byte enables can be driven by the MPC821 signal $\overline{OE}$.

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals.

- CLKI is a clock input which is required by the S1D13706 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, SYSCLK from the Motorola MPC821 is used for CLKI.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the MPC821 address (A[15:31]) and data bus (D[0:15]), respectively. CNF4 must be set to select big endian mode.
- Chip Select (CS#) must be driven low by $\overline{CS4}$ whenever the S1D13706 is accessed by the Motorola MPC821.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A14 to be connected to the M/R# line.
- WE0# connects to $\overline{WE1}$ (the low byte enable signal from the MPC821) and must be driven low when the MPC821 is writing the low byte to the S1D13706.
- WE1# connects to $\overline{WE0}$ (the high byte enable signal from the MPC821) and must be driven low when the MPC821 is writing the high byte to the S1D13706.
- RD# and RD/WR# are read enables for the low-order and high-order bytes, respectively. Both signals are driven low by $\overline{OE}$ when the Motorola MPC821 is reading data from the S1D13706.
- WAIT# connects to $\overline{TA}$ and is a signal which is output from the S1D13706 which indicates the MPC821 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since MPC821 accesses to the S1D13706 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13706 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Status (BS#) signal is not used in this implementation of the MPC821 interface using the Generic #1 Host Bus Interface. This pin must be tied high (connected to HIO V_{DD}).

4 MPC821 to S1D13706 Interface

4.1 Hardware Description

The interface between the S1D13706 and the MPC821 requires no external glue logic. The polarity of the WAIT# signal must be selected as active high by connecting CNF5 to NIO V_{DD} (see Table 4-2:, "Summary of Power-On/Reset Configuration Options," on page 18).

BS# (bus start) is not used in this implementation and should be tied high (connected to HIO V_{DD}).

The following diagram shows a typical implementation of the MPC821 to S1D13706 interface.

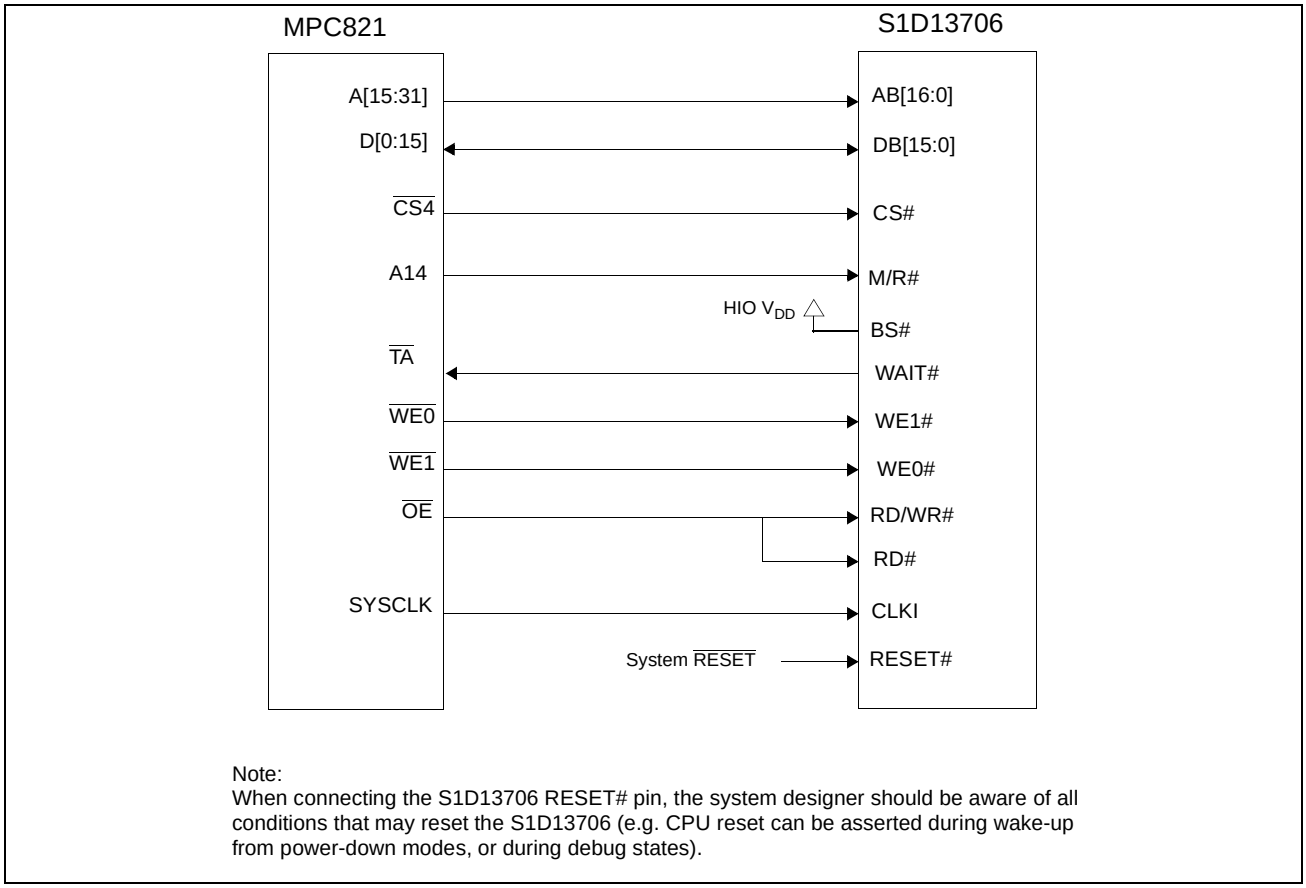


Figure 4-1: Typical Implementation of MPC821 to S1D13706 Interface

Table 4-1:, *"List of Connections from MPC821ADS to S1D13706"* on page 16 shows the connections between the pins and signals of the MPC821 and the S1D13706.

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 S1D13706. In a true 3 volt system, no buffering is necessary.

4.2 MPC821ADS Evaluation Board Hardware Connections

The following table details the connections between the pins and signals of the MPC821 and the S1D13706.

Table 4-1: List of Connections from MPC821ADS to S1D13706

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13706 Signal Name
Vcc	P6-A1, P6-B1	COREVDD, HIOVDD, NIOVDD
A15	P6-D20	A16
A16	P6-B24	A15
A17	P6-C24	A14
A18	P6-D23	A13
A19	P6-D22	A12
A20	P6-D19	A11
A21	P6-A19	A10
A22	P6-D28	A9
A23	P6-A28	A8
A24	P6-C27	A7
A25	P6-A26	A6
A26	P6-C26	A5
A27	P6-A25	A4
A28	P6-D26	A3
A29	P6-B25	A2
A30	P6-B19	A1
A31	P6-D17	A0
D0	P12-A9	D15
D1	P12-C9	D14
D2	P12-D9	D13
D3	P12-A8	D12
D4	P12-B8	D11
D5	P12-D8	D10
D6	P12-B7	D9
D7	P12-C7	D8
D8	P12-A15	D7
D9	P12-C15	D6
D10	P12-D15	D5

Table 4-1: List of Connections from MPC821ADS to S1D13706 (Continued)

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13706 Signal Name
D11	P12-A14	D4
D12	P12-B14	D3
D13	P12-D14	D2
D14	P12-B13	D1
D15	P12-C13	D0
SRESET	P9-D15	RESET#
SYSCLK	P9-C2	CLKI
CS4	P6-D13	CS#
TA	P6-B6 to inverter enabled by CS#	WAIT#
WE0	P6-B15	WE1#
WE1	P6-A14	WE0#
OE	P6-B16	RD/WR#, RD#
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

The bit numbering of the Motorola MPC821 bus signals is reversed from the normal convention, e.g.: the most significant address bit is A0, the next is A1, A2, etc.

4.3 S1D13706 Hardware Configuration

The S1D13706 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13706 to Motorola MPC821 microprocessor.

Table 4-2: Summary of Power-On/Reset Configuration Options

S1D13706 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[2:0]	011 = Generic #1 Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-3: “CLKI to BCLK Divide Selection” for recommended settings	

= configuration for MPC821 microprocessor

Table 4-3: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for MPC821 microprocessor

4.4 Register/Memory Mapping

The DRAM on the MPC821 ADS board extends from address 0 through 3F FFFFh, so the S1D13706 is addressed starting at 40 0000h. The S1D13706 uses two 128K byte blocks which are selected using A14 from the MPC821 (A14 is connected to the S1D13706 M/R# pin). The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block.

4.5 MPC821 Chip Select Configuration

Chip select 4 is used to control the S1D13706. The following options are selected in the base address register (BR4).

- BA (0:16) = 0000 0000 0100 0000 0 – set starting address of S1D13706 to 40 0000h
- AT (0:2) = 0 – ignore address type bits.
- PS (0:1) = 1:0 – memory port size is 16 bits
- 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; S1D13706 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 ½ clock cycle from address lines
- BI = 1 – 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 S1D13706 generates an external transfer acknowledge using the WAIT# line
- TRLX = 0 – normal timing
- EHTR = 0 – normal timing

4.6 Test Software

The test software to exercise this interface is very simple. It configures chip select 4 (CS4) on the MPC821 to map the S1D13706 to an unused 256K byte block of address space and loads the appropriate values into the option register for CS4. Then the software runs a tight loop reading the 13706 Revision Code Register REG[00h]. 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, its on-chip caches and MMU are disabled. If the data cache is enabled, then the MMU must be set up so that the S1D13706 memory block is tagged as non-cacheable, to ensure that accesses to the S1D13706 occurs in proper order, and also to ensure that the MPC821 does not attempt to cache any data read from or written to the S1D13706 or its display buffer.

