EPSON®

ENERGY SAVING EPSON

S1D13704 Embedded Memory Color LCD Controller

S1D13704
TECHNICAL MANUAL

Issue Date: 01/02/12

Document Number: X26A-Q-001-04

THIS PAGE LEFT BLANK

Customer Support Information

Comprehensive Support Tools

Seiko Epson Corp. provides to the system designer and computer OEM manufacturer a complete set of resources and tools for the development of graphics systems.

Evaluation / Demonstration Board

- Assembled and fully tested graphics evaluation board with installation guide and schematics.
- To borrow an evaluation board, please contact your local Seiko Epson Corp. sales representative.

Chip Documentation

- Technical manual includes Data Sheet, Application Notes, and Programmer's Reference.

Software

- OEM Utilities.
- User Utilities.
- Evaluation Software.
- To obtain these programs, contact Application Engineering Support.

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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

Table of Contents

THIS PAGE LEFT BLANK

February 2001

S1D13704 EMBEDDED MEMORY COLOR LCD CONTROLLER

DESCRIPTION

The S1D13704 is a color/monochrome LCD graphics controller with an embedded 40K Byte SRAM display buffer. The high integration of the S1D13704 provides a low cost, low power, single chip solution to meet the requirements of embedded markets such as Office Automation equipment, Mobile Communications devices, and Palm-size PCs where board size and battery life are major concerns.

Products requiring a “Portrait” display can take advantage of the Hardware Portrait Mode feature of the S1D13704. Virtual and Split Screen are just some of the display modes supported. The above features, combined with the Operating System independence of the S1D13704, make it the ideal solution for a wide variety of applications.

FEATURES

Memory Interface

- Embedded 40K byte SRAM display buffer.

CPU Interface

- Direct support of the following interfaces:
  - Hitachi SH-3.
  - Hitachi SH-4.
  - Motorola M68K.
  - MPU bus interface with programmable READY.
- Direct memory mapping of internal registers.
- CPU write buffer.

Display Support

- 4/8-bit monochrome LCD interface.
- 4/8-bit color LCD interface.
- Single-panel, single-drive passive displays.
- Dual-panel, dual-drive passive displays.
- Active Matrix TFT / TFD interface.
- Register level suport for EL panels.
- Example resolutions:
  - 640x480 at a color depth of 1 bpp
  - 640x240 at a color depth of 2 bpp
  - 320x240 at a color depth of 4 bpp
  - 240x160 at a color depth of 8 bpp

Power Down Modes

- Hardware and software Suspend modes.
- LCD power-down sequencing.

Display Modes

- Hardware Portrait Mode: direct hardware rotation of display image for portrait mode display.
- 1/2/4 bit-per-pixel (bpp), 2/4/16-level grayscale display.
- 1/2/4/8 bit-per-pixel, 2/4/16/256-level color display.
- Up to 16 shades of gray by FRM on monochrome passive LCD panels.
- 256 simultaneous of 4096 colors on color passive and active matrix LCD panels.
- Split screen display for all panel modes allows two different images to be simultaneously displayed.
- Virtual display support (displays images larger than the panel size through the use of panning).

Clock Source

- Single clock input for both pixel and memory clocks.
- The S1D13704 clock source can be internally divided down for a higher frequency clock input.
- Dynamic switching of memory clocks in portrait mode.

General Purpose IO Pins

- Five General Purpose Input / Output pins available.

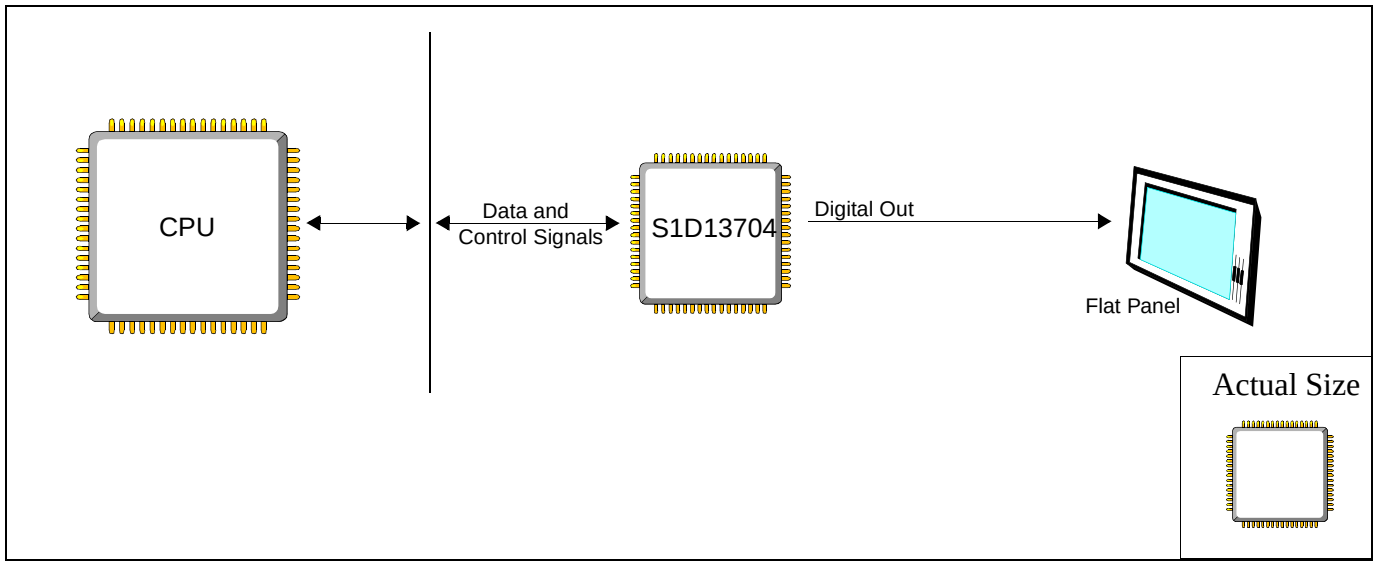
Operating Voltage

- 2.7 volts to 5.5 volts.

Package

- 80-pin QFP14 surface mount package.

■ SYSTEM BLOCK DIAGRAM



CONTACT YOUR SALES REPRESENTATIVE FOR THESE COMPREHENSIVE DESIGN TOOLS:

- S1D13704 Technical Manual
- S5U13704 Evaluation Boards
- Windows® CE Display Driver
- CPU Independent Software Utilities

FOR SYSTEM INTEGRATION SERVICES FOR WINDOWS® CE CONTACT:
Epson Research & Development, Inc.
Suite #320 - 11120 Horseshoe Way
Richmond, B.C., Canada V7A 5H7
Tel: (604) 275-5151
Fax: (604) 275-2167
Email: wince@erd.epson.com
http://www.erd.epson.com

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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

EPSON®

S1D13704 Embedded Memory LCD Controller

Hardware Functional Specification

Document Number: X26A-A-001-04

THIS PAGE LEFT BLANK

Table of Contents

List of Tables

THIS PAGE LEFT BLANK

List of Figures

1 Introduction

1.1 Scope

This is the Functional Specification for the S1D13704 Embedded Memory LCD Controller Chip. 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.

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.

1.2 Overview Description

The S1D13704 is a color / monochrome LCD graphics controller with an embedded 40K Byte SRAM display buffer. The high integration of the S1D13704 provides a low cost, low power, single chip solution to meet the requirements of embedded markets such as Office Automation equipment, Mobile Communications devices, and Hand-Held PCs where board size and battery life are major concerns.

Products requiring a “Portrait” display can take advantage of the Swivelview™ (90° Hardware Rotate) feature of the S1D13704. Virtual and Split Screen are just some of the display modes supported. The above features, combined with the Operating System independence of the S1D13704, make it the ideal solution for a wide variety of applications.

2 Features

2.1 Integrated Frame Buffer

- Embedded 40K byte SRAM display buffer.

2.2 CPU Interface

- Direct support of the following interfaces:
  Hitachi SH-3.
  Hitachi SH-4.
  Motorola M68K.
  MPU bus interface using WAIT# signal.
- Direct memory mapping of internal registers.
- Single level CPU write buffer.
- Registers are mapped into upper 32 bytes of 64K byte address space.
- The complete 40K byte frame buffer is directly and contiguously available through the 16-bit address bus.

2.3 Display Support

- 4/8-bit monochrome LCD interface.
- 4/8-bit color LCD interface.
- Single-panel, single-drive passive displays.
- Dual-panel, dual-drive passive displays.
- Active Matrix TFT / D-TFD interface
- Register level support for EL panels.
- Example resolutions:
  640x480 at a color depth of 1 bpp
  640x240 at a color depth of 2 bpp
  320x240 at a color depth of 4 bpp
  240x160 at a color depth of 8 bpp

2.4 Display Modes

- SwivelView™: direct 90° hardware rotation of display image for portrait mode display.
- 1/2/4 bit-per-pixel (bpp), 2/4/16-level grayshade display.
- 1/2/4/8 bit-per-pixel, 2/4/16/256-level color display.
- Up to 16 shades of gray by FRM on monochrome passive LCD panels; a 16x4 Look-Up-Table is used to map 1/2/4-bpp modes into these shades.
- 256 simultaneous of 4096 colors on color passive and active matrix LCD panels; three 16x4 Look-Up Tables are used to map 1/2/4/8-bpp modes into these colors.
- Split screen display for all landscape panel modes allows two different images to be simultaneously displayed.
- Virtual display support (displays images larger than the panel size through the use of panning).

2.5 Clock Source

- Maximum operating clock (CLK) frequency of 25MHz.
- Operating clock (CLK) is derived from CLKI input.
  CLK = CLKI
  or
  CLK = CLKI/2
- Pixel Clock (PCLK) and Memory Clock (MCLK) are derived from CLK.

2.6 Miscellaneous

- Hardware/Software Video Invert.
- Software Power Save mode.
- Hardware Power Save mode.
- LCD power-down sequencing.
- 5 General Purpose Input/Output pins are available.
  - GPIO0 is available if Hardware Power Save is not required.
  - GPIO[4:1] are available if upper LCD data pins (FPDAT[11:8]) are not required for TFT/D-TFD support or Hardware Video Invert.
- IO Operates from 3.0 volts to 5.5 volts
- Core operates from 3.0 volts to 3.6 volts.

2.7 Package

- 80 pin QFP14 package.

3 Typical System Implementation Diagrams

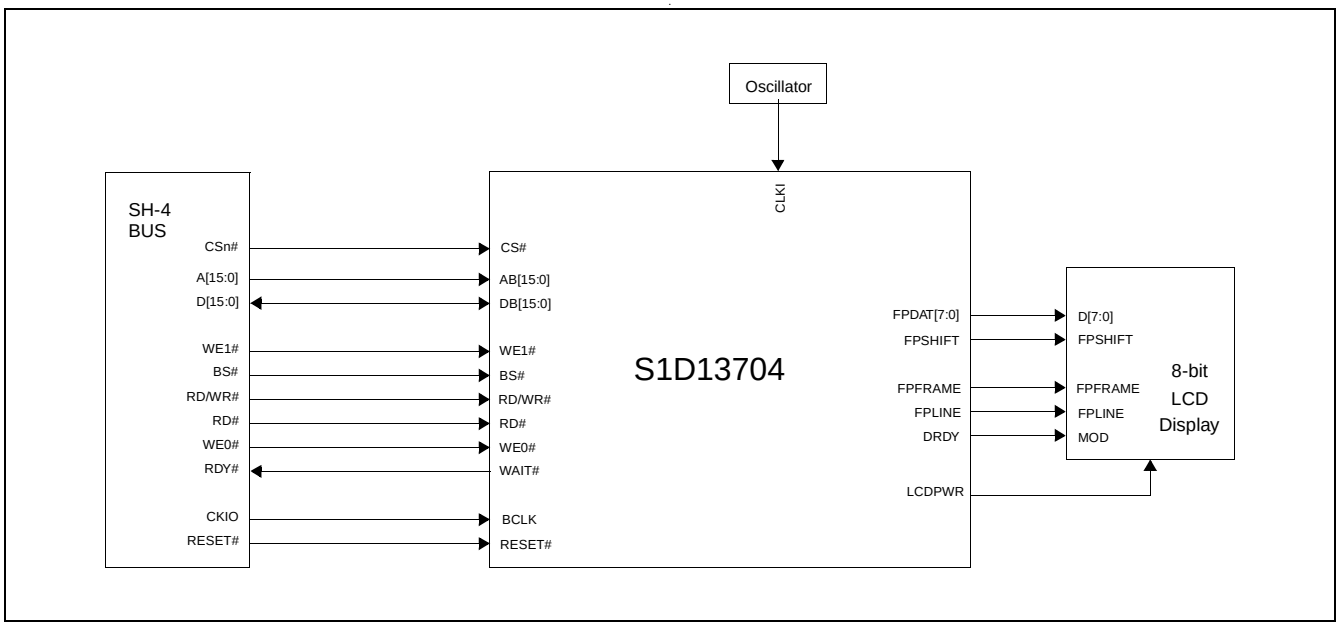


Figure 3-1: Typical System Diagram (SH-4 Bus)

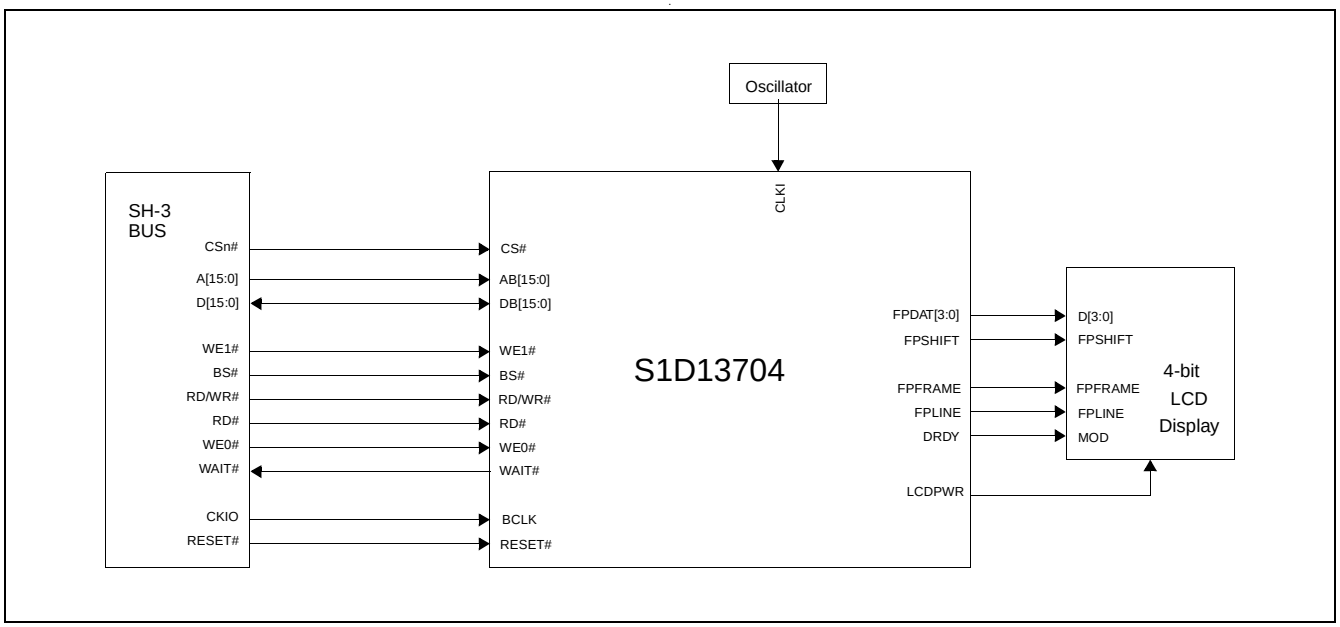


Figure 3-2: Typical System Diagram (SH-3 Bus)

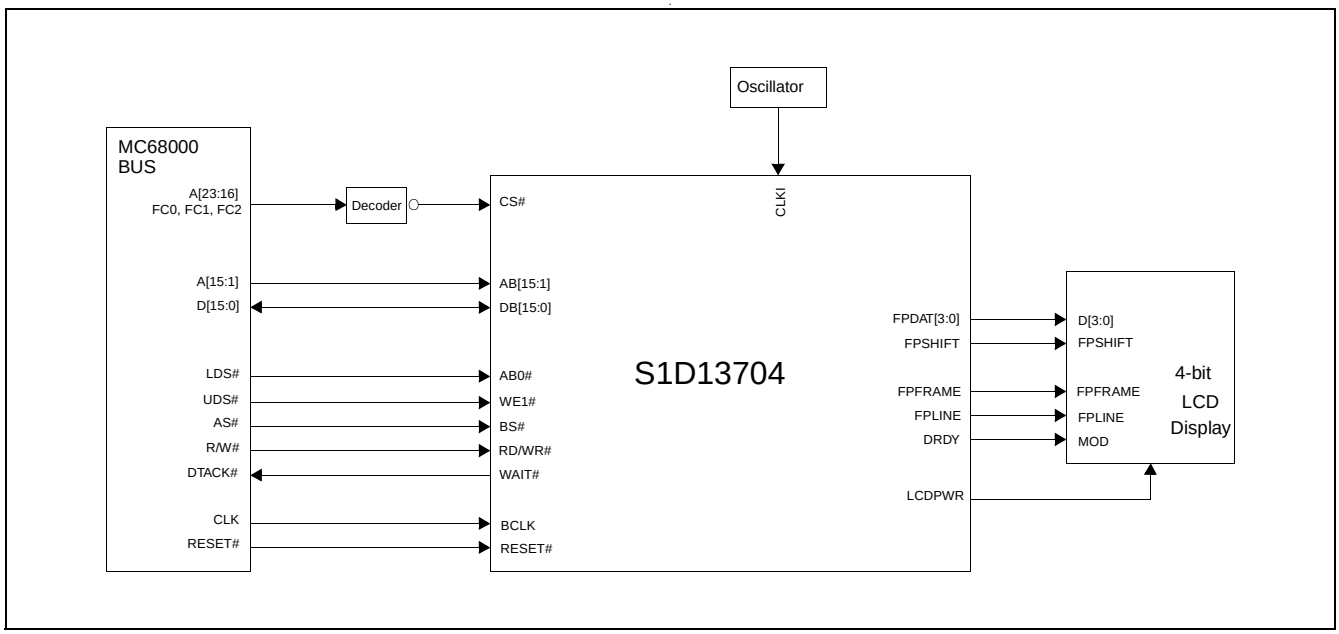


Figure 3-3: Typical System Diagram (M68K #1 Bus)

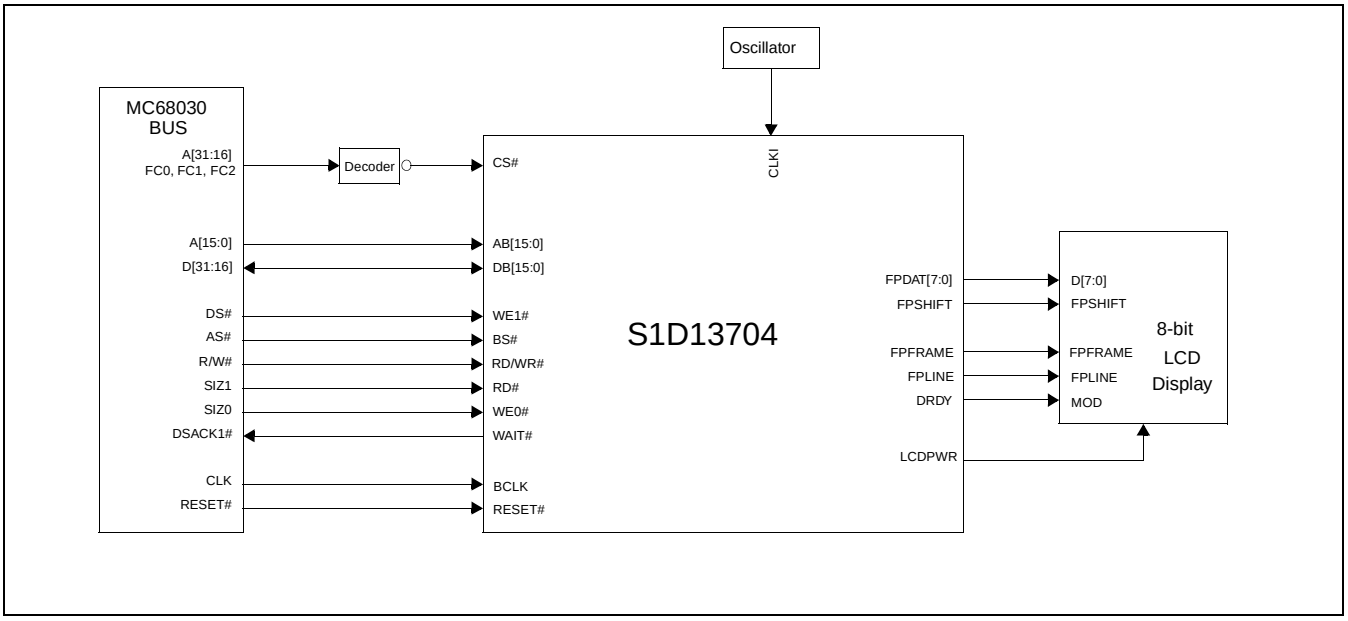


Figure 3-4: Typical System Diagram (M68K #2 Bus)

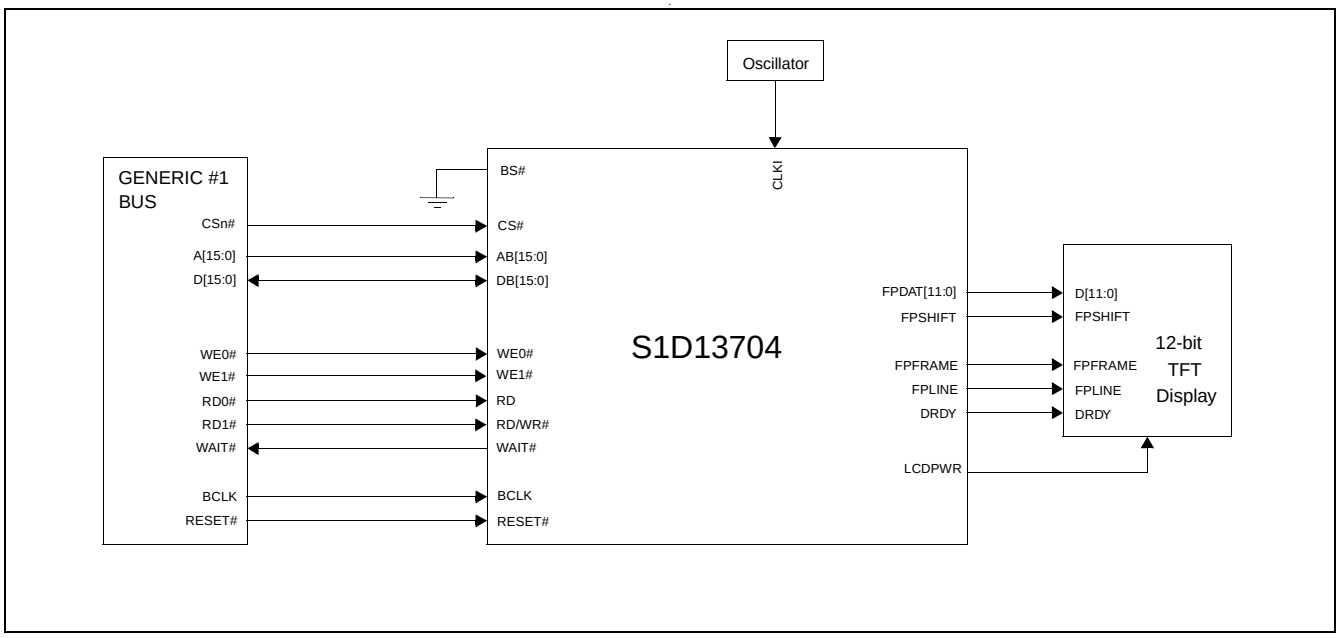


Figure 3-5: Typical System Diagram (Generic #1 Bus)

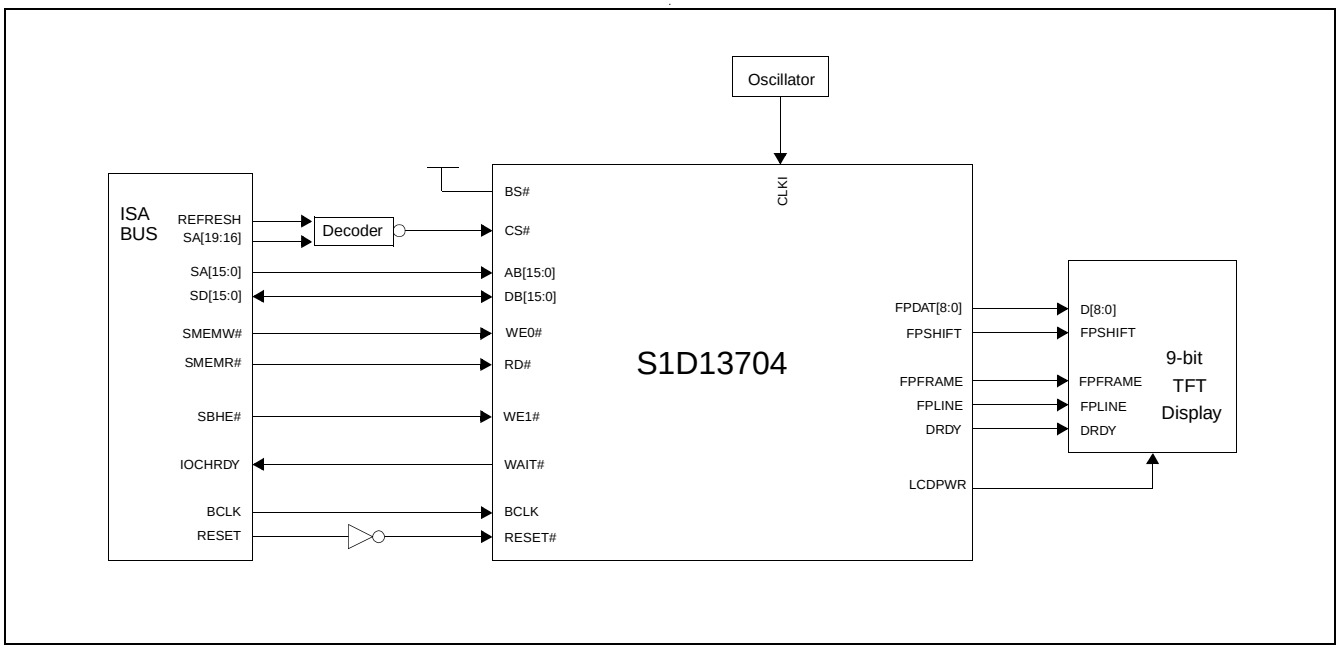


Figure 3-6: Typical System Diagram (Generic #2 Bus - e.g. ISA Bus)

4 Functional Block Diagram

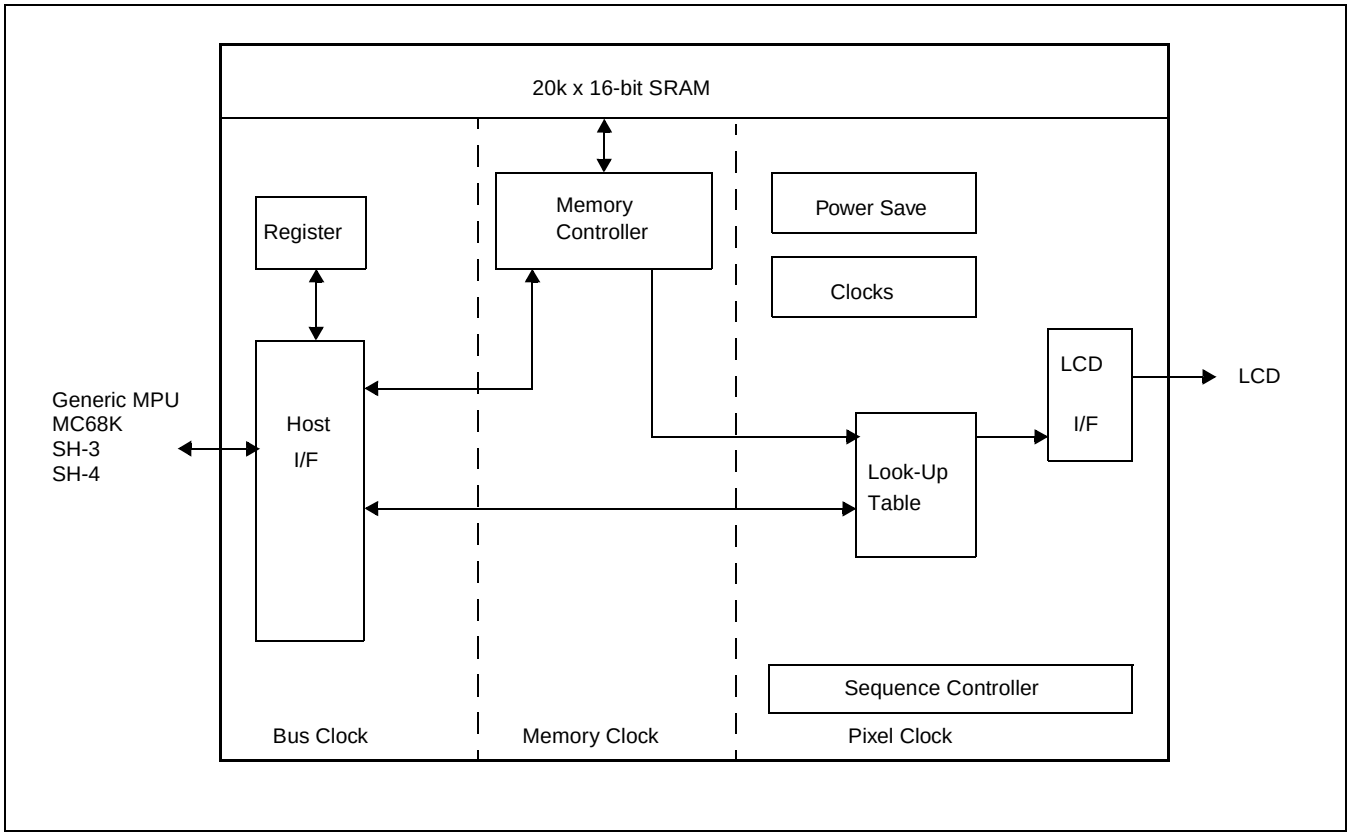


Figure 4-1: System Block Diagram Showing Data Paths

4.1 Functional Block Descriptions

4.1.1 Host Interface

The Host Interface provides the means for the CPU/MPU to communicate with the display memory and internal registers.

4.1.2 Memory Controller

The Memory Controller arbitrates between CPU accesses and display refresh accesses. It also generates the necessary signals to control the SRAM frame buffer.

4.1.3 Sequence Controller

The Sequence Controller controls data flow from the Memory Controller through the Look-Up Table and to the LCD Interface. It also generates memory addresses for display refresh accesses.

4.1.4 Look-Up Table

The Look-Up Table contains three 16x4 Look-Up Tables or palettes, one for each primary color. In monochrome mode only one of these Look-Up Tables is used.

4.1.5 LCD Interface

The LCD Interface performs frame rate modulation for passive LCD panels. It also generates the correct data format and timing control signals for various LCD and TFT/D-TFD panels.

4.1.6 Power Save

Power Save contains the power save mode circuitry.

5 Pins

5.1 Pinout Diagram

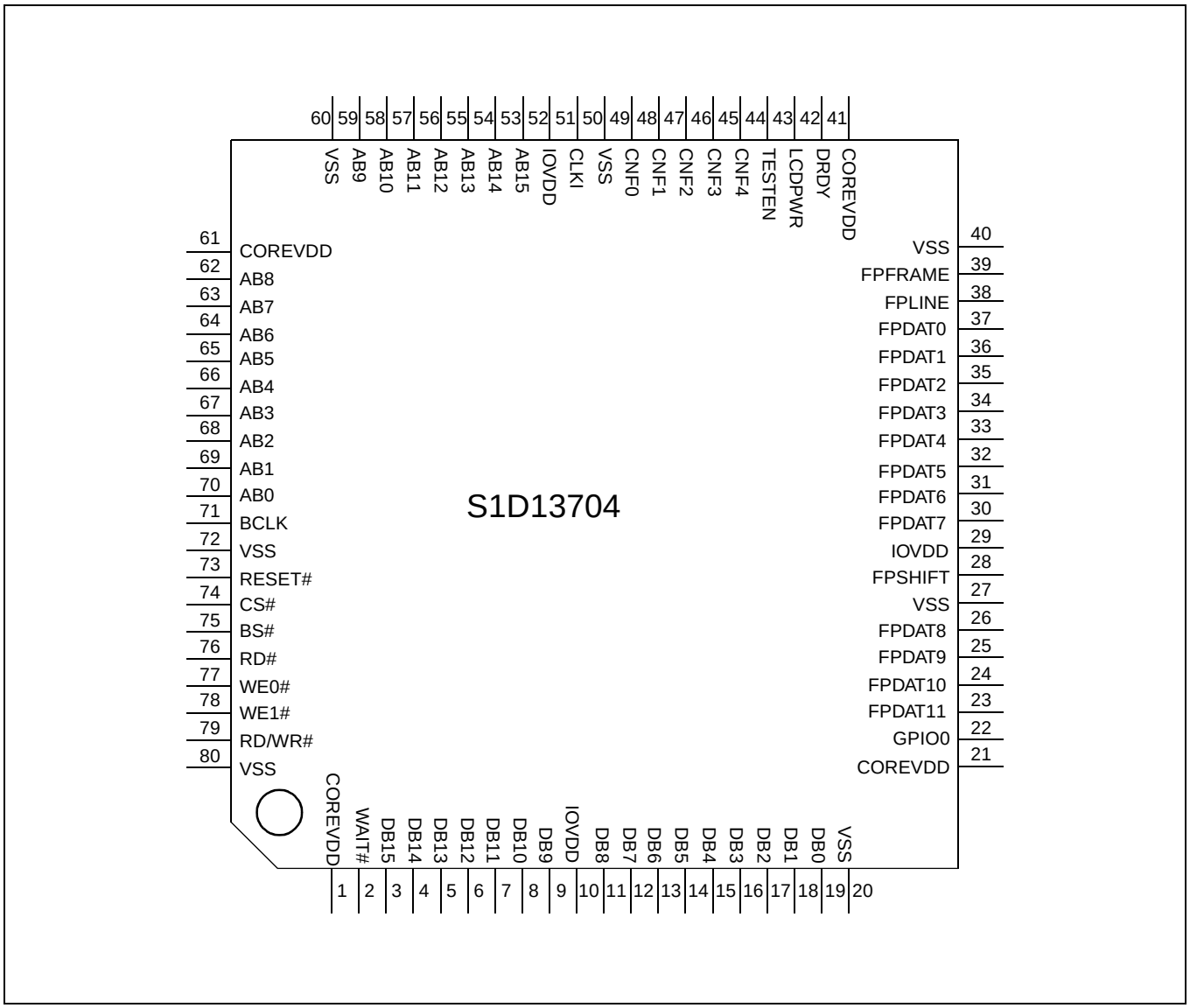


Figure 5-1: Pinout Diagram

Note
Package type: 80 pin surface mount QFP14

5.2 Pin Description

Key:

I	=	Input
O	=	Output
I/O	=	Bi-Directional (Input/Output)
P	=	Power pin
C	=	CMOS level input
CD	=	CMOS level input with pull down resistor (typical values of 100KΩ/180KΩ at 5V/3.3V respectively)
CS	=	CMOS level Schmitt input
COx	=	CMOS output driver, x denotes driver type (1=3/-1.5mA, 2=6/-3mA, 3=12/-6mA)
TSx	=	Tri-state CMOS output driver, x denotes driver type (1=3/-1.5mA, 2=6/-3mA, 3=12/-6mA)
TSxD	=	Tri-state CMOS output driver with pull down resistor (typical values of 100KΩ/180KΩ at 5V/3.3V respectively), x denotes driver type (1=3/-1.5mA, 2=6/-3mA, 3=12/-6mA)
CNx	=	CMOS low-noise output driver, x denotes driver type (1=3/-1.5mA, 2=6/-3mA, 3=12/-6mA)

5.2.1 Host Interface

Pin Names	Type	Pin #	Cell	RESET# State	Description
AB0	I	70	CS	Input	This pin has multiple functions. • For SH-3/SH-4 mode, this pin inputs system address bit 0 (A0). • For MC68K #1, this pin inputs the lower data strobe (LDS#). • For MC68K #2, this pin inputs system address bit 0 (A0). • For Generic #1, this pin inputs system address bit 0 (A0). • For Generic #2, this pin inputs system address bit 0 (A0). See *"Host Bus Interface Pin Mapping"* for summary.
AB[15:1]	I	53, 54, 55, 56, 57, 58, 59, 62, 63, 64, 65, 66, 67, 68, 69	C	Input	These pins input the system address bits 15 through 1 (A[15:1]).
DB[15:0]	I/O	3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19	C/TS2	High Impedance	These pins have multiple functions. • For SH-3/SH-4 mode, these pins are connected to [D15:0]. • 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 Generic #1, these pins are connected to D[15:0]. • For Generic #2, these pins are connected to D[15:0]. See *"Host Bus Interface Pin Mapping"* for summary.

Pin Names	Type	Pin #	Cell	RESET# State	Description
WE0#	I	77	CS	Input	This pin has multiple functions. • For SH-3/SH-4 mode, this pin inputs the write enable signal for the lower data byte (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 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#) See *"Host Bus Interface Pin Mapping"* for summary.
WE1#	I	78	CS	Input	This pin has multiple functions. • For SH-3/SH-4 mode, this pin inputs the write enable signal for the upper data byte (WE1#). • For MC68K #1, this pin inputs the upper data strobe (UDS#). • For MC68K #2, this pin inputs the data strobe (DS#). • 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#). See *"Host Bus Interface Pin Mapping"* for summary.
CS#	I	74	C	Input	This pin inputs the chip select signal.
BCLK	I	71	C	Input	This pin inputs the system bus clock.
BS#	I	75	CS	Input	This pin has multiple functions. • For SH-3/SH-4 mode, 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 Generic #1, this pin must be tied to V_{SS}. • For Generic #2, this pin must be tied to IO V_{DD}. See *"Host Bus Interface Pin Mapping"* for summary.
RD/WR#	I	79	CS	Input	This pin has multiple functions. • For SH-3/SH-4 mode, this pin inputs the RD/WR# signal. The S1D13704 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 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}. See *"Host Bus Interface Pin Mapping"* for summary.

Pin Names	Type	Pin #	Cell	RESET# State	Description
RD#	I	76	CS	Input	This pin has multiple functions. • For SH-3/SH-4 mode, 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 Generic #1, this pin inputs the read command for the lower data byte (RD0#). • For Generic #2, this pin inputs the read command (RD#). See *"Host Bus Interface Pin Mapping"* for summary.
WAIT#	O	2	TS2	High Impedance	This pin has multiple functions. • 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 Generic #1, this pin outputs the wait signal (WAIT#). • For Generic #2, this pin outputs the wait signal (WAIT#). See *"Host Bus Interface Pin Mapping"* for summary.
RESET#	I	73	CS	0	Active low input to set all internal registers to the default state and to force all signals to their inactive states.

5.2.2 LCD Interface

Pin Name	Type	Pin #	Cell	RESET# State	Description
FPDAT[7:0]	O	30, 31, 32, 33, 34, 35, 36, 37	CN3	0	Panel Data
FPDAT[10:8]	O, I/O	24, 25, 26	CN3	Input	These pins have multiple functions. • Panel Data bits [10:8] for TFT/D-TFD panels. • General Purpose Input/Output pins GPIO[3:1]. These pins should be connected to IO V_{DD} when unused. See *"LCD Interface Pin Mapping"* for summary.
FPDAT11	O, I/O	23	CN3	Input	This pin has multiple functions. • Panel Data bit 11 for TFT/D-TFD panels. • General Purpose Input/Output pin GPIO4. • Inverse Video select pin. This pin should be connected to IO V_{DD} when unused. See *"LCD Interface Pin Mapping"* for summary.
FPFRAME	O	39	CN3	0	Frame Pulse

Pin Name	Type	Pin #	Cell	RESET# State	Description
FPLINE	O	38	CN3	0	Line Pulse
FPSHIFT	O	28	CN3	0	Shift Clock
LCDPWR	O	43	CO1	0 if CNF4 = 1 1 if CNF4 = 0	LCD Power Control
DRDY	O	42	CN3	0	This pin has multiple functions. • TFT/D-TFD Display Enable (DRDY). • LCD Backplane Bias (MOD). • Second Shift Clock (FPSHIFT2). See *"LCD Interface Pin Mapping"* for summary.

5.2.3 Clock Input

Pin Name	Type	Pin #	Driver	Description
CLKI	I	51	C	Input Clock

5.2.4 Miscellaneous

Pin Name	Type	Pin #	Cell	RESET# State	Description
CNF[4:0]	I	45, 46, 47, 48, 49	C	As set by hardware	These inputs are used to configure the S1D13704 - see *"Summary of Configuration Options"*. Must be connected directly to IO V_{DD} or V_{SS}.
GPIO0	I/O, I	22	CS/ TS1	Input	This pin has multiple functions - see REG[03h] bit 2. • General Purpose Input/Output pin. • Hardware Power Save.
TESTEN	I	44	CD	High Impedance	Test Enable input. This input must be connected to V_{SS}.

5.2.5 Power Supply

Pin Name	Type	Pin #	Driver	Description
COREVDD	P	1, 21, 41, 61	P	Core V_{DD}
IOVDD	P	10, 29, 52	P	IO V_{DD}
VSS	P	20, 27, 40, 50, 60, 72, 80	P	Common V_{SS}

5.3 Summary of Configuration Options

Table 5-1: Summary of Power On/Reset Options

Configuration Pin	Power On/Reset State	
	1	**0**
CNF4	Active high (On) LCDPWR polarity	Active low (On) LCDPWR polarity
CNF3	Big Endian	Little Endian
CNF[2:0]	Select host bus interface as follows: **CNF2 CNF1 CNF0 BS# Host Bus** 0 0 0 X SH-4 interface 0 0 1 X SH-3 interface 0 1 0 X reserved 0 1 1 X MC68K #1, 16-bit 1 0 0 X reserved 1 0 1 X MC68K #2, 16-bit 1 1 0 0 reserved 1 1 0 1 reserved 1 1 1 0 Generic #1, 16-bit 1 1 1 1 Generic #2, 16-bit	

5.4 Host Bus Interface Pin Mapping

Table 5-2: Host Bus Interface Pin Mapping

S1D13704 Pin Names	SH-3	SH-4	MC68K #1	MC68K #2	Generic #1	Generic #2
AB[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]
AB0	A0	A0	LDS#	A0	A0	A0
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[31:16]	D[15:0]	D[15:0]
WE1#	WE1#	WE1#	UDS#	DS#	WE1#	BHE#
CS#	CSn#	CSn#	External Decode	External Decode	External Decode	External Decode
BCLK	CKIO	CKIO	CLK	CLK	BCLK	BCLK
BS#	BS#	BS#	AS#	AS#	connect to V_{SS}	connect to IO V_{DD}
RD/WR#	RD/WR#	RD/WR#	R/W#	R/W#	RD1#	connect to IO V_{DD}
RD#	RD#	RD#	connect to IO V_{DD}	SIZ1	RD0#	RD#
WE0#	WE0#	WE0#	connect to IO V_{DD}	SIZ0	WE0#	WE#
WAIT#	WAIT#	RDY#	DTACK#	DSACK1#	WAIT#	WAIT#
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	RESET#

5.5 LCD Interface Pin Mapping

Table 5-3: LCD Interface Pin Mapping

S1D13704 Pin Name	Monochrome Passive Panel			Color Passive Panel				Color TFT/D-TFD	
	4-bit Single	8-bit Single	8-bit Dual	4-bit Single	8-bit Single Format 1	8-bit Single Format 2	8-bit Dual	9-bit	12-bit
FPFRAME	FPFRAME								
FPLINE	FPLINE								
FPSHIFT	FPSHIFT								
DRDY	MOD	MOD	MOD	MOD	FPSHIFT2	MOD	MOD	DRDY	
FPDAT0	driven 0	D0	LD0	driven 0	D0	D0	LD0	R2	R3
FPDAT1	driven 0	D1	LD1	driven 0	D1	D1	LD1	R1	R2
FPDAT2	driven 0	D2	LD2	driven 0	D2	D2	LD2	R0	R1
FPDAT3	driven 0	D3	LD3	driven 0	D3	D3	LD3	G2	G3
FPDAT4	D0	D4	UD0	D0	D4	D4	UD0	G1	G2
FPDAT5	D1	D5	UD1	D1	D5	D5	UD1	G0	G1
FPDAT6	D2	D6	UD2	D2	D6	D6	UD2	B2	B3
FPDAT7	D3	D7	UD3	D3	D7	D7	UD3	B1	B2
FPDAT8	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	B0	B1
FPDAT9	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	R0
FPDAT10	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	G0
FPDAT11	GPIO4/ HW Video Invert	GPIO4/ HW Video Invert	GPIO4/ HW Video Invert	GPIO4/ HW Video Invert	GPIO4/ HW Video Invert	GPIO4/ HW Video Invert	GPIO4/ HW Video Invert	GPIO4	B0

Note

1. Unused GPIO pins must be connected to IO V_{DD}.
2. Hardware Video Invert is enabled on FPDAT11 by REG[02h] bit 1.

6 D.C. Characteristics

Table 6-1: Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
Core V_{DD}	Supply Voltage	V_{SS} - 0.3 to 4.6	V
IO V_{DD}	Supply Voltage	V_{SS} - 0.3 to 6.0	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 6-2: Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
Core V_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	3.6	V
IO V_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3/5.0	5.5	V
V_{IN}	Input Voltage		V_{SS}		IO V_{DD}	V
T_{OPR}	Operating Temperature		-40	25	85	° C

Table 6-3: Input Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage CMOS inputs	IO V_{DD} = 3.3 5.0			0.8 1.0	V V
V_{IH}	High Level Input Voltage CMOS inputs	IO V_{DD} = 3.3 5.0	2.0 3.5			V V
V_{T+}	Positive-going Threshold CMOS Schmitt inputs	IO V_{DD} = 3.3 5.0	1.1 2.0		2.4 4.0	V V
V_{T-}	Negative-going Threshold CMOS Schmitt inputs	IO V_{DD} = 3.3 5.0	0.6 0.8		1.8 3.1	V V
I_{IZ}	Input Leakage Current	V_{DD} = Max V_{IH} = V_{DD} V_{IL} = V_{SS}	-1		1	µA
C_{IN}	Input Pin Capacitance				10	pF
HR_{PD}	Pull Down Resistance	V_I = V_{DD}	50	100	300	kΩ

Table 6-4: Output Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{OL}	Low Level Output Voltage Type 1 - TS1, CO1 Type 2- TS2, CO2 Type 3 - TS3, CO3	 I_{OL} = 3mA I_{OL} = 6mA I_{OL} = 12mA			0.4	V
V_{OH}	High Level Output Voltage Type 1 - TS1, CO1 Type 2- TS2, CO2 Type 3 - TS3, CO3	 I_{OL} = -1.5 mA I_{OL} = -3 mA I_{OL} = -6 mA	IO V_{DD} - 0.4			V
I_{OZ}	Output Leakage Current	V_{DD} = MAX V_{OH} = V_{DD} V_{OL} = V_{SS}	-1		1	µA
C_{OUT}	Output Pin Capacitance				10	pF
C_{BID}	Bidirectional Pin Capacitance				10	pF

7 A.C. Characteristics

Conditions: IO V_{DD} = 3.3V ± 10% or IO V_{DD} = 5V ± 10%
T_A = -40° C to 85° C
T_{rise} and T_{fall} for all inputs must be ≤ 5 nsec (10% ~ 90%)
C_L = 60pF (Bus/MPU Interface)
C_L = 60pF (LCD Panel Interface)

7.1 Bus Interface Timing

7.1.1 SH-4 Interface Timing

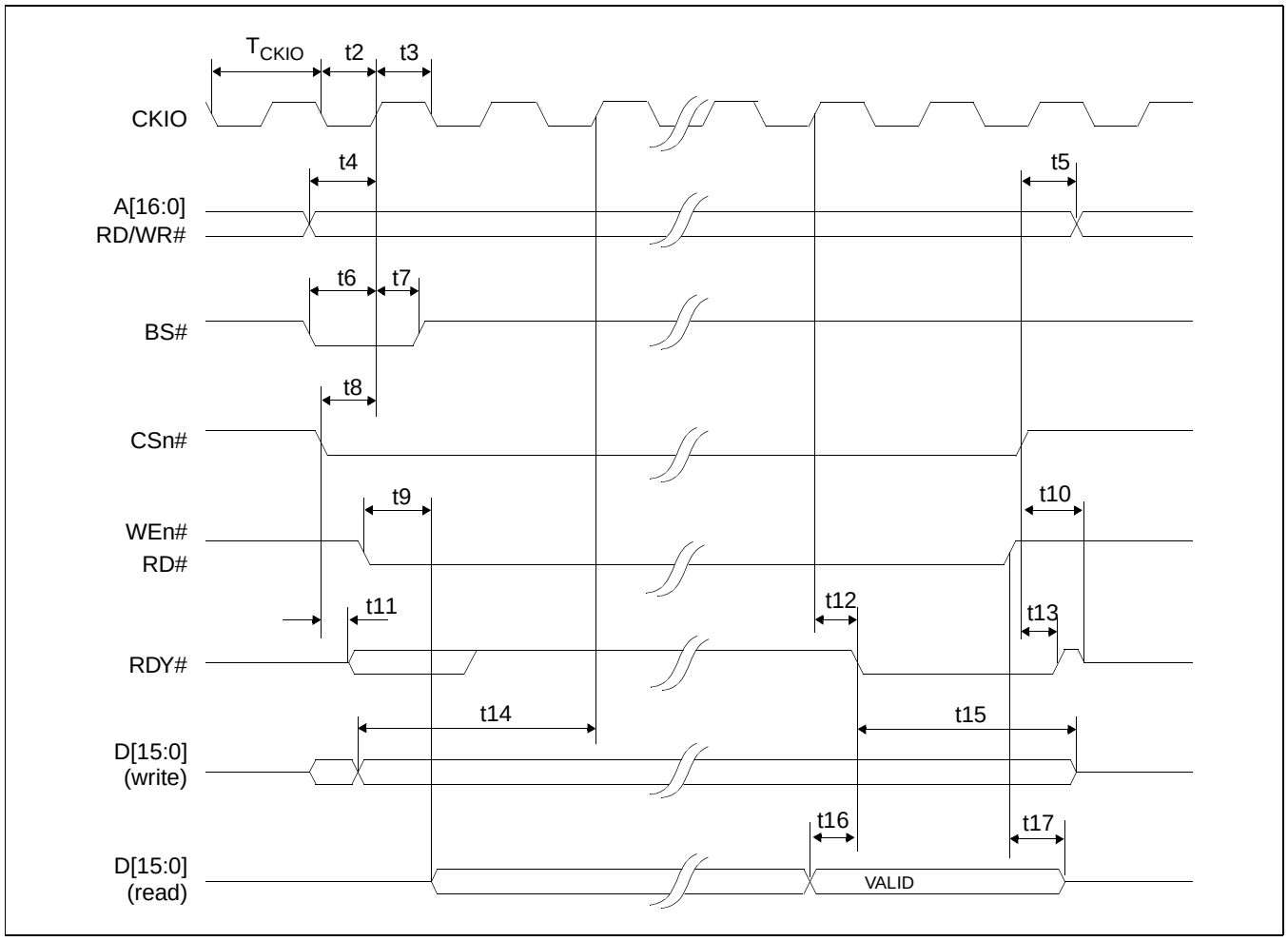


Figure 7-1: SH-4 Timing

Note

The SH-4 Wait State Control Register for the area in which the S1D13704 resides must be set to a non-zero value. The SH-4 read-to-write cycle transition must be set to a non-zero value (with reference to BUSCLK).

Table 7-1: SH-4 Timing

Symbol	Parameter	Min	Max	Units
f_{CKIO}	Bus Clock frequency	0	50	MHz
T_{CKIO}	Bus Clock period	$1/f_{CKIO}$		
t2	Clock pulse width high	17		ns
t3	Clock pulse width low	16		ns
t4	A[15:0], RD/WR# setup to CKIO	0		ns
t5	A[15:0], RD/WR# hold from CS#	0		ns
t6	BS# setup	5		ns
t7	BS# hold	5		ns
t8	CSn# setup	0		ns
t9	Falling edge RD# to DB[15:0] driven		25	ns
t10	Rising edge CSn# to RDY# high impedance		t1	ns
t11	Falling edge CSn# to RDY# driven		20	ns
t12	CKIO to RDY# low		20	ns
t13	Rising edge CSn# to RDY# high		20	ns
t14	DB[15:0] setup to 2nd CKIO after BS# (write cycle)	0		ns
t15	DB[15:0] hold (write cycle)	0		ns
t16	DB[15:0] valid to RDY# falling edge setup time (read cycle)	0		ns
t17	Rising edge RD# to DB[15:0] high impedance (read cycle)		10	ns

Note

CKIO may be turned off (held low) between accesses - see Section 13.5, “Turning Off BCLK Between Accesses” on page 86

7.1.2 SH-3 Interface Timing

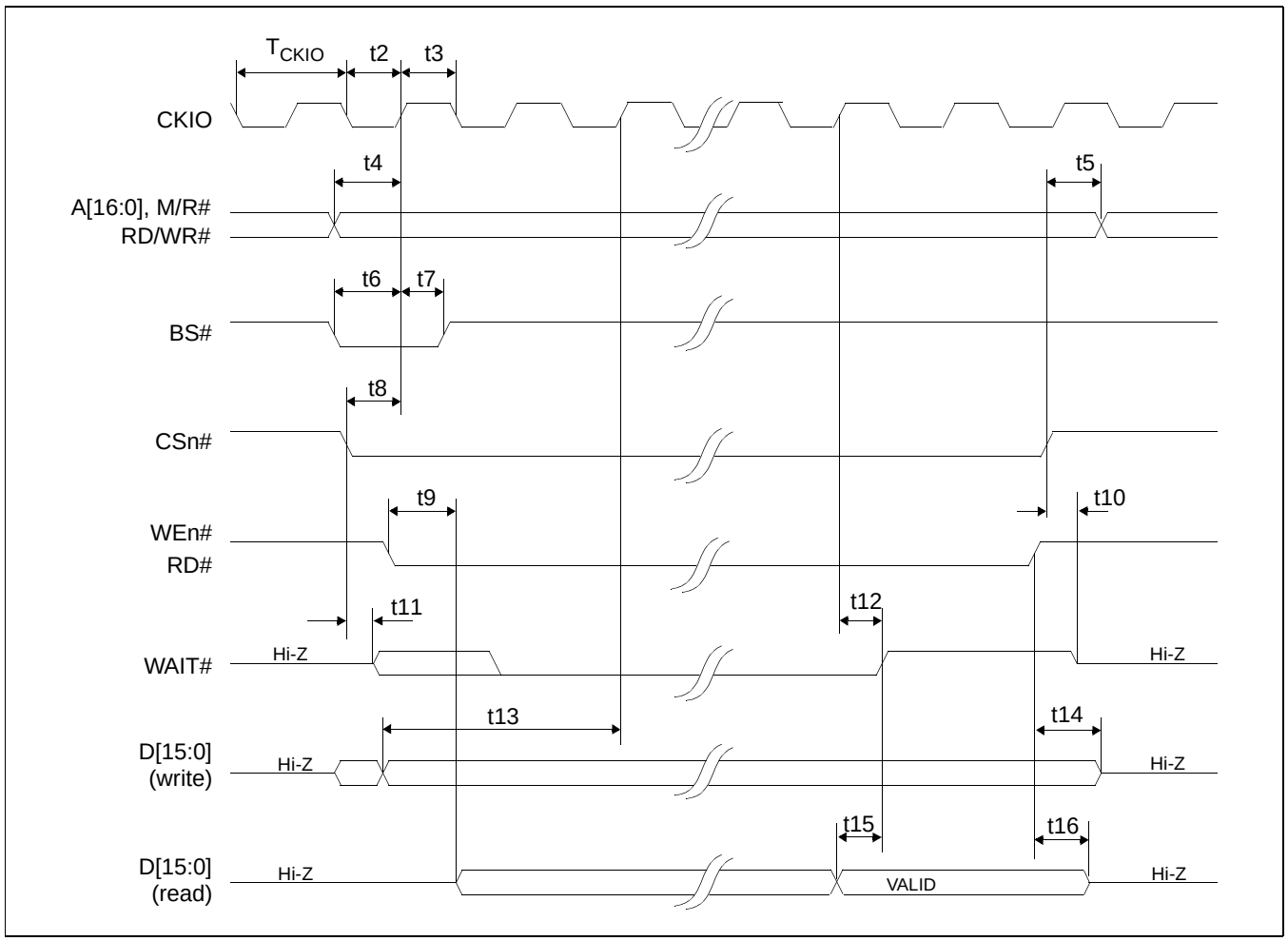


Figure 7-2: SH-3 Bus Timing

Note

The SH-3 Wait State Control Register for the area in which the S1D13704 resides must be set to a non-zero value.

Table 7-2: SH-3 Bus Timing

Symbol	Parameter	Min	Max[a]	Units
f_{CKIO}	Bus Clock frequency	0	50	MHz
T_{CKIO}	Bus Clock period	$1/f_{CKIO}$		
t2	Clock pulse width high	17		ns
t3	Clock pulse width low	16		ns
t4	A[15:0], RD/WR# setup to CKIO	0		ns
t5	A[15:0], RD/WR# hold from CS#	0		ns
t6	BS# setup	5		ns
t7	BS# hold	5		ns
t8	CSn# setup	0		ns
t9	Falling edge RD# to DB[15:0] driven		25	ns
t10	Rising edge CSn# to WAIT# high impedance		10	ns
t11	Falling edge CSn# to WAIT# driven		15	ns
t12	CKIO to WAIT# delay		20	ns
t13	DB[15:0] setup to 2nd CKIO after BS# (write cycle)	0		ns
t14	DB[15:0] hold from rising edge of WEn# (write cycle)	0		ns
t15	DB[15:0] valid to RDY# falling edge setup time (read cycle)	0		ns
t16	Rising edge RD# to DB[15:0] high impedance (read cycle)		10	ns

[a] One Software WAIT State Required

Note

CKIO may be turned off (held low) between accesses - see Section 13.5, “Turning Off BCLK Between Accesses” on page 86

7.1.3 Motorola M68K #1 Interface Timing

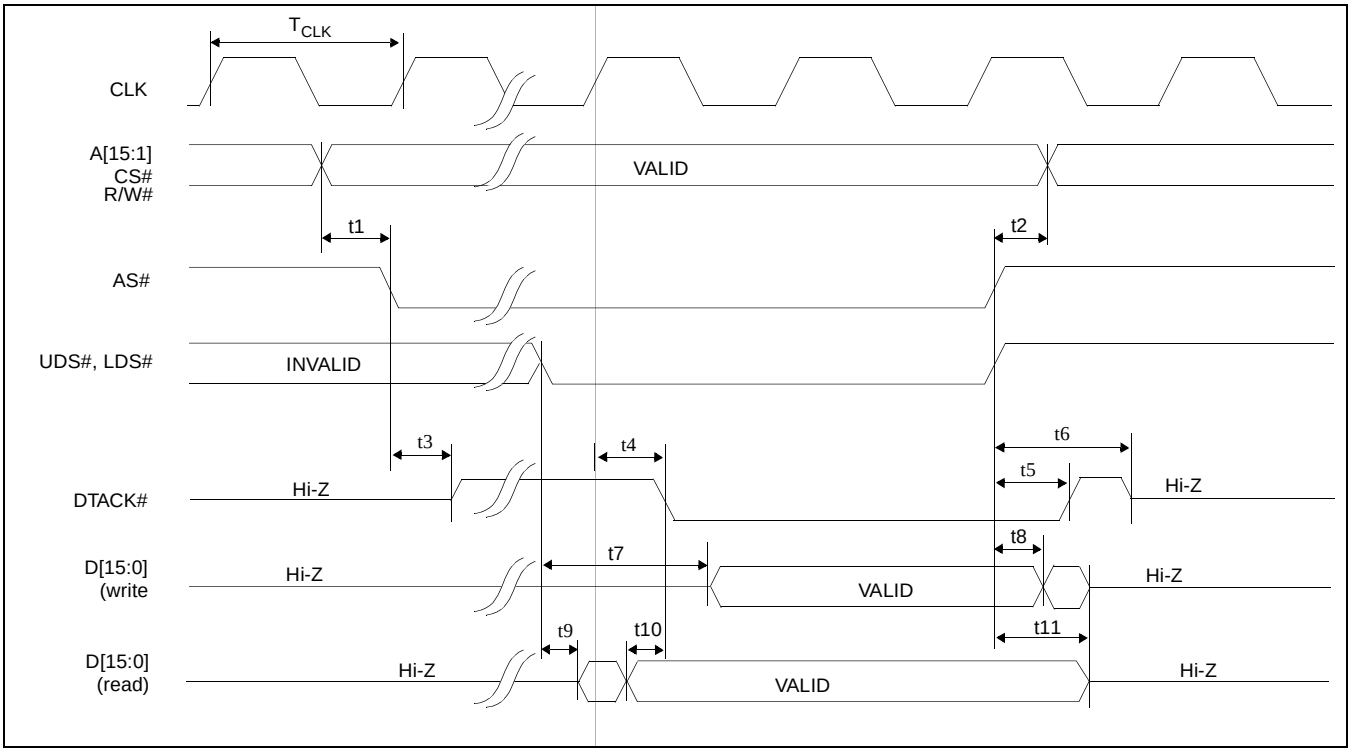


Figure 7-3: M68K #1 Bus Timing (MC68000)

Table 7-3: M68K #1 Bus Timing (MC68000)

Symbol	Parameter	Min	Max	Units
f_{CLK}	Bus Clock Frequency	0	33	MHz
T_{CLK}	Bus Clock period	$1/f_{CLK}$		
t1	A[15:1], CS# valid before AS# falling edge	0		ns
t2	A[15:1], CS# hold from AS# rising edge	0		ns
t3	AS# low to DTACK# driven high		16	ns
t4	CLK to DTACK# low		15	ns
t5	AS# high to DTACK# high		20	ns
t6	AS# high to DTACK# high impedance		T_{CLK}	
t7	UDS#, LDS# falling edge to D[15:0] valid (write cycle)		T_{CLK}	
t8	D[15:0] hold from AS# rising edge (write cycle)	0		ns
t9	UDS#, LDS# falling edge to D[15:0] driven (read cycle)		15	ns
t10	D[15:0] valid to DTACK# falling edge (read cycle)	0		ns
t11	UDS#, LDS# rising edge to D[15:0] high impedance		10	ns

Note

CLK may be turned off (held low) between accesses - see Section 13.5, “Turning Off BCLK Between Accesses” on page 86

7.1.4 Motorola M68K #2 Interface Timing

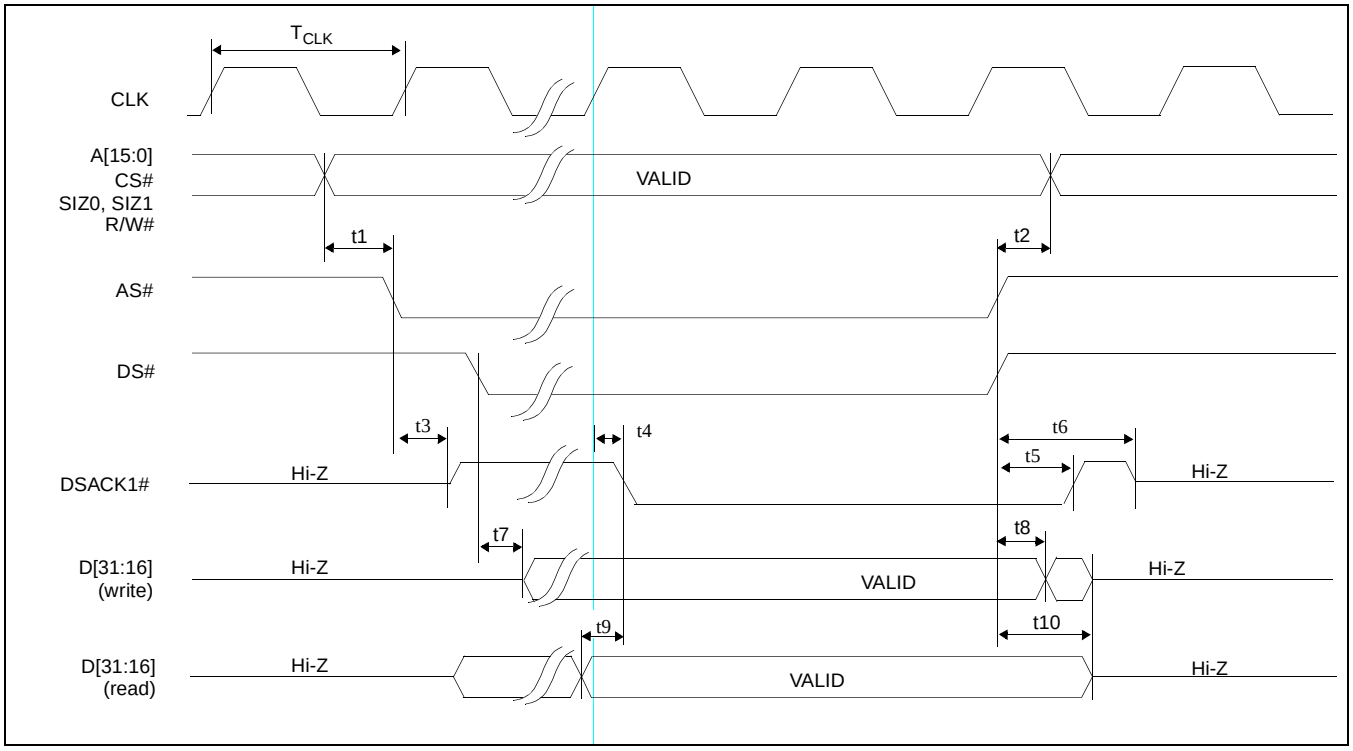


Figure 7-4: M68K #2 Timing (MC68030)

Table 7-4: M68K #2 Timing (MC68030)

Symbol	Parameter	Min	Max	Units
f_{CLK}	Bus Clock frequency	0	33	MHz
T_{CLK}	Bus Clock period	$1/f_{CLK}$		
t1	A[15:0], CS#, SIZ0, SIZ1 valid before AS# falling edge	0		ns
t2	A[15:0], CS#, SIZ0, SIZ1 hold from AS#, DS# rising edge	0		ns
t3	AS# low to DSACK1# driven high		22	ns
t4	CLK to DSACK1# low		18	ns
t5	AS# high to DSACK1# high		26	ns
t6	AS# high to DSACK1# high impedance		T_{CLK}	
t7	DS# falling edge to D[31:16] valid (write cycle)		T_{CLK} / 2	
t8	AS#, DS# rising edge to D[31:16] invalid (write cycle)	0		ns
t9	D[31:16] valid to DSACK1# low (read cycle)	0		ns
t10	AS#, DS# rising edge to D[31:16] high impedance		20	ns

Note

CLK may be turned off (held low) between accesses - see Section 13.5, “Turning Off BCLK Between Accesses” on page 86

7.1.5 Generic #1 Interface Timing

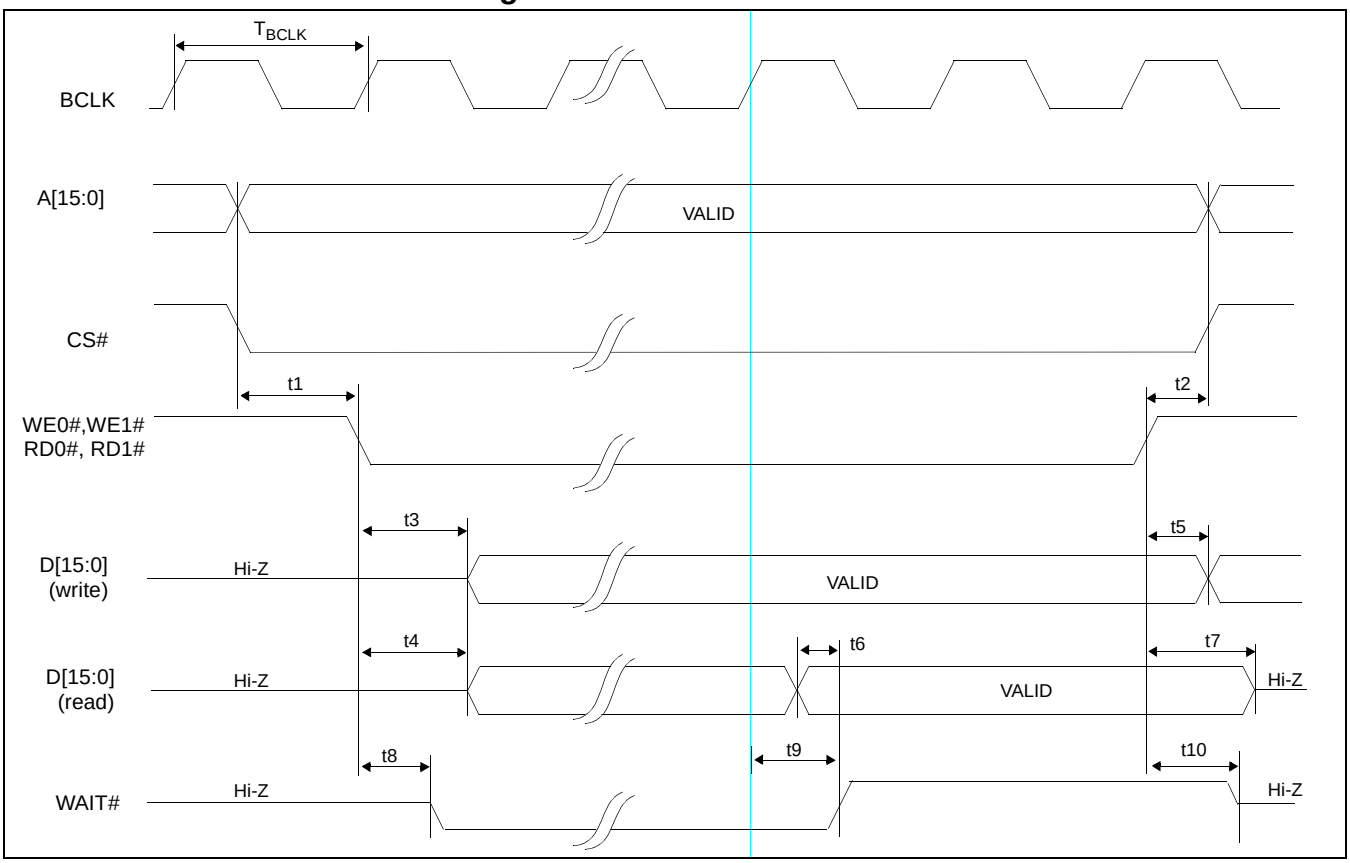


Figure 7-5: Generic #1 Timing

Table 7-5: Generic #1 Timing

Symbol	Parameter	Min	Max	Units
f_{BCLK}	Bus Clock frequency	0	50	MHz
T_{BCLK}	Bus Clock period	$1/f_{BCLK}$		MHz
t1	A[15:0], CS# valid to WE0#, WE1# low (write cycle) or RD0#, RD1# low (read cycle)	0		ns
t2	WE0#, WE1# high (write cycle) or RD0#, RD1# high (read cycle) to A[15:0], CS# invalid	0		ns
t3	WE0#, WE1# low to D[15:0] valid (write cycle)		T_{BCLK}	
t4	RD0#, RD1# low to D[15:0] driven (read cycle)		17	ns
t5	WE0#, WE1# high to D[15:0] invalid (write cycle)	0		ns
t6	D[15:0] valid to WAIT# high (read cycle)	0		ns
t7	RD0#, RD1# high to D[15:0] high impedance (read cycle)		10	ns
t8	WE0#, WE1# low (write cycle) or RD0#, RD1# low (read cycle) to WAIT# driven low		16	ns
t9	BCLK to WAIT# high		16	ns
t10	WE0#, WE1# high (write cycle) or RD0#, RD1# high (read cycle) to WAIT# high impedance		11	ns

Note

BCLK may be turned off (held low) between accesses - see Section 13.5, “Turning Off BCLK Between Accesses” on page 86

7.1.6 Generic #2 Interface Timing

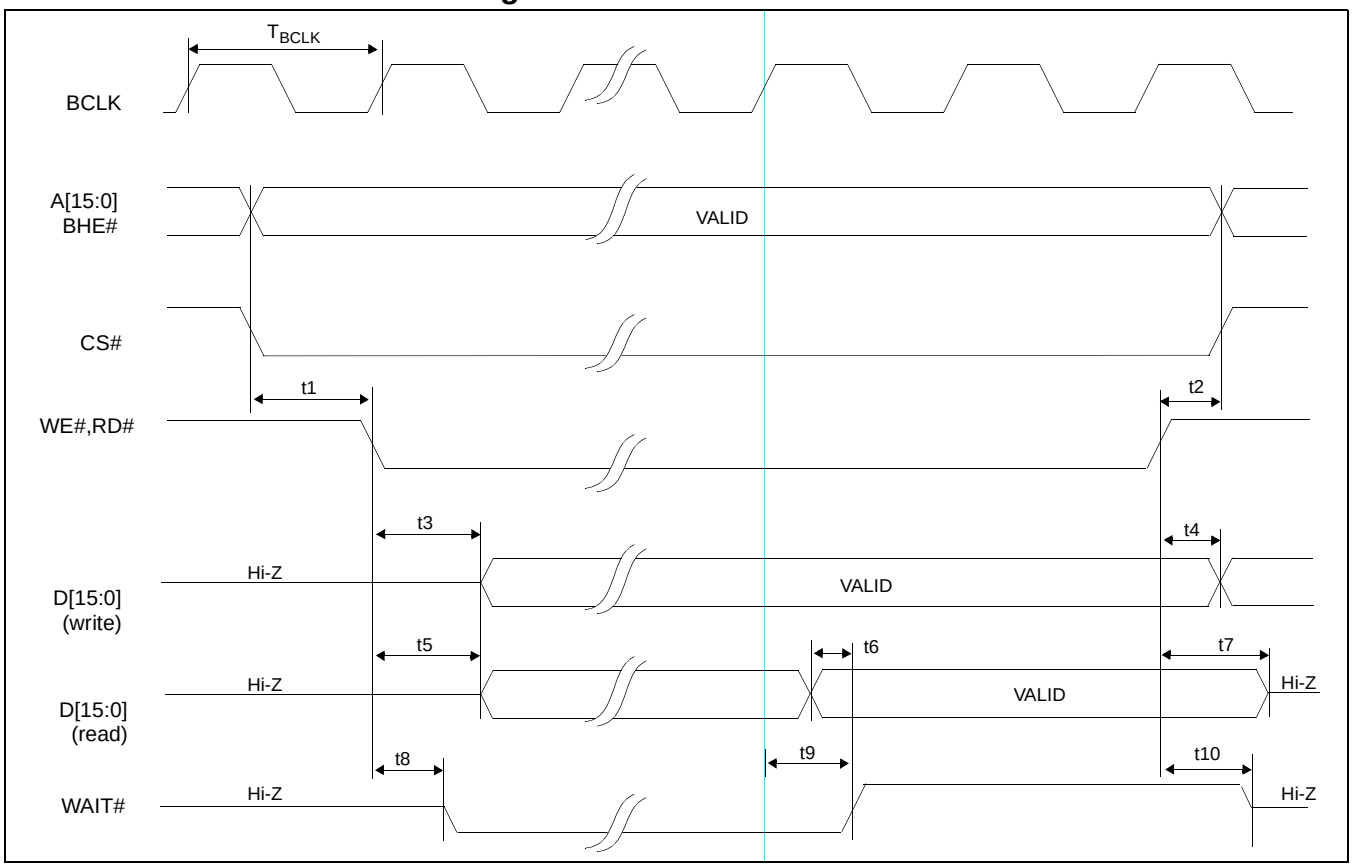


Figure 7-6: Generic #2 Timing

Table 7-6: Generic #2 Timing

Symbol	Parameter	Min	Max	Units
f_{BCLK}	Bus Clock frequency	0	50	MHz
T_{BCLK}	Bus Clock period	$1/f_{BCLK}$		
t1	A[15:0], BHE#, CS# valid to WE#, RD# low	0		ns
t2	WE#, RD# high to A[15:0], BHE#, CS# invalid	0		ns
t3	WE# low to D[15:0] valid (write cycle)		T_{BCLK}	
t4	WE# high to D[15:0] invalid (write cycle)	0		ns
t5	RD# low to D[15:0] driven (read cycle)		16	ns
t6	D[15:0] valid to WAIT# high (read cycle)	0		ns
t7	RD# high to D[15:0] high impedance (read cycle)		10	ns
t8	WE#, RD# low to WAIT# driven low		14	ns
t9	BCLK to WAIT# high		16	ns
t10	WE#, RD# high to WAIT# high impedance		11	ns

Note

BCLK may be turned off (held low) between accesses - see Section 13.5, “Turning Off BCLK Between Accesses” on page 86

7.2 Clock Input Requirements

Clock Input Waveform

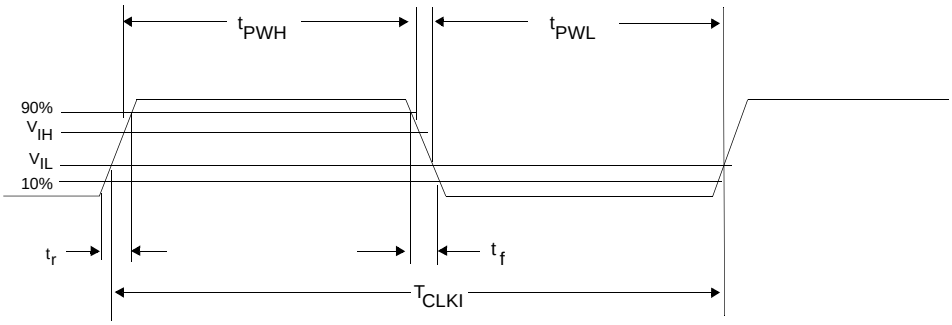


Figure 7-7: Clock Input Requirements

Table 7-7: Clock Input Requirements

Symbol	Parameter	Min	Max	Units
f_{CLKI}	Input Clock Frequency (CLKI)	0	50	MHz
T_{CLKI}	Input Clock period (CLKI)	$1/f_{CLKI}$		
t_{PWH}	Input Clock Pulse Width High (CLKI)	8		ns
t_{PWL}	Input Clock Pulse Width Low (CLKI)	8		ns
t_f	Input Clock Fall Time (10% - 90%)		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5	ns

Note

When CLKI is > 25MHz it must be divided by 2 (REG[02h] bit 4 = 1).