The source code for this test routine is as follows:

```
BR4           equ         $120              ; CS4 base register
OR4           equ         $124              ; CS4 option register
MemStart      equ         $42 0000          ; address of S1D13706 display buffer
RevCodeReg    equ         $40 0000          ; 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,$0708       ; normal CS negation; delay CS ½ clock;
                                            ; inhibit burst
              stw         r2,OR4(r1)        ; write to option register
              andis.      r1,r0,0           ; clear r1
              oris        r1,r1,MemStart    ; point r1 to start of S1D13706 mem space
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 only.

5 Software

Test utilities and Windows® CE v2.11/2.12 display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE v2.11/2.12 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE v2.11/2.12 display drivers are available from your sales support contact or on the internet at http://www.eea.epson.com.

6 References

6.1 Documents

- Motorola Inc., *Power PC MPC821 Portable Systems Microprocessor User's Manual*, Motorola Publication no. MPC821UM/; available on the Internet at http://www.mot.com/SPS/ADC/pps/_subpgs/_documentation/821/821UM.html.
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, Document Number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 Evaluation Board User Manual*, Document Number X31B-G-004-xx.
- Epson Research and Development, Inc., *Programming Notes and Examples*, Document Number X31B-G-003-xx.

6.2 Document Sources

- Motorola Inc. Literature Distribution Center: (800) 441-2447.
- Motorola Inc. Website: http://www.mot.com.
- Epson Electronics America website: http://www.eea.epson.com.

7 Technical Support

7.1 EPSON LCD/CRT Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

7.2 Motorola MPC821 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Interfacing to the Motorola MCF5307 "ColdFire" Microprocessor

Document Number: X31B-G-010-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13706 Embedded Memory LCD Controller and the Motorola MCF5307 Processor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at techpubs@erd.epson.com.

2 Interfacing to the MCF5307

2.1 The MCF5307 System Bus

The MCF5200/5300 family of processors feature a high-speed synchronous system bus typical of modern microprocessors. This section is an overview of the operation of the CPU bus in order to establish interface requirements.

2.1.1 Overview

The MCF5307 microprocessor family uses a synchronous address and data bus, very similar in architecture to the MC68040 and MPC8xx. All outputs and inputs are timed with respect to a square-wave reference clock called BCLK0 (Master Clock). This clock runs at a software-selectable divisor rate from the machine cycle speed of the CPU core (typically 20 to 33 MHz). Both the address and the data bus are 32 bits in width. All IO accesses are memory-mapped; there is no separate IO space in the Coldfire architecture.

The bus can support two types of cycles, normal and burst. Burst memory cycles are used to fill on-chip cache memories, and for certain on-chip DMA operations. Normal cycles are used for all other data transfers.

2.1.2 Normal (Non-Burst) Bus Transactions

A data transfer is initiated by the bus master by placing the memory address on address lines A31 through A0 and driving $\overline{TS}$ (Transfer Start) low for one clock cycle. Several control signals are also provided with the memory address:

- SIZ[1:0] (Transfer Size) — indicates whether the bus cycle is 8, 16, or 32-bit.
- R/$\overline{W}$ — set high for read cycles and low for write cycles.
- TT[1:0] (Transfer Type Signals) — provides more detail on the type of transfer being attempted.
- $\overline{TIP}$ (Transfer In Progress) — asserts whenever a bus cycle is active.

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 MCF5307 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 2-1: “MCF5307 Memory Read Cycle,” illustrates a typical memory read cycle on the MCF5307 system bus.

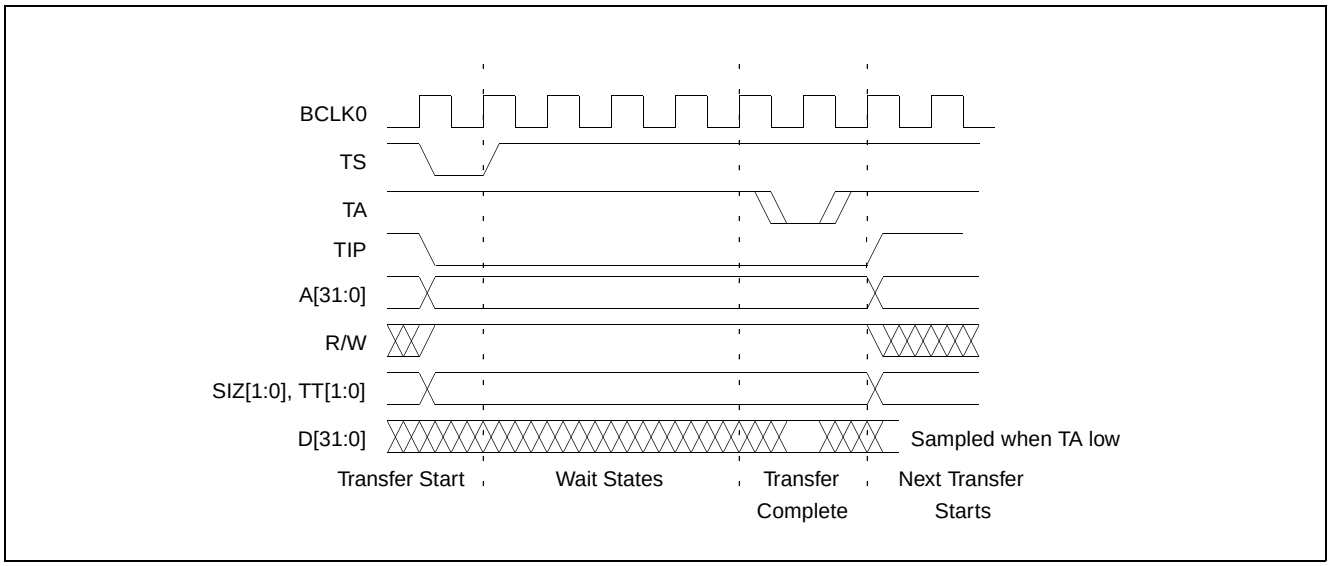


Figure 2-1: MCF5307 Memory Read Cycle

Figure 2-2: “MCF5307 Memory Write Cycle,” illustrates a typical memory write cycle on the MCF5307 system bus.

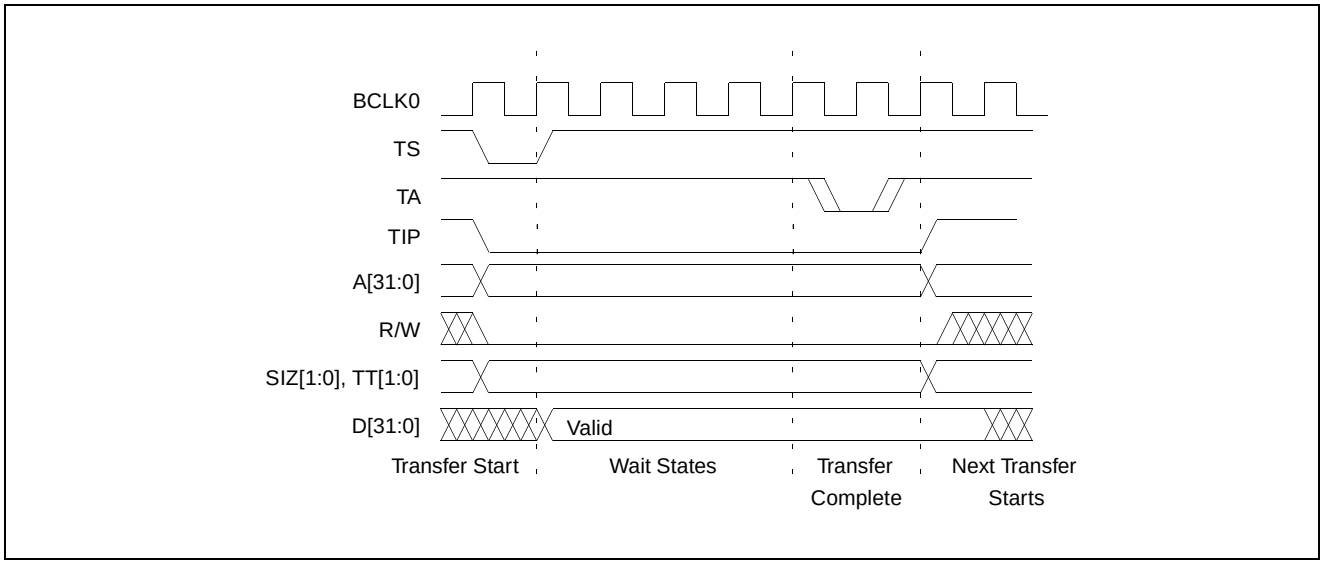


Figure 2-2: MCF5307 Memory Write Cycle

2.1.3 Burst Cycles

Burst cycles are very similar to normal cycles, except that they occur as a series of four back-to-back, 32-bit memory reads or writes. The $\overline{\text{TIP}}$ (Transfer In Progress) output is asserted continuously through the burst. Burst memory cycles are mainly intended to fill

caches from program or data memory. They are typically not used for transfers to or from IO peripheral devices such as the S1D13706. The MCF5307 chip selects provide a mechanism to disable burst accesses for peripheral devices which are not burst capable.

2.2 Chip-Select Module

In addition to generating eight independent chip-select outputs, the MCF5307 Chip Select Module can generate active-low Output Enable ($\overline{OE}$) and Write Enable ($\overline{BWE}$) signals compatible with most memory and x86-style peripherals. The MCF5307 bus controller also provides a Read/Write (R/$\overline{W}$) signal which is compatible with most 68K peripherals.

Chip selects 0 and 1 can be programmed independently to respond to any base address and block size. Chip select 0 can be active immediately after reset, and is typically used to control a boot ROM. Chip select 1 is likewise typically used to control a large static or dynamic RAM block.

Chip selects 2 through 7 have fixed block sizes of 2M bytes each. Each has a unique, fixed offset from a common, programmable starting address. These chip selects are well-suited to typical IO addressing requirements.

Each chip select may be individually programmed for:

- port size (8/16/32-bit).
- up to 15 wait states or external acknowledge.
- address space type.
- burst or non-burst cycle support.
- write protect.

Figure 2-3: “Chip Select Module Outputs Timing” illustrates a typical cycle for a memory mapped device using the GPCM of the Power PC.

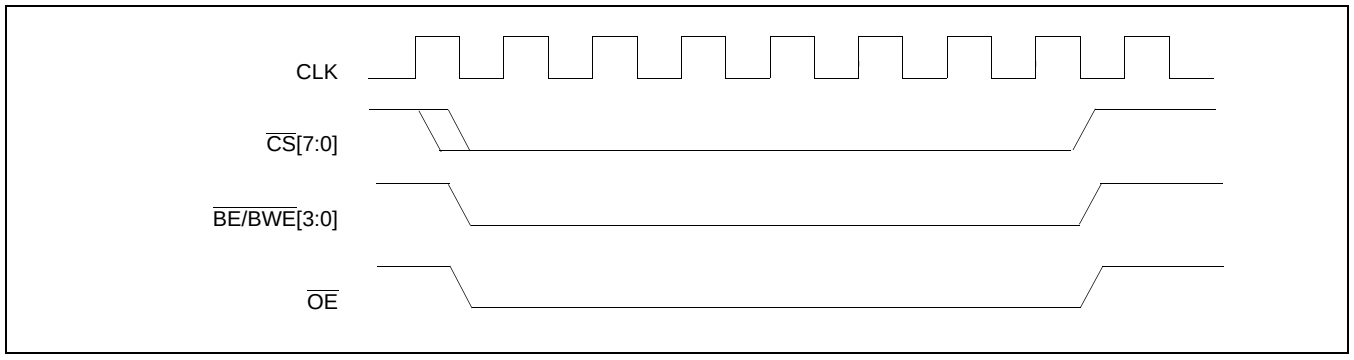


Figure 2-3: Chip Select Module Outputs Timing

3 S1D13706 Host Bus Interface

The S1D13706 directly supports multiple processors. The S1D13706 implements a 16-bit Generic #1 Host Bus Interface which is most suitable for direct connection to the Motorola MFC5307 microprocessor. Generic #1 supports a Chip Select and an individual Read Enable/Write Enable for each byte.

The Generic #1 Host Bus Interface is selected by the S1D13706 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13706 configuration, see Section 4.2, “S1D13706 Hardware Configuration” on page 14.

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

S1D13706 Pin Names	Motorola MCF5307
AB[16:0]	A[16:0]
DB[15:0]	D[31:16]
WE1#	$\overline{\text{BWE1}}$
CS#	$\overline{\text{CS4}}$
M/R#	A17
CLKI	BCLK0
BS#	Connect to HIO V_{DD}
RD/WR#	$\overline{\text{OE}}$
RD#	$\overline{\text{OE}}$
WE0#	$\overline{\text{BWE0}}$
WAIT#	$\overline{\text{TA}}$
RESET#	system $\overline{\text{RESET}}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals.

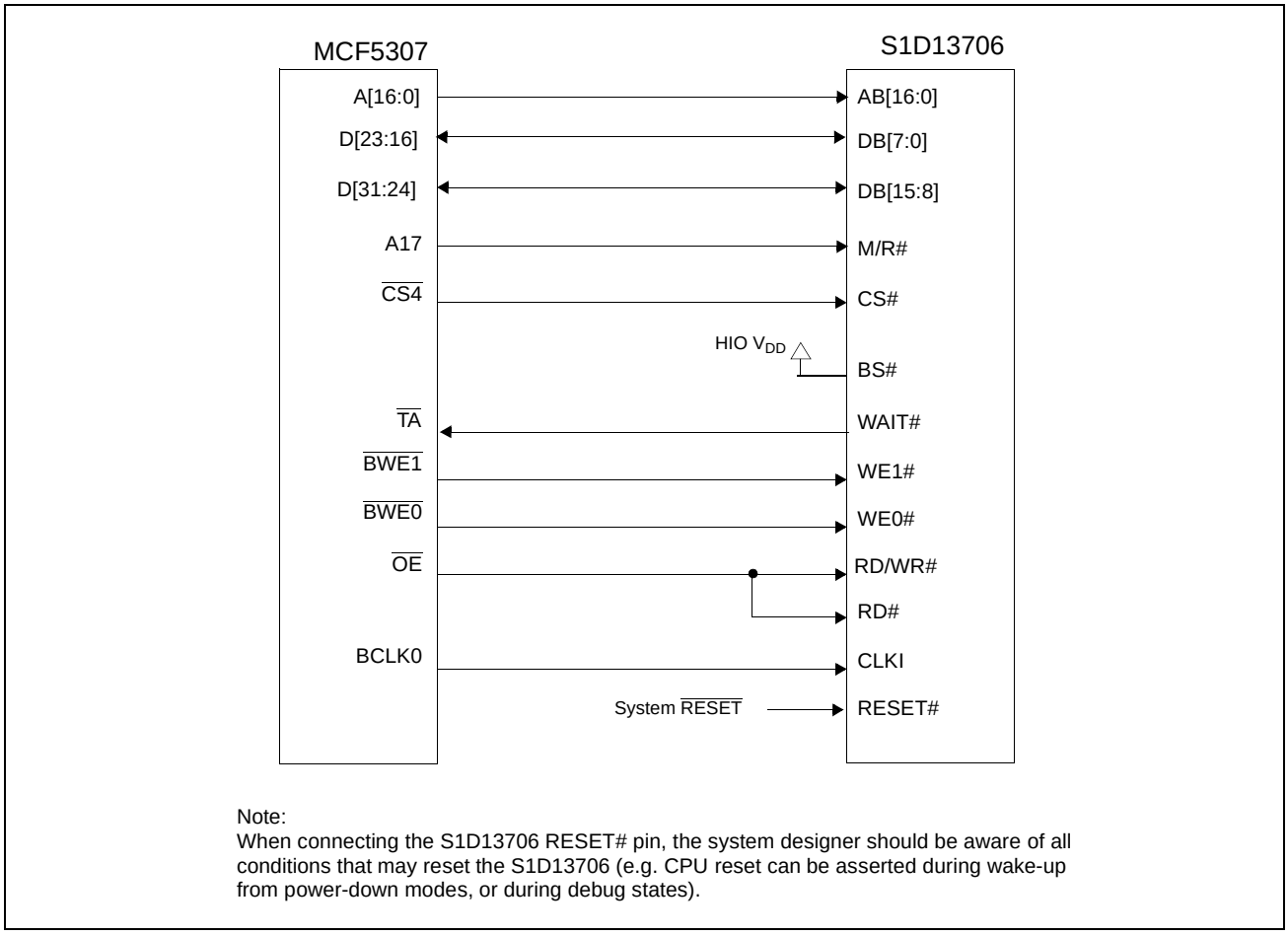
- CLKI is a clock input which is required by the S1D13706 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, BCLK0 from the Motorola MCF5307 is used for CLKI.
- The address inputs AB[16:0] connect directly to the MCF5307 address bus (A[16:0]).
- DB[7:0] connects D[23:16] (the MCF5307 low order byte). DB[15:8] connects to D[31:24] (the MCF5307 high order byte). CNF4 must be set to select big endian mode.
- Chip Select (CS#) must be driven low by $\overline{CS4}$ whenever the S1D13706 is accessed by the Motorola MCF5307.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A17 to be connected to the M/R# line.
- WE0# connects to $\overline{BWE0}$ (the low byte enable signal from the MCF5307) and must be driven low when the MCF5307 is writing the low byte to the S1D13706.
- WE1# connects to $\overline{BWE1}$ (the high byte enable signal from the MCF5307) and must be driven low when the MCF5307 is writing the high byte to the S1D13706.
- RD# and RD/WR# are read enables for the low-order and high-order bytes, respectively. Both signals are driven low by $\overline{OE}$ when the Motorola MCF5307 is reading data from the S1D13706.
- WAIT# connects to $\overline{TA}$ and is a signal which is output from the S1D13706 that indicates the host CPU must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since host CPU accesses to the S1D13706 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13706 internal registers and/or refresh memory. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete. This signal is active low and may need to be inverted if the host CPU wait state signal is active high.
- The Bus Status (BS#) signal is not used in the bus interface for Generic #1 mode and must be tied high to HIO V_{DD}.

4 MCF5307 To S1D13706 Interface

4.1 Hardware Description

The interface between the S1D13706 and the MCF5307 requires no external glue logic. The polarity of the WAIT# signal must be selected as active high by connecting CNF5 to NIO V_{DD} (see Table 4-1:, "Summary of Power-On/Reset Configuration Options," on page 14).

The following diagram shows a typical implementation of the MCF5307 to S1D13706 interface.



Note:
When connecting the S1D13706 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13706 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 4-1: Typical Implementation of MCF5307 to S1D13706 Interface

4.2 S1D13706 Hardware Configuration

The S1D13706 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13706 to Motorola MFC5307 microprocessor.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13706 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	**0**
CNF[2:0]	011 = Generic #1 Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	See Table 4-2: “CLKI to BCLK Divide Selection” for recommended setting	

= configuration for MFC5307 host bus interface

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for MFC5307 host bus interface

4.3 Register/Memory Mapping

The S1D13706 uses two 128K byte blocks which are selected using A17 from the MCF5307 (A17 is connected to the S1D13706 M/R# pin). The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block. These two blocks of memory are aliased over the entire 2M byte space.

Note
If aliasing is not desirable, the upper addresses must be fully decoded.

4.4 MCF5307 Chip Select Configuration

Chip Selects 0 and 1 have programmable block sizes from 64K bytes through 2G bytes. However, these chip selects would normally be needed to control system RAM and ROM. Therefore, one of the IO chip selects CS2 through CS7 is required to address the entire address space of the S1D13706. These IO chip selects have a fixed, 2M byte block size. In the example interface, chip select 4 is used to control the S1D13706. The CSBAR register should be set to the upper 8 bits of the desired base address.

The following options should be selected in the chip select mask registers (CSMR4/5).

- WP = 0 – disable write protect
- AM = 0 - enable alternate bus master access to the S1D13706
- C/I = 1 - disable CPU space access to the S1D13706
- SC = 1 - disable Supervisor Code space access to the S1D13706
- SD = 0 - enable Supervisor Data space access to the S1D13706
- UC = 1 - disable User Code space access to the S1D13706
- UD = 0 - enable User Data space access to the S1D13706
- V = 1 - global enable (“Valid”) for the chip select

The following options should be selected in the chip select control registers (CSCR4/5).

- WS0-3 = 0 - no internal wait state setting
- AA = 0 - no automatic acknowledgment
- PS (1:0) = 1:0 – memory port size is 16 bits
- BEM = 0 – Byte enable/write enable active on writes only
- BSTR = 0 – disable burst reads
- BSTW = 0 – disable burst writes

5 Software

Test utilities and Windows® CE v2.11/2.12 display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE v2.11/2.12 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE v2.11/2.12 display drivers are available from your sales support contact or on the internet at http://www.eea.epson.com.