7.3 Display Interface

7.3.1 Power On/Reset Timing

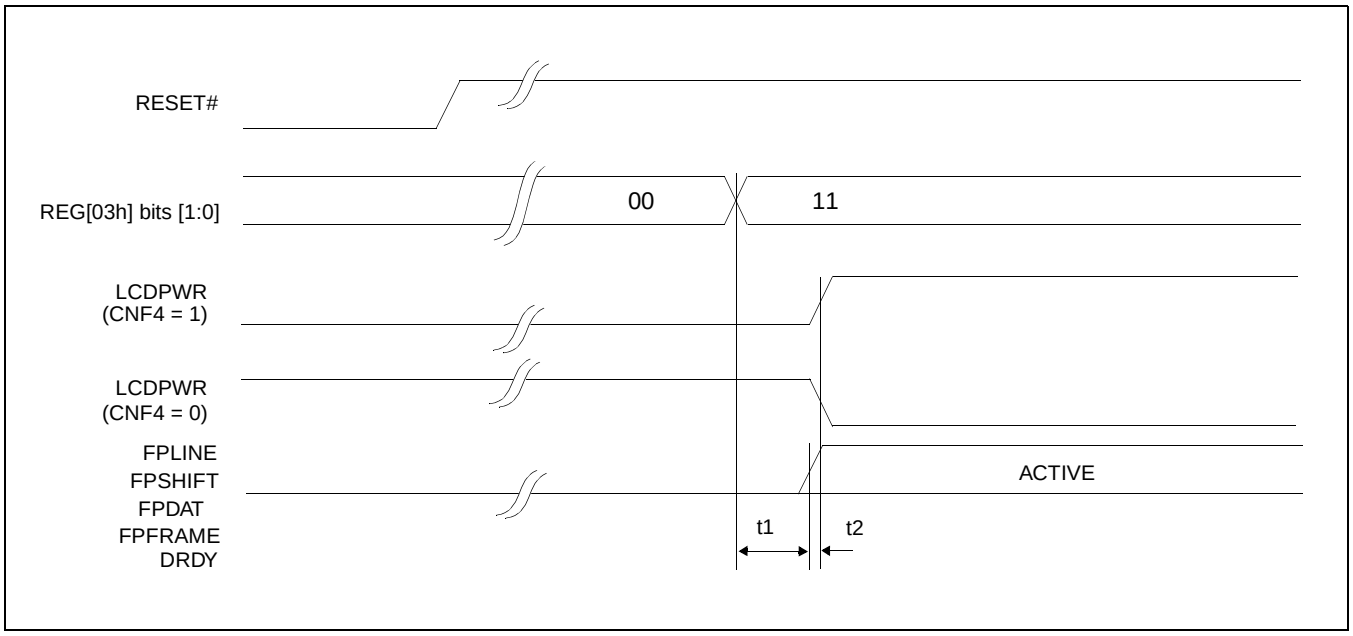


Figure 7-8: LCD Panel Power On/Reset Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	REG[03h] to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY active			$T_{FPFRAME}$	ns
t2	FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY active to LCDPWR		0		Frames

Note

Where $T_{FPFRAME}$ is the period of FPFRAME and T_{PCLK} is the period of the pixel clock.

7.3.2 Power Down/Up Timing

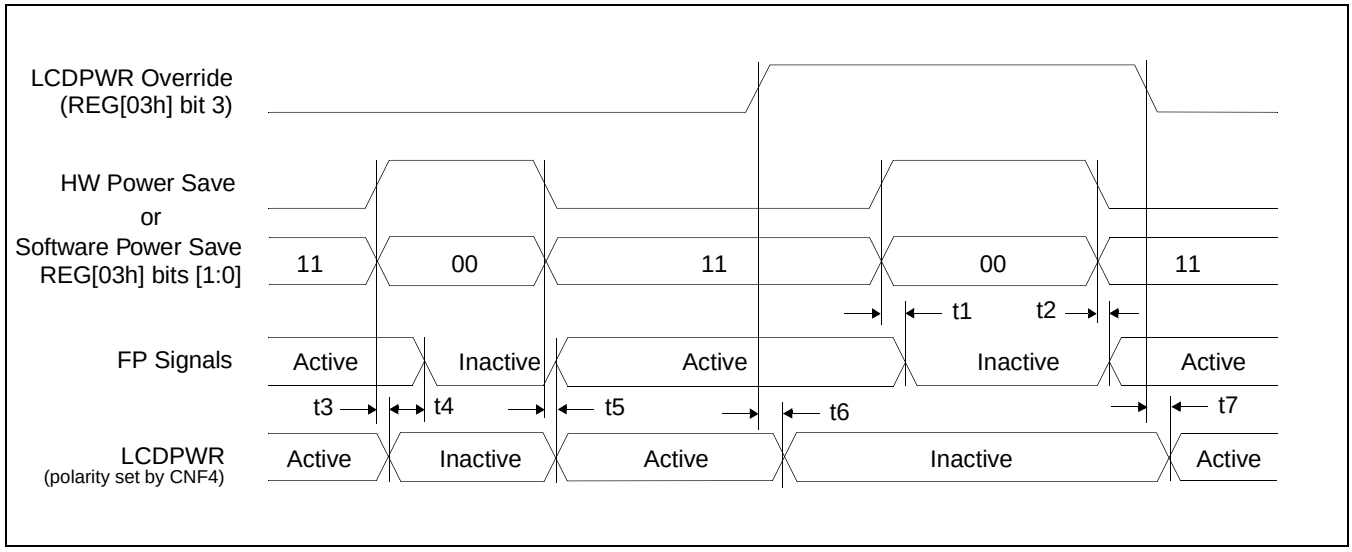
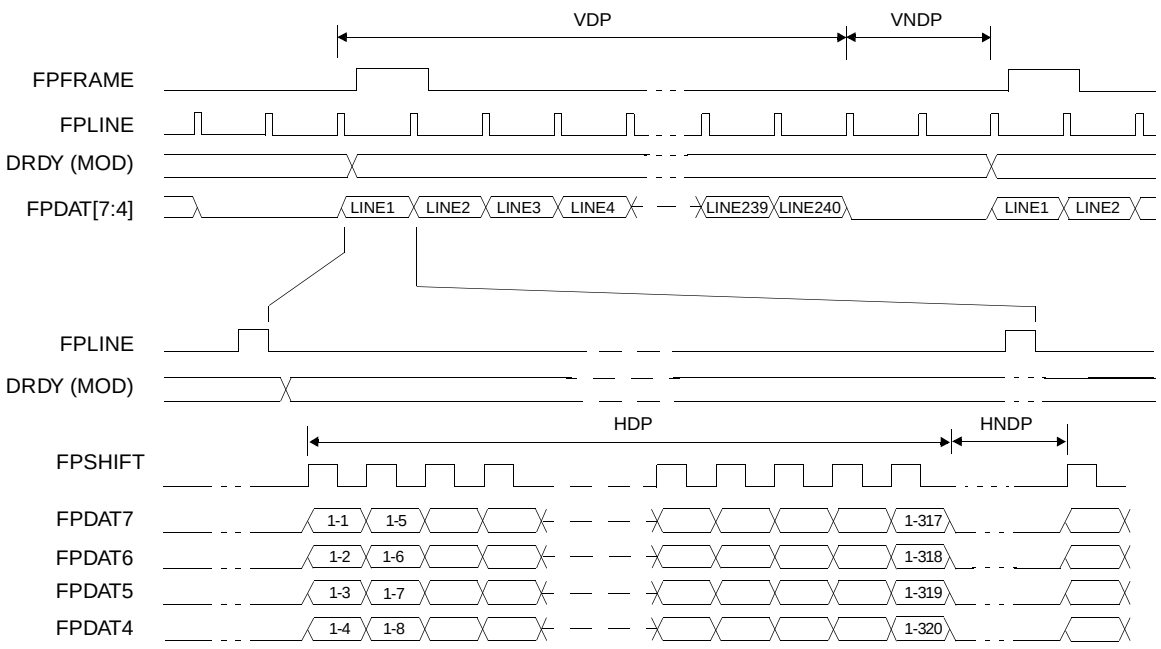


Figure 7-9: Power Down/Up Timing

Table 7-8: Power Down/Up Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	HW Power Save active to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY inactive - LCDPWR Override = 1			1	Frame
t2	HW Power Save inactive to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY active - LCDPWR Override = 1			1	Frame
t3	HW Power Save active to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY inactive - LCDPWR Override = 0			1	Frame
t4	LCDPWR low to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY inactive - LCDPWR Override = 0		127		Frame
t5	HW Power Save inactive to FPLINE, FPFRAME, FPSHIFT, FPDAT, DRDY, LCDPWR active - LCDPWR Override = 0		0		Frame
t6	LCDPWR Override active (1) to LCDPWR inactive			1	Frame
t7	LCDPWR Override inactive (1) to LCDPWR active			1	Frame

7.3.3 Single Monochrome 4-Bit Panel Timing



* Diagram drawn with 2 FPLINE vertical blank period
Example timing for a 320x240 panel
For this timing diagram Mask FPSHIFT, REG[01h] bit 3, is set to 1

Figure 7-10: Single Monochrome 4-Bit Panel Timing

VDP =	Vertical Display Period	= (REG[06h] bits 1-0, REG[05h] bits 7-0) + 1 Lines
VNDP =	Vertical Non-Display Period	= REG[0Ah] bits 5-0 Lines
HDP =	Horizontal Display Period	= ((REG[04h] bits 6-0) + 1) x 8Ts
HNDP =	Horizontal Non-Display Period	= (REG[08h] + 4) x 8Ts

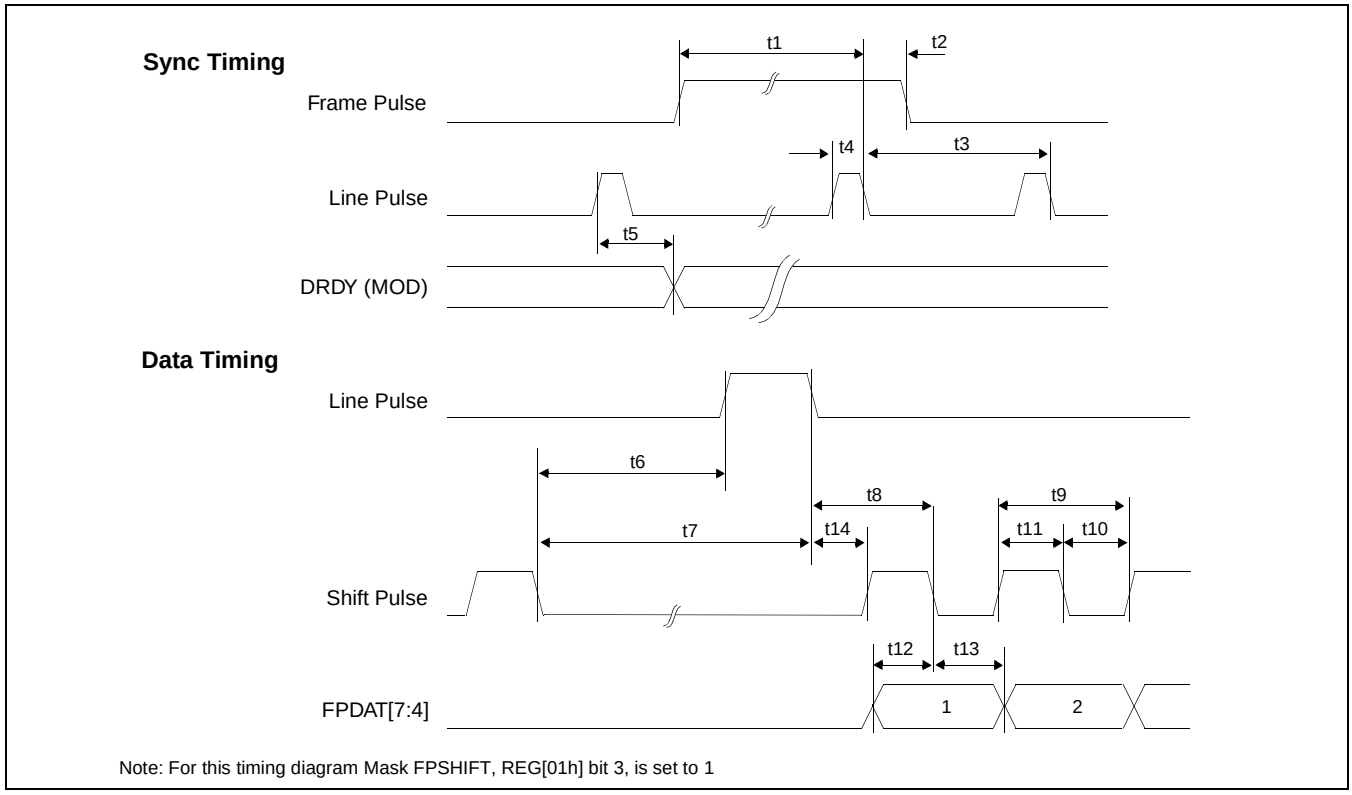


Figure 7-11: Single Monochrome 4-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t3	Line Pulse period	note 3			
t4	Line Pulse pulse width	9			Ts
t5	MOD delay from Line Pulse rising edge	1			Ts
t6	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t7	Shift Pulse falling edge to Line Pulse falling edge	note 5			
t8	Line Pulse falling edge to Shift Pulse falling edge	t14 + 2			Ts
t9	Shift Pulse period	4			Ts
t10	Shift Pulse pulse width low	2			Ts
t11	Shift Pulse pulse width high	2			Ts
t12	FPDAT[7:4] setup to Shift Pulse falling edge	2			Ts
t13	FPDAT[7:4] hold to Shift Pulse falling edge	2			Ts
t14	Line Pulse falling edge to Shift Pulse rising edge	23			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = $t3_{min}$ - 9Ts
3. $t3_{min}$ = [((REG[04h] bits 6-0)+1) x 8 + ((REG[08h] bits 4-0) + 4) x 8]Ts
4. $t6_{min}$ = [(REG[08h] bits 4-0) x 8 + 2]Ts
5. $t7_{min}$ = [(REG[08h] bits 4-0) x 8 + 11]Ts

7.3.4 Single Monochrome 8-Bit Panel Timing

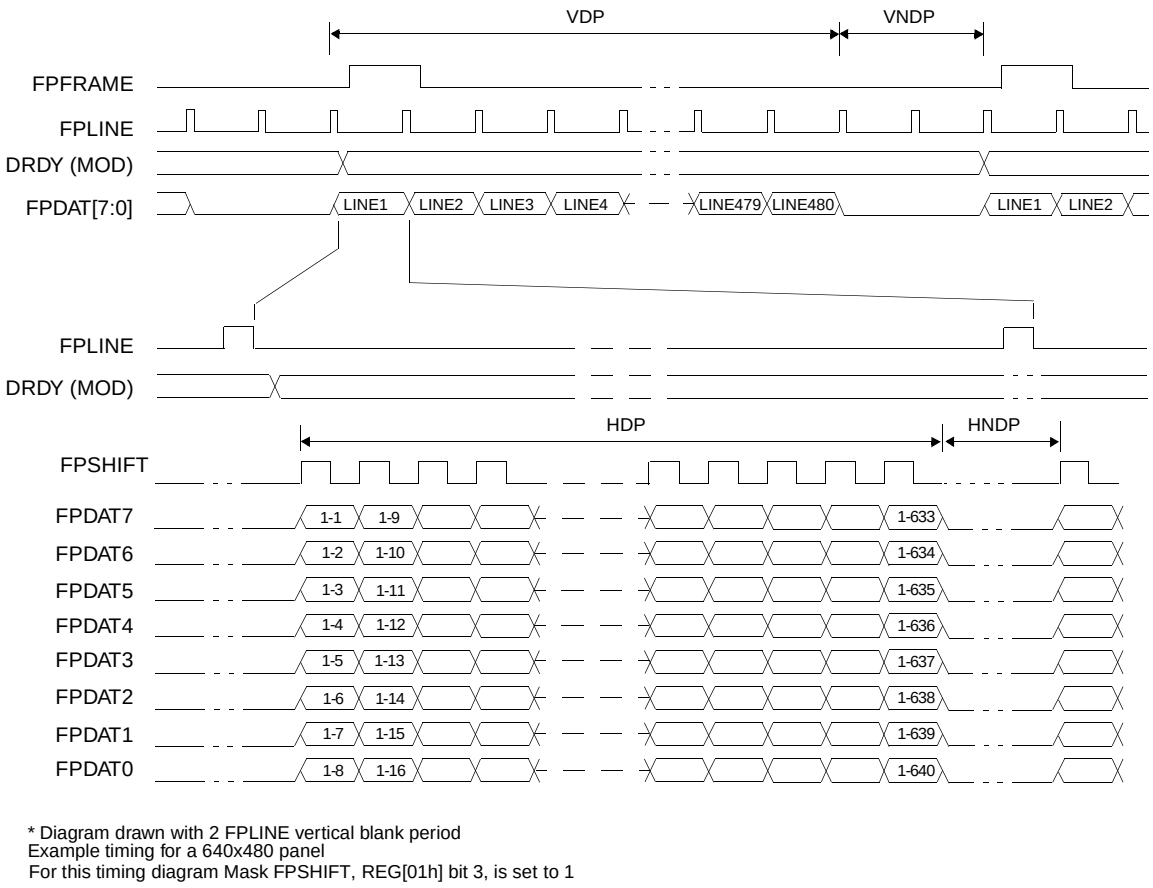


Figure 7-12: Single Monochrome 8-Bit Panel Timing

VDP =	Vertical Display Period	= (REG[06h] bits 1-0, REG[05h] bits 7-0) + 1 Lines
VNDP =	Vertical Non-Display Period	= REG[0Ah] bits 5-0 Lines
HDP =	Horizontal Display Period	= ((REG[04h] bits 6-0) + 1) x 8Ts
HNDP =	Horizontal Non-Display Period	= (REG[08h] + 4) x 8Ts

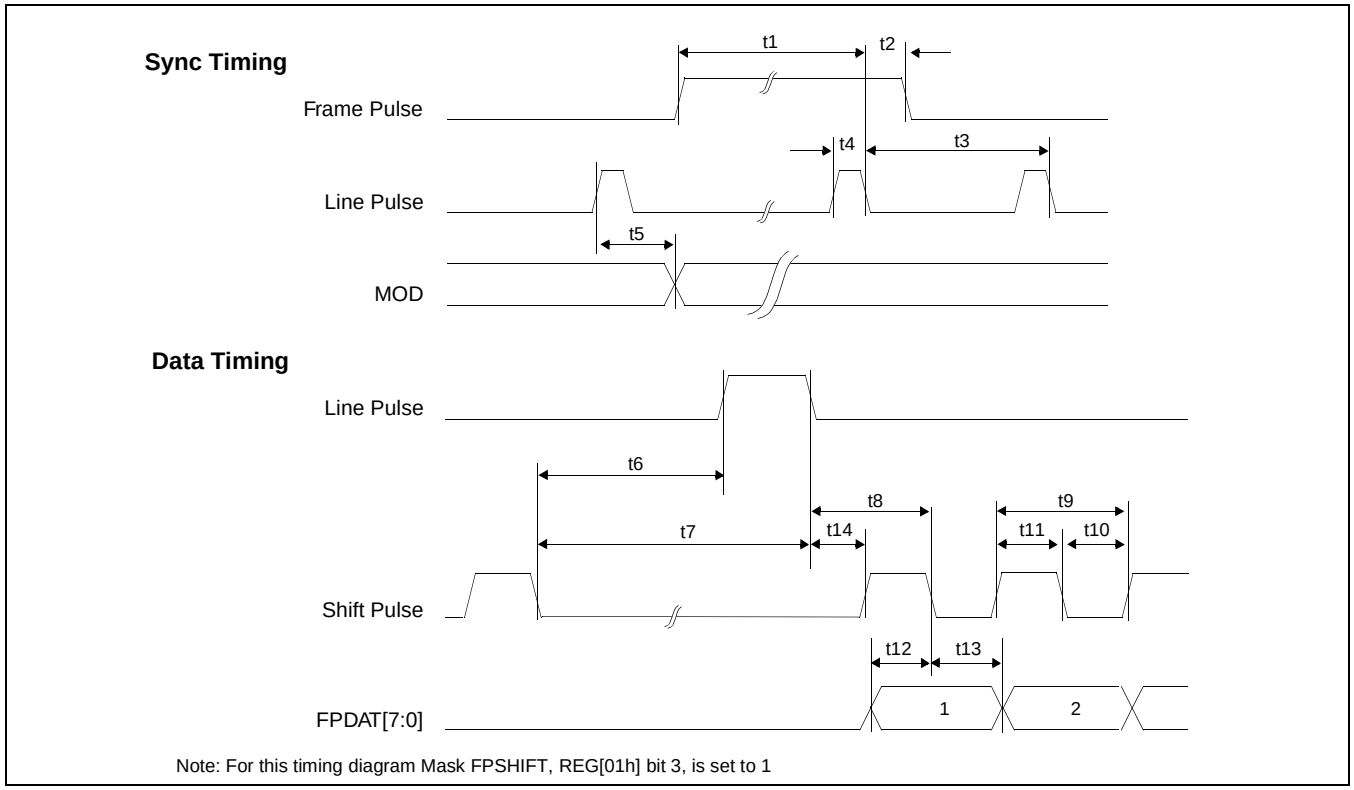


Figure 7-13: Single Monochrome 8-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t3	Line Pulse period	note 3			
t4	Line Pulse pulse width	9			Ts
t5	MOD delay from Line Pulse rising edge	1			Ts
t6	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t7	Shift Pulse falling edge to Line Pulse falling edge	note 5			
t8	Line Pulse falling edge to Shift Pulse falling edge	t14 + 4			Ts
t9	Shift Pulse period	8			Ts
t10	Shift Pulse pulse width low	4			Ts
t11	Shift Pulse pulse width high	4			Ts
t12	FPDAT[7:0] setup to Shift Pulse falling edge	4			Ts
t13	FPDAT[7:0] hold to Shift Pulse falling edge	4			Ts
t14	Line Pulse falling edge to Shift Pulse rising edge	23			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = $t3_{min}$ - 9Ts
3. $t3_{min}$ = [((REG[04h] bits 6-0)+1) x 8 + ((REG[08h] bits 4-0) + 4) x 8]Ts
4. $t6_{min}$ = [(REG[08h] bits 4-0) x 8 + 4]Ts
5. $t7_{min}$ =[(REG[08h] bits 4-0) x 8 + 13]Ts

7.3.5 Single Color 4-Bit Panel Timing

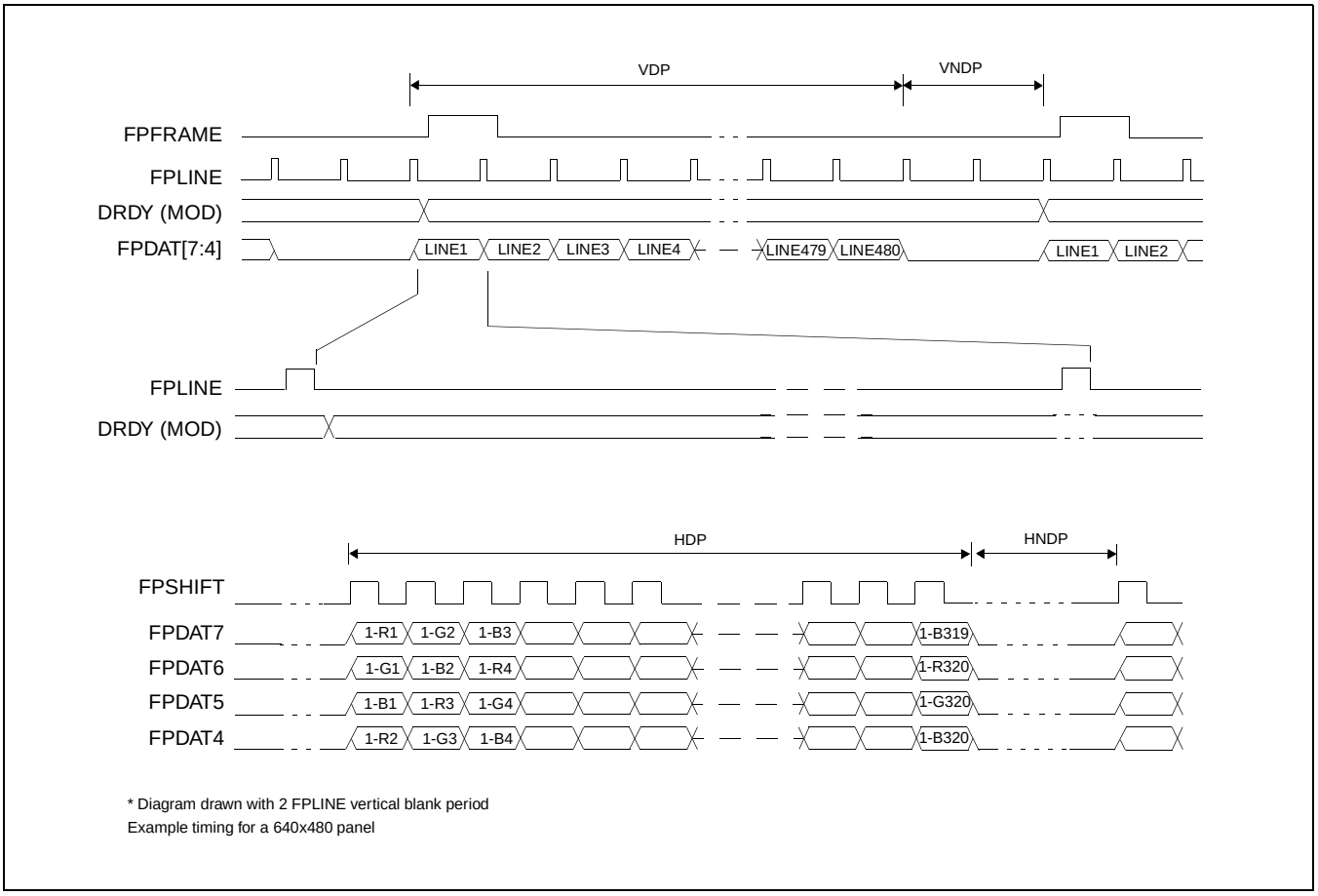


Figure 7-14: Single Color 4-Bit Panel Timing

VDP =	Vertical Display Period	= (REG[06h] bits 1-0, REG[05h] bits 7-0) + 1 Lines
VNDP =	Vertical Non-Display Period	= REG[0Ah] bits 5-0 Lines
HDP =	Horizontal Display Period	= ((REG[04h] bits 6-0) + 1) x 8Ts
HNDP =	Horizontal Non-Display Period	= (REG[08h] + 4) x 8Ts

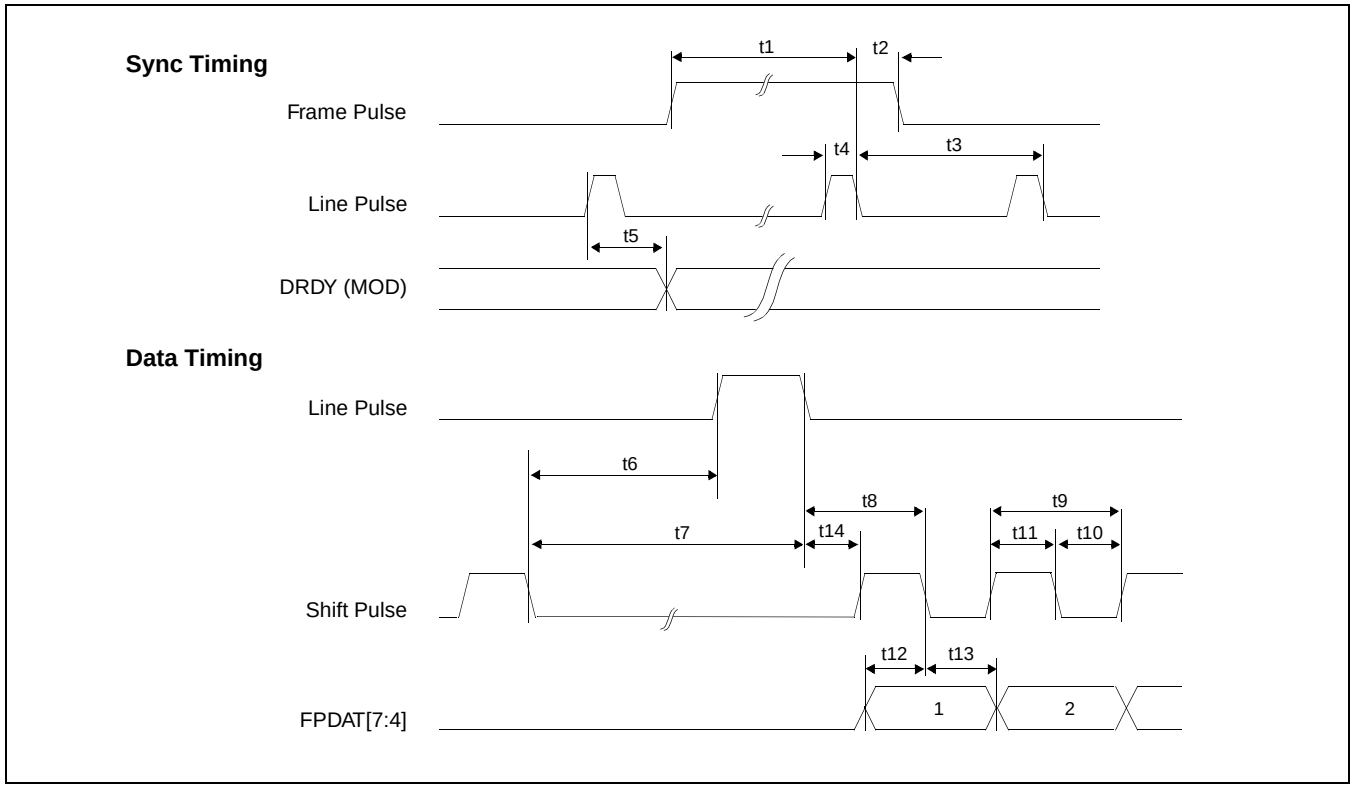
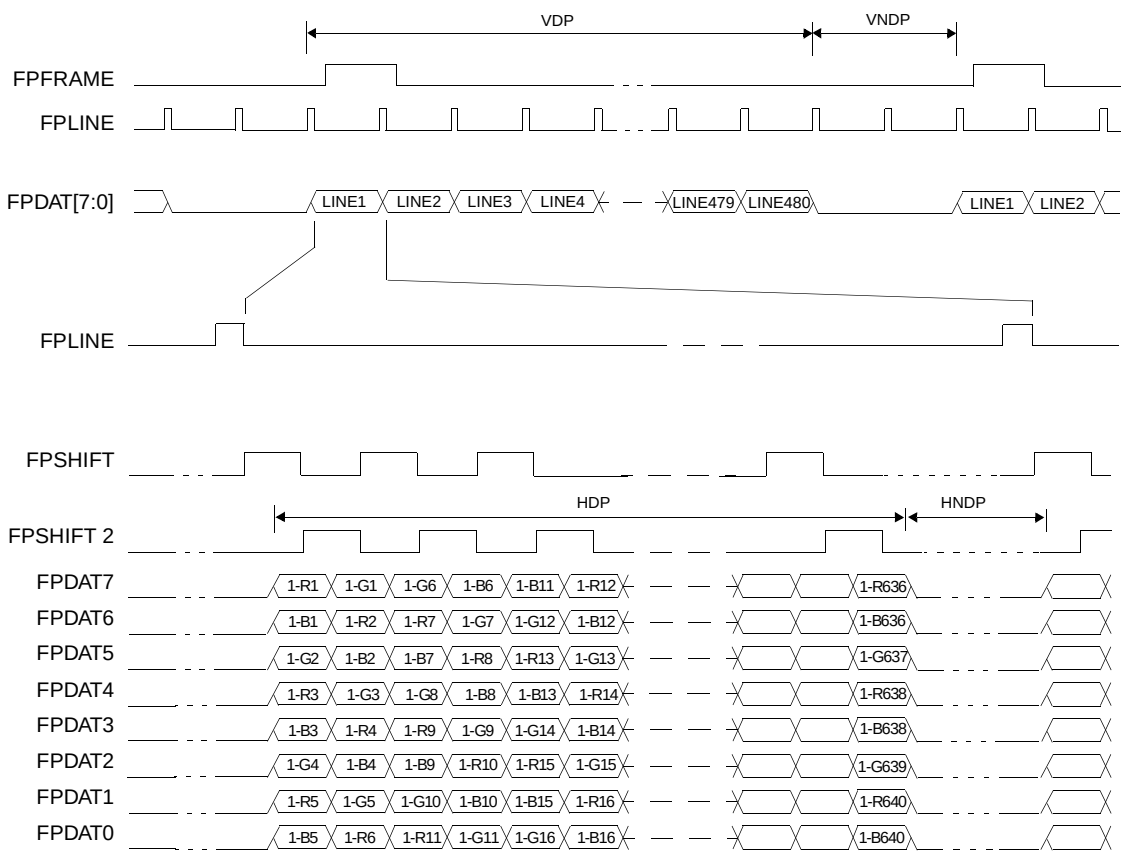


Figure 7-15: Single Color 4-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t3	Line Pulse period	note 3			
t4	Line Pulse pulse width	9			Ts
t5	MOD delay from Line Pulse rising edge	1			Ts
t6	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t7	Shift Pulse falling edge to Line Pulse falling edge	note 5			
t8	Line Pulse falling edge to Shift Pulse falling edge	t14 + 0.5			Ts
t9	Shift Pulse period	1			Ts
t10	Shift Pulse pulse width low	0.5			Ts
t11	Shift Pulse pulse width high	0.5			Ts
t12	FPDAT[7:4] setup to Shift Pulse falling edge	0.5			Ts
t13	FPDAT[7:4] hold to Shift Pulse falling edge	0.5			Ts
t14	Line Pulse falling edge to Shift Pulse rising edge	23			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = $t3_{min}$ - 9Ts
3. $t3_{min}$ = [((REG[04h] bits 6-0)+1) x 8 + ((REG[08h] bits 4-0) + 4) x 8]Ts
4. $t6_{min}$ = [(REG[08h] bits 4-0) x 8 + 0.5]Ts
5. $t7_{min}$ = [(REG[08h] bits 4-0) x 8 + 9.5]Ts

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



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

Figure 7-16: Single Color 8-Bit Panel Timing (Format 1)

VDP =	Vertical Display Period	= (REG[06h] bits 1-0, REG[05h] bits 7-0) + 1 Lines
VNDP =	Vertical Non-Display Period	= REG[0Ah] bits 5-0 Lines
HDP =	Horizontal Display Period	= ((REG[04h] bits 6-0) + 1) x 8Ts
HNDP =	Horizontal Non-Display Period	= (REG[08h] + 4) x 8Ts

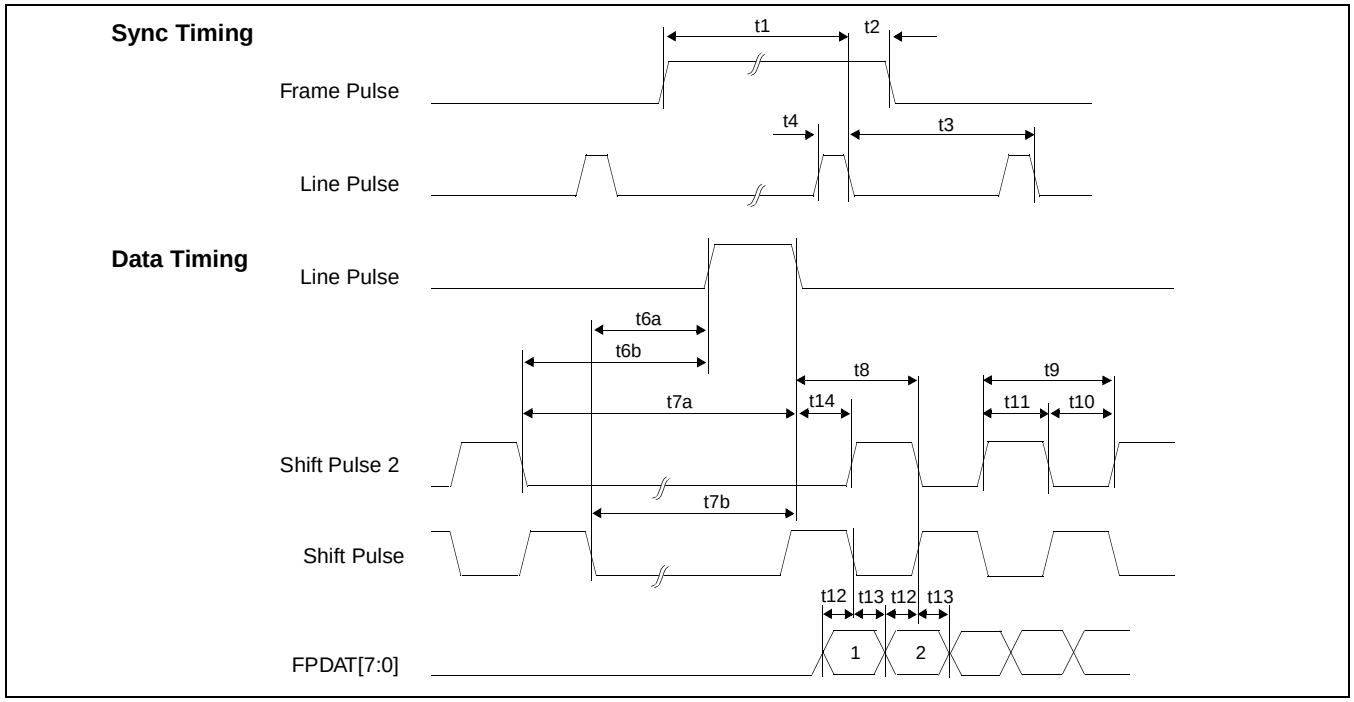
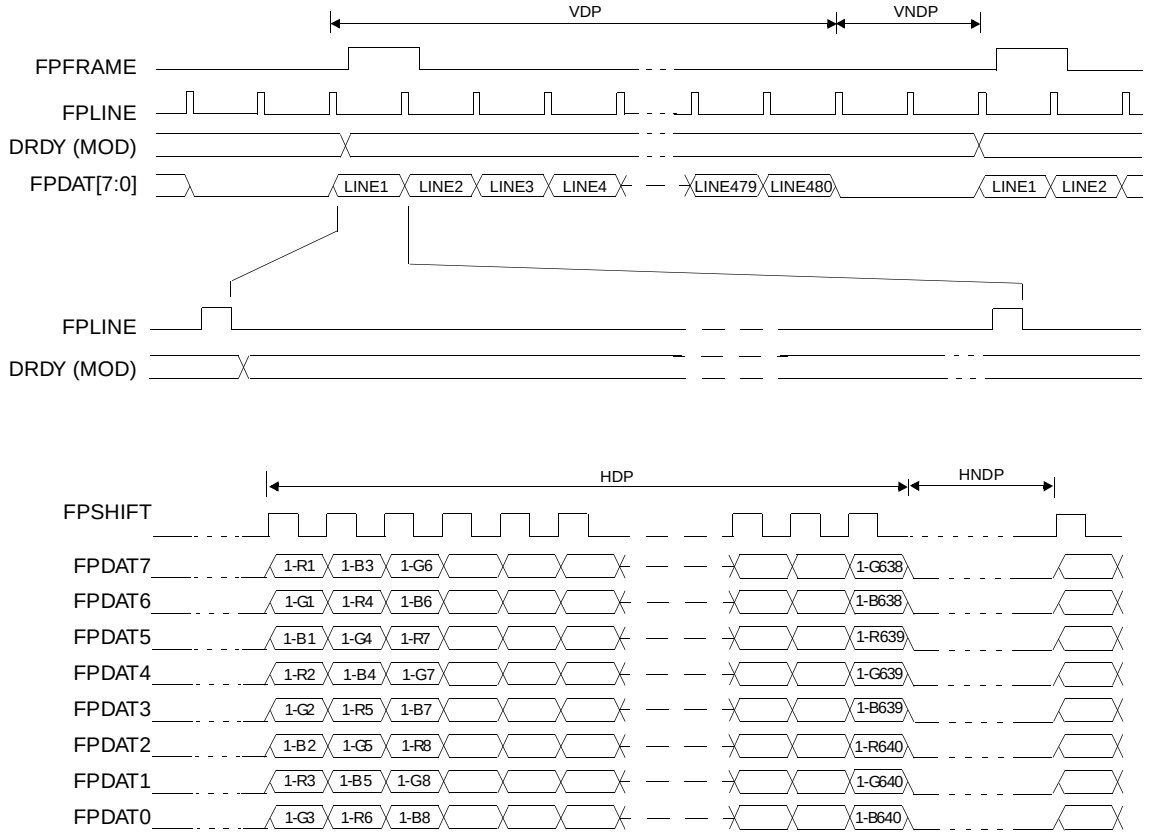


Figure 7-17: Single Color 8-Bit Panel A.C. Timing (Format 1)

Symbol	Parameter	Min	Typ	Max	Units
t1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t3	Line Pulse period	note 3			
t4	Line Pulse pulse width	9			Ts
t6a	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t6b	Shift Pulse 2 falling edge to Line Pulse rising edge	note 5			
t7a	Shift Pulse 2 falling edge to Line Pulse falling edge	note 6			
t7b	Shift Pulse falling edge to Line Pulse falling edge	note 7			
t8	Line Pulse falling edge to Shift Pulse rising, Shift Pulse 2 falling edge	t14 + 2			Ts
t9	Shift Pulse 2, Shift Pulse period	4			Ts
t10	Shift Pulse 2, Shift Pulse pulse width low	2			Ts
t11	Shift Pulse 2, Shift Pulse pulse width high	2			Ts
t12	FPDAT[7:0] setup to Shift Pulse 2, Shift Pulse falling edge	1			Ts
t13	FPDAT[7:0] hold to Shift Pulse 2, Shift Pulse falling edge	1			Ts
t14	Line Pulse falling edge to Shift Pulse rising edge	23			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = $t3_{min}$ - 9Ts
3. $t3_{min}$ = [((REG[04h] bits 6-0)+1) x 8 + ((REG[08h] bits 4-0) + 4) x 8]Ts
4. $t6a_{min}$ = [(REG[08h] bits 4-0) x 8 + t13 - t10]Ts
5. $t6b_{min}$ = [(REG[08h] bits 4-0) x 8 + t13]Ts
6. $t7a_{min}$ = [(REG[08h] bits 4-0) x 8 + 11]Ts
7. $t7b_{min}$ = [(REG[08h] bits 4-0) x 8 + 11] - t10]Ts

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



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

Figure 7-18: Single Color 8-Bit Panel Timing (Format 2)

VDP =	Vertical Display Period	= (REG[06h] bits 1-0, REG[05h] bits 7-0) + 1 Lines
VNDP =	Vertical Non-Display Period	= REG[0Ah] bits 5-0 Lines
HDP =	Horizontal Display Period	= ((REG[04h] bits 6-0) + 1) x 8Ts
HNDP =	Horizontal Non-Display Period	= (REG[08h] + 4) x 8Ts

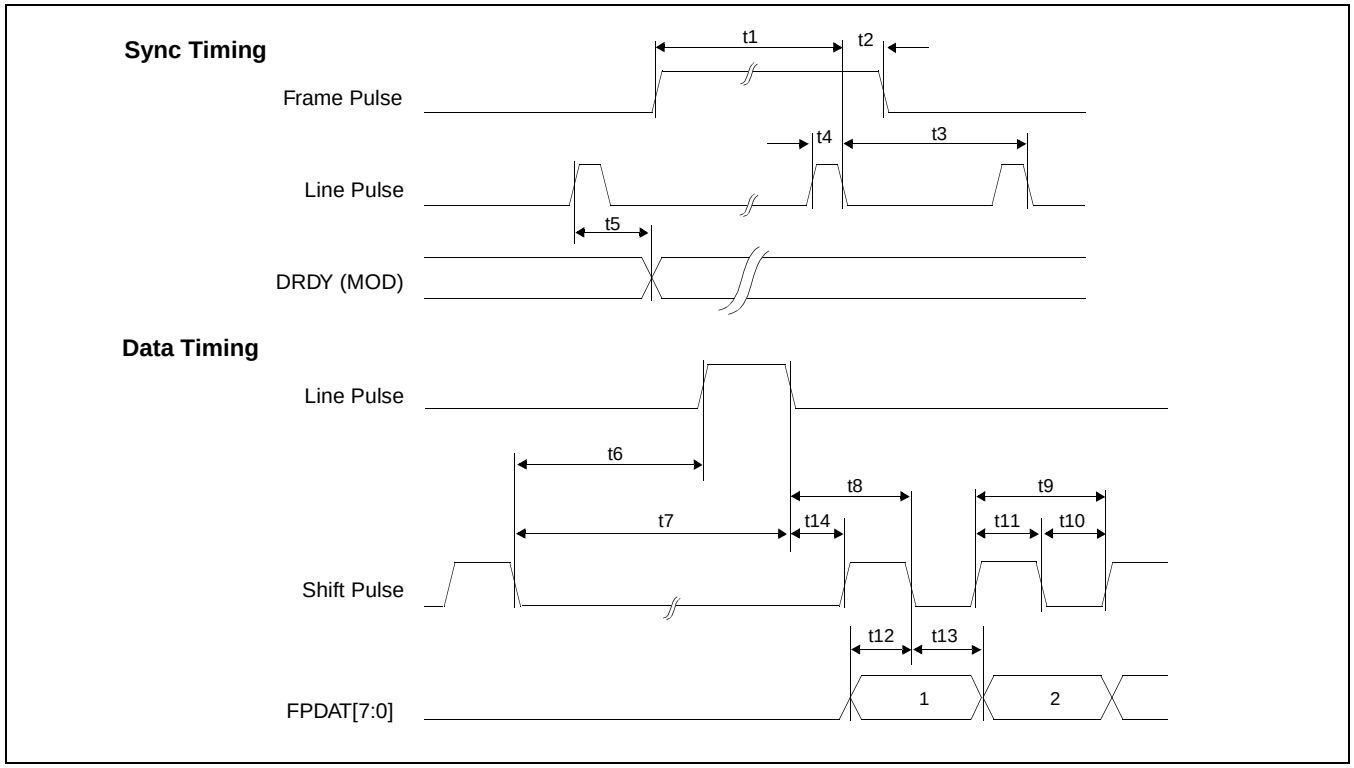
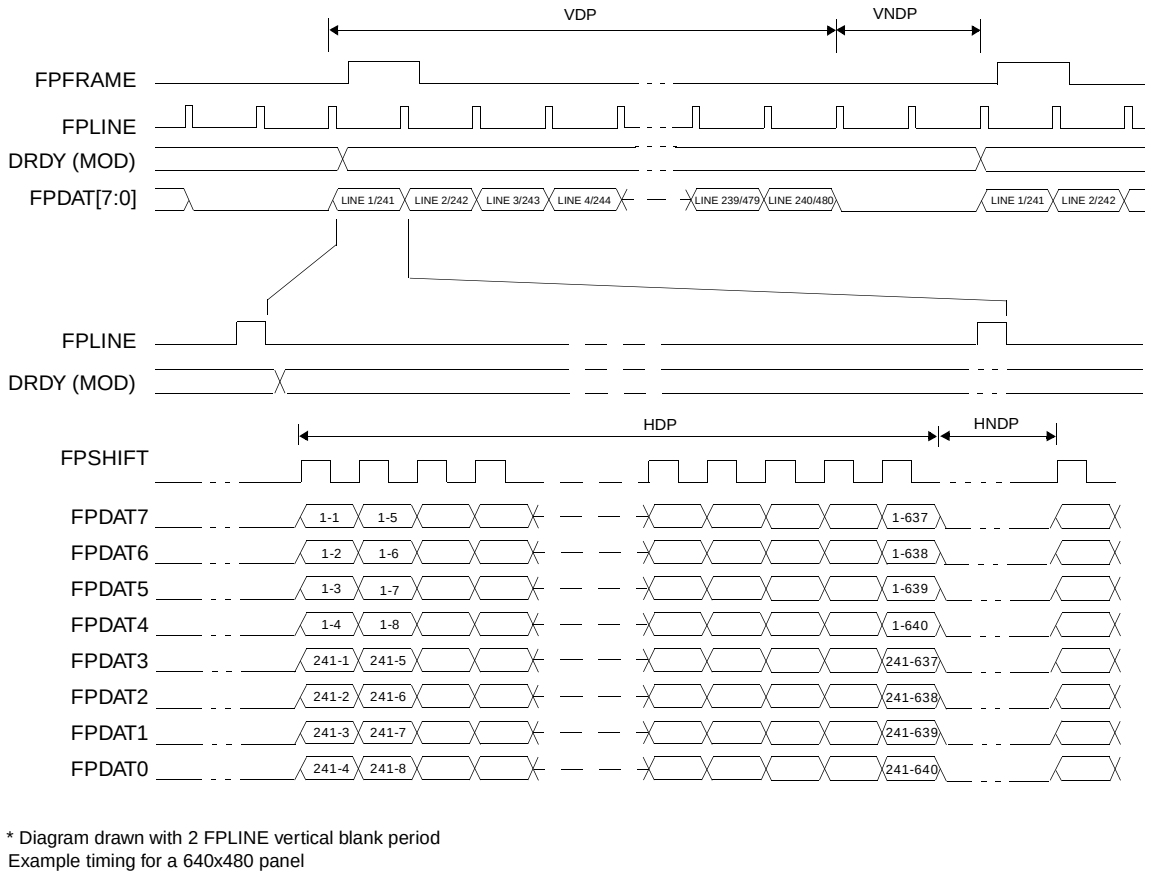


Figure 7-19: Single Color 8-Bit Panel A.C. Timing (Format 2)

Symbol	Parameter	Min	Typ	Max	Units
t1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t3	Line Pulse period	note 3			
t4	Line Pulse pulse width	9			Ts
t5	MOD delay from Line Pulse rising edge	1			Ts
t6	Shift Pulse falling edge to Line Pulse rising edge	note 4			
t7	Shift Pulse falling edge to Line Pulse falling edge	note 5			
t8	Line Pulse falling edge to Shift Pulse falling edge	t14 + 2			Ts
t9	Shift Pulse period	2			Ts
t10	Shift Pulse pulse width low	1			Ts
t11	Shift Pulse pulse width high	1			Ts
t12	FPDAT[7:0] setup to Shift Pulse falling edge	1			Ts
t13	FPDAT[7:0] hold to Shift Pulse falling edge	1			Ts
t14	Line Pulse falling edge to Shift Pulse rising edge	23			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = $t3_{min}$ - 9Ts
3. $t3_{min}$ = [((REG[04h] bits 6-0)+1) x 8 + ((REG[08h] bits 4-0) + 4) x 8]Ts
4. $t6_{min}$ = [(REG[08h] bits 4-0) x 8 + 1]Ts
5. $t7_{min}$ = [(REG[08h] bits 4-0) x 8 + 10]Ts

7.3.8 Dual Monochrome 8-Bit Panel Timing



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

Figure 7-20: Dual Monochrome 8-Bit Panel Timing

VDP =	Vertical Display Period	= (REG[06h] bits 1-0, REG[05h] bits 7-0) + 1 Lines
VNDP =	Vertical Non-Display Period	= REG[0Ah] bits 5-0 Lines
HDP =	Horizontal Display Period	= ((REG[04h] bits 6-0) + 1) x 8Ts
HNDP =	Horizontal Non-Display Period	= (REG[08h] + 4) x 8Ts

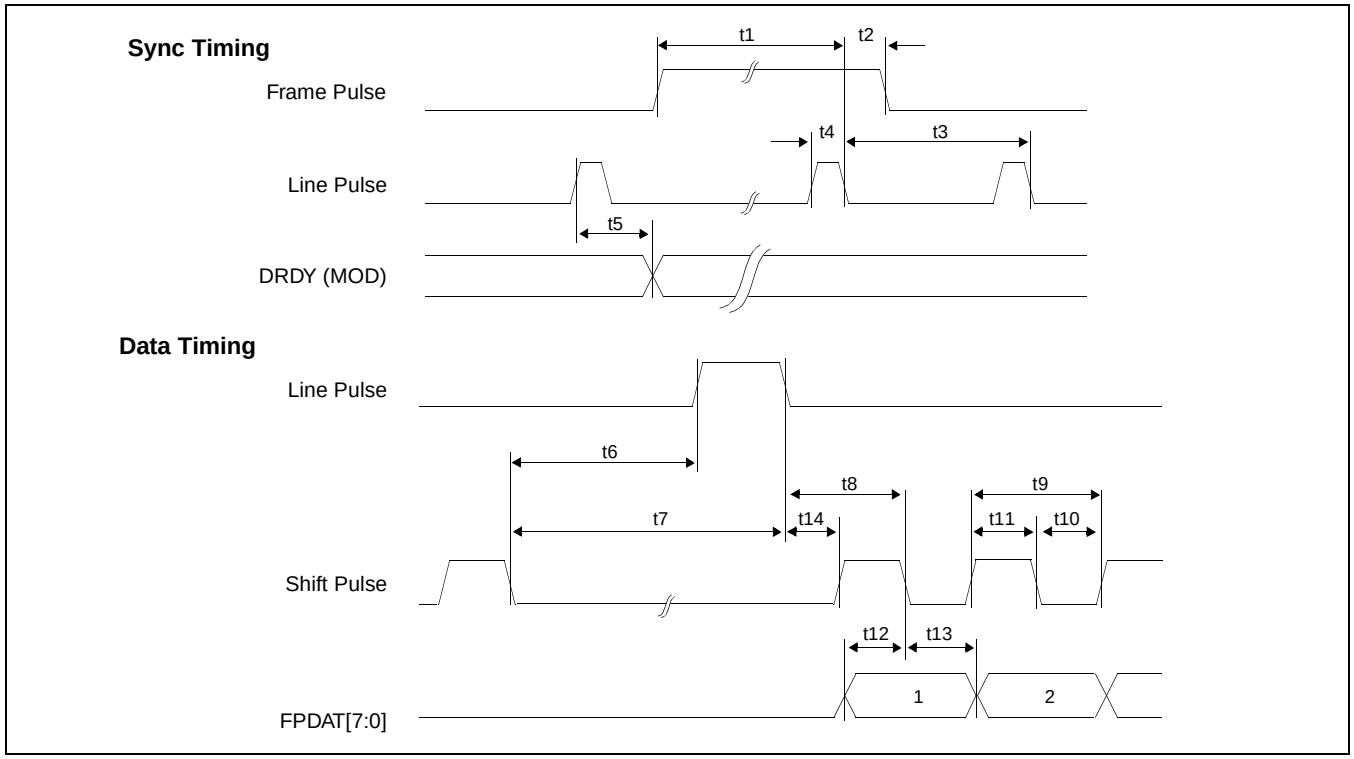
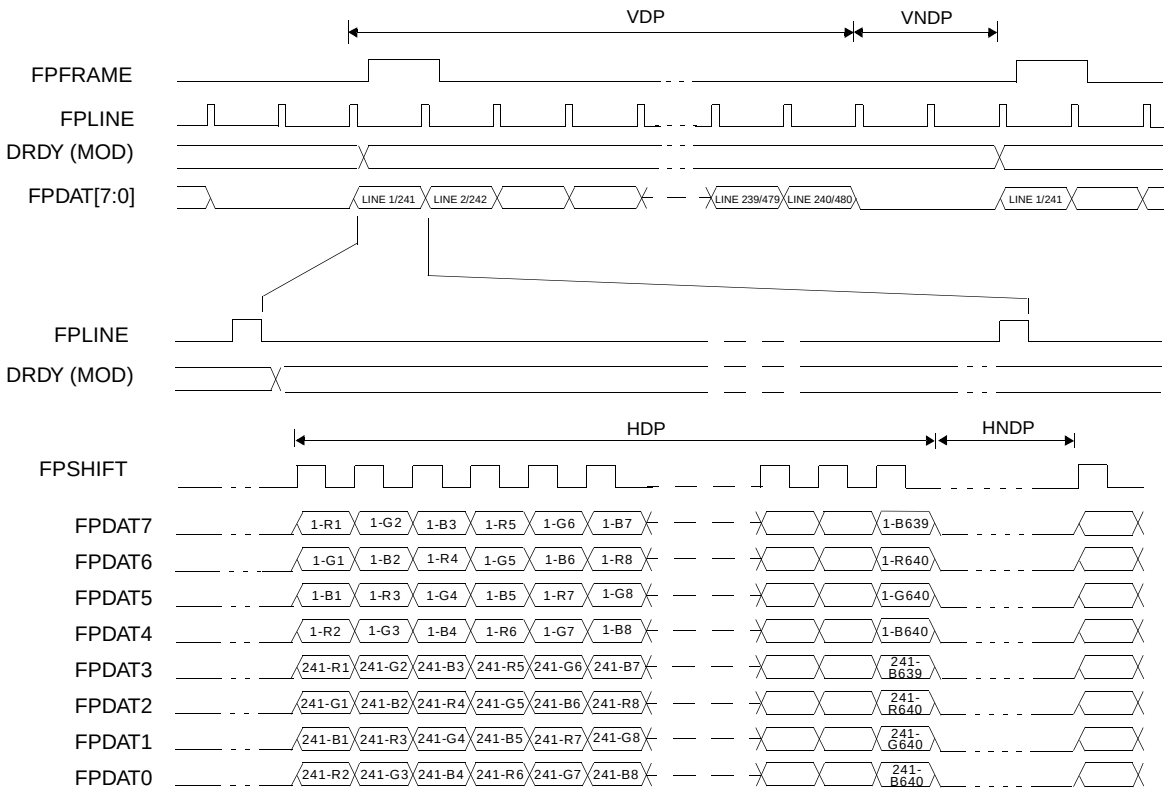


Figure 7-21: Dual Monochrome 8-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t3	Line Pulse period	note 3			
t4	Line Pulse pulse width	9			Ts
t5	MOD delay from Line Pulse falling edge	1			Ts
t6	Shift Pulse falling edge to Line Pulse rising edge	note 5			
t7	Shift Pulse falling edge to Line Pulse falling edge	note 6			
t8	Line Pulse falling edge to Shift Pulse falling edge	t14 + 4			Ts
t9	Shift Pulse period	8			Ts
t10	Shift Pulse pulse width low	4			Ts
t11	Shift Pulse pulse width high	4			Ts
t12	FPDAT[7:0] setup to Shift Pulse falling edge	4			Ts
t13	FPDAT[7:0] hold to Shift Pulse falling edge	4			Ts
t14	Line Pulse falling edge to Shift Pulse rising edge	39			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = $t3_{min}$ - 9Ts
3. $t3_{min}$ = [(((REG[04h] bits 6-0)+1) x 8 + ((REG[08h] bits 4-0) + 4) x 8) x 2]Ts
5. $t6_{min}$ = [((REG[08h] bits 4-0) x 2)x 8 + 20]Ts
6. $t7_{min}$ = [((REG[08h] bits 4-0) x 2)x 8 + 29]Ts

7.3.9 Dual Color 8-Bit Panel Timing



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

Figure 7-22: Dual Color 8-Bit Panel Timing

VDP =	Vertical Display Period	= (REG[06h] bits 1-0, REG[05h] bits 7-0) + 1 Lines
VNDP =	Vertical Non-Display Period	= REG[0Ah] bits 5-0 Lines
HDP =	Horizontal Display Period	= ((REG[04h] bits 6-0) + 1) x 8Ts
HNDP =	Horizontal Non-Display Period	= (REG[08h] + 4) x 8Ts