6 References

6.1 Documents

- Motorola Inc., *MCF5307 ColdFire® Integrated Microprocessor User's Manual*, Motorola Publication no. MCF5307UM; available on the Internet at http://www.mot.com/SPS/HPESD/prod/coldfire/5307UM.html.
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 Evaluation Board User Manual*, document number X31B-G-004-xx.
- Epson Research and Development, Inc., *S1D13706 Programming Notes and Examples*, document number X31B-G-003-xx.

6.2 Document Sources

- Motorola Inc.: Motorola Literature Distribution Center, (800) 441-2447.
- Motorola website: http://www.mot.com.
- Epson Electronics America website: http://www.eea.epson.com

7 Technical Support

7.1 EPSON LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

7.2 Motorola MCF5307 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Connecting to the Sharp HR-TFT Panels

Document Number: X31B-G-011-04

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to connect to the Sharp HR-TFT panels directly supported by the S1D13706. These panels are:

- Sharp LQ031B1DDXX 160 x 160 HR-TFT panel.
- Sharp LQ039Q2DS01 320 x 240 HR-TFT panel.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Connecting to the Sharp LQ039Q2DS01 HR-TFT

2.1 External Power Supplies

The S1D13706 provides all necessary data and control signals to connect to the Sharp LQ039Q2DS01 320 x 240 HR-TFT panel. However, it does not provide any of the voltages required for gray scaling, gate driving, or for the digital and analog supplies. Therefore, external supplies must be designed for any device utilizing the LQ039Q2DS01.

2.1.1 Gray Scale Voltages for Gamma Correction

The standard gray scale voltages can be generated using a precise resistor divider network that supplies two sets (A and B) of nine reference voltages to a National Semiconductor 9-Channel Buffer Amplifier (LMC6009). The LMC6009 buffers these nine reference voltages and outputs them to the panel column drivers. The A/B inputs allow the two sets of reference voltages to be alternated, compensating for asymmetrical gamma characteristics during row inversion. This input is controlled by the S1D13706 output signal REV which toggles every time a horizontal sync signal is sent to the panel.

The REV signal is also used to generate the highest gray scale voltage (V0 or black) by buffering REV and shifting its maximum level to the maximum gray scale voltage (CON_POWER). CON_POWER is supplied by a National Semiconductor micropower Voltage Regulator (LP2951). Figure 2-1: “Sharp LQ039Q2DS01 Gray Scale Voltage (V0-V9) Generation” shows the schematic for gray scale voltage generation.

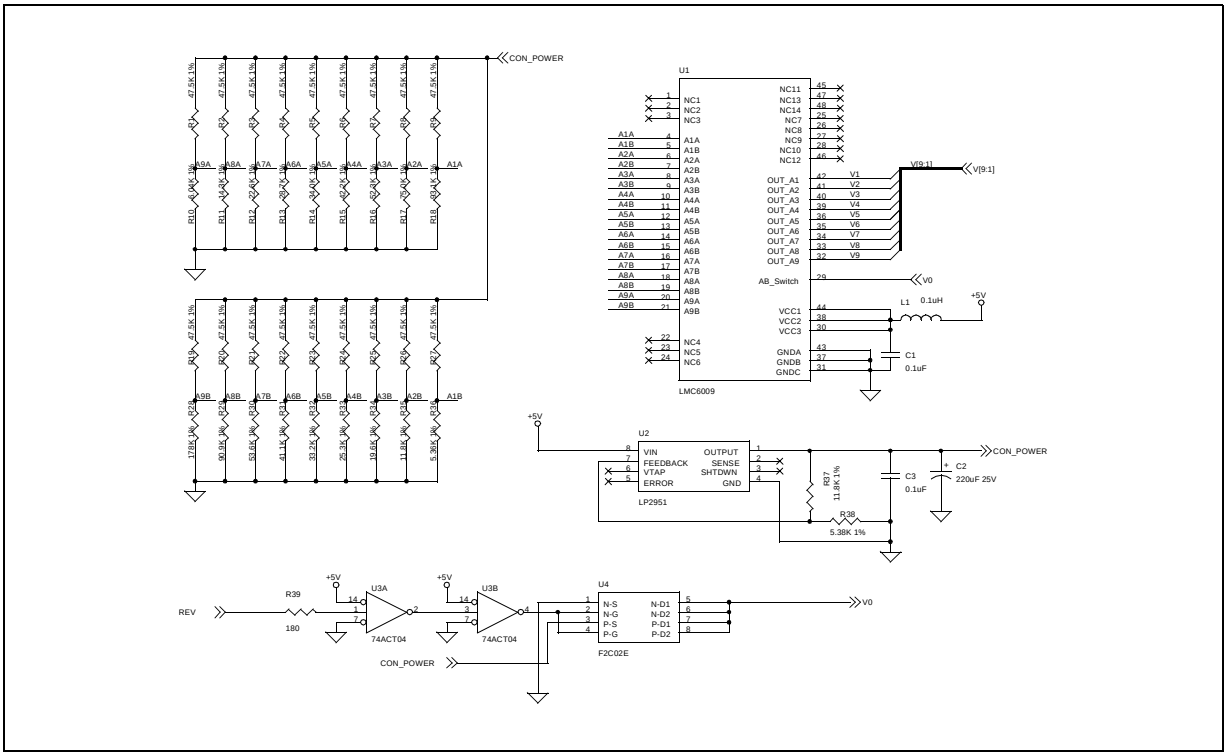


Figure 2-1: Sharp LQ039Q2DS01 Gray Scale Voltage (V0-V9) Generation

2.1.2 Digital/Analog Power Supplies

The digital power supply (VSHD) must be connected to a 3.3V supply. The analog power supply (VSHA) must be connected to a 5.0V supply.

2.1.3 DC Gate Driver Power Supplies

The gate driver high level power supply (V_{DD}) and the gate driver logic low power supply (V_{SS}) have typical values of +15V and -15V respectively. These power supplies can be provided by a Linear Technology high efficiency switching regulator (LT1172). The two power supplies can be adjusted through their allowable ranges using the potentiometer VR1.

The gate driver logic high power supply (V_{CC}) is defined as V_{SS} + VSHD. The typical V_{CC} voltage of -11.7V can be supplied from V_{SS} using a 3.3V zener diode which provides the necessary voltage change.

Figure 2-2: "Panel Gate Driver DC Power Supplies" shows the schematic for V_{SS}, V_{DD} and V_{CC}.

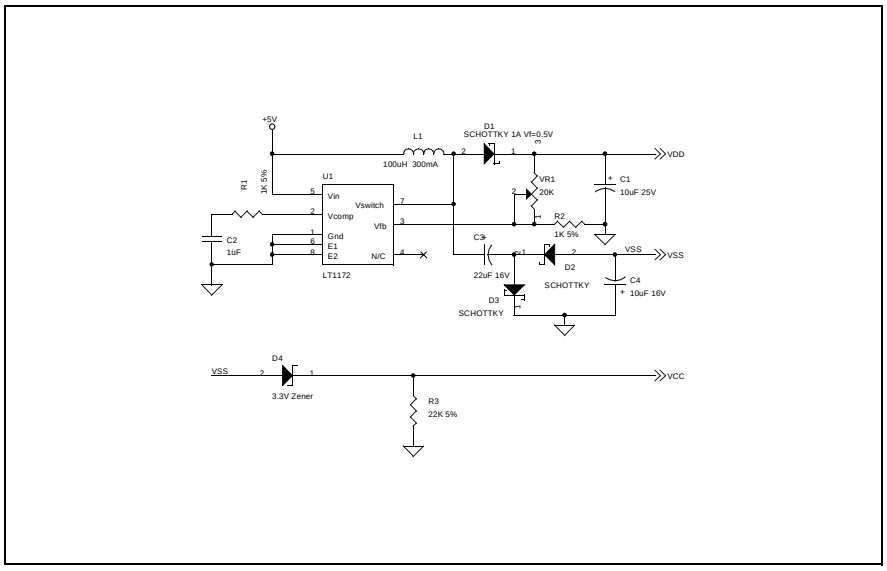


Figure 2-2: Panel Gate Driver DC Power Supplies

2.1.4 AC Gate Driver Power Supplies

The gate drive low level power supply (V_{EE}) is an AC power supply with a DC offset voltage (offset typically -9.0V). The AC component is the common electrode driving signal (V_{COM}) which has a voltage of ±2.5V. V_{COM} must be alternated every horizontal period and every vertical period. The S1D13706 output signal REV accomplishes this function and generates the alternating V_{COM} signal which is superimposed onto V_{EE}. Figure 2-3: "Panel Gate Driver AC Power Supplies," on page 10 shows the schematic for generating V_{COM} and V_{EE}.

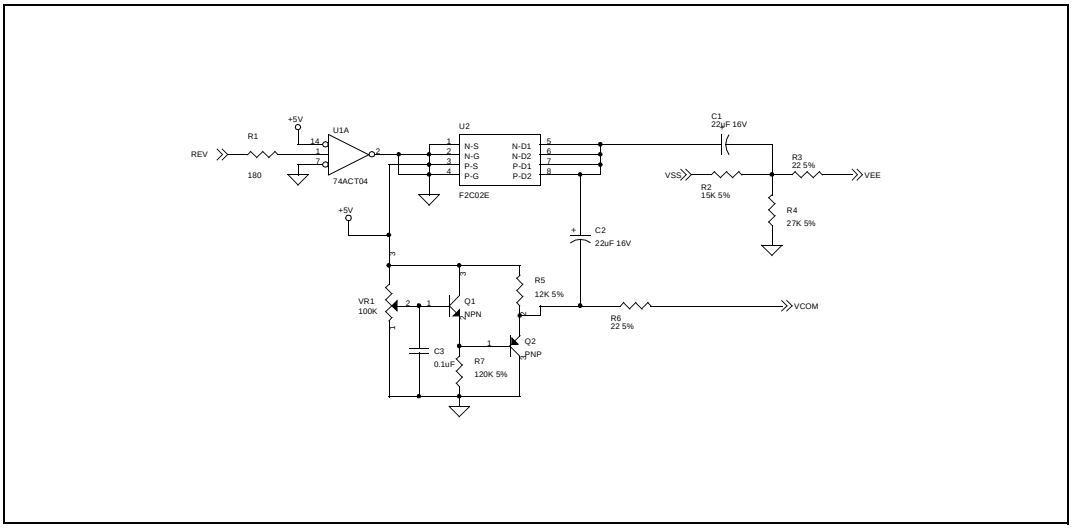
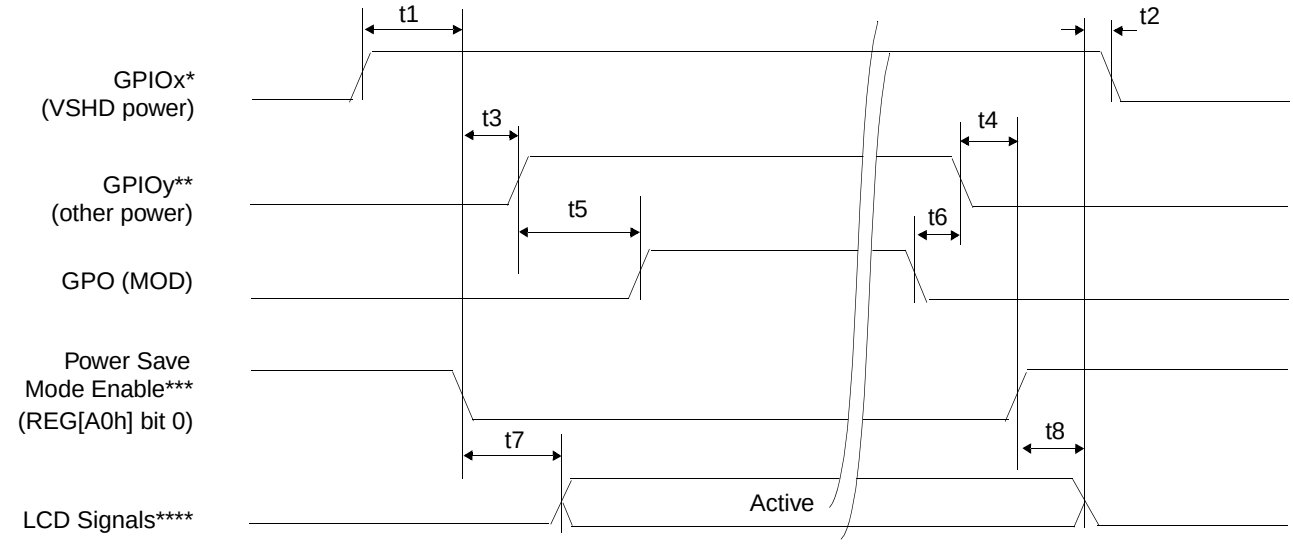


Figure 2-3: Panel Gate Driver AC Power Supplies

2.2 HR-TFT MOD Signal

The HR-TFT panel uses an input signal (MOD) to control the power-on sequencing of the panel. This HR-TFT signal should not be confused with the S1D13706 signal DRDY (referred to as MOD for passive panels).

To power-on the HR-TFT panel, MOD must be held low until the power supply has been turned on for more than two FRAMES. To power-off the HR-TFT panel, MOD must be forced low before the power supply is turned off. This sequencing requires two software controlled GPIO pins from the S1D13706 (see Figure 2-4: "HR-TFT Power-On/Off Sequence Timing").



*It is recommended to use one of the general purpose IO pins GPIO[6:4] to control the digital power supply VSHD.
** It is recommended to use one of the general purpose IO pins GPIO[6:4] to control the other power supplies required by the HR-TFT panel.
***The S1D13706 LCD power-on/off sequence is activated by programming the Power Save Mode Enable bit (REG[A0h] bit 0
****LCD Signals include: FPDAT[17:0], FPSHIFT, FPLINE, FPFRAME, and GPIO[3:0].

Figure 2-4: HR-TFT Power-On/Off Sequence Timing

Table 2-1: HR-TFT Power-On/Off Sequence Timing

Symbol	Parameter	Min	Max	Units
t1	LCD Power (VSHD) active to Power Save Mode disabled	0		ns
t2	LCD signals low to LCD Power (VSHD) inactive	0		ns
t3	Power Save Mode disabled to LCD Power (other) active	0		ns
t4	LCD Power (other) inactive to Power Save Mode enabled	0		ns
t5	LCD Power (other) active to MOD active	2		FRAME
t6	MOD inactive to LCD Power (other) inactive	0		ns
t7	Power Save Mode disabled to LCD signals active		20	ns
t8	Power Save Mode enabled to LCD signals low		20	ns

2.3 S1D13706 to LQ039Q2DS01 Pin Mapping

Table 2-2: S1D13706 to LQ039Q2DS01 Pin Mapping

LCD Pin No.	LCD Pin Name	S1D13706 Pin Name	Description	Remarks
1	VDD	-	Power supply of gate driver (high level)	See Section 2.1, “External Power Supplies” on page 8
2	VCC	-	Power supply of gate driver (logic high)	See Section 2.1, “External Power Supplies” on page 8
3	MOD	-	Control signal of gate driver	See Section 2.2, “HR-TFT MOD Signal” on page 11
4	MOD	-	Control signal of gate driver	See Section 2.2, “HR-TFT MOD Signal” on page 11
5	U/L	-	Selection for vertical scanning direction	Connect to VSHD (top / bottom scanning)
6	SPS	FPFRAME	Start signal of gate driver	
7	CLS	GPIO1	Clock signal of gate driver	
8	VSS	-	Power supply of gate driver (logic low)	See Section 2.1, “External Power Supplies” on page 8
9	VEE	-	Power supply of gate driver (low level)	See Section 2.1, “External Power Supplies” on page 8
10	VEE	-	Power supply of gate driver (low level)	See Section 2.1, “External Power Supplies” on page 8
11	VCOM	-	Common electrode driving signal	See Section 2.1, “External Power Supplies” on page 8
12	VCOM	-	Common electrode driving signal	See Section 2.1, “External Power Supplies” on page 8
13	SPL	GPIO3	Sampling start signal for left / right scanning	
14	R0	FPDAT11	Red data signal (LSB)	
15	R1	FPDAT10	Red data signal	
16	R2	FPDAT9	Red data signal	
17	R3	FPDAT2	Red data signal	
18	R4	FPDAT1	Red data signal	
19	R5	FPDAT0	Red data signal (MSB)	
20	G0	FPDAT14	Green data signal (LSB)	
21	G1	FPDAT13	Green data signal	
22	G2	FPDAT12	Green data signal	
23	G3	FPDAT5	Green data signal	
24	G4	FPDAT4	Green data signal	
25	G5	FPDAT3	Green data signal (MSB)	

Table 2-2: S1D13706 to LQ039Q2DS01 Pin Mapping (Continued)

LCD Pin No.	LCD Pin Name	S1D13706 Pin Name	Description	Remarks
26	B0	FPDAT17	Blue data signal (LSB)	
27	B1	FPDAT16	Blue data signal	
28	B2	FPDAT15	Blue data signal	
29	B3	FPDAT8	Blue data signal	
30	B4	FPDAT7	Blue data signal	
31	B5	FPDAT6	Blue data signal (MSB)	
32	VSHD	-	Digital power supply	See Section 2.1, “External Power Supplies” on page 8
33	DGND	V_{SS}	Digital ground	Ground pin of S1D13706
34	PS	GPIO0	Power save signal	
35	LP	FPLINE	Data latch signal of source driver	
36	DCLK	FPSHIFT	Data sampling clock signal	
37	LBR	-	Selection for horizontal scanning direction	Connect to VSHD (left / right scanning)
38	SPR	-	Sampling start signal for right / left scanning	Right to left scanning not supported
39	VSHA	-	Analog power supply	See Section 2.1, “External Power Supplies” on page 8
40	V0	-	Standard gray scale voltage (black)	See Section 2.1, “External Power Supplies” on page 8
41	V1	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
42	V2	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
43	V3	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
44	V4	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
45	V5	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
46	V6	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
47	V7	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
48	V8	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
49	V9	-	Standard gray scale voltage (white)	See Section 2.1, “External Power Supplies” on page 8
50	AGND	V_{SS}	Analog ground	Ground pin of S1D13706

3 Connecting to the Sharp LQ031B1DDxx HR-TFT

3.1 External Power Supplies

The S1D13706 provides all necessary data and control signals to connect to the Sharp LQ031B1DDxx 160x160 HR-TFT panel(s). However, it does not provide any of the voltages required for the backlight, gray scaling, gate driving, or for the digital and analog supplies. Therefore, external supplies must be designed for any device utilizing the LQ031B1DDxx.

The LQ031B1DDxx (160x160) has the same voltage requirements as the LQ039Q2DS01 (320x240). All the circuits used to generate the various voltages for the LQ039Q2DS01 panel also apply to the LQ031B1DDxx panel. This section provides additional circuits for generating some of these voltages.

3.1.1 Gray Scale Voltages for Gamma Correction

The standard gray scale voltages can be generated using a precise resistor divider network as described in Section 2.1.1, “Gray Scale Voltages for Gamma Correction” on page 8. Alternately, they can be generated using a Sharp gray scale IC. The Sharp IR3E203 eliminates the large resistor network used to provide the 10 gray scale voltages and combines their function into a single IC.

The S1D13706 output signal REV is used to alternate the gray scale voltages and connects to the SW input of the IR3E203 IC. The COM signal is used in generating the gate driver panel AC voltage, V_{COM} and is explained in Section 3.1.4, “AC Gate Driver Power Supplies” on page 15. Figure 3-1: “Sharp LQ031B1DDxx Gray Scale Voltage (V0-V9) Generation” shows the circuit that generates the gray scale voltages using the Sharp IR3E203 IC.

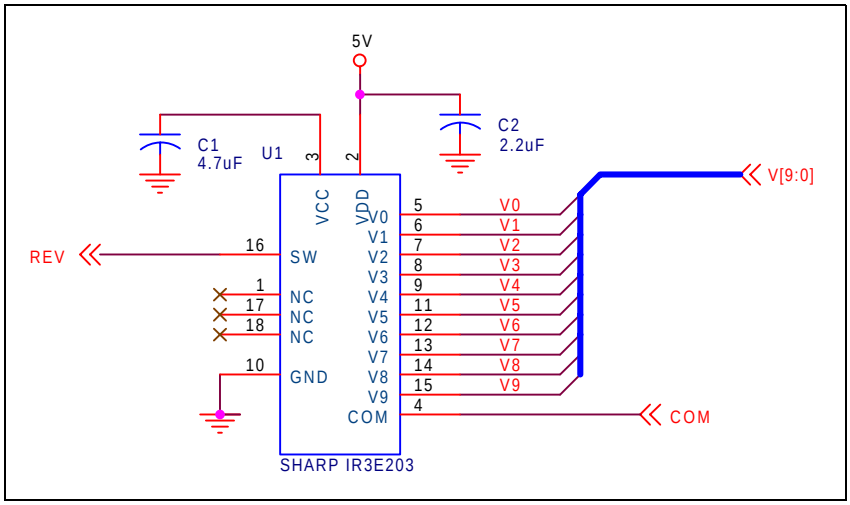


Figure 3-1: Sharp LQ031B1DDxx Gray Scale Voltage (V0-V9) Generation

3.1.2 Digital/Analog Power Supplies

The digital power supply (VSHD) must be connected to a 3.3V supply. The analog power supply (VSHA) must be connected to a 5.0V supply.

3.1.3 DC Gate Driver Power Supplies

See Section 2.1.3, “DC Gate Driver Power Supplies” on page 9 and Figure 2-2: “Panel Gate Driver DC Power Supplies,” on page 9 for details on generating V_{SS}, V_{DD}, and V_{CC}.

3.1.4 AC Gate Driver Power Supplies

See Section 2.1.4, “AC Gate Driver Power Supplies” on page 10 and Figure 2-3: “Panel Gate Driver AC Power Supplies,” on page 10 for details on generating V_{EE} and V_{COM}. If the Sharp IR3E203 is used to generate the gray scale voltages, the COM signal can be connected to the input of the F2C02E MOSFET instead of the buffered REV signal.

3.2 HR-TFT MOD Signal

See Section 2.2, “HR-TFT MOD Signal” on page 11 for details on controlling the MOD signal through software.

3.3 S1D13706 to LQ031B1DDxx Pin Mapping

Table 3-1: S1D13706 to LQ031B1DDxx Pin Mapping

LCD Pin No.	LCD Pin Name	S1D13706 Pin Name	Description	Remarks
1	VDD	-	Power supply of gate driver (high level)	See Section 3.1, “External Power Supplies” on page 14