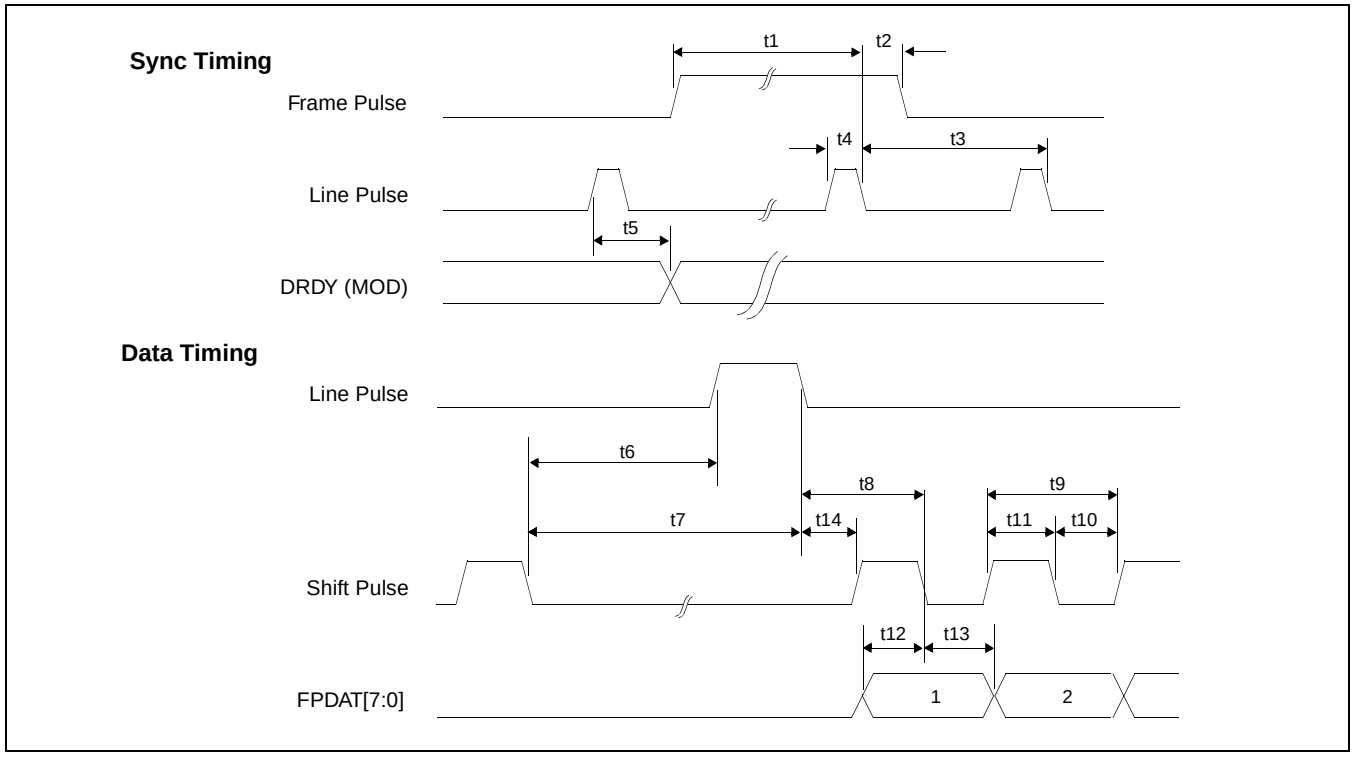


Figure 7-23: Dual Color 8-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Frame Pulse setup to Line Pulse falling edge	note 2			(note 1)
t2	Frame Pulse hold from Line Pulse falling edge	9			Ts
t3	Line Pulse period	note 3			
t4	Line Pulse pulse width	9			Ts
t5	MOD delay from Line Pulse falling edge	1			Ts
t6	Shift Pulse falling edge to Line Pulse rising edge	note 5			
t7	Shift Pulse falling edge to Line Pulse falling edge	note 6			
t8	Line Pulse falling edge to Shift Pulse falling edge	t14 + 1			Ts
t9	Shift Pulse period	2			Ts
t10	Shift Pulse pulse width low	1			Ts
t11	Shift Pulse pulse width high	1			Ts
t12	FPDAT[7:0] setup to Shift Pulse falling edge	1			Ts
t13	FPDAT[7:0] hold to Shift Pulse falling edge	1			Ts
t14	Line Pulse falling edge to Shift Pulse rising edge	39			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = $t3_{min}$ - 9Ts
3. $t3_{min}$ = [(((REG[04h] bits 6-0)+1) x 8 + ((REG[08h] bits 4-0) + 4) x 8) x 2]Ts
5. $t6_{min}$ = [((REG[08h] bits 4-0) x 2)x 8 + 17]Ts
6. $t7_{min}$ = [((REG[08h] bits 4-0) x 2)x 8 + 26]Ts

7.3.10 9/12-Bit TFT/D-TFD Panel Timing

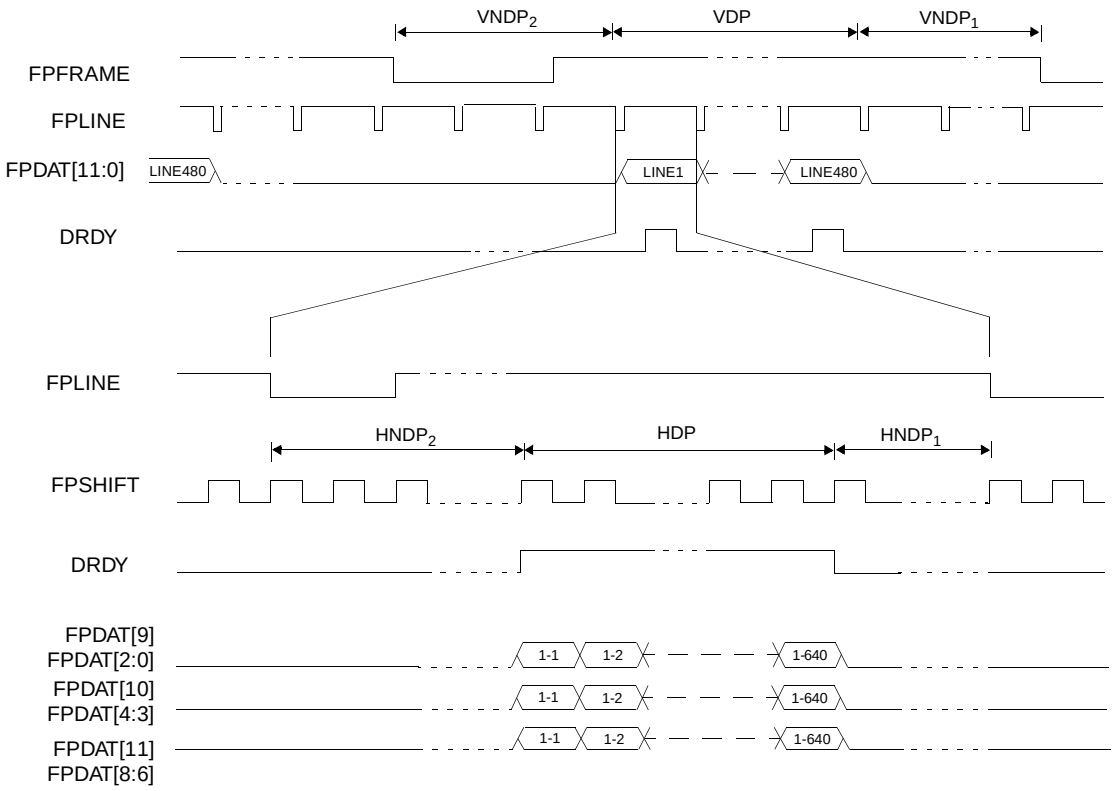


Figure 7-24: 12-Bit TFT/D-TFD Panel Timing

VDP =	Vertical Display Period	= (REG[06h] bits 1-0, REG[05h] bits 7-0) + 1 Lines
VNDP =	Vertical Non-Display Period	= VNDP1 + VNDP2 = (REG[0Ah] bits 5-0) Lines
VNDP1 =	Vertical Non-Display Period 1	= REG[09h] bits 5-0 Lines
VNDP2 =	Vertical Non-Display Period 2	= (REG[0Ah] bits 5-0) - (REG[09Ah] bits 5-0) Lines
HDP =	Horizontal Display Period	= ((REG[04h] bits 6-0) + 1) x 8Ts
HNDP =	Horizontal Non-Display Period	= HNDP1 + HNDP2 = (REG[08h] + 4) x 8Ts
HNDP1=	Horizontal Non-Display Period 1	= ((REG[07h] bits4-0) x 8) +16Ts
HNDP2=	Horizontal Non-Display Period 2	= (((REG[08h] bits4-0) - (REG[07h] bits 4-0)) x 8) +16Ts

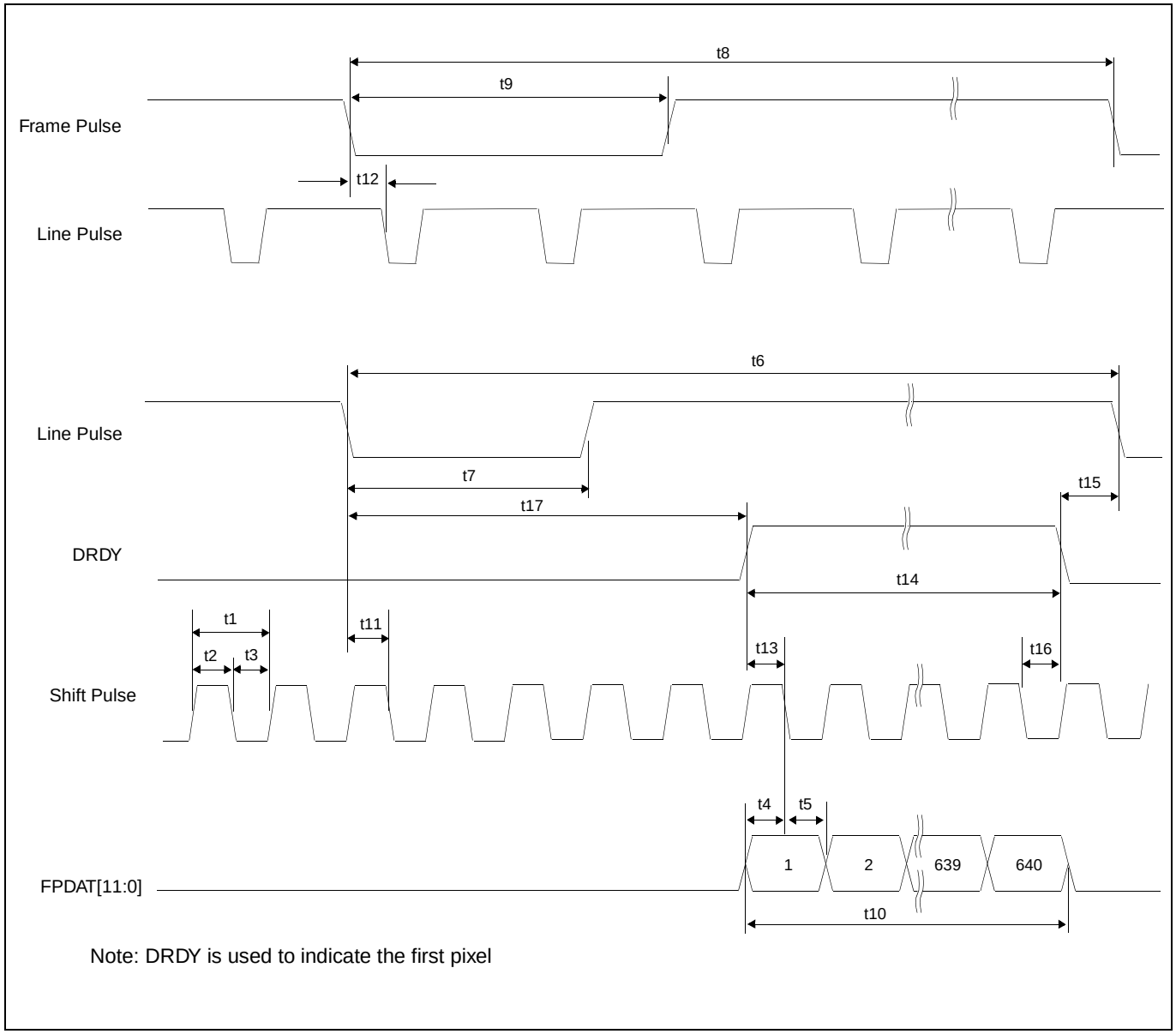


Figure 7-25: TFT/D-TFD A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Shift Pulse period	1			(note 1)
t2	Shift Pulse pulse width high	0.5			Ts
t3	Shift Pulse pulse width low	0.5			Ts
t4	Data setup to Shift Pulse falling edge	0.5			Ts
t5	Data hold from Shift Pulse falling edge	0.5			Ts
t6	Line Pulse cycle time	note 2			
t7	Line Pulse pulse width low	9			Ts
t8	Frame Pulse cycle time	note 3			
t9	Frame Pulse pulse width low	2t6			
t10	Horizontal display period	note 4			
t11	Line Pulse setup to Shift Pulse falling edge	0.5			Ts
t12	Frame Pulse falling edge to Line Pulse falling edge phase difference	t6 - 18Ts			
t13	DRDY to Shift Pulse falling edge setup time	0.5			Ts
t14	DRDY pulse width	note 5			
t15	DRDY falling edge to Line Pulse falling edge	note 6			
t16	DRDY hold from Shift Pulse falling edge	0.5			Ts
t17	Line Pulse Falling edge to DRDY active	note 7		250	

1. Ts = pixel clock period
2. t6min = [((REG[04h] bits 6-0)+1) x 8 + ((REG[08h] bits 4-0)+4) x 8] Ts
3. t8 min = [((REG[06h] bits 1-0, REG[05h] bits 7-0)+1) + (REG[0Ah] bits 6-0)] Lines
4. t10min = [((REG[04h] bits 6-0)+1) x 8] Ts
5. t14min = [((REG[04h] bits 6-0)+1) x 8] Ts
6. t15min = [(REG[07h] bits 4-0) x 8 + 16] Ts
7. t17min = [(REG[08h] bits 4-0) - (REG[07]) x 8 + 16] Ts

8 Registers

8.1 Register Mapping

The S1D13704 registers are located in the upper 32 bytes of the 64K byte S1D13704 address range. The registers are accessible when CS# = 0 and AB[15:0] are in the range FFE0h through FFFFh.

8.2 Register Descriptions

Unless specified otherwise, all register bits are reset to 0 during power up.

REG[00h] Revision Code Register
Address = FFE0h — Read Only

Product Code Bit 5	Product Code Bit 4	Product Code Bit 3	Product Code Bit 2	Product Code Bit 1	Product Code Bit 0	Revision Code Bit 1	Revision Code Bit 0

bits 7-2 Product Code
This is a read-only register that indicates the product code of the chip. The product code is 000110.

bits 1-0 Revision Code
This is a read-only register that indicates the revision code of the chip. The revision code is 00.

REG[01h] Mode Register 0
Address = FFE1h — Read/Write.

TFT/STN	Dual/Single	Color/Mono	FPLine Polarity	FPFrame Polarity	Mask FPSHIFT	Data Width Bit 1	Data Width Bit 0

bit 7 TFT/STN
When this bit = 0, STN (passive) panel mode is selected. When this bit = 1, TFT/D-TFD panel mode is selected. If TFT/D-TFD panel mode is selected, Dual/Single (REG[01h] bit 6) and Color/Mono (REG[01h] bit5) are ignored. See Table 8-1: “Panel Data Format” below.

bit 6 Dual/Single
When this bit = 0, Single LCD panel drive is selected. When this bit = 1, Dual LCD panel drive is selected. See Table 8-1: “Panel Data Format” below.

bit 5 Color/Mono
When this bit = 0, Monochrome LCD panel drive is selected. When this bit = 1, Color LCD panel drive is selected. See Table 8-1: “Panel Data Format” below.

bit 4 FPLINE Polarity
This bit controls the polarity of FPLINE in TFT/D-TFD mode (no effect in passive panel mode). When this bit = 0, FPLINE is active low. When this bit = 1, FPLINE is active high.

bit 3 FPFRAME Polarity
This bit controls the polarity of FPFRAME in TFT/D-TFD mode (no effect in passive panel mode). When this bit = 0, FPFRAME is active low. When this bit = 1, FPFRAME is active high.

bit 2 Mask FPSHIFT
FPSHIFT is masked during non-display periods if either of the following two criteria is met:

1. Color passive panel is selected (REG[01h] bit 5 = 1)
2. This bit (REG[01h] bit 2) = 1

bits 1-0 Data Width Bits [1:0]
These bits select the display data format. See Table 8-1: “Panel Data Format” below.

Table 8-1: Panel Data Format

TFT/STN REG[01h] bit 7	Color/Mono REG[01h] bit 5	Dual/Single REG[01h] bit 6	Data Width Bit 1 REG[01h] bit 1	Data Width Bit 0 REG[01h] bit 0	Function
0	0	0	0	0	Mono Single 4-bit passive LCD
				1	Mono Single 8-bit passive LCD
			1	0	reserved
				1	reserved
		1	0	0	reserved
				1	Mono Dual 8-bit passive LCD
			1	0	reserved
				1	reserved
	1	0	0	0	Color Single 4-bit passive LCD
				1	Color Single 8-bit passive LCD format 1
			1	0	reserved
				1	Color Single 8-bit passive LCD format 2
		1	0	0	reserved
				1	Color Dual 8-bit passive LCD
			1	0	reserved
				1	reserved
1	X (don’t care)			0	9-bit TFT/D-TFD panel
				1	12-bit TFT/D-TFD panel

REG[02h] Mode Register 1 Address = FFE2h							Read/Write.
Bit-Per-Pixel Bit 1	Bit-Per-Pixel Bit 0	High Performance	Input Clock divide (CLKI/2)	Display Blank	Frame Repeat	Hardware Video Invert Enable	Software Video Invert

bits 7-6 Bit-Per-Pixel Bits [1:0]

These bits select the color or gray-shade depth (Display Mode).

Table 8-2: Gray Shade/Color Mode Selection

Color/Mono **REG[01h] bit 6**	**Bit-Per-Pixel Bit 1** **REG[02h] bit 7**	**Bit-Per-Pixel Bit 0** **REG[02h] bit 6**	**Display Mode**	
0	0	0	2 Gray shade	1 bit-per-pixel
		1	4 Gray shade	2 bit-per-pixel
	1	0	16 Gray shade	4 bit-per-pixel
		1	reserved	
1	0	0	2 Colors	1 bit-per-pixel
		1	4 Colors	2 bit-per-pixel
	1	0	16 Colors	4 bit-per-pixel
		1	256 Colors	8 bit-per-pixel

bit 5 High Performance (Landscape Modes Only)

When this bit = 0, the internal Memory clock (MCLK) is a divided-down version of the Pixel clock (PCLK). The denominator is dependent on the bit-per-pixel mode - see the table below.

Table 8-3: High Performance Selection

High Performance	**BPP Bit 1**	**BPP Bit 0**	**Display Modes**	
0	0	0	MClk = PClk/8	1 bit-per-pixel
		1	MClk = PClk/4	2 bit-per-pixel
	1	0	MClk = PClk/2	4 bit-per-pixel
		1	MClk = PClk	8 bit-per-pixel
1	X	X	MClk = PClk	

When this bit = 1, MCLK is fixed to the same frequency as PCLK for all bit-per-pixel modes. This provides a faster screen update performance in 1, 2, 4 bit-per-pixel modes, but also increases power consumption. This bit can be set to 1 just before a major screen update, then set back to 0 to save power after the update. This bit has no effect in Swivel-View mode. Refer to REG[1Bh] SwivelView Mode Register on page 68 for SwivelView mode clock selection.

bit 4 Input Clock Divide

When this bit = 0, the operating clock(CLK) is same as the input clock (CLKI). When this bit = 1, CLK = CLKI/2.

In landscape mode PCLK=CLK and MCLK is selected as per Table 8-3: “High Performance Selection”.

In SwivelView mode MCLK and PCLK are derived from CLK as shown in Table 8-9: “Selection of PCLK and MCLK in SwivelView Mode,” on page 69.

bit 3 Display Blank

This bit blanks the display image. When this bit = 1, the display is blanked (FPDAT lines to the panel are driven low). When this bit = 0, the display is enabled.

bit 2 Frame Repeat (EL support)

This feature is used to improve Frame Rate Modulation of EL panels. When this bit = 1, an internal frame counter runs from 0 to 3FFFFh. When the frame counter rolls over, the modulated image pattern is repeated (every 1 hour when the frame rate is 72Hz). When this bit = 0, the modulated image pattern is never repeated.

bit 1 Hardware Video Invert Enable

In passive panel modes (REG[01h] bit 7 = 0) FPDAT11 is available as either GPIO4 or hardware video invert. When this bit = 1, Hardware Video Invert is enabled via the FPDAT11 pin. When this bit = 0, FPDAT11 operates as GPIO4. See Table 8-4: “Inverse Video Mode Select Options” below.

Note
Video data is inverted after the Look-Up Table.

bit 0 Software Video Invert

When this bit = 1, Inverse video mode is selected. When this bit = 0, standard video mode is selected. See Table 8-4: “Inverse Video Mode Select Options” below.

Note
Video data is inverted after the Look-Up Table.

Table 8-4: Inverse Video Mode Select Options

Hardware Video Invert Enable	Software Video Invert (Passive and Active Panels)	FPDAT11 (Passive Panels Only)	Video Data
0	0	X	Normal
0	1	X	Inverse
1	X	0	Normal
1	X	1	Inverse

REG[03h] Mode Register 2 Address = FFE3h							Read/Write
Look-Up Table Bypass	n/a	n/a	n/a	LCDPWR Override	Hardware Power Save Enable	Software Power Save Bit 1	Software Power Save Bit 0

bit 7 Look-Up Table Bypass

When the Look-Up Table Bypass bit = 0, the Green Look-Up Table is used for display data output in gray shade modes. When this bit = 1, the Look-Up Table is bypassed for display data output in gray shade modes (for power save purposes). See "Look-Up Table Architecture" on pag e72.

There is no effect on changing this bit in color modes. In color display mode the Look-Up Table cannot be bypassed.

bit 3 LCDPWR Override

This bit is used to override the panel on/off sequencing logic. When this bit = 0, LCDPWR and the panel interface signals are controlled by the sequencing logic. When this bit 1, LCDPWR is forced to off and the panel interface signals are forced low immediately upon entering power save mode. See Section 7.3.2, "Power Down/Up Timing" on page 36 for further information.

bit 2 Hardware Power Save Enable

When this bit = 1 GPIO0 is used as the Hardware Power Save input pin. When this bit = 0 GPIO0 operates normally.

Table 8-5: Hardware Power Save/GPIO0 Operation

RESET# State	**Hardware Power Save Enable** REG[03h] bit 2	**GPIO0 Config** REG[18h] bit 0	**GPIO0 Status/Control** REG[19h] bit 0	**GPIO0 Operation**
0	X	X	X	
1	0	0	reads pin status	GPIO0 Input (high impedance)
1	0	1	0	GPIO0 Output = 0
1	0	1	1	GPIO0 Output = 1
1	1	X	X	Hardware Power Save Input (active high)

bits 1-0 Software Power Save Bits [1: 0]

These bits select the Power Save Mode as shown in the following table.

Table 8-6: Software Power Save Mode Selection

Bit 1	**Bit 0**	**Mode**
0	0	Software Power Save
0	1	reserved
1	0	reserved
1	1	Normal Operation

Refer to Power Save Modes on page 84 for a complete description.

REG[04h] Horizontal Panel Size Register Address = FFE4h							Read/Write
n/a	Horizontal Panel Size Bit 6	Horizontal Panel Size Bit 5	Horizontal Panel Size Bit 4	Horizontal Panel Size Bit 3	Horizontal Panel Size Bit 2	Horizontal Panel Size Bit 1	Horizontal Panel Size Bit 0

bits 6-0 Horizontal Panel Size Bits [6:0]
This register determines the horizontal resolution of the panel. This register must be programmed with a value calculated as follows:

$$\text{HorizontalPanelSizeRegister} = \left(\frac{\text{HorizontalPanelResolution(pixels)}}{8}\right) - 1$$

This register must not be set to a value less than 03h.

REG[05h] Vertical Panel Size Register (LSB) Address = FFE5h							Read/Write
Vertical Panel Size Bit 7	Vertical Panel Size Bit 6	Vertical Panel Size Bit 5	Vertical Panel Size Bit 4	Vertical Panel Size Bit 3	Vertical Panel Size Bit 2	Vertical Panel Size Bit 1	Vertical Panel Size Bit 0

.

REG[06h] Vertical Panel Size Register (MSB) Address = FFE6h							Read/Write
n/a	n/a	n/a	n/a	n/a	n/a	Vertical Panel Size Bit 9	Vertical Panel Size Bit 8

REG[05h] bits 7-0 Vertical Panel Size Bits [9:0]

REG[06h] bits 1-0 This 10-bit register determines the vertical resolution of the panel. This register must be programmed with a value calculated as follows:

$$\text{VerticalPanelSizeRegister} = \text{VerticalPanelResolution(lines)} - 1$$

3FFh is the maximum value of this register for a vertical resolution of 1024 lines.

REG[07h] FPLINE Start Position
Address = FFE7h — Read/Write

n/a	n/a	n/a	FPLINE Start Position Bit 4	FPLINE Start Position Bit 3	FPLINE Start Position Bit 2	FPLINE Start Position Bit 1	FPLINE Start Position Bit 0

bits 4-0 FPLINE Start Position

These bits are used in TFT/D-TFD mode to specify the position of the FPLINE pulse. These bits specify the delay, in 8-pixel resolution, from the end of a line of display data (FPDAT) to the leading edge of FPLINE. This register is effective in TFT/D-TFD mode only (REG[01h] bit 7 = 1). This register is programmed as follows:

$$FPLINEposition(pixels) = (REG[07h] + 2) \times 8$$

The following constraint must be satisfied:

$$REG[07h] \leq REG[08h]$$

REG[08h] Horizontal Non-Display Period
Address = FFE8h — Read/Write

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

bits 4-0 Horizontal Non-Display Period

These bits specify the horizontal non-display period in 8-pixel resolution.

$$HorizontalNonDisplayPeriod(pixels) = (REG[08h] + 4) \times 8$$

REG[09h] FPFRAME Start Position
Address = FFE9h — Read/Write

n/a	n/a	FPFRAME Start Position Bit 5	FPFRAME Start Position Bit 4	FPFRAME Start Position Bit 3	FPFRAME Start Position Bit 2	FPFRAME Start Position Bit 1	FPFRAME Start Position Bit 0

bits 5-0 FPFRAME Start Position

These bits are used in TFT/D-TFD mode to specify the position of the FPFRAME pulse. These bits specify the number of lines between the last line of display data (FPDAT) and the leading edge of FPFRAME. This register is effective in TFT/D-TFD mode only (REG[01h] bit 7 = 1).

$$FPFRAMEposition(lines) = REG[09h]$$

The contents of this register must be greater than zero and less than or equal to the Vertical Non-Display Period Register, i.e.

$$1 \leq REG[09h] \leq REG[0Ah]$$

REG[0Ah] Vertical Non-Display Period							
Address = FFEAh							Read/Write
Vertical Non-Display Status	n/a	Vertical Non-Display Period Bit 5	Vertical Non-Display Period Bit 4	Vertical Non-Display Period Bit 3	Vertical Non-Display Period Bit 2	Vertical Non-Display Period Bit 1	Vertical Non-Display Period Bit 0

bit 7 Vertical Non-Display Status
This bit =1 during the Vertical Non-Display period.

bits 5-0 Vertical Non-Display Period
These bits specify the vertical non-display period.

VerticalNonDisplayPeriod(lines) = REG[0Ah]

Note
This register should be set only once, on power-up during initialization.

.

REG[0Bh] MOD Rate Register							
Address = FFEBh							Read/Write
n/a	n/a	MOD Rate Bit 5	MOD Rate Bit 4	MOD Rate Bit 3	MOD Rate Bit 2	MOD Rate Bit 1	MOD Rate Bit 0

bits 5-0 MOD Rate Bits [5:0]
When the value of this register is 0, the MOD output signal toggles every FPFRAME. For a non-zero value, the value in this register + 1 specifies the number of FPLINEs between toggles of the MOD output signal. These bits are for passive LCD panels only.

REG[0Ch] Screen 1 Start Address Register (LSB)							
Address = FFECh							Read/Write
Screen 1 Start Address Bit 7	Screen 1 Start Address Bit 6	Screen 1 Start Address Bit 5	Screen 1 Start Address Bit 4	Screen 1 Start Address Bit 3	Screen 1 Start Address Bit 2	Screen 1 Start Address Bit 1	Screen 1 Start Address Bit 0

REG[0Dh] Screen 1 Start Address Register (MSB)							
Address = FFEDh							Read/Write
Screen 1 Start Address Bit 15	Screen 1 Start Address Bit 14	Screen 1 Start Address Bit 13	Screen 1 Start Address Bit 12	Screen 1 Start Address Bit 11	Screen 1 Start Address Bit 10	Screen 1 Start Address Bit 9	Screen 1 Start Address Bit 8

REG[0Dh] bit 6-0 Screen 1 Start Address Bits [14:0]
REG[0Ch] bit 7-0 These bits determine the **word address** of the start of Screen 1 in landscape modes or the **byte address** of the start of Screen 1 in SwivelView modes.

REG[0Dh] bit 7 Screen 1 Start Address Bit 15
This bit is for SwivelView mode only and has no effect in Landscape mode.

REG[0Fh] Screen 2 Start Address Register (LSB)							
Address = FFEFh							Read/Write
Screen 2 Start Address Bit 7	Screen 2 Start Address Bit 6	Screen 2 Start Address Bit 5	Screen 2 Start Address Bit 4	Screen 2 Start Address Bit 3	Screen 2 Start Address Bit 2	Screen 2 Start Address Bit 1	Screen 2 Start Address Bit 0

REG[10h] Screen 2 Start Address Register (MSB)							
Address = FFF0h							Read/Write
Screen 2 Start Address Bit 15	Screen 2 Start Address Bit 14	Screen 2 Start Address Bit 13	Screen 2 Start Address Bit 12	Screen 2 Start Address Bit 11	Screen 2 Start Address Bit 10	Screen 2 Start Address Bit 9	Screen 2 Start Address Bit 8

REG[10h] bit 6-0 Screen 2 Start Address Bits [14:0]

REG[0Fh] bit 7-0 These bits determine the **word address** of the start of Screen 2 in landscape modes or the **byte address** of the start of Screen 2 in SwivelView modes.

REG[10h] bit 7 Screen 2 Start Address Bit 15

This bit is for SwivelView mode only and has no effect in Landscape mode.

REG[12h] Memory Address Offset Register							
Address = FFF2h							Read/Write
Memory Address Offset Bit 7	Memory Address Offset Bit 6	Memory Address Offset Bit 5	Memory Address Offset Bit 4	Memory Address Offset Bit 3	Memory Address Offset Bit 2	Memory Address Offset Bit 1	Memory Address Offset Bit 0

bits 7-0 Memory Address Offset Bits [7:0] (Landscape Modes Only)

This register is used to create a virtual image by setting a word offset between the last address of one line and the first address of the following line. If this register is not equal to zero, then a virtual image is formed. The displayed image is a window into the larger virtual image. See Figure 8-1: “Screen-Register Relationship, Split Screen,” on page 64.

This register has no effect in SwivelView modes. See “REG[1Ch] Line Byte Count Register for SwivelView Mode” on page69.

.

REG[13h] Screen 1 Vertical Size Register (LSB) Address = FFF3h							Read/Write
Screen 1 Vertical Size Bit 7	Screen 1 Vertical Size Bit 6	Screen 1 Vertical Size Bit 5	Screen 1 Vertical Size Bit 4	Screen 1 Vertical Size Bit 3	Screen 1 Vertical Size Bit 2	Screen 1 Vertical Size Bit 1	Screen 1 Vertical Size Bit 0

REG[14h] Screen 1 Vertical Size Register (MSB) Address = FFF4h							Read/Write
n/a	n/a	n/a	n/a	n/a	n/a	Screen 1 Vertical Size Bit 9	Screen 1 Vertical Size Bit 8

REG[14h] bits 1-0 Screen 1 Vertical Size Bits [9:0]

REG[13h] bits 7-0 This register is used to implement the Split Screen feature of the S1D13704. These bits determine the height (in lines) of Screen 1. On reset this register is set to 0h.

In landscape modes, if this register is programmed with a value, n, where n is less than the Vertical Panel Size (REG[06h], REG[05h]), then lines 0 to n of the panel contain Screen 1 and lines n+1 to REG[06h], REG[05h] of the panel contain Screen 2. See Figure 8-1: “Screen-Register Relationship, Split Screen,” on page 64. If Split Screen is not desired, this register must be programmed greater than, or equal to the Vertical Panel Size, REG[06h] and REG[05h].

In SwivelView modes this register must be programmed greater than, or equal to the Vertical Panel Size, REG[06h] and REG[05h]. See “SwivelView™” on page 79.

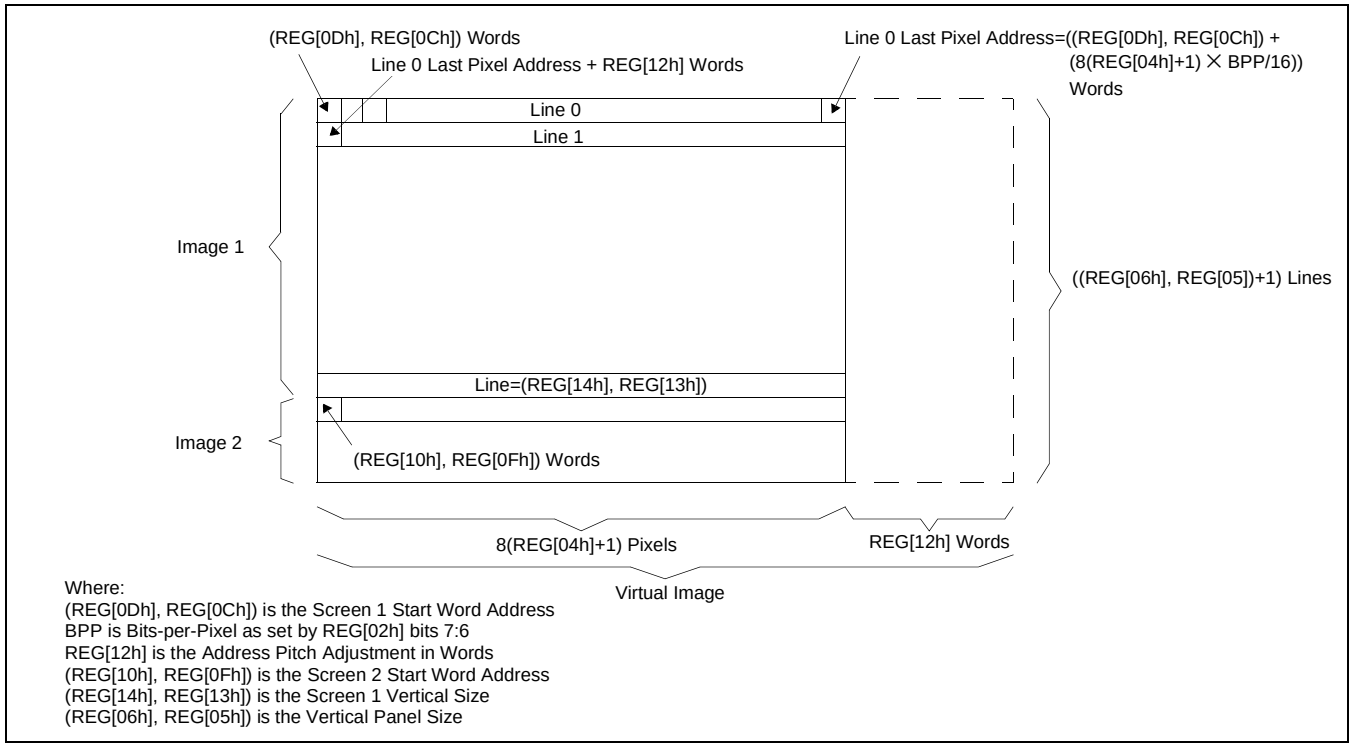


Figure 8-1: Screen-Register Relationship, Split Screen

Consider an example where REG[14h], REG[13h]= 0CEh for a 320x240 display system. The upper 207 lines (CEh + 1) of the panel show an image from the Screen 1 Start Word Address. The remaining 33 lines show an image from the Screen 2 StartWord Address.

REG[15h] Look-Up Table Address Register Address = FFF5h							Read/Write
n/a	n/a	RGB Index Bit 1	RGB Index Bit 0	Look-Up Table Address Bit 3	Look-Up Table Address Bit 2	Look-Up Table Address Bit 1	Look-Up Table Address Bit 0

The S1D13704 has three 16-position, 4-bit wide Look-Up Tables, one each for red, green, and blue. Refer to *"Look-Up Table Architecture"* for details. This register selects which Look-Up Table position is read/write accessible through the Look-Up Table Data Register (REG[17h]).

bits 5-4 RGB Index Bits [1:0]
These bits select between the Red, Green, and Blue Look-Up Tables, and Auto-Increment mode. The Green Look-Up Table is used in monochrome mode with these bits set to 10b. See Note below.

bits 3-0 Look-Up Table Address Bits [3:0]
These 4 bits select one of the 16 positions in the selected Look-Up Table. These bits are automatically changed as the Look-Up Table Data Register is accessed. See Note below.

Note

Accesses to the Look-Up Table Data Register automatically increment a pointer into the RGB Look-Up Tables. The pointer sequence varies as shown in the table below.

Table 8-7: Look-Up Table Access

REG[01h]	**REG[15h]**		**Look-Up Table Selected**	**Pointer Sequence**
bit 5	**bit 5**	**bit 4**		
0	1	0	Green/Gray Look-Up Table	G[n], G[n+1], G[n+2],...
1	0	0	Auto-Increment	R[n], G[n], B[n], R[n+1], G[n+1],...
1	0	1	Red Look-Up Table	R[n], R[n+1], R[n+2],...
1	1	0	Green/Gray Look-Up Table	G[n], G[n+1], G[n+2],...
1	1	1	Blue Look-Up Table	B[n], B[n+1], B[n+2],...

In Auto-Increment mode, writing the Look-Up Table Address Register automatically sets the pointer to the Red Look-Up Table. For example, writing a value 03 into the Look-Up Table Address Register selects Auto-Increment mode and sets the pointer to R[3]. Subsequent accesses to the Look-Up Table Data Register move the pointer onto G[3], B[3], R[4], etc.

.

REG[16h] Look-Up Table Bank Select Register Address = FFF6h							Read/Write
n/a	n/a	Red Bank Select Bit 1	Red Bank Select Bit 0	Green Bank Select Bit 1	Green Bank Select Bit 0	Blue Bank Select Bit 1	Blue Bank Select Bit 0

bits 7-6 n/a

bits 5-4 Red Bank Select Bits [1:0]
In 1 bit-per-pixel (bpp) color mode the lower 8 positions of the Red Look-UpTable is arranged into four banks, each with two positions. These two bits select which bank is used for display data.

In 2 bpp color mode the 16 position Red Look-Up Table is arranged into four banks, each with four positions. These two bits select which bank is used for display data.

These bits have no effect in 4 bpp color/gray modes.

In 8 bpp color mode the 16 position, Red Look-Up Table is arranged into two banks, each with eight positions. Red Bank Select bit 0 selects which bank is used for display data.

bits 3-2 Green Bank Select Bits [1:0]
In 1 bit-per-pixel (bpp) color/gray mode the lower 8 positions of the Green Look-Up Table is arranged into four banks, each with two positions. These two bits select which bank is used for display data.

In 2 bpp color/gray mode, the 16 position Green Look-UpTable is arranged into four banks, each with four positions. These two bits select which bank is used for display data.

These bits have no effect in 4 bpp color/gray modes.

In 8 bpp color mode, the 16 position Green Look-UpTable is arranged into two banks, each with eight positions. Green Bank Select bit 0 selects which bank is used for display data.

bit 1-0 Blue Bank Select Bits [1:0]
In 1 bit-per-pixel (bpp) color mode the lower 8 positions of the Blue Look-UpTable is arranged into four banks, each with two positions. These two bits select which bank is used for display data.

In 2 bpp color mode, the 16 position Blue Look-Up Table is arranged into four banks, each with four positions. These two bits select which bank is used for display data.

These bits have no effect in 4 bpp color/gray modes.

In 8 bpp color mode, the 16 position Blue Look-Up Table is arranged into four banks, each with four positions. These two bits select which bank is used for display data.

REG[17h] Look-Up Table Data Register Address = FFF7h							Read/Write
n/a	n/a	n/a	n/a	Look-Up Table Data Bit 3	Look-Up Table Data Bit 2	Look-Up Table Data Bit 1	Look-Up Table Data Bit 0

bits 3-0 Look-Up Table Data Bits [3:0]
This register is used to read/write the RGB Look-Up Tables. This register is an aperture into the three 16-position Look-Up Tables. The Look-Up Table Address Register (REG[16h]) selects which Look-Up Table position is accessible. See REG[16h] Look-Up Table Bank Select Register on page 66.

REG[18h] GPIO Configuration Control Register Address = FFF8h							Read/Write
n/a	n/a	n/a	GPIO4 Pin IO Configuration	GPIO3 Pin IO Configuration	GPIO2 Pin IO Configuration	GPIO1 Pin IO Configuration	GPIO0 Pin IO Configuration

bits 4-0 GPIO[4:0] Pin IO Configuration
These bits determine the direction of the GPIO[4:0] pins.
When GPIOn Pin IO Configuration bit = 0, the corresponding GPIOn pin is configured as an input. The input can be read at the GPIOn Status/Control Register bit. See REG[19h] below.

When GPIOn Pin IO Configuration bit = 1, the corresponding GPIOn pin is configured as an output. The output can be controlled by writing the GPIOn Status/Control Register bit.

Note
These bits have no effect when the GPIOn pin is configured for a specific function (i.e. as FPDAT[11:8] for TFT/D-TFD operation). All unused GPIO pins must be tied to IO V_{DD}.

REG[19h] GPIO Status/Control Register Address = FFF9h							Read/Write
n/a	n/a	n/a	GPIO4 Pin IO Status	GPIO3 Pin IO Status	GPIO2 Pin IO Status	GPIO1 Pin IO Status	GPIO0 Pin IO Status

bits 4-0 GPIO[4:0] Status
When the GPIOn pin is configured as an input, the corresponding GPIO Status bit is used to read the pin input. See REG[18h] above.

When the GPIOn pin is configured as an output, the corresponding GPIO Status bit is used to control the pin output.

REG[1Ah] Scratch Pad Register Address = FFFAh							Read/Write
Scratch bit 7	Scratch bit 6	Scratch bit 5	Scratch bit 4	Scratch bit 3	Scratch bit 2	Scratch bit 1	Scratch bit 0

bits 7-0 Scratch Pad Register
This register contains general use read/write bits. These bits have no effect on hardware.

REG[1Bh] SwivelView Mode Register Address = FFFBh							Read/Write
SwivelView Mode Enable	SwivelView Mode Select	n/a	n/a	n/a	reserved	SwivelView Mode Pixel Clock Select Bit 1	SwivelView Mode Pixel Clock Select Bit 0

bit 7 SwivelView Mode Enable
When this bit = 1, SwivelView Mode is enabled. When this bit = 0, Landscape Mode is enabled.

bit 6 SwivelView Mode Select
When this bit = 0, Default SwivelView Mode is selected. When this bit = 1, Alternate SwivelView Mode is selected. See Section 12, “SwivelView™” on page 79 for further information on SwivelView Mode.

The following table shows the selection of SwivelView Mode.

Table 8-8: Selection of SwivelView Mode

SwivelView Mode Enable (REG[1Bh] bit 7)	**SwivelView Mode Select** (REG[1Bh] bit 6)	**Mode**
0	X	Landscape
1	0	Default SwivelView
1	1	Alternate SwivelView

bit 2 reserved
reserved bits must be set to 0.

bits 1-0 SwivelView Mode Pixel Clock Select Bits [1:0]
These two bits select the Pixel Clock (PCLK) source in SwivelView Mode - these bits have no effect in Landscape Mode. The following table shows the selection of PCLK and MCLK in SwivelView Mode - see Section 12, “SwivelView™” on page 79 for details.

Table 8-9: Selection of PCLK and MCLK in SwivelView Mode

SwivelView Mode Enable (REG[1Bh] bit 7)	SwivelView Mode Select (REG[1Bh] bit 6)	Pixel Clock (PCLK) Select (REG[1Bh] bits [1:0]		PCLK =	MCLK =
		Bit 1	Bit 0		
0	X	X	X	CLK	See Reg[02h] bit 5
1	0	0	0	CLK	CLK
1	0	0	1	CLK/2	CLK/2
1	0	1	0	CLK/4	CLK/4
1	0	1	1	CLK/8	CLK/8
1	1	0	0	CLK/2	CLK
1	1	0	1	CLK/2	CLK
1	1	1	0	CLK/4	CLK/2
1	1	1	1	CLK/8	CLK/4

Where CLK is CLKI (REG[02h] bit 4 = 0) or CLKI/2 (REG[02h] bit 4 = 1)

REG[1Ch] Line Byte Count Register for SwivelView Mode
Address = FFFCh — Read/Write

Line Byte Count bit 7	Line Byte Count bit 6	Line Byte Count bit 5	Line Byte Count bit 4	Line Byte Count bit 3	Line Byte Count bit 2	Line Byte Count bit 1	Line Byte Count bit 0

bits 7-0 Line Byte Count Bits [7:0]
This register is the byte count from the beginning of one line to the beginning of the next consecutive line (commonly called “stride” by programmers). This register may be used to create a virtual image in SwivelView mode.

REG[1Eh] and REG[1Fh]

REG[1Eh] and REG[1Fh] are reserved for factory S1D13704 testing and should not be written. Any value written to these registers may result in damage to the S1D13704 and/or any panel connected to the S1D13704.

9 Frame Rate Calculation

The following formulae are used to calculate the display frame rate.

TFT/D-TFD and Passive Single-Panel modes

$$\text{FrameRate} = \frac{f_{PCLK}}{(HDP + HNDP) \times (VDP + VNDP)}$$

Where: f_{PCLK} = PClk frequency (Hz)
HDP = Horizontal Display Period = ((REG[04h] bits 6-0) + 1) x 8 Pixels
HNDP = Horizontal Non-Display Period = ((REG[08h] bits 4-0) + 4) x 8 Pixels
VDP = Vertical Display Period = ((REG[06h] bits 1-0, REG[05h] bits 7-0) + 1) Lines
VNDP = Vertical Non-Display Period = (REG[0Ah] bits 5-0) Lines

Passive Dual-Panel mode

$$\text{FrameRate} = \frac{f_{PCLK}}{2 \times (HDP + HNDP) \times \left(\frac{VDP}{2} + VNDP\right)}$$

Where: f_{PCLK} = PClk frequency (Hz)
HDP = Horizontal Display Period = ((REG[04h] bits 6-0) + 1) x 8 Pixels
HNDP = Horizontal Non-Display Period = ((REG[08h] bits 4-0) + 4) x 8 Pixels
VDP = Vertical Display Period = ((REG[06h] bits 1-0, REG[05h] bits 7-0) + 1) Lines
VNDP = Vertical Non-Display Period = (REG[0Ah] bits 5-0) Lines

10 Display Data Formats

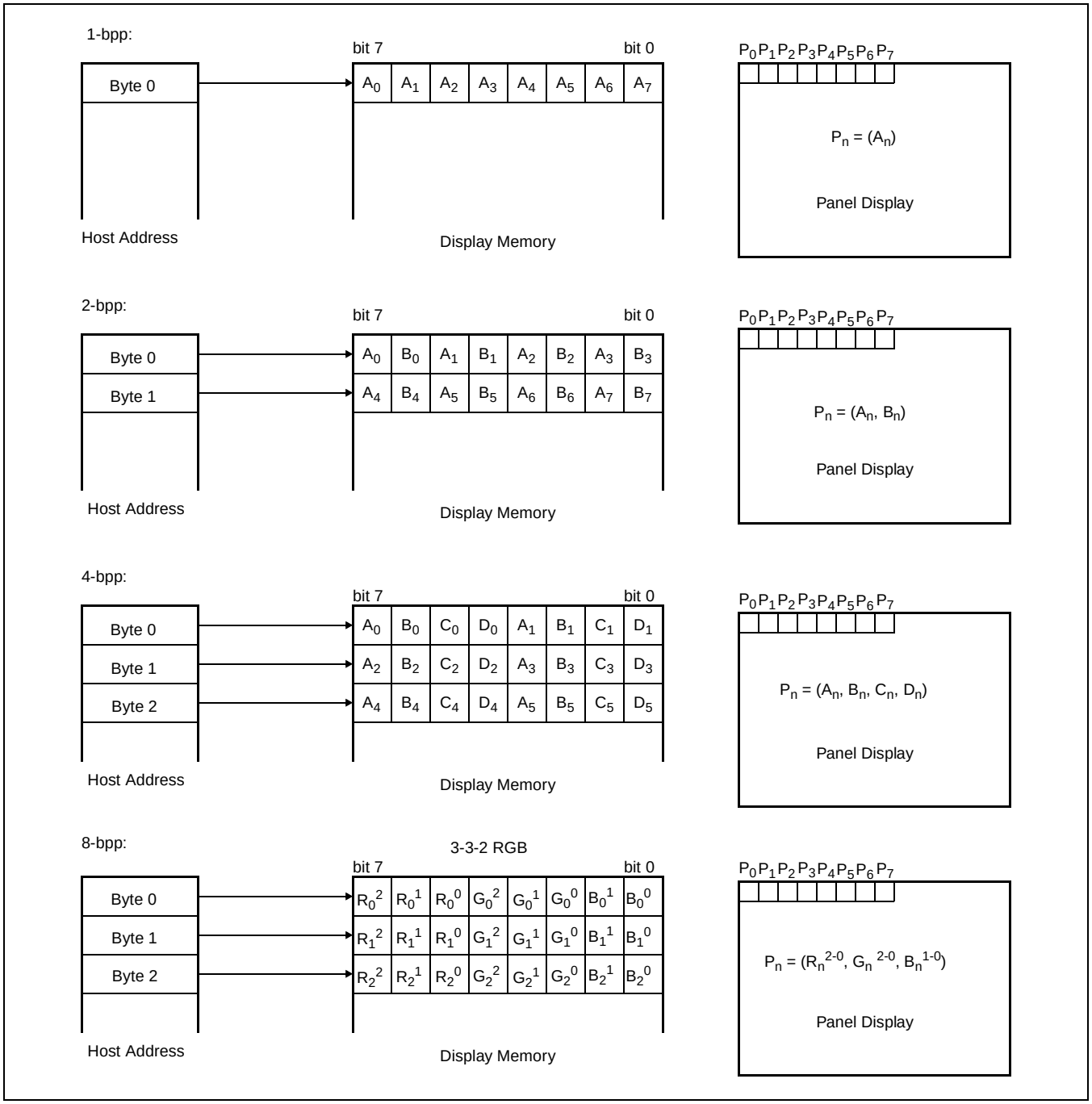


Figure 10-1: 1/2/4/8 Bit-Per-Pixel Display Data Memory Organization

11 Look-Up Table Architecture

Table 11-1: Look-Up Table Configurations

Display Mode	4-bit wide Look-Up Table		
	RED	GREEN	BLUE
2-level gray		4 banks of 2	
4-level gray		4 banks of 4	
16-level gray		1 bank of 16	
2 color	4 bank of 2	4 bank of 2	4 bank of 2
4 color	4 banks of 4	4 banks of 4	4 banks of 4
16 color	1 bank of 16	1 bank of 16	1 bank of 16
256 color	2 banks of 8	2 banks of 8	4 banks of 4

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

The following figures are intended to show the display data output path only. The CPU R/W access to the individual Look-Up Tables is not affected by the various 'banking' configurations.

11.1 Gray Shade Display Modes

2-Level Gray Shade Mode

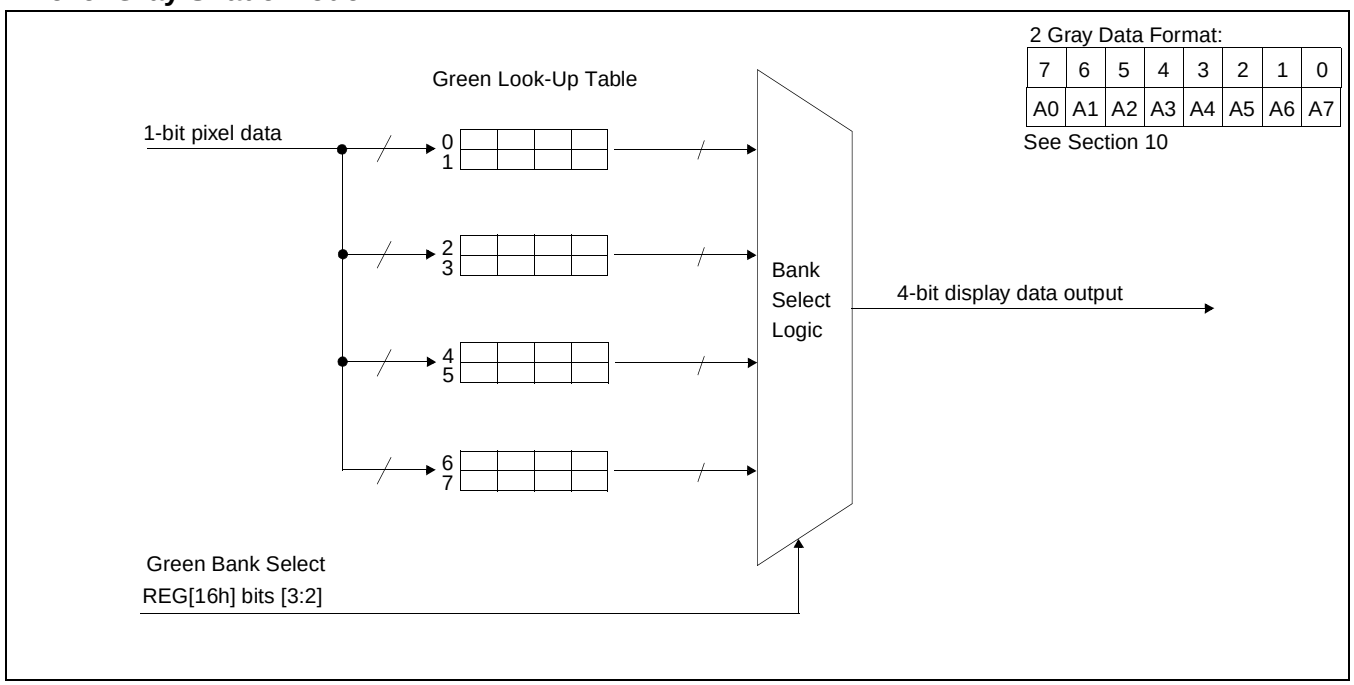


Figure 11-1: 2-Level Gray-Shade Mode Look-Up Table Architecture

4-Level Gray Shade Mode

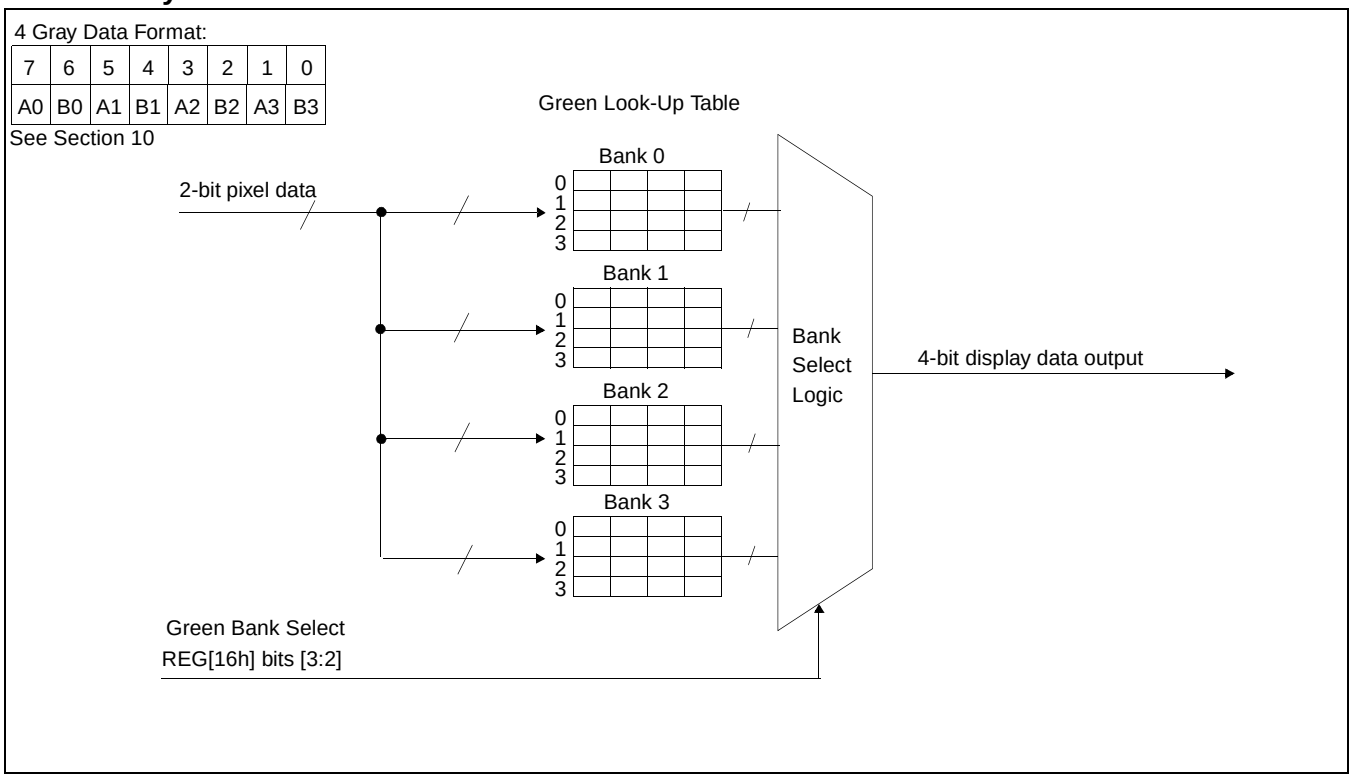


Figure 11-2: 4-Level Gray-Shade Mode Look-Up Table Architecture

16-Level Gray Shade Mode

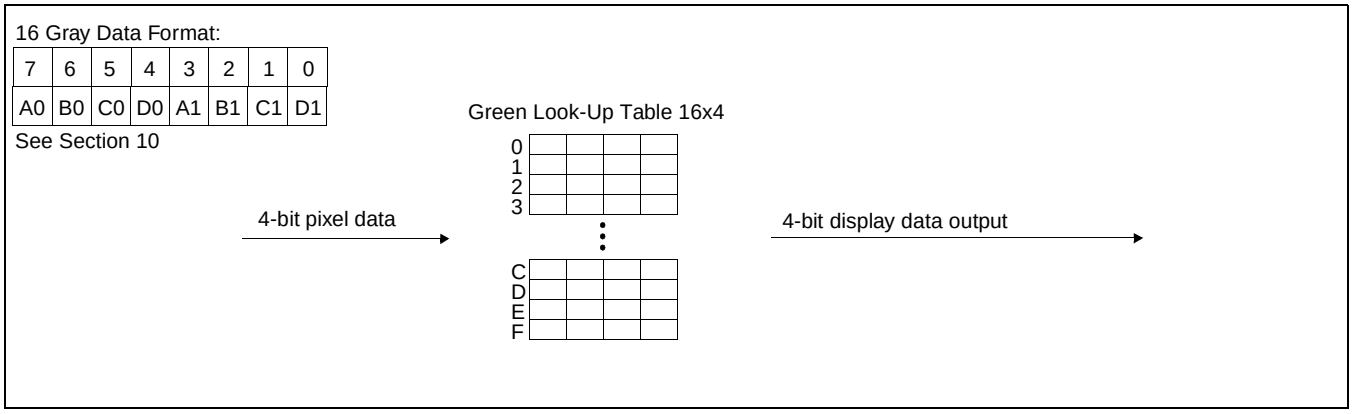


Figure 11-3: 16-Level Gray-Shade Mode Look-Up Table Architecture

Look-Up Table Bypass Mode

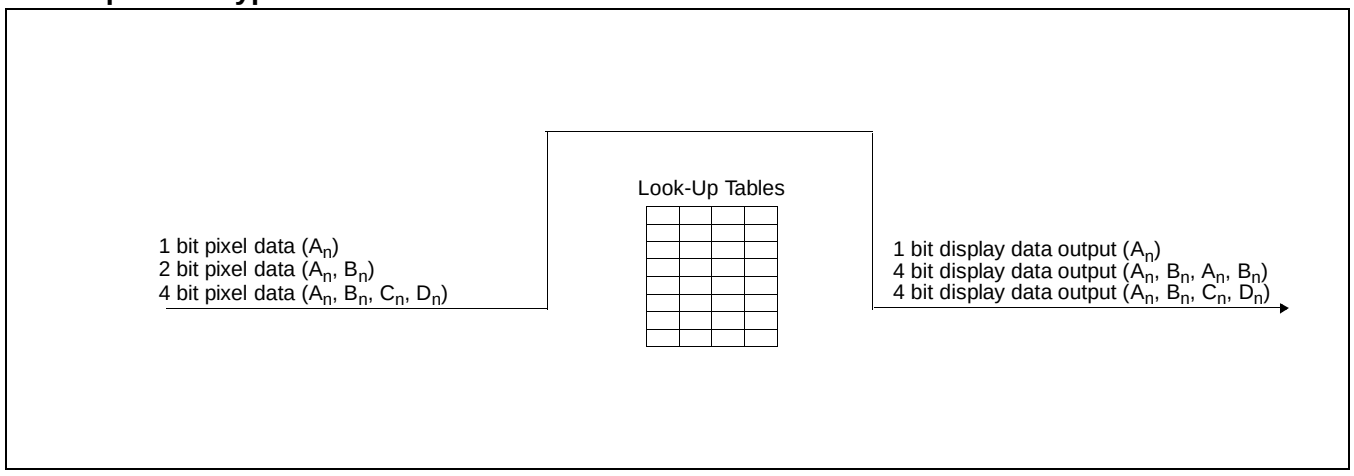


Figure 11-4: Look-Up Table Bypass Mode Architecture

Note

In 1 bit-per-pixel display mode, Look-Up Table Bypass mode will turn off the FRM circuitry and place the S1D13704 in Black-and-White mode.

In 2 bit-per-pixel mode the Display Data Output values are 0, 5, A, and F (in hex).

11.2 Color Display Modes

2-Level Color Mode

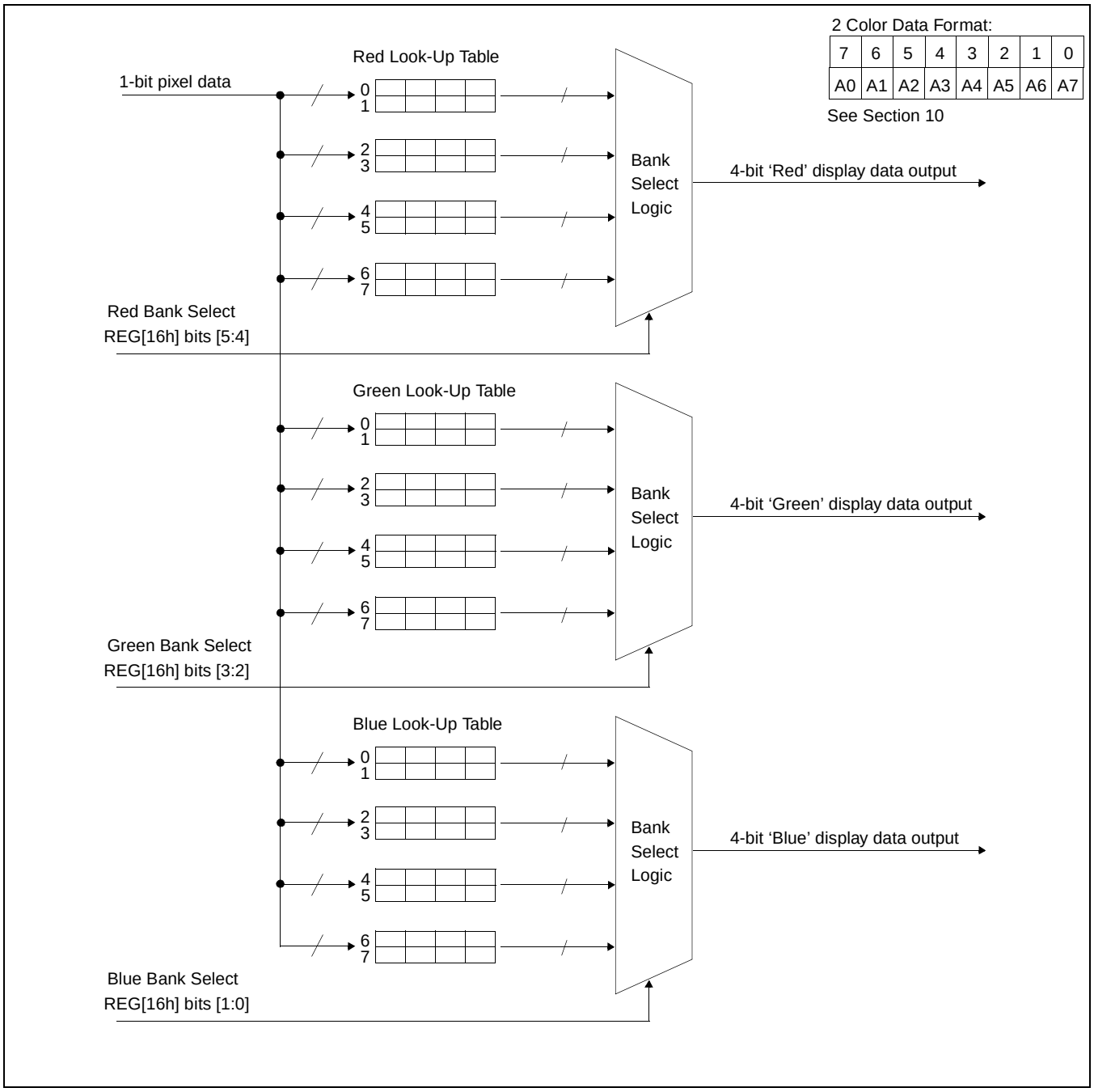


Figure 11-5: 2-Level Color Look-Up Table Architecture

4-Level Color Mode

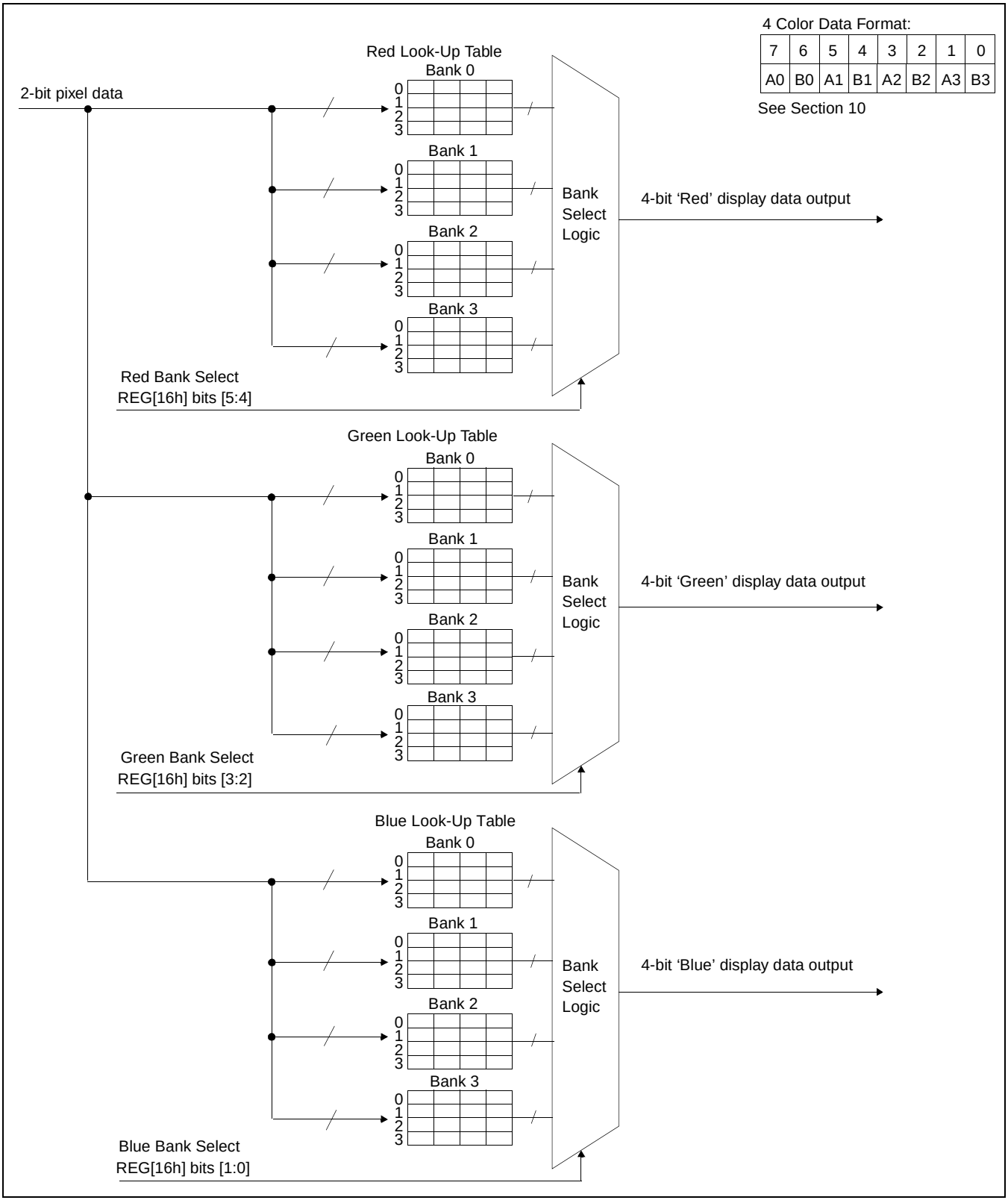


Figure 11-6: 4-Level Color Mode Look-Up Table Architecture

16-Level Color Mode

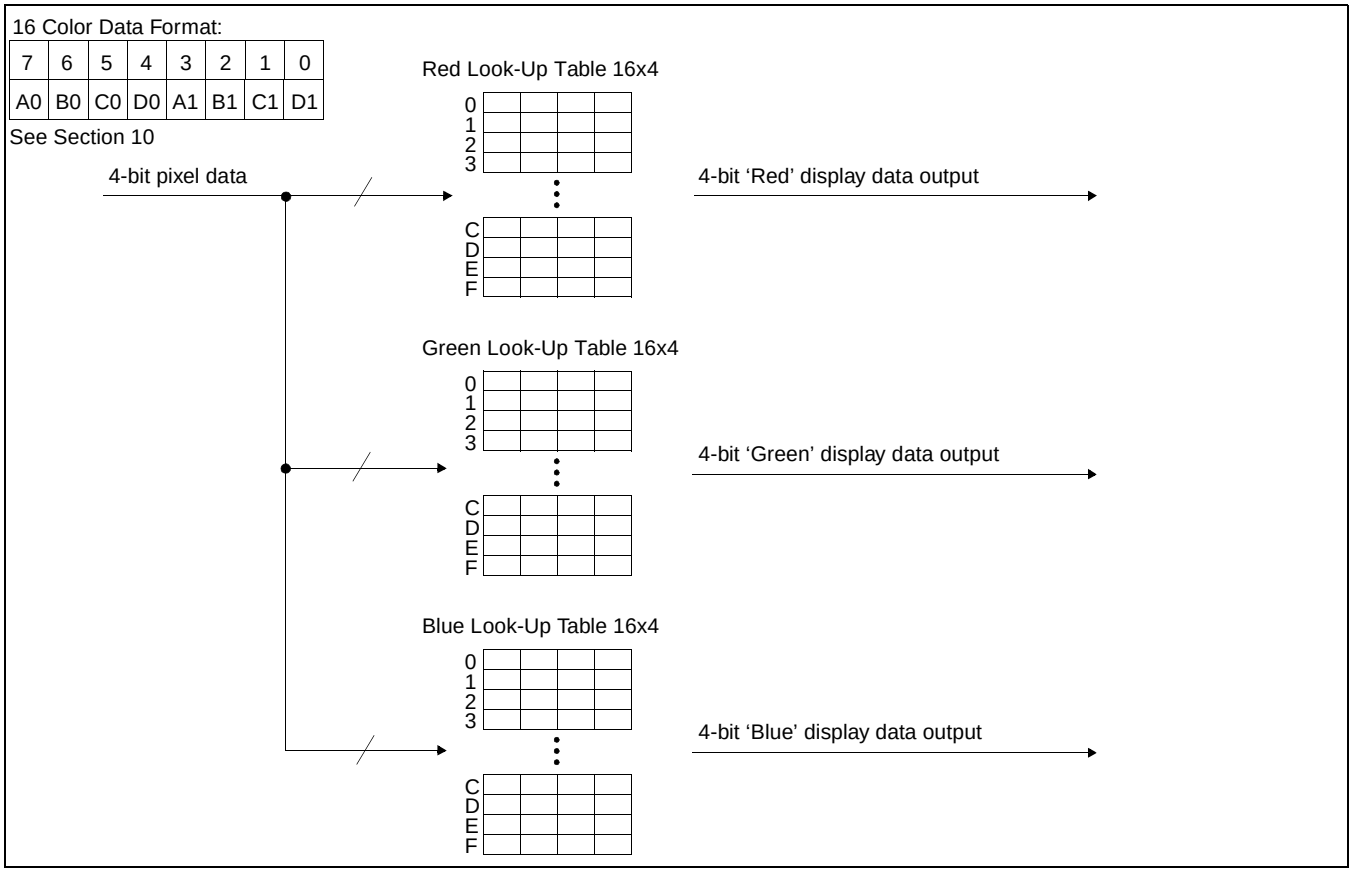


Figure 11-7: 16-Level Color Mode Look-Up Table Architecture

256-Level Color Mode

256 Color Data Format:

7	6	5	4	3	2	1	0
R2	R1	R0	G2	G1	G0	B1	B0

See Section 10

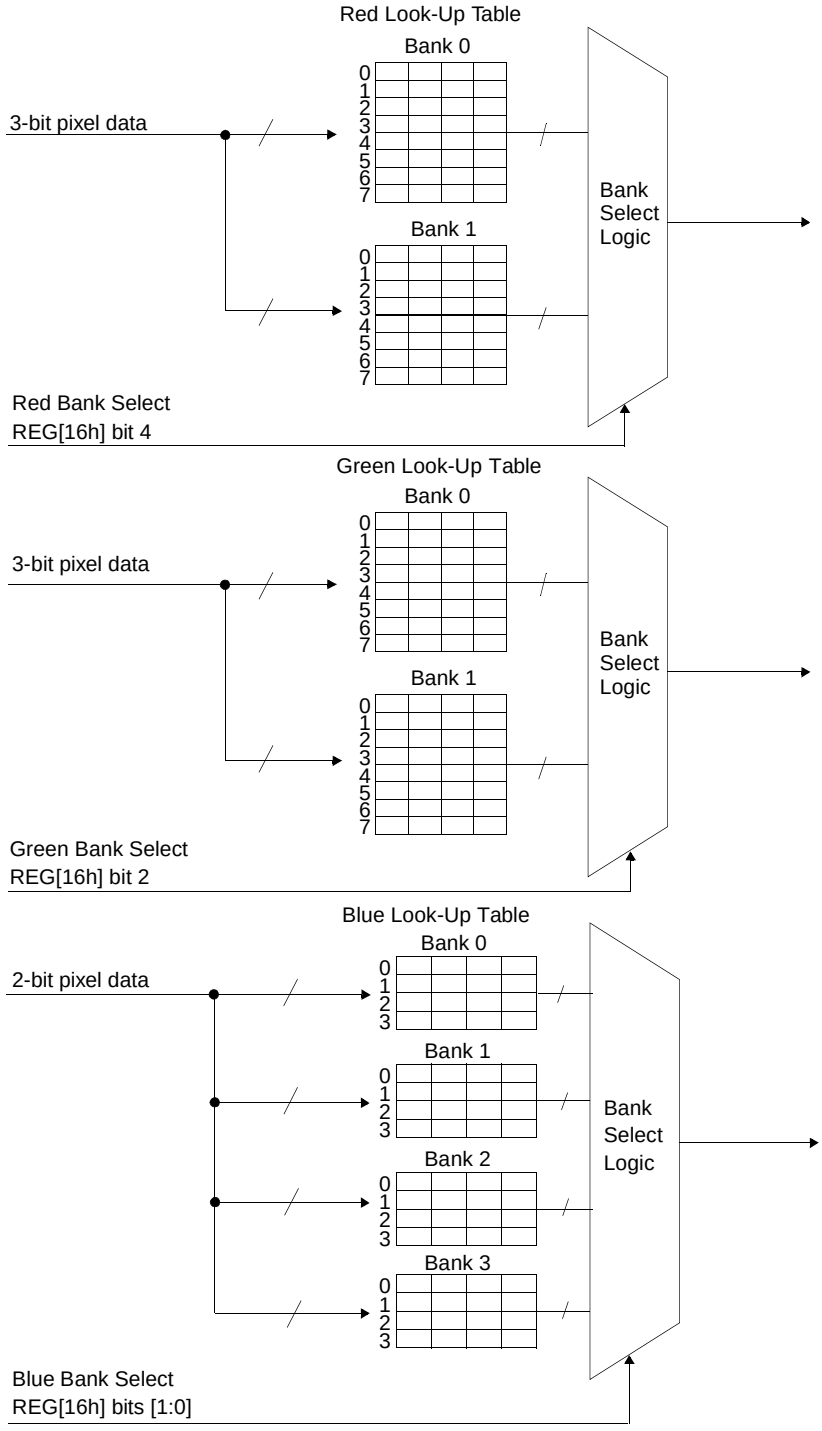


Figure 11-8: 256-Level Color Mode Look-Up Table Architecture

12 SwivelView™

Many of todays applications use the LCD panel in a portrait orientation. In this case it becomes necessary to "rotate" the displayed image. This rotation can be done by software at the expense of performance or, as with the S1D13704, it can be done by hardware with no CPU penalty.

There are two SwivelView modes: Default SwivelView and Alternate SwivelView.

12.1 Default SwivelView Mode

Default SwivelView Mode requires the portrait image width be a power of two, e.g. a 240-line panel requires a minimum virtual image width of 256. This mode should be used whenever the required virtual image can be contained within the integrated display buffer (i.e. virtual image size ≤ 40k bytes), as it consumes less power than the Alternate SwivelView mode.

For example, the panel size is 320x240 and the display mode is 4 bit-per-pixel. The virtual image size is 320x256 which can be contained within the 40k Byte display buffer.

Default SwivelView Mode also requires memory clock (MCLK) ≥ pixel clock (PCLK).

The following figure shows how the programmer sees a 240x320 image and how the image is displayed. The application image is written to the S1D13704 in the following sense: A–B–C–D. The display is refreshed by the S1D13704 in the following sense: B-D-A-C.

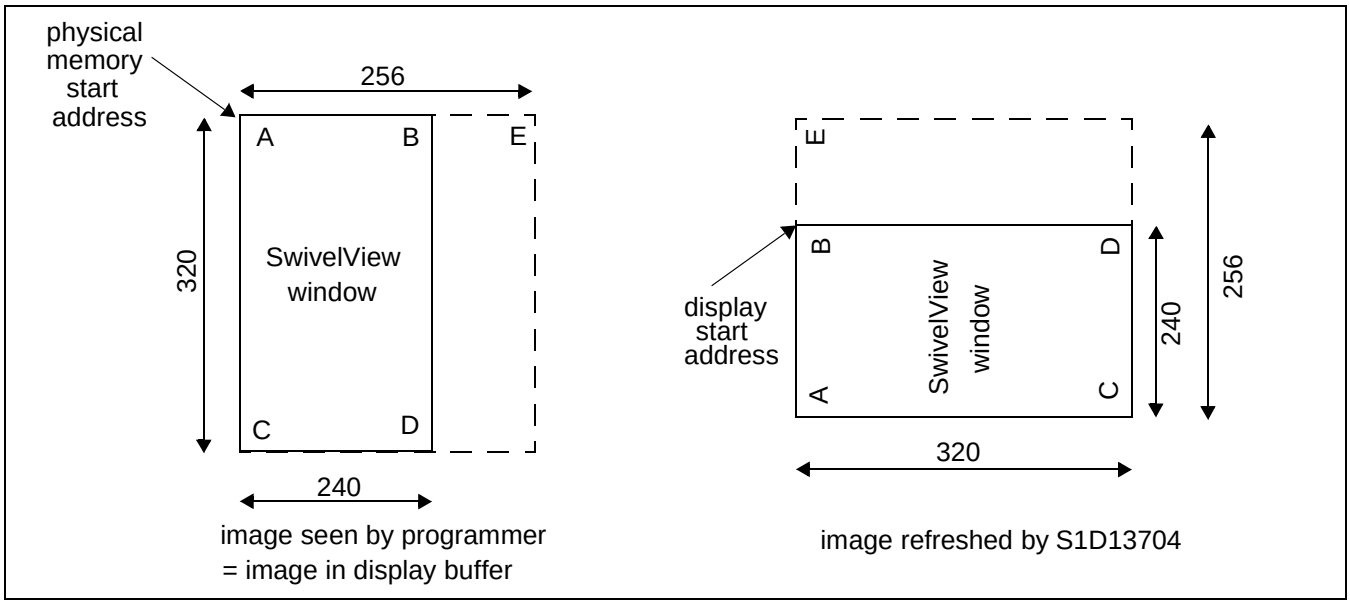


Figure 12-1: Relationship Between The Screen Image and the Image Refreshed by S1D13704

12.1.1 How to Set Up Default SwivelView Mode

The following describes the register settings needed to set up Default SwivelView Mode for a 240x320x4 bpp image:

- Select Default SwivelView Mode: REG[1Bh] bit 7 = 1 and bit 6 = 0
- The display refresh circuitry starts at pixel “B”, therefore the Screen 1 Start Address register must be programmed with the address of pixel “B”, i.e.

$$\begin{aligned} REG[0Dh], REG[0Ch] &= AddressOfPixelB \\ &= (AddressOfPixelA + ByteOffset) \\ &= AddressOfPixelA + \left(\frac{240pixels \times 4bpp}{8bpb}\right) - 1 \\ &= AddressOfPixelA + 77h \end{aligned}$$

Where bpp is bits-per-pixel and bpb is bits-per-byte.

- The Line Byte Count Register for SwivelView Mode must be set to the virtual-image width in bytes, i.e.

$$REG[1Ch] = \frac{256}{(8bpb) \div (4bpp)} = \frac{256}{2} = 128 = 80h$$

Where bpb is bits-per-byte and bpp is bits-per-pixel.

- Panning is achieved by changing the Screen 1 Start Address register:
  - Increment the register by 1 to pan horizontally by one byte, e.g. two pixels in 4 bpp mode
  - Increment the register by twice the value in the Line Byte Count register to pan vertically by two lines, e.g. add 100h to pan by two lines in the example above.

Note
Vertical panning by a single line is not supported in Default SwivelView Mode.

12.2 Alternate SwivelView Mode

Alternate SwivelView Mode may be used when the virtual image size of Default SwivelView Mode cannot be contained in the 40kByte integrated frame buffer. For example, the panel size is 240x160 and the display mode is 8 bit-per-pixel. The minimum virtual image size for Default SwivelView Mode would be 240x256 which requires 60K bytes. Alternate SwivelView Mode requires a panel size of only 240x160 which needs only 38,400 bytes.

Alternate SwivelView Mode requires the memory clock (MCLK) to be at least twice the frequency of the pixel clock (PCLK), i.e. MCLK ≥ 2 x PCLK. Because of this, the power consumption in Alternate SwivelView Mode is higher than in Default SwivelView Mode.

The following figure shows how the programmer sees a 240x160 image and how the image is being displayed. The application image is written to the S1D13704 in the following sense: A–B–C–D. The display is refreshed by the S1D13704 in the following sense: B-D-A-C.

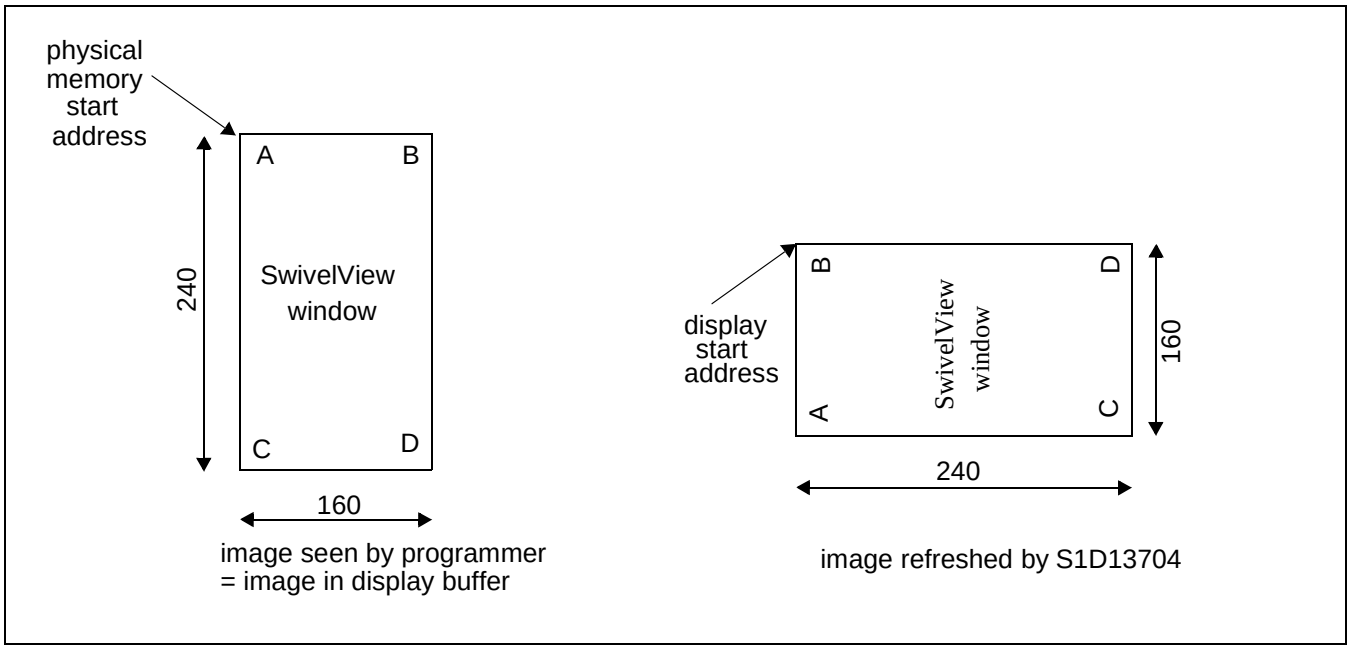


Figure 12-2: Relationship Between The Screen Image and the Image Refreshed by S1D13704

12.2.1 How to Set Up Alternate SwivelView Mode

The following describes the register settings needed to set up Alternate SwivelView Mode for a 160x240x8 bpp image.

- Select Alternate SwivelView Mode:
  REG[1Bh] bit 7 = 1 and bit 6 = 1
- The display refresh circuitry starts at pixel “B”, therefore the Screen 1 Start Address register must be programmed with the address of pixel “B”, or

$$\begin{aligned} \text{REG[0Dh], REG[0Ch]} &= \text{AddressOfPixelB} \\ &= (\text{AddressOfPixelA} + \text{ByteOffset}) \\ &= \text{AddressOfPixelA} + \left(\frac{160\text{pixels} \times 8\text{bpp}}{8\text{bpb}}\right) - 1 \\ &= \text{AddressOfPixelA} + 9\text{Fh} \end{aligned}$$

  Where bpp is bits-per-pixel and bpb is bits-per-byte.

- The Line Byte Count Register for SwivelView Mode must be set to the image width in bytes, i.e.

$$\text{REG[1Ch]} = \frac{160}{(8\text{bpb}) \div (8\text{bpp})} = \frac{160}{1} = 160 = \text{A0h}$$

  Where bpb is bits-per-byte and bpp is bits-per-pixel.

- Panning is achieved by changing the Screen 1 Start Address register:
  - Increment the register by 1 to pan horizontally by one byte, e.g. one pixel in 8 bpp mode
  - Increment the register by the value in the Line Byte Count register to pan vertically by one line, e.g. add A0h to pan by one line in the example above

12.3 Comparison Between Default and Alternate SwivelView Modes

Table 12-1: Default and Alternate SwivelView Mode Comparison

Item	Default SwivelView Mode	Alternate SwivelView Mode
Memory Requirements	The width of the rotated image must be a power of 2. In most cases, a virtual image is required where the right-hand side of the virtual image is unused and memory is wasted. For example, a 160x240x8bpp image would normally require only 38,400 bytes - possible within the 40K byte address space, but the virtual image is 256x240x8bpp which needs 61,440 bytes - not possible.	Does not require a virtual image.
Clock Requirements	CLK need only be as fast as the required PCLK.	MCLK, and hence CLK, need to be 2x PCLK. For example, if the panel requires a 3MHz PCLK, then CLK must be 6MHz. Note that 25MHz is the maximum CLK, so PCLK cannot be higher than 12.5MHz in this mode.
Power Consumption	Lowest power consumption.	Higher than Default Mode.
Panning	Vertical panning in 2 line increments.	Vertical panning in 1 line increments.
Performance	Nominal performance.	Higher performance than Default Mode.

12.4 SwivelView Mode Limitations

The only limitation to using SwivelView mode on the S1D13705. is that split screen operation is not supported.

13 Power Save Modes

Two Power Save Modes have been incorporated into the S1D13704 to accommodate the need for power reduction in the hand-held devices market. These modes are enabled as follows:

Table 13-1: Power Save Mode Selection

Hardware Power Save	Software Power Save Bit 1	Software Power Save Bit 0	Mode
Not Configured or 0	0	0	Software Power Save Mode
Not Configured or 0	0	1	reserved
Not Configured or 0	1	0	reserved
Not Configured or 0	1	1	Normal Operation
Configured and 1	X	X	Hardware Power Save Mode

13.1 Software Power Save Mode

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

Table 13-2: Software Power Save Mode Summary

• Registers read/write accessible
• Memory read/write accessible
• LCD outputs are forced low

13.2 Hardware Power Save Mode

Hardware Power Save Mode saves power by powering down the panel, stopping accesses to the display buffer and registers, and disabling the Host Bus Interface.

Table 13-3: Hardware Power Save Mode Summary

• Host Interface not accessible
• Memory read/write not accessible
• LCD outputs are forced low

13.3 Power Save Mode Function Summary

Table 13-4: Power Save Mode Function Summary

	Hardware Power Save	Software Power Save	Normal
IO Access Possible?	No	Yes	Yes
Memory Access Possible?	No	Yes	Yes
Sequence Controller Running?	No	No	Yes
Display Active?	No	No	Yes
LCDPWR	Inactive	Inactive	Active
FPDAT[11:0], FPSHIFT (see note)	Forced Low	Forced Low	Active
FPLINE, FPFRAME, DRDY	Forced Low	Forced Low	Active

Note

When FPDAT[11:8] are designated as GPIO outputs, the output state prior to enabling the Power Save Mode is maintained. When FPDAT[11:8] are designated as GPIO inputs, unused inputs must be tied to either IO V_{DD} or GND - see Table 5-3: “LCD Interface Pin Mapping,” on page 23.

13.4 Panel Power Up/Down Sequence

After chip reset or when entering/exiting a power save mode, the Panel Interface signals follow a power on/off sequence shown below. This sequence is essential to prevent damage to the LCD panel.

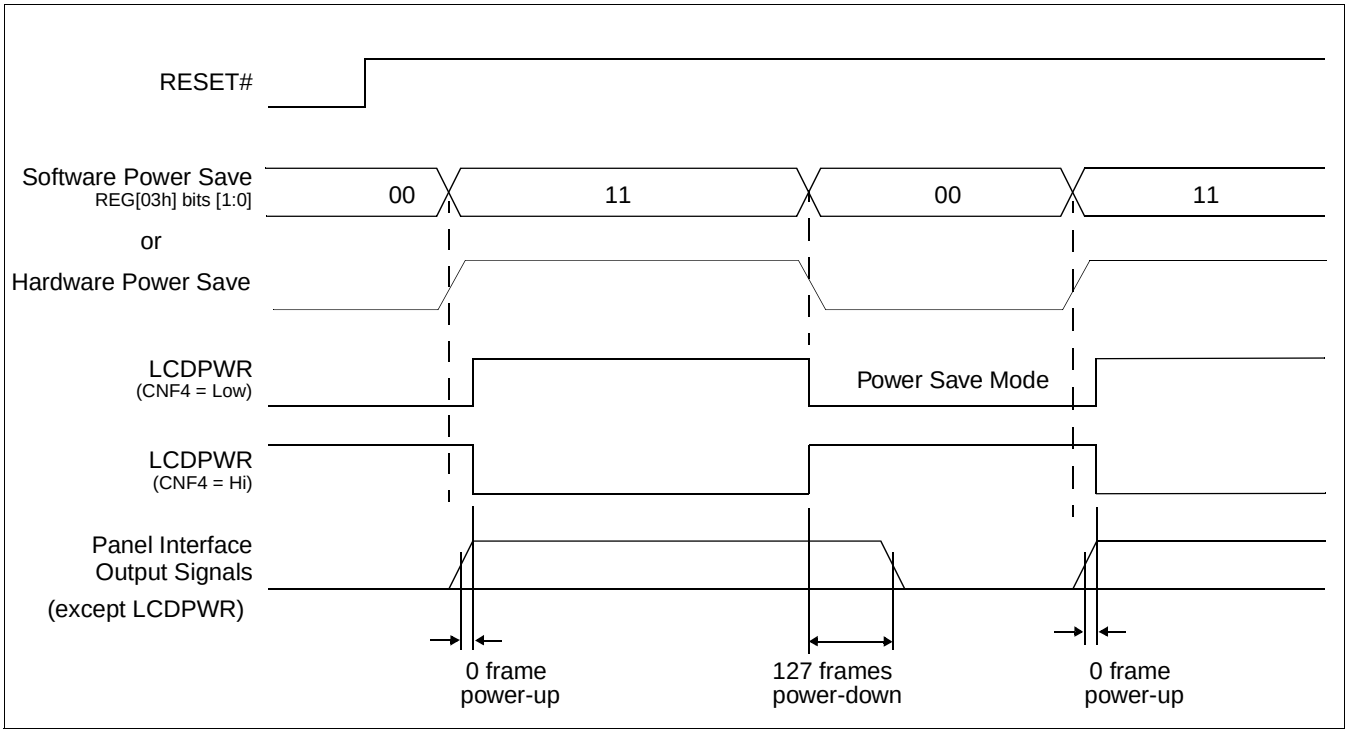


Figure 13-1: Panel On/Off Sequence

After chip reset, LCDPWR is inactive and the rest of the panel interface output signals are held 'low'. Software initializes the chip (i.e. programs the registers) and then - as a last step set - programs REG[03h] bits [1:0] to 11. This starts the power-up sequence as shown. The power-up/power-down sequence delay is 127 frames.

The power-up/power-down sequence also occurs when exiting/entering Software Power Save Mode.

13.5 Turning Off BCLK Between Accesses

BCLK may be turned off (held low) between accesses if the following rules are observed:

1. BCLK must be turned off/on in a glitch free manner
2. BCLK must continue for a period equal to [$8T_{BCLK} + 12T_{MCLK}$] after the end of the access (RDY# asserted or WAIT# deasserted).
3. BCLK must be present for at least one T_{BCLK} before the start of an access.

13.6 Clock Requirements

The following table shows what clock is required for which function in the S1D13704.

Table 13-5: S1D13704 Internal Clock Requirements

Function	BCLK	CLKI
Register Read/Write	Is required during register accesses. BCLK can be shut down between accesses: allow eight BCLK pulses plus 12 MCLK pulses ($8T_{BCLK} + 12T_{MCLK}$) after the last access before shutting BCLK off. Allow one BCLK pulse after starting up BCLK before the next access	Not Required
Memory Read/Write	Is required during memory accesses. BCLK can be shut down between accesses: allow eight BCLK pulses plus 12 MCLK pulses ($8T_{BCLK} + 12T_{MCLK}$) after the last access before shutting BCLK off. Allow one BCLK pulse after starting up BCLK before the next access	Required
Software Power Save	Required	Can be stopped after 128 frames from entering Software Power Save, i.e. after REG[03h] bits 1-0 = 11
Hardware Power Save	Not Required	Can be stopped after 128 frames from entering Hardware Power Save

14 Mechanical Data

QFP14 - 80 pin

Unit: mm

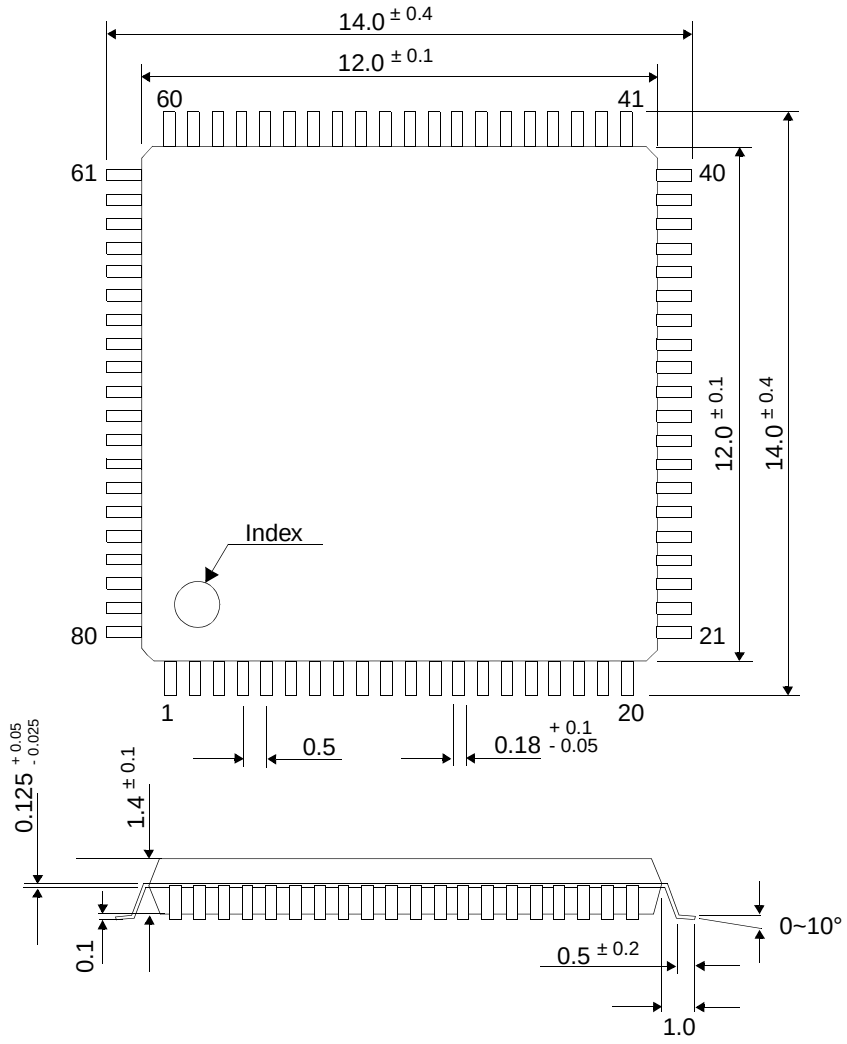


Figure 14-1: Mechanical Drawing QFP14

EPSON®

S1D13704 Embedded Memory Color LCD Controller

Programming Notes and Examples

THIS PAGE LEFT BLANK

Table of Contents

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This guide describes how to program various features of the S1D13704 Embedded Memory Color LCD controller. The demonstrations include descriptions of how to calculate register values and explanations of how or why you might want to do certain procedures.

This guide also introduces the Hardware Abstraction Layer (HAL), which is designed to simplify the programming of the S1D13704. Most S1D1350x, S1D1370x, and 1380x products support the HAL allowing OEMs to switch chips with relative ease.

2 Initialization

This section describes the register settings and steps needed to initialize the S1D13704. The first step toward initializing the S1D13704 is to set the control registers. The S1D13704 then generates the proper control signals for the display. After setting the control registers, the Look-up Table must be programmed with meaningful values. This section does not cover setting Look-Up Table values. See Section 4 on page 14 of this manual for Look-up Table programming details.

The following initialization, presented in table form, provides the sequences and values to set the registers. The notes column comments the reason for the particular value being written.

This example writes to all the control registers. In practice, it may be possible to write to only a subset of the registers. When the S1D13704 is first powered up all registers, unless noted otherwise in the specification, are set to zero. This example programs these registers to zero to establish a known state.

The initialization enables the S1D13704 to control a panel with the following specifications:

- 320x240 color dual passive panel at 75Hz.
- Color Format 2, 8-bit data interface.
- 4 bit-per-pixel (bpp) - 16 colors.
- 25 MHz input clock (CLKI).

Table 2-1: S1D13704 Initialization Sequence

Register	Value (hex)	Notes	See Also
[01]	0010 0000 (20)	Select an passive, Single, Color panel with a data width of 4-bits	
[02]	1010 0000 (B0)	Select 4-bpp color depth and high performance.	
[03]	0000 0011 (03)	Select normal power operation	
[04]	0010 0111 (27)	Horizontal display size = (Reg[04]+1)*8 = (39+1) * 8 = 320 pixels	
[05]	1110 1111 (EF)	Vertical display size = Reg[06][05] + 1 = 0000 0000 1110 1111 + 1 = 239 +1 = 240 lines	
[06]	0000 0000 (00)		
[07]	0000 0000 (00)	FPLINE start position (not used by STN)	
[08]	0001 1110 (1E)	Horizontal non-display period = (Reg[08] + 4) * 8 = (30 + 4) * 8 = 272 pixels	Frame Rate Calculation
[09]	0000 0000 (00)	FPFRAME start position (not used by STN)	
[0A]	0010 0110 (26)	Vertical non-display period = REG[0A] = 38 lines	Frame Rate Calculation
[0B]	0000 0000 (00)	MOD rate - not required for this panel	
[0C]	0000 0000 (00)	Screen 1 Start Address - set to 0 for initialization	Split Screen on page 30
[0D]	0000 0000 (00)		

Table 2-1: S1D13704 Initialization Sequence (Continued)

Register	Value (hex)	Notes	See Also
[0F]	0000 0000 (00)	Screen 2 Start Address - set to 0 for initialization	
[10]	0000 0000 (00)		
[12]	0000 0000 (00)	Memory Address offset - not virtual setup so set to 0	
[13]	1111 1111 (FF)	Set the vertical size to the maximum value.	Split Screen on page 30
[14]	0000 0011 (03)		
[15]	0000 0000 (00)	SetLUT control registers to 0 for this example.	Look-Up Table (LUT) on page 14
[16]	0000 0000 (00)		
[17]	0000 0000 (00)		
[18]	0000 0000 (00)	GPIO control and status registers - set to “0”	
[19]	0000 0000 (00)		
[1A]	0000 0000 (00)	Set the scratch pad bits to “0”.	
[1B]	0000 0000 (00)	We are not setting up SwivelView mode so set this register to “0”.	
[1C]	0000 0000 (00)	Line Byte Count is only required for SwivelView mode.	
[1E],[1F]	0000 0000 (00)	These registers are reserved and should not be written to.	

2.1 Frame Rate Calculation

The system the S1D13704 is being configured for dictates certain physical constraints such as the width and height of the panel and the video system input clock.

The following are the formulae for determining the frame rate of a panel. The frame rate for a single passive or TFT panel is calculated as follows:

$$\text{FrameRate} = \frac{\text{PCLK}}{(\text{HDP} + \text{HNDP}) \times (\text{VDP} + \text{VNDP})}$$

for a dual passive panel the formula is:

$$\text{FrameRate} = \frac{\text{PCLK}}{2 \times (\text{HDP} + \text{HNDP}) \times \left(\frac{\text{VDP}}{2} + \text{VNDP}\right)}$$

where: PCLK = Pixel clock (in Hz)
HDP = Horizontal Display Period (in pixels)
HNDP = Horizontal Non-Display Period (in pixels)
VDP = Vertical Display Period (in lines)
VNDP = Vertical Non-Display Period (in lines)

To achieve the desired frame rate the HNDP and VNDP values can be manipulated. The example below is a generic routine to calculate HNDP and VNDP from a desired frame rate.

This routine first performs a formula rearrangement so that HNDP or VNDP can be solved for. Start with VNDP set to a small value. Loop increasing VNDP and solving the equation for HNDP until satisfactory HNDP and VNDP values are found. If no satisfactory values are found then divide CLKI and repeat the process. If a satisfactory frame rate still can't be reached - return an error.

In C the code looks like the following snip:

```
for (int loop = 0; loop < 2; loop++)
{
      for (VNDP = 2; VNDP < 0x3F; VNDP += 3)
      {
            // Solve for HNDP
             HNDP = (PCLK / (FrameRate * (VDP + VNDP))) - HDP;
            if ((HNDP >= 32) && (HNDP <= 280))
            {
                  // Solve for VNDP.
                  VNDP = (PCLK / (FrameRate * (HDP + HNDP))) - VDP;
                  // If we have satisfied VNDP then we're done.
                  if ((VNDP >= 0) && (VNDP <= 0x3F))
                        goto DoneCalc;
            }
      }
      // Divide ClkI and try again.
      // (Reg[02] allows us to dived CLKI by 2)
      PCLK /= 2;
}
// If we still can't hit the frame rate - throw an error.
if ((VNDP < 0) || (VNDP > 0x3F) || (HNDP < 32) || (HNDP > 280))
{
      sprintf("ERROR: Unable to set the desired frame rate.\n");
      exit(1);
}
```

3 Memory Models

The S1D13704 is capable of operating at four different color depths. The data format for each color depth is packed pixel. S1D13704 packed pixel modes can range from one byte containing eight adjacent pixels (1-bpp) to one byte containing just one pixel (8-bpp).

Packed pixel data memory may be envisioned as a stream of data. Pixels fill this stream with one pixel packed in adjacent to the next. If a pixel requires four bits then it will be located in the four most significant bits of a byte. The pixel to the immediate right on the display will occupy 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 S1D13704 contains 40 kilobytes of internal display memory. External support logic must be employed to determine the starting address for this display memory in CPU address space. On the S5U13704B00C PC platform evaluation boards the address is usually fixed at D0000h.

3.1.1 1 Bit-Per-Pixel (2 Colors/Gray Shades)

1-bit pixels support two color/gray shades. In this memory format each byte of display buffer contains eight adjacent pixels. Setting or resetting any pixel requires reading the entire byte, masking out appropriate bits and, if necessary, setting bits to "1".

With color panels the two colors are derived by indexing into positions 0 and 1 of the Look-Up Table. For monochrome panels the two gray shades are generated by indexing into the first two elements of the green component of the Look-Up Table (LUT).

If the first two LUT elements are set to black (RGB = 0 0 0) and white (RGB = F F F) then each "0" bit of display memory will display as a black pixel and each "1" bit will display as a white pixel. The two LUT entries can be set to any desired colors, for instance red/green or cyan/yellow.

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 (2 Colors/Gray Shades) in One Byte of Display Buffer

3.1.2 2 Bit-Per-Pixel (4 Colors/Gray Shades)

2-bit pixels support four color/gray shades. In this memory format each byte of display buffer contains four adjacent pixels. Setting or resetting any pixel requires reading the entire byte, masking out the appropriate bits and, if necessary, setting bits to "1".

For color panels the four colors are derived by indexing into positions 0 through 3 of the Look-Up Table. For monochrome panels the four gray shades are generated by indexing into the first four elements of the green component of the Look-Up Table.

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

Figure 3-2: Pixel Storage for 2 Bpp (4 Colors/Gray Shades) in One Byte of Display Buffer

3.1.3 4 Bit-Per-Pixel (16 Colors/Gray Shades)

Four bit pixels support 16 color/gray shades. In this memory format 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".

For color panels the 16 colors are derived by indexing into the first 16 positions of the Look-Up Table. For monochrome panels the gray shades are generated by indexing into the first 16 elements of the green component of the Look-Up Table.

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

Figure 3-3: Pixel Storage for 4 Bpp (16 Colors/Gray Shades) in One Byte of Display Buffer

3.1.4 Eight Bit-Per-Pixel (256 Colors)

In eight bit-per-pixel mode one byte of display buffer represents one pixel on the display. At this color depth the read-modify-write cycles, required by the lessor pixel depths, are eliminated.

Each byte of display memory consists of three pointers into the Look-Up Table. The three most significant bits form an index into the first eight red values. The next three bits are an index into the first eight green values. The last two bits form an index into the first four blue Look-Up Table entries.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Red bit 2	Red bit 1	Red bit 0	Green bit 2	Green bit 1	Green bit 0	Blue bit 1	Blue bit 0

Figure 3-4: Pixel Storage for 8 Bpp (256 Colors) in One Byte of Display Buffer

4 Look-Up Table (LUT)

This section is supplemental to the description of the Look-Up Table (LUT) architecture found in the S1D13704 Hardware Functional Specification. Covered here is a review of the LUT registers, recommendations for the color and monochrome LUT values, and additional programming considerations for the LUT.

The S1D13704 Look-Up Table consists of sixteen 4-bit wide entries for each of red, green and blue. The Look-Up Table is controlled by three registers. REG[15h] forms the index into the table. REG[16h] determines which bank is active during display. REG[17h] is the register where the Look-Up Table data is read and written.

The currently configured color depth affects how many indices will be used for image display. In color modes, pixel values are used as indices to an RGB value stored in the Look-Up Table. In monochrome modes only the green component of the LUT is used.

4.1 Look-Up Table Registers

REG[15h] Look-Up Table Address Register							**Read/Write**
n/a	n/a	RGB Index bit 1	RGB Index bit 0	LUT Address Bit 3	LUT Address Bit 2	LUT Address Bit 1	LUT Address Bit 0

RGB Index

The RGB Index bits determine how the S1D13704 will handle automatic LUT Address updates.

When the RGB Index is set to auto-increment (00) then three consecutive accesses of REG[17h] will read/write the red, green, and then the blue elements at the Look-Up Table index specified by the LUT Address. After three accesses of REG[17h] the LUT Address is incremented. The next access of REG[17h] will be the red element from the new Look-Up Table address.

By altering the RGB Index the sequence can be changed such that three accesses of REG[17h] will affect just the reds or just the greens or just the blues at three different LUT addresses.

When configured for monochrome panels the mechanism in which writes are handled is slightly different. One to three reads are still required to update the LUT Address depending on the setting of the RGB Index bits. If the RGB Index bits are set to auto-increment then three writes to REG[17h] are required to bump the LUT Address. Only the last write will affect the display appearance; it is copied across all three RGB elements. If the RGB Index is set to access just red, just green or just blue then a single write to REG[17h] is copied to the red, green and blue elements of the lookup address and the LUT Address is incremented.

Look-Up Table Address

The Look-Up Table (LUT) consists of 16 indexed entries each consisting three 4-bit elements (red, green, blue). The LUT Address bits select which of the 16 entries is accessed. Upon setting the LUT Address an internal pointer is set to the red element. Dependent on the RGB Index setting one to three accesses of the Look-Up Table Data register cause the LUT Address to automatically increment to the next index.

REG[16h] Look-Up Table Bank Select Register							**Read/Write**
n/a	n/a	Red Bank Select bit 1	Red Bank Select bit 0	Green Bank Select bit 1	Green Bank Select bit 0	Blue Bank Select bit 1	Blue Bank Select bit0

Look-Up Table Bank Select

The Look-Up Table Bank Select register affects displayed colors.

Depending on the color mode, not all of the sixteen Look-Up Table (LUT) entries are required. This register determines which entries will be displayed.

At 1-bpp only the lower eight Look-Up Table addresses are used. These are further divided into four banks of two colors. The bank selects determine which of the four red, green and blue banks the displayed colors will come from. For instance: Assume the Look-Up Table Bank Select register was set to 18h (0001 1000 b). Red pixels would come from the 2nd red lookup bank (red LUT Addresses 2 and 3). Green would be taken from the 3rd green lookup bank (green LUT addresses 4 and 5). Blue pixels would be taken from the 1st blue lookup bank (blue LUT addresses 0 and 1).

At 2-bpp, sixteen Look-Up Table addresses are used. The Look-Up Table is a now arranged into four banks of four colors each. As with 1-bpp, the bank select bits determine the initial offset into the Look-Up Table. Incrementing a bank select by one bumps the Look-Up Table offset by four.

Table 4-1: 2 Bpp Banking Scheme

Bank	Red LUT Addresses	Green LUT Addresses	Blue LUT Addresses
0	0	0	0
	1	1	1
	2	2	2
	3	3	3
1	4	4	4
	5	5	5
	6	6	6
	7	7	7
2	8	8	8
	9	9	9
	A	A	A
	B	B	B
3	C	C	C
	D	D	D
	E	E	E
	F	F	F

At 4-bpp the pixel data is a direct index to the color to be displayed. At this color depth the Look-Up Table Bank Select bits have no effect on the display colors. For instance: If the data was 7Bh then the first pixel color would be from the RGB values of the 8th Look-Up Table address. The second pixel would be the colored by the RGB value at the 12th (0Bh) Look-Up Table address.

Table 4-2: 4 Bpp Banking Scheme

Red LUT Addresses	Green LUT Addresses	Blue LUT Addresses
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
A	A	A
B	B	B
C	C	C
D	D	D
E	E	E
F	F	F

At 8-bpp the lookup scheme gets a little more complicated. Each byte of display data contains 3 bits of red lookup, 3 bits of green lookup and 2 bits of blue lookup. The 16 addresses of the Look-Up Table are divided into 2 eight-element banks for the red and green components and 4 four-element banks for the blue component.

Table 4-3: 8 Bpp Banking Scheme

Red/Green Bank	Red LUT Addresses	Green LUT Addresses	Blue Bank	Blue LUT Addresses
0	0	0	0	0
	1	1		1
	2	2		2
	3	3		3
	4	4	1	4
	5	5		5
	6	6		6
	7	7		7
1	8	8	2	8
	9	9		9
	A	A		A
	B	B		B
	C	C	3	C
	D	D		D
	E	E		E
	F	F		F

REG[17h] Look-Up Table Data Register							Read/Write
n/a	n/a	n/a	n/a	LUT Data Bit 3	LUT Data Bit 2	LUT Data Bit 1	LUT Data Bit 0

LUT Data

This register is where the 4-bit red/green/blue data value is written/read. With each successive read or write the internal RGB selector is incremented. Depending on the RGB Index setting, one to three access of this register will result in the Look-Up Table Address incrementing.

4.2 Look-Up Table (LUT) Organization

Color and monochrome operation is slightly different. Both Look-Up Table schemes are described here.

- 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; a pixel value of 7 would point to the eighth LUT entry.
- The value inside each LUT entry represents the intensity of the given color or gray shade. This intensity can range in value between 00 and 0Fh.

The following table shows how many elements from each Look-Up Table index are used at the different color depths.

Table 4-4: Look-Up Table Configurations

Display Mode	4-Bit Wide Look-Up Table		
	Red	Green	Blue
1 Bpp Gray		4 banks of 2	
2 Bpp Gray		4 banks of 4	
4 Bpp Gray		1 bank of 16	
1 Bpp Color	4 banks of 2	4 banks of 2	4 banks of 2
2 Bpp Color	4 banks of 4	4 banks of 4	4 banks of 4
4 Bpp Color	1 bank of 16	1 bank of 16	1 bank of 16
8 Bpp Color	2 banks of 8	2 banks of 8	4 banks of 4

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

Color Modes

1 Bpp Color

When the S1D13704 is configured for 1 bit-per-pixel color mode, only the first two colors from the active bank are displayed. The two entries can be set to any color but are typically set to black and white.

Each byte in the display buffer contains 8 bits, each bit represents an individual pixel. A bit value of "0" results in the Look-Up Table 0 value being displayed. A bit set to "1" results in the Look-Up Table index 1 value displayed.

The following table shows the recommended values for 1 bpp on a color panel.

Table 4-5: Recommended LUT Values for 1 Bpp Color Mode

Index	Red	Green	Blue
00	00	00	00
01	0F	0F	0F
02	00	00	00
...	00	00	00
0F	00	00	00

	Normally unused entries

2 Bpp Color

When the S1D13704 is configured for 2 bit-per-pixel color mode, only the first four colors from the active bank are displayed. The four entries can be set to any color.

Each byte in the display buffer contains 4 adjacent pixels. Each pair of bits in the byte are used as an index into the LUT. The following table shows example values for 2 bpp color mode.

Table 4-6: LUT Values for 2 Bpp Color Mode

Index	Red	Green	Blue
00	00	00	0F
01	0F	00	00
02	00	0F	00
03	0F	0F	0F
04	00	00	00
...	00	00	00
0F	00	00	00

	Normally unused entries

4 Bpp Color

When the S1D13704 is configured for 4 bit-per-pixel operation all sixteen Look-Up Table entries 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	00	00	0A
02	00	0A	00
03	00	0A	0A
04	0A	00	00
05	0A	00	0A
06	0A	0A	00
07	0A	0A	0A
08	00	00	00
09	00	00	0F
0A	00	0F	00
0B	00	0F	0F
0C	0F	00	00
0D	0F	00	0F
0E	0F	0F	00
0F	0F	0F	0F

8 Bpp Color

When the S1D13704 is configured for 8 bit-per-pixel color mode, 8 colors from red and green and 4 colors from the blue active banks are displayed. The eight red, eight green and four blue entries can be set to any color.

The S1D13704 LUT has four bits (16 levels) of intensity control per primary color while a standard VGA RAMDAC has six bits (64 levels). This four to one difference has to be considered when attempting to match colors between a VGA RAMDAC and the S1D13704 LUT. (i.e. VGA levels 0 - 3 map to LUT level 0, VGA levels 4 - 7 map to LUT level 1... etc.).

The following table shows LUT values that approximate the default 256 color VGA palette.

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

Index	Red	Green	Blue
00	00	00	00
01	02	02	05
02	04	04	0A
03	06	06	0F
04	09	09	00
05	0B	0B	00
06	0D	0D	00
07	0F	0F	00
08	00	00	00
...	00	00	00
0F	00	00	00

	Normally unused entries

Gray Shade Modes

1 Bpp Gray Shade (Black-and-White)

In 1 bpp gray shade mode only the first two entries of the green LUT are used. All other LUT entries are unused.

Table 4-9: Recommended LUT Values for 1 Bpp Gray Shade

Address	Red	Green	Blue
00	00	00	00
01	0F	0F	0F
02	00	00	00
...	00	00	00
0F	00	00	00

	Normally unused entries

2 Bpp Gray Shade

In 2 bpp gray shade mode the first four green elements are used to provide values to the panel. The remaining indices are unused.

Table 4-10: Suggested Values for 2 Bpp Gray Shade

Index	Red	Green	Blue
0	00	00	00
1	05	05	05
2	0A	0A	0A
3	0F	0F	0F
4	00	00	00
...	00	00	00
F	00	00	00

	Normally unused entries

4 Bpp Gray Shade

The 4 bpp gray shade mode uses all 16 LUT elements.

Table 4-11: Suggested LUT Values for 4 Bpp Gray Shade

Index	Red	Green	Blue
00	00	00	00
01	01	01	01
02	02	02	02
03	03	03	03
04	04	04	04
05	05	05	05
06	06	06	06
07	07	07	07
08	08	08	08
09	09	09	09
0A	0A	0A	0A
0B	0B	0B	0B
0C	0C	0C	0C
0D	0D	0D	0D
0E	0E	0E	0E
0F	0F	0F	0F

	Normally unused entries

5 Advanced Techniques

This section contains information on the following:

- virtual display
- panning and scrolling
- split screen display

5.1 Virtual Display

Virtual display refers to the situation where the image to be viewed is larger than the physical display. The difference can be in the horizontal, vertical or both dimensions. To view the image, the display is used as a window into the display buffer. At any given time only a portion of the image is visible. Panning and scrolling are used to view the full image.

The Memory Address Offset register determines the number of horizontal pixels in the virtual image. The offset register can be used to specify from 0 to 255 additional words for each scan line. At 1 bpp, 255 words span an additional 4,080 pixels. At 8 bpp, 255 words span an additional 510 pixels.

The maximum vertical size of the virtual image is the result of dividing 40960 bytes of display memory by the number of bytes on each line (i.e. at 1 bpp with a 320x240 panel set for a virtual width of 640x480 there is enough memory for 512 lines).

Figure 5-1: “Viewport Inside a Virtual Display,” depicts a typical use of a virtual display. The display panel is 320x240 pixels, an image of 640x480 pixels can be viewed by navigating a 320x240 pixel viewport around the image using panning and scrolling.

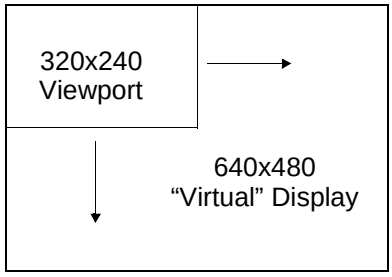


Figure 5-1: Viewport Inside a Virtual Display

5.1.1 Registers

REG[12h] Memory Address Offset Register							
Memory Address Offset Bit 7	Memory Address Offset Bit 6	Memory Address Offset Bit 5	Memory Address Offset Bit 4	Memory Address Offset Bit 3	Memory Address Offset Bit 2	Memory Address Offset Bit 1	Memory Address Offset Bit 0

Figure 5-2: Memory Address Offset Register

REG[12h] forms an 8-bit value called the Memory Address Offset. This offset is the number of additional bytes on each line of the display. If the offset is set to zero there is no virtual width.

Note
This value does not represent the number of words to be shown on the display. The display width is set in the Horizontal Display Width register.

5.1.2 Examples

Example 1: In this example we go through the calculations to display a 640x480 image on a 320x240 panel at 2 bpp.

Step 1: Calculate the number of pixels per word for this color depth.

At 2 bpp each byte is comprised of 4 pixels, therefore each word contains 8 pixels.

pixels_per_word = 16 / bpp = 16 / 2 = 8

Step 2: Calculate the Memory Address Offset register value

We require a total of 640 pixels. The horizontal display register will account for 320 pixels, this leaves 320 pixels for the Memory Address Offset register to account for.

offset = pixels / pixels_per_word = 320 / 8 = 40 = 28h

The Memory Address Offset register, REG[12h], will have to be set to 28h to satisfy the above condition.

Example 2: From the above, what is the maximum number of lines our image can contain?

Step 1: Calculate the number of bytes on each line.

bytes_per_line = pixels_per_line / pixels_per_byte = 640 / 4 = 160

Each line of the display requires 160 bytes.

Step 2: Calculate the number of lines the S1D13704 is capable of.

total_lines = memory / bytes_per_line = 40960 / 160 = 256

The the maximum number of lines which can be accommodated by our image can contain is 256. This example will not “fit” in available display memory. We must reduce either the color depth or the virtual image size.

5.2 Panning and Scrolling

Panning and scrolling describe the actions of appearing to move the image in a virtual display so that all the image can be viewed. After correctly setting up a virtual display (see above) and loading an image into display memory, panning and scrolling allow viewing the entire image a portion at a time.

Panning describes the horizontal (side to side) motion of the viewport. When panning to the right the image in the viewport appears to slide to the left. When panning to the left the image to appears to slide to the right. Scrolling describes the vertical (up and down) motion of the viewport. Scrolling down causes the image to appear to slide up and scrolling up causes the image to appear to slide down.

Both panning and scrolling are performed by modifying the start address register. Start address refers to the word offset in the display buffer where the image will start being displayed from. The start address registers in the S1D13704 are an offset to the first word to be displayed in the top left corner of every frame.

Keep in mind that the start address is a word offset. Changing the start address by one means a change of one words worth of pixels. The number of pixels in word varies according to the color depth. At 1 bit-per-pixel a word contains sixteen pixels. At 2 bit-per-pixel there are eight pixels, at 4 bit-per-pixel there are four pixels and at 8 bit-per-pixel there are two pixels. The number of pixels in each word represent the finest panning step the S1D13704 is capable of. (i.e. at 4 bit-per-pixel the display will move sideways by four pixels for each change to the start address registers)

When SwivelView mode (see SwivelView™ on page 36) is enabled the start address registers become offsets to bytes. In this mode the step rate for the start address registers if halved making for smoother panning.

5.2.1 Registers

REG[0Ch] Screen 1 Display Start Address 0 (LSB)							
Start Addr Bit 7	Start Addr Bit 6	Start Addr Bit 5	Start Addr Bit 4	Start Addr Bit 3	Start Addr Bit 2	Start Addr Bit 1	Start Addr Bit 0

REG[0Dh] Screen 1 Display Start Address 1 (MSB)							
reserved	Start Addr Bit 14	Start Addr Bit 13	Start Addr Bit 12	Start Addr Bit 11	Start Addr Bit 10	Start Addr Bit 9	Start Addr Bit 8

Figure 5-3: Screen 1 Start Address Registers

In landscape mode these two registers form the offset to the word in display memory to be displayed in the upper left corner of the screen. Screen 1 is always the top of a display frame, starting in the upper left corner and descending downward. Changing these registers by one will shift the display 2 to 16 pixels, depending on the current color depth.

In SwivelView mode these registers form the offset to the byte in display memory from where screen 1 will start displaying. Changing these registers in SwivelView mode will result in a shift of 1 to 8 pixels depending on the color depth.

Refer to Table 5-1: “Number of Pixels Panned Using Start Address” to see the minimum number of pixels affected by a change of one to these registers

Table 5-1: Number of Pixels Panned Using Start Address

Color Depth (bpp)	Pixels per Word	Landscape Mode Number of Pixels Panned	SwivelView Mode Number of Pixels Panned
1	16	16	8
2	8	8	4
4	4	4	2
8	2	2	1

5.2.2 Examples

For the following examples assume the display system has been set up to view a 320x240 4 bpp image in a 256x64 viewport. Refer to Section 2, “Initialization” on page 8 and Section 5.1, “Virtual Display” on page 25 for assistance with these settings. The examples are shown in a C-like syntax.

Example 3: Panning (Right and Left)

To pan to the right increase the start address value by one. To pan to the left decrease the start address value. Keep in mind that, with the exception of 8 bit-per-pixel SwivelView mode, the display will jump by more than one pixel as a result of changing the start address registers.

Panning to the right.

```
StartWord = GetStartAddress();
StartWord ++;
SetStartAddress(StartWord);
```

Panning to the left.

```
StartWord = GetStartAddress();
StartWord --;
if (StartWord < 0)
      StartWord = 0;
SetStartAddress(StartWord);
```

Example 4: Scrolling (Up and Down)

To scroll down, increase the value in the Screen 1 Display Start Address Register by the number of words in one *virtual* scan line. To scroll up, decrease the value in the Screen 1 Display Start Address Register by the number of words in one *virtual* scan line.

Step 1: Determine the number of words in one virtual scanline.

bytes_per_line = pixels_per_line / pixels_per_byte = 320 / 2 = 160

words_per_line = bytes_per_line / 2 = 160 /2 = 80

Step 2: Scroll up or down

To scroll up.

```
StartWord = GetStartAddress();
StartWord -= words_per_line;
if (StartWord < 0)
      StartWord = 0;
SetStartAddress(StartWord);
```

To scroll down.

```
StartWord = GetStartAddress();
StartWord += words_per_line;
SetStartAddress(StartWord);

long GetStartAddress (void)
{
      return (REG[0D] * 256 +REG[0C];
}

void SetStartAddress (long StartWord)
{
      REG[0C] = StartWord & 0xFF;
      REG[0D] = StartWord / 256;
}
```

5.3 Split Screen

Occasionally the need arises to display two different but related images. For example, a game where the main play area requires rapid updates and game status displayed at the bottom of the screen. The status area updates far less often than the main play area.

The Split Screen feature of the S1D13704 allows a programmer to setup a display for such an application. The figure below illustrates setting a 320x240 panel to have Image 1 displaying from scan line 0 to scan line 199 and image 2 displaying from scan line 200 to scan line 239. Although this example picks specific values, the split between image 1 and image 2 can occur anywhere on the display.

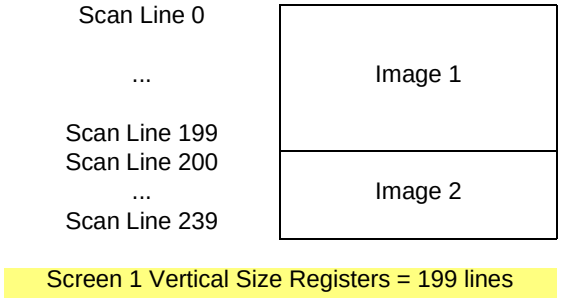


Figure 5-4: 320x240 Single Panel For Split Screen

In split screen operation “Image 1" is taken from the display memory location pointed to by the Screen 1 Start Address registers and always is located at the top of the screen. “Image 2" is taken from the display memory location pointed to by the Screen 2 Start Address registers and begins after Screen 1 Vertical Size lines.

5.3.1 Registers

The other registers required for split screen operations, REG[0Ch] through REG[0Dh] (Screen 1 Start Word Address) and REG[0Fh] through REG[10h] (Screen 2 Start Word Address) are described in Section 5.2.1 on page 28.

REG[13] Screen 1 Vertical Size (LSB)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[14] Screen 1 Vertical Size (MSB)							
n/a	n/a	n/a	n/a	n/a	n/a	Bit 9	Bit 8

Figure 5-5: Screen 1 Vertical Size

These two registers form a ten bit value which determines the size of screen 1. When the vertical size is equal to or greater than the physical number of lines being displayed there is no visible effect on the display. When the vertical size value is less than the number of physically displayed lines, display operation works like this:

1. From the end of vertical non-display (beginning of a frame) to the number of lines indicated by vertical size the display data will come from the memory pointed to by the Screen 1 Display Start Address.

2. After *vertical size* lines have been displayed the system will begin displaying data from Screen 2 Display Start Address memory.

Screen 1 memory is **always** displayed at the top of the screen followed by screen 2 memory. The start address for the screen 2 image may be lower in memory than that of screen 1 (i.e. screen 2 could be coming from offset 0 in the display buffer while screen 1 was coming from an offset located several thousand bytes into the display buffer). While not particularly useful, it is even possible to set screen 1 and screen 2 to the same address.

REG[0Fh] Screen 2 Display Start Address 0 (LSB)							
Start Addr Bit 7	Start Addr Bit 6	Start Addr Bit 5	Start Addr Bit 4	Start Addr Bit 3	Start Addr Bit 2	Start Addr Bit 1	Start Addr Bit 0

REG[10h] Screen 2 Display Start Address 0 (LSB)							
reserved	Start Addr Bit 14	Start Addr Bit 13	Start Addr Bit 12	Start Addr Bit 11	Start Addr Bit 10	Start Addr Bit 9	Start Addr Bit 8

Figure 5-6: Screen 2 Start Address Registers

In landscape mode these two registers form the offset to the word in display memory to be displayed immediately after the screen 1 area of display memory. Changing these registers by one will shift the display 2 to 16 pixels, depending on the current color depth.

Split screen operation is not supported in SwivelView mode, leaving this register un-used. Refer to Table 5-1:, “Number of Pixels Panned Using Start Address,” on page 28 to see the minimum number of pixels affected by a change of one to these registers

5.3.2 Examples

Example 5: Display 200 scanlines of image 1 and 40 scanlines of image 2. Image 2 is located first (offset 0) in the display buffer followed immediately by image 1. Assume a 320x240 display and a color depth of 4 bpp.

1. Calculate the Screen 1Vertical Size register values.

   vertical_size = 200 = C8h

   Write the Vertical Size LSB, REG[13h], with C8h and Vertical Size MSB, REG[14h], with a 00h.

2. Calculate the Screen 1 Start Word Address register values.

   Screen 2 is located first in display memory, therefore we must calculate the number of bytes taken up by the screen 2 data.

   bytes_per_line = pixels_per_line / pixels_per_byte = 320 / 2 = 160

   total bytes = bytes_per_line x lines = 160 x 40 = 6400.

   Screen 2 requires 6400 bytes (0 to 6399) therefore the start address offset for screen 1 must be 6400 bytes. (6400 bytes = 3200 words = C80h words)

   Set the Screen 1 Start Word Address MSB, REG[0Dh], to 0Ch and the Screen 1 Start Word Address LSB, REG[0Ch], to 80h.

3. Calculate the Screen 2 Start Word Address register values.

Screen 2 display data is coming from the very beginning of the display buffer. All there is to do here is ensure that both the LSB and MSB of the Screen 2 Start Word Address registers are set to zero.

6 LCD Power Sequencing and Power Save Modes

6.1 LCD Power Sequencing

LCD Power Sequencing allows the LCD power supply to discharge prior to shutting down the LCD logic signals. Power sequencing is required to prevent long term damage to the panel and to avoid unsightly “lines” on power-down and power-up.

The S1D13704 performs automatic power sequencing when the LCD is enabled or disabled through the Power Save bits in REG[03h] or in response to a hardware power save request. For most applications the internal power sequencing is the appropriate choice.

Proper LCD power sequencing dictates there must be a time delay between the LCD power being disabled and the LCD signals being shut down. During power-up the LCD signals must be active prior to or when power is applied to the LCD. The time intervals vary depending on the power supply design.

One frame after a power save mode has been enabled the S1D13704 disables LCD power. One hundred and twenty seven frames later the LCD logic signals are disabled. There may be situations where the internal time delay is insufficient to discharge the LCD power supply before the LCD signals are shut down. This section details the sequences to manually power-up and power-down the LCD interface.

During the power up sequence the LCD power should not be applied before the LCD logic signals. Usually the power and logic can begin at the same time. There may be times when the LCD logic signals must begin before LCD power is applied.

6.2 Registers

REG[03h] Mode Register 2							
				LCDPWR Override	Hardware Power Save Enable	Software Power Save bit 1	Software Power Save bit 0

The LCD Power (LCDPWR) Override bit forces LCD power to inactive one frame after being toggled. The LCD logic signals to the panel are still active and are controlled by enabling or disabling a power save mode. After enabling a power save mode there are still 128 frames before LCD logic signals are disabled.

The Hardware Power Save Enable bit must be set in order for a hardware power save request (on GPIO0) to have any affect. Without enabling this bit toggling GPIO0 will have no power save effect.

The Software Power Save bits are used to set the software power save mode. The two valid states are "00" for power save and "11" for normal operation.

6.3 LCD Enable/Disable

The descriptions below cover manually powering the LCD panel up and down. Use them only if the power supply connected to the panel requires more than 127 frames to discharge on power-down or if the panel requires starting the LCD logic well in advance of enabling LCD power.

Power On/Enable Sequence

The following is the recommended sequence for manually powering-up an LCD panel. These steps would be used if LCD power had to be applied later than LCD logic.

1. Set REG[03h] bit 3, LCDPWR Override, to "1" (ensures that LCD power is disabled).
2. Enable LCD logic. This is done by either setting GPIO0 to 0 for hardware power save mode and/or by setting REG[03h] bits 1-0, software power save, to "11".
3. Count "x" Vertical Non-Display Periods.
   "x" corresponds the length of time LCD logic must be enabled before LCD power-up, converted to the equivalent vertical non-display periods. For example, at 72 HZ counting 36 non-display periods results in a one half second delay.
4. Set REG[03h] bit 3 to "0" (enable LCD Power).

Power Off/Disable Sequence

The following is the recommended sequence for manually powering-down an LCD panel. These steps would be used if power supply timing requirements were larger than the timings built into the S1D13704 power disable sequence.

1. Set REG[03h] bit 3, LCDPWR Override, to "1" (disables LCD Power).
2. Count "x" Vertical Non-Display Periods.
   "x" corresponds to the power supply discharge time converted to the equivalent vertical non-display periods.
3. Disable the LCD logic by setting the software power save in REG[03h] or setting hardware power save via GPIO0.

7 SwivelView™

7.1 Introduction To SwivelView

Many of todays applications use the LCD panel in a portrait orientation. In this case it becomes necessary to “rotate” the displayed image. This rotation can be done by software at the expense of performance or, as with the S1D13704, it can be done by hardware with no performance penalty.

There are two hardware rotated modes: Default SwivelView Mode and Alternate SwivelView Mode.

7.2 Default SwivelView Mode

Default SwivelView Mode requires the portrait image width be a power of two, e.g. a 240-line panel requires a minimum virtual image width of 256. This mode should be used whenever the required virtual image can be contained within the integrated display buffer (i.e. virtual image size ≤ 40k bytes), as it consumes less power than the Alternate SwivelView mode.

For example, the panel size is 320x240 and the display mode is 4 bit-per-pixel. The virtual image size is 320x256 which can be contained within the 40k Byte display buffer.

Default SwivelView Mode also requires memory clock (MCLK) ≥ pixel clock (PCLK).

The following figures show how the programmer sees a 240x320 image and how the image is displayed. The application image is written to the S1D13704 in the following sense: A–B–C–D. The display is refreshed by the S1D13704 in the following sense: B-D-A-C.

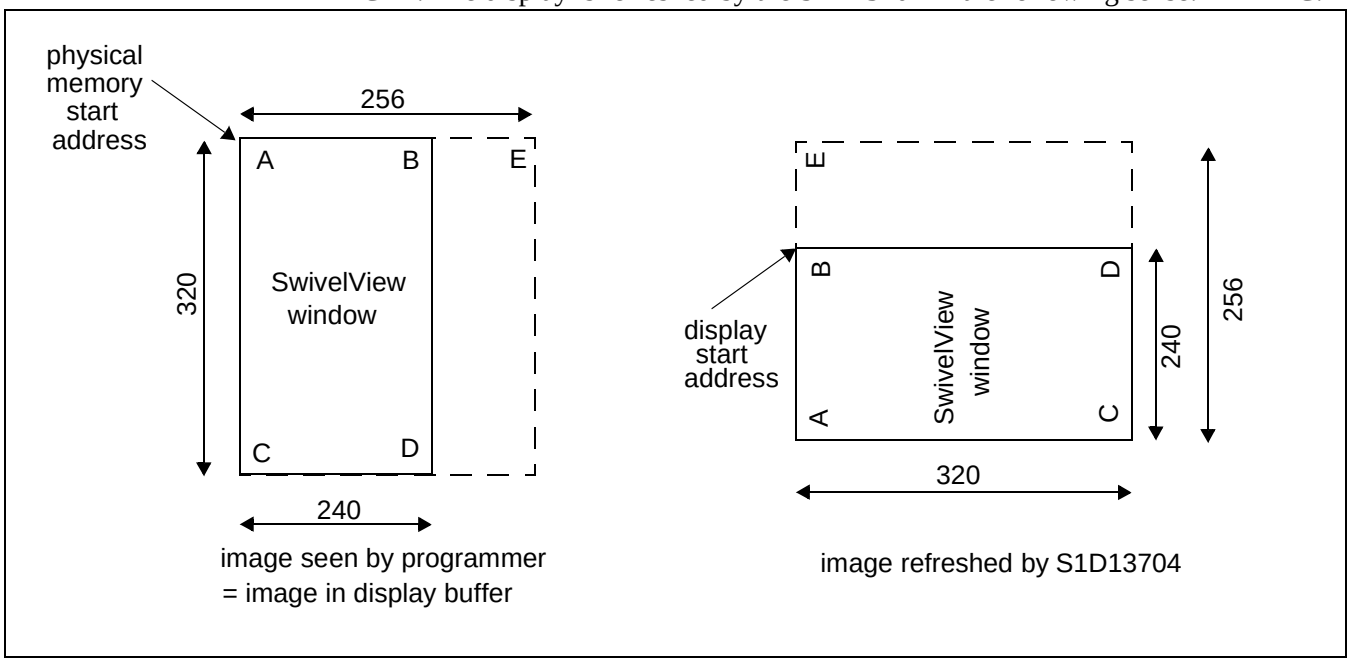


Figure 7-1: Relationship Between The Screen Image and the Image Refreshed by S1D13704

7.3 Alternate SwivelView Mode

Alternate SwivelView Mode may be used when the virtual image size of Default SwivelView Mode cannot be contained in the 40k Byte integrated frame buffer. For example, when the panel size is 240x160 and the display mode is 8 bit-per-pixel the minimum virtual image size for Default SwivelView Mode would be 240x256 which requires 60K bytes. Alternate SwivelView Mode requires a panel size of only 240x160 which needs only 38,400 bytes.

Alternate SwivelView Mode requires the memory clock (MCLK) to be at least twice the frequency of the pixel clock (PCLK), i.e. MCLK ≥ 2 x PCLK.

Because of this, the power consumption in Alternate SwivelView Mode is higher than in Default SwivelView Mode.

The following figure shows how the programmer sees a 240x160 image and how the image is being displayed. The application image is written to the S1D13704 in the following sense: A–B–C–D. The display is refreshed by the S1D13704 in the following sense: B-D-A-C.