2	VCC	-	Power supply of gate driver (logic high)	See Section 3.1, “External Power Supplies” on page 14
3	MOD	-	Control signal of gate driver	See Section 3.2, “HR-TFT MOD Signal” on page 15
4	MOD	-	Control signal of gate driver	See Section 3.2, “HR-TFT MOD Signal” on page 15
5	U/L	-	Selection for vertical scanning direction	Connect to VSHD (top / bottom scanning)
6	SPS	FPFRAME	Start signal of gate driver	
7	CLS	GPIO1	Clock signal of gate driver	
8	VSS	-	Power supply of gate driver (logic low)	See Section 3.1, “External Power Supplies” on page 14
9	VEE	-	Power supply of gate driver (low level)	See Section 3.1, “External Power Supplies” on page 14
10	VEE	-	Power supply of gate driver (low level)	See Section 3.1, “External Power Supplies” on page 14
11	VCOM	-	Common electrode driving signal	See Section 3.1, “External Power Supplies” on page 14
12	VCOM	-	Common electrode driving signal	See Section 3.1, “External Power Supplies” on page 14
13	SPL	GPIO3	Sampling start signal for left / right scanning	
14	R0	FPDAT11	Red data signal (LSB)	
15	R1	FPDAT10	Red data signal	
16	R2	FPDAT9	Red data signal	
17	R3	FPDAT2	Red data signal	
18	R4	FPDAT1	Red data signal	
19	R5	FPDAT0	Red data signal (MSB)	
20	G0	FPDAT14	Green data signal (LSB)	
21	G1	FPDAT13	Green data signal	
22	G2	FPDAT12	Green data signal	
23	G3	FPDAT5	Green data signal	
24	G4	FPDAT4	Green data signal	
25	G5	FPDAT3	Green data signal (MSB)	

Table 3-1: S1D13706 to LQ031B1DDxx Pin Mapping (Continued)

LCD Pin No.	LCD Pin Name	S1D13706 Pin Name	Description	Remarks
26	B0	FPDAT17	Blue data signal (LSB)	
27	B1	FPDAT16	Blue data signal	
28	B2	FPDAT15	Blue data signal	
29	B3	FPDAT8	Blue data signal	
30	B4	FPDAT7	Blue data signal	
31	B5	FPDAT6	Blue data signal (MSB)	
32	VSHD	-	Digital power supply	See Section 3.1, “External Power Supplies” on page 14
33	DGND	V_{SS}	Digital ground	Ground pin of S1D13706
34	PS	GPIO0	Power save signal	
35	LP	FPLINE	Data latch signal of source driver	
36	DCLK	FPSHIFT	Data sampling clock signal	
37	LBR	-	Selection for horizontal scanning direction	Connect to VSHD (left / right scanning)
38	SPR	-	Sampling start signal for right / left scanning	Right to left scanning not supported
39	VSHA	-	Analog power supply	See Section 3.1, “External Power Supplies” on page 14
40	V0	-	Standard gray scale voltage (black)	See Section 3.1, “External Power Supplies” on page 14
41	V1	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
42	V2	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
43	V3	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
44	V4	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
45	V5	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
46	V6	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
47	V7	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
48	V8	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
49	V9	-	Standard gray scale voltage (white)	See Section 3.1, “External Power Supplies” on page 14
50	AGND	V_{SS}	Analog ground	Ground pin of S1D13706

4 Test Software

Test utilities and Windows CE display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and display drivers are available from your sales support contact or www.eea.epson.com.

5 References

5.1 Documents

- Sharp Electronics Corporation, *LQ039Q2DS01 Specification*.
- Sharp Electronics Corporation, *LQ031B1DDxx Specification*.
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, Document Number X31B-A-001-xx.
- Epson Research and Development, Inc., *S1D13706 Programming Notes and Examples*, Document Number X31B-G-003-xx.

5.2 Document Sources

- Sharp Electronics Corporation Website: http://www.sharpsma.com.
- Epson Electronics America Website: http://www.eea.epson.com.

6 Technical Support

6.1 EPSON LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

6.2 Sharp HR-TFT Panel

http://www.sharpsma.com

EPSON®

S1D13706 Embedded Memory LCD Controller

Connecting to the Epson D-TFD Panels

Document Number: X31B-G-012-03

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software required to connect the S1D13706 to two Epson D-TFD (Digital Thin Film Diode) panels, the 320 x 240 LF37SQT and the 160 x 240 LF26SCT.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 External Power Supplies

The S1D13706 provides all necessary data and control signals to connect to the Epson LF37SQT and LF26SCT D-TFD panels. However, it does not provide any of the vertical and horizontal logic voltages, contrast or brightness voltages, or the horizontal and vertical liquid crystal driving voltages. Therefore, external supplies must be designed for any device utilizing these D-TFD panels.

2.1 VDDH and VDD - Horizontal and Vertical Analog Voltages

VDDH and VDD control the horizontal and vertical drivers that activate the liquid crystals in the D-TFD display. The range of VDDH is from 4.5V to 5.5V and VDD is from 4.0V to 5.0V. These voltages should be set to 4.5V.

VDDH and VDD must be activated after all D-TFD control signals are active, and should be deactivated after the control signals are inactive.

Figure 2-1: "VDDH and VDD Voltage Generation" shows an example implementation which generates VDDH and VDD from 3.3V.

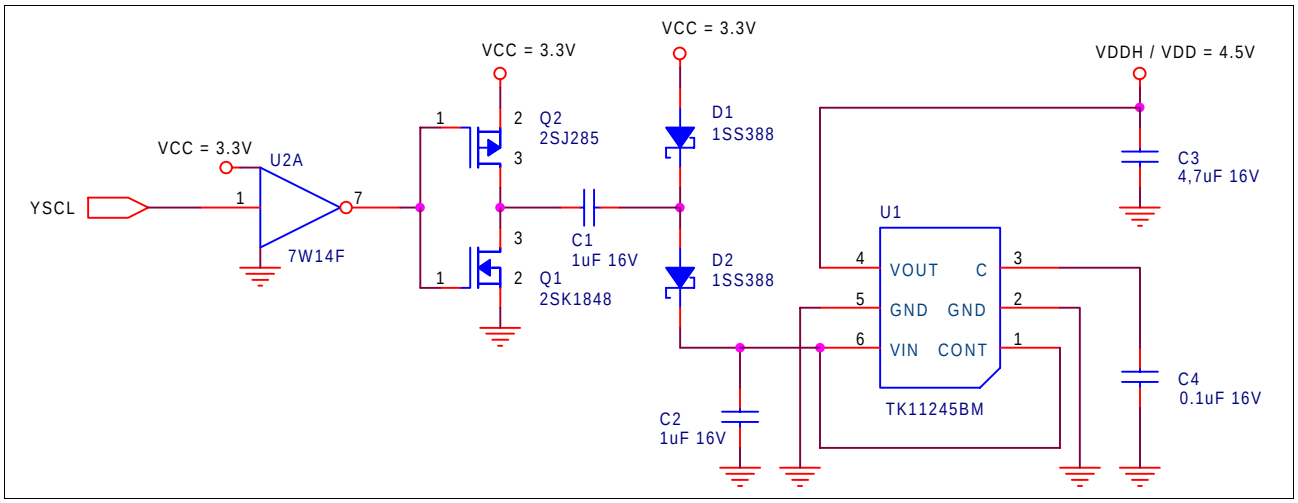


Figure 2-1: VDDH and VDD Voltage Generation

The circuit in Figure 2-1: "VDDH and VDD Voltage Generation" uses the Vertical Shift Clock control signal, YSCL, to control a pair of ultrahigh-speed P and N-channel MOSFET transistors. These transistors are used to generate a 3.3V square wave which is passed through C1. This blocks any DC component in the signal. The 3.3V square wave is then added to 3.0V from diode D1 (VF = 0.3V) and passed through diode D2 (VF = 0.3V) to produce a 6.0V DC input voltage to the linear regulator (Toko part TK11245BM). This regulator provides the high precision output of 4.5V required for VDDH and VDD.

An alternative method would be to use a switching regulator IC to generate 4.5V from 3.3V. If 5.0V is available, a low dropout linear regulator may also be used.

2.2 VEEY - LCD Panel Drive Voltage for Vertical Power Supplies - Brightness Reference

A negative voltage potential (VEEY) must be provided as a brightness reference and a temperature compensator to the vertical logic and vertical liquid crystal driving power supplies. The recommended voltage is -32.0V with a minimum allowable value of -37.0V. Figure 2-2: "VEE Switching Power Supply" shows a standard topology buck-boost switching power supply controlled by the S1D13706 output signal GPIO5 (DD_P1).

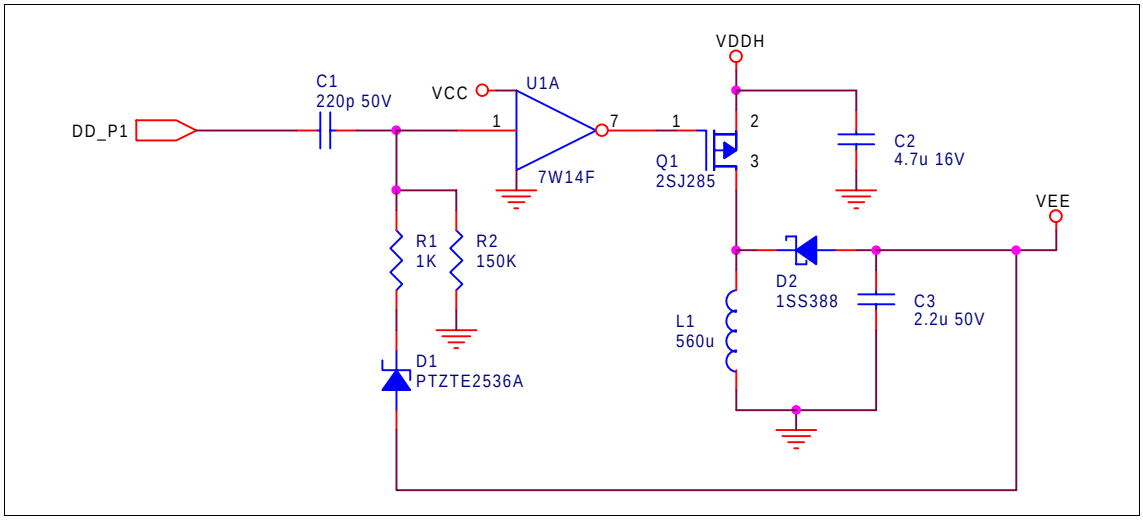


Figure 2-2: VEE Switching Power Supply

The circuit in Figure 2-2: "VEE Switching Power Supply" uses GPIO5 (DD_P1), a S1D13706 output that has a 200KHz - 96% duty cycle signal, as the switching control of the switching power supply. The duty cycle of the input to the gate of Q1 is varied by the feedback of VEE through D1. This diode feedback causes an overshoot on the rising edge of GPIO5 (DD_P1) that is proportional to the output level of VEE. This overshoot settles to a steady level after a variable time depending on how high the overshoot was. This variable time causes the high speed CMOS inverter (U1) to trigger at different times, thereby varying the duty cycle of the control input to Q1.

When Q1 turns on, the inductor L1 builds up its magnetic field using current from VDDH, and D2 is reversed biased. When Q2 turns off, current flows from L1 causing the voltage across it to reverse polarity and forward bias D2. The output capacitor C3 is charged and holds the output voltage with an acceptable output ripple when the cycle repeats (Q1 turns on). The output voltage is regulated by the feedback controlling the on/off times of Q1. The longer Q1 is turned on, the more current is stored in L1 and the resulting polarity change when Q1 is turned off is greater.

The power supply is configured to generate a voltage (VEE) of -34.0V. This voltage is used as an input into the temperature compensation circuit shown in Figure 2-3: “Temperature Compensated VEEY” , which generates VEEY for use by the vertical power supplies.

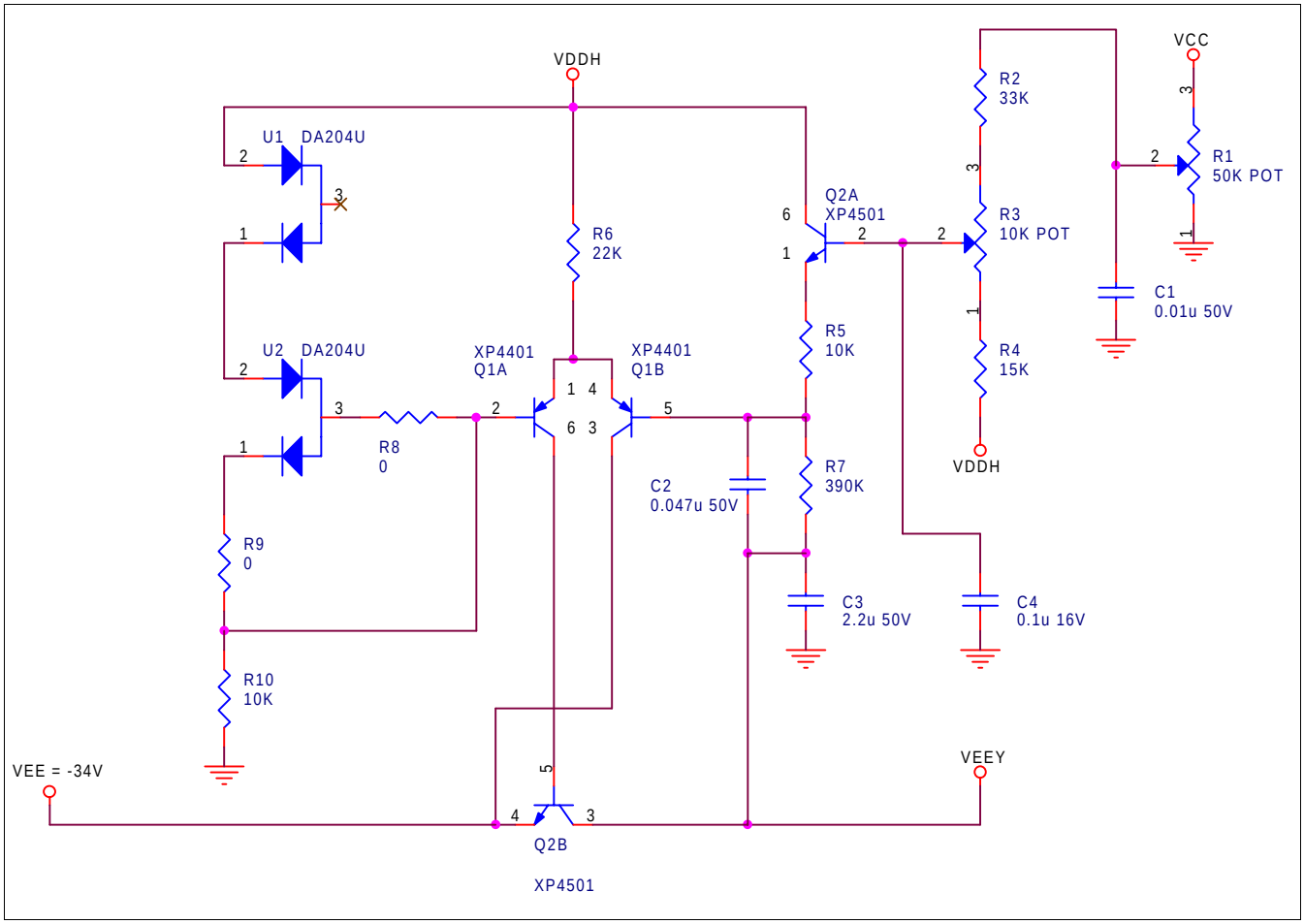


Figure 2-3: Temperature Compensated VEEY

The brightness reference VEEY must be temperature compensated to ensure the D-TFD display remains stable over a range of temperatures. The compensation circuit shown in Figure 2-3: “Temperature Compensated VEEY” uses temperature dependent diode forward voltage drops to adjust the output level of VEEY. The three serially connected diodes are connected to VDDH and grounded through resistor R10, which causes them to be forward biased. At room temperature, the forward voltage of each diode is approximately 0.7V which sets the base voltage of Q1A at 2.4V. When the temperature changes, the base voltage changes according to the characteristics of the diodes.

The base voltage at Q1A also appears at the base of Q1B which, along with potentiometers R1 and R2, determine the current flowing into resistor R7. The current flowing into R7 sets the output voltage VEEY. Therefore, any change in temperature results in a corresponding change in the output of VEEY.

2.3 VCC - Horizontal Logic Power Supply

The power supply for the horizontal logic circuitry must be set at 3.3V. The panel must be ready for use before this supply is turned on. A general purpose output pin may be used to control VCC (GPO on the S1D13706). Figure 2-4: “VCC Power Supply” shows an example of this power supply. The control signal (GPO) in this implementation activates VCC when it is low.

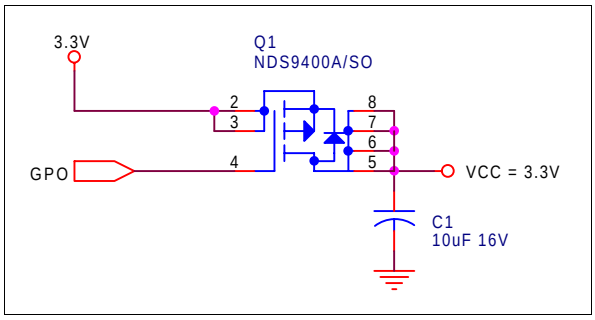


Figure 2-4: VCC Power Supply

2.4 Swing Power Supply for the Vertical Drive (V0Y) and Logic (VCCY / V5Y) Voltages

The vertical drive voltage (V0Y) and vertical logic voltages (VCCY and V5Y) require a swing power supply. To obtain the required voltage range, VEEY is used to swing the vertical system voltages through the recommended -32V to 32V range. The swing circuit is shown in Figure 2-5: “Swing Power Supply for Vertical System Voltages” .

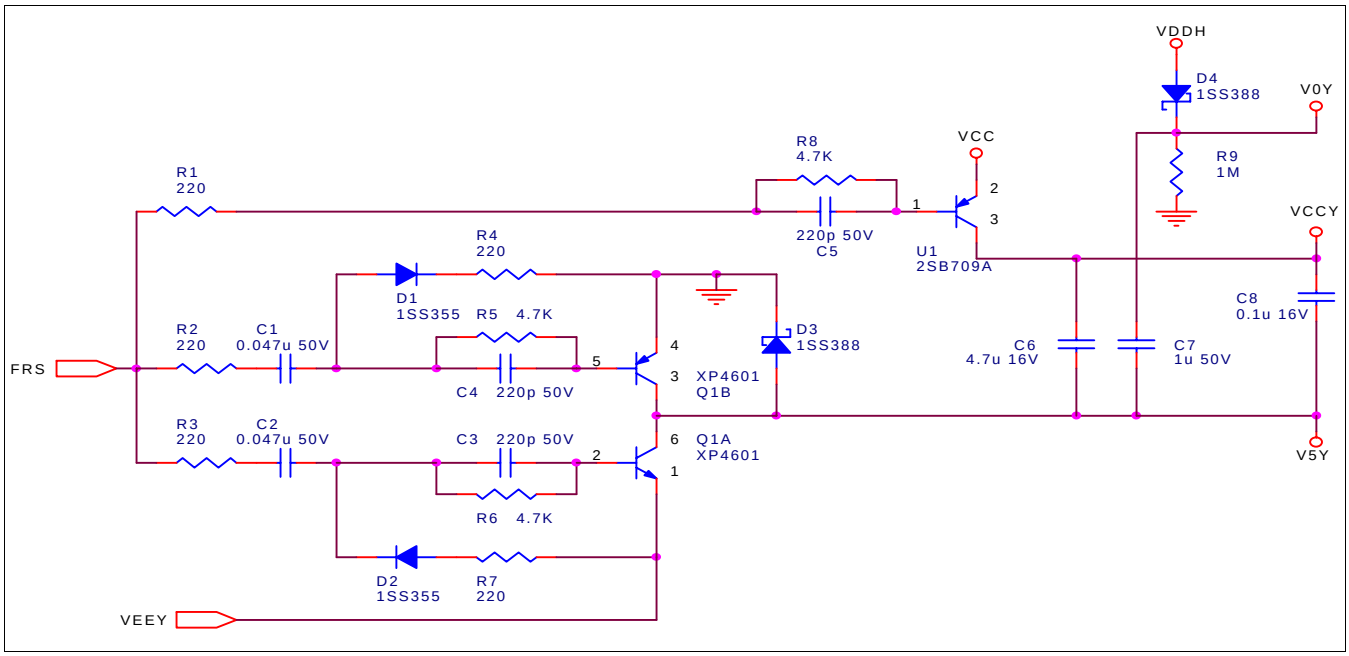


Figure 2-5: Swing Power Supply for Vertical System Voltages

The swing power supply is controlled by the S1D13706 output signal GPIO3 (FRS). When GPIO3 is low, transistor Q1B turns on and Q1A turns off. V5Y (vertical logic low potential) goes to GND. Transistor U1 also turns on and VCCY (vertical logic high potential) = VCC = 3.3V. V0Y (vertical liquid crystal drive supply) swings to | VEEY | + 4.5 when GPIO3 goes low since the reference changes to VEEY from GND for this signal.

When GPIO3 is high, transistor Q1A turns on and Q1b turns off. V5Y goes to the level of VEEY. VCCY is now referenced to VEEY and its level goes to VEEY + VCC. Diode D8 forward biases and sets V0Y = VDDH = 4.5V.

The following table shows the values of V5Y, V0Y, and VCCY for the high and low values of the control signal GPIO3 (FRS).

Table 2-1: Swing Power Supply Values

FRS	GPIO3 (FRS) Low (GND)	GPIO3 (FRS) High (3.3V)
Power Supply Potential V0Y	|VEEY| + VDDH	VDDH
Power Supply Potential VCCY	VCC	VEEY + VCC
Power Supply Potential V5Y	GND	VEEY

2.5 Level Shift and Clamp Circuit for Vertical Logic Control Signals

The vertical system power supplies are swung between positive and negative values. However, the vertical control signals from the S1D13706 are between GND and VCC. Signals going to the panel must be level shifted to the swinging power supply levels. The transition from high to low, and low to high for these control signals must take place at the same time that the swing power supply switches states. Figure 2-6: “Logic for Vertical Control Signals” shows the circuitry required for the vertical control signals. The control signals on the left are outputs from the S1D13706 and the derived control signals on the right are connected to the LCD panel.

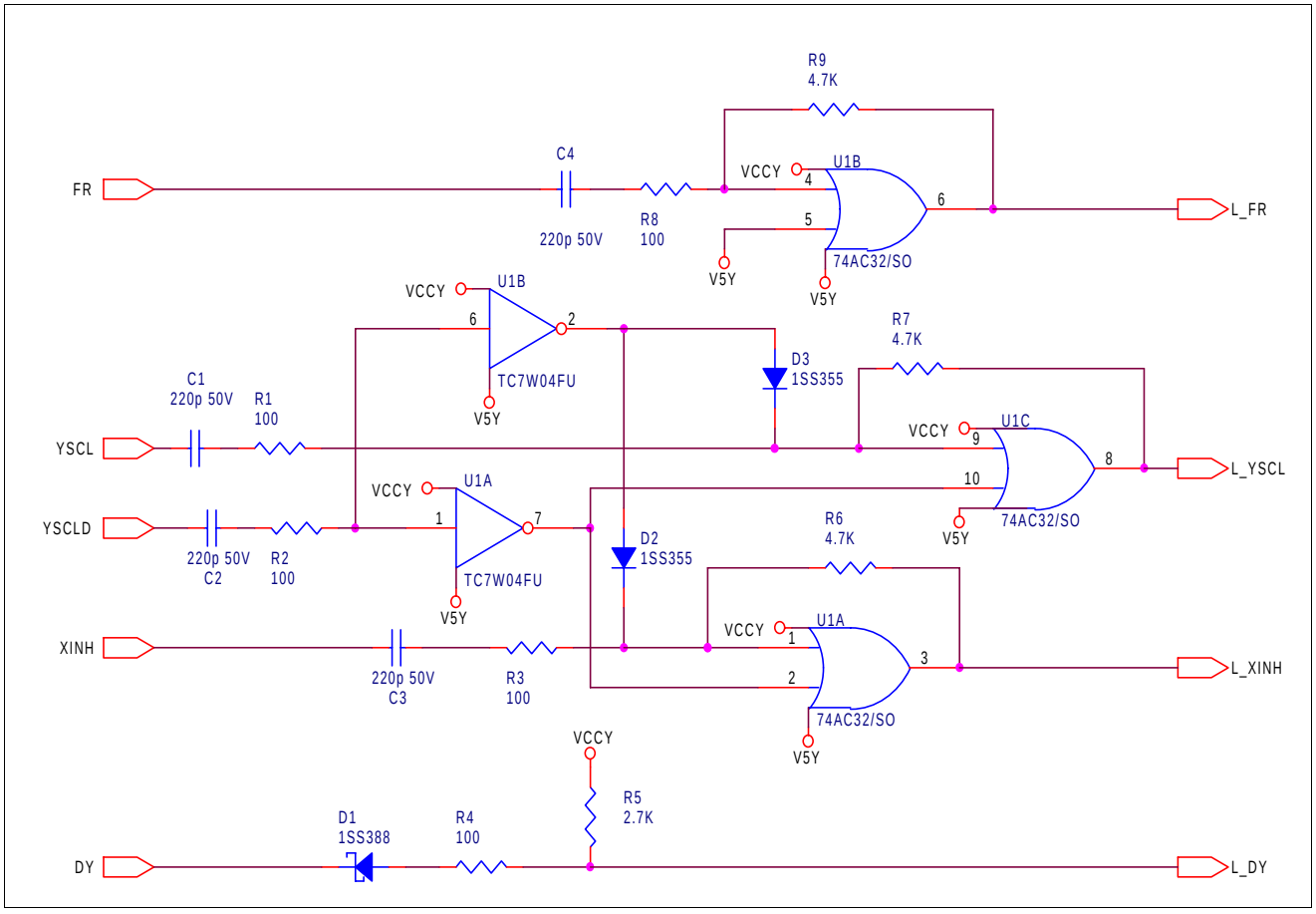


Figure 2-6: Logic for Vertical Control Signals

3 S1D13706 to D-TFD Panel Pin Mapping