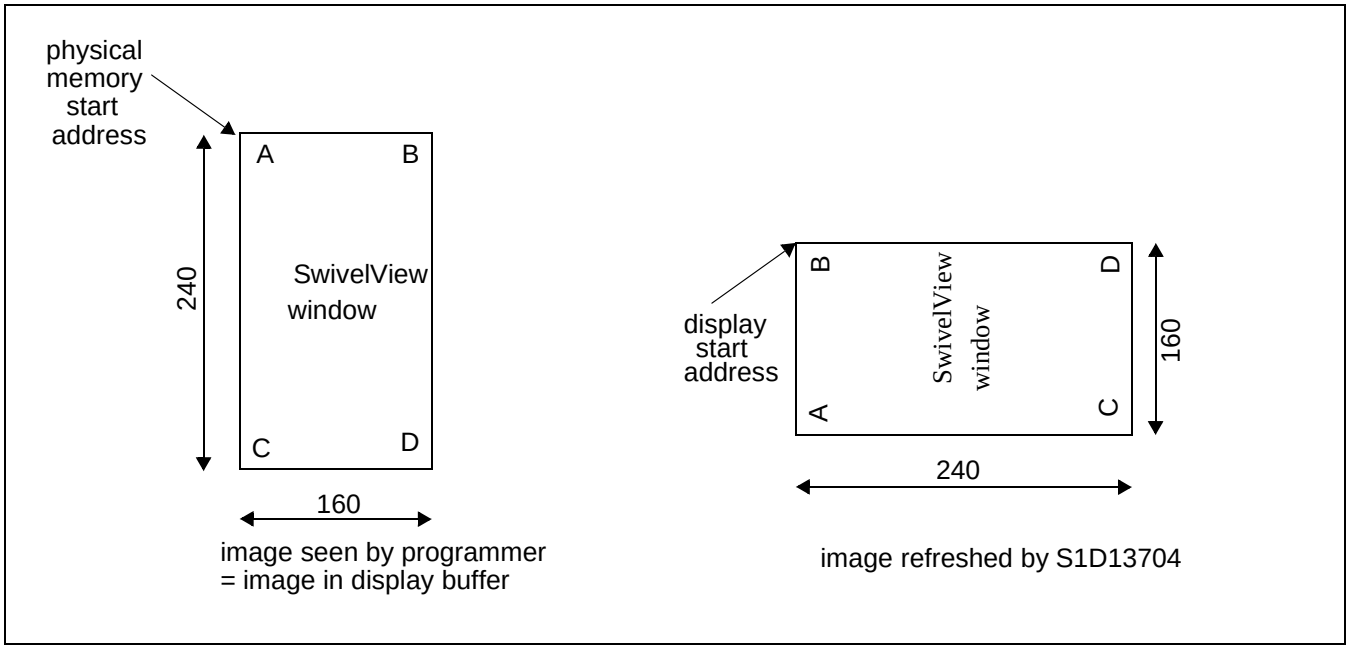


Figure 7-2: Relationship Between The Screen Image and the Image Refreshed by S1D13704

7.4 Registers

This section describes the registers used to set SwivelView mode operation.

REG[0Ch] Screen 1 Start Word Address LSB							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

REG[0Dh] Screen 1 Start Word Address MSB							
reserved	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8

The start address registers must be set for SwivelView mode. In SwivelView mode the offset contained in the start address points to a byte.

REG[1Ch] Line Byte Count Register							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

The line byte count register informs the S1D13704 of the stride, in bytes, between two consecutive lines of display in SwivelView mode. The Line Byte Count register only affects SwivelView mode operation. The contents of this register are ignored when the S1D13704 is in landscape display mode.

REG[1Bh] SwivelView Mode Register							
SwivelView Mode Enable	SwivelView Mode Select	n/a	n/a	n/a	reserved	SwivelView Mode Pixel Clock Select Bit 1	SwivelView Mode Pixel Clock Select Bit 0

The SwivelView mode register contains several items for SwivelView mode support.

The first is the SwivelView Mode Enable bit. When this bit is “0” the S1D13704 is in landscape mode and the remainder of the settings in this register as well as the Line Byte Count in REG[1Ch] are ignored. When this bit is a “1” SwivelView mode is enabled.

There are two SwivelView mode display schemes available. The SwivelView mode select bit selects between the “Default Mode” and the "Alternate Mode”. The default mode offers the lowest power consumption with some display mode limitations. The alternate mode uses more power but offers greater display flexibility.

In return for using less power the default SwivelView imposes the restriction that the SwivelView display width must be a power of two (e.g. 64, 128, 256, 512). The physical display does not need to be a power of two wide. The difference can be treated as a virtual width. In addition, scrolling in default SwivelView mode is restricted to two lines.

Alternate SwivelView mode requires more power as the internal clocks are run faster. In return for a higher power consumption the "power of two" width-restriction is removed. Also, the display can be scrolled one line at a time. One benefit to removing the power of two width restriction is that panels which might not have been able to be used in SwivelView mode due to a lack of memory may now be used.

Clocking for the S1D13704 works as follows:
An external clock source supplies CLKI, the input clock. CLKI is routed through the Input Clock Divide from Mode Register 1 (REG[02h] bit 4) and is either divided by two or passed on. This signal is now the Operating Clock (CLK) from which PCLK and MCLK are derived. In SwivelView mode the CLK signal may be divided down by 0, 2, 4, or 8 before PCLK and MCLK are derived.

SwivelView mode offers additional clocking control over landscape mode. One reason for the additional support is to maintain a register set that was backward compatible with previous Epson LCD controllers.

When setting SwivelView mode, it is possible that the horizontal and vertical non-display time must be recalculated as a result of PCLK changing in response to the SwivelView mode selected or the clock selection method.

7.5 Limitations

The only limitation to using SwivelView mode on the S1D13704 is that split screen operation is not supported.

A comparison of the two SwivelView modes is as follows:

Table 7-1: Default and Alternate SwivelView Mode Comparison

Item	Default SwivelView Mode	Alternate SwivelView Mode
Memory Requirements	The width of the rotated image must be a power of 2. In most cases, a virtual image is required where the right-hand side of the virtual image is unused and memory is wasted. For example, a 160x240x8bpp image would normally require only 38,400 bytes - possible within the 40K byte address space, but the virtual image is 256x240x8bpp which needs 61,440 bytes - not possible.	Does not require a virtual image.
Clock Requirements	CLK need only be as fast as the required PCLK.	MCLK, and hence CLK, need to be 2x PCLK. For example, if the panel requires a 3MHz PCLK, then CLK must be 6MHz. Note that 25MHz is the maximum CLK, so PCLK cannot be higher than 12.5MHz in this mode.
Power Consumption	Lowest power consumption.	Typically 20% higher than Default Mode.
Panning	Vertical panning in 2-line increments.	Vertical panning in 1-line increments.
Performance	Nominal performance.	Slightly higher performance than Default Mode.

7.6 Examples

Example 6: Enable default SwivelView mode for a 320x240 panel at 4 bpp.

Before switching to SwivelView mode from landscape mode, display memory should be cleared to make the user perceived transition smoother. Images in display memory are not rotated automatically by hardware and the garbled image would be visible for a short period of time if video memory is not cleared.

In this example we will bypass having to recalculate the horizontal and vertical non-display times (frame rate) by selecting the default SwivelView mode scheme.

1. Calculate and set the Screen 1 Start Word Address register.

   OffsetBytes = (Width x BitsPerPixel / 8) - 1 = (256 x 4 / 8) -1 = 127 = 007Fh

   (“Width” is the width of the SwivelView mode display - in this case the next power of two greater than 240 pixels or 256.)

   Set Screen1 Display Start Word Address LSB (REG [0Ch]) to 7Fh and Screen1 Display Start Word Address MSB (REG[0Dh]) to 00h.

2. Calculate the Line Byte Count

   The Line Byte Count also must be based on the power of two width.

   LineByteCount = Width x BitsPerPixel / 8 = 256 x 4 / 8 = 128 = 80h.

   Set the Line Byte Count (REG[1C]) to 80h.

3. Enable SwivelView mode.

This example uses the default SwivelView mode scheme. If we do not change the SwivelView Mode Pixel Clock Select bits then we will not have to recalculate the non-display timings to correct the frame rate.

Write 80h to the SwivelView Mode Register (REG[1Bh]).

The display is now configured for SwivelView mode use. Offset zero into display memory will corresponds to the upper left corner of the display. The only thing to keep in mind is that the count from the first pixel of one line to the first pixel of the next line (refered to as the "stride") is 128 bytes.

Example 7: Enable alternate SwivelView mode for a 320x240 panel at 4 bpp.

Note
As we have to perform a frame rate calculation for this mode we need to know the following panel characteristics: 320x240 8-bit color to be run at 80 Hz with a 16 MHz input clock.

As in the previous example, before switching to SwivelView mode, display memory should be cleared. Images in display memory are not rotated automatically by hardware and the garbled image would be visible for a short period of time if video memory is not cleared.

1. Calculate and set the Screen 1 Start Word Address register.

   OffsetBytes = (Width x BitsPerPixel / 8) - 1 = (240 x 4 / 8) - 1 = 119 = 0077h

   Set Screen1 Display Start Word Address LSB (REG [0Ch]) to 77h and Screen1 Display Start Word Address MSB (REG[0Dh]) to 00h.

2. Calculate the Line Byte Count.

   LineByteCount = Width x BitsPerPixel / 8 = 240 x 4 / 8 = 120 = 78h.

   Set the Line Byte Count (REG[1C]) to 78h.

3. Enable SwivelView mode.

   This example uses the alternate SwivelView mode scheme. We will not change the Pixel Clock Select settings.

   Write C0h to the SwivelView Mode register (REG[1Bh])

4. Recalculate the frame rate dependents.

This example assumes the alternate SwivelView mode scheme. In this scheme, without touching the Pixel Clock Select bits the PCLK value will be equal to CLK/2.

Note
These examples don’t use the Pixel Clock Select bits. The ability to divide the PCLK value down further than the default values was added to the S1D13704 to support SwivelView mode on very small panels.

The Pixel Clock value has changed so we must calculate horizontal and vertical non-display times to reach the desired frame rate. Rather than perform the frame rate calculations here I will refer the reader to the frame rate calculations in Frame Rate Calculation on page 9 and simply "arrive" at the following:

Horizontal Non-Display Period = 88h

Vertical Non-Display Period = 03h

Plugging the values into the frame rate calculations yields:

$$\text{FrameRate} = \frac{\text{PCLK}}{(\text{HDP} + \text{HNDP}) \times (\text{VDP} + \text{VNDP})}$$

$$\text{FrameRate} = \frac{\frac{16,000,000}{2}}{(320 + 88) \times (240 + 3)} = 80.69$$

For this example the Horizontal Non-Display register [REG[08h]) needs to be set to 07h and the Vertical Non-Display register (REG[0Ah]) needs to be set to 03h.

The 16,000,000/2 in the formula above represents the input clock being divided by two when this alternate SwivelView mode is selected. With the values given for this example we must ensure the Input Clock Divide bit (REG[02h] b4) is reset (with the given values it was likely set as a result of the frame rate calculations for landscape display mode).

No other registers need to be altered.

The display is now configured for SwivelView mode use. Offset zero of display memory corresponds to the upper left corner of the display. Display memory is accessed exactly as it was for landscape mode.

As this is the alternate SwivelView mode the power of two stride issue encountered with the default SwivelView mode is no longer an issue. The stride is the same as the SwivelView mode width. In this case 120 bytes.

Example 8: Pan the above SwivelView mode image to the right by 4 pixels then scroll it up by 6 pixels.

To pan by four pixels the start address needs to be advanced.

1. Calculate the amount to change start address by.

   Bytes = Pixels x BitsPerPixel / 8 = 4 x 4 / 8 = 2 bytes

2. Increment the start address registers by the just calculated value.

   In this case the value writen to the start address register will be 81h (7Fh + 2 = 81h)

To scroll by 4 lines we have to change the start address by the offset of four lines of display.

1. Calculate the amount to change start address by.

   BytesPerLine = LineByteCount = 128

   Bytes = Lines x BytesPerLine = 4 x 128 = 512 = 200h

2. Increment the start address registers by the just calculated value

   In this case 281h (81h + 200h) will be written to the Screen 1 Start Word Address register pair.

Set Screen1 Display Start Word Address LSB (REG[0Ch]) to 81h and Screen1 Display Start Word Address MSB (REG[0Dh]) to 02h.

8 Identifying the S1D13704

As there are several similar products in the 1350X and 1370X LCD controller families, which can for the most part share the same code base. It may be important for a program to identify between products at run time.

Identification of the S1D13704 can be performed any time after the system has been powered up by reading REG[00h], the Revision Code register. The six most significant bits form the product identification code and the two least significant bits form the product revision.

From reset (power on) the steps to identifying the S1D13704 are as follows:

1. Read REG[00h]. Mask off the lower two bits, the revision code, to obtain the product code.
2. The product code for the S1D13704 is 018h.

9 Hardware Abstraction Layer (HAL)

9.1 Introduction

The HAL is a processor independent programming library provided by Epson with support for several different computing platforms. The HAL was developed to aid implementation of internal test programs and provides an easy, consistent method of programming S1D1350x, S1D1370x, and S1D1380x products on different processor platforms.

The HAL keeps sample code simpler, although end programmers may find the HAL functions to be limited in their scope, and may wish to ignore the HAL.

9.2 API for 13704HAL

The following is a description of the HAL library. Updates and revisions to the HAL may include new functions not included in the following documentation.

The original design philosophy of the HAL was that function return values would be status of the call. Most functions simple return ERR_OK. If a value had to be returned then a pointer of the appropriate type was passed to the function.

9.2.1 Initialization

The following section describes the HAL functions dealing with S1D13704 initialization. Typically a programmer has only to concern themselves with calls to seRegisterDevice() and seSetInit().

int seRegisterDevice(const LPHAL_STRUC lpHalInfo, int * pDevID)

Description: Registers the S1D13704 device parameters with the HAL library. The device parameters have been configured with address range, register values, desired frame rate, etc., and have been saved in the HAL_STRUCT structure pointed to by lpHalInfo.

Parameters: lpHalInfo - pointer to HAL_STRUCT information structure
pDevice - pointer to the integer to receive the device ID

Return Value: ERR_OK - operation completed with no problems
ERR_UNKNOWN_DEVICE - the HAL was unable to find an S1D13704.

Note

No S1D13704 registers are changed by calling seRegisterDevice().

seSetInit(int DevID)

Description: Configures the S1D13704 for operation. This function sets all the S1D13704 control registers to their default values.

Initialization of the S1D13704 was made a stand-alone step to accommodate those programs (e.g. 13704PLAY.EXE) which needed the ability to start and examine the system before changing register contents.

Parameters: DevID - registered device ID

Return Value: ERR_OK - operation completed with no problems

Note

After this call the Look-Up Table will be set to a default state appropriate to the display type.

int seInitHal(void)

Description: This function initializes variables used by the HAL library. Call this function once when the application starts.

Normally, programmers will never need to call seInitHal(). On PC platforms, seRegisterDevice() automatically calls seInitHal(). Consecutive calls to seRegisterDevice() will not call seInitHal() again. On non-PC platforms the start-up code, supplied by Seiko, will call seInitHal(). If support code for a new CPU platform is written the programmer must ensure that seInitHAL() is called prior to calling other HAL functions.

Parameters: None

Return Value: ERR_OK - operation completed with no problems

9.2.2 Miscellaneous HAL Support

Functions in this group do not fit into any specific category of support. They provide a miscellaneous range of support for working with the S1D13704

int seGetId(int DevID, int * pId)

Description: Reads the S1D13704 revision code register to determine the chip product and revisions. The interpreted value is returned in pID.

Parameters: DevID - registered device ID
pId - pointer to an integer which will receive the controller ID.

S1D13704 values returned in pID are:
- ID_S1D13704
- ID_S1D13704F00A
- ID_UNKNOWN

Other HAL libraries will return their respective controller IDs upon detection of their controller.

Return Value: ERR_OK - operation completed with no problems
ERR_UNKNOWN_DEVICE - the HAL was unable to identify the display controller. Returned when pID returns ID_UNKNOWN.

void seGetHalVersion(const char ** pVersion, const char ** pStatus, const char **pStatusRevision)

Description: Retrieves the HAL library version information. The return values are ASCII strings. A typical return would be: “1.01 B 5” - HAL version 1.01, 'B' is the beta designator, this example would be Beta 5. If pStatus is NULL then pStatusRevision should be NULL too.

Parameters: pVersion - Pointer to string to return the version in.
- must point to an allocated string of size VER_SIZE

pStatus - Pointer to a string to return the release status in.
- must point to an allocated string of size STATUS_SIZE

pStatusRevision - Pointer to return the current revision of status.
- must point to an allocated string of size STAT_REV_SIZE

Return Value: None

int seSetBitsPerPixel(int DevID, int BitsPerPixel)

Description: This routine sets the color depth the S1D13704 displays in.

After performing validity checks to ensure the requested video mode can be set the appropriate registers are changed and the Look-Up table is set its default values appropriate to the color depth.

This call is similar to a mode set call on a standard VGA.

Parameter: DevID - registered device ID
BitsPerPixel - desired color depth in bits per pixel.
- Valid arguments are: 1, 2, 4, and 8.

Return Value: ERR_OK - operation completed with no problems
ERR_FAILED- possible causes for this error include:

1) the desired frame rate may not be attainable with the specified input clock
2) the combination of width, height and color depth may require more memory than is available on the S1D13704.

int seGetBitsPerPixel(int DevID, int * pBitsPerPixel)

Description: This function reads the S1D13704 registers to determine the current color depth and returns the result in *pBitsPerPixel*.

Parameters: DevID - registered device ID
pBitsPerPixel - pointer to an integer to receive current color depth.
- return values will be: 1, 2, 4, or 8.

Return Value: ERR_OK - operation completed with no problems

int seGetBytesPerScanline(int DevID, int * pBytes)

Description: Returns the number of bytes use by each scan line in the integer pointed to by pBytes. The number of bytes per scanline will include the number of non-displayed bytes, if applicable.

Prior to calling seGetBytesPerScanline() the S1D13704 control registers must have been correctly initialized.

Parameters: DevID - registered device ID
pBytes - pointer to an integer to receive the number of bytes per scan line

Return Value: ERR_OK - operation completed with no problems

int seGetScreenSize(int DevID, int * Width, int * Height)

Description: Retrieves the width and height in pixels of the display surface. The width and height are derived by reading the horizontal and vertical size registers and calculating the dimensions. Virtual dimensions are not taken into account for this calculation.

When the display is in SwivelView mode the dimensions will be swapped. (i.e. a 640x480 display in SwivelView mode will return a width and height of 480 and height of 640.

Parameters: DevID - registered device ID
Width - pointer to an integer to receive the display width
Height - pointer to an integer to receive the display height

Return value: ERR_OK - the operation completed successfully

int seDelay(int MilliSeconds)

Description: This function will delay for the length of time specified in "MilliSeconds" before returning to the caller.

This function was originally intended for non-PC platforms. Information about how to access the timers was not always available however we do know frame rate and can use that for timing calculations.

The S1D13704 registers must be initialized for this function to work correctly. On the PC platform this is simply a call to the C timing functions and is therefore independent of the register settings.

Parameters: DevID - registered device ID
MilliSeconds- time to delay in seconds

Return Value: ERR_OK - operation completed with no problems
ERR_FAILED- returned on non-PC platforms when the S1D13704 registers have not bee initialized

int seGetLastUsableByte(int DevID, long * plLastByte)

Description: This functions returns a pointer, as a long integer, to the last byte of usable display memory.

This function is a holdover from 1350X products which support different amounts of memory.

Parameters: DevID - registered device ID
plLastByte - pointer to a long integer to receive the offset to the last byte of display memory

Return Value: ERR_OK - operation completed with no problems

int seSetHightPerformance(int DevID, BOOL OnOff)

Description: This function call enables or disable the high performance bit of the S1D13704.

When high performance is enabled then MClk equals PClk for all video display resolutions. In the high performance state CPU to video memory performance is improved at the cost of higher power consumption.

When high performance is disabled then MClk ranges from PClk/1 at 8 bit-per-pixel to PClk/8 at 1 bit-per-pixel. Without high performance CPU to video memory accessed speeds are slower but the S1D13704 uses less power.

Parameters: DevID - registered device ID
OnOff - a boolean value (defined in HAL.H) to indicate whether to enable of disable high performance.

Return Value: ERR_OK - operation completed with no problems

9.2.3 Advanced HAL Functions

Advanced HAL functions include the functions to support split, virtual and rotated displays. While the concept for using these features is advanced the HAL makes actually using them easy.

int seSetHWRotate(int DevID, int Rotate)

Description: This function sets the rotation scheme according to the value of 'Rotate'. When SwivelView mode is selected as the display rotation the scheme selected is the 'non-X2' scheme.

Parameters: DevID - registered device ID
Rotate - the direction to rotate the display
- Valid arguments for Rotate are: LANDSCAPE and PORTRAIT.

Return Value: ERR_OK - operation completed with no problems
ERR_FAILED - the operation failed to complete.
The most likely reason for failing to set a SwivelView mode is an inability to set the desired frame rate when setting the mode. Other factors which can cause a failure include having configured for a 0 Hz frame rate or specifying something other than LANDSCAPE or PORTRAIT for the rotation scheme.

int seSplitInit(int DevID, WORD Scrn1Addr, WORD Scrn2Addr)

Description: This function prepares the system for split screen operation. In order for split screen to function the starting address in display buffer for the upper portion(screen 1) and the lower portion (screen 2) must be specified. Screen 1 is always displayed above screen 2 on the display regardless of the location of their start addresses.

Parameters: DevID - registered device ID
Scrn1Addr - offset, in bytes, to the start of screen 1
Scrn2Addr - offset, in bytes, to the start of screen 2

Return Value: ERR_OK - operation completed with no problems

Note

It is assumed that the system has been properly initialized prior to calling seSplitInit().

int seSplitScreen(int DevID, int Screen, int VisibleScanlines)

Description: Changes the relevant registers to adjust the split screen according to the number of visible lines requested. 'WhichScreen' determines which screen, 1 or 2, to base the changes on.

The smallest surface screen 1 can display is one line. This is due to the way the S1D13704 operates. Setting Screen 1 Vertical Size to zero results in one line of screen 1 being displayed. The remainder of the display will be screen 2 image.

Parameters: DevID - registered device ID
Screen - must be set to 1 or 2 (or use the constants SCREEN1 or SCREEN2)
VisibleScanlines- number of lines to display for the selected screen

Return Value: ERR_OK - operation completed with no problems
ERR_HAL_BAD_ARG- argument VisibleScanlines is negative or is greater than vertical panel size or WhichScreen is not SCREEN1 or SCREEN 2.

Note

seSplitInit() must be called before calling seSplitScreen()

int seVirtInit(int DevID, DWORD VirtX, DWORD * VirtY)

Description: This function prepares the system for virtual screen operation. The programmer passes the desired virtual width, in pixels, as *VirtX*. When the routine returns *VirtY* will contain the maximum number of line that can be displayed at the requested virtual width.

Parameter:
- DevID - registered device ID
- VirtX - horizontal size of virtual display in pixels. (Must be greater or equal to physical size of display)
- VirtY - pointer to an integer to receive the maximum number of displayable lines of 'VirtX' width.

Return Value: ERR_OK - operation completed with no problems
ERR_HAL_BAD_ARG - returned in three situations:
1) the virtual width (VirtX) is greater than the largest possible width (VirtX varies with color depth and ranges from 4096 pixels wider than the panel at 1 bit-per-pixel down to 512 pixels wider than the panel at 8 bit-per-pixel)
2) the virtual width is less than the physical width or
3) the maximum number of lines becomes less than the physical number of lines

Note

The system must have been properly initialized prior to calling seVirtInit()

int seVirtMove(int DevID, int Screen, int x, int y)

Description: This routine pans and scrolls the display after a virtual display has bee setup. In the case where split screen operation is being used the WhichScreen argument specifies which screen to move. The x and y parameters specify, in pixels, the starting location in the virtual image for the top left corner of the applicable display.

Parameter:
- DevID - registered device ID
- Screen - must be set to 1 or 2, or use the constants SCREEN1 or SCREEN2, to identify which screen to base calculations on
- x - new starting X position in pixels
- y - new starting Y position in pixels

Return Value: ERR_OK - operation completed with no problems
ERR_HAL_BAD_ARG- there are several reasons for this return value:
1) WhichScreen is not SCREEN1 or SCREEN2.
2) the y argument is greater than the last available line less the screen height.

Note

seVirtInit() must be been called before calling seVirtMove().

9.2.4 Register / Memory Access

The Register/Memory Access functions provide access to the S1D13704 registers and display buffer through the HAL.

int seGetReg(int DevID, int Index, BYTE * pValue)

Description: Reads the value in the register specified by index.

Parameters:
DevID - registered device ID
Index - register index to read
pValue - pointer to a BYTE to receive the register value.

Return Value: ERR_OK - operation completed with no problems

int seSetReg(int DevID, int Index, BYTE Value)

Description: Writes value specified in Value to the register specified by Index.

Parameters:
DevID - registered device ID
Index - register index to set
Value - value to write to the register

Return Value: ERR_OK - operation completed with no problems

int seReadDisplayByte(int DevID, DWORD Offset, BYTE *pByte)

Description: Reads a byte from the display buffer at the specified offset and returns the value in pByte.

Parameters:
DevID - registered device ID
Offset - offset, in bytes from start of the display buffer, to read from
pByte - pointer to a BYTE to return the value in

Return Value: ERR_OK - operation completed with no problems
ERR_HAL_BAD_ARG - if the value for Addr is greater 40 kb

int seReadDisplayWord(int DevID, DWORD Offset, WORD *pWord)

Description: Reads a word from the display buffer at the specified offset and returns the value in pWord.

Parameters:
DevID - registered device ID
Offset - offset, in bytes from start of the display buffer, to read from
pWord - pointer to a WORD to return the value in

Return Value: ERR_OK - operation completed with no problems.
ERR_HAL_BAD_ARG - if the value for Addr is greater than 40 kb.

int seReadDisplayDword(int DevID, DWORD Offset, DWORD *pDword)

Description: Reads a dword from the display buffer at the specified offset and returns the value in pDword.

Parameters:
DevID - registered device ID
Offset - offset from start of the display buffer to read from
pDword - pointer to a DWORD to return the value in

Return Value: ERR_OK - operation completed with no problems.
ERR_HAL_BAD_ARG - if the value for Addr is greater than 40 kb.

int seWriteDisplayBytes(int DevID, DWORD Offset, BYTE Value, DWORD Count)

Description: This routine writes one or more bytes to display buffer at the offset specified by Addr. If a count greater than one is specified all bytes will have the same value.

Parameters:
DevID - registered device ID
Offset - offset from start of the display buffer to start writing at
Value - BYTE value to write
Count - number of bytes to write

Return Value: ERR_OK - operation completed with no problems
ERR_HAL_BAD_ARG - if the value for Addr or the value of Addr plus Count is greater than 40 kb.

int seWriteDisplayWords(int DevID, DWORD Offset, WORD Value, DWORD Count)

Description: Writes one or more WORDS to the display buffer at the offset specified by Addr. If a count greater than one is specified all WORDS will have the same value.

Parameters:
DevID - registered device ID
Offset - offset from start of the display buffer
Value - WORD value to write
Count - number of words to write

Return Value: ERR_OK - operation completed with no problems
ERR_HAL_BAD_ARG - if the value for Addr or if Addr plus Count is greater than 40 kb.

int seWriteDisplayDwords(int DevID, DWORD Offset, DWORD Value, DWORD Count)

Description: Writes one or more DWORDS to the display buffer at the offset specified by Addr. If a count greater than one is specified all DWORDSs will have the same value.

Parameters:
DevID - registered device ID
Offset - offset from start of the display buffer
Value - DWORD value to write
Count - number of dwords to write

Return Value: ERR_OK - operation completed with no problems
ERR_HAL_BAD_ARG - if the value for Addr or if Addr plus Count is greater than 40 kb.

9.2.5 Power Save

This section covers the HAL functions dealing with the Power Save features of the S1D13704.

int seSetPowerSaveMode(int DevID, int PwrSaveMode)

Description: This function sets on the S1D13704's software selectable power save modes.

Parameters: DevID - a registered device ID
PwrSaveMode - integer value specifying the desired power save mode.

Acceptable values for PwrSaveMode are:
0 - (software power save mode) in this mode registers and memory are read/writable. LCD output is forced low.
3 - (normal operation) all outputs function normally.

Return Value: ERR_OK - operation completed with no problems

9.2.6 Drawing

The Drawing routines cover HAL functions that deal with displaying pixels, lines and shapes.

int seDrawLine(int DevID, int x1, int y1, int x2, int y2, DWORD Color)

Description: This routine draws a line on the display from the endpoints defined by x1,y1 to the endpoint x2,y2 in the requested 'Color'.

Currently seDrawLine() only draws horizontal and vertical lines.

Parameters:
Device - registered device ID.
(x1, y1) - first endpoint of the line in pixels
(x2, y2) - second endpoint of the line in pixels (see note below)
Color - color to draw with. 'Color' is an index into the LUT.

Return Value: ERR_OK - operation completed with no problems

int seDrawRect(int DevID, long x1, long y1, long x2, long y2, DWORD Color, BOOL SolidFill)

Description: This routine draws and optionally fills a rectangular area of display buffer. The upper right corner is defined by x1,y1 and the lower right corner is defined by x2,y2. The color, defined by *Color*, applies both to the border and to the optional fill.

Parameters:

DevID	- registered device ID
x1, y1	- top left corner of the rectangle (in pixels)
x2, y2	- bottom right corner of the rectangle (in pixels)
Color	- The color to draw the rectangle outline and fill with - Color is an index into the Look-Up Table.
SolidFill	- Flag whether to fill the rectangle or simply draw the border. - Set to 0 for no fill, set to non-0 to fill the inside of the rectangle

Return Value: ERR_OK - operation completed with no problems.

9.2.7 LUT Manipulation

These functions deal with altering the color values in the Look-Up Table.

int seSetLut(int DevID, BYTE *pLut, int Count)

Description: This routine writes one or more LUT entries. The writes always start with Look-Up Table index 0 and continue for 'Count' entries.

A Look-Up Table entry consists of three bytes, one each for Red, Green, and Blue. The color information is stored in the four least significant bits of each byte.

Parameters:

DevID	- registered device ID
pLut	- pointer to an array of BYTE lut[16][3] lut[x][0] == RED component lut[x][1] == GREEN component lut[x][2] == BLUE component
Count	- the number of LUT entries to write.

Return Value: ERR_OK - operation completed with no problems

int seGetLut(int DevID, BYTE *pLUT, int Count)

Description: This routine reads one or more LUT entries and puts the result in the byte array pointed to by pLUT.

A Look-Up Table entry consists of three bytes, one each for Red, Green, and Blue. The color information is stored in the four least significant bits of each byte.

Parameters:

DevID	- registered device ID
pLUT	- pointer to an array of BYTE lut[16][3]
	- pLUT must point to enough memory to hold 'Count' x 3 bytes of data.
Count	- the number of LUT elements to read.

Return Value: ERR_OK - operation completed with no problems

int seSetLutEntry(int DevID, int Index, BYTE *pEntry)

Description: This routine writes one LUT entry. Unlike seSetLut, the LUT entry indicated by 'Index' can be any value from 0 to 15.

A Look-Up Table entry consists of three bytes, one each for Red, Green, and Blue. The color information is stored in the four least significant bits of each byte.

Parameters:

DevID	- registered device ID
Index	- index to LUT entry (0 to 15)
pLUT	- pointer to an array of three bytes.

Return Value: ERR_OK - operation completed with no problems

int seGetLutEntry(int DevID, int index, BYTE *pEntry)

Description: This routine reads one LUT entry from any index.

A Look-Up Table entry consists of three bytes, one each for Red, Green, and Blue. The color information is stored in the four least significant bits of each byte.

Parameters:

DevID	- registered device ID
Index	- index to LUT entry (0 to 15)
pEntry	- pointer to an array of three bytes

Return Value: ERR_OK - operation completed with no problems

10 Sample Code

10.1 Introduction

Included in the sample code section are two examples of programing the S1D13704. The first sample uses the HAL to draw a red square, wait for user input then rotates to SwivelView mode and draws a blue square. The second sample code performs the same procedures but directly accesses the registers of the S1D13704. These code samples are for example purposes only.

10.1.1 Sample code using the S1D13704 HAL API

```
/*
**========================================================================
** SAMPLE1.C - Sample code demonstating a program using the S1D13704 HAL.
**------------------------------------------------------------------------
** Created 1998, Vancouver Design Centre
** Copyright (c) 1998 Epson Research and Development, Inc.
** All Rights Reserved.
**------------------------------------------------------------------------
**
** The HAL API code is configured for the following:
**
** 320x240 Single Color 8-bit STN (format 2)
** 4 bpp - 70 Hz Frame Rate (25 MHz CLKi)
** High Performance enabled
**
**========================================================================
*/
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "hal.h"               /* Structures, constants and prototypes. */
#include "appcfg.h"            /* HAL configuration information. */
/*------------------------------------------------------------------------*/
void main(void)
{
       int ChipId;
       int Device;

       /*
       ** Initialize the HAL.
       ** The call to seRegisterDevice() actually prepares the HAL library
       ** for use. The S1D13704 is not accessed.
       */
       if (ERR_OK != seRegisterDevice(&HalInfo, &Device))
```

```
{
      printf("\nERROR: Could not register S1D13704 device.");
      exit(1);
}

/*
** Get the product code to verify this is an S1D13704.
** NOTE: If the S1D13704 design is modified then the
**       product identification change. Additional IDs
**       will have to be checked for.
*/
seGetId(Device, &ChipId);
if (ID_S1D13704F00A != ChipId)
{
      printf("\nERROR: Did not detect an S1D13704.");
      exit(1);
}

/*
** Initialize the S1D13704.
** This step programs the registers with values taken from
** the HalInfo struct in appcfg.h.
*/
if (ERR_OK != seSetInit(Device))
{
      printf("\nERROR: Could not initialize device.");
      exit(1);
}

/*
** The default initialization cleared the display.
** Draw a 100x100 red rectangle in the upper left corner (0,0)
** of the display.
*/
seDrawRect(Device, 0, 0, 100, 100, 1, TRUE);

/*
** Pause here.
*/
getch();

/*
** Clear the display. Do this by writing 40960 bytes
*/
seWriteDisplayBytes(Device, 0, 0, FORTY_K);

/*
** Setup SwivelView mode.
*/
```

```
        seSetHWRotate(Device, PORTRAIT);

        /*
        ** Draw a solid blue 100x100 rectangle in center of the display.
        ** This starting co-ordinates, assuming a 320x240 display is
        ** (320-100)/2 , (240-100)/2 = 110,70.
        */
        seDrawRect(Device, 110, 70, 210, 170, 2, TRUE);

        /*
        ** Done!
        */
        exit(0);
}
```

10.1.2 Sample code without using the S1D13704 HAL API

This second sample demonstrates exactly the same sequence as the first howerver the HAL is not used, all manipulation is done by manually adjusting the registers.

```
/*
**========================================================================
** SAMPLE2.C - Sample code demonstating a direct access of the S1D13704.
**------------------------------------------------------------------------
** Created 1998, Vancouver Design Centre
** Copyright (c) 1998 Epson Research and Development, Inc.
** All Rights Reserved.
**------------------------------------------------------------------------
**
** The sample code using direct S1D13704 access
** will configure for the following:
**
** 320x240 Single Color 8-bit STN (format 2)
** 4 bpp - 70 Hz Frame Rate (25 MHz CLKi)
** High Performance enabled
**
** Notes:
** 1) This code is pseudo-C code intended to show technique.
**    It is assumed that pointers can access the relevant memory addresses.
** 2) Register setup is done with discreet writes rather than being table
**    driven. This allows for clearer commenting. It is more efficient to
**    loop through the array writing each element to a control register.
** 3) The array of register values as produced by 1374CFG.EXE is included
**    here. I used the values directly rather than refer to the register
**    array in the sample code.
**
**========================================================================
*/

#include <conio.h>
/*
** Look-up table for 4 bpp color.
*/
unsigned char Color_4BPP[16*3] =
{
        0x00, 0x00, 0x00,/* BLACK  */
        0x00, 0x00, 0x0A,/* BLUE   */
        0x00, 0x0A, 0x00,/* GREEN  */
        0x00, 0x0A, 0x0A,/* CYAN   */
        0x0A, 0x00, 0x00,/* RED    */
        0x0A, 0x00, 0x0A,/* PURPLE */
        0x0A, 0x0A, 0x00,/* YELLOW */
        0x0A, 0x0A, 0x0A,/* WHITE  */
        0x00, 0x00, 0x00,/* BLACK     */
```

```
        0x00, 0x00, 0x0F,/* LT BLUE   */
        0x00, 0x0F, 0x00,/* LT GREEN  */
        0x00, 0x0F, 0x0F,/* LT CYAN   */
        0x0F, 0x00, 0x00,/* LT RED    */
        0x0F, 0x00, 0x0F,/* LT PURPLE */
        0x0F, 0x0F, 0x00,/* LT YELLOW */
        0x0F, 0x0F, 0x0F/* LT WHITE  */
};

/*
** Register data for the configuratin described above.
** These values were generated using 1374CFG.EXE.
** The sample code uses these values but does not refer to this array.
*/
unsigned char Reg[0x20] = {
  0x00,  0x23,  0xB0,  0x03,  0x27,  0xEF,  0x00,  0x00,
  0x1E,  0x00,  0x3B,  0x00,  0x00,  0x00,  0x00,  0x00,
  0x00,  0x00,  0x00,  0xFF,  0x03,  0x00,  0x00,  0x00,
  0x00,  0x00,  0x00,  0x00,  0x00,  0x00,  0x00,  0x00,
};

/*
** Useful definitions, constants and macros to make the sample code
** easier to follow.
*/
#define MEM_OFFSET 0x01374B0B        /* Location is platform dependent */
#define REG_OFFSET MEM_OFFSET + 0xFFE0/* Memory offset + 64K - 0x20     */
#define MEM_SIZE  0xA000             /* 40 kb display buffer.          */
typedef unsigned char BYTE;          /* Some usefule typedefs */
typedef BYTE far * LPBYTE;
typedef unsigned short WORD;
#define LOBYTE(w)          ((BYTE)(w))
#define HIBYTE(w)          ((BYTE)(((WORD)(w) >> 8) & 0xFF))
#define SET_REG(idx, val)  (*(LPBYTE)(REG_OFFSET + idx)) = (val)

/*-----------------------------------------------------------------------*/

void main(void)
{
        LPBYTE pRegs = (LPBYTE)REG_OFFSET;
        LPBYTE pMem  = (LPBYTE)MEM_OFFSET;
        LPBYTE pLUT;
        int   LUTcount, RGBcount;
        int x, y, tmp;
        int BitsPerPixel = 4;
        int Width      = 320;
        int Height     = 240;
        int OffsetBytes;
```

```
/*
** Check the revision code. Exit if we don't find an S1D13704.
*/
if (0x18 != *pRegs)
      return;

/*
** Initialize the chip.
** Each register is individually programmed to make comments clearer.
*/
/*
** Register 01h: Mode Register 0 - Color, 8-bit format 2
*/
SET_REG(0x01, 0x23);

/*
** Register 02h: Mode Register 1 - 4BPP,  High Performance, CLKi/2
*/
SET_REG(0x02, 0xB0);

/*
** Register 03h: Mode Register 2 - Normal power mode
*/
SET_REG(0x03, 0x03);

/*
** Register 04h: Horizontal Panel Size - 320 pixels - (320/8)-1 = 39 = 27h
*/
SET_REG(0x04, 0x27);

/*
** Register 05h: Vertical Panel Size LSB - 240 pixels
** Register 06h: Vertical Panel Size MSB - (240 - 1) = 239 = EFh
*/
SET_REG(0x05, 0xEF);
SET_REG(0x06, 0x00);

/*
** Register 07h - FPLINE Start Position - not used by STN
*/
SET_REG(0x07, 0x00);

/*
** Register 08h - Horizontal Non-Display Period
**              - HNDP and VNDP are calculated to achieve the
**                desired frame rate according to:
**
**                                      PCLK
```

```
**               Frame Rate = ---------------------------
**                            (HDP + HNDP) * (VDP + VNDP)
*/
SET_REG(0x08, 0x1E);

/*
** Register 09h - FPFRAME Start Position - not used by STN
*/
SET_REG(0x09, 0x00);

/*
** Register 0Ah - Vertical Non-Display Register
**              - CAlculated in conjunction with register 08h (HNDP) to
**                achieve the desired frame rate.
*/
SET_REG(0x0A, 0x3B);

/*
** Register 0Bh - MOD Rate - not used by this panel
*/
SET_REG(0x0B, 0x00);

/*
** Register 0Ch - Screen 1 Start Word Address LSB
** Register 0Dh - Screen 1 Start Word Address MSB
**              - Start address should be set to 0
*/
SET_REG(0x0C, 0x00);
SET_REG(0x0D, 0x00);

/*
** Register 0Fh - Screen 2 Start Word Address LSB
** Register 10h - Screen 2 Start Word Address MSB
**              - Set this start address to 0 too
*/
SET_REG(0x0F, 0x00);
SET_REG(0x10, 0x00);

/*
** Register 12h - Memory Address Offset
**              - Used for setting memory to a width greater than the
**                display size. Usually set to 0 during initialization
**                and programmed to desired value later.
*/
SET_REG(0x12, 0x00);

/*
** Register 13h - Screen 1 Vertical Size LSB
** Register 14h - Screen 1 Vertical Size MSB
```

```
**              - Set to maximum (i.e. 0x3FF). This register is used
**                for split screen operation and should be set to 0
**                during initialization.
*/
SET_REG(0x13, 0xFF);
SET_REG(0x14, 0x03);

/*
** Look-Up Table
** In this example the LUT will be programmed in the register sequence.
** In practice the LUT would probably be done after the other registers.
*/
/*
** Register 15h - Look-Up Table Address
**              - Set to 0 to start RGB sequencing at the first LUT entry.
*/
SET_REG(0x15, 0x00);

/*
** Register 16h - Look-Up Table Bank Select
**              - Set all the banks to 0.
**              - At 4BPP this makes no difference however it will affect
**                appearance at other color depths.
*/
SET_REG(0x16, 0x00);

/*
** Register 17h - Look-Up Table Data
**              - Write 16 RGB triplets to setup the LUT for 4BPP operation.
**              - The LUT is 16 elements deep, 4BPP uses all the idices.
*/
pLUT = Color_4BPP;
for (LUTcount = 0; LUTcount < 16; LUTcount++)
{
      for (RGBcount = 0; RGBcount < 3; RGBcount++)
      {
            SET_REG(0x17, *pLUT);
            pLUT++;
      }
}

/*
** Register 18h - GPIO Configuration - set to 0
**              - '0' configures the GPIO pins for input (power on default)
*/
SET_REG(0x18, 0x00);
/*
** Register 19h - GPIO Status - set to 0
**              - This step has no reason other than it programs the GPIO
```

```
**                values low should the pins get configured as outputs.
*/
SET_REG(0x19, 0x00);

/*
** Register 1Ah - Scratch Pad - set to 0
*/
SET_REG(0x1A, 0x00);

/*
** Register 1Bh - SwivelView Mode - set to 0 - disable SwivelView mode
*/
SET_REG(0x1B, 0x00);

/*
** Register 1Ch - Line Byte Count - set to 0 - Not used by landscape mode
*/
SET_REG(0x0C, 0x00);

/*
** Register 1Fh - TestMode - set to 0
*/
SET_REG(0x1F, 0x00);

/*
** Draw a 100x100 red rectangle in the upper left corner (0,0)
** of the display.
*/
for (y = 0; y < 100; y++)
{
      /*
      ** Set the memory pointer at the start of each line.
      **   Pointer = MEM_OFFSET + (Y * Line_Width * BPP / 8) + (X * BPP / 8)
      */
      pMem = (LPBYTE)MEM_OFFSET + (y * 320 * BitsPerPixel / 8) + 0;
      for (x = 0; x < 100; x+=2)
      {
            *pMem = 0x44;           /* Draws 2 pixels with LUT color 4 */
            pMem++;
      }
}

/*
** Pause here.
*/
getch();

/*
** Clear the display, and all of video memory, by writing 40960 bytes of 0.
```

```
** This is done because an image in display memory is not rotated with the
** switch to SwivelView mode we are about to make.
*/
pMem = (LPBYTE)MEM_OFFSET;
do {
 *pMem = 0;
 pMem++;
} while (pMem < (LPBYTE)(MEM_OFFSET + MEM_SIZE));

/*
** SwivelView mode.
*/
/*
** We will use the default SwivelView mode scheme so we have to adjust
** the ROTATED width to be a power of 2.
** (NOTE: current height will become the rotated width)
*/
tmp = 1;
while (Height > (1 << tmp))
      tmp++;
Height = (1 << tmp);
OffsetBytes = Height * BitsPerPixel / 8;

/*
** Set:
** 1) Line Byte Count to size of the ROTATED width (i.e. current height)
** 2) Start Address to the offset of the width of the ROTATED display.
**    (in SwivelView mode the start address registers point to bytes)
*/
SET_REG(0x1C, (BYTE)OffsetBytes);
OffsetBytes--;
SET_REG(0x0C, LOBYTE(OffsetBytes));
SET_REG(0x0D, HIBYTE(OffsetBytes));

/*
** Set SwivelView mode.
** Use the non-X2 (default) scheme so we don't have to re-calc the frame
** rate. MCLK will be <= 25 MHz so we can leave auto-switch enabled.
*/
SET_REG(0x1B, 0x80);

/*
** Draw a solid blue 100x100 rectangle centered on the display.
** Starting co-ordinates, assuming a 320x240 display are:
**   (320-100)/2 , (240-100)/2 = 110,70.
*/
for (y = 70; y < 180; y++)
{
      /*
```

```
            ** Set the memory pointer at the start of each line.
            **    Pointer = MEM_OFFSET + (Y * Line_Width * BPP / 8) + (X * BPP / 8)
            ** NOTICE that in SwivelView mode we will use a value of 256
            ** for the line width value (not 240).
            */
            x = 110;
            pMem = (LPBYTE)MEM_OFFSET + (y * 256 * BitsPerPixel / 8) +
                                        (x * BitsPerPixel / 8);
            for (x = 110; x < 210; x+=2)
            {
                  *pMem = 0x11;           /* Draws 2 pixels in LUT color 1 */
                  pMem++;
            }
      }
}
```

10.1.3 Header Files

The header files included here are the required for the HAL sample to compile correctly.

```
/*
**========================================================================
** HAL.H - Typical HAL header file for use with programs written to
**         use the S1D13704 HAL.
**------------------------------------------------------------------------
** Created 1998, Vancouver Design Centre
** Copyright (c) 1998 Epson Research and Development, Inc.
** All Rights Reserved.
**========================================================================
*/
#ifndef _HAL_H_
#define _HAL_H_
#pragma warning(disable:4001)   // Disable the 'single line comment' warning.
#include "hal_regs.h"

/*------------------------------------------------------------------------*/

typedef unsigned char  BYTE;
typedef unsigned short WORD;
typedef unsigned long  DWORD;
typedef unsigned int   UINT;
typedef          int   BOOL;
#ifdef INTEL
  typedef BYTE  far *LPBYTE;
  typedef WORD  far *LPWORD;
  typedef UINT  far *LPUINT;
  typedef DWORD far *LPDWORD;
#else
  typedef BYTE      *LPBYTE;
  typedef WORD      *LPWORD;
  typedef UINT      *LPUINT;
  typedef DWORD     *LPDWORD;
#endif
#ifndef LOBYTE
#define LOBYTE(w)    ((BYTE)(w))
#endif
#ifndef HIBYTE
#define HIBYTE(w)    ((BYTE)(((UINT)(w) >> 8) & 0xFF))
#endif
#ifndef LOWORD
#define LOWORD(l)    ((WORD)(DWORD)(l))
#endif
#ifndef HIWORD
#define HIWORD(l)    ((WORD)((((DWORD)(l)) >> 16) & 0xFFFF))
#endif
```

```
#ifndef MAKEWORD
#define MAKEWORD(lo, hi) ((WORD)(((WORD)(lo)) | (((WORD)(hi)) << 8)) )
#endif
#ifndef MAKELONG
#define MAKELONG(lo, hi) ((long)(((WORD)(lo)) | (((DWORD)((WORD)(hi))) << 16)))
#endif
#ifndef TRUE
#define TRUE   1
#endif
#ifndef FALSE
#define FALSE  0
#endif
#define OFF 0
#define ON  1
#define SCREEN1 1
#define SCREEN22
/*

** Constants for HW rotate support
*/
#define DEFAULT0
#define LANDSCAPE 1
#define PORTRAIT2
#ifndef NULL
#ifdef __cplusplus
#define NULL    0
#else
#define NULL    ((void *)0)
#endif
#endif
/*-----------------------------------------------------------------------*/
/*
** SIZE_VERSION  is the size of the version string (eg. "1.00")
** SIZE_STATUS   is the size of the status string (eg. "b" for beta)
** SIZE_REVISION is the size of the status revision string (eg. "00")
*/
#define SIZE_VERSION5
#define SIZE_STATUS 2
#define SIZE_REVISION3
#ifdef ENABLE_DPF   /* Debug_printf() */
#define DPF(exp)  printf(#exp "\n")
#define DPF1(exp) printf(#exp " = %d\n", exp)
#define DPF2(exp1, exp2) printf(#exp1 "=%d  " #exp2 "=%d\n", exp1, exp2)
#define DPFL(exp) printf(#exp " = %x\n", exp)
#else
#define DPF(exp) ((void)0)
#define DPF1(exp) ((void)0)
#define DPFL(exp) ((void)0)
#endif
```

```
/*-----------------------------------------------------------------------*/
enum
{
        ERR_OK = 0,             /* No error, call was successful. */
        ERR_FAILED,             /* General purpose failure.       */
        ERR_UNKNOWN_DEVICE,     /* */
        ERR_INVALID_PARAMETER,  /* Function was called with invalid parameter. */
        ERR_HAL_BAD_ARG,
        ERR_TOOMANY_DEVS
};
/****************************************
 * Definitions for seGetId()
 ****************************************/
#define PRODUCT_ID 0x18
enum
{
        ID_UNKNOWN,
        ID_S1D13704,
        ID_S1D13704F00A
};

#define MAX_MEM_ADDR 40960 -1
#define FORTY_K                         40960
#define MAX_DEVICE   10
#define SE_RSVD                          0

/*
** DetectEndian is used to determine whether the most significant
** and least significant bytes are reversed by the given compiler.
 */
#define ENDIAN       0x1234
#define REV_ENDIAN   0x3412
/****************************************
 * Definitions for Internal calculations.
 ****************************************/
#define MIN_NON_DISP_X     32
#define MAX_NON_DISP_X     256
#define MIN_NON_DISP_Y     2
#define MAX_NON_DISP_Y     64
/****************************************
 * Definitions for seSetFont
 ****************************************/
enum
{
        HAL_STDOUT,
        HAL_STDIN,
        HAL_DEVICE_ERR
};
#define FONT_NORMAL          0x00
```

```
#define FONT_DOUBLE_WIDTH    0x01
#define FONT_DOUBLE_HEIGHT   0x02
enum
{
        RED,
        GREEN,
        BLUE
};
/*************************************************************************/
typedef struct tagHalStruct
{
        char  szIdString[16];
        WORD  wDetectEndian;
        WORD  wSize;
        BYTE Reg[MAX_REG + 1];
        DWORD dwClkI;          /* Input Clock Frequency (in kHz) */
        DWORD dwDispMem;                          /* */
        WORD  wFrameRate;                         /* */
} HAL_STRUCT;
typedef HAL_STRUCT * PHAL_STRUCT;
#ifdef INTEL
typedef HAL_STRUCT far * LPHAL_STRUCT;
#else
typedef HAL_STRUCT     * LPHAL_STRUCT;
#endif
/*=======================================================================*/
/*                          FUNCTION  PROTO-TYPES                        */
/*=======================================================================*/

/*---------------------------- Initialization ----------------------------*/
int seRegisterDevice( const LPHAL_STRUCT lpHalInfo, int *Device );
int seSetInit( int device );
int    seInitHal( void );

/*---------------------------- Miscellaneous ----------------------------*/
int  seGetId( int nDevID, int *pId );
void seGetHalVersion( const char **pVersion, const char **pStatus, const char **pSta-
tusRevision );
int seSetBitsPerPixel( int nDevID, int nBitsPerPixel );
int seGetBitsPerPixel( int nDevID, int *pBitsPerPixel );
int seGetBytesPerScanline( int nDevID, int *pBytes );
int seGetScreenSize( int nDevID, int *width, int *height );
void seDelay( int nMilliSeconds );
int seGetLastUsableByte( int nDevID, long *LastByte );
int seSetHighPerformance( int nDevID, BOOL OnOff );

/*---------------------------- Advanced ----------------------------*/

int seSetHWRotate( int nDevID, int nMode );
```

```
int seSplitInit( int nDevID, WORD Scrn1Addr, WORD Scrn2Addr );
int seSplitScreen( int nDevID, int WhichScreen, int VisibleScanlines );
int seVirtInit( int nDevID, int xVirt, long *yVirt );
int seVirtMove( int nDevID, int nWhichScreen, int x, int y );

/*------------------------ Register/Memory Access -------------------------*/
int seGetReg( int nDevID, int index, BYTE *pValue );
int seSetReg( int nDevID, int index, BYTE value );
int seReadDisplayByte( int nDevID, DWORD offset, BYTE *pByte );
int seReadDisplayWord( int nDevID, DWORD offset, WORD *pWord );
int seReadDisplayDword( int nDevID, DWORD offset, DWORD *pDword );
int seWriteDisplayBytes( int nDevID, DWORD addr, BYTE val, DWORD count );
int seWriteDisplayWords( int nDevID, DWORD addr, WORD val, DWORD count );
int seWriteDisplayDwords( int nDevID, DWORD addr, DWORD val, DWORD count );

/*--------------------------------- Power Save ----------------------------*/
int seHWSuspend( int nDevID, BOOL val );
int seSetPowerSaveMode( int nDevID, int PowerSaveMode );

/*---------------------------------- Drawing ------------------------------*/
// int seSetPixel( int nDevID, int x, int y, DWORD color );
// int seGetPixel( int nDevID, int x, int y, DWORD *pVal );
int seDrawLine( int nDevID, int x1, int y1, int x2, int y2, DWORD color );
int seDrawRect( int nDevID, int x1, int y1, int x2, int y2, DWORD color, BOOL Solid-
fill );
// int seDrawCircle( int nDevID, int xCenter, int yCenter, int radius, DWORD color,
BYTE SolidFill );

/*------------------------------ Text -------------------------------------*/
// int seDrawText( int nDevID, char *fmt, ... );
// int seSetCursor( int row, int col);
// int seSetColor( DWORD foreground, DWORD background);
// int seSetFont( BYTE FontSize, BYTE FontAttr);

/*------------------------------ Color ------------------------------------*/
int seSetLut( int nDevID, BYTE *pLut );
int seGetLut( int nDevID, BYTE *pLut );
int seSetLutEntry( int nDevID, int index, BYTE *pEntry );
int seGetLutEntry( int nDevID, int index, BYTE *pEntry );

#endif      /* _HAL_H_ */

/*-----------------------------------------------------------------------*/

/*
**=======================================================================
** APPCFG.H - Application configuration information.
**-----------------------------------------------------------------------
** Created 1998, Vancouver Design Centre
```

```
** Copyright (c) 1998 Epson Research and Development, Inc.
** All Rights Reserved.
**--------------------------------------------------------------------------
**
** The data in this file was generated using 1374CFG.EXE.
**
** The configureation parameters chosen were:
**   320x240 Single Color 8-bit STN (format 2)
**   4 bpp - 70 Hz Frame Rate (25 MHz CLKi)
**   High Performance enabled
**
**==========================================================================
*/

/*************************************************************/
/*  1374 HAL HDR       (do not remove)                       */
/*  HAL_STRUCT Information generated by 1374CFG.EXE          */
/*  Copyright (c) 1998 Seiko Epson Corp. All rights reserved. */
/*                                                           */
/*  Include this file ONCE in your primary source file       */
/*************************************************************/
HAL_STRUCT HalInfo =
{
  "1374 HAL EXE",     /* ID string   */
  0x1234,             /* Detect Endian */
  sizeof(HAL_STRUCT), /* Size          */
  0x00,  0x23,  0xB0,  0x03,  0x27,  0xEF,  0x00,  0x00,
  0x1E,  0x00,  0x3B,  0x00,  0x00,  0x00,  0x00,  0x00,
  0x00,  0x00,  0x00,  0xFF,  0x03,  0x00,  0x00,  0x00,
  0x00,  0x00,  0x00,  0x00,  0x00,  0x00,  0x00,  0x00,
  25000,                                  /* ClkI (kHz)      */
  0xD0000,                                /* Display Address */
  70,                                     ** Panel Frame Rate (Hz) */
};

/*--------------------------------------------------------------------------*/

/*
**==========================================================================
**  HAL_REGS.H
**--------------------------------------------------------------------------
**  Created 1998, Epson Research & Development
**                Vancouver Design Center.
**  Copyright(c) Seiko Epson Corp. 1998.  All rights reserved.
**==========================================================================
*/
#ifndef __HAL_REGS_H__
#define __HAL_REGS_H__
/*
```

```
**      1374 register names
*/
#define REG_REVISION_CODE                           0x00
#define REG_MODE_REGISTER_0                         0x01
#define REG_MODE_REGISTER_1                         0x02
#define REG_MODE_REGISTER_2                         0x03
#define REG_HORZ_PANEL_SIZE                         0x04
#define REG_VERT_PANEL_SIZE_LSB                     0x05
#define REG_VERT_PANEL_SIZE_MSB                     0x06
#define REG_FPLINE_START_POS                        0x07
#define REG_HORZ_NONDISP_PERIOD                     0x08
#define REG_FPFRAME_START_POS                       0x09
#define REG_VERT_NONDISP_PERIOD                     0x0A
#define REG_MOD_RATE                                0x0B
#define REG_SCRN1_START_ADDR_LSB                    0x0C
#define REG_SCRN1_START_ADDR_MSB                    0x0D
#define REG_RESERVED_1                              0x0E
#define REG_SCRN2_START_ADDR_LSB                    0x0F
#define REG_SCRN2_START_ADDR_MSB                    0x10
#define REG_RESERVED_2                              0x11
#define REG_PITCH_ADJUST                            0x12
#define REG_SCRN1_VERT_SIZE_LSB                     0x13
#define REG_SCRN1_VERT_SIZE_MSB                     0x14
#define REG_LUT_ADDR                                0x15
#define REG_LUT_BANK_SELECT                         0x16
#define REG_LUT_DATA                                0x17
#define REG_GPIO_CONFIG                             0x18
#define REG_GPIO_STATUS                             0x19
#define REG_SCRATCHPAD                              0x1A
#define REG_PORTRAIT_MODE                           0x1B
#define REG_LINE_BYTE_COUNT                         0x1C
#define REG_NOT_PRESENT_1                           0x1D
#define REG_FRAMING                                 0x1E
#define REG_TEST_MODE                               0x1F
/*
** WARNING!!! MAX_REG must be the last available register!!!
*/
#define MAX_REG                                     0x1F
#endif        /*  __HAL_REGS_H__ */
```

Appendix A Supported Panel Values

A.1 Introduction

Future versions of this document will supply example tables for programming the S1D13704 for different panels.

THIS PAGE LEFT BLANK

S1D13704 Register Summary

REG[00h] Revision Code Register [1] IO address = FFE0h [2], RO

Product Code = 000110						Revision Code = 00	
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0

REG[01h] Mode Register 0 IO address = FFE1h, RW

TFT/STN	Dual/Single	Color/Mono [3]	FPLine Polarity	FPFrame Polarity	Mask FPSHIFT	Data Width [4]	
						Bit 1	Bit 0

REG[02h] Mode Register 1 IO address = FFE2h, RW

Bit-Per-Pixel [3]		High [5] Performance	Input Clock Div (CLKI/2)	Display Blank	Frame Repeat	Hw Video Invert Enable	Software Video Invert
Bit 1	Bit 0						

REG[03h] Mode Register 2 IO address = FFE3h, RW

Look-Up Table Bypass	n/a	n/a	n/a	LCDPWR Override	Hardware PS Enable	Sw Power Save [6]	
						Bit 1	Bit 0

REG[04h] Horizontal Panel Size Register IO address = FFE4h, RW

n/a	Horizontal Panel Size = 8(REG + 1)						
	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[05h] Vertical Panel Size Register (LSB) IO address = FFE5h, RW

Vertical Panel Size = (REG[05h], REG[06h]) + 1							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[06h] Vertical Panel Size Register (MSB) IO address = FFE6h, RW

n/a	n/a	n/a	n/a	n/a	n/a	Vertical Panel Size	
						Bit 9	Bit 8

REG[07h] FPLine Start Position IO address = FFE7h, RW

n/a	n/a	n/a	FPLine Start Position = 8(REG[07h] + 2)				
			Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[08h] Horizontal Non-Display Period IO address = FFE8h, RW

n/a	n/a	n/a	Horizontal Non-Display Period = 8(REG + 4)				
			Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[09h] FPFrame Start Position IO address = FFE9h, RW

n/a	n/a	FPFrame Start Position					
		Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[0Ah] Vertical Non-Display Period Register IO address = FFEAh, RW

Vert Non-Disp Status	n/a	Vertical Non-Display Period					
		Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[0Bh] MOD Rate Register IO address = FFEBh, RW

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

REG[0Ch] Screen 1 Start Word Address Register (LSB) IO address = FFECh, RW

Screen 1 Start Word Address = (REG[0Ch], REG[0Dh])							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[0Dh] Screen 1 Start Word Address Register (MSB) IO address = FFEDh, RW

reserved	Screen 1 Start Word Address						
	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

REG[0Fh] Screen 2 Start Word Address Register (LSB) IO address = FFEFh, RW

Screen 2 Start Word Address = (REG[0F], REG[10h])							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[10h] Screen 2 Start Word Address Register (MSB) IO address = FFF0h, RW

Screen 2 Start Word Address							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

REG[12h] Memory Address Offset Register IO address = FFF2h, RW

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

REG[13h] Screen 1 Vertical Size Register (LSB) IO address = FFF3h, RW

Screen 1 Vertical Size = (REG[13h], REG[14h])							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[14h] Screen 1 Vertical Size Register (MSB) IO address = FFF4h, RW

n/a	n/a	n/a	n/a	n/a	n/a	Screen 1 Vertical Size	
						Bit 9	Bit 8

REG[15h] Look-Up Table Address Register [7] IO address = FFF5h, RW

n/a	n/a	RGB Index		Look-Up Table Address			
		Bit 1	Bit 0	Bit 3	Bit 2	Bit 1	Bit 0

REG[16h] Look-Up Table Bank Select Register IO address = FFF6h, RW

n/a	n/a	Red Bank Select		Green Bank Select		Blue Bank Select	
		Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0

REG[17h] Look-Up Table Data Register IO address = FFF7h, RW

n/a	n/a	n/a	n/a	Look-Up Table Data			
				Bit 3	Bit 2	Bit 1	Bit 0

REG[18h] GPIO Configuration Control Register IO address = FFF8h, RW

n/a	n/a	n/a	GPIO4 Pin IO Config	GPIO3 Pin IO Config	GPIO2 Pin IO Config	GPIO1 Pin IO Config	GPIO0 Pin IO Config

REG[19h] GPIO Status / Control Register IO address = FFF9h, RW

n/a	n/a	n/a	GPIO4 Pin IO Status	GPIO3 Pin IO Status	GPIO2 Pin IO Status	GPIO1 Pin IO Status	GPIO0 Pin IO Status

REG[1Ah] Scratch Pad Register IO address = FFFAh, RW

Scratch Pad Register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

REG[1Bh] SwivelView Mode Register IO address = FFFBh, RW

SwivelView Mode En.	SwivelView Mode Sel.	n/a	n/a	n/a	reserved	SwivelView Mode PCLK Select	
						Bit 1	Bit 0

REG[1Ch] Line Byte Count Register IO address = FFFCh, RW

Line Byte Count							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Notes

1 These bits are used to identify the S1D13704 at power on / reset.

2 IO addresses are relative to the beginning of display memory.

3 Gray Shade/Color Mode Selection

Color/Mono REG[01] bit 5	Bit-Per-Pixel Bit 1 REG[02] bit 7	Bit-Per-Pixel Bit 0 REG[02] bit 6	Display Mode	
1	0	0	2 Colors	1 Bit-Per-Pixel
		1	4 Colors	2 Bit-Per-Pixel
	1	0	16 Colors	4 Bit-Per-Pixel
		1	256 Colors	8 Bit-Per-Pixel
0	0	0	2 Gray Shade	1 Bit-Per-Pixel
		1	4 Gray Shade	2 Bit-Per-Pixel
	1	0	16 Gray Shade	4 Bit-Per-Pixel
		1	reserved	

4 Panel Data Format

TFT/STN REG[01] bit 7	Color/ Mono REG[01] bit 5	Dual/ Single REG[01 bit 6
0	0	0
		1
	1	0
		1
1	don't ca	

5 High Performance Selection

High Performance	Bit-Per-P Bit 1 REG[02]
0	0
	1
1	X

6 Power Save Mode Selection

Power Save Bit 1	Po
0	
0	
1	
1	

7 Look-Up Table Access

Color/Mono REG[01h] bit 5	REG[15h] bit 5	bit
0	X	X
1	0	0
1	0	1
1	1	0
1	1	1

S1D13704 Register Summary

EPSON®

S1D13704 Embedded Memory Color LCD Controller

13704CFG.EXE Configuration Program

Document No. X26A-B-001-02

THIS PAGE LEFT BLANK

Table of Contents

List of Figures

THIS PAGE LEFT BLANK

Introduction

13704CFG is a Win 32 program which gives developers an easy means to modify panel types, clock rates, color depths, etc. for S1D13704 demonstration programs.

13704CFG can:

- Read programs, based on the 13704 Hardware Abstraction Layer (HAL), modify the settings and write the changes back to the file. The ability to read, modify and write bypasses having to recompile after every change.
- Write C header files containing register settings which can be used to initialize the 13704 registers in programs which do not use the HAL.

Program Requirements

This program is designed to run under Windows 95/98 or Windows NT 4.0

Installation

There is no installation program for 13704CFG. Installation to a local drive is done by copying 13704CFG.EXE and 13704CFG.HLP to your hard drive and optionally creating a link on the Windows desktop for easy access to the program.

Usage

Open the drive and folder where you copied 13704CFG.EXE and double click the icon to start the program. Optionally, if you created a link to the program on your desktop, double click the link icon.

13704CFG

The 13704CFG window has four main sections: Panel information (includes Dimensions), LookUp Table, Miscellaneous Options, and System settings.

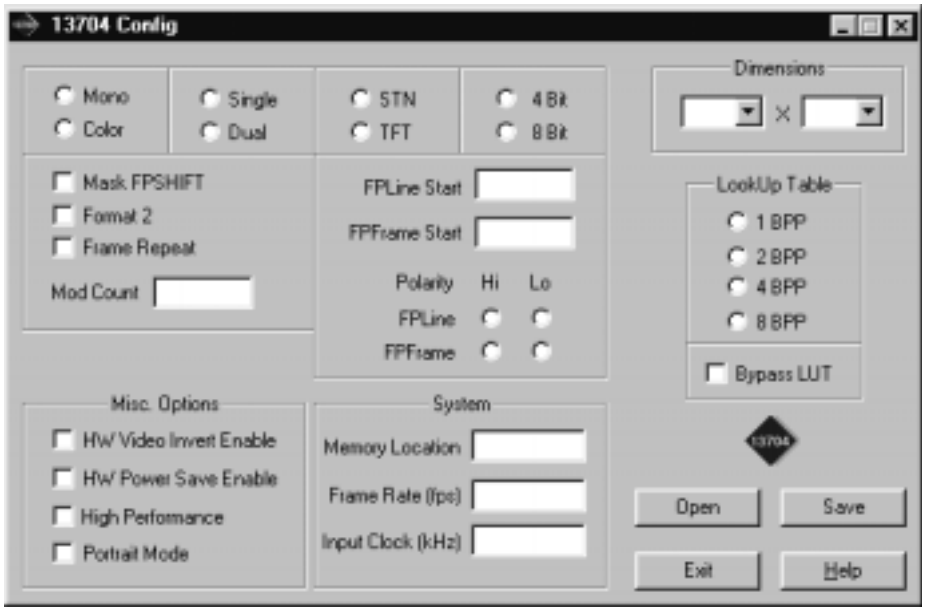


Figure 1: 13704CFG Window

The following sections describe each of the main sections of the configuration dialog box.

Panel Information

Figure 2: Panel Information

This section of the 13704CFG dialog describes the panel connected to the S1D13704. Each of the settings are described briefly below.

- **Mono / Color** – select mono for monochrome panels or color for color panels.

  This option is STN specific and is disabled if TFT is selected.

- **Single / Dual** – select single when connected to a single panel or dual for connection to a dual panel.

  This option is STN specific and is disabled if TFT is selected.

- **STN / TFT** – select STN for passive panels or TFT for active panels. Switching between these two panel types causes visible changes to take place to the configuration dialog box.

- **4 Bit / 8 Bit** – here the panel data width is selected. When STN panel types are selected the options are 4-bit and 8-bit. When TFT panels are selected the options will be 9-bit and 12-bit.

- **Dimensions** – in the left selection box horizontal pixels can be chosen from the list or typed in; in the right selection box, vertical lines, in pixels, can be selected from the list or typed in.

- **Mask FPSHIFT** – when selected the panel clocking signal FPSHIFT is masked off. This option is required for most newer monochrome panels. When color panel type is selected this option is disabled.

  This option is STN specific and is disabled if TFT is selected.

- **Format 2** – There are two data clocking formats in use by 8-bit color panels. The original clocking scheme was designated to be format 1 and the newer scheme was designated format 2. Select this option for most 8 bit color panels. To date all color panels smaller than 640x480 have been found to be format 2.

  Setting this attribute incorrectly will result in a garbled display but will not damage the panel. The display may appear “cut in half” or possibly horizontally skewed.

  This option is STN specific and is disabled if TFT is selected. It is also disabled if the panel type is selected to be 4-bit or monochrome.

- **Frame Repeat** – is a feature for EL panel support. EL panels use a frame of repeated data as the cue to change their polarization. Without this change in polarization panel quality deteriorates.

  When Frame Repeat is selected an internal counter causes the periodic repeat of one frame of modulated panel. At a frame rate of 72 Hz the repeat period is roughly one hour. When not selected the modulated image is never consecutively repeated.

  This option is STN specific and is disabled if TFT is selected.

- **MOD Count** – the mod count value specifies the number of FPLINEs between toggles of the MOD output signal. When set to “0” (default) the MOD output signal toggles every FPFRAME.

  This field is for passive panels only and is generally only required for older monochrome panels.

- **FPLINE Start** – this field specifies the delay, in an 8 pixel resolution, from the end of a line of display data (FPDAT) to the leading edge of FPLINE.

  This field is a TFT specific setting and is disabled if an STN panel is chosen.

- **FPFRAME Start** - this field specifies the number of lines between the last line of display data (FPDAT) and the leading edge of FPFRAME.

  This field is a TFT specific setting and is disabled if an STN panel is chosen.

- **FPLINE / FPFRAME Polarity** - these settings control the sync pulse direction of the FPLINE and FPFRAME pulses in TFT modes.