The S1D13706 outputs and the external signals are sent to the D-TFD panels through two flat cable connectors. A 30-pin connector is used for the horizontal drivers and a 12-pin connector for the vertical drivers. Both D-TFD panels use the same horizontal 30-pin connector, but their vertical driver connectors are different. The 320x240 LF37SQT connector pins are swapped compared to the 160x240 LF26SCT panel connector.The following tables provide pin mapping for the various connectors.

3.1 LCD Pin Mapping for Horizontal Connector (LF37SQT and LF26SCT)

Table 3-1: LCD Pin Mapping for Horizontal Connector: Pins for Horizontal Driver

LCD Pin No.	LCD Pin Name	S1D13706 Pin Name	Description	Remarks
X-1	EIO2	No Connect	I/O enable signal	Forward Scanning: Low Reverse Scanning: Open
X-2	VCC	NIOVDD	Power supply for logic (High)	See Section 2.3, “VCC - Horizontal Logic Power Supply” on page 11.
X-3	VDDH	-	Power supply for liquid crystal drive	See Section 2.1, “VDDH and VDD - Horizontal and Vertical Analog Voltages” on page 8.
X-4	D25	FPDAT6	Blue digital data signal (MSB)	
X-5	D24	FPDAT7	Blue digital data signal	
X-6	D23	FPDAT8	Blue digital data signal	
X-7	D22	FPDAT15	Blue digital data signal	
X-8	D21	FPDAT16	Blue digital data signal	
X-9	D20	FPDAT17	Blue digital data signal (LSB)	
X-10	GCP	DRDY	PWM output - pulse width setting signal	
X-11	FR	GPIO2	AC signal for output	
X-12	LP	FPLINE	Data load and start pulse	
X-13	RES	GPIO4	Reset signal for GCP signal	
X-14	D05	FPDAT0	Red digital data signal (MSB)	
X-15	D04	FPDAT1	Red digital data signal	
X-16	D03	FPDAT2	Red digital data signal	
X-17	D02	FPDAT9	Red digital data signal	
X-18	D01	FPDAT10	Red digital data signal	
X-19	D00	FPDAT11	Red digital data signal (LSB)	
X-20	XSCL	FPSHIFT	Shift clock signal	
X-21	SHL	NIOVDD	Shift direction selection for shift registers	Forward Scanning: Low Reverse Scanning: High
X-22	D15	FPDAT3	Green digital data signal (MSB)	
X-23	D14	FPDAT4	Green digital data signal	
X-24	D13	FPDAT5	Green digital data signal	
X-25	D12	FPDAT12	Green digital data signal	
X-26	D11	FPDAT13	Green digital data signal	
X-27	D10	FPDAT14	Green digital data signal (LSB)	
X-28	GND	VSS (GND)	Common power supply logic low and liquid crystal drive	
X-29	GND	VSS (GND)	Common power supply logic low and liquid crystal drive	
X030	EIO1	VSS (GND)	I/O enable signal	Forward Scanning: Open Reverse Scanning: Low

3.2 LCD Pin Mapping for Y Connector (LF37SQT)

Table 3-2: LCD Pin Mapping for Y Connector: Pins for Y - Driver (LF37SQT)

LCD Pin No.	LCD Pin Name	S1D13706 Pin Name	Description	Remarks
Y-1	GND	VSS (GND)	Ground and power supply for liquid crystal drive	
Y-2	SHF	-	Shift direction selection for shift registers	Forward scanning: V5Y Reverse scanning: VCCY Connect to VCCY.
Y-3	XINH	GPIO0	Thinning control signal	See Section 2.5, “Level Shift and Clamp Circuit for Vertical Logic Control Signals” on page 13.
Y-4	YSCL	GPIO1	Shift clock signal	See Section 2.5, “Level Shift and Clamp Circuit for Vertical Logic Control Signals” on page 13.
Y-5	FRY	GPIO2	AC signal for output	See Section 2.5, “Level Shift and Clamp Circuit for Vertical Logic Control Signals” on page 13.
Y-6	VCCY	-	Power supply for logic high	See Section 2.4, “Swing Power Supply for the Vertical Drive (V0Y) and Logic (VCCY / V5Y) Voltages” on page 12.
Y-7	V5Y	-	Power supply for logic low and liquid crystal drive	See Section 2.4, “Swing Power Supply for the Vertical Drive (V0Y) and Logic (VCCY / V5Y) Voltages” on page 12.
Y-8	NC	-	No Connect	
Y-9	V0Y	-	Power supply for liquid crystal drive	See Section 2.4, “Swing Power Supply for the Vertical Drive (V0Y) and Logic (VCCY / V5Y) Voltages” on page 12.
Y-10	VDD	-	Power supply for liquid crystal drive	See Section 2.1, “VDDH and VDD - Horizontal and Vertical Analog Voltages” on page 8.
Y-11	DYIO2	No Connect	Start pulse signal	Forward scanning: Active low pulse Reverse scanning: Open
Y-12	DYIO1	FPFRAME	Start pulse signal	Forward scanning: Open Reverse scanning: Active low pulse See Section 2.5, “Level Shift and Clamp Circuit for Vertical Logic Control Signals” on page 13.

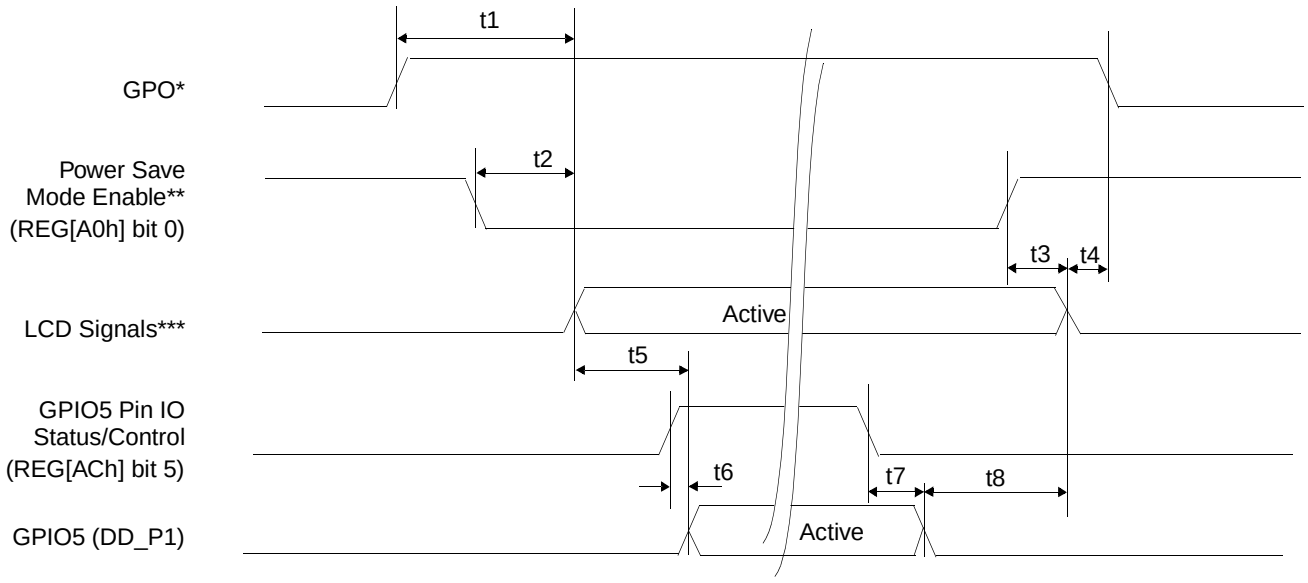
3.3 LCD Pin Mapping for Y Connector (LF26SCT)

Table 3-3: LCD Pin Mapping for Y Connector: Pins for Y - Driver (LF26SCT)

LCD Pin No.	LCD Pin Name	S1D13706 Pin Name	Description	Remarks
Y-1	DYIO1	No Connect	Start pulse signal	Forward scanning: Open Reverse scanning: Active low pulse using VCCY and V5Y for logic level.
Y-2	DYIO2	FPFRAME	Start pulse signal	Forward scanning: Active low pulse using VCCY and V5Y for logic level. Reverse scanning: Open See Section 2.5, “Level Shift and Clamp Circuit for Vertical Logic Control Signals” on page 13.
Y-3	VDD	-	Power supply for liquid crystal drive	See Section 2.4, “Swing Power Supply for the Vertical Drive (V0Y) and Logic (VCCY / V5Y) Voltages” on page 12.
Y-4	V0Y	-	Power supply for liquid crystal drive	See Section 2.4, “Swing Power Supply for the Vertical Drive (V0Y) and Logic (VCCY / V5Y) Voltages” on page 12.
Y-5	NC	No Connect	No Connect	
Y-6	V5Y	-	Power supply for logic low and liquid crystal drive	See Section 2.4, “Swing Power Supply for the Vertical Drive (V0Y) and Logic (VCCY / V5Y) Voltages” on page 12.
Y-7	VCCY	-	Power supply for logic high	See Section 2.4, “Swing Power Supply for the Vertical Drive (V0Y) and Logic (VCCY / V5Y) Voltages” on page 12.
Y-8	FRY	GPIO2	AC signal for output	See Section 2.5, “Level Shift and Clamp Circuit for Vertical Logic Control Signals” on page 13.
Y-9	YSCL	GPIO1	Shift clock signal	See Section 2.5, “Level Shift and Clamp Circuit for Vertical Logic Control Signals” on page 13.
Y-10	XINH	GPIO0	Thinning control signal	See Section 2.5, “Level Shift and Clamp Circuit for Vertical Logic Control Signals” on page 13.
Y-11	SHF	-	Shift direction selection for shift registers	Forward scanning: V5Y Reverse scanning: VCCY Connect to V5Y.
Y-12	GND	VSS (GND)	Ground and power supply for liquid crystal drive	

4 Power-On/Off Sequence

The D-TFD panel requires a specific sequence to power-on/off. For further information on power sequencing the D-TFD panel, see the specification for each specific panel.



*It is recommended that LCD power be controlled using the general output pin GPO.
**The LCD power-off sequence is activated by programming the Power Save Mode Enable bit (REG[A0h] bit 0) to 1.
***LCD Signals include: FPDAT[17:0], FPSHIFT, FPLINE, FPFRAME, DRDY, GPIO6, and GPIO[4:0].

Figure 4-1: D-TFD Power-On/Off Sequence Timing

Table 4-1: D-TFD Power-On/Off Sequence Timing

Symbol	Parameter	Min	Max	Units
t1	LCD power active to LCD signals active		Note 1	
t2	Power Save Mode Enable bit low to LCD signals active	0	20	ns
t3	Power Save Mode Enable bit high to LCD signals low		20	ns
t4	LCD signals low to LCD power inactive		Note 1	
t5	LCD signals active to GPIO5 active	2		FRAME
t6	GPIO5 Pin IO Status high to GPIO5 active		20	ns
t7	GPIO5 Pin IO Status low to GPIO5 inactive		20	ns
t8	GPIO5 inactive to LCD signals low	3		FRAME

1. t1 and t 4 are controlled by software and must be determined from the timing requirements of the panel connected.

5 GCP Data Signal

The D-TFD panel uses a 256-bit bit chain to control the pixel/FPSHIFT (XSCL) positions relative to the falling edge of the GPIO4 (RES) signal. A one in each bit indicates the presence of a GCP pulse at that pixel/XSCL position. A zero indicates the absence of a GCP pulse. For D-TFD AC Timing required by the S1D13706, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

5.1 GCP Data Structure

The S1D13706 uses two registers to program the GCP Data:

- D-TFD GCP Index Register (REG[28h]
- D-TFD GCP Data Register (REG[2Ch])

The 256-bit GCP data is organized into 32 8-bit data registers, each addressable by the D-TFD GCP Index register (REG[28h]).

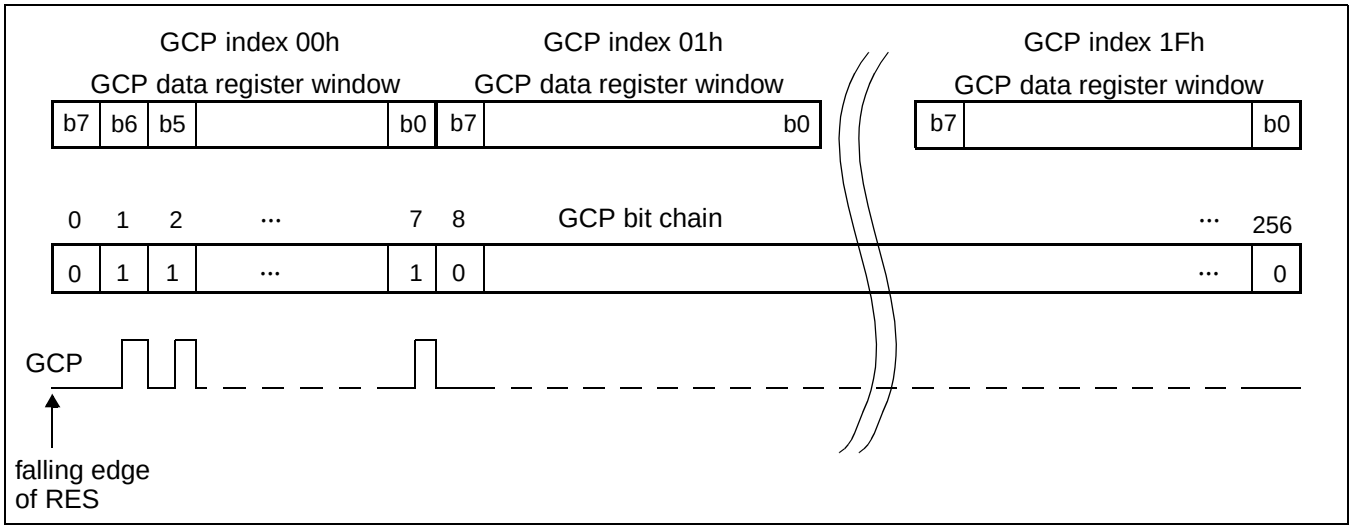


Figure 5-1: GCP Data

5.2 Programming GCP Data

To program the GCP Data bit chain the following procedure must be followed.

1. Program the D-TFD GCP Index Register (REG[28h]).
2. Program the D-TFD GCP Data Register (REG[2Ch]).
3. Increment the D-TFD GCP Index Register (REG[28h]).
4. Return to step 2 and repeat until all 32 8-bit segments are programmed.

The following values must be programmed into the GCP data bit chain for the LF37SQT and LF26SCT D-TFD panels.

Table 5-1: GCP Data Bit Chain Values for LF37SQT and LF26SCT

Index	Value	Index	Value	Index	Value	Index	Value
00h	52h	08h	49h	10h	2Ah	18h	00h
01h	2Ah	09h	24h	11h	52h	19h	00h
02h	92h	0Ah	92h	12h	49h	1Ah	00h
03h	22h	0Bh	49h	13h	24h	1Bh	00h
04h	48h	0Ch	49h	14h	48h	1Ch	00h
05h	88h	0Dh	4Ah	15h	84h	1Dh	00h
06h	91h	0Eh	52h	16h	00h	1Eh	00h
07h	22h	0Fh	A5h	17h	00h	1Fh	00h

6 Test Software

Test utilities and display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and display drivers are available from your sales support contact or www.eea.epson.com.

7 References

7.1 Documents

- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, Document Number X31B-A-001-xx.
- Epson Research and Development, Inc., *S1D13706 Programming Notes and Examples*, Document Number X31B-G-003-xx.

7.2 Document Sources

- Epson Electronics America Website: http://www.eea.epson.com.

8 Technical Support

8.1 EPSON LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Interfacing to the Motorola MC68030 Microprocessor

Document Number: X31B-G-013-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13706 Embedded Memory LCD Controller and the Motorola MC68030 microprocessor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at techpubs@erd.epson.com.

2 Motorola MC68030 Bus Interface

2.1 Overview

The MC68030 is a second generation enhanced microprocessor from the Motorola M68000 family of devices. The MC68030 is a 32-bit microprocessor with a 32-bit address bus and a 32-bit data bus. The microprocessor supports both asynchronous and synchronous bus cycles and burst data transfers. The bus also supports dynamic bus sizing which automatically determines device port size on a cycle-by-cycle basis.

2.2 Dynamic Bus Sizing

The MC68030 supports dynamic bus sizing using the following signals.

- SIZ1 and SIZ0 indicate the number of bytes remaining to be transfered for the current bus cycle.

Table 2-1: SIZ Signal Encoding

SIZ1	SIZ0	Bytes Remaining
0	1	1 Byte
1	0	2 Bytes (Word)
1	1	3 Bytes
0	0	4 Bytes (Double Word)

- DSACK1 and DSACK0 (the data transfer size acknowledge signals) indicate the size of the external port and acknowledge the end of the cycle.

Table 2-2: $\overline{DSACK}$ Decoding

$\overline{DSACK1}$	$\overline{DSACK0}$	Result
1	1	Insert Wait States in the Current Bus Cycle
1	0	Complete Cycle - Data Bus Port Size is 8 bits
0	1	Complete Cycle - Data Bus Port Size is 16 bits
0	0	Complete Cycle - Data Bus Port Size is 32 bits

- A0 and A1 determine which portion of the data bus the data is transferred on and whether the address is misaligned.

2.3 Asynchronous / Synchronous Bus Operation

The MC68030 bus can operate asynchronously or synchronously. Asynchronous operation requires $\overline{DSACK0}$, $\overline{DSACK1}$, $\overline{AS}$, and $\overline{DS}$ to control transfers. The $\overline{DSACK}$ signals specify the port width and insert wait states in the current bus cycle. $\overline{AS}$ (the address strobe)

signals the start of a bus cycle by indicating a valid address has been placed on the bus. $\overline{DS}$ (the data strobe) is used as a condition for valid data on the data bus. SIZ selects the active portions of the data bus. R/$\overline{W}$ indicates a read or write operation.

Synchronous bus cycles operate much like asynchronous cycles except only 32-bit port sizes are allowed. In this mode the $\overline{DSACK}$ signals are not required. Wait states are inserted with the synchronous signal ($\overline{STERM}$) which signals that the data is to be latched on the next clock when asserted.

3 S1D13706 Host Bus Interface

The S1D13706 directly supports multiple processors. The S1D13706 implements a MC68K #2 Host Bus Interface which directly supports the Motorola MC68030 microprocessor.

The MC68K #2 Host Bus Interface is selected by the S1D13706 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13706 configuration, see Section 4.2, “S1D13706 Hardware Configuration” on page 13.

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

S1D13706 Pin Names	Motorola MC68030
AB[16:0]	A[16:0]
DB[15:0]	D[31:16]
WE1#	$\overline{DS}$
CS#	External Decode
M/R#	External Decode
CLKI	CLK
BS#	$\overline{AS}$
RD/WR#	R/$\overline{W}$