  Select the appropriate pulse direction for the panel being connected. Selecting 'Lo' results in an active low sync pulse while 'Hi' results in an active high pulse.

  These settings are TFT specific and are disabled when STN panel is selected. When STN panel type is selected the pulse directions are preset to +ve, +ve.

Miscellaneous Options

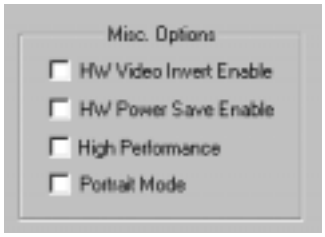


Figure 3: Miscellaneous Options

Miscellaneous options are several items which do not fit into any other category.

- **HW Video Invert Enable** - the S1D13704 supports inverted color output. The color inversion can be toggled by software or in response to a signal applied to pin FPDAT11. In order for the hardware color inversion to succeed this option must be selected.

  - The color inversion is performed on the output from the LUT.
  - HW Video Invert is not availlable for TFT operation.

- **HW Power Save Enable** - the S1D13704 supports two power save modes. One is initiated by software, the second in response to input on the GPIO0 pin. In order for the hardware power save mode to function this option must be selected.

- **High Performance** - improves chip throughput at the expense of power consumption.

  When not selected the internal MCLK signal is divided down version of the internal PCLK signal. Table 1 depicts the ratios when high performance is not selected. The slower MCLKs result in lower power use.

Table 1: MCLK to PCLK ratios

Color Depth (bpp)	Ratio
1	MCLK = PCLK / 8
2	MCLK = PCLK / 4
4	MCLK = PCLK / 2
8	MCLK = PCLK

When this option is selected MCLK == PCLCK at all pixel depths. Running MCLK at higher frequencies results in greater power use.

- **Portrait Mode** - selecting Portrait Mode causes register settings and timings to be saved for portrait mode operation.

  The HAL is capable of performing rotations “on the fly”. Most programs written for the HAL will ignore this setting and set Portrait or Landscape display modes as desired. This setting is useful when the configuration is saved into a C header file to be used by non-HAL programs.

System

The options in the System section describe the items which are required for frame rate calculations and where in CPU address space the S1D13704 will be located.

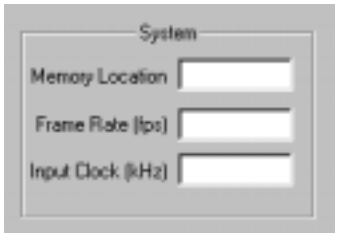


Figure 4: System Options

- **Memory Location** - this describes where in CPU address space the S1D13704 will be located. This setting is required by the HAL to locate the S1D13704. If the settings from 13704CFG will be saved to a C header file for use in a non-HAL program this value does not have to be filled in.
- **Frame Rate** - indicate the desired frame rate here. 13704CFG will attempt to write register settings which result in the requested frame rate. If the frame rate cannot be reached then the following dialog inform the user of the problem

Figure 5: ERROR: Frame Rate

A Frame rate must be entered in order for 13704CFG to complete the frame rate calculations. If no frame rate is entered or the frame rate is set to 0 then the following dialog box will inform the user when they try to save the configuration.

Figure 6: ERROR: Zero Frame Rate

- **Input Clock** - this field specifies the clock rate being applied to the S1D13704 in kHz.

LUT Control

The items in this section control the color depth for the S1D13704 after initialization.

Figure 7: LUT Control

The color depth selections in this section will become enabled or disabled in response to the panel dimensions entered. (i.e. there is only enough memory to operate a 640x480 panel at 1 bit per pixel so the selections for 2 BPP, 4 BPP and 8 BPP would be disabled if this size pane was selected)

- **1 BPP** – sets the color depth to 1 bit per pixel.
- **2 BPP** – sets the color depth to 2 bit per pixel.
- **4 BPP** – sets the color depth to 4 bit per pixel.
- **8 BPP** – sets the color depth to 8 bit per pixel.

- **Bypass LUT** – when selected this option causes the lookup table to be bypassed. Selecting to bypass the lookup table results in a power saving as the lookup table section of the S1D13704 is powered down when this option is selected.

  This option is only applicable for monochrome displays. If a color panel is selected this option is disabled.

  When the lookup table is not enabled then display intensities are dependent on the values in the lookup table. A smaller numerical value in display memory may be displayed with a greater intensity than a larger value.

  When the lookup table is bypassed the colors displayed on the panel are directly proportional to their memory value. (i.e. at 4 bit per pixel; 00h will display as black and 0Fh will display as full intensity)

Open

Click on the Open button to read the settings saved in an executable program based on the S1D13704 hardware abstraction layer.

Clicking the Open button brings up the standard Windows file open dialog.

Figure 8: 13704CFG File Open Dialog

From here the user selects the file to be opened. 13704CFG is capable of opening executable files based on the S1D13704 HAL. Typically the file extension for these file are .EXE for intel platform executables and .S9 for 68k and SH3 platform executables.

Opening a file reads that files HAL configuration information. Use the data read as a starting point in configuring this or other files or to check on the current configuration.

If 13704CFG is unable locate the HAL information in the selected file the following dialog box is displayed.

Figure 9: ERROR: Unable to read HAL

Save

Click on the Close button to save the current configuration settings. When clicked the standard Windows file “Save As” dialog box is displayed.

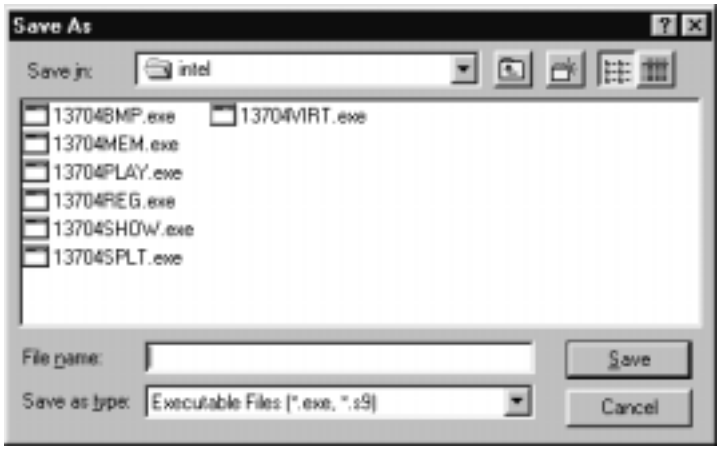


Figure 10: 13704CFG Save As Dialog

From the save as dialog box first select the type of file to save to in the “Save as type:” edit field. 13704CFG currently saves in three file formats.

- .EXE files - are binary images containing a HAL structure for execution on Intel platforms
- .S9 files - are ASCII binary format files used by several embedded systems. The .S9 file is a variation of .S19 files.
- .H files - are ASCII C header files which can be included in other programs.

If an executable file (.EXE or .S9) is selected as the type of file to save to the file being saved to must already exist and be an S1D13704 HAL based program. 13704CFG is cannot save to a non-existent program. If 13704CFG is unable to locate the HAL information in the file being saved to the following dialog box is displayed.

Figure 11: ERROR: Unable to read HAL

Help

Clicking on the Help button will start the help file for S1D13704CFG.

Exit

Clicking on the Exit button exits 13704CFG immediately. The user is not prompted to save any changes they may have made.

Comments

It is assumed that the 13704CFG user is familiar with S1D13704 hardware and software. Refer to the S1D13704 “Functional Hardware Specification,” drawing office number X22A-A-001-xx, and the S1D13704 “Programming Notes and Examples” manual, drawing office number X22A-G-002-xx for information.

EPSON®

S1D13704 Embedded Memory Color LCD Controller

13704SHOW Demonstration Program

Document No. X26A-B-002-02

THIS PAGE LEFT BLANK

13704SHOW

13704SHOW demonstrates S1D13704 display capabilities by drawing a pattern image at different pixel depths (1, 2, 4, and 8 bits-per-pixel) on the display.

13704SHOW must be configured to work with each different hardware platform. Consult documentation for the program 13704CFG.EXE which can be used to configure 13704SHOW.

This software is designed to work in a variety of embedded and personal computer (PC) environments. For embedded environments the model employed is that of host-target. It is assumed that the system has a means of downloading software from the host to the target platform. Typically this is done by a serial communication link. Alternative methods include EPROM, parallel port connection or network connection. It is beyond the scope of this document to provide support for target/host configurations.

S1D13704 Supported Evaluation Platforms

13704SHOW has been tested with the following S1D13704 supported evaluation platforms:

- PC system with an Intel 80x86 processor.
- M68332BCC (Business Card Computer) board, revision B, with a Motorola MC68332 processor.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.

If the platform you are using is different from the above, please see the S1D13704 Programming Notes and Examples manual, document number X26A-G-002-xx.

Installation

PC platform: copy the file 13704SHOW.EXE to a directory that is in the DOS path on your hard drive.

Embedded platform: download the program 13704SHOW to the system.

Usage

PC platform: at the prompt, type:

`13704show [/a][b=n][/l][/p][/vertical][/noinit][/?]`

Embedded platform: execute **`13704show`** and at the prompt, type the command line argument(s).

Where:	`/a`	automatically cycle through all video modes.
	`b=?`	starts 13704SHOW at a user specified bit-per-pixel (bpp) level, where ? can be: 1, 2, 4, 8.
	`/l`	set landscape mode.
	`/p`	set portrait mode.
	`/vertical`	displays vertical line pattern.
	`/noinit`	bypass register initialization and use values which are currently in the registers.
	`/?`	displays the help screen.

Program Messages

`ERROR: Too many devices registered.`

There are too many display devices attached to the HAL. The HAL currently can manage only one device.

`ERROR: Could not register 13704 device.`

A 13704 device was not found at the configured addresses. Check the configuration address using the 13704CFG configuration program.

`ERROR: Did not find a 13704 device.`

The HAL was unable to read the revision code register on the S1D13704. Ensure that the S1D13704 hardware is installed and that the hardware platform has been set up correctly.

`ERROR: Could not initialize device.`

The HAL failed to initialize the registers.

EPSON®

S1D13704 Embedded Memory Color LCD Controller

13704SPLT Display Utility

Document No. X26A-B-003-02

THIS PAGE LEFT BLANK

13704SPLT

13704SPLT demonstrates S1D13704 split screen capability by showing two different areas of display memory on the screen simultaneously.

Screen 1 memory is located at the start of the display buffer and is filled with horizontal bars. Screen 2 memory is located immediately after Screen 1 in the display buffer and is filled with vertical bars. On either user input or elapsed time, the line compare register value is changed to adjust the amount of display area taken up by each screen.

13704SPLT must be configured to work with each different hardware platform. Consult documentation for the program 13704CFG.EXE which can be used to configure 13704SPLT.

This software is designed to work with a variety of embedded and personal computer (PC) environments. For embedded environments the model employed is that of host-target. It is assumed that the system has a means of downloading software from the host to the target platform. Typically this is done by a serial communication link. Alternative methods include EPROM, parallel port connection or network connection. It is beyond the scope of this document to provide support for target/host configurations.

S1D13704 Supported Evaluation Platforms

13704SPLT has been tested with the following S1D13704 supported evaluation platforms:

- PC system with an Intel 80x86 processor.
- M68332BCC (Business Card Computer) board, revision B, with a Motorola MC68332 processor.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.

If the platform you are using is different from the above, please see the S1D13704 Programming Notes and Examples manual, document number X26A-G-002-xx.

Installation

PC platform: Copy the file 13704SPLT.EXE to a directory that is in the DOS path on your hard drive.

Embedded platform: Download the program 13704SPLT to the system.

Usage

PC platform: at the prompt, type **13704SPLT [/a] [/l] [/p] [/?]**

Embedded platform: execute **`13704splt`** and at the prompt, type the command line argument.

Where:	`no argument`	enables manual split screen operation
	`/a`	enables automatic split screen operation (a timer is used to move screen 2)
	/?	display the help screen

After starting 13704SPLT the following keyboard commands are available.

Manual mode:	↑, u	move Screen 2 up
	↓, d	move Screen 2 down
	`HOME`	covers Screen 1 with Screen 2
	`END`	displays only Screen 1
Automatic mode:	`any key`	change the direction of split screen movement (for PC only)
Both modes:	`b`	changes the color depth (bits-per-pixel)
	`ESC`	exits 13704SPLT

13704SPLT Example

1. Type “13704splt /a” to automatically move the split screen.
2. Press “b” to change the color depth from 1 bit-per-pixel to 2 bit-per-pixel.
3. Repeat step 2 for the remaining color depths (4 and 8 bit-per-pixel).
4. Press <ESC> to exit the program.

Program Messages

`ERROR: Too many devices registered.`

There are too many display devices attached to the HAL. The HAL currently can manage only one device.

`ERROR: Could not register 13704 device.`

A 13704 device was not found at the configured addresses. Check the configuration address using the 13704CFG configuration program.

`ERROR: Did not detect 13704.`

The HAL was unable to read the revision code register on the S1D13704. Ensure that the S1D13704 hardware is installed and that the hardware platform has been set up correctly.

THIS PAGE LEFT BLANK

EPSON®

S1D13704 Embedded Memory Color LCD Controller

13704VIRT Display Utility

Document No. X26A-B-004-02

THIS PAGE LEFT BLANK

13704VIRT

13704VIRT demonstrates the virtual display capability of the S1D13704. A virtual display is where the image to be displayed is larger than the physical display device. The display surface is used a viewing window. The entire image can be seen only by panning and scrolling.

13704VIRT must be configured to work with each different hardware platform. Consult documentation for the program 13704CFG.EXE which can be used to configure 13704VIRT.

This software is designed to work with a variety of embedded and personal computer (PC) environments. For embedded environments the model employed is that of host/target. It is assumed that the system has a means of downloading software from the host to the target platform. Typically this is done by a serial communication link. Alternative methods include EPROM, parallel port connection or network connection. It is beyond the scope of this document to provide support for target/host configurations.

S1D13704 Supported Evaluation Platforms

13704VIRT has been tested with the following S1D13704 supported evaluation platforms:

- PC system with an Intel 80x86 processor.
- M68332BCC (Business Card Computer) board, revision B, with a Motorola MC68332 processor.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.

If the platform you are using is different from the above, please see the S1D13704 Programming Notes and Examples manual, document number X26A-G-002-xx.

Installation

PC platform: copy the file 13704VIRT.EXE to a directory that is in the DOS path on your hard drive.

Embedded platform: download the program 13704VIRT to the system.

Usage

PC platform: at the prompt, type **`13704virt [/a] [/w=???]`**.

Embedded platform: execute **`13704virt`** and at the prompt, type the command line argument.

Where:	`no argument`	panning and scrolling is performed manually (defaults to virtual width = = physical width x 2 and maximum virtual height)
	`/a`	panning and scrolling is performed automatically
	`/w=???`	specifies the virtual display width which includes both on-screen and off-screen size
		the maximum virtual display width for each display mode is: 1 bpp – 4096 pixels 2 bpp – 2048 pixels 4 bpp – 1024 pixels 8 bpp – 512 pixels

The following keyboard commands are for navigation within the program.

Manual mode:	↑	scrolls up
	↓	scrolls down
	←	pans to the left
	→	pans to the right
	`HOME`	moves the display screen so that the upper right of the virtual screen shows in the upper right of the display
	`END`	moves the display screen so that the lower left of the virtual screen shows in the lower left of the display
Automatic mode:	`any key`	changes the direction of screen
Both modes:	`b`	changes the color depth (bits-per-pixel)
	`ESC`	exits 13704VIRT

13704VIRT Example

1. Type “13704virt /a” to automatically pan and scroll.
2. Press "b" to change the bits-per-pixel from 1 bit-per-pixel to 2 bits-per-pixel.
3. Repeat steps 1 and 2 for the remaining color depths (4 and 8 bit-per-pixel).
4. Press <ESC> to exit the program.

Program Messages

`ERROR: Too many devices registered.`

There are too many display devices attached to the HAL. The HAL currently can manage only one device.

`ERROR: Could not register 13704 device.`

A 13704 device was not found at the configured addresses. Check the configuration address using the 13704CFG configuration program.

`ERROR: Did not detect 13704.`

The HAL was unable to read the revision code register on the S1D13704. Ensure that the S1D13704 hardware is installed and that the hardware platform has been set up correctly.

THIS PAGE LEFT BLANK

EPSON®

S1D13704 Embedded Memory Color LCD Controller

13704PLAY Diagnostic Utility

Document No. X26A-B-005-03

THIS PAGE LEFT BLANK

13704PLAY

13704PLAY is a utility which allows the user to easily read/write the S1D13704 registers, Look-up Table and display memory.

The user interface for 13704PLAY 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 and terminal for embedded platforms). This utility requires the target platform to support standard I/O.

13704PLAY commands can be entered interactively using a keyboard/monitor or they can be executed from a script file. Scripting is a powerful feature which allows command sequences played back from a file thus avoiding having to retype lengthy sequences.

13704PLAY must be configured to work with each different hardware platform. Consult documentation for the program 13704CFG.EXE which can be used to configure 13704PLAY.

This software is designed to work with a variety of embedded and personal computer (PC) environments. For embedded environments the model employed is that of host.target. It is assumed that the system has a means of downloading software from the host to the target platform. Typically this is done by a serial communication link. Alternative methods include EPROM, parallel port connection or network connection. It is beyond the scope of this document to provide support for target/host configurations.

S1D13704 Supported Evaluation Platforms

13704PLAY has been tested with the following S1D13704 supported evaluation platforms:

- PC system with an Intel 80x86 processor.
- M68332BCC (Business Card Computer) board, revision B, with a Motorola MC68332 processor.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.

If the platform you are using is different from the above, please see the S1D13704 Programming Notes and Examples manual, document number X26A-G-002-xx.

Installation

PC platform: copy the file 13704PLAY.EXE to a directory that is in the DOS path on your hard drive.

Embedded platform: download the program 13704PLAY to the system.

Usage

PC platform: at the prompt, type **`13704play [/?]`**.

Embedded platform: execute **`13704play`** and at the prompt, type the command line argument.

Where: `/?` displays program revision information.

The following commands are valid within the 13704PLAY program.

`X index [data]`	Reads/writes the registers. Writes data to the register specified by the index when "data" is specified; otherwise the register is read.
`XA`	Reads all registers.
`L index [data1 data2 data3]`	Reads/writes Look-Up Table (LUT) values. Writes data to the LUT index when "data" is specified; otherwise the LUT index is read. Data must consist of 3 bytes: 1 red, 1 green, 1 blue. and range in value from 0x00 to 0x0F.
`LA`	Reads all LUT values.
`F[W] addr1 addr2 data . . .`	Fills bytes or words from address 1 to address 2 with data. Data can be multiple values (e.g. `F 0 20 1 2 3 4` fills address 0 to 0x20 with a repeating pattern of 1 2 3 4).
`R[W] addr [count]`	Reads "count" of bytes or words from the address specified by "addr". If "count" is not specified, then 16 bytes/words are read.
`W[W] addr data . . .`	Writes bytes or words of data to address specified by "addr". Data can be multiple values e.g. `W 0 1 2 3 4` writes the byte values 1 2 3 4 starting at address 0).
`I`	Initializes the chip with user specified configuration.

`M [bpp]`	Returns information about the current mode. If “bpp” is specified then set the requested color depth.
`P 0|1|2`	Sets software power save mode 0-2. Power save mode 0 is normal operation.
`H [lines]`	Halts after specified lines of display. This feature halts the display during long read operations to prevent data from scrolling off the display. Set 0 to disable.
`Q`	Quits this utility.
`?`	Displays Help information.

13704PLAY Example

1. Type “13704PLAY” to start the program.
2. Type "?" for help.
3. Type "i" to initialize the registers.
4. Type "xa" to display the contents of the registers.
5. Type "x 5" to read register 5.
6. Type "x 3 10" to write 10 hex to register 3.
7. Type "f 0 400 aa" to fill the first 400 hex bytes of display memory with AA hex.
8. Type "f 0 a000 aa" to fill 40k bytes of display memory.
9. Type "r 0 ff" to read the first 100 hex bytes of display memory.
10. Type "q" to exit the program.

Scripting

13704PLAY can be driven by a script file. This is useful when:

- there is no standard display output to monitor command entry and results.
- various registers must be quickly changed faster than can achieved by typing.
- The same series of keystrokes is being entered time and again.

A script file is an ASCII text file with one 13704PLAY command per line. All scripts must end with a “q” (quit) command in order to return control to the operating system. The semi-colon is used as a comment delimitor. Everything on a line after the semi-colon will be ignored.

On a PC platform, a typical script command line is: “13704PLAY < dumpregs.scr > results”.

This causes the script file “dumpregs.scr” to be interpreted and the results to be sent to the file “results.”

Example 1: The script file “dumpregs.scr” can be created with and text editor and will look like the following:

```
; This file initializes the S1D13704 and reads the registers
i       ; Initialize the registers.
xa      ; Dump all the registers
la      ; And the LUT
q       ; Exit
```

Comments

- All numeric values are considered to be hexadecimal unless identified otherwise. For example, 10 = 10h = 16 decimal; 10t = 10 decimal; 010b = 2 decimal.
- Redirecting commands from a script file (PC platform) allows those commands to be executed as though they were typed.

Program Messages

ERROR: Too many devices registered.

There are too many display devices attached to the HAL. The HAL currently can manage only one device.

ERROR: Could not register 13704 device.

A 13704 device was not found at the configured addresses. Check the configuration address using the 13704CFG configuration program.

WARNING: Did not detect 13704.

The HAL did not detect an S1D13704, however 13704PLAY will continue to function.

THIS PAGE LEFT BLANK

EPSON®

S1D13704 Embedded Memory Color LCD Controller

13704BMP Demonstration Program

Document No. X26A-B-006-02

THIS PAGE LEFT BLANK

13704BMP

13704BMP demonstrates S1D13704 display capabilities by rendering bitmap images on the display.

The 13704BMP display utility is designed to operate in a personal computer DOS environment and must be configured to work with your display hardware. Consult documentation for the program 13704CFG.EXE which can be used to configure 13704BMP.

13704BMP is not supported on non-PC platforms.

Installation

Copy the file 13704BMP.EXE to a directory that is in the DOS path on your hard drive.

Usage

At the prompt, type:

`13704bmp bmp_file [/a[time]] [/l] [/p] [/?]`.

Where:	`bmp_file`	the name of the file to display
	`/a[time]`	automatic mode returns to the operating system after “time” seconds. If time is not specified the default is 5 seconds. This option is intended for use with batch files to automate displaying a series of images.
	`/l`	override default configuration settings and set landscape display mode.
	`/p`	override default configuration settings and set portrait display mode.
	`/?`	displays the Help screen

Comments

- 13704BMP currently views only Windows BMP format images.

Program Messages

ERROR: Too many devices registered.

There are too many display devices attached to the HAL. The HAL currently can manage only one device.

ERROR: Could not register 13704 device.

A 13704 device was not found at the configured addresses. Check the configuration address using the 13704CFG configuration program.

ERROR: Did not detect 13704.

The HAL was unable to read the revision code register on the S1D13704. Ensure that the S1D13704 hardware is installed and that the hardware platform has been set up correctly.

EPSON®

S1D13704 Embedded Memory Color LCD Controller

13704PWR Power Save Utility

Document Number: X26A-B-007-02

THIS PAGE LEFT BLANK

13704PWR

The 13704PWR Power Save Utility is a tool to assist in the testing of the software and hardware power save modes.

Refer to the section titled “Power Save Modes” in the S1D13704 Programming Notes and Examples manual, document number X26A-G-002-xx, and the S1D13704 Functional Hardware Specification, document number X26A-A-001-xx for further information.

The 13704PWR utility must be configured and/or compiled to work with your hardware platform. Consult documentation for the program 13704CFG.EXE which can be used to configure 13704PWR.

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.

S1D13704 Supported Evaluation Platforms

13704PWR has been designed to work with the following S1D13704 supported evaluation platforms:

- PC system with an Intel 80x86 processor.
- M68332BCC (Business Card Computer) board, revision B, with a Motorola MC68332 processor.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.

If the platform you are using is different from the above, please see the S1D13704 “Programming Notes and Examples” manual, document number X26A-G-002-xx.

Installation

PC platform: copy the file 13704PWR.EXE to a directory that is in the DOS path on your hard drive.

Embedded platform: download the program 13704PWR to the system.

Usage

PC platform: at the prompt, type **`13704pwr [s0] [s1] [h0] [h1]`**.

Embedded platform: execute **`13704pwr`** and at the prompt, type the command line argument.

Where:

`s0`	resets software power save mode
`s1`	sets software power save mode
`h0`	resets (disables) hardware power save mode (REG[03h] bit 2)
`h1`	sets **(**enables) hardware power save mode (REG[03h] bit 2)
`/?`	displays this usage message

Program Messages

`ERROR: Unknown command line argument.`

An invalid command line argument was entered. Enter a valid command line argument.

`ERROR: Too many devices registered.`

There are too many display devices attached to the HAL. The HAL currently can manage only one device.

`ERROR: Could not register 13704 device.`

A 13704 device was not found at the configured addresses. Check the configuration address using the 13704CFG configuration program.

`ERROR: Did not detect 13704.`

The HAL was unable to read the revision code register on the S1D13704. Ensure that the S1D13704 hardware is installed and that the hardware platform has been set up correctly.

EPSON®

S1D13704 Embedded Memory Color LCD Controller

Windows® CE Display Drivers

Document Number: X26A-E-001-02

THIS PAGE LEFT BLANK

1 WINDOWS® CE DISPLAY DRIVERS

The Windows CE display drivers are designed to support the S1D13704 Embedded Memory LCD Controller running under the Microsoft Windows CE operating system. Available drivers include: 4 bit-per-pixel landscape mode, and 4 bit-per-pixel portrait mode.

For updated source code, visit Epson Research and Development on the World Wide Web at www.erd.epson.com, or contact your Seiko Epson sales representative.

1.1 Program Requirements

Video Controller	:	S1D13704
Display Type	:	LCD
Windows Version	:	CE Version 2.0/2.1

1.2 Example Driver Build

Build For CEPC (X86) Version 2.0/2.1

To build a Windows CE v2.0/2.1 display driver for the CEPC (X86) platform using a S5U13704B00C evaluation board, follow the instructions below:

1. Install Microsoft Windows NT v4.0.
2. Install Microsoft Visual C/C++ v5.0.
3. Install the Microsoft Windows CE Embedded Toolkit (ETK) by running SETUP.EXE from the ETK compact disc #1.
4. Create a new project by following the procedure documented in “Creating a New Project Directory” from the Windows CE ETK. Alternately, use the current “DEMO7” project included with the ETK. Follow the steps below to create a “X86 DEMO7” shortcut on the Windows NT v4.0 desktop which uses the current “DEMO7” project:
    a. Right click on the “Start” menu on the taskbar.
    b. Click on the item “Open All Users” and the “Start Menu” window will come up.
    c. Click on the icon “Programs”.
    d. Click on the icon “Windows CE Embedded Development Kit”.
    e. Drag the icon “X86 DEMO1” onto the desktop using the right mouse button.
    f. Click on “Copy Here”.
    g. Rename the icon “X86 DEMO1” on the desktop to “X86 DEMO7” by right clicking on the icon and choosing “rename”.

h. Right click on the icon “X86 DEMO7” and click on “Properties” to bring up the “X86 DEMO7 Properties” window.

i. Replace the string “DEMO1” under the entry “Target” with “DEMO7”.

j. Click on “OK” to finish.

1. Create a sub-directory named 4BPP13704 under \wince\platform\cepc\drivers\display.

2. Copy the source code to the 4BPP13704 subdirectory.

3. Add an entry for the 4BPP13704 in the file \wince\platform\cepc\drivers\display\dirs.

4. Modify the file CONFIG.BIB (using any text editor such as NOTEPAD) to set the system RAM size, the S1D13704 IO port and display buffer address mapping. Note that CONFIG.BIB is located in X:\wince\platform\cepc\files (where X: is the drive letter). Since the S5U13704B00C maps the 64K byte region from D0000h to DFFFFh, make sure no other devices occupy this area. The following lines should be in CONFIG.BIB:

   NK 80200000 00500000 RAMIMGE

   RAM 80700000 00500000 RAM

Note
DISPDRVR.C should include the following:

#define PhysicalPortAddr 0x000DF000L

#define PhysicalVmemAddr 0x000D0000L

5. Edit the file PLATFORM.BIB (located in X:\wince\platform\cepc\files) to set the default display driver to the file 4BPP13704.DLL. 4BPP13704.DLL will be created during the build in step 13.

   You may replace the following lines in PLATFORM.BIB:

```
IF CEPC_DDI_VGA2BPP
ddi.dll    $(_FLATRELEASEDIR)\ddi_vga2.dll  NK SH
ENDIF
IF CEPC_DDI_VGA8BPP
ddi.dll    $(_FLATRELEASEDIR)\ddi_vga8.dll  NK SH
ENDIF
IF CEPC_DDI_VGA2BPP !
IF CEPC_DDI_VGA8BPP !
ddi.dll    $(_FLATRELEASEDIR)\ddi_s364.dll  NK SH
ENDIF
ENDIF
```

   with this line:

```
ddi.dll    $(_FLATRELEASEDIR)\4BPP13704.dll NK SH
```

6. Edit the file DISPDRVR.C (located in X:\wince\platform\odo\drivers\display\4BPP13704) to set the desired screen resolution, color depth (bpp) and panel type. The sample code defaults to a 320x240 color single passive 4-bit LCD panel. To support one of the other listed panels, change the #define statement.

7. Generate the proper building environment by double-clicking on the sample project icon (i.e. X86 DEMO7).

8. Type BLDDEMO <ENTER> at the DOS prompt of the X86 DEMO7 window to generate a Windows CE image file (NK.BIN).

1.3 Example Installation

Installation for CEPC Environment

Windows CE v2.0 can be loaded on a PC using a floppy drive or a hard drive. The two methods are described below:

To load CEPC from a floppy drive:

1. Create a DOS bootable floppy disk.

2. Edit CONFIG.SYS on the floppy disk to contain the following line only.
   device=a:\himem.sys

3. Edit AUTOEXEC.BAT on the floppy disk to contain the following lines.
   mode com1:9600,n,8,1
   loadcepc /B:9600 /C:1 c:\wince\release\nk.bin

4. Copy LOADCEPC.EXE from c:\wince\public\common\oak\bin to the bootable floppy disk.

5. Confirm that NK.BIN is located in c:\wince\release.

6. Reboot the system from the bootable floppy disk.

To load CEPC from a hard drive:

1. Copy LOADCEPC.EXE to the root directory of the hard drive.

2. Edit CONFIG.SYS on the hard drive to contain the following line only.
   device=c:\himem.sys

3. Edit AUTOEXEC.BAT on the hard drive to contain the following lines.
   mode com1:9600,n,8,1
   loadcepc /B:9600 /C:1 c:\wince\release\nk.bin

4. Confirm that NK.BIN is located in c:\wince\release.

5. Reboot the system from the hard drive.

1.4 Comments

- At the time of this printing, the drivers have been tested on the x86 CPUs and have only been run with version 2.0 of the ETK. We are constantly updating the drivers so please check our website at www.erd.epson.com, or contact your Seiko Epson or Epson Electronics America sales representative.

EPSON®

S1D13704 Embedded Memory Color LCD Controller

S5U13704B00C Rev. 1.0 ISA Bus Evaluation Board User Manual

Document Number: X26A-G-005-03

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This manual describes the setup and operation of the S5U13704B00C Rev. 1.0 Evaluation Board. Implemented using the S1D13704 Embedded Memory Color LCD Controller, the S5U13704B00C board is designed for the 16-bit ISA bus environment. To accommodate other bus architectures, the S5U13704B00C board also provides CPU/Bus interface connectors.

For more information regarding the S1D13704, refer to the S1D13704 Hardware Functional Specification, document number X26A-A-001-xx.

1.1 Features

- 80-pin QFP14 package.
- SMT technology for all appropriate devices.
- 4/8-bit monochrome and color passive LCD panel support.
- 9/12-bit LCD TFT/D-TFD panel support.
- Selectable 3.3V or 5.0V LCD panel support.
- Oscillator support for CLKI (up to 50MHz with internal clock divide or 25MHz with no internal clock divide).
- Embedded 40K byte SRAM display buffer for 1/2/4 bit-per-pixel (bpp), 2/4/16 level gray shade display and 1/2/4/8 bpp, 2/4/16/256 level color display.
- Support for software and hardware power save modes.
- On-board adjustable LCD bias positive power supply (+23V to 40V).
- On-board adjustable LCD bias negative power supply (-14V to -24V).
- 16-bit ISA bus support.
- CPU/Bus interface header strips for non-ISA bus support.

2 Installation and Configuration

The S1D13704 has five configuration inputs, CNF[4:0], which are read on the rising edge of RESET# and are fully configurable on this evaluation board. One six-position DIP switch is provided on the board to configure these five configuration inputs and to enable/disable hardware power save mode.

The following settings are recommended when using the S5U13704B00C with the ISA bus.

Table 2-1: Configuration DIP Switch Settings

Switch	Signal	Closed (0 or low)	Open (1 or high)
SW1-1	CNF0	See “Host Bus Selection” table below	See “Host Bus Selection” table below
SW1-2	CNF1		
SW1-3	CNF2		
SW1-4	CNF3	Little Endian	Big Endian
SW1-5	CNF4	Active low LCDPWR signal	Active high LCDPWR signal
SW1-6	GPIO0	Hardware Suspend Disable	Hardware Suspend Enable

= recommended settings (configured for ISA bus support)

Table 2-2: Host Bus Selection

S1-3	S1-2	S1-1	BS#	Host Bus Interface
0	0	0	X	SH-4 bus interface
0	0	1	X	SH-3 bus interface
0	1	0	X	reserved
0	1	1	X	MC68K bus interface #1, 16-bit
1	0	0	X	reserved
1	0	1	X	MC68K bus interface #2, 16-bit
1	1	0	0	reserved
1	1	0	1	reserved
1	1	1	0	Generic #1, 16-bit
1	1	1	1	Generic #2, 16-bit

= recommended settings (configured for ISA bus support)

Table 2-3: Jumper Settings

	Description	1-2	2-3
JP1	IOVDD Selection	5.0V IOVDD	3.3V IOVDD
JP2	RD/WR# Signal Selection	Pulled up to IOVDD	No Connection
JP3	BS# Signal Selection	Pulled up to IOVDD	No Connection
JP4	LCD Panel Voltage Selection	5.0V LCD Panel	3.3V LCD Panel

= recommended settings (configured for ISA bus support)

3 LCD Interface Pin Mapping

Table 3-1: LCD Signal Connector (J5) Pinout

Connector		Single Passive Panel					Dual Passive Panel		Color TFT/D-TFD	
		Color			Mono		Color	Mono		
Pin Name	Pin #	4-bit	8-bit	8-bit Alternate Format	4-bit	8-bit	8-bit	8-bit	9-bit	12-bit
BFPDAT0	1		D0	D0		D0	LD0	LD0	R2	R3
BFPDAT1	3		D1	D1		D1	LD1	LD1	R1	R2
BFPDAT2	5		D2	D2		D2	LD2	LD2	R0	R1
BFPDAT3	7		D3	D3		D3	LD3	LD3	G2	G3
BFPDAT4	9	D0	D4	D4	D0	D4	UD0	UD0	G1	G2
BFPDAT5	11	D1	D5	D5	D1	D5	UD1	UD1	G0	G1
BFPDAT6	13	D2	D6	D6	D2	D6	UD2	UD2	B2	B3
BFPDAT7	15	D3	D7	D7	D3	D7	UD3	UD3	B1	B2
BFPDAT8	17								B0	B1
BFPDAT9	19									R0
BFPDAT10	21									G0
BFPDAT11	23									B0
BFPSHIFT	33	FPSHIFT	FPSHIFT	FPSHIFT	FPSHIFT	FPSHIFT	FPSHIFT	FPSHIFT	FPSHIFT	FPSHIFT
BFPSHIFT2	35		FPSHIFT2							
BFPLINE	37	FPLINE	FPLINE	FPLINE	FPLINE	FPLINE	FPLINE	FPLINE	FPLINE	FPLINE
BFPFRAME	39	FPFRAME	FPFRAME	FPFRAME	FPFRAME	FPFRAME	FPFRAME	FPFRAME	FPFRAME	FPFRAME
GND	2-26 (Even Pins)	GND	GND	GND	GND	GND	GND	GND	GND	GND
N / C	28									
VLCD	30	LCD panel negative bias voltage (-18V to -23V)								
LCDVCC	32	+3.3V or +5V (selectable with JP4)								
+12V	34	+12V	+12V	+12V	+12V	+12V	+12V	+12V	+12V	+12V
VDDH	36	LCD panel positive bias voltage (+24V to +38V)								
BDRDY	38	MOD		MOD	MOD	MOD	MOD	MOD	DRDY	DRDY
BLCDPWR	40	LCDPWR	LCDPWR	LCDPWR	LCDPWR	LCDPWR	LCDPWR	LCDPWR	LCDPWR	LCDPWR

4 CPU/Bus Interface Connector Pinouts

Table 4-1: CPU/BUS Connector (H1) Pinout

Connector Pin No.	CPU/BUS Pin Name	Comments
1	SD0	Connected to DB0 of the S1D13704
2	SD1	Connected to DB1 of the S1D13704
3	SD2	Connected to DB2 of the S1D13704
4	SD3	Connected to DB3 of the S1D13704
5	GND	Ground
6	GND	Ground
7	SD4	Connected to DB4 of the S1D13704
8	SD5	Connected to DB5 of the S1D13704
9	SD6	Connected to DB6 of the S1D13704
10	SD7	Connected to DB7 of the S1D13704
11	GND	Ground
12	GND	Ground
13	SD8	Connected to DB8 of the S1D13704
14	SD9	Connected to DB9 of the S1D13704
15	SD10	Connected to DB10 of the S1D13704
16	SD11	Connected to DB11 of the S1D13704
17	GND	Ground
18	GND	Ground
19	SD12	Connected to DB12 of the S1D13704
20	SD13	Connected to DB13 of the S1D13704
21	SD14	Connected to DB14 of the S1D13704
22	SD15	Connected to DB15 of the S1D13704
23	RESET#	Connected to the RESET# signal of the S1D13704
24	GND	Ground
25	GND	Ground
26	GND	Ground
27	+12V	12 volt supply
28	+12V	12 volt supply
29	WE0#	Connected to the WE0# signal of the S1D13704
30	WAIT#	Connected to the WAIT# signal of the S1D13704
31	CS#	Connected to the CS# signal of the S1D13704
32	NC	Not connected
33	WE1#	Connected to the WE1# signal of the S1D13704
34	NC	Not connected

Table 4-2: CPU/BUS Connector (H2) Pinout

Connector Pin No.	CPU/BUS Pin Name	Comments
1	SA0	Connected to AB0 of the S1D13704
2	SA1	Connected to AB1 of the S1D13704
3	SA2	Connected to AB2 of the S1D13704
4	SA3	Connected to AB3 of the S1D13704
5	SA4	Connected to AB4 of the S1D13704
6	SA5	Connected to AB5 of the S1D13704
7	SA6	Connected to AB6 of the S1D13704
8	SA7	Connected to AB7 of the S1D13704
9	GND	Ground
10	GND	Ground
11	SA8	Connected to AB8 of the S1D13704
12	SA9	Connected to AB9 of the S1D13704
13	SA10	Connected to AB10 of the S1D13704
14	SA11	Connected to AB11 of the S1D13704
15	SA12	Connected to AB12 of the S1D13704
16	SA13	Connected to AB13 of the S1D13704
17	GND	Ground
18	GND	Ground
19	SA14	Connected to AB14 of the S1D13704
20	SA15	Connected to AB14 of the S1D13704
21	SA16	Connected to AB16 of the S1D13704
22	SA17	Connected to AB17 of the S1D13704
23	SA18	Connected to AB18 of the S1D13704
24	SA19	Connected to AB19 of the S1D13704
25	GND	Ground
26	GND	Ground
27	VCC	5 volt supply
28	VCC	5 volt supply
29	RD/WR#	Connected to the R/W# signal of the S1D13704
30	BS#	Connected to the BS# signal of the S1D13704
31	BUSCLK	Connected to the BCLK signal of the S1D13704
32	RD#	Connected to the RD# signal of the S1D13704
33	NC	Not connected
34	NC	Not connected

5 Host Bus Interface Pin Mapping

Table 5-1: Host Bus Interface Pin Mapping

S1D13704 Pin Names	SH-3	SH-4	MC68K #1	MC68K #2	Generic Bus #1	Generic Bus #2
AB[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]
AB0	A0	A0	LDS#	A0	A0	A0
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0]
WE1#	WE1#	WE1#	UDS#	DS#	WE1#	BHE#
CS#	CSn#	CSn#	External Decode	External Decode	External Decode	External Decode
BCLK	CKIO	CKIO	BCLK	BCLK	BCLK	BCLK
BS#	BS#	BS#	AS#	AS#	Connect to V_{SS}	Connect to IO V_{DD}
RD/WR#	RD/WR#	RD/WR#	R/W#	R/W#	RD1#	Connect to IO V_{DD}
RD#	RD#	RD#	Connect to IO V_{DD}	SIZ1	RD0#	RD#
WE0#	WE0#	WE0#	Connect to IO V_{DD}	SIZ0	WE0#	WE#
WAIT#	WAIT#	RDY#	DTACK#	DSACK1#	WAIT#	WAIT#
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	RESET#

6 Technical Description

6.1 ISA Bus Support

This board has been designed to support the 16-bit ISA bus environment and can be used in conjunction with either a VGA or a monochrome display adapter card.

There are 5 configuration inputs associated with the Host Interface (CNF[3:0] and BS#). Refer to Table 2-3: “Jumper Settings” and Table 5-1: “Host Bus Interface Pin Mapping” for complete details.

6.1.1 Display Adapter Card Support

When using the S5U13704B00B in conjunction with another primary Display Adapter (VGA or Monochrome) the following applies:

ISA or VL Bus VGA Display Adapter

When the S5U13704B00B board is used with an ISA or VL Bus VGA display adapter, the VGA card must have a 16-bit BIOS to prevent conflicts during 16-bit accesses (MEMCS16#). If an 8-bit VGA adapter card is installed in the system being used, it must be removed and the screen display routed through a COM port to a terminal display device.

PCI Bus VGA Display Adapter

All PCI based VGA display adapters can be used in conjunction with the S5U13704B00C board.

Monochrome Display Adapter

All monochrome display adapters can be used in conjunction with the S5U13704B00C board.

6.1.2 Expanded Memory Manager

If a memory manager is being used for system memory, the address range D0000h to DFFFFh must be excluded from use as this range is used by the S5U13704B00C.

6.2 Non-ISA Bus Support

The S5U13704B00C board is specifically designed to support the standard 16-bit ISA bus; however, the S1D13704 directly supports many other host bus interfaces. Header strips H1 and H2 are provided and contain all the necessary IO pins to interface to these host buses. See CPU/Bus Interface Connector Pinouts on page 11; Table 2-1: “Configuration DIP Switch Settings,” on page 8; and Table 2-3: “Jumper Settings,” on page 9 for details.

When using the header strips to provide the bus interface observe the following:

- All IO signals on the ISA bus card edge must be isolated from the ISA bus (do not plug the card into a computer). Voltage lines are provided on the header strips.
- U7, a TIBPAL16L8-15 PAL, is currently used to provide the S1D13704 CS# (pin 74), RESET# (pin 73) and other decode logic signals for ISA bus use. This functionality must now be provided externally; remove the PAL from its socket to eliminate conflicts resulting from two different outputs driving the same input. Refer to Table 5-1: “Host Bus Interface Pin Mapping” for connection details.

Note
When using a 3.3V host bus interface, IOVDD must be set to 3.3V by setting jumper (JP1) to the 2-3 position. Refer to Table 2-3: “Jumper Settings,” on page 9.

6.3 Embedded Memory Support

The S1D13704 contains 40K bytes of 16-bit SRAM used for the display buffer. The SRAM starting address is set at D0000h. Starting at this address, the board design decodes a 64K byte segment accommodating both the 40K byte display buffer and the S1D13704 internal register set.

The S1D13704 registers are mapped into the upper 32 bytes of the 64K byte segment (DFFE0h to DFFFFh).

6.4 Decode Logic

All the required decode logic is provided through a TIBPAL16L8-15 PAL (U7, socketed). This PAL contains the following equations.

```
!CS = (Address >= ^hD0000) & (Address <= ^hDFFFF) & REFRESH & !RESET;
!MEMCS16= (Address1 >= ^h0C0000) & (Address1 <= ^h0DFFFF);
RESET_  = !RESET;
```

6.5 Clock Input Support

The input clock (CLKI) frequency can be up to 50.0MHz for the S1D13704 if the internal clock divide by 2 is set. If the clock divide is not used, the maximum CLKI frequency is 25MHz.

A 25.0MHz oscillator (U2, socketed) is provided as the default clock source.

6.6 LCD Panel Voltage Setting

The S5U13704B00C board supports both 3.3V and 5.0V LCD panels through the single LCD connector J5. The voltage level is selected by setting jumper J4 to the appropriate position. Refer to Table 2-3: “Jumper Settings,” on page 9 for setting this jumper.

6.7 Monochrome LCD Panel Support

The S1D13704 directly supports 4 and 8-bit, dual and single, monochrome passive LCD panels. All necessary signals are provided on the 40-pin ribbon cable header J5. The interface signals on the cable are alternated with grounds to reduce crosstalk and noise.

Refer to Table 3-1: “LCD Signal Connector (J5) Pinout,” on page 10 for specific connection information.

6.8 Color Passive LCD Panel Support

The S1D13704 directly supports 4 and 8, dual and single, color passive LCD panels. All the necessary signals are provided on the 40-pin ribbon cable header J5. The interface signals on the cable are alternated with grounds to reduce crosstalk and noise.

Refer to Table 3-1: “LCD Signal Connector (J5) Pinout,” on page 10 for specific connection information.

6.9 Color TFT/D-TFD LCD Panel Support

The S1D13704 directly supports 9 and 12-bit active matrix color TFT/D-TFD panels. All the necessary signals can also be found on the 40-pin LCD connector J5. The interface signals on the cable are alternated with grounds to reduce crosstalk and noise.

Refer to Table 3-1: “LCD Signal Connector (J5) Pinout,” on page 10 for connection information.

6.10 Power Save Modes

The S1D13704 supports one hardware and one software power save mode. These modes are controlled by the utility 13704PWR. The hardware power save mode needs to be enabled by 13704PWR and then activated by DIP switch SW1-6. See Table 2-1: "Configuration DIP Switch Settings," on page 8 for details on setting this switch.

6.11 Adjustable LCD Panel Negative Power Supply

Most monochrome passive LCD panels require a negative power supply to provide between -18V and -23V (I_{out}=45mA). For ease of implementation, such a power supply has been provided as an integral part of this design. The VLCD power supply can be adjusted by R21 to give an output voltage from -14V to -23V, and is enabled/disabled by the S1D13704 control signal LCDPWR.

LCDPWR is an S1D13704 output signal which is configurable as active high or active low by the CNF4 signal status on the rising edge of the RESET# signal. For the proper operation of the VLCD power supply, LCDPWR must be configured as active low.

Determine the panel's specific power requirements and set the potentiometer accordingly before connecting the panel.

6.12 Adjustable LCD Panel Positive Power Supply

Most color passive LCD panels and most single monochrome 640x480 passive LCD panels require a positive power supply to provide between +23V and +40V (I_{out}=45mA). For ease of implementation, such a power supply has been provided as an integral part of this design. The V_{DDH} power supply can be adjusted by R15 to provide an output voltage from +23V to +40V and is enabled/disabled by the S1D13704 control signal LCDPWR.

LCDPWR is an S1D13704 output signal which is configurable as active high or active low by the CNF4 signal status on the rising edge of the RESET# signal. For the proper operation of the VDDH power supply, LCDPWR must be configured as active low.

Determine the panel's specific power requirements and set the potentiometer accordingly before connecting the panel.

6.13 CPU/Bus Interface Header Strips

All of the CPU/Bus interface pins of the S1D13704 are connected to the header strips H1 and H2 for easy interface to a CPU/Bus other than ISA.

Refer to Table 4-1: “CPU/BUS Connector (H1) Pinout,” on page 11 and Table 4-2: “CPU/BUS Connector (H2) Pinout,” on page 12 for specific settings.

Note

These headers only provide the CPU/Bus interface signals from the S1D13704. When another host bus interface is selected by CNF[3:0] and BS#, appropriate external decode logic MUST be used to access the S1D13704. Refer to Table 5-1: “Host Bus Interface Pin Mapping,” on page 13 for connection details.

7 Parts List

Item #	Qty/board	Designation	Part Value	Description
1	13	C1-C11, C15,C16	0.1uF, 5%, 50V	0805 ceramic capacitor
2	1	C12	1uF, 10%, 16V	Tantalum capacitor size A
3	2	C13, C14	10uF, 10%, 25V	Tantalum capacitor size D
4	2	C17, C21	47uF, 10%, 16V	Tantalum capacitor size D
5	3	C18-C20	4.7uF, 10%, 50V	Tantalum capacitor size D
6	1	C22	56uF, 20%, 63V	Electrolytic, radial, low ESR
7	2	H1,H2	CON34A Header	0.1” 17x2 header, PTH
8	4	JP1-JP4	HEADER 3	0.1” 1x3 header, PTH
9	1	J1	AT CON-A	ISA Bus gold-fingers
10	1	J2	AT CON-B	ISA Bus gold-fingers
11	1	J3	AT CON-C	ISA Bus gold-fingers
12	1	J4	AT CON-D	ISA Bus gold-fingers
13	1	J5	CON40A	Shrouded header 2x20, PTH, center key
14	1	L1	1µH	MCI-1812 inductor
15	3	L2-L4	Ferrite bead	Philips BDS3/3/8.9-4S2
16	1	Q1	2N3906	PNP signal transistor, SOT23
17	1	Q2	2N3903	NPN signal transistor, SOT23
18	6	R1-R6	15K, 5%	0805 resistor
19	7	R7-R13	10K, 5%	0805 resistor
20	1	R14	470K, 5%	0805 resistor
21	1	R15	200K	200K Trim POT Spectrol 63S204T607 (or equivalent)
22	1	R16	14K, 1%	0805 resistor
23	2	R17, R18	1K, 5%	0805 resistor
24	2	R19, R20	100K, 5%	0805 resistor
25	1	R21	100K	100K Trim POT Spectrol 63S104T607 (or equivalent)
26	1	S1	SW DIP-6	6 position DIP switch
27	1	U1	S1D13704F00A	QFP14-80, 80 pin, SMT
28	1	U2	25.0 MHz oscillator	FOX 25MHz oscillator or equiv., 14 pin DIP socketed
29	3	U3-U5	74AHC244	SO-22, TI74AHC244
30	1	U6	LT1117CM-3.3	Linear Technology 5V to 3.3V regulator, 800mA
31	1	U7	TIBPAL16L8-15	TI PAL, 20 Pin DIP, socketed.
32	1	U8	74ALS125	SO-20, TI74ALS125
33	1	U9	RD-0412	Xentek RD-0412, positive PS
34	1	U10	EPN001	Xentek EPN001 negative PS

8 Schematic Diagrams

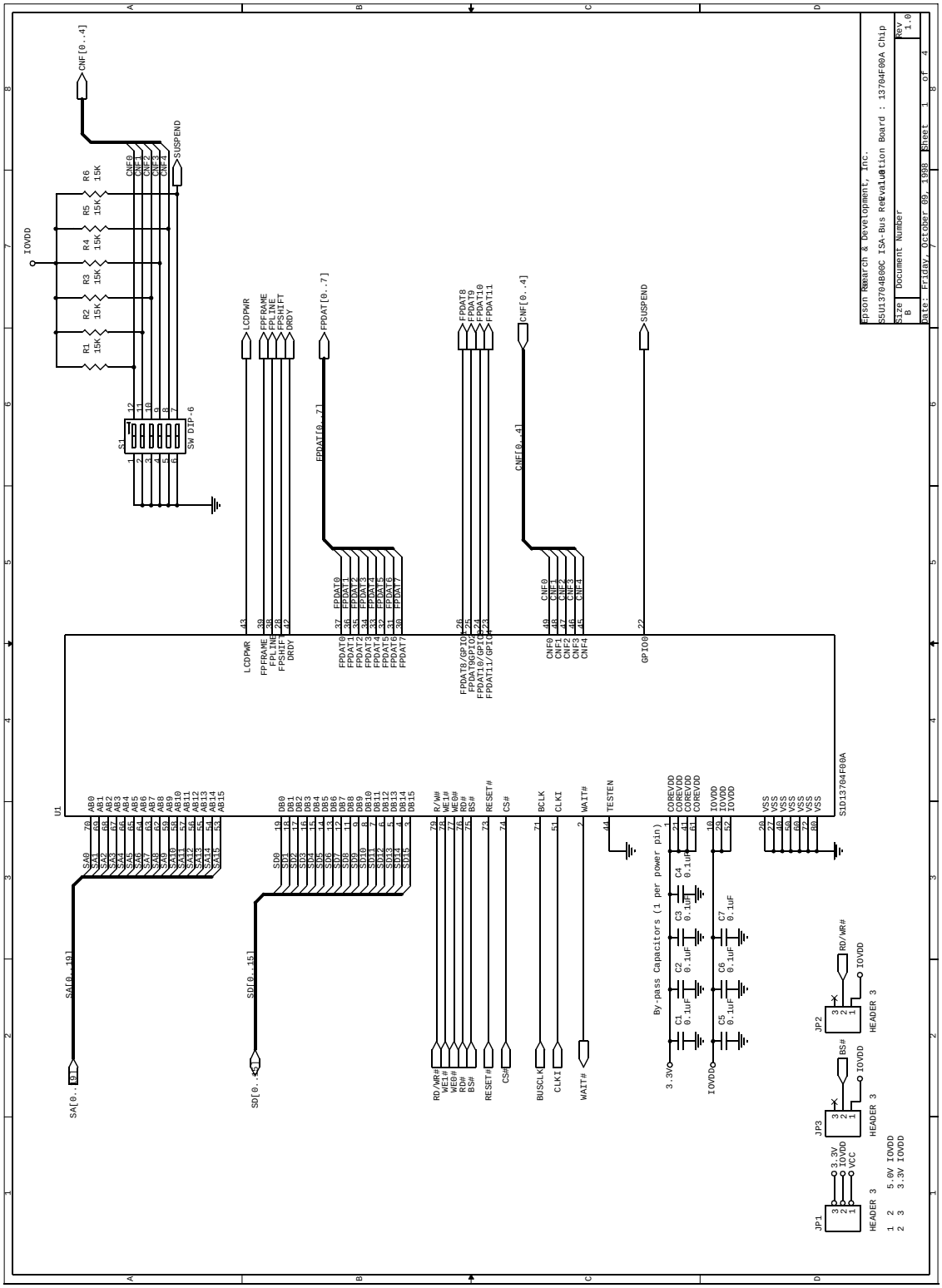


Figure 8-1: S1D13704B00C Schematic Diagram (1 of 4)

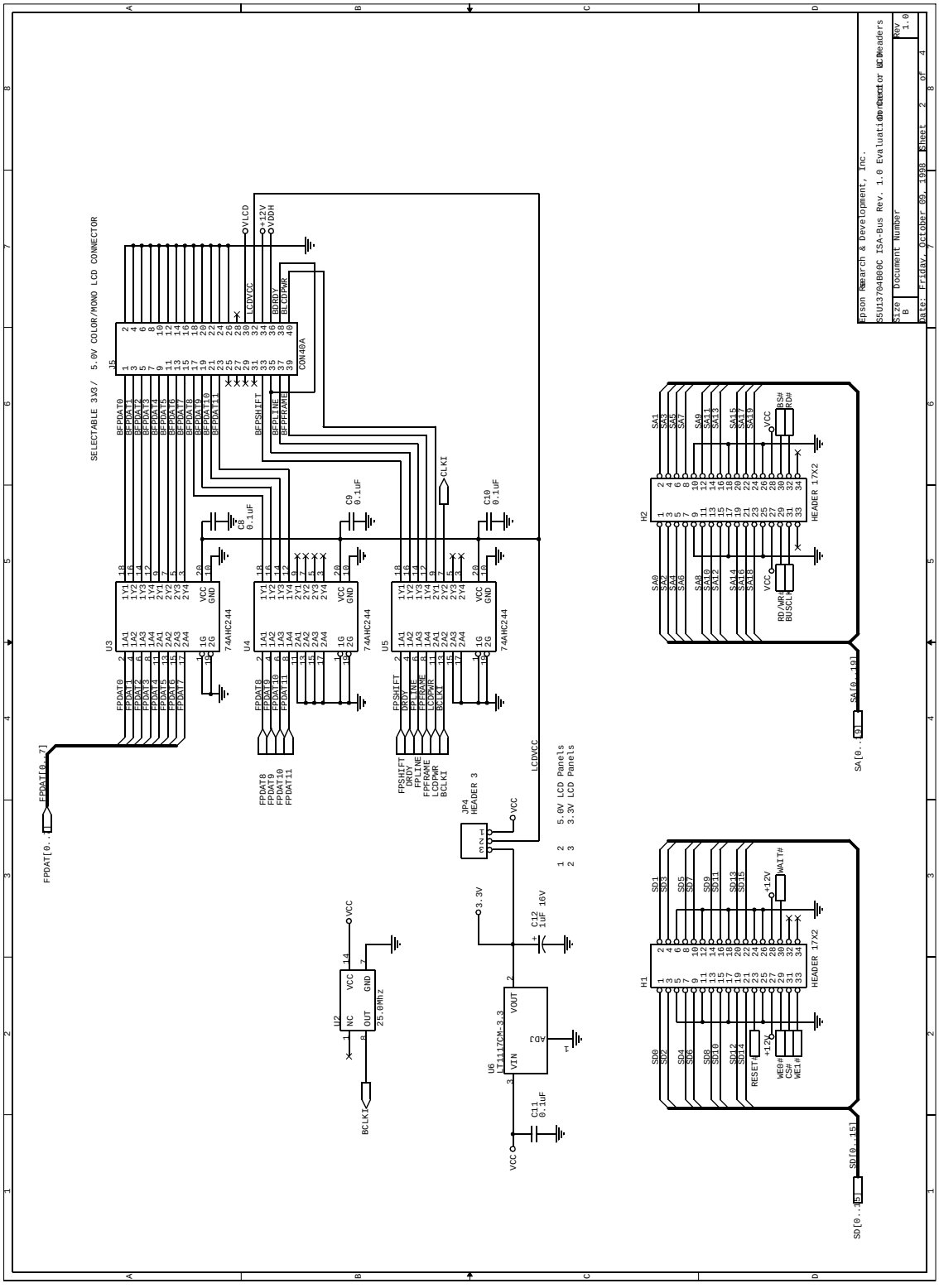


Figure 8-2: S1D13704B00C Schematic Diagram (2 of 4)

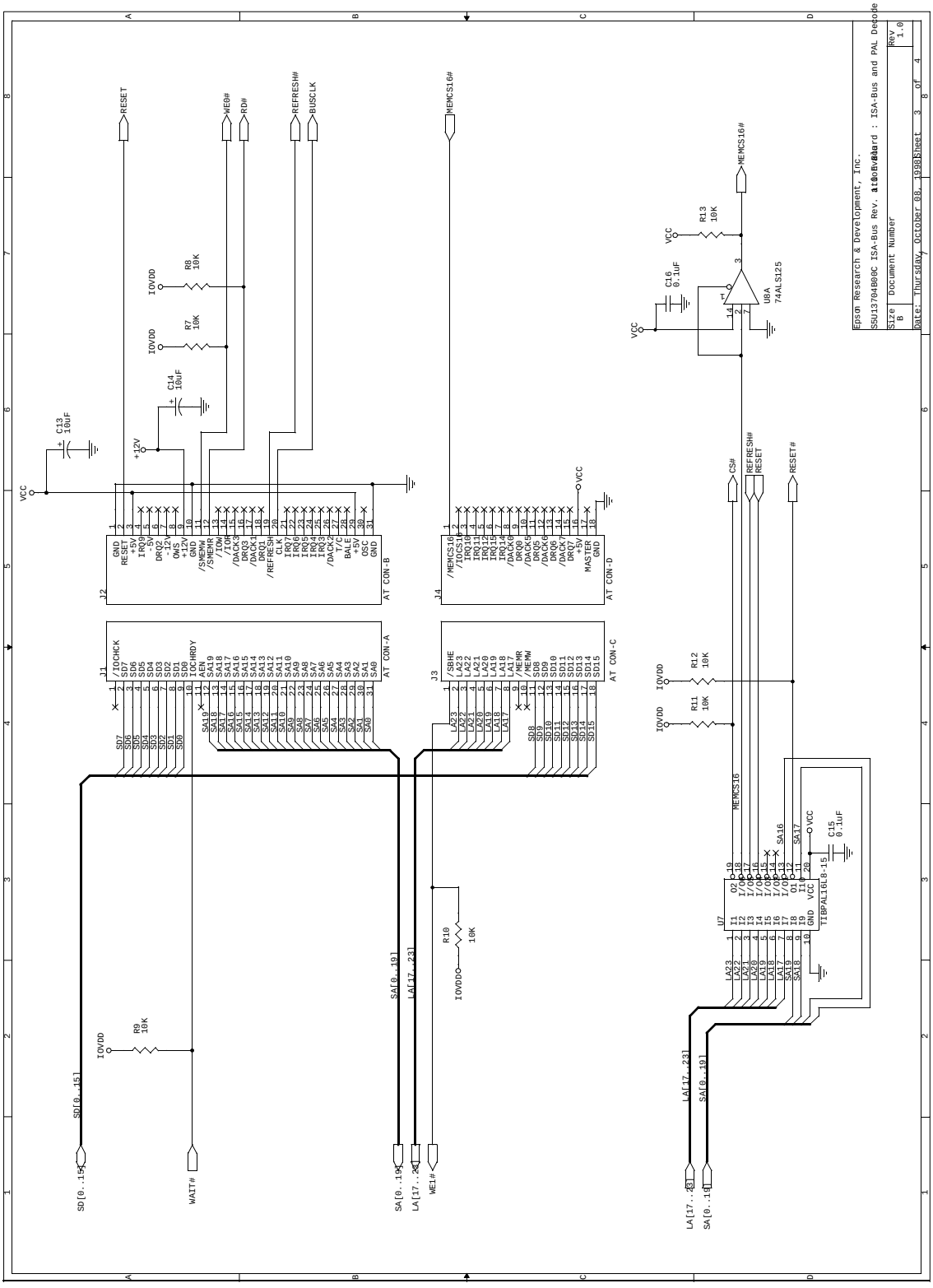


Figure 8-3: S1D13704B00C Schematic Diagram (3 of 4)

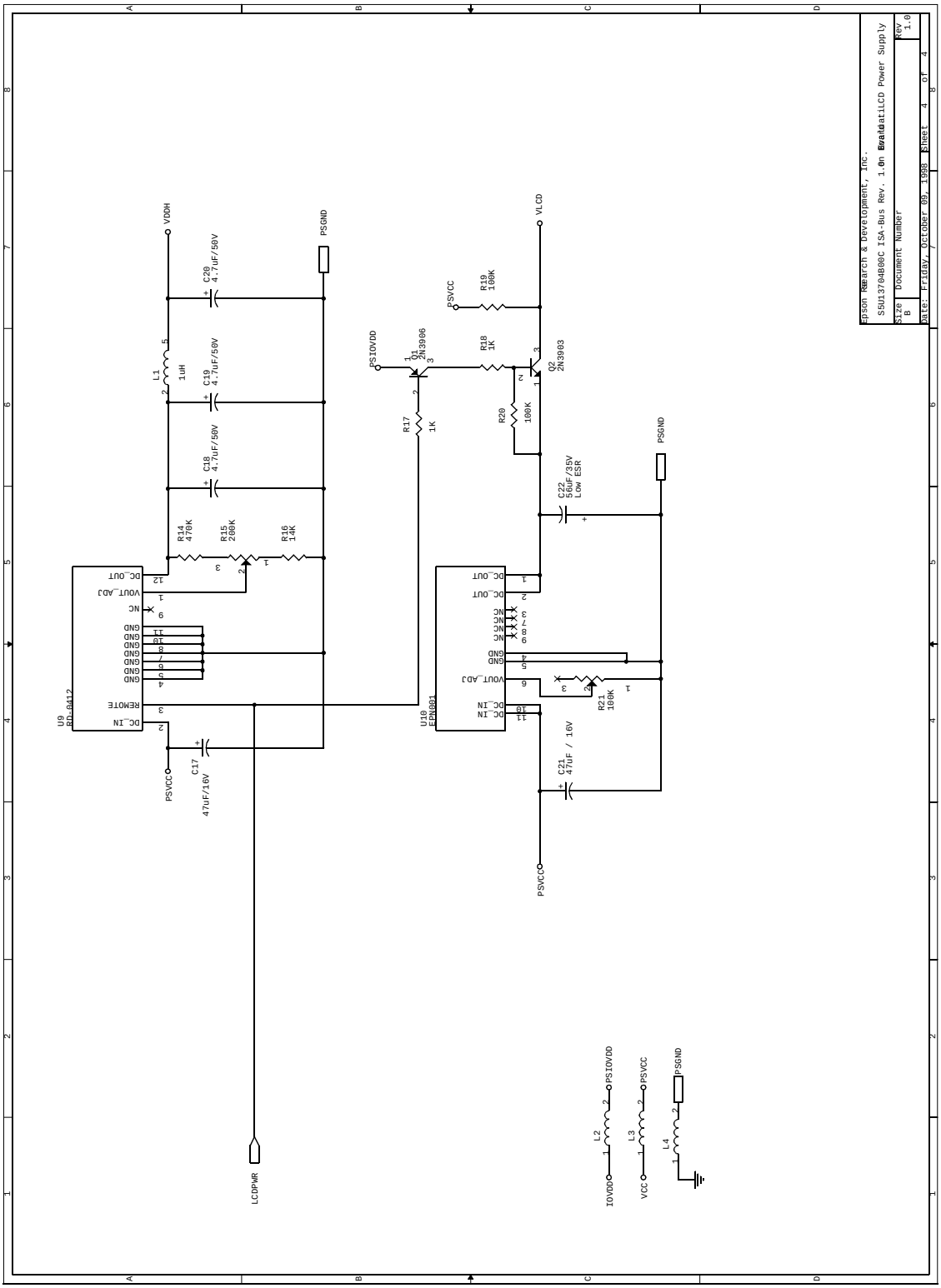


Figure 8-4: S1D13704B00C Schematic Diagram (4 of 4)

THIS PAGE LEFT BLANK

EPSON®

S1D13704 Embedded Memory Color LCD Controller

Interfacing to the Toshiba MIPS TX3912 Processor

Document Number: X26A-G-004-02

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 S1D13704 Embedded Memory Color Graphics LCD Controller and the Toshiba MIPS TX3912 Processor.

For further information on the S1D13704, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

For further information on the TX3912, contact Toshiba or refer to the Toshiba website under semiconductors at http://www.toshiba.com/taec/nonflash/indexproducts.html.

For further information on the ITE IT8368E, refer to the *IT8368E PC Card / GPIO Buffer Chip Specification*.

1.1 General Description

The Toshiba MIPS TX3912 processor supports up to two PC Card (PCMCIA) slots. It is through this host bus interface that the S1D13704 connects to the TX3912 processor.

The S1D13704 can be successfully interfaced using one of two configurations:

- Direct connection to TX3912 (see Section 2, *"Direct Connection to the Toshiba TX3912"* on page 8).
- System design using one ITE IT8368E PC Card/GPIO buffer chip (see Section 3, *"System Design Using the ITE IT8368E PC Card Buffer"* on page 10).

2 Direct Connection to the Toshiba TX3912

2.1 General Description

In this example implementation, the S1D13704 occupies the TX3912 PC Card slot #1.

The S1D13704 is easily interfaced to the TX3912 with minimal additional logic. The address bus of the TX3912 PC Card interface is multiplexed and can be demultiplexed using an advanced CMOS latch (e.g., 74ACT373). The direct connection approach makes use of the S1D13704 in its "Generic Interface #2" configuration.

The following diagram demonstrates a typical implementation of the interface.

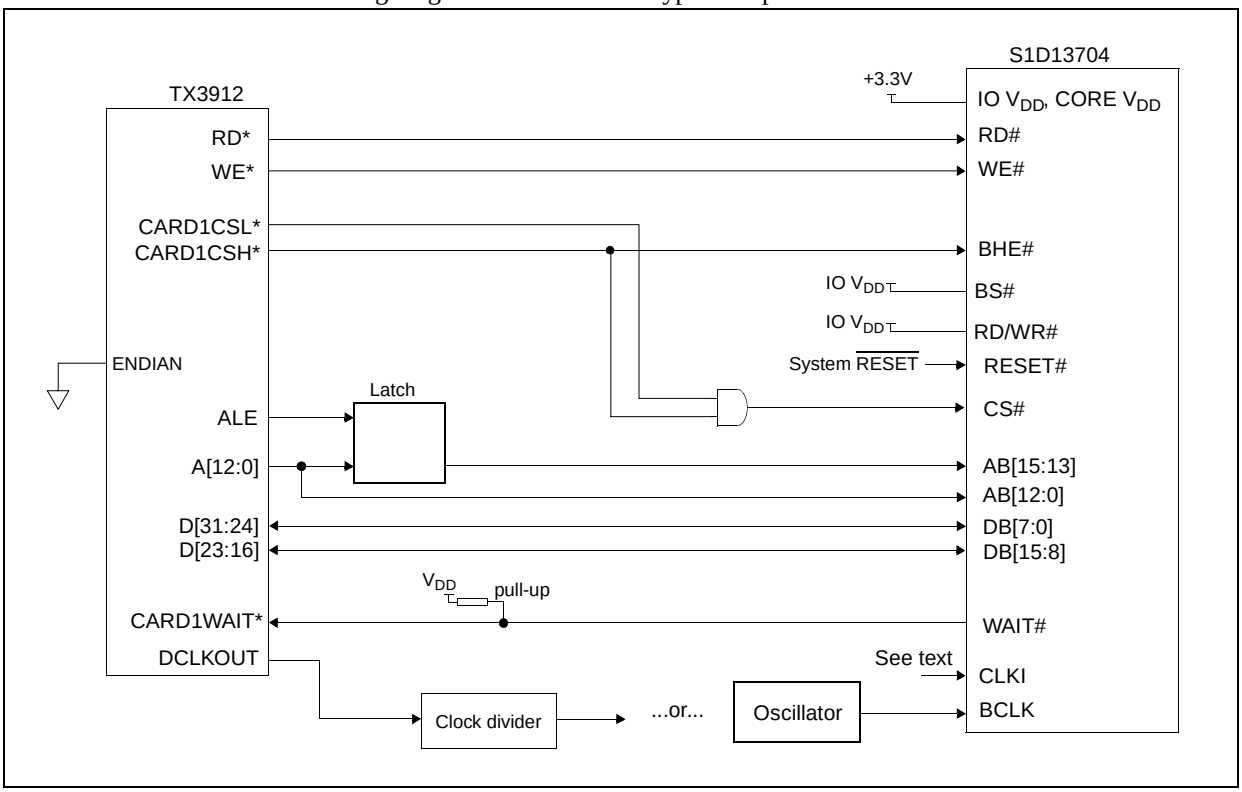


Figure 2-1: S1D13704 to TX3912 Direct Connection

The "Generic #2" host interface control signals of the S1D13704 are asynchronous with respect to the S1D13704 bus clock. This gives the system designer full flexibility to choose the appropriate source (or sources) for CLKI and BCLK. 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 S1D13704 clock frequencies.