RD#	External Decode of SIZ1 and SIZ0
WE0#	SIZ0
WAIT#	$\overline{DSACK1}$
RESET#	System $\overline{RESET}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals.

- CLKI is a clock input which is required by the S1D13706 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, CLK from the Motorola MC68030 is used for CLKI.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the MC68030 address (A[16:0) and data bus (D[31:16]), respectively. CNF4 must be set to select big endian mode.
- Chip Select (CS#) must be driven low by the external address decode circuitry whenever the S1D13706 is accessed by the Motorola MC68030.
- M/R# (memory/register) selects between memory or register accesses. This signal is generated by the external address decode circuitry.
- WE0# connects to SIZ0, one of the transfer size signals of the MC68030. Along with SIZ1 this signal indicates how many bytes are to be transferred during the current cycle.
- WE1# connects to $\overline{DS}$ (the data strobe signal from the MC68030) and must be driven low when valid data has been placed on the bus.
- RD# connects to external decode circuitry of SIZ1, one of the transfer size signals of the MC68030. Along with SIZ0 this signal indicates how many bytes are to be transferred during the current cycle.
- RD/WR# is the read or write enable signal and connects to R/$\overline{W}$ of the MC68030. This signal drives low when the MC68030 is writing to the S1D13706 and drives high when the MC68030 is reading from the S1D13706.
- WAIT# connects to $\overline{DSACK1}$ and is a signal which is output from the S1D13706 which indicates the MC68030 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since MC68030 accesses to the S1D13706 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13706 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- BS# connects to AS (the address strobe from the MC68030) and must be driven low when a valid address has been placed on the address bus.

4 MC68030 to S1D13706 Interface

4.1 Hardware Description

The interface between the S1D13706 and the MC68030 requires external glue logic. Address decoding logic is required to provide the chip select (CS#) and memory/register (M/R#) signals to the S1D13706 since the MC68030 does not have a chip select module.

SIZ1 is modified to signal the S1D13706 that 24-bit and 32-bit accesses are to converted into word-byte and word-word accesses, respectively. Misaligned operands for 24-bit and 32-bit cycles are not supported with this external circuitry for SIZ1.RD# must be connected to the following logic circuitry instead of directly to SIZ1.

RD# = ’(SIZ0 & SIZ1’).

The polarity of the WAIT# signal must be selected as active high by connecting CNF5 to NIO V_{DD} (see Table 4-1:, “Summary of Power-On/Reset Configuration Options,” on page 13).

The diagram below shows a typical implementation of the MC68030 to S1D13706 interface.

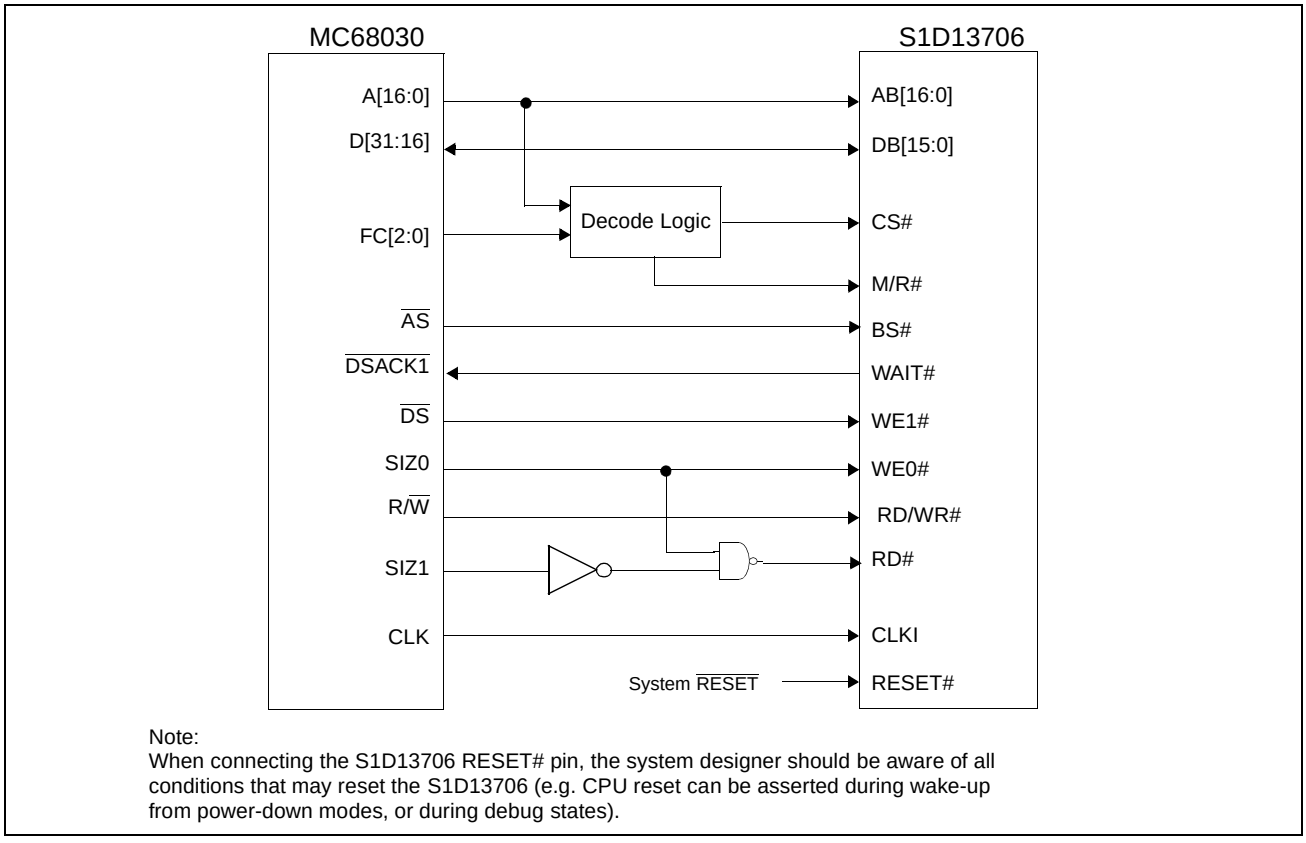


Figure 4-1: Typical Implementation of MC68030 to S1D13706 Interface

Note

The interface was designed using a Motorola MC68030 Integrated Development Platform (IDP).

4.2 S1D13706 Hardware Configuration

The S1D13706 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13706 to Motorola MC68030 microprocessor.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13706 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[2:0]	010 = MC68K #2 Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table "" for recommended settings	

= configuration for MC68030 microprocessor

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for MC68030 microprocessor

4.3 Register/Memory Mapping

The MC68030 IDP board uses the first 256M bytes of address space, therefore the S1D13706 can be mapped anywhere beyond this boundary. The S1D13706 uses two 128K byte blocks which are selected using M/R# from the address decoder. The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block. Registers were located at memory location 10A0 0000h and the display buffer at memory location 10E0 0000h. The address space for the S1D13706 is user dependent.

5 Software

Test utilities and Windows® CE v2.11/2.12 display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE v2.11/2.12 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE v2.11/2.12 display drivers are available from your sales support contact or on the internet at http://www.eea.epson.com.

6 References

6.1 Documents

- Motorola Inc., *MC68030 32-bit Enhanced Microprocessor User's Manual*, Motorola Publication no. MC68030UM/; available on the Internet at http://www.mot.com/SPS/ADC/pps/_subpgs/_documentation/
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, Document Number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 Evaluation Board User Manual*, Document Number X31B-G-004-xx.
- Epson Research and Development, Inc., *Programming Notes and Examples*, Document Number X31B-G-003-xx.

6.2 Document Sources

- Motorola Inc. Literature Distribution Center: (800) 441-2447.
- Motorola Inc. Website: http://www.mot.com.
- Epson Electronics America website: http://www.eea.epson.com.

7 Technical Support

7.1 EPSON LCD/CRT Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

7.2 Motorola MC68030 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Interfacing to the Motorola RedCap2 DSP With Integrated MCU

Document Number: X31B-G-014-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to provide an interface between the S1D13706 Embedded Memory LCD Controller and the Motorola REDCAP2 processor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America Website at www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at techpubs@erd.epson.com.

2 Interfacing to the REDCAP2

2.1 The REDCAP2 System Bus

REDCAP2 integrates a RISC microprocessor (MCU) and a general purpose digital signal processor (DSP) on a single chip. The External Interface Module (EIM) handles the interface to external devices. This section provides an overview of the operation of the REDCAP2 bus in order to establish interface requirements.

2.2 Overview

REDCAP2 uses a 22-bit address bus (A[21:0]) and 16-bit data bus (D[15:0]). All IO is synchronous to a square wave reference clock called CKO. The CKO source can be the DSP clock or the MCU clock and is selected/disabled in the Clock Control Register (CKCTL).

REDCAP2 can generate up to 6 independent chip select outputs. Each chip select has a memory range of 16M bytes and can be independently programmed for wait-states and port size.

Note
REDCAP2 does not provide a wait or termination acknowledge signal to external devices. Therefore, all external devices must guarantee a fixed cycle length.

2.3 Bus Transactions

The chip initiates a data transfer by placing the memory address on address lines A0 through A21. Several control signals are provided with the memory address.

- R/$\overline{W}$ — set high for read cycles and low for write cycles.
- $\overline{EB0}$ — active low signal indicates access to data byte 0 (D[15:8]) during read or write cycles.
- $\overline{EB1}$ — active low signal indicates access to data byte 1 (D[7:0]) during read or write cycles.
- $\overline{OE}$ — active low signal indicates read accesses and enables slave devices to drive the data bus.

Figure 2-1: “REDCAP2 Memory Read Cycle” on page 9 illustrates a typical memory read cycle on the REDCAP2 bus.

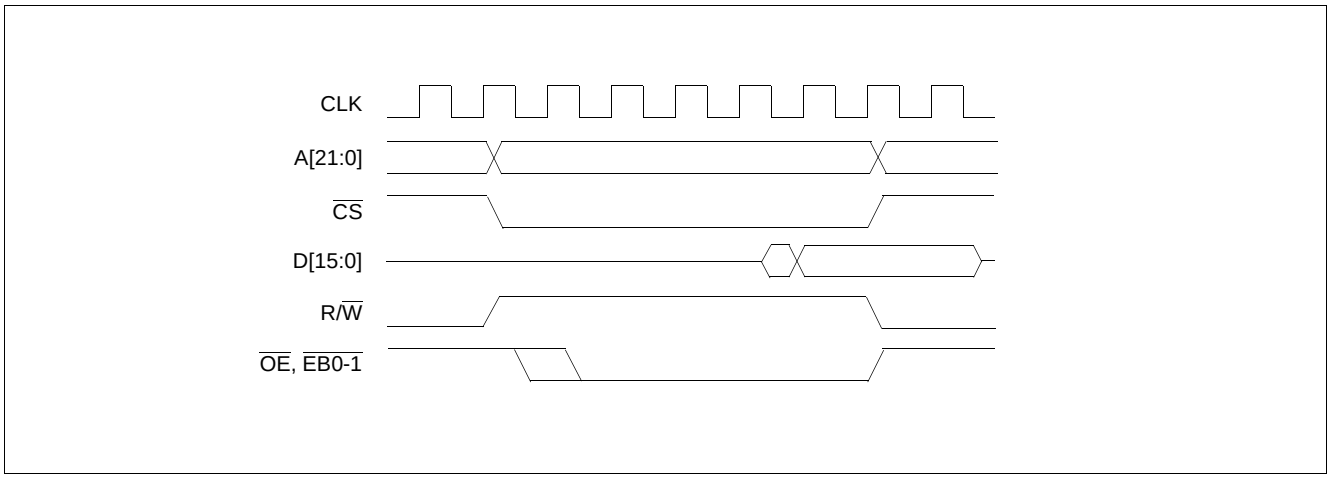


Figure 2-1: REDCAP2 Memory Read Cycle

Figure 2-2: “REDCAP2 Memory Write Cycle” on page 9 illustrates a typical memory write cycle on the REDCAP2 bus.

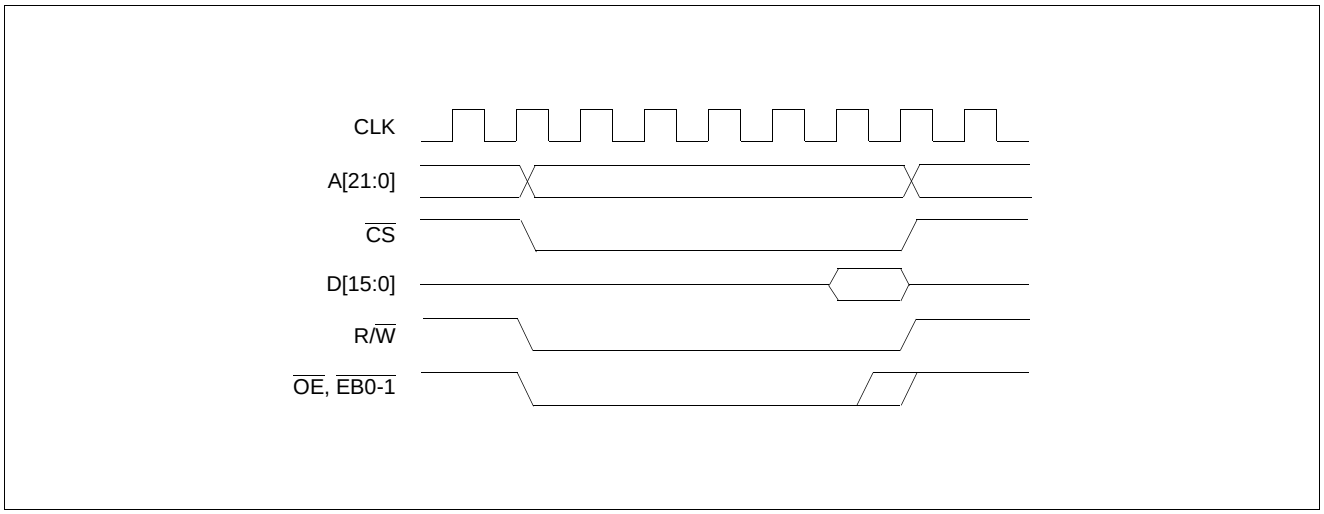


Figure 2-2: REDCAP2 Memory Write Cycle

3 S1D13706 Host Bus Interface

The S1D13706 implements a 16-bit native REDCAP2 host bus interface which is used to interface to the REDCAP2 processor.

The REDCAP2 host bus interface is selected by the S1D13706 on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. For details on S1D13706 configuration, see Section 4.3, “S1D13706 Hardware Configuration” on page 15.

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

S1D13706 Pin Names	REDCAP2
AB[16:0]	A[16:0]
DB[15:0]	D[15:0]
WE1#	$\overline{EB0}$
M/R#	A17
CS#	REDCAP2 Internal Chip Select
CLKI	CKO
BS#	Connected to HIO V_{DD}
RD/WR#	R/$\overline{W}$
RD#	$\overline{OE}$
WE0#	$\overline{EB1}$
WAIT#	N/A
RESET#	$\overline{RST\_OUT}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals:

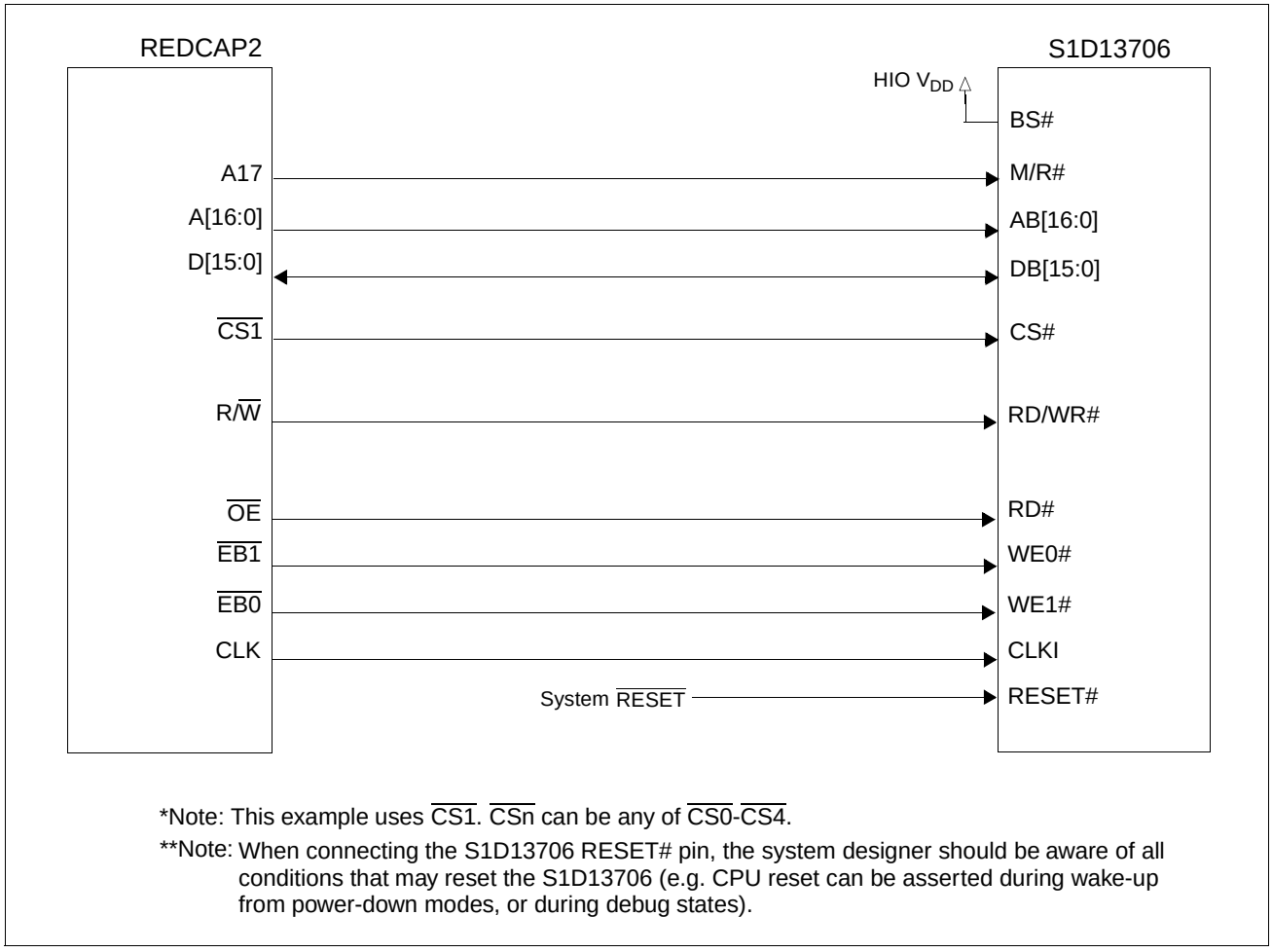
- CLKI is a clock input which is required by the S1D13706 host bus interface and connects to CKO of the REDCAP2.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the REDCAP2 bus address (A[16:0]) and data bus (D[15:0]), respectively. CNF[2:0] and CNF4 must be set to select the REDCAP2 host bus interface with big endian mode.
- M/R# (memory/register) selects between memory or register access. It may be connected to an address line, allowing REDCAP2 bus address A17 to be connected to the M/R# line.
- CS# (Chip Select) must be driven low whenever the S1D13706 is accessed by the REDCAP2 bus.
- RD/WR# connects to R/$\overline{W}$ which indicates whether a read or a write access is being performed on the S1D13706.
- WE1# and WE0# connect to $\overline{EB0}$ and $\overline{EB1}$ (Enable Byte 0 and 1) for byte steering.
- RD# connects to $\overline{OE}$ (Output Enable). This signal must be driven by the REDCAP2 bus to indicate the bus access is a read and enables slave devices to drive the data bus with read data.
- The BS# and WAIT# signals are not needed for this bus interface, they should be connected to HIO V_{DD}.

4 REDCAP2 to S1D13706 Interface

4.1 Hardware Description

The interface between the S1D13706 and the REDCAP2 requires no external glue logic. The information in this section describes the environment necessary to connect the S5U13706B00C Evaluation Board and the Motorola DSP56654 Application Development Module (ADM). For a list of connections between the pins and signals of the REDCAP2 and the S1D13706 see Table 4-1:,"List of Connections from REDCAP2 ADM to S5U13706B00C" on page 13.

The following figure demonstrates a typical implementation of the S1D13706 to REDCAP2 interface.



*Note: This example uses CS1. CSn can be any of CS0-CS4.

**Note: When connecting the S1D13706 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13706 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 4-1: Typical Implementation of REDCAP2 to S1D13706 Interface

4.2 Hardware Connections

The following table details the connections between the pins and signals of the REDCAP2 and the S1D13706.

Table 4-1: List of Connections from REDCAP2 ADM to S5U13706B00C

REDCAP2 Signal Name	REDCAP2ADS Connector and Pin Name	S1D13706 Signal Name
A17	P9-34	M/R#
A16	P9-33	AB20
A15	P9-32	AB19
A14	P9-31	AB18
A13	P9-30	AB17
A12	P9-29	AB16
A11	P9-28	AB15
A10	P9-27	AB14
A9	P9-26	AB13
A8	P9-25	AB12
A7	P9-24	AB11
A6	P9-23	AB10
A5	P9-22	AB9
A4	P9-21	AB8
A3	P9-20	AB7
A2	P9-19	AB6
A1	P9-18	AB5
A0	P9-17	AB4
D15	P9-16	DB15
D14	P9-15	DB14
D13	P9-14	DB13
D12	P9-13	DB12
D11	P9-12	DB11
D10	P9-11	DB10
D9	P9-10	DB9
D8	P9-9	DB8
D7	P9-8	DB7
D6	P9-7	DB6
D5	P9-6	DB5
D4	P9-5	DB4
D3	P9-4	DB3
D2	P9-3	DB2
D1	P9-2	DB1
D0	P9-1	DB0
RES_OUT	P24-6	RESET#

Table 4-1: List of Connections from REDCAP2 ADM to S5U13706B00C (Continued)

REDCAP2 Signal Name	REDCAP2ADS Connector and Pin Name	S1D13706 Signal Name
CLK0	P24-3	BUSCLK
$\overline{CS1}$	P9-40	CS#
R/$\overline{W}$	P9-47	RD/WR#
$\overline{OE}$	P9-48	RD#
$\overline{EB1}$	P9-46	WE0#
$\overline{EB0}$	P9-45	WE1#
Gnd	P24-20 / P9-50	Vss

Note

In order for the S5U13706B00C evaluation board to work with the ADM, pin 5 and pin 13 of U28 on the ADM must be connected to $V_{DD.}$ This ensures that the DIR signal of transceivers U17 and U18 is low only during read access, even when EBC in the CS1 Control Register is set to 0.

4.3 S1D13706 Hardware Configuration

The S1D13706 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13706 to Motorola REDCAP2 microprocessor.

Table 4-2: Summary of Power-On/Reset Options

S1D13706 Pin Name	**state of this pin at rising edge of RESET# is used to configure:**(1/0)	
	1	**0**
CNF[2:0]	101 = REDCAP2 Host Bus Interface	
CNF3	GPIO pins as inputs at power-on	GPIO pins as HR-TFT/ D-TFD outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	WAIT# is active high	WAIT# is active low