The S1D13704 also has internal clock dividers providing additional flexibility.

2.2 Memory Mapping and Aliasing

The S1D13704 requires an addressing space of 64K bytes. The on-chip display memory occupies the range 0 through 9FFFh. The registers occupy the range FFE0h through FFFFh. The TX3912 demultiplexed address lines A16 and above are ignored, thus the S1D13704 is aliased 1024 times at 64K byte intervals over the 64M byte PC Card slot #1 memory space. In this example implementation, the TX3912 control signal CARDREG* is ignored, the S1D13704 also takes up the entire PC Card slot 1 configuration space.

Note
If aliasing is undesirable, additional decoding circuitry must be added.

2.3 S1D13704 Configuration and Pin Mapping

The S1D13704 is configured at power up by latching the state of the CNF[4:0] pins. Pin BS# also plays a role in host bus interface configuration. For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

The table below shows those configuration settings relevant to the direct connection approach.

Table 2-1: S1D13704 Configuration for Direct Connection

S1D13704 Configuration Pin	Value hard wired on this pin is used to configure:	
	1 (IO V_{DD})	**0 (V_{SS})**
BS#	Generic #2	Generic #1
CNF3	Big Endian	Little Endian
CNF[2:0]	111: Generic #1 or #2	

= configuration for Toshiba TX3912 host bus interface

When the S1D13704 is configured for “Generic #2” interface, the host interface pins are mapped as in the table below.

Table 2-2: S1D13704 Generic #2 Interface Pin Mapping

Pin Name	Pin Function
WE1#	BHE#
BS#	Connect to IO V_{DD}
RD/WR#	Connect to IO V_{DD}
RD#	RD#
WE0#	WE#

3 System Design Using the ITE IT8368E PC Card Buffer

If the system designer uses the ITE IT8368E PC Card and multiple-function I/O buffer, the S1D13704 can be interfaced so that it 'shares' a PC Card slot. The S1D13704 is mapped to a rarely-used 16M byte portion of the PC Card slot buffered by the IT8368E. This makes the S1D13704 virtually transparent to PC Card devices that use the same slot.

3.1 Hardware Description

The ITE8368E has been specially designed to support EPSON LCD controllers. The ITE IT8368E provides eleven Multi-Function IO pins (MFIO). Configuration registers may be used to allow these MFIO pins to provide the control signals required to implement the S1D13704 CPU interface.

The TX3912 processor only provides addresses A[12:0], therefore devices requiring more address space must use an external device to latch A[25:13]. The IT8368E's MFIO pins can be configured to provide this latched address.

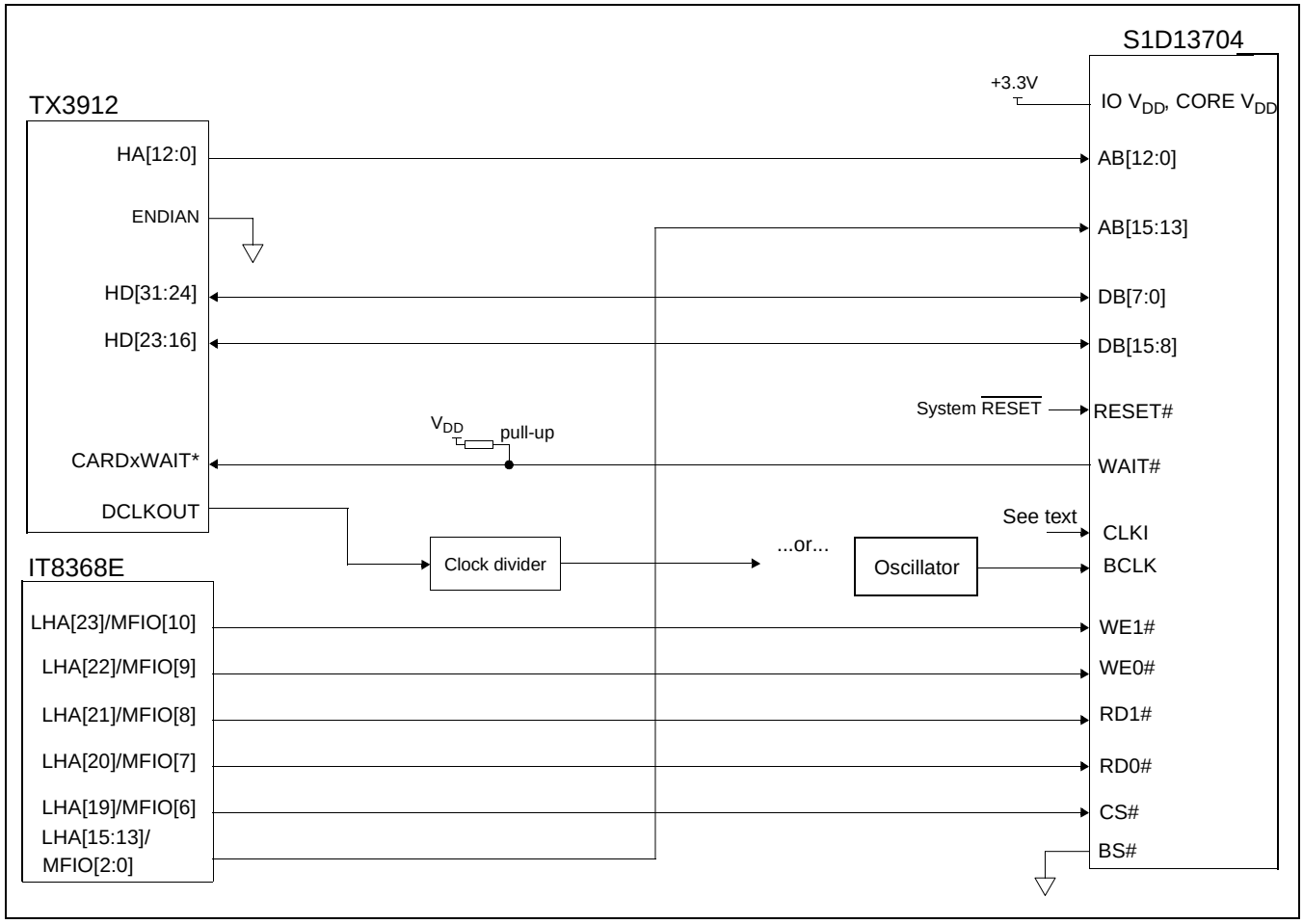


Figure 3-1: S1D13704 to TX3912 Connection Using an IT8368E

The “Generic #1” host interface control signals of the S1D13704 are asynchronous with respect to the S1D13704 bus clock. This gives the system designer full flexibility to choose the appropriate source (or sources) for CLKI and BCLK. The choice of whether both clocks should be the same, and whether to use DCLKOUT (divided) as clock source, should be based on pixel and frame rates, power budget, part count and maximum S1D13704 respective clock frequencies. Also, internal S1D13704 clock dividers provide additional flexibility.

3.2 IT8368E Configuration

The IT8368E provides eleven multi-function IO pins (MFIO). The IT8368E must have both “Fix Attribute/IO” and “VGA” modes on. When both these modes are enabled, the MFIO pins provide control signals needed by the S1D13704 host bus interface, and a 16M byte portion of the system PC Card attribute and IO space is allocated to address the S1D13704. When accessing the S1D13704 the associated card-side signals are disabled in order to avoid any conflicts.

For mapping details, refer to section 3.3: “Memory Mapping and Aliasing.” For connection details see Figure 3-1: “S1D13704 to TX3912 Connection Using an IT8368E,” on page 10. For further information on the IT8368E, refer to the *IT8368E PC Card/GPIO Buffer Chip Specification.*

Note

When a second IT8368E is used, that circuit should not be set in VGA mode.

3.3 Memory Mapping and Aliasing

When the TX3912 accesses the PC Card slots *without* the ITE IT8368E, its system memory is mapped as in Table 3-1:, “TX3912 to Unbuffered PC Card Slots System Address Mapping”.

Note

Bits CARD1IOEN and CARD2IOEN need to be set in TX3912 Memory Configuration Register 3.

Table 3-1: TX3912 to Unbuffered PC Card Slots System Address Mapping

TX3912 Address	Size	Function (CARDnIOEN=0)	Function (CARDnIOEN=1)
0800 0000h	64M byte	Card 1 Attribute	Card 1 IO
0C00 0000h	64M byte	Card 2 Attribute	Card 2 IO
6400 0000h	64M byte	Card 1 Memory	
6400 0000h	64M byte	Card 2 Memory	

When the TX3912 accesses the PC Card slots buffered through the ITE IT8368E, bits CARD1IOEN and CARD2IOEN are ignored and the attribute/IO space of the TX3912 is divided into Attribute, I/O and S1D13704 access. Table 3-2:, “TX3912 to PC Card Slots Address Remapping Using the IT8368E” provides all details of the Attribute/IO address reallocation by the IT8368E.

Table 3-2: TX3912 to PC Card Slots Address Remapping Using the IT8368E

IT8368E Uses PC Card Slot #	TX3912 Address	Size	Function
1	0800 0000h	16M byte	Card 1 IO
	0900 0000h	16M byte	S1D13704 (aliased 256 times at 64K byte intervals)
	0A00 0000h	32M byte	Card 1 Attribute
	6400 0000h	64M byte	Card 1 Memory
2	0C00 0000h	16M byte	Card 2 IO
	0D00 0000h	16M byte	S1D13704 (aliased 256 times at 64K byte intervals)
	0E00 0000h	32M byte	Card 2 Attribute
	6800 0000h	64M byte	Card 2 Memory

3.4 S1D13704 Configuration

The S1D13704 is configured at power up by latching the state of the CNF[4:0] pins. Pin BS# also plays a role in host bus interface configuration. For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

The table below shows those configuration settings relevant to this specific interface.

Table 3-3: S1D13704 Configuration Using the IT8368E

S1D13704 Configuration Pin	Value hard wired on this pin is used to configure:	
	1 (IO V_{DD})	**0 (V_{SS})**
BS#	Generic #2	Generic #1
CNF3	Big Endian	Little Endian
CNF[2:0]	111: Generic #1 or #2	

= configuration for connection using ITE IT8368E

When the S1D13704 is configured for “Generic #1” interface, the host interface pins are mapped as in the table below.

Table 3-4: S1D13704 Generic #1 Interface Pin Mapping

Pin Name	Pin Function
WE1#	WE1#
BS#	connect to V_{SS}
RD/WR#	RD1#
RD#	RD0#
WE0#	WE0#

4 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13704. 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 1357CFG, or by directly modifying the source. The Windows CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13704 test utilities and Windows CE v2.0 display drivers are available from your sales support contact or www.erd.epson.com.

5 Technical Support

5.1 EPSON LCD Controllers (S1D13704)

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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

5.2 Toshiba MIPS TX3912 Processor

http://www.toshiba.com/taec/nonflash/indexproducts.html

5.3 ITE IT8368E

Integrated Technology Express, Inc.
Sales & Marketing Division
2710 Walsh Avenue
Santa Clara, CA 95051, USA
Tel: (408) 980-8168
Fax: (408) 980-9232
http://www.iteusa.com

THIS PAGE LEFT BLANK

EPSON®

S1D13704 Embedded Memory Color LCD Controller

S1D13704
Power Consumption

Document Number: X26A-G-006-02

THIS PAGE LEFT BLANK

1 S1D13704 Power Consumption

S1D13704 power consumption is affected by many system design variables.

- Input clock frequency (CLKI): the CLKI frequency and the internal clock divide register determine the operating clock (CLK) frequency of the S1D13704. The higher CLK is, the higher the frame fate, performance, and power consumption.
- CPU interface: the S1D13704 current consumption depends on the BUSCLK frequency, data width, number of toggling pins, and other factors – the higher the BUSCLK, the higher the CPU performance and power consumption.
- V_{DD} voltage levels (Core and IO): the voltage level of the Core and IO sections in the S1D13704 affects power consumption – the higher the voltage, the higher the consumption.
- Display mode: the resolution, panel type, and color depth affect power consumption. The higher the resolution/color depth and number of LCD panel signals, the higher the power consumption.

Note

If the High Performance option is turned on, the power consumption increases to that of 8 bit-per-pixel mode for all color depths.

There are two power save modes in the S1D13704: Software and Hardware Power Save. The power consumption of these modes is affected by various system design variables.

- CPU bus state during Power Save: the state of the CPU bus signals during Power Save has a substantial effect on power consumption. An inactive bus (e.g. BUSCLK = low, Addr = low etc.) reduces overall system power consumption.
- CLKI state during Power Save: disabling the CLKI during Power Save has substantial power savings.

1.1 Conditions

Table 1-1: "S1D13704 Total Power Consumption" below gives an example of a specific environment and its effects on power consumption.

Table 1-1: S1D13704 Total Power Consumption

	Test Condition *Core V_{DD} = 3.3V, IO V_{DD} = 3.3V* BUSCLK = 8.33MHz	**Gray Shades / Colors**	**Power Consumption**				
			Active			**Power Save Mode**	
			Core	**IO**	**Total**	**Software**	**Hardware**
1	Input Clock = 6MHz LCD Panel = 320x240 4-bit Single Monochrome	Black-and-White 4 Gray Shades 16 Gray Shades	5.29mW 6.86mW 8.15mW	0.3mW 0.43mW 0.55mW	5.59mW 7.29mW 8.70mW	1.58mW[1]	1.19mW[2]
2	Input Clock = 6MHz LCD Panel = 320x240 4-bit Single Color	2 Colors 4 Colors 16 Colors	6.82mW 7.58mW 8.98mW	1.13mW 2.29mW 2.25mW	7.95mW 9.86mW 11.23mW	1.58mW[1]	1.19mW[2]
3	Input Clock = 25MHz LCD Panel = 640x480 8-bit Single Monochrome	Black-and-White	21.38mW	0.92mW	22.30mW	3.09mW[1]	2.71mW[2]
4	Input Clock = 25MHz LCD Panel = 640x480 8-bit Single Color	2 Colors	23.66mW	2.40mW	26.07mW	3.09mW[1]	2.71mW[2]
5	Input Clock = 25MHz LCD Panel = 640x480 8-bit Dual Monochrome	Black-and-White	20.93mW	0.88mW	21.81mW	3.09mW[1]	2.71mW[2]
6	Input Clock = 25MHz LCD Panel = 640x480 8-bit Dual Color	2 Colors	23.78mW	1.93mW	25.72mW	3.09mW[1]	2.71mW[2]
7	Input Clock = 25MHz LCD Panel = 640x480 9-bit TFT	2 Colors	16.48mW	8.07mW	24.55mW	3.09mW[1]	2.71mW[2]

Note

1. Conditions for Software Power Save:
   - CPU interface active (signals toggling)
   - CLKI active

2. Conditions for Hardware Power Save:
   - CPU interface inactive (high impedance)
   - CLKI active

2 Summary

The system design variables in Section 1, “S1D13704 Power Consumption” and in Table 1-1: “S1D13704 Total Power Consumption” show that S1D13704 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 S1D13704 can be configured to be an extremely power-efficient LCD Controller with high performance and flexibility.

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13704 Embedded Memory Color LCD Controller

Interfacing to the Motorola MC68328 ‘Dragonball’ Microprocessor

Document Number: X26A-G-007-03

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 required to provide an interface between the S1D13704 Embedded Memory LCD Controller and the Motorola MC68328 “Dragonball” Microprocessor. By implementing a dedicated display refresh memory, the S1D13704 can reduce system power consumption, improve image quality, and increase system performance as compared to the Dragonball’s on-chip LCD controller.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note will be updated as appropriate. Please check the Epson Research and Development Website at http://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 techpubs@erd.epson.com.

2 Interfacing to the MC68328

2.1 The MC68328 System Bus

The MC68328 is an integrated controller for handheld products, based upon the MC68EC000 microprocessor core. It implements a 16-bit data bus and a 32-bit address bus. The bus interface consists of all the standard MC68000 bus interface signals, plus some new signals intended to simplify the task of interfacing to typical memory and peripheral devices.

The MC68000 bus control signals are well documented in Motorola's user manuals, and will not be described here. A brief summary of the new signals appears below:

- Output Enable ($\overline{OE}$) is asserted when a read cycle is in process; it is intended to connect to the output enable control of a typical static RAM, EPROM, or Flash EPROM device.
- Upper Write Enable and Lower Write Enable ($\overline{UWE/LWE}$) 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.

The S1D13704 implements the MC68000 bus interface using its "MC68K #1" mode, so this mode may be used to connect the MC68328 directly to the S1D13704 with no glue logic. However, several of the MC68000 bus control signals are multiplexed with IO and interrupt signals on the MC68328, and in many applications it may be desirable to make these pins available for these alternate functions. This requirement may be accommodated through use of the Generic #1 interface mode on the S1D13704.

2.2 Chip-Select Module

The MC68328 can generate up to 16 chip select outputs, organized into four groups "A" through "D".

Each chip select group has a common base address register and address mask register, to set the base address and block size of the entire group. 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. Finally, each chip select may be individually programmed to control an 8 or 16-bit device, and each may be individually programmed to generate from 0 through 6 wait states internally, or allow the memory or peripheral device to terminate the cycle externally through use of the standard MC68000 $\overline{DTACK}$ signal.

Groups A and B can have a minimum block size of 64K bytes, so these are typically used to control memory devices. Chip select A0 is active immediately after reset, so it is typically used to control a boot EPROM device. Groups C and D have a minimum block size of 4K bytes, so they are well-suited to controlling peripheral devices. Chip select D3 is associated with the MC68328 on-chip PCMCIA control logic.

3 S1D13704 Host Bus Interface

This section is a summary of the host bus interface modes available on the S1D13704 and offers some detail on the Generic #1 and MC68K #1 host bus interfaces that may be used to implement the interface to the MC68328.

3.1 Bus Interface Modes

The S1D13704 implements a 16-bit interface to the host microprocessor which may operate in one of several modes compatible with most of the popular embedded microprocessor families. Six bus interface modes are supported:

- Hitachi SH-4.
- Hitachi SH-3
- Motorola MC68000 (using Upper Data Strobe/Lower Data Strobe).
- Motorola MC68020/MC68030/MC683xx (using Data Strobe/DSACKx).
- Generic #1 (Chip Select, plus individual Read Enable/Write Enable for each byte).
- Generic #2 (External Chip Select, shared Read/Write Enable for high byte, individual Read/Write Enable for low byte).

The S1D13704 latches CNF[2:0] and BS# to allow selection of the host bus interface on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. The following table shows the functions of each host bus interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13704 Pin Names	SH-3	SH-4	MC68K #1	MC68K #2	Generic #1	Generic #2
AB[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]
AB0	A0	A0	LDS#	A0	A0	A0
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[31:16]	D[15:0]	D[15:0]
WE1#	WE1#	WE1#	UDS#	DS#	WE1#	BHE#
CS#	CSn#	CSn#	External Decode	External Decode	External Decode	External Decode
BCLK	CKIO	CKIO	CLK	CLK	BCLK	BCLK
BS#	BS#	BS#	AS#	AS#	connect to V_{SS}	connect to IO V_{DD}
RD/WR#	RD/WR#	RD/WR#	R/W#	R/W#	RD1#	connect to IO V_{DD}
RD#	RD#	RD#	connect to IO V_{DD}	SIZ1	RD0#	RD#
WE0#	WE0#	WE0#	connect to IO V_{DD}	SIZ0	WE0#	WE#
WAIT#	WAIT#	RDY#	DTACK#	DSACK1#	WAIT#	WAIT#
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	RESET#

Two other configuration options (CNF[4:3]) are also made at time of hardware reset:

- endian mode setting (big endian or little endian).
- polarity of the LCDPWR signal.

The capability to select the endian mode independent of the host bus interface offers more flexibility in configuring the S1D13704 with other CPUs.

For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

3.2 Generic #1 Interface Mode

Generic #1 interface mode is the most general and least processor-specific interface mode on the S1D13704. The Generic # 1 interface mode was chosen for this interface due to the simplicity of its timing.

The interface requires the following signals:

- BUSCLK is a clock input which is required by the S1D13704 host interface. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB0 through AB15, and the data bus DB0 through DB15, connect directly to the CPU address and data bus, respectively. On 32-bit big endian architectures such as the Power PC, the data bus would connect to the high-order data lines; on little endian hosts, or 16-bit big endian hosts, they would connect to the low-order data lines. The hardware engineer must ensure that CNF3 selects the proper endian mode upon reset.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper IO or memory address space.
- WE0# and WE1# are write enables for the low-order and high-order bytes, respectively, to be driven low when the host CPU is writing data to the S1D13704. These signals must be generated by external hardware based on the control outputs from the host CPU.
- RD# and RD/WR# are read enables for the low-order and high-order bytes, respectively, to be driven low when the host CPU is reading data from the S1D13704. These signals must be generated by external hardware based on the control outputs from the host CPU.
- WAIT# is a signal output from the S1D13704 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 S1D13704 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13704 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. However, BS# is used to configure the S1D13704 for Generic #1 mode and should be tied low (connected to GND).

3.3 MC68K #1 Interface Mode

The MC68K #1 Interface Mode can be used to interface to the MC68328 microprocessor if the previously mentioned, multiplexed, bus signals will not be used for other purposes.

The interface requires the following signals:

- BUSCLK is a clock input which synchronizes transfers between the host CPU and the S1D13704. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB1 through AB15, and the data bus DB0 through DB15, connect directly to the CPU address and data bus, respectively. On 32-bit big endian architectures such as the Power PC, the data bus would connect to the high-order data lines; on little endian hosts, or 16-bit big endian hosts, they would connect to the low-order data lines. The hardware engineer must ensure that CNF3 selects the proper endian mode upon reset.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper IO or memory address space.
- A0 and WE1# are the enables for the low-order and high-order bytes, respectively, to be driven low when the host CPU is reading or writing data to the S1D13704. These must be generated by external decode hardware based upon the control outputs from the host CPU.
- RD/WR# is the read/write signal that is driven low when the CPU writes to the S1D13704 and is driven high when the CPU is doing a read from the S1D13704. This signal must be generated by external decode hardware based upon the control output from the host CPU.
- WAIT# is a signal which is output from the S1D13704 to the host CPU that indicates when data is ready (read cycle) or accepted (write cycle) on the host bus. Since host CPU accesses to the S1D13704 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13704 internal registers and/or refresh memory. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Status (BS#) signal indicates that the address on the address bus is valid. This signal must be generated by external decode hardware based upon the control outputs from the host CPU.
- The WE0# signal is not used in the bus interface for MC68K #1 and must be tied high (tied to IO V_{DD}).

4 MC68328 To S1D13704 Interface

4.1 Hardware Description

The interface between the MC68328 and the S1D13704 can be implemented using either the MC68K #1 or Generic #1 host bus interface of the S1D13704.

4.1.1 Using The MC68K #1 Host Bus Interface

The MC68328 multiplexes dual functions on some of its bus control pins (specifically UDS, LDS, and DTACK). In implementations where all of these pins are available for use as bus control pins, then the S1D13704 interface is a straightforward implementation of the "MC68K #1" host bus interface. For further information on this host bus interface, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

The following diagram shows a typical implementation of the MC68328 to S1D13704 using the MC68K #1 host bus interface.

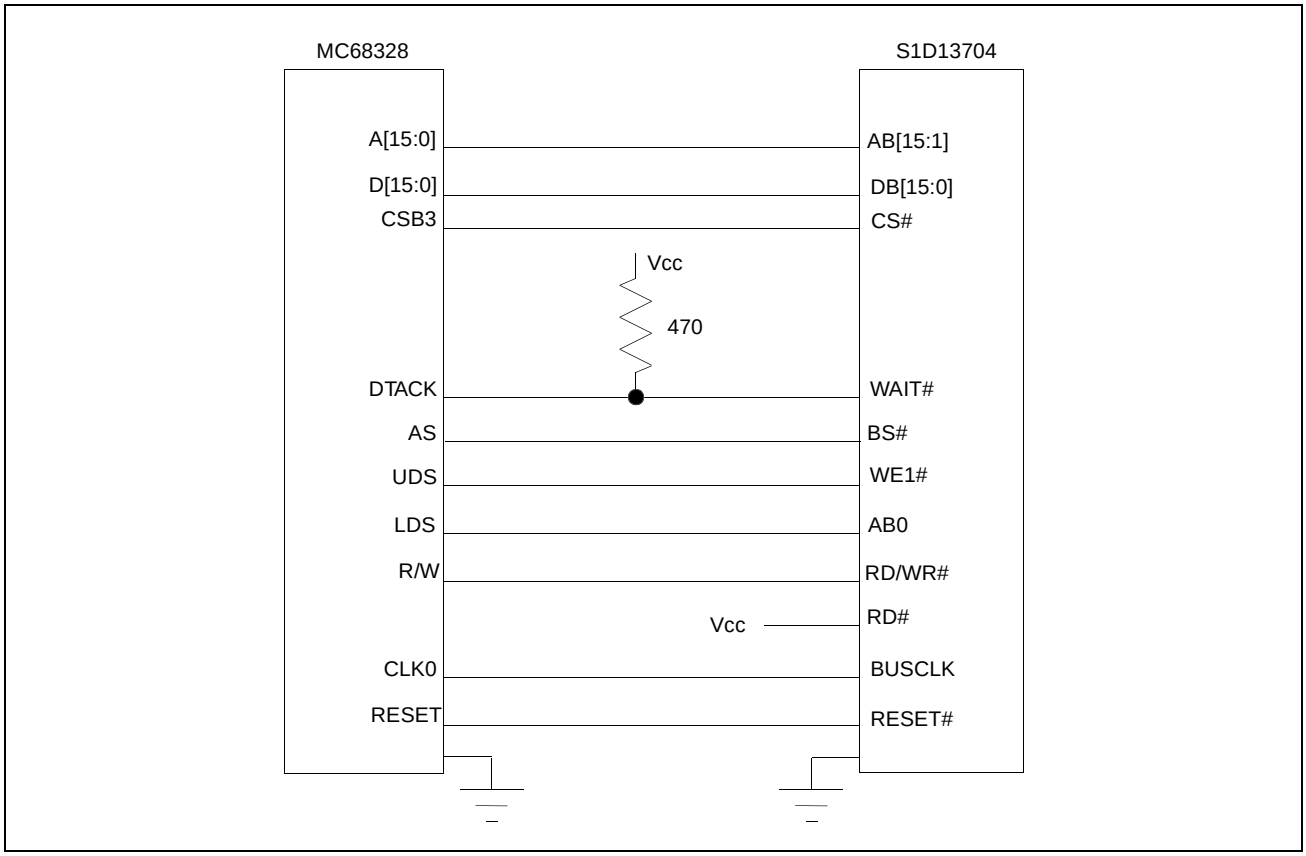


Figure 4-1: Typical Implementation of MC68328 to S1D13704 Interface - MC68K #1

4.1.2 Using The Generic #1 Host Bus Interface

If UDS and/or LDS are required for their alternate IO functions, then the MC68328 to S1D13704 interface may be implemented using the S1D13704 Generic #1 host bus interface. Note that in either case, the DTACK signal must be made available for the S1D13704, since it inserts a variable number of wait states depending upon CPU/LCD synchronization and the LCD panel display mode. WAIT# must be inverted (using an inverter enabled by CS#) to make it an active high signal and thus compatible with the MC68328 architecture. 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 MC68328 to S1D13704 using the Generic #1 host bus interface.

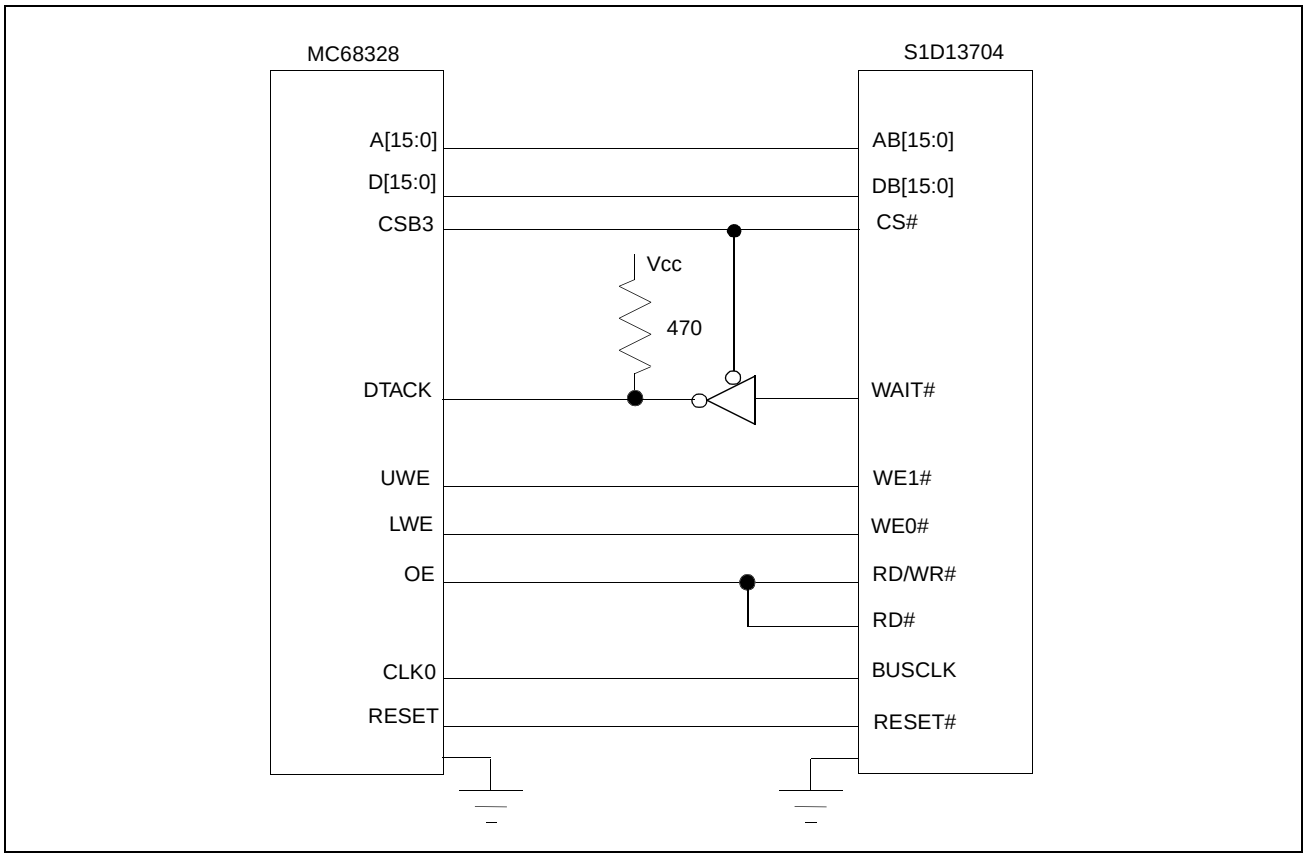


Figure 4-2: Typical Implementation of MC68328 to S1D13704 Interface - Generic #1

4.2 S1D13704 Hardware Configuration

The S1D13704 uses CNF4 through CNF0 and BS# to allow selection of the bus mode and other configuration data on the rising edge of RESET#. Refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx for details.

The tables below show those configuration settings important to the MC68K #1 and Generic #1 host bus interfaces.

Table 4-1: Summary of Power-On/Reset Options

S1D13704 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	0	1
CNF0	See "Host Bus Selection" table below	See "Host Bus Selection" table below
CNF1		
CNF2		
CNF3	Little Endian	Big Endian
CNF4	Active low LCDPWR signal	Active high LCDPWR signal

= configuration for MC68328 support

Table 4-2: Host Bus Interface Selection

CNF2	CNF1	CNF0	BS#	Host Bus Interface
0	0	0	X	SH-4 interface
0	0	1	X	SH-3 interface
0	1	0	X	reserved
0	1	1	X	MC68K #1, 16-bit
1	0	0	X	reserved
1	0	1	X	MC68K #2, 16-bit
1	1	0	0	reserved
1	1	0	1	reserved
1	1	1	0	Generic #1, 16-bit
1	1	1	1	Generic #2, 16-bit

= configuration for MC68328 using Generic #1 host bus interface
= configuration for MC68328 using MC68K #1 host bus interface

4.3 MC68328 Chip Select Configuration

The S1D13704 requires a 64K byte address space 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. Virtually any chip select other than CSA0 or CSD3 would be suitable for the S1D13704 interface.

In the example interface, chip select CSB3 is used to control the S1D13704. A 64K byte address space is used with the S1D13704 control registers mapped into the top 32 bytes of the 64K byte block and the 40K bytes of display buffer mapped to the starting address of the block. The chip select should have its RO (Read Only) bit set to 0, and the WAIT field (Wait states) should be set to 111b to allow the S1D13704 to terminate bus cycles externally.

5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13704. 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 13704CFG, or by directly modifying the source. The Windows CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13704 test utilities and Windows CE v2.0 display drivers are available from your sales support contact or on the internet at http://www.erd.epson.com.

6 References

6.1 Documents

- Motorola Inc., *MC68328 DragonBall® Integrated Microprocessor User's Manual*, Motorola Publication no. MC68328UM/AD; available on the Internet at http://www.mot.com/SPS/WIRELESS/products/MC68328.html.
- Epson Research and Development, Inc., *S1D13704 Hardware Functional Specification*; Document Number X26A-A-001-xx.
- Epson Research and Development, Inc., *S5U13704B00C Rev. 1.0 ISA Bus Evaluation Board User Manual;* Document Number X26A-G-005-xx.
- Epson Research and Development, Inc., *S1D13704 Programming Notes and Examples*; Document Number X26A-G-002-xx.

6.2 Document Sources

- Motorola Inc.: Motorola Literature Distribution Center, (800) 441-2447.
- Motorola Website: http://www.mot.com.
- Epson Research and Development Website: http://www.erd.epson.com.

7 Technical Support

7.1 EPSON LCD Controllers (S1D13704)

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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 MC68328 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13704 Embedded Memory Color LCD Controller

Interfacing to the NEC VR4102™ Microprocessor

Document Number: X26A-G-008-05

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 required to provide an interface between the S1D13704 Embedded Memory LCD Controller and the NEC VR4102 Microprocessor (*u*PD30102). The NEC VR4102 Microprocessor is specifically designed to support an external LCD controller and the pairing of these two devices results in an embedded system offering impressive display capability with very low power consumption.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note will be updated as appropriate. Please check the Epson Research and Development Website at http://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 techpubs@erd.epson.com.

2 Interfacing to the NEC VR4102

2.1 The NEC VR4102 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 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 VR4102 is designed around the RISC architecture developed by MIPS. This microprocessor is designed around the 66MHz VR4100 CPU core which supports 64-bit processing. The CPU communicates with the Bus Control Unit (BCU) with its internal SysAD bus. The BCU in turn communicates with external devices with its ADD and DAT buses that can be dynamically sized to 16 or 32-bit operation.

The NEC VR4102 has direct support for an external LCD controller. Specific control signals are assigned for an external LCD controller that 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

Figure 2-1: "NEC VR4102 Read/Write Cycles," on page 9 shows the read and write cycles to the LCD Controller Interface.

Once an address in the LCD block of memory is placed on the external address bus, ADD[25:0], the LCD chip select, LCDCS#, is driven low. The read or write enable signals, RD# and WR#, are driven low for the appropriate cycle and LCDRDY is driven low to insert wait states into the cycle. The high byte enable is driven low for 16-bit transfers and high for 8-bit transfers.

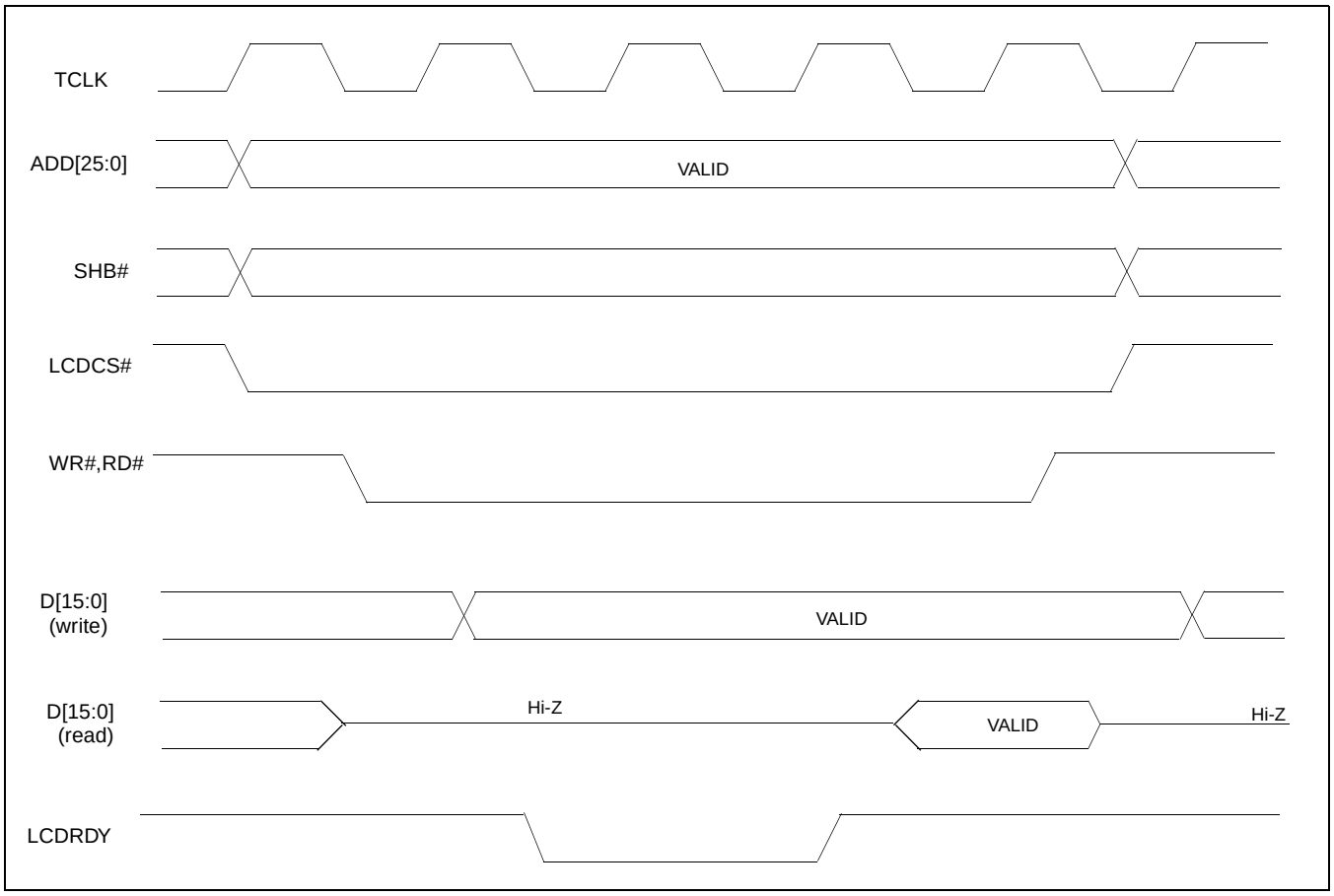


Figure 2-1: NEC VR4102 Read/Write Cycles

3 S1D13704 Host Bus Interface

This section is a summary of the host bus interface modes available on the S1D13704 and offers some detail on the Generic #2 host bus interface used to implement the interface to the VR4102.

3.1 Bus Interface Modes

The S1D13704 implements a 16-bit interface to the host microprocessor which may operate in one of several modes compatible with most of the popular embedded microprocessor families. Six bus interface modes are supported:

- Hitachi SH-4.
- Hitachi SH-3
- Motorola MC68000 (using Upper Data Strobe/Lower Data Strobe).
- Motorola MC68020/MC68030/MC683xx (using Data Strobe/DSACKx).
- Generic #1 (Chip Select, plus individual Read Enable/Write Enable for each byte).
- Generic #2 (External Chip Select, shared Read/Write Enable for high byte, individual Read/Write Enable for low byte).

The S1D13704 latches CNF[2:0] and BS# to allow selection of the host bus interface on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. The following table shows the functions of each host bus interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13704 Pin Names	SH-3	SH-4	MC68K #1	MC68K #2	Generic #1	Generic #2
AB[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]
AB0	A0	A0	LDS#	A0	A0	A0
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[31:16]	D[15:0]	D[15:0]
WE1#	WE1#	WE1#	UDS#	DS#	WE1#	BHE#
CS#	CSn#	CSn#	External Decode	External Decode	External Decode	External Decode
BCLK	CKIO	CKIO	CLK	CLK	BCLK	BCLK
BS#	BS#	BS#	AS#	AS#	connect to V_{SS}	connect to IO V_{DD}
RD/WR#	RD/WR#	RD/WR#	R/W#	R/W#	RD1#	connect to IO V_{DD}
RD#	RD#	RD#	connect to IO V_{DD}	SIZ1	RD0#	RD#
WE0#	WE0#	WE0#	connect to IO V_{DD}	SIZ0	WE0#	WE#
WAIT#	WAIT#	RDY#	DTACK#	DSACK1#	WAIT#	WAIT#
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	RESET#

Two other configuration options (CNF[4:3]) are also made at time of hardware reset:

- endian mode setting (big endian or little endian).
- polarity of the LCDPWR signal.

The capability to select the endian mode independent of the host bus interface offers more flexibility in configuring the S1D13704 with other CPUs.

For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

3.2 Generic #2 Interface Mode

Generic #2 interface mode is a general and non-processor-specific interface mode on the S1D13704. The Generic # 2 interface mode was chosen for this interface due to the simplicity of its timing and compatibility with the VR4102 control signals.

The interface requires the following signals:

- BUSCLK is a clock input which synchronizes transfers between the host CPU and the S1D13704. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB0 through AB15, and the data bus DB0 through DB15, connect directly to the CPU address and data bus, respectively. On 32-bit big endian architectures such as the Power PC, the data bus would connect to the high-order data lines; on little endian hosts, or 16-bit big endian hosts, they would connect to the low-order data lines. The hardware engineer must ensure that CNF3 selects the proper endian mode upon reset.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper IO or memory address space.
- WE1# is the high byte enable for both read and write cycles and WE0# is the enable signal for a write access. These must be generated by external decode hardware based upon the control outputs from the host CPU.
- RD# is the read enable for the S1D13704, to be driven low when the host CPU is reading data from the S1D13704. RD# must be generated by external decode hardware based upon the control outputs from the host CPU.
- WAIT# is a signal which is output from the S1D13704 to the host CPU that indicates when data is ready (read cycle) or accepted (write cycle) on the host bus. Since host CPU accesses to the S1D13704 may occur asynchronously to the display update, it is possible that contention may occur in accessing the 13704 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#) and Read/Write (RD/WR#) signals are not used in the bus interface for Generic #2 mode. However, BS# is used to configure the S1D13704 for Generic #2 mode and should be tied high (connected to IO V_{DD}). RD/WR# should also be tied high.

4 VR4102 to S1D13704 Interface

4.1 Hardware Description

The NEC VR4102 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 interface, only one inverter is required to change the polarity of the system reset signal to active low. A pull-up resistor is attached to WAIT# to speed up its rise time when terminating a cycle.

The following diagram shows a typical implementation of the VR4102 to S1D13704 interface.

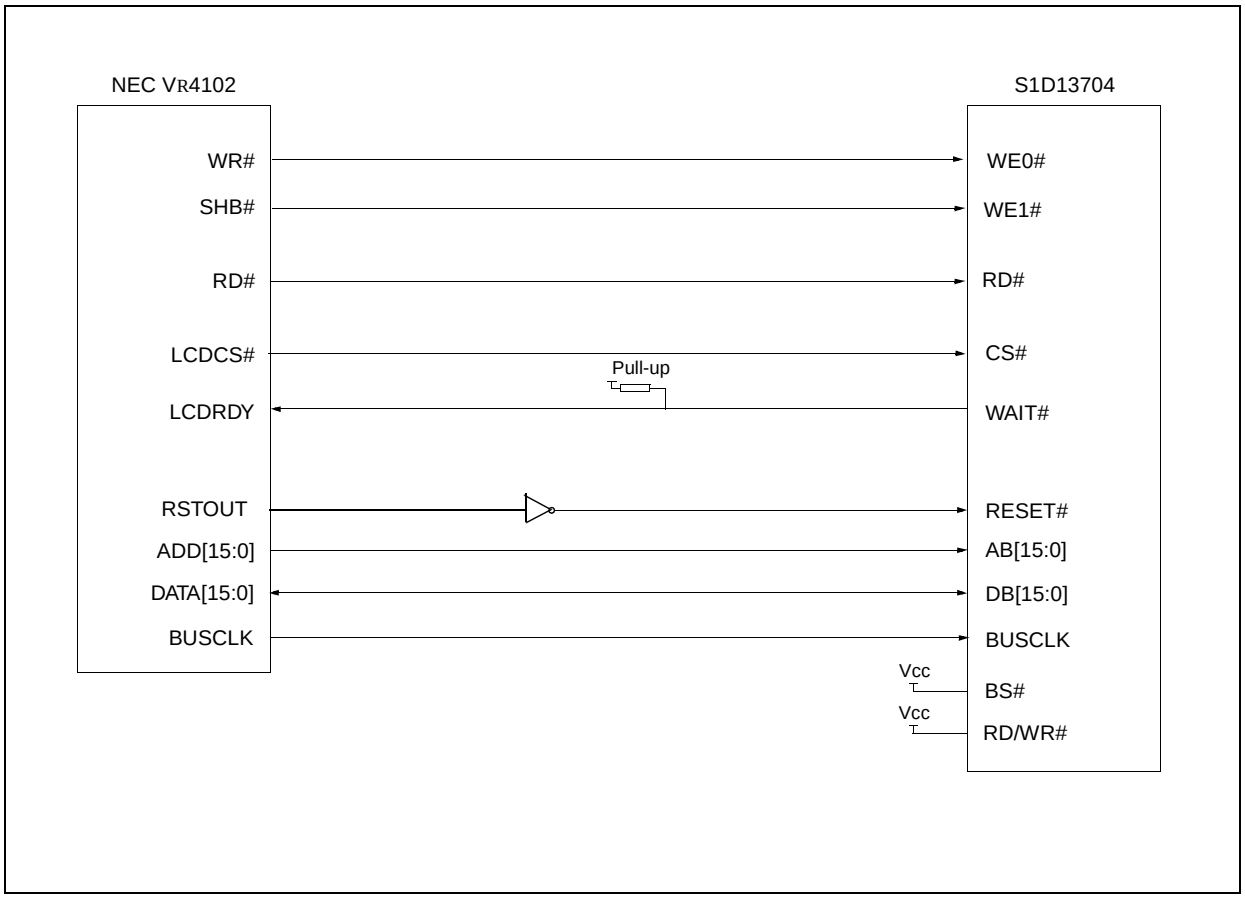


Figure 4-1: Typical Implementation of VR4102 to S1D13704 Interface

4.2 S1D13704 Hardware Configuration

The S1D13704 uses CNF4 through CNF0 and BS# to allow selection of the bus mode and other configuration data on the rising edge of RESET#. Refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx for details.

The tables below show those configuration settings important to the Generic #2 host bus interface.

Table 4-1: Summary of Power-On/Reset Options

Signal	value on this pin at the rising edge of RESET# is used to configure: (0/1)	
	0	1
CNF0	See “Host Bus Selection” table below	See “Host Bus Selection” table below
CNF1		
CNF2		
CNF3	Little Endian	Big Endian
CNF4	Active low LCDPWR signal	Active high LCDPWR signal

= configuration for NEC VR4102 support

Table 4-2: Host Bus Selection

CNF2	CNF1	CNF0	BS#	Host Bus Interface
0	0	0	X	SH-4 interface
0	0	1	X	SH-3 interface
0	1	0	X	reserved
0	1	1	X	MC68K #1, 16-bit
1	0	0	X	reserved
1	0	1	X	MC68K #2, 16-bit
1	1	0	0	reserved
1	1	0	1	reserved
1	1	1	0	Generic #1, 16-bit
1	1	1	1	Generic #2, 16-bit

= configuration for NEC VR4102 support

4.3 NEC VR4102 Configuration

The NEC VR4102 provides the internal address decoding necessary to map to an external LCD controller. Physical address 0A000000h to 0AFFFFFFh (16M bytes) is reserved for an external LCD controller.

The S1D13704 supports up to 40K bytes of display buffer memory and 32 bytes for internal registers. Therefore, the S1D13704 will be shadowed over the entire 16M byte memory range at 64K byte segments. The starting address of the display buffer is 0A000000h and register 0 of the S1D13704 (REG[00h]) resides at 0A00FFE0h.

The NEC VR4102 has a 16-bit internal register named BCUCNTREG2 located at address 0B000002h. It must be set to the value of 0001h to indicate that LCD controller accesses use a non-inverting data bus.

The 16-bit internal register named BCUCNTREG1, located at address 0B000000h, must have bit D[13] (ISA/LCD bit) set to 0 to reserve the 16M bytes space, 0A000000h to 0AFFFFFFh, for LCD use and not as ISA bus memory space.

5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13704. 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 13704CFG, or by directly modifying the source. The Windows CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13704 test utilities and Windows CE v2.0 display drivers are available from your sales support contact or on the internet at http://www.erd.epson.com.

6 References

6.1 Documents

- NEC VR4102 64/32-bit Microprocessor Preliminary User's Manual.
- Epson Research and Development, Inc., *S1D13704 Embedded Memory Color LCD Controller Hardware Functional Specification*; Document Number X26A-A-001-xx.
- Epson Research and Development, Inc., *S5U13704B00C Rev. 1.0 ISA Bus Evaluation Board User Manual;* Document Number X26A-G-005-xx.
- Epson Research and Development, Inc., *S1D13704 Programming Notes and Examples*; Document Number X26A-G-002-xx.

6.2 Document Sources

- NEC web page : http://www.nec.com.
- Epson Research and Development web page: http://www.erd.epson.com

7 Technical Support

7.1 Epson LCD Controllers (S1D13704)

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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.)

Santa Clara
California
Tel: (800) 366-9782
Fax: (800) 729-9288
http://www.nec.com

EPSON®

ENERGY SAVING EPSON

S1D13704 Embedded Memory Color LCD Controller

Interfacing to the PC Card Bus

Document Number: X26A-G-009-03

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 environment required to provide an interface between the S1D13704 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 will be updated as appropriate. Please check the Epson Research and Development Website at http://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 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 micro-processors, 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 16.

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: and Figure 2-2: illustrate typical memory access cycles on the PC Card bus.

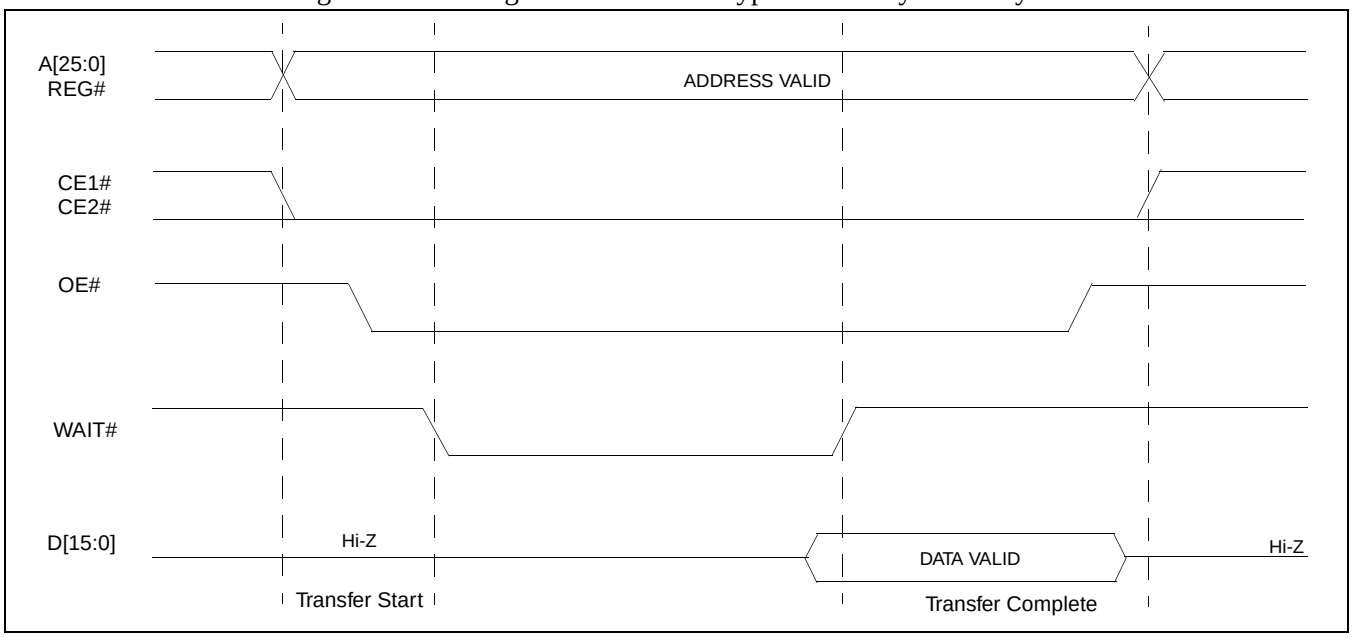


Figure 2-1: PC Card Read Cycle

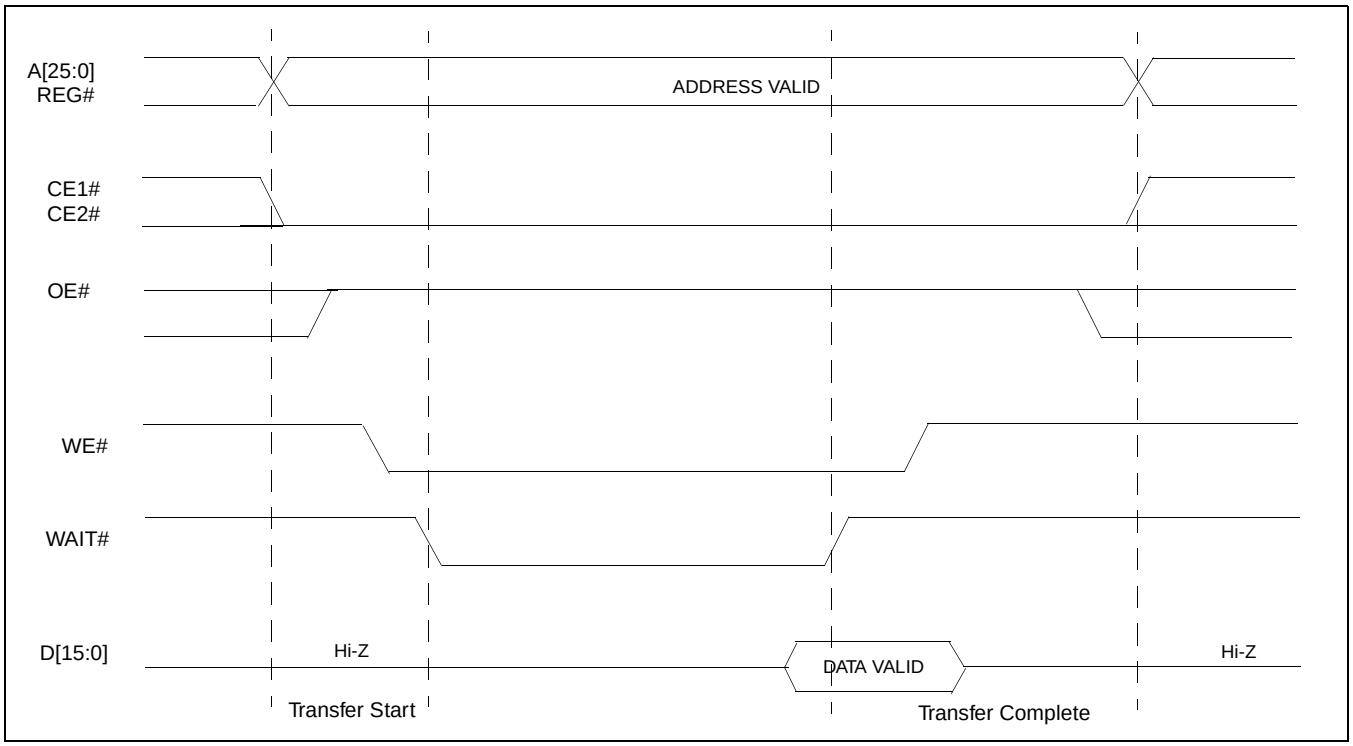


Figure 2-2: PC Card Write Cycle

3 S1D13704 Bus Interface

This section is a summary of the host bus interface modes available on the S1D13704 and offers some detail on the Generic #1 host bus interface used to implement the interface to the PC Card bus.

3.1 Bus Interface Modes

The S1D13704 implements a 16-bit interface to the host microprocessor which may operate in one of several modes compatible with most of the popular embedded microprocessor families. Six bus interface modes are supported:

- Hitachi SH-4.
- Hitachi SH-3
- Motorola MC68000 (using Upper Data Strobe/Lower Data Strobe).
- Motorola MC68020/MC68030/MC683xx (using Data Strobe/DSACKx).
- Generic #1 (Chip Select, plus individual Read Enable/Write Enable for each byte).
- Generic #2 (External Chip Select, shared Read/Write Enable for high byte, individual Read/Write Enable for low byte).

The S1D13704 latches CNF[2:0] and BS# to allow selection of the host bus interface on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. The following table shows the functions of each host bus interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13704 Pin Names	SH-3	SH-4	MC68K #1	MC68K #2	Generic #1	Generic #2
AB[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]
AB0	A0	A0	LDS#	A0	A0	A0
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[31:16]	D[15:0]	D[15:0]
WE1#	WE1#	WE1#	UDS#	DS#	WE1#	BHE#
CS#	CSn#	CSn#	External Decode	External Decode	External Decode	External Decode
BCLK	CKIO	CKIO	CLK	CLK	BCLK	BCLK
BS#	BS#	BS#	AS#	AS#	connect to V_{SS}	connect to IO V_{DD}
RD/WR#	RD/WR#	RD/WR#	R/W#	R/W#	RD1#	connect to IO V_{DD}
RD#	RD#	RD#	connect to IO V_{DD}	SIZ1	RD0#	RD#
WE0#	WE0#	WE0#	connect to IO V_{DD}	SIZ0	WE0#	WE#
WAIT#	WAIT#	RDY#	DTACK#	DSACK1#	WAIT#	WAIT#
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	RESET#

Two other configuration options (CNF[4:3]) are also made at time of hardware reset:

- endian mode setting (big endian or little endian).
- polarity of the LCDPWR signal.

The capability to select the endian mode independent of the host bus interface offers more flexibility in configuring the S1D13704 with other CPUs.

For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

3.2 Generic #1 Interface Mode

Generic #1 interface mode is the most general and least processor-specific interface mode on the S1D13704. The Generic # 1 interface mode was chosen for this interface due to the simplicity of its timing.

The interface requires the following signals:

- BUSCLK is a clock input which is required by the S1D13704 host interface. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB0 through AB15, and the data bus DB0 through DB15, connect directly to the CPU address and data bus, respectively. On 32-bit big endian architectures such as the Power PC, the data bus would connect to the high-order data lines; on little endian hosts, or 16-bit big endian hosts, they would connect to the low-order data lines. The hardware engineer must ensure that CNF3 selects the proper endian mode upon reset.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper IO or memory address space.
- WE0# and WE1# are write enables for the low-order and high-order bytes, respectively, to be driven low when the host CPU is writing data to the S1D13704. These signals must be generated by external hardware based on the control outputs from the host CPU.
- RD# and RD/WR# are read enables for the low-order and high-order bytes, respectively, to be driven low when the host CPU is reading data from the S1D13704. These signals must be generated by external hardware based on the control outputs from the host CPU.
- WAIT# is a signal output from the S1D13704 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 S1D13704 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13704 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. However, BS# is used to configure the S1D13704 for Generic #1 mode and should be tied low (connected to GND).

4 PC Card to S1D13704 Interface

4.1 Hardware Connections

The S1D13704 is interfaced to the PC Card interface with a minimal amount of glue logic. A PAL is used to decode the write and read signals of the PC Card bus to generate RD#, RD/WR#, WE0#, WE1#, and CS# for the S1D13704. The PAL also inverts the reset signal of the PC card since it is active high and the S1D13704 uses an active low reset. For PAL equations for this implementation refer to Section 4.3, "PAL Equations" on page 14.

In this implementation, the address inputs (AB[15:0]) and data bus (DB[15:0] connect directly to the CPU address (A[15:0]) and data bus (D[15:0]).

The PC Card interface does not provide a bus clock, so one must be supplied for the S1D13704. Since the bus clock frequency is not critical, nor does it have to be synchronous to the bus signals, it may be the same as CLKI.

BS# (bus start) is not used by Generic #1 mode but is used to configure the S1D13704 for Generic #1 and should be tied low (connected to GND).

The following diagram shows a typical implementation of the PC Card to S1D13704 interface.

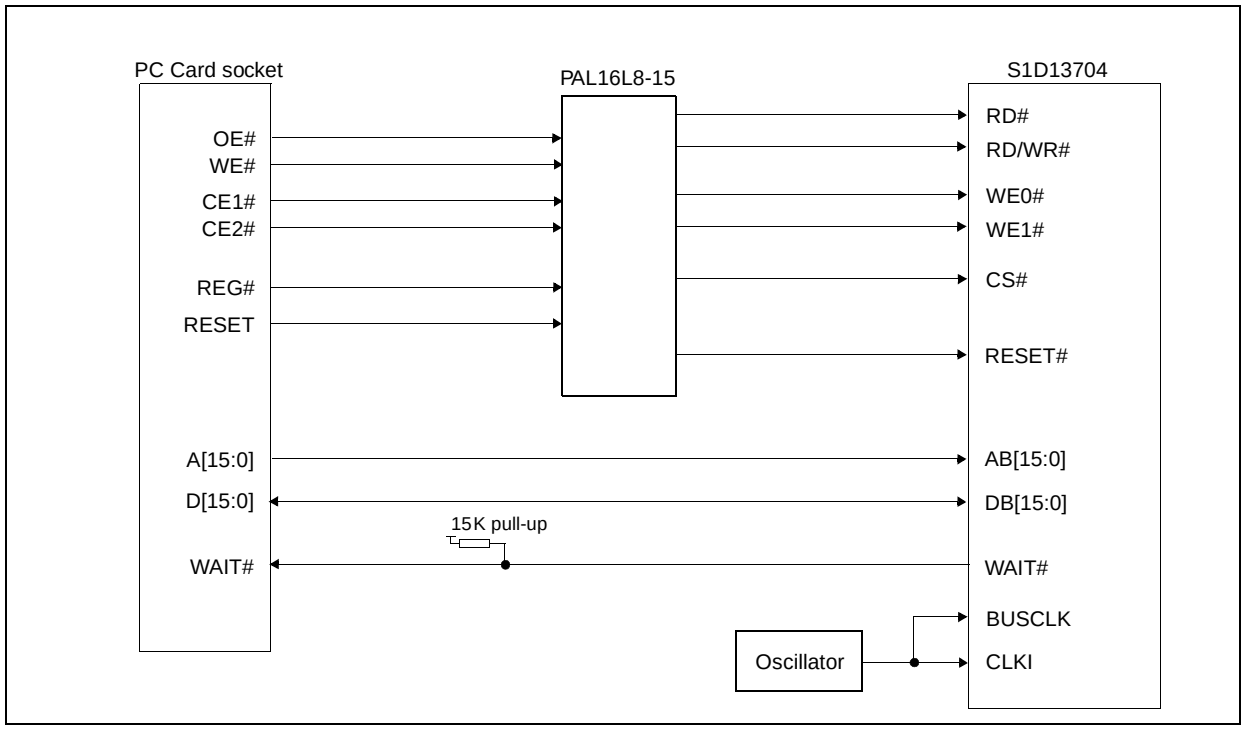


Figure 4-1: Typical Implementation of PC Card to S1D13704 Interface

4.2 S1D13704 Hardware Configuration

The S1D13704 uses CNF4 through CNF0 and BS# to allow selection of the bus mode and other configuration data on the rising edge of RESET#. Refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx for details.

The tables below show only those configuration settings important to the PC Card host bus interface.

Table 4-1: Summary of Power-On/Reset Options

Signal	Low	High
CNF0	See “Host Bus Selection” table below	See “Host Bus Selection” table below
CNF1		
CNF2		
CNF3	Little Endian	Big Endian
CNF4	Active low LCDPWR signal	Active high LCDPWR signal

= configuration for PC Card host bus interface

Table 4-2: Host Bus Interface Selection

CNF2	CNF1	CNF0	BS#	Host Bus Interface
0	0	0	X	SH-4 bus interface
0	0	1	X	SH-3 bus interface
0	1	0	X	reserved
0	1	1	X	MC68K bus interface #1, 16-bit
1	0	0	X	reserved
1	0	1	X	MC68K bus interface #2, 16-bit
1	1	0	0	reserved
1	1	0	1	reserved
1	1	1	0	Generic #1, 16-bit
1	1	1	1	Generic #2, 16-bit

= configuration for PC Card host bus interface

4.3 PAL Equations

The PAL equations for the implementation presented in this document are as follows.

```
PAL  device  '16L8';

OE                              PIN   1;
WE                              PIN   2;
CE1                             PIN   3;
CE2                             PIN   4;
REG                             PIN   5;
PCRESET                         PIN   6;
RESET                           PIN  14;
WE0                             PIN  15;
WE1                             PIN  16;
RD                              PIN  17;
RDWR                            PIN  18;
CS                              PIN  19;

equations

!WE0 = !WE & !CE1 & REG;
!WE1 = !WE & !CE2 & REG;
!CS = REG & (!RD # !RDWR # !WE0 # !WE1);
!RD = !OE & !CE1 & REG;
!RDWR = !OE & !CE2 & REG;
!RESET = PCRESET;
```

4.4 Register/Memory Mapping

The S1D13704 is a memory mapped device. The S1D13704 memory may be addressed starting at 0000h, or on consecutive 64K byte blocks, and its internal registers are located in the upper 32 bytes of the 64K byte block (i.e. REG[0] = FFE0h).

While the PC Card socket provides 64M bytes of address space, the S1D13704 only needs a 64K byte block of memory to accommodate its 40K byte display buffer and its 32 byte register set. For this reason only address bits A[15:0] are used while A[25:16] are ignored. Because the entire 64M bytes of memory is available, the S1D13704's memory and registers will be aliased every 64K bytes for a total of 1024 times.

Note

If aliasing is not desirable, the upper addresses must be fully decoded.

5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13704. 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 13704CFG, or by directly modifying the source. The Windows CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13704 test utilities and Windows CE v2.0 display drivers are available from your sales support contact or on the internet at http://www.erd.epson.com.

6 References

6.1 Documents

- PC Card (PCMCIA) Standard March 1997
- Epson Research and Development, Inc., *S1D13704 Embedded Memory Color LCD Controller Hardware Functional Specification*; Document Number X26A-A-001-xx.
- Epson Research and Development, Inc., *S5U13704B00C Rev. 1.0 ISA Bus Evaluation Board User Manual;* Document Number X26A-G-005-xx.
- Epson Research and Development, Inc., *S1D13704 Programming Notes and Examples*; Document Number X26A-G-002-xx.

6.2 Document Sources

- PC Card web page: http://www.pc-card.com.
- EPSON Research and Development web page: http://www.erd.epson.com

7 Technical Support

7.1 EPSON LCD Controllers (S1D13704)

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13704 Embedded Memory Color LCD Controller

Interfacing to the Motorola MPC821 Microprocessor

Document Number: X26A-G-010-03

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 environment required to provide an interface between the S1D13704 Embedded Memory LCD Controller and the Motorola MPC821 Processor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note will be updated as appropriate. Please check the Epson Research and Development Website at http://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 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 MPC821 Bus Overview

The MPC8xx microprocessor family uses a synchronous address and data bus. All IO is synchronous to a square-wave reference clock called MCLK (Master Clock). This clock runs at the machine cycle speed of the CPU core (typically 25 to 50 MHz). Most outputs from the processor change state on the rising edge of this clock. Similarly, most inputs to the processor are sampled on the rising edge.

Note
The external bus can run at one-half the CPU core speed using the clock control register. This is typically used when the CPU core is operated above 50 MHz.

The MPC821 can generate up to eight independent chip select outputs, each of which may be controlled by one of two types of timing generators: the General Purpose Chip Select Module (GPCM) or the User-Programmable Machine (UPM). Examples are given using the GPCM.

It should be noted that all Power PC microprocessors, including the MPC8xx family, use bit notation opposite from the convention used by most other microprocessor systems. Bit numbering for the MPC8xx always starts with zero as the most significant bit, and increments in value to the least-significant bit. For example, the most significant bits of the address bus and data bus are A0 and D0, while the least significant bits are A31 and D31.

The MPC8xx uses both a 32-bit address and data bus. A parity bit is supported for each of the four byte lanes on the data bus. Parity checking is done when data is read from external memory or peripherals, and generated by the MPC8xx bus controller on write cycles. All IO accesses are memory-mapped meaning there is no separate IO space in the Power PC architecture.

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 23.

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.