CNF[7:6]	CLKI to BCLK divide select: CNF7 CNF6 CLKI to BCLK Divide Ratio 0 0 1 : 1 0 1 2 : 1 1 0 3 : 1 1 1 4 : 1	

= configuration for REDCAP2 microprocessor

4.4 Register/Memory Mapping

The S1D13706 is a memory mapped device. The S1D13706 uses two 128K byte blocks which are selected using A17 from the REDCAP2 bus (A17 is connected to the S1D13706 M/R# pin). The internal registers occupy the first 128K byte block and the 80K byte display buffer occupies the second 128K byte block. In this example, the S1D13706 internal registers are accessed starting at address 4100 0000h and the display buffer is accessed starting at address 4102 0000h.

Each Chip Select on the REDCAP2 is allocated a 16M byte block. However, the S1D13706 only needs a 256K byte block of memory to accommodate its register set and 80K byte display buffer. For this reason, only address bits A[17:0] are used while A[21:18] are ignored. The S1D13706's memory and register are aliased every 256K bytes in the 16M byte CS1 address space.

Note
If aliasing is not desirable, the upper addresses must be fully decoded.

4.5 REDCAP2 Chip Select Configuration

In this example, Chip Select 1 controls the S1D13706. The following options are selected in the CS1 Control Register.

- CSEN = 1 — Chip Select function enabled.
- WP = 0 — writes allowed.
- SP = 0 — user mode access allowed.
- DSZ = 10 — 16-bit Port.
- EBC = 0 — assert $\overline{\text{EB0-1}}$ for both reads and writes.
- WEN = 1 — $\overline{\text{EB0-1}}$ negated half a clock earlier during write cycle.
- OEA = 1 — $\overline{\text{OE}}$ asserted half a clock later during a read cycle.
- CSA = 0 — Chip Select asserted as early as possible. No idle cycle inserted between back-to-back external transfers.
- EDC = 1 — an idle cycle is inserted after a read cycle for back-to-back external transfers, unless the next cycle is a read cycle to the same $\overline{\text{CS}}$ bank.
- WWS = 0 — same length for reads and writes.

5 Software

Test utilities and Windows® CE v2.11/2.12 display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE v2.11/2.12 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE v2.11/2.12 display drivers are available from your sales support contact or on the internet at www.eea.epson.com.

6 References

6.1 Documents

- Motorola Inc., *REDCAP2 Digital Signal Processor Integrated With MCU Product Specifications Rev. 1.2ext.*
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, Document Number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 Evaluation Board User Manual*, Document Number X31B-G-004-xx.
- Epson Research and Development, Inc., *S1D13706 Programming Notes and Examples*, Document Number X31B-G-003-xx.

6.2 Document Sources

- Motorola Literature Distribution Center: (800) 441-2447.
- Epson Electronics America Website: www.eea.epson.com.

7 Technical Support

7.1 EPSON LCD/CRT Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

7.2 Motorola REDCAP2 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

THIS PAGE LEFT BLANK

EPSON®

S1D13706 Embedded Memory LCD Controller

Interfacing to 8-bit Processors

Document Number: X31B-G-015-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13706 Embedded Memory LCD Controller and 8-bit processors. This document is not intended to cover all possible implementation, but provides a generic example of how such an interface can be accomplished.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to an 8-bit Processor

2.1 The Generic 8-bit Processor System Bus

Although the S1D13706 does not directly support an 8-bit CPU, an 8-bit interface can be achieved with minimal external logic.

Typically, the bus of an 8-bit microprocessor is straight forward with minimal CPU and system control signals. To connect a memory mapped device such as the S1D13706, only the write, read, and wait control signals, plus the data and address lines, need to be interfaced. Since the S1D13706 is a 16-bit device, some external logic is required.

3 S1D13706 Host Bus Interface

The S1D13706 directly supports multiple processors. The S1D13706 implements a 16-bit Generic #2 Host Bus Interface which can be adapted for use with an 8-bit processor.

The Generic #2 Host Bus Interface is selected by the S1D13706 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13706 configuration, see Section 4.2, "S1D13706 Hardware Configuration" on page 12.

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

S1D13706 Pin Names	Generic #2	Comments
AB[16:0]	A[16:0]	—
DB[15:0]	D[15:0]	—
WE1#	Byte High Enable (BHE#)	External decode required
CS#	Chip Select	External decode required
M/R#	Memory/Register Select	External decode required
CLKI	BUSCLK	—
BS#	connect to HIO V_{DD}	—
RD/WR#	connect to HIO V_{DD}	—
RD#	RD#	—
WE0#	WE#	—
WAIT#	WAIT#	—
RESET#	Inverted RESET	—

3.2 Host Bus Interface Signals

The S1D13706 Generic #2 Host Bus Interface requires the following signals from an 8-bit processor.

- CLKI is a clock input which is required by the S1D13706 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock.
- The address inputs AB[16:0] connect directly to the 8-bit processor address lines (A[16:0). If the specific 8-bit processor cannot implement all 17 address lines required by the S1D13706, only a portion of the 80K byte S1D13706 display buffer is accessible. For example, if only AB[15:0] are supported, only the first 64K byte of the display buffer is available.
- The data bus DB[15:0] must be connected so that the 8-bit processor data lines (D[7:0]) are connected to both DB[15:8] and DB[7:0] of the S1D13706. CNF4 must be set to select little endian mode.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper register and memory address space.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A17 to be connected to the M/R# line.

  Note
  If A17 is unavailable on the 8-bit processor, an external decode must be used to generate the M/R# signal.

- BHE# is the high byte enable for both read and write cycles and connects to the high byte chip select signal.

  Note
  In an 8-bit environment, this signal is driven by inverting address line A0 thus indicating that odd addresses are to be read/write on the high byte of the data bus.

- WE# connects to WE# (the write enable signal) and must be driven low when the 8-bit processor is writing data to the S1D13706.
- RD# connects to RD# (the read enable signal) and must be driven low when the 8-bit processor is reading data from the S1D13706.
- WAIT# is a signal output from the S1D13706 that indicates the 8-bit processor must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since host CPU bus accesses to the S1D13706 may occur asynchronously to the display update, it is possible that contention may occur in accessing the 13706 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete. This signal is active low and may need to be inverted if the host CPU wait state signal is active high.
- The Bus Status (BS#) and Read/Write (RD/WR#) signals are not used in this implementation of a generic 8-bit processor using the Generic #2 Host Bus Interface. These pins must be tied high (connected to HIO V_{DD}).

4 8-Bit Processor to S1D13706 Interface

4.1 Hardware Connections

The interface between the S1D13706 and an 8-bit processor requires minimal glue logic. A decoder is used to generate the chip select for the S1D13706 based on where the S1D13706 is mapped into memory. Alternatively, if the processor supports a chip select module, it can be programmed to generate a chip select for the S1D13706 without the need of an address decoder.

An inverter inverts A0 to generate the BHE# signal for the S1D13706. If the 8-bit host interface has an active high WAIT signal, it must be inverted as well.

BS# (bus start) and RD/WR# are not used by the Generic #2 Host Bus Interface and should be tied high (connected to HIO V_{DD}).

In order to support an 8-bit processor with a 16-bit peripheral, the low and high order bytes of the data bus must be connected together. The following diagram shows a typical implementation of an 8-bit processor to S1D13706 interface.

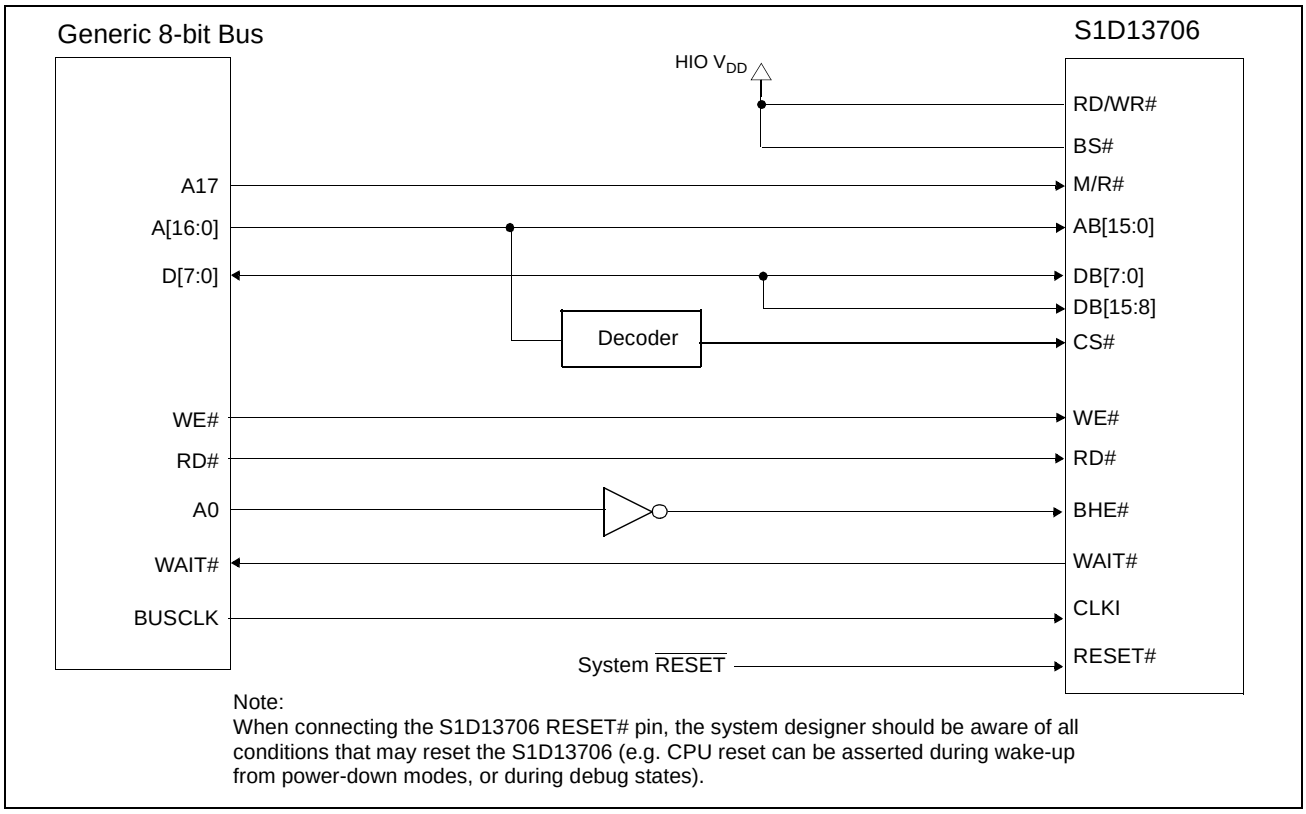


Figure 4-1: Typical Implementation of 8-bit Processor to S1D13706 Interface

4.2 S1D13706 Hardware Configuration

The S1D13706 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13706 to generic 8-bit processor.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13706 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[2:0]	100 = Generic #2 Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-2: "CLKI to BCLK Divide Selection" for recommended setting	

= configuration for generic 8-bit processor

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for generic 8-bit processor

4.3 Register/Memory Mapping

The S1D13706 is a memory mapped device. The S1D13706 uses two 128K byte blocks which are selected using A17 from the 8-bit processor (A17 is connected to the S1D13706 M/R# pin). The internal registers occupy the first 128K byte block and the 80K byte display buffer occupies the second 128K byte block.

An external decoder can be used to decode the address lines and generate a chip select for the S1D13706 whenever the selected 128k byte memory block is accessed. If the processor supports a general chip select module, its internal registers can be programmed to generate a chip select for the S1D13706 whenever the S1D13706 memory block is accessed.

5 Software

Test utilities and Windows® CE v2.11/2.12 display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE v2.11/2.12 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE v2.11/2.12 display drivers are available from your sales support contact or on the internet at http://www.eea.epson.com.

6 References

6.1 Documents

- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 Evaluation Board User Manual*, document number X31B-G-004-xx.
- Epson Research and Development, Inc., *S1D13706 Programming Notes and Examples*, Document Number X31B-G-003-xx.

6.2 Document Sources

- Epson Electronics America website: http://www.eea.epson.com

7 Technical Support

7.1 EPSON LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

THIS PAGE LEFT BLANK

EPSON®

S1D13706 Embedded Memory LCD Controller

Interfacing to the Motorola MC68VZ328 Dragonball Microprocessor

Document Number: X31B-G-016-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13706 Embedded Memory LCD Controller and the Motorola MC68VZ328 Dragonball VZ microprocessor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Electronics America website at http://www.eea.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to the MC68VZ328

2.1 The MC68VZ328 System Bus

The Motorola MC68VZ328 "Dragonball VZ" is the third generation in the Dragonball microprocessor family. The Dragonball VZ is an integrated controller designed for handheld products. It is based upon the FLX68000 microprocessor core and uses a 24-bit address bus and 16-bit data bus. The Dragonball VZ is faster than its predecessors and the DRAM controller now supports SDRAM. The bus interface consists of all the standard MC68000 bus interface signals except $\overline{AS}$, plus some new signals intended to simplify the interface to typical memory and peripheral devices. The 68000 signals are multiplexed with IrDA, SPI and LCD controller signals.

The MC68000 bus control signals are well documented in the Motorola user manuals, and are not be described here. The new signals are as follows.

- Output Enable ($\overline{OE}$) is asserted when a read cycle is in progress. It is intended to connect to the output enable control signal of a typical static RAM, EPROM, or Flash EPROM device.
- Upper Write Enable and Lower Write Enable ($\overline{UWE}$ / $\overline{LWE}$) are asserted during memory write cycles for the upper and lower bytes of the 16-bit data bus. They may be directly connected to the write enable inputs of a typical memory device.

2.2 Chip-Select Module

The MC68VZ328 can generate up to 8 chip select outputs which are organized into four groups (A through D).