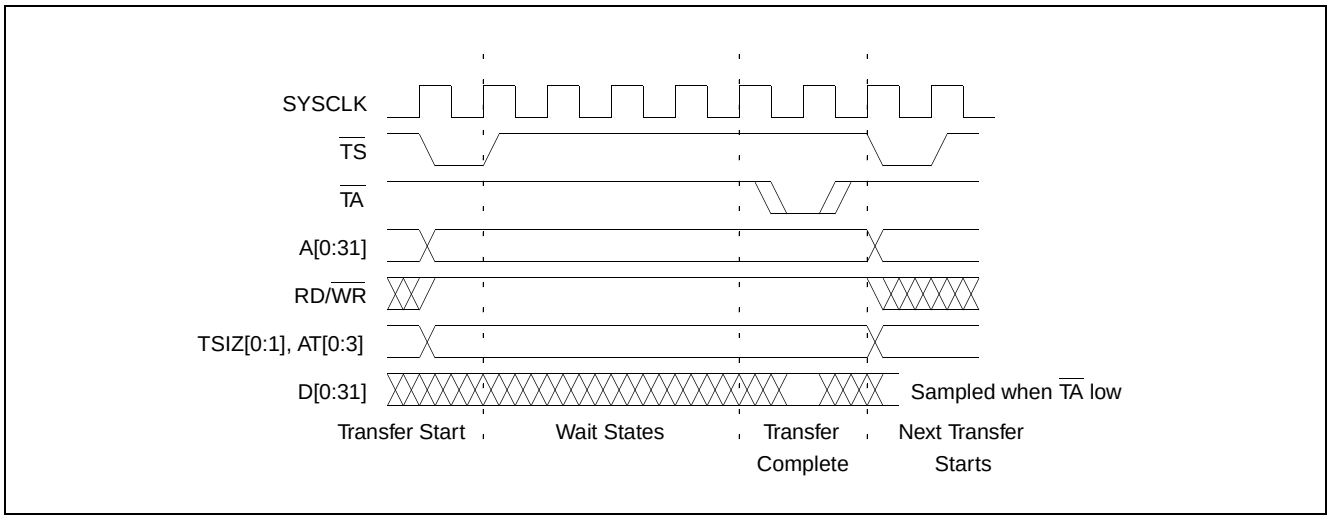


Figure 2-1: Power PC Memory Read Cycle

Figure 2-2: “Power PC Memory Write Cycle” illustrates a typical memory write cycle on the Power PC system bus.

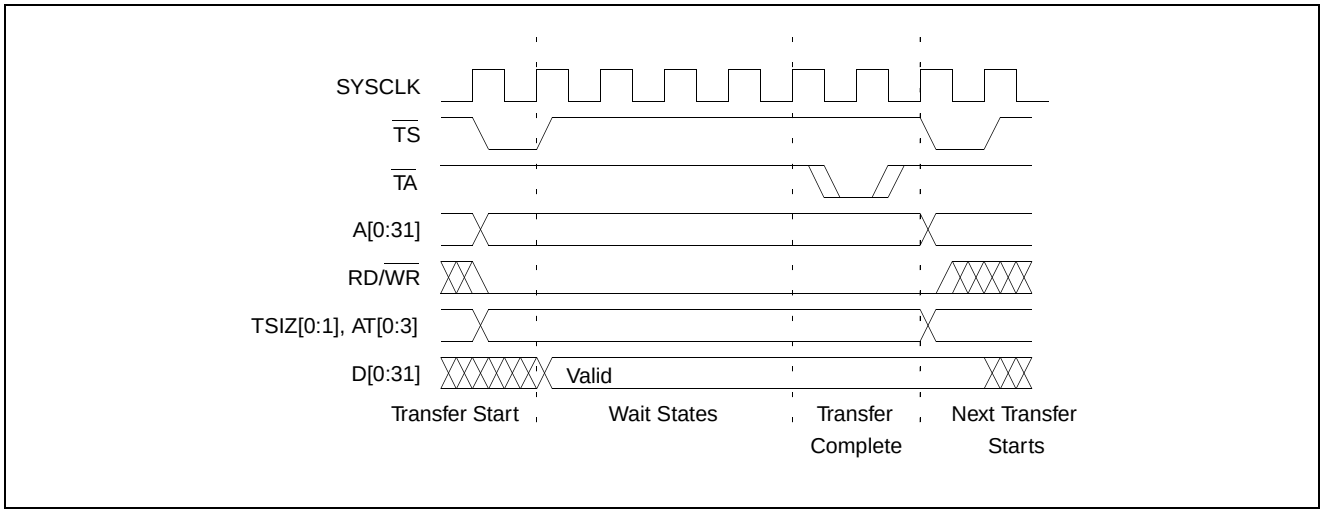


Figure 2-2: Power PC Memory Write Cycle

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 A30 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 will revert to normal bus cycles for the remaining data transfers.

Burst cycles are mainly intended to facilitate cache line fills from program or data memory. They are normally not used for transfers to/from IO peripheral devices such as the S1D13504, therefore the interfaces described in this document do not attempt to support burst cycles. However, the example interfaces include circuitry to detect the assertion of $\overline{BDIP}$ and respond with $\overline{BI}$ if caching is accidently enabled for the S1D13504 address space.

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 will insert an additional one clock delay between assertion of the address bus and chip select. This accommodates memory and peripherals with long setup times.
- The EHTR (Extended hold time) bit will insert an additional 1-clock delay on the first access to a chip select.
- Up to 15 wait states may be inserted, or the peripheral can terminate the bus cycle itself by asserting $\overline{TA}$ (Transfer Acknowledge).
- Any chip select may be programmed to assert $\overline{BI}$ (Burst Inhibit) automatically when its memory space is addressed by the processor core.

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 S1D13504 and it is desirable to leave the UPM free to handle other interfacing duties, such as EDO DRAM.

3 S1D13704 Host Bus Interface

This section is a summary of the host bus interface modes available on the S1D13704 and offers some detail on the Generic #1 host bus interface used to implement the interface to the MPC821 bus.

3.1 Host Bus Interface Modes

The S1D13704 implements a 16-bit interface to the host microprocessor which may operate in one of several modes compatible with most of the popular embedded microprocessor families. Six host bus interface modes are supported:

- Hitachi SH-4.
- Hitachi SH-3
- Motorola MC68000 (using Upper Data Strobe/Lower Data Strobe).
- Motorola MC68020/MC68030/MC683xx (using Data Strobe/DSACKx).
- Generic #1 (Chip Select, plus individual Read Enable/Write Enable for each byte).
- Generic #2 (External Chip Select, shared Read/Write Enable for high byte, individual Read/Write Enable for low byte).

The S1D13704 latches CNF[2:0] and BS# to allow selection of the host bus interface on the rising edge of RESET#. After releasing reset, the host bus interface signals assume their selected configuration. The following table shows the functions of each host bus interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13704 Pin Names	SH-3	SH-4	MC68K #1	MC68K #2	Generic #1	Generic #2
AB[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]
AB0	A0	A0	LDS#	A0	A0	A0
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[31:16]	D[15:0]	D[15:0]
WE1#	WE1#	WE1#	UDS#	DS#	WE1#	BHE#
CS#	CSn#	CSn#	External Decode	External Decode	External Decode	External Decode
BCLK	CKIO	CKIO	CLK	CLK	BCLK	BCLK
BS#	BS#	BS#	AS#	AS#	connect to V_{SS}	connect to IO V_{DD}
RD/WR#	RD/WR#	RD/WR#	R/W#	R/W#	RD1#	connect to IO V_{DD}
RD#	RD#	RD#	connect to IO V_{DD}	SIZ1	RD0#	RD#
WE0#	WE0#	WE0#	connect to IO V_{DD}	SIZ0	WE0#	WE#
WAIT#	WAIT#	RDY#	DTACK#	DSACK1#	WAIT#	WAIT#
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	RESET#

Two other configuration options (CNF[4:3]) are also made at time of hardware reset:

- endian mode setting (big endian or little endian).
- polarity of the LCDPWR signal.

The capability to select the endian mode independent of the host bus interface offers more flexibility in configuring the S1D13704 with other CPUs.

For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

3.2 Generic #1 Host Bus Interface Mode

Generic #1 host bus interface mode is the most general and least processor-specific host bus interface mode on the S1D13704. The Generic # 1 host bus interface mode was chosen for this interface due to the simplicity of its timing.

The host bus interface requires the following signals:

- BUSCLK is a clock input which is required by the S1D13704 host interface. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB0 through AB15, and the data bus DB0 through DB15, connect directly to the CPU address and data bus, respectively. On 32-bit big endian architectures such as the Power PC, the data bus would connect to the high-order data lines; on little endian hosts, or 16-bit big endian hosts, they would connect to the low-order data lines. The hardware engineer must ensure that CNF3 selects the proper endian mode upon reset.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper IO or memory address space.
- WE0# and WE1# are write enables for the low-order and high-order bytes, respectively, to be driven low when the host CPU is writing data to the S1D13704. These signals must be generated by external hardware based on the control outputs from the host CPU.
- RD# and RD/WR# are read enables for the low-order and high-order bytes, respectively, to be driven low when the host CPU is reading data from the S1D13704. These signals must be generated by external hardware based on the control outputs from the host CPU.
- WAIT# is a signal output from the S1D13704 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 S1D13704 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13704 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. However, BS# is used to configure the S1D13704 for Generic #1 mode and should be tied low (connected to GND).

4 MPC821 to S1D13704 Interface

4.1 Hardware Description

The interface between the S1D13704 and the MPC821 requires minimal glue logic. One inverter is required to change the polarity of the WAIT# signal (an active low signal) to insert wait states in the bus cycle. The MPC821 Transfer Acknowledge signal ($\overline{TA}$) is an active low signal which ends the current bus cycle. The inverter is enabled using CS# so that $\overline{TA}$ is not driven by the S1D13704 during non-S1D13704 bus cycles. A single resistor is used to speed up the rise time of the WAIT# ($\overline{TA}$) signal when terminating the bus cycle.

BS# (bus start) is not used in this implementation and should be tied low (connected to GND).

The following diagram shows a typical implementation of the MPC821 to S1D13704 interface.

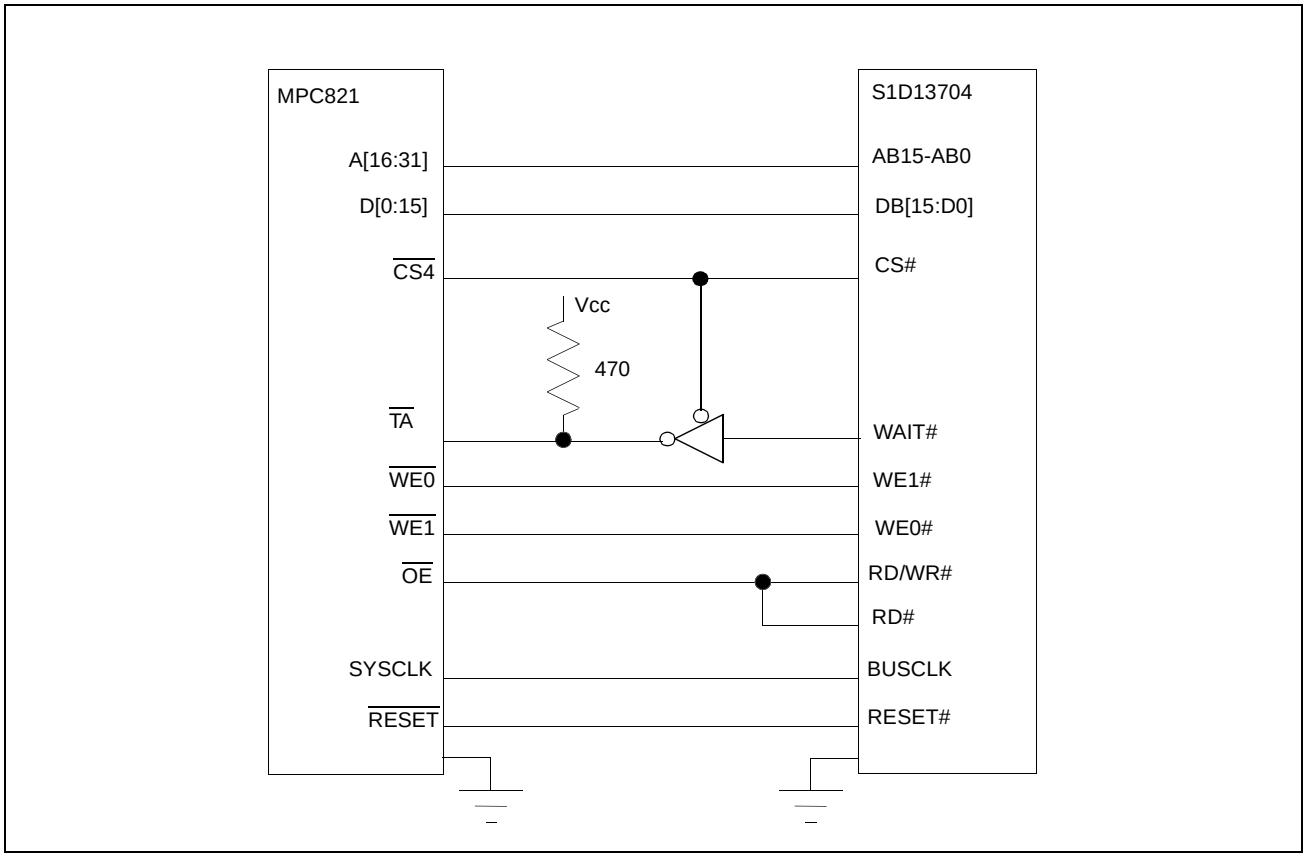


Figure 4-1: Typical Implementation of MPC821 to S1D13704 Interface

4.2 Hardware Connections

The following table details the connections between the pins and signals of the MPC821 and the S1D13704.

Table 4-1: List of Connections from MPC821ADS to S1D13704

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13704 Signal Name
Vcc	P6-A1, P6-B1	Vcc
A16	P6-B24	SA15
A17	P6-C24	SA14
A18	P6-D23	SA13
A19	P6-D22	SA12
A20	P6-D19	SA11
A21	P6-A19	SA10
A22	P6-D28	SA9
A23	P6-A28	SA8
A24	P6-C27	SA7
A25	P6-A26	SA6
A26	P6-C26	SA5
A27	P6-A25	SA4
A28	P6-D26	SA3
A29	P6-B25	SA2
A30	P6-B19	SA1
A31	P6-D17	SA0
D0	P12-A9	SD15
D1	P12-C9	SD14
D2	P12-D9	SD13
D3	P12-A8	SD12
D4	P12-B8	SD11
D5	P12-D8	SD10
D6	P12-B7	SD9
D7	P12-C7	SD8
D8	P12-A15	SD7
D9	P12-C15	SD6
D10	P12-D15	SD5
D11	P12-A14	SD4
D12	P12-B14	SD3
D13	P12-D14	SD2
D14	P12-B13	SD1
D15	P12-C13	SD0
SRESET	P9-D15	RESET#
SYSCLK	P9-C2	BUSCLK
CS4	P6-D13	CS#

Table 4-1: List of Connections from MPC821ADS to S1D13704 (Continued)

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13704 Signal Name
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 Power PC 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 S1D13704 Hardware Configuration

The S1D13704 uses CNF4 through CNF0 and BS# to allow selection of the bus mode and other configuration data on the rising edge of RESET#. Refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx for details.

The tables below show only those configuration settings important to the MPC821 interface. The settings are very similar to the ISA bus with the following exceptions:

- the WAIT# signal is active high rather than active low.
- the Power PC is big endian rather than little endian.

Table 4-2: Configuration Settings

Signal	Low	High
CNF0	See “Host Bus Selection” table below	See “Host Bus Selection” table below
CNF1		
CNF2		
CNF3	Little Endian	Big Endian
CNF4	Active low LCDPWR signal	Active high LCDPWR signal

= configuration for MPC821 host bus interface

Table 4-3: Host Bus Selection

CNF2	CNF1	CNF0	BS#	Host Bus Interface
0	0	0	X	SH-4 interface
0	0	1	X	SH-3 interface
0	1	0	X	reserved
0	1	1	X	MC68K #1, 16-bit
1	0	0	X	reserved
1	0	1	X	MC68K #2, 16-bit
1	1	0	0	reserved
1	1	0	1	reserved
1	1	1	0	Generic #1, 16-bit
1	1	1	1	Generic #2, 16-bit

= configuration for MPC821 host bus interface

4.4 MPC821 Chip Select Configuration

The DRAM on the MPC821 ADS board extends from address 0 through 3F FFFFh, so the S1D13704 is addressed starting at 40 0000h. The S1D13704 uses a 64K byte segment of memory starting at this address, with the first 40K bytes used for the display buffer and the upper 32 bytes of this memory block used for the S1D13704 internal registers.

Chip select 4 is used to control the S1D13704. The following options are selected in the base address register (BR4):

- BA (0:16) = 0000 0000 0100 0000 0 – set starting address of S1D13704 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; S1D13704 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 S1D13704 generates an external transfer acknowledge using the WAIT# line
- TRLX = 0 – normal timing
- EHTR = 0 – normal timing

4.5 Test Software

The test software to exercise this interface is very simple. It configures chip select 4 on the MPC821 to map the S1D13704 to an unused 64k byte block of address space and loads the appropriate values into the option register for CS4. At that point the software runs in a tight loop reading the 13704 Revision Code Register REG[00h], which allows monitoring of the bus timing on a logic analyzer.

The source code for this test routine is as follows:

```
BR4             equ             $120            ; CS4 base register
OR4             equ             $124            ; CS4 option register
MemStart        equ             $40             ; upper word of S1D13704 start
address
RevCodeReg      equ             FFE0            ; address of Revision Code Regis-
ter

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 S1D13704
mem space
Loop            lbz             r0,RevCodeReg(r1) ; read revision code into r1
                b               Loop            ; branch forever

end
```

This code was entered into the memory of the MPC821ADS using the line-by-line assembler in MPC8BUG, the debugger provided with the ADS board. It was executed on the ADS and a logic analyzer was used to verify operation of the interface hardware.

Note

MPC8BUG does not support comments or symbolic equates; these have been added for clarity.

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 S1D13704 memory block is tagged as non-cacheable, to ensure that accesses to the S1D13704 will occur in proper order, and also to ensure that the MPC821 does not attempt to cache any data read from or written to the S1D13704 or its display buffer.

5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13704. 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 13704CFG, or by directly modifying the source. The Windows CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13704 test utilities and Windows CE v2.0 display drivers are available from your sales support contact or on the internet at http://www.erd.epson.com.

6 References

6.1 Documents

- Motorola Inc., *Power PC MPC821 Portable Systems Microprocessor User's Manual*, Motorola Publication no. MPC821UM/AD; available on the Internet at http://www.mot.com/SPS/ADC/pps/_subpgs/_documentation/821/821UM.html.
- Epson Research and Development, Inc., *S1D13704 Embedded Memory LCD Controller Hardware Functional Specification*; Document Number X126A-A-002-xx.
- Epson Research and Development, Inc., *S5U13704B00C Rev. 1.0 ISA Bus Evaluation Board User Manual;* Document Number X26A-G-005-xx.
- Epson Research and Development, Inc., *Programming Notes and Examples*; Document Number X26A-G-002-xx.

6.2 Document Sources

- Motorola Inc. Literature Distribution Center: (800) 441-2447.
- Motorola Inc. Website: http://www.mot.com.
- Epson Research and Development Website: http://www.erd.epson.com.

7 Technical Support

7.1 EPSON LCD/CRT Controllers (S1D13704)

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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.

EPSON®

ENERGY SAVING EPSON

S1D13704 Embedded Memory Color LCD Controller

Interfacing to the Motorola MCF5307 "ColdFire" Microprocessor

Document Number: X26A-G-011-03

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 required to provide an interface between the S1D13704 Embedded Memory LCD Controller and the Motorola MCF5307 Processor. The pairing of these two devices results in an embedded system offering impressive display capability with very low power consumption.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note will be updated as appropriate. Please check the Epson Research and Development Website at http://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 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 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), which indicate whether the bus cycle is 8, 16, or 32 bits in width.
- R/$\overline{W}$, which is high for read cycles and low for write cycles.
- A set of transfer type signals (TT[1:0]) which provide more detail on the type of transfer being attempted.
- $\overline{TIP}$ (Transfer In Progress), which is asserted 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, completing 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 illustrates a typical memory read cycle on the MCF5307 system bus, and Figure 2-2 illustrates a memory write cycle.

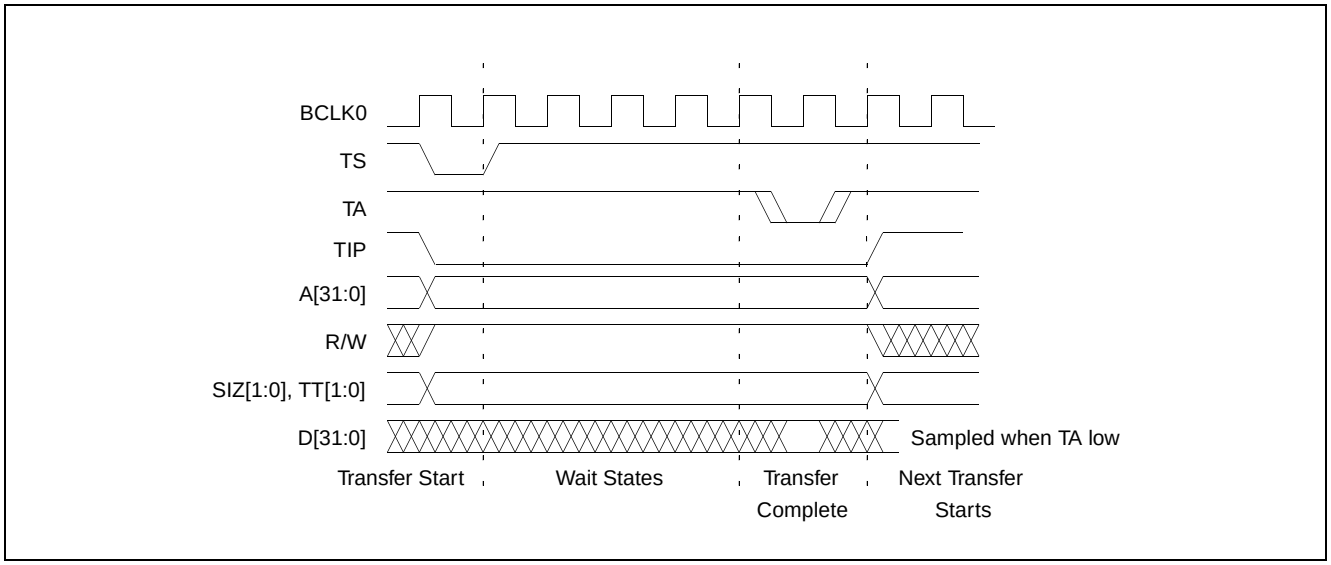


Figure 2-1: MCF5307 Memory Read Cycle

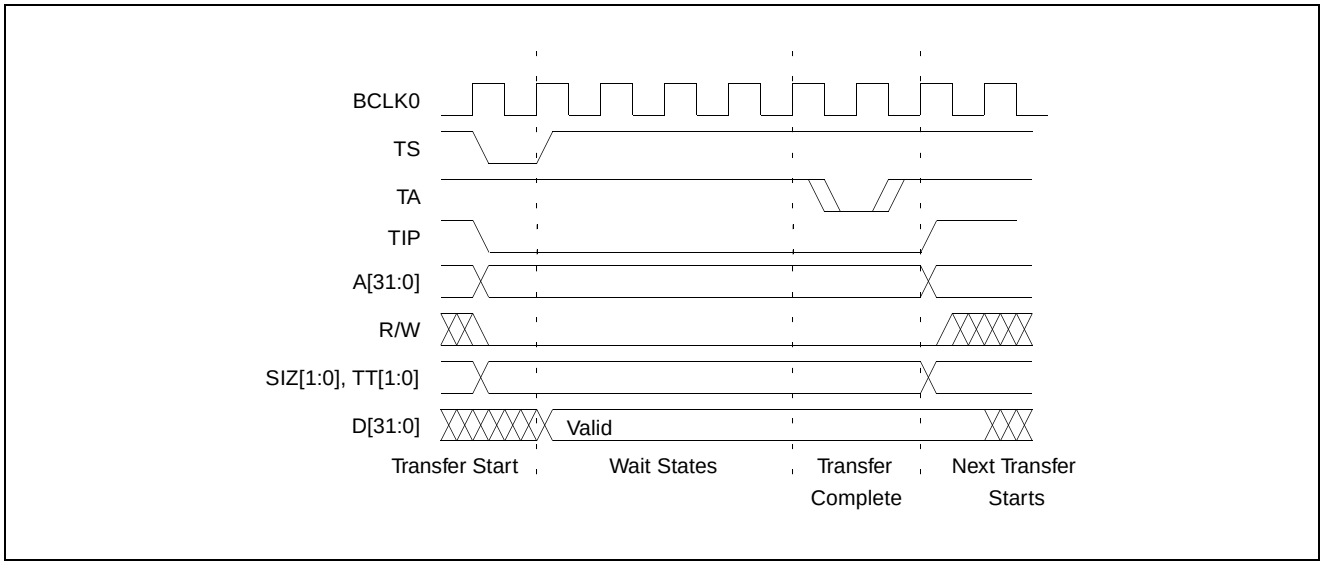


Figure 2-2: MCF5307 Memory Write Cycle

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, with the $\overline{\text{TIP}}$ (Transfer In Progress) output asserted continuously through the burst. Burst memory cycles are mainly intended to facilitate cache line fill from program or data memory; they are typically not used for transfers to or from IO peripheral devices such as the S1D13704. The MCF5307 chip selects provide a mechanism to disable burst accesses for peripheral devices which are not able to support them.

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 ($\overline{WE}$) 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 bits), 0-15 wait states or external acknowledge, address space type, burst or non-burst cycle support, and write protect.

3 S1D13704 Bus Interface

This section is a summary of the host bus interface modes available on the S1D13704 and offers some detail on the Generic #1 host bus interface used to implement the interface to the MCF5307.

3.1 Bus Interface Modes

The S1D13704 implements a 16-bit interface to the host microprocessor which may operate in one of several modes compatible with most of the popular embedded microprocessor families. Six bus interface modes are supported:

- Hitachi SH-4.
- Hitachi SH-3
- Motorola MC68000 (using Upper Data Strobe/Lower Data Strobe).
- Motorola MC68020/MC68030/MC683xx (using Data Strobe/DSACKx).
- Generic #1 (Chip Select, plus individual Read Enable/Write Enable for each byte).
- Generic #2 (External Chip Select, shared Read/Write Enable for high byte, individual Read/Write Enable for low byte).

The S1D13704 latches CNF[2:0] and BS# to allow selection of the host bus interface on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. The following table shows the functions of each host bus interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13704 Pin Names	SH-3	SH-4	MC68K #1	MC68K #2	Generic #1	Generic #2
AB[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]	A[15:1]
AB0	A0	A0	LDS#	A0	A0	A0
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[31:16]	D[15:0]	D[15:0]
WE1#	WE1#	WE1#	UDS#	DS#	WE1#	BHE#
CS#	CSn#	CSn#	External Decode	External Decode	External Decode	External Decode
BCLK	CKIO	CKIO	CLK	CLK	BCLK	BCLK
BS#	BS#	BS#	AS#	AS#	connect to V_{SS}	connect to IO V_{DD}
RD/WR#	RD/WR#	RD/WR#	R/W#	R/W#	RD1#	connect to IO V_{DD}
RD#	RD#	RD#	connect to IO V_{DD}	SIZ1	RD0#	RD#
WE0#	WE0#	WE0#	connect to IO V_{DD}	SIZ0	WE0#	WE#
WAIT#	WAIT#	RDY#	DTACK#	DSACK1#	WAIT#	WAIT#
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	RESET#

Two other configuration options (CNF[4:3]) are also made at time of hardware reset:

- endian mode setting (big endian or little endian).
- polarity of the LCDPWR signal.

The capability to select the endian mode independent of the host bus interface offers more flexibility in configuring the S1D13704 with other CPUs.

For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

3.2 Generic #1 Interface Mode

Generic #1 interface mode is the most general and least processor-specific interface mode on the S1D13704. The Generic # 1 interface mode was chosen for this interface due to the simplicity of its timing.

The interface requires the following signals:

- BUSCLK is a clock input which is required by the S1D13704 host interface. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB0 through AB15, and the data bus DB0 through DB15, connect directly to the CPU address and data bus, respectively. On 32-bit big endian architectures such as the Power PC, the data bus would connect to the high-order data lines; on little endian hosts, or 16-bit big endian hosts, they would connect to the low-order data lines. The hardware engineer must ensure that CNF3 selects the proper endian mode upon reset.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper IO or memory address space.
- WE0# and WE1# are write enables for the low-order and high-order bytes, respectively, to be driven low when the host CPU is writing data to the S1D13704. These signals must be generated by external hardware based on the control outputs from the host CPU.
- RD# and RD/WR# are read enables for the low-order and high-order bytes, respectively, to be driven low when the host CPU is reading data from the S1D13704. These signals must be generated by external hardware based on the control outputs from the host CPU.
- WAIT# is a signal output from the S1D13704 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 S1D13704 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13704 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. However, BS# is used to configure the S1D13704 for Generic #1 mode and should be tied low (connected to GND).

4 MCF5307 To S1D13704 Interface

4.1 Hardware Description

The S1D13704 is interfaced to the MCF5307 with a minimal amount of glue logic. One inverter is required to change the polarity of the WAIT# signal, which is an active low signal to insert wait states in the bus cycle, while the MCF5307's Transfer Acknowledge signal ($\overline{TA}$) is an active low signal to end the current bus cycle. The inverter is enabled by CS# so that $\overline{TA}$ is not driven by the S1D13704 during non-S1D13704 bus cycles. A single resistor is used to speed up the rise time of the WAIT# ($\overline{TA}$) signal when terminating the bus cycle.

The following diagram shows a typical implementation of the MCF5307 to S1D13704 interface.

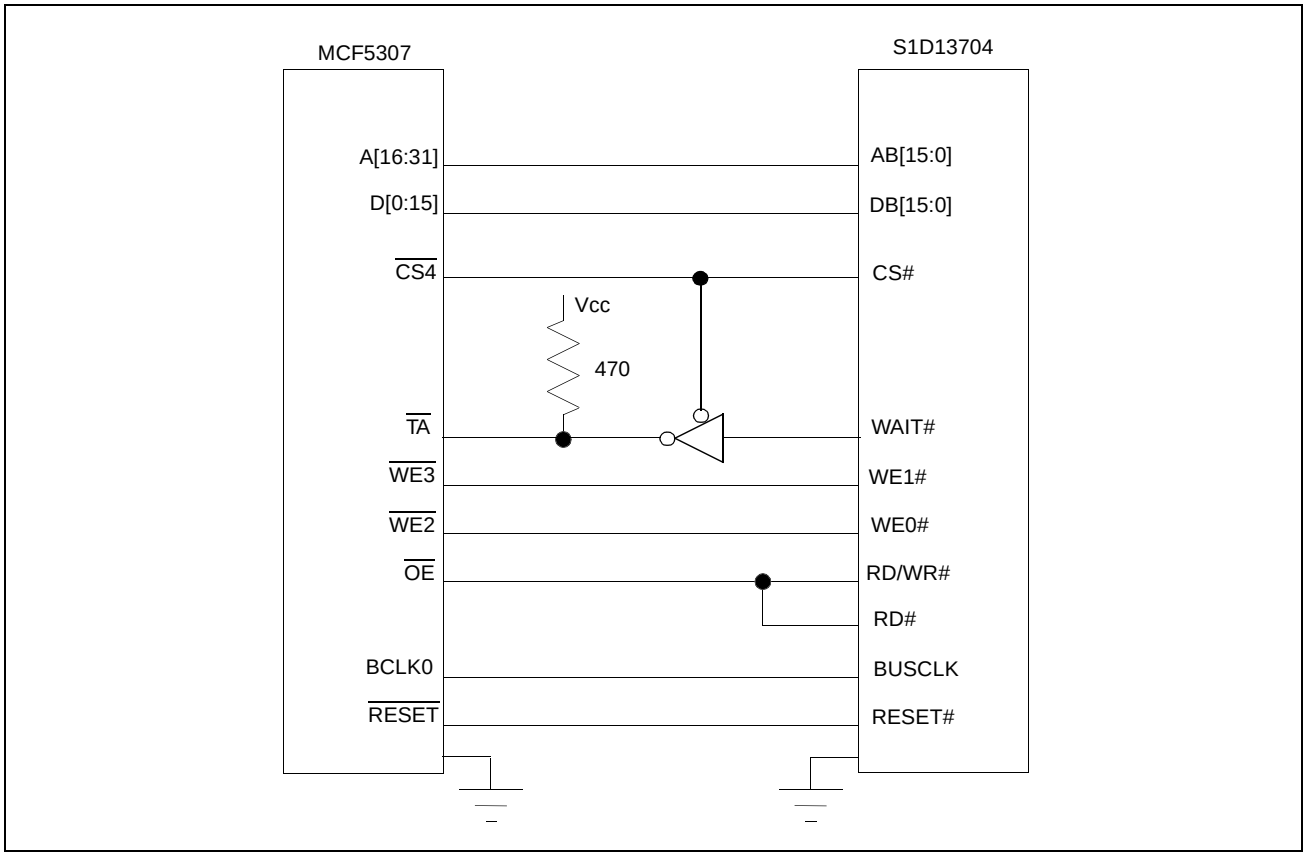


Figure 4-1: Typical Implementation of MCF5307 to S1D13704 Interface

4.2 S1D13704 Hardware Configuration

The S1D13704 uses CNF0 through CNF4 and BS# to allow selection of the bus mode and other configuration data on the rising edge of RESET#. Table 4-1: "Summary of Power-On/Reset Options" and Table 4-2: "Host Bus Interface Selection" shows the settings used for the S1D13704 in this interface.

Table 4-1: Summary of Power-On/Reset Options

S1D13704 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (0/1)	
	0	1
CNF0	See "Host Bus Selection" table below	See "Host Bus Selection" table below
CNF1		
CNF2		
CNF3	Little Endian	Big Endian
CNF4	Active low LCDPWR signal	Active high LCDPWR signal

= configuration for MFC5307 support

Table 4-2: Host Bus Interface Selection

CNF2	CNF1	CNF0	BS#	Host Bus Interface
0	0	0	X	SH-4 bus interface
0	0	1	X	SH-3 bus interface
0	1	0	X	reserved
0	1	1	X	MC68K bus interface #1, 16-bit
1	0	0	X	reserved
1	0	1	X	MC68K bus interface #2, 16-bit
1	1	0	0	reserved
1	1	0	1	reserved
1	1	1	0	Generic #1, 16-bit
1	1	1	1	Generic #2, 16-bit

= configuration for MFC5307 support

4.3 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 S1D13704. These IO chip selects have a fixed, 2M byte block size. In the example interface, chip select 4 is used to control the S1D13704. The S1D13704 only uses a 64K byte block with its 40K byte display buffer residing at the start of this 64K byte block and its internal registers occupying the last 32 bytes of this block. This 64K byte block of memory will be shadowed over the entire 2M byte space. 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 S1D13704
- C/I = 1 - disable CPU space access to the S1D13704
- SC = 1 - disable Supervisor Code space access to the S1D13704
- SD = 0 - enable Supervisor Data space access to the S1D13704
- UC = 1 - disable User Code space access to the S1D13704
- UD = 0 - enable User Data space access to the S1D13704
- 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 v2.0 display drivers are available for the S1D13704. 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 13704CFG, or by directly modifying the source. The Windows CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13704 test utilities and Windows CE v2.0 display drivers are available from your sales support contact or on the internet at http://www.erd.epson.com.

6 References

6.1 Documents

- Motorola Inc., *MCF5307 ColdFire® Integrated Microprocessor User's Manual*, Motorola Publication no. MCF5307UM/AD; available on the Internet at http://www.mot.com/SPS/HPESD/prod/coldfire/5307UM.html.
- Epson Research and Development, Inc., *S1D13704 Hardware Functional Specification*; Document Number X26A-A-002-xx.
- Epson Research and Development, Inc., *S5U13704B00C Rev. 1.0 ISA Bus Evaluation Board User Manual;* Document Number X26A-G-005-xx.
- Epson Research and Development, Inc., *S1D13704 Programming Notes and Examples*; Document Number X26A-G-002-xx.

6.2 Document Sources

- Motorola Inc.: Motorola Literature Distribution Center, (800) 441-2447.
- Motorola Website: http://www.mot.com.
- Epson Research and Development Website: http://www.erd.epson.com

7 Technical Support

7.1 EPSON LCD Controllers (S1D13704)

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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®

S1D13704 Embedded Memory Color LCD Controller

Interfacing to the Philips MIPS PR31500/PR31700 Processor

Document Number: X26A-G-012-02

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 S1D13704 Embedded Memory Color Graphics LCD Controller and the Philips MIPS PR31500/PR31700 Processor.

For further information on the S1D13704, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

For further information on the PR31500/PR31700, contact Philips or refer to the Philips website at http://www.philips.com.

For further information on the ITE IT8368E, refer to the *IT8368E PC Card / GPIO Buffer Chip Specification*.

1.1 General Description

The Philips MIPS PR31500/PR31700 processor supports up to two PC Card (PCMCIA) slots. It is through this host bus interface that the S1D13704 connects to the PR31500/PR31700 processor.

The S1D13704 can be successfully interfaced using one of two configurations:

- Direct connection to PR31500/PR31700 (see Section 2, *"Direct Connection to the Philips PR31500/PR31700"* on page 8).
- System design using one ITE IT8368E PC Card/GPIO buffer chip (see Section 3, *"System Design Using the ITE IT8368E PC Card Buffer"* on page 10).

2 Direct Connection to the Philips PR31500/PR31700

2.1 General Description

In this example implementation the S1D13704 occupies the PR31500/PR31700 PC Card slot #1.

The S1D13704 is easily interfaced to the PR31500/PR31700 with minimal additional logic. The address bus of the PR31500/PR31700 PC Card interface is multiplexed and can be demultiplexed using an advanced CMOS latch (e.g., 74ACT373). The direct connection approach makes use of the S1D13704 in its "Generic Interface #2" configuration.

The following diagram demonstrates a typical implementation of the interface.

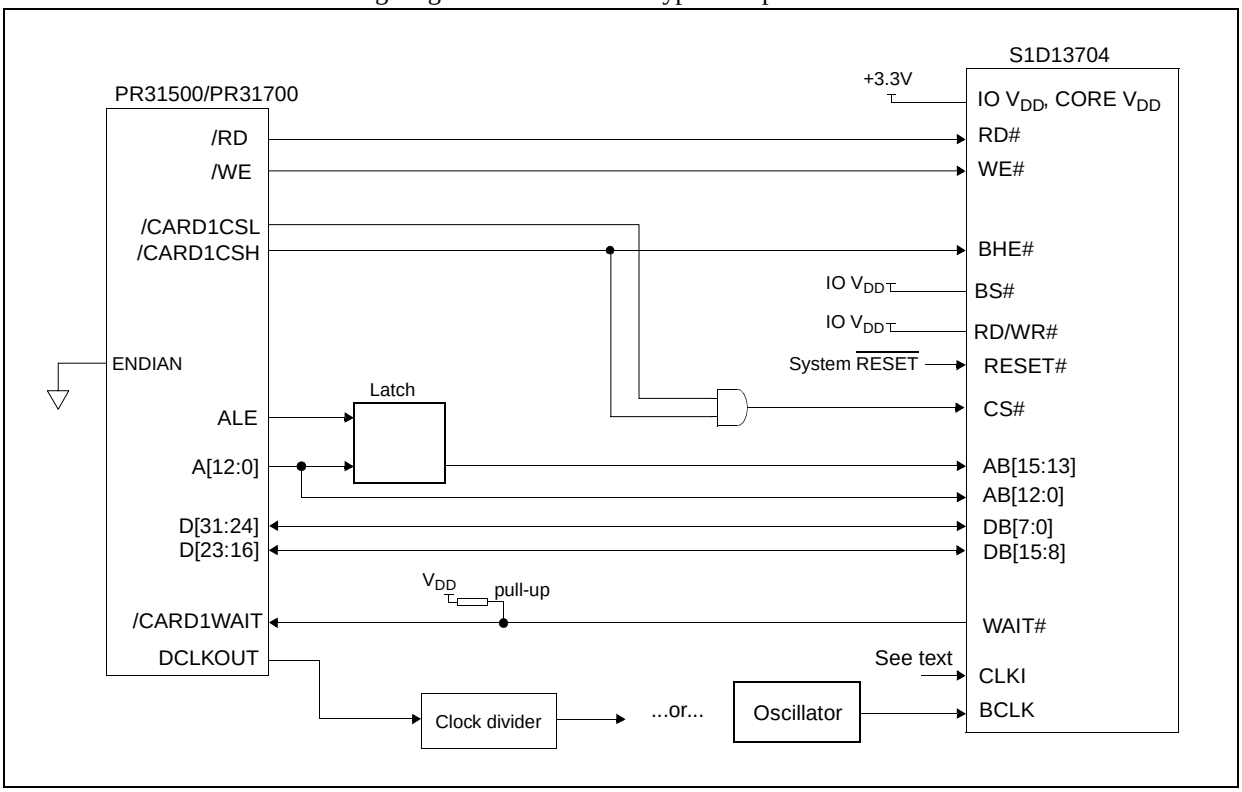


Figure 2-1: S1D13704 to PR31500/PR31700 Direct Connection

The "Generic #2" host interface control signals of the S1D13704 are asynchronous with respect to the S1D13704 bus clock. This gives the system designer full flexibility to choose the appropriate source (or sources) for CLKI and BCLK. 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 S1D13704 clock frequencies.

The S1D13704 also has internal clock dividers providing additional flexibility.

2.2 Memory Mapping and Aliasing

The S1D13704 requires an addressing space of 64K bytes. The on-chip display memory occupies the range 0 through 9FFFh. The registers occupy the range FFE0h through FFFFh. The PR31500/PR31700 demultiplexed address lines A16 and above are ignored, thus the S1D13704 is aliased 1024 times at 64K byte intervals over the 64M byte PC Card slot #1 memory space. In this example implementation, the PR31500/PR31700 control signal /CARDREG is ignored, the S1D13704 also takes up the entire PC Card slot 1 configuration space.

Note
If aliasing is undesirable, additional decoding circuitry must be added.

2.3 S1D13704 Configuration and Pin Mapping

The S1D13704 is configured at power up by latching the state of the CNF[4:0] pins. Pin BS# also plays a role in host bus interface configuration. For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

The table below shows those configuration settings relevant to the direct connection approach.

Table 2-1: S1D13704 Configuration for Direct Connection

S1D13704 Configuration Pin	Value hard wired on this pin is used to configure:	
	1 (IO V_{DD})	**0 (V_{SS})**
BS#	Generic #2	Generic #1
CNF3	Big Endian	Little Endian
CNF[2:0]	111: Generic #1 or #2	

= configuration for Philips PR31500/PR31700 host bus interface

When the S1D13704 is configured for “Generic #2” interface, the host interface pins are mapped as in the table below.

Table 2-2: S1D13704 Generic #2 Interface Pin Mapping

Pin Name	Pin Function
WE1#	BHE#
BS#	Connect to IO V_{DD}
RD/WR#	Connect to IO V_{DD}
RD#	RD#
WE0#	WE#

3 System Design Using the ITE IT8368E PC Card Buffer

If the system designer uses the ITE IT8368E PC Card and multiple-function I/O buffer, the S1D13704 can be interfaced so that it 'shares' a PC Card slot. The S1D13704 is mapped to a rarely-used 16M byte portion of the PC Card slot buffered by the IT8368E. This makes the S1D13704 virtually transparent to PC Card devices that use the same slot.

3.1 Hardware Description

The ITE8368E has been specially designed to support EPSON LCD controllers. The ITE IT8368E provides eleven Multi-Function IO pins (MFIO). Configuration registers may be used to allow these MFIO pins to provide the control signals required to implement the S1D13704 CPU interface.

The PR31500/PR31700 processor only provides addresses A[12:0], therefore devices requiring more address space must use an external device to latch A[25:13]. The IT8368E's MFIO pins can be configured to provide this latched address.

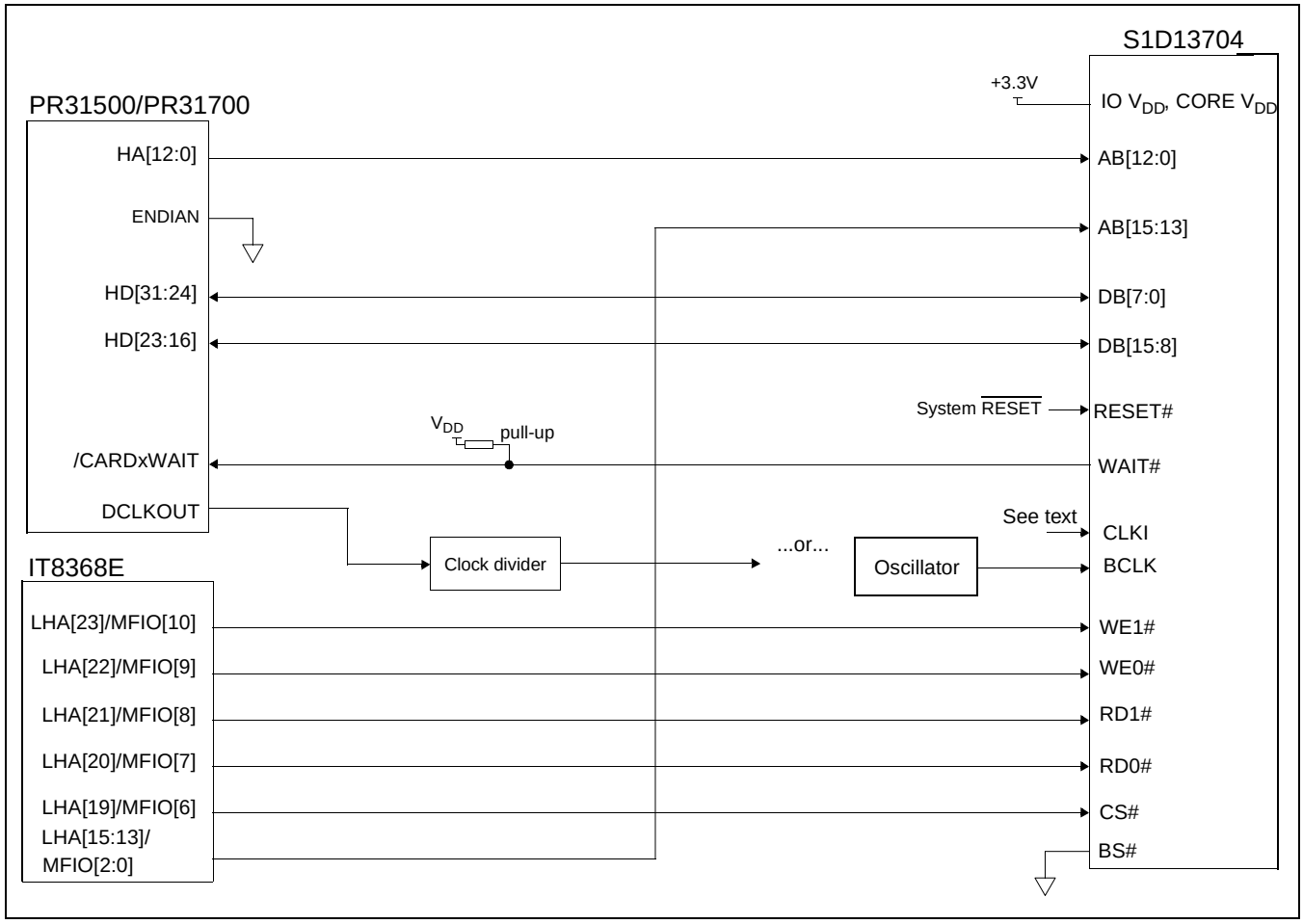


Figure 3-1: S1D13704 to PR31500/PR31700 Connection Using an IT8368E

The “Generic #1” host interface control signals of the S1D13704 are asynchronous with respect to the S1D13704 bus clock. This gives the system designer full flexibility to choose the appropriate source (or sources) for CLKI and BCLK. 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 S1D13704 clock frequencies.

The S1D13704 also has internal clock dividers providing additional flexibility.

3.2 IT8368E Configuration

The IT8368E provides eleven multi-function IO pins (MFIO). The IT8368E must have both “Fix Attribute/IO” and “VGA” modes on. When both these modes are enabled, the MFIO pins provide control signals needed by the S1D13704 host bus interface, and a 16M byte portion of the system PC Card attribute and IO space is allocated to address the S1D13704. When accessing the S1D13704 the associated card-side signals are disabled in order to avoid any conflicts.

For mapping details, refer to section 3.3: “Memory Mapping and Aliasing.” For connection details see Figure 3-1: “S1D13704 to PR31500/PR31700 Connection Using an IT8368E,” on page 10. For further information on the IT8368E, refer to the *IT8368E PC Card/GPIO Buffer Chip Specification.*

Note
When a second IT8368E is used, that circuit should not be set in VGA mode.

3.3 Memory Mapping and Aliasing

When the PR31500/PR31700 accesses the PC Card slots *without* the ITE IT8368E, its system memory is mapped as in Table 3-1:, “PR31500/PR31700 to Unbuffered PC Card Slots System Address Mapping”.

Note

Bits CARD1IOEN and CARD2IOEN need to be set in PR31500/PR31700 Memory Configuration Register 3.

Table 3-1: PR31500/PR31700 to Unbuffered PC Card Slots System Address Mapping

Philips Address	Size	Function (CARDnIOEN=0)	Function (CARDnIOEN=1)
0800 0000h	64M byte	Card 1 Attribute	Card 1 IO
0C00 0000h	64M byte	Card 2 Attribute	Card 2 IO
6400 0000h	64M byte	Card 1 Memory	
6400 0000h	64M byte	Card 2 Memory	

When the PR31500/PR31700 accesses the PC Card slots buffered through the ITE IT8368E, bits CARD1IOEN and CARD2IOEN are ignored and the attribute/IO space of the PR31500/PR31700 is divided into Attribute, I/O and S1D13704 access. Table 3-2:, “PR31500/PR31700 to PC Card Slots Address Remapping Using the IT8368E” provides all details of the Attribute/IO address reallocation by the IT8368E.

Table 3-2: PR31500/PR31700 to PC Card Slots Address Remapping Using the IT8368E

IT8368E Uses PC Card Slot #	Philips Address	Size	Function
1	0800 0000h	16M byte	Card 1 IO
	0900 0000h	16M byte	S1D13704 (aliased 256 times at 64K byte intervals)
	0A00 0000h	32M byte	Card 1 Attribute
	6400 0000h	64M byte	Card 1 Memory
2	0C00 0000h	16M byte	Card 2 IO
	0D00 0000h	16M byte	S1D13704 (aliased 256 times at 64K byte intervals)
	0E00 0000h	32M byte	Card 2 Attribute
	6800 0000h	64M byte	Card 2 Memory

3.4 S1D13704 Configuration

The S1D13704 is configured at power up by latching the state of the CNF[4:0] pins. Pin BS# also plays a role in host bus interface configuration. For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

The table below shows those configuration settings relevant to this specific interface.

Table 3-3: S1D13704 Configuration Using the IT8368E

S1D13704 Configuration Pin	Value hard wired on this pin is used to configure:	
	1 (IO V_{DD})	**0 (V_{SS})**
BS#	Generic #2	Generic #1
CNF3	Big Endian	Little Endian
CNF[2:0]	111: Generic #1 or #2	

= configuration for connection using ITE IT8368E

When the S1D13704 is configured for “Generic #1” interface, the host interface pins are mapped as in the table below.

Table 3-4: S1D13704 Generic #1 Interface Pin Mapping

Pin Name	Pin Function
WE1#	WE1#
BS#	connect to V_{SS}
RD/WR#	RD1#
RD#	RD0#
WE0#	WE0#

4 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13704. 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 1357CFG, or by directly modifying the source. The Windows CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13704 test utilities and Windows CE v2.0 display drivers are available from your sales support contact or www.erd.epson.com.

5 Technical Support

5.1 EPSON LCD Controllers (S1D13704)

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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

5.2 Philips MIPS PR31500/PR31700 Processor

Philips Semiconductors

Handheld Computing Group
4811 E. Arques Avenue
M/S 42, P.O. Box 3409
Sunnyvale, CA 94088-3409
Tel: (408) 991-2313
http://www.philips.com

5.3 ITE IT8368E

Integrated Technology Express, Inc.
Sales & Marketing Division
2710 Walsh Avenue
Santa Clara, CA 95051, USA
Tel: (408) 980-8168
Fax: (408) 980-9232
http://www.iteusa.com

THIS PAGE LEFT BLANK

EPSON®

S1D13704/5 Embedded Memory Color LCD Controller

S5U13704/5 - TMPR3912/22U CPU Module

Document Number: X00A-G-004-02

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This manual describes the interface between the S1D13704/5 LCD Controller (LCDC) and the TMPR3912/22U microprocessor as implemented on the Toshiba 3912/22 and S1D13704/5 CPU Module. This module is uS1D in conjunction with the Toshiba TX RISC Reference Platform.

For more information regarding the S1D13704 or S1D13705 refer to their respective Hardware Functional Specification, document number X26A-A-001-xx and X27A-A-001-xx respectively.

For more information regarding the TMPR3912/22U, refer to the TMPR3912/22U 32-Bit MIPS RISC Processor User's Manual. See the Toshiba website under semiconductors at http://toshiba.com/taec/nonflash/indexproducts.html.

1.1 General Description

The Toshiba TX RISC Reference Kit consists of 6 boards which include : a main board, a CPU board, a EPROM board, a FMEM board, a debug board, and an analog board. The main board acts as the motherboard for all the other add-on boards. In addition to these boards, there is an LCD module that connects to the CPU board. In order to support the add-on LCD panel that connects to the LCD module, the CPU board microprocessor must have an internal LCD controller or the CPU board must have an LCD controller on it that interfaces to the microprocessor.

For the TMPR3912/22U microprocessor, the S1D13704 or S1D13705 LCDC is used to provide support for LCD panels. The LCDC is socketed so that it can be interchanged between the S1D13704 and the S1D13705. These controllers are very similar, with the main differences being the amount of embedded display memory and the lookup-table architecture (LUT). The S1D13704 has 40K bytes of display memory and the S1D13705 has 80K bytes.

The Toshiba TMPR3912/22U processor supports two PC Card (PCMCIA) slots on the TX RISC Reference Platform. The S1D13704 or S1D13705 LCD controller uses the PC Card slot 1 to interface to the TMPR3912/22U, therefore, this slot is unavailable for use on the TX RISC Reference Platform.

2 S1D13704/5 Bus Interface

This section is summary of the bus interface modes available on the S1D13704 and S1D13705 LCDCs, and offers some detail on the Generic #2 bus mode used to implement the interface to the TMPR3912/22U.

2.1 Bus Interface Modes

The S1D13704/5 implements a general-purpose 16-bit interface to the host microprocessor, which may operate in one of several modes compatible with most of the popular embedded microprocessor families.

Bus interface mode selections are made during reset by sampling the state of the configuration pins CNF[2:0] and the BS# line. Table 5-1 in the S1D13704 or S1D13705 Hardware Functional Specification details the values needed for the configuration pins and BS# to select the desired mode.

2.2 Generic #2 Interface Mode

Generic #2 interface mode is a general and non-processor-specific interface mode on the S1D13704/5. The Generic # 2 interface mode was chosen for this interface due to its compatibility with the PC Card interface.

The interface requires the following signals:

- BUSCLK is a clock input which synchronizes transfers between the host CPU and the S1D13704/5. BUSCLK is separate from the input clock (CLKI) and is typically driven by the host CPU system clock.
- The address inputs AB0 through AB15, and the data bus DB0 through DB15, connect directly to the CPU address and data bus, respectively. On 32-bit big endian architectures such as the Power PC, the data bus would connect to the high-order data lines; on little endian hosts, or 16-bit big endian hosts, they would connect to the low-order data lines. The hardware engineer must ensure that CNF3 selects the proper endian mode upon reset.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper memory address space.
- WE1# is the high byte enable for both read and write cycles and WE0# is the enable signal for a write access. These must be generated by external decode hardware based upon the control outputs from the host CPU.
- RD# is the read enable for the S1D13704/5, to be driven low when the host CPU is reading data from the S1D13704/5. RD# must be generated by external decode hardware based upon the control outputs from the host CPU.
- WAIT# is a signal which is output from the S1D13704/5 to the host CPU that indicates when data is ready (read cycle) or accepted (write cycle) on the host bus. Since host CPU accesses to the S1D13704/5 may occur asynchronously to the display update, it is possible that contention may occur in accessing the 13704/5 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#) and Read/Write (RD/WR#) signals are not used in the bus interface for Generic #2 mode. However, BS# is used to configure the S1D13704/5 for Generic #2 mode and must be tied high (connected to IOVDD = 3.3V). RD/WR# must also be tied high.

3 TMPR3912/22U and S1D13704/5 Interface

3.1 Hardware Connections

The S1D13704/5 occupies the TMPR3912/22U's PC Card slot #1. Therefore, this slot cannot be used for other devices on the main board. The Generic # 2 bus mode of the S1D13704/5 is used to interface to this PC Card slot #1.

The S1D13704/5 is interfaced to the TMPR3912/22U with minimal glue logic. Since the address bus of the TMPR3912/22U is multiplexed, it is demultiplexed using an advanced CMOS latch (74ACT373) to obtain the higher address bits needed for the S1D13704/5.

The following diagram demonstrates the implementation of the interface.

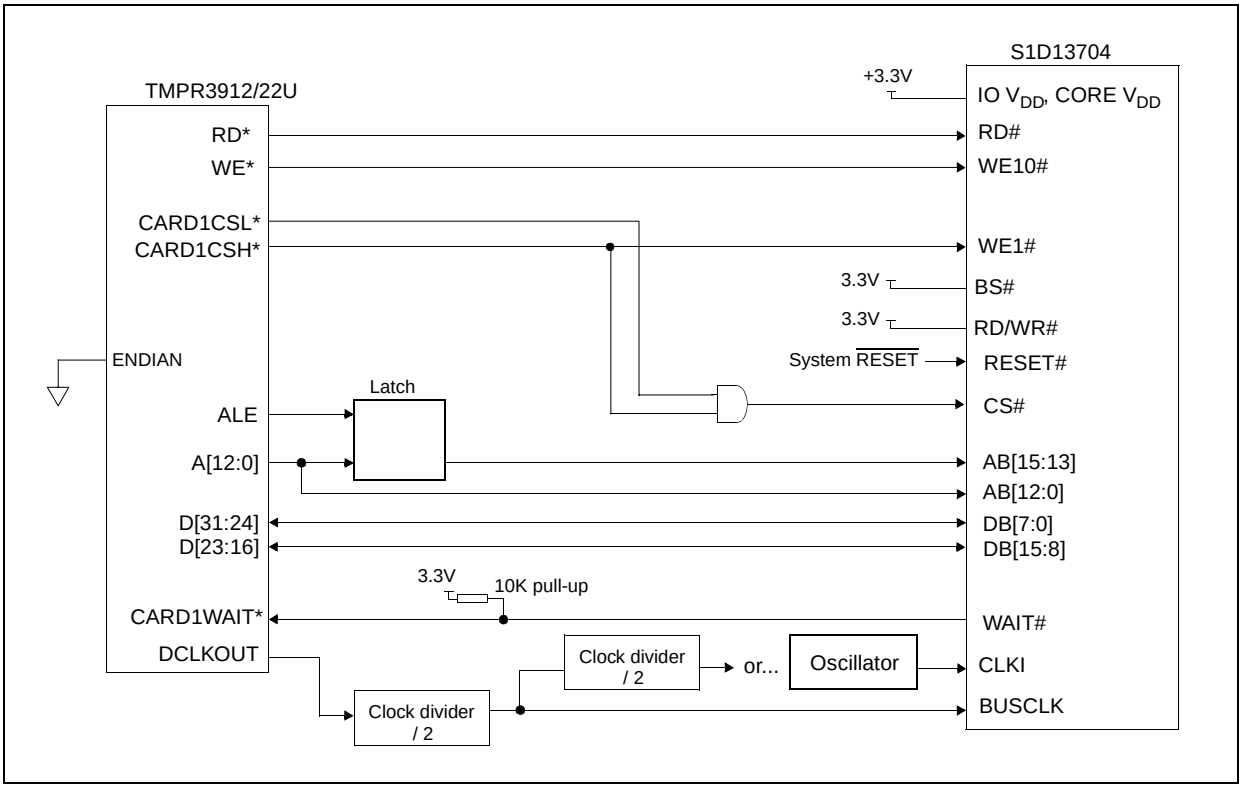


Figure 3-1: S1D13704 to TMPR3912/22U Interface

3.2 Memory Mapping and Aliasing

The S1D13704 requires an addressing space of 64K bytes while the S1D13705 requires 128K. The on-chip display memory occupies the range 0 through 9FFFh. The registers occupy the range FFE0h through FFFFh. The TMPR3912/22U demultiplexed address lines A16 and above are ignored if the S1D13704 is used, thus it is aliased 1024 times at 64K byte intervals over the 64M byte PC Card slot #1 memory space. If the S1D13705 is used, address lines A17 and above are ignored; therefore the S1D13705 is aliased 512 times at 128K byte intervals. The TMPR3912/22U control signal CARDREG# is ignored; therefore the S1D13704 also takes up the entire PC Card slot #1 configuration space.

Note
If aliasing is undesirable, additional decoding circuitry must be added.

3.3 S1D13704/5 Configuration and Pin Mapping

The S1D13704/5 host bus interface is configured at power up by latching the state of the CNF[3:0] pins. Pin BS# also plays a role in host bus interface configuration. One additional configuration pin for the S1D13704, CNF4, is also used to set the polarity of the LCDPWR signal.

The table below shows the configuration pin connections to configure the S1D13704/5 for use with the TMPR3912/22U microprocessor.

Table 3-1: S1D13704/5 Configuration for Generic #2 Bus Interface

S1D13704 Configuration Pin	Value hard wired on this pin is used to configure:	
	1 (IO V_{DD})	0 (V_{SS})
BS#	Generic #2	Generic #1
CNF3	Big Endian	Little Endian
CNF[2:0]	111: Generic #1 or #2	

= configuration for Toshiba TMPR3912/22U host bus interface

When the S1D13704/5 is configured for Generic #2 bus interface mode, the host interface pins are mapped as in the table below.

Table 3-2: S1D13704/5 Generic #2 Interface Pin Mapping

Pin Name	Pin Function
WE1#	BHE#
BS#	Connect to IO V_{DD}
RD/WR#	Connect to IO V_{DD}
RD#	RD#
WE0#	WE#

4 CPU Module Description

This section will describe the various parts of the CPU module that pertain to the S1D13704/5 LCD Controller.

4.1 Clock Signals

4.1.1 BUSCLK

Because the bus clock for the S1D13704/5 does not need to be synchronous with the bus interface control signals, a lot of flexibility is available in the choice for BUSCLK. In this CPU module, BUSCLK is a divided by two version of the SDRAM clock signal, DCLKOUT. Since DCLKOUT equals 73.728MHz, BUSCLK = 36.864MHz.

4.1.2 CLKI

The pixel clock for the S1D13704/5, CLKI, is also asynchronous with respect to the interface control signals. This clock is selected based upon panel frame rates, power vs performance budget, and maximum input frequencies. The maximum CLKI input is 25MHz if the internal CLKI/2 isn't used, and if it is used the maximum input is 50MHz.

On the CPU module, CLKI's default input is a divided by four version of DCLKOUT, which gives a CLKI = 18.432MHZ. This frequency gives good performance for 320x240 resolution panels for both portrait and landscape modes. If power saving is desired, the CLKI can be reduced by using the internal CLKI/2 and the various PCLK and MCLK dividers for portrait mode.

A socket for an external oscillator is also provided if a different frequency is required. This option is selected by positioning jumper JP8 in the 2 3 position and adding a standard 14-DIP type oscillator in the socket U10.

4.2 LCD Connectors

4.2.1 50-pin LCD Module Connector, J3

The standard connector used on Toshiba's CPU Modules to connect to the LCD module is included in this CPU module. All twelve LCD data lines, FPDAT[11:0], from the S1D13704/5, as well as the five video control signals, FPFRAME, FPSHIFT, FPLINE, DRDY, LCDPWR, are passed through this connector. Through this connector, the S1D13704/5 supports monochrome and color STN panels up to a resolution of 640x480 as well as color TFT/D-TFT up to a resolution of 640x480. All touch panel signals from the main board have also been routed through this connector.

4.2.2 Standard Epson LCD Connector, J4

A shrouded 40-pin header, J4, is also added to the CPU module to connect to LCD panels. This header is the standard LCD connector used on Epson Research and Development evaluation boards and can be used to directly connect LCD panels to the S1D13704/5 controller. All LCD signals are buffered to allow 3.3V or 5.0V logic LCD panels to be connected. Jumper, JP9, selects between these two types of panels.

A positive power supply for panels requiring a positive bias voltage is supplied to header J4, by the LCD module through the 50-pin LCD module connector, J3. No negative power supply is available on the LCD module, therefore only panels which have their own bias voltage supply, or those that use a positive supply, can be connected to J4. The LCD module can only support these panels as well.

Header, J4, and its associated buffers and components have been left unpopulated on the CPU module. These parts can be added by the user if desired.

4.3 LCD Controller

4.3.1 S1D13704 vs. S1D13705

The LCD controller used in conjunction with the TMPR3912/22U microprocessor can either be a S1D13704 or a S1D13705. If a S1D13704 is used, jumper JP7 must be set to position 1 2. This setting allows CNF4 to be configured for the S1D13704. CNF4 controls the polarity of the LCDPWR signal and can be set either high or low with jumper, JP11. If a S1D13705 is used, jumper JP7 must be set to position 2 3. This setting allows pin 45 of the LCDC to be used as address bit, AB16, which is needed on the S1D13705 to accommodate the larger display memory.

4.3.2 LCDPWR Polarity

The power supply on the LCD module used LCDON, an active low signal to turn on the supply. This signal is connected to LCDPWR. Since LCDPWR is configurable on the S1D13704 and is set active high on the S1D13705, a facility must be provided to invert this signal if it is active high so that LCDON will be the right polarity to turn on the LCD power supply. Jumper, JP10 must be set to position 1 2 if LCDPWR is active low and to position 2 3 if LCDPWR is active high.

4.3.3 S1D13704\75 Chip Select

Minimal glue logic is used on the CPU module to provide the chip select signal, CS#, for the LCDC. A simple AND gate activates the S1D13704/5 whenever the PC Card slot #1 is accessed, whether it be memory space or attribute space.

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13704 Embedded Memory Color LCD Controller

Interfacing to an 8-bit Processor

Document Number: X26A-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 environment required to provide an interface between the S1D13704 Embedded Memory LCD Controller and a generic 8-bit microprocessor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note will be 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 an 8-bit Processor

2.1 The Generic 8-bit Processor System Bus

Although the S1D13704 does not directly support an 8-bit CPU, with minimal external logic, an 8-bit interface can be achieved.

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 S1D13704, only the write, read, and wait control signals, as well as the data and address lines, need to be interfaced. Since the S1D13704 is a 16-bit device, some external logic is required.

3 S1D13704 Bus Interface

This section is a summary of the host bus interface modes available on the S1D13704 and offers some detail on the Generic #2 Host Bus Interface used to implement the interface to an 8-bit processor.

The S1D13704 provides a 16-bit interface to the host microprocessor which may operate in one of several modes compatible with most of the popular embedded microprocessor families. The bus interface mode used in this example is:

- Generic #2 (this bus interface is ISA-like and can easily be modified to support an 8-bit CPU).

3.1 Host Bus Pin Connection

The following table shows the functions of each host bus interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13704 Pin Names	Generic #2	Description
AB[15:1]	A[15:1]	Address [15:1]
AB0	A0	Address A0
DB[15:0]	D[15:0]	Data
WE1#	BHE#	Byte High Enable
CS#	External Decode	Chip Select
BCLK	BCLK	Bus Clock
BS#	n/c	Must be tied to IO V_{DD}
RD/WR#	n/c	Must be tied to IO V_{DD}
RD#	RD#	Read
WE0#	WE#	Write
WAIT#	WAIT#	
RESET#	RESET#	

For details on configuration, refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx.

3.2 Generic #2 Interface Mode

Generic #2 Host Bus Interface is a general, non-processor specific interface mode on the S1D13704 that is ideally suited to interface to an 8-bit processor bus.

The interface requires the following signals:

- BUSCLK is a clock input which synchronizes transfers between the host CPU and the S1D13704. It is separate from the input clock (CLKI) and is typically driven by the host CPU system clock. If the host CPU bus does not provide this clock, an asynchronous clock can be provided.
- The address inputs AB0 through AB15, and the data bus DB0 through DB15, connect directly to the CPU address and data bus, respectively. On 32-bit big endian architectures such as the Power PC, the data bus would connect to the high-order data lines; on little endian hosts, or 16-bit big endian hosts, they would connect to the low-order data lines. The hardware engineer must ensure that CNF3 selects the proper endian mode upon reset.

Note
In an 8-bit environment D[7:0] must also be connected to D[15:8] respectively (see Figure 4-1: “Typical Implementation of an 8-bit Processor to the S1D13704 Generic #2 Interface”)

- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper memory address space.
- BHE# (WE1#) is the high byte enable for both read and write cycles.

Note
In an 8-bit environment, this signal is driven by inverting address line A0 thus indicating that odd addresses are to be R/W on the high byte of the data bus.

- WE0# is the enable signal for a write access, to be driven low when the host CPU is writing the 13704 memory or registers.
- RD# is the read enable for the S1D13704, to be driven low when the host CPU is reading data from the S1D13704.
- WAIT# is a signal which is output from the S1D13704 to the host CPU that indicates when data is ready (read cycle) or accepted (write cycle) on the host bus. Since host CPU accesses to the S1D13704 may occur asynchronously to the display update, it is possible that contention may occur in accessing the 13704 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#) and Read/Write (RD/WR#) signals are not used in the bus interface for Generic #2 mode. However, BS# is used to configure the S1D13704 for Generic #2 mode and should be tied high (connected to IO V_{DD}). RD/WR# should also be tied high.