Each chip select group has a common base address register and address mask register allowing the base address and block size of the entire group to be set. In addition, each chip select within a group has its own address compare and address mask register to activate the chip select for a subset of the group’s address block. Each chip select may also be individually programmed to control an 8 or 16-bit device. Lastly, each chip select can either generate from 0 through 6 wait states internally, or allow the memory or peripheral device to terminate the cycle externally using the standard MC68000 $\overline{DTACK}$ signal.

Chip select groups A and B are used to control ROM, SRAM, and Flash memory devices and have a block size of 128K bytes to 16M bytes. Chip select A0 is active immediately after reset and is a global chip select so it is typically used to control a boot EPROM device. A0 ceases to decode globally once its chip select registers are programmed. Groups C and D are special in that they can also control DRAM interfaces. These last two groups have block size of 32K bytes to 4M bytes.

3 S1D13706 Host Bus Interface

The S1D13706 directly supports multiple processors. The S1D13706 implements a Dragonball Host Bus Interface which directly supports the Motorola MC68VZ328 microprocessor.

The Dragonball Host Bus Interface is selected by the S1D13706 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13706 configuration, see Section 4.2, “S1D13706 Hardware Configuration” on page 12.

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

S1D13706 Pin Names	Motorola MC68VZ328
AB[16:0]	A[16:0]
DB[15:0]	D[15:0]
WE1#	$\overline{\text{UWE}}$
CS#	$\overline{\text{CSx}}$
M/R#	External Decode
CLKI	CLKO
BS#	Connect to HIO_{VDD} from the S1D13706
RD/WR#	Connect to HIO_{VDD} from the S1D13706
RD#	$\overline{\text{OE}}$
WE0#	$\overline{\text{LWE}}$
WAIT#	$\overline{\text{DTACK}}$
RESET#	System $\overline{\text{RESET}}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals.

- CLKI is a clock input required by the S1D13706 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, CLK0 from the Motorola MC68VZ328 is used for CLKI.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the MC68VZ328 address (A[16:0]) and data bus (D[15:0]), respectively. CNF4 must be set to one to select big endian mode.
- Chip Select (CS#) must be driven low by one of the Dragonball VZ chip select outputs from the chip select module whenever the S1D13706 is accessed by the MC68VZ328.
- M/R# (memory/register) selects between memory or register accesses. This signal is generated by the external address decode circuitry. For this example, M/R# may be connected to an address line, allowing system address A17 to be connected to the M/R# line.
- WE0# connects to $\overline{\text{LWE}}$ (the low data byte write strobe enable of the MC68VZ328) and is asserted when valid data is written to the low byte of a 16-bit device.
- WE1# connects to $\overline{\text{UWE}}$ (the upper data byte write strobe enable of the MC68VZ328) and is asserted when valid data is written to the high byte of a 16-bit device.
- RD# connects to $\overline{\text{OE}}$ (the read output enable of the MC68VZ328) and is asserted during a read cycle of the MC68VZ328 microprocessor.
- RD/WR# is not used for the Dragonball host bus interface and must be tied high to HIO V_{DD}.
- WAIT# connects to $\overline{\text{DTACK}}$ and is a signal which is output from the S1D13706 indicating the MC68VZ328 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. The MC68VZ328 accesses to the S1D13706 may occur asynchronously to the display update.
- BS# is not used for the Dragonball host bus interface and must be tied high to HIO V_{DD}.

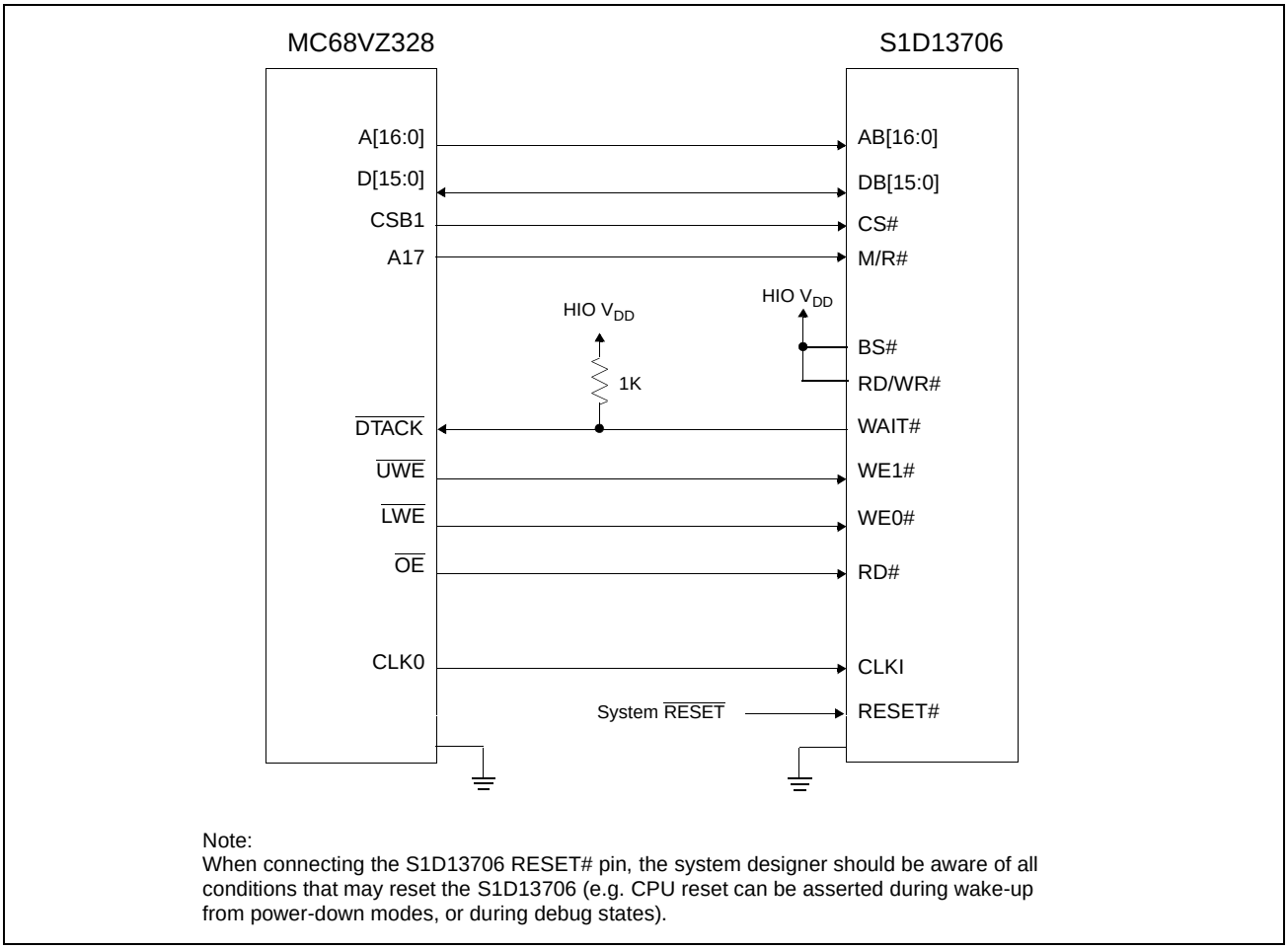
4 MC68VZ328 to S1D13706 Interface

4.1 Hardware Description

The interface between the S1D13706 and the MC68VZ328 does not requires any external glue logic. Chip select module B is used to provide the S1D13706 with a chip select and A17 is used to select between memory and register accesses.

In this example, the $\overline{\text{DTACK}}$ signal is made available for the S1D13706. Alternately, the S1D13706 can guarantee a maximum cycle length that the Dragonball VZ handles by inserting software wait states (see Section 4.2.2, "MC68VZ328 Chip Select and Pin Configuration" on page 13). A single resistor is used to speed up the rise time of the WAIT# ($\overline{\text{DTACK}}$) signal when terminating the bus cycle.

The following diagram shows a typical implementation of the MC68VZ328 to S1D13706 using the Dragonball host bus interface. For further information on the Dragonball Host Bus interface and AC Timing, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.



Note:
When connecting the S1D13706 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13706 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 4-1: Typical Implementation of MC68VZ328 to S1D13706 Interface

4.2 S1D13706 Hardware Configuration

The S1D13706 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13706 to Motorola MC68VZ328 microprocessor.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13706 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[2:0]	110 = Dragonball Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF4	Big Endian bus interface	Little Endian bus interface
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-2: “CLKI to BCLK Divide Selection” for recommended settings	

= configuration for MC68VZ328 microprocessor

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for MC68VZ328 microprocessor

4.2.1 Register/Memory Mapping

The S1D13706 requires two 128K byte segments in memory for the display buffer and its internal registers. To accommodate this block size, it is preferable (but not required) to use one of the chip selects from groups A or B. Groups A and B can have a size range of 128K bytes to 16M bytes and groups C and D have a size range of 32K bytes to 16M bytes. Therefore, any chip select other than CSA0 would be suitable for the S1D13706 interface.

In the example interface, chip select CSB1 controls the S1D13706. A 256K byte address space is used with the S1D13706 internal registers occupying the first 128K byte block and the 80K byte display buffer located in the second 128K byte block. A17 from the MC68VZ328 is used to select between these two 128K byte blocks.

4.2.2 MC68VZ328 Chip Select and Pin Configuration

The chip select used to map the S1D13706 (in this example CSB1) must have its RO (Read Only) bit set to 0, its BSW (Bus Data Width) set to 1 for a 16-bit bus, and the WS (Wait states) bits should be set to 111b to allow the S1D13706 to terminate bus cycles externally with $\overline{\text{DTACK}}$. The $\overline{\text{DTACK}}$ pin function must be enabled with Register FFFFF433, Port G Select Register, bit 0.

If $\overline{\text{DTACK}}$ is not used, then the the WS bits should be set to either 4, 6, 10, or 12 software wait states depending on the divide ratio between the S1D13706 MCLK and BCLK. The WS bits should be set as follows.

Table 4-3: WS Bit Programming

S1D13706 MCLK to BCLK Divide Ratio	WS Bits (wait states)
MCLK = BCLK	4
MCLK = BCLK ÷ 2	6
MCLK = BCLK ÷ 3	10
MCLK = BCLK ÷ 4	12

5 Software

Test utilities and Windows® CE display drivers are available for the S1D13706. 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 13706CFG, or by directly modifying the source. The Windows CE display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13706 test utilities and Windows CE display drivers are available from your sales support contact or on the internet at http://www.eea.epson.com.

6 References

6.1 Documents

- Motorola Inc., *MC68VZ328 DragonBall-VZ® Integrated Processor User's Manual*, Motorola Publication no. MC683VZ28UM; available on the Internet at http://www.mot.com/SPS/WIRELESS/products/MC68VZ328.html.
- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification*, Document Number X31B-A-001-xx.
- Epson Research and Development, Inc., *S5U13706B00C Rev. 1.0 Evaluation Board User Manual*, Document Number X31B-G-004-xx.
- Epson Research and Development, Inc., *Programming Notes and Examples*, Document Number X31B-G-003-xx.

6.2 Document Sources

- Motorola Inc. Literature Distribution Center: (800) 441-2447.
- Motorola Inc. Website: http://www.mot.com.
- Epson Electronics America website: http://www.eea.epson.com.

7 Technical Support

7.1 EPSON LCD/CRT Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Taiwan
Epson Taiwan Technology & Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716

7.2 Motorola MC68VZ328 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

EPSON®

ENERGY SAVING EPSON

S1D13706 Embedded Memory LCD Controller

Integrating the CFLGA 104-pin Chip Scale Package

Document Number: X31B-G-018-02

THIS PAGE LEFT BLANK

Table of Contents

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This manual provides an example for integrating the CFLGA 104-pin chip scale package (CSP) available for the S1D13706. It includes an overview of the package and provides an example of how to route the pads.

This application note is updated as appropriate. Please check the Epson Electronics America website at www.eea.epson.com or the Epson Research and Development website at www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Package Description

Designing a Chip Scale Package part (i.e. S1D13706) into a printed circuit board requires the use of microvia technology. Before starting development of a PCB, consult with the board manufacturer for information about the particular microvia technology they use.

Microvias are commonly defined as vias that have holes less than 0.15mm (0.006") in diameter. Microvia technology typically uses the layer located just below the outermost layer. Microvias are blind vias that go down only one layer and connect the outer layer with the microvia specific layer. The traces on microvia specific layers are connected to the other layers of the board by standard vias.

The S1D13706 CSP has the following characteristics.

- reinforced land type footprint.
- 4 reinforced pads, land size 1.05mm (.042") in diameter.
- 104 pads.
- land size 0.3mm (0.012") in diameter.
- distributed on a 11 x 11 grid with a 0.65mm (0.025") pitch.
- solder mask 0.43mm (0.017") in diameter.

Note

The reinforcement pads located in the corner of the footprint, provide extra mechanical strength once the chip has been mounted.

For pinout diagrams and mechanical drawings, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

3 Routing

3.1 Perimeter Pads

Perimeter pads of the S1D13706 CSP are usually fanned out on the top layer using 0.004" traces with 0.0045" spaces at the passage between pads. The traces are terminated using standard via technology (i.e. 0.025" via with 0.012" hole).

The following diagram shows an example for perimeter pad routing.

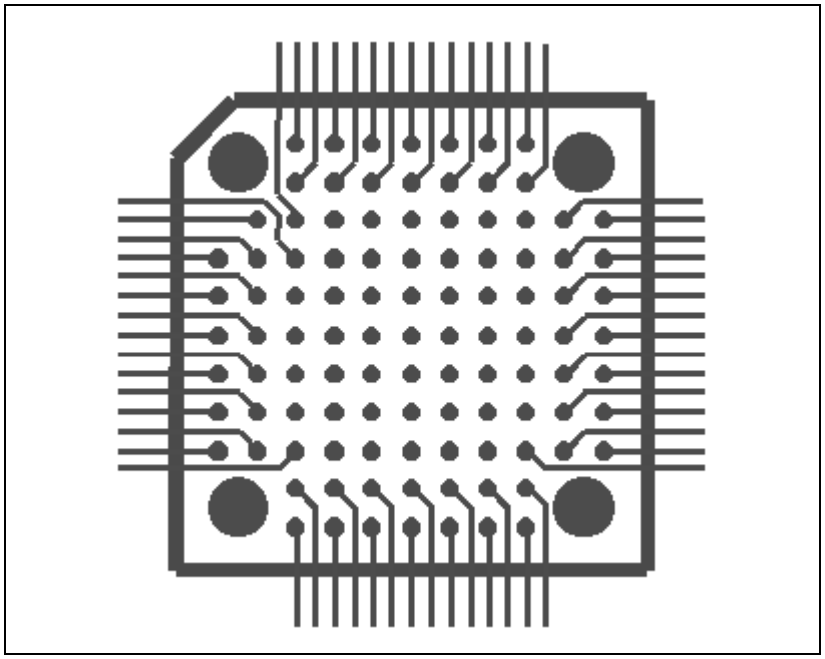


Figure 3-1: Example Perimeter Pad Routing

3.2 Inner Pads

The inner pads on top layer require microvias connecting them with the microvia specific layer located just below the top layer. The pads on the microvia specific layer have a land size of 0.254mm (0.010") in diameter and are fanned out with 0.005" traces with 0.005" spaces at the passage between pads.

All the Vss pins are inner pins and require connection with the microvia specific layer. On this layer, all the Vss pads are connected together and are fanned out with multiple traces.

All the traces on the microvia specific layer must be terminated to a standard through-hole via for connection to the rest of the board (i.e. bottom layer, power and ground planes).

The following diagram shows an example for inner pad routing.

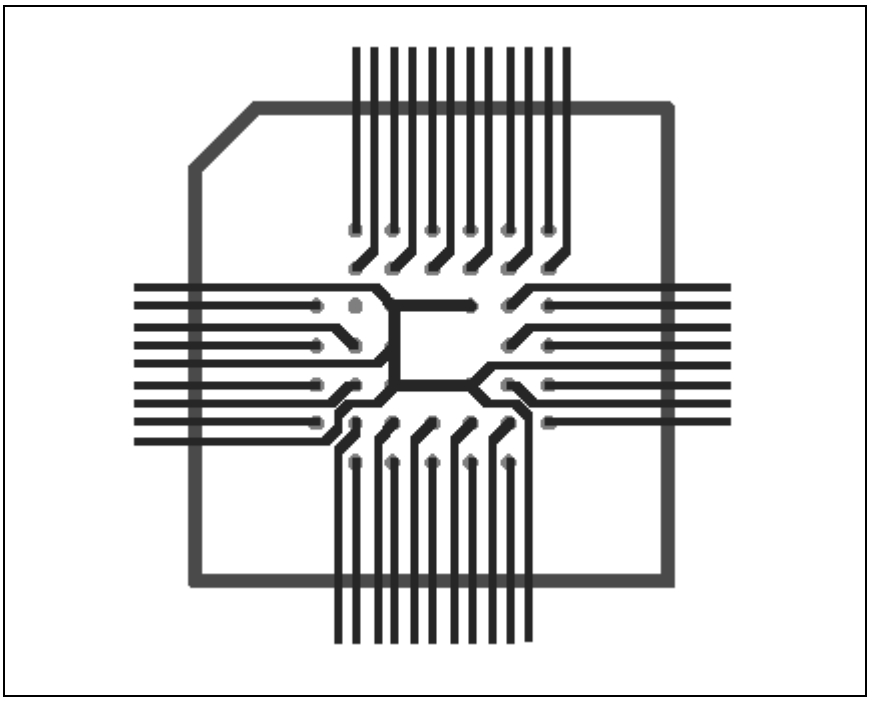


Figure 3-2: Example Inner Pad Routing

4 References

4.1 Documents

- Epson Research and Development, Inc., *S1D13706 Hardware Functional Specification,* Document Number X31B-A-001-xx.

4.2 Document Sources

- Epson Electronics America website: http://www.eea.epson.com.
- Epson Research and Development website: http://www.erd.epson.com.

5 Technical Support

5.1 EPSON LCD Controllers (S1D13706)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de

Taiwan
Epson Taiwan Technology & Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg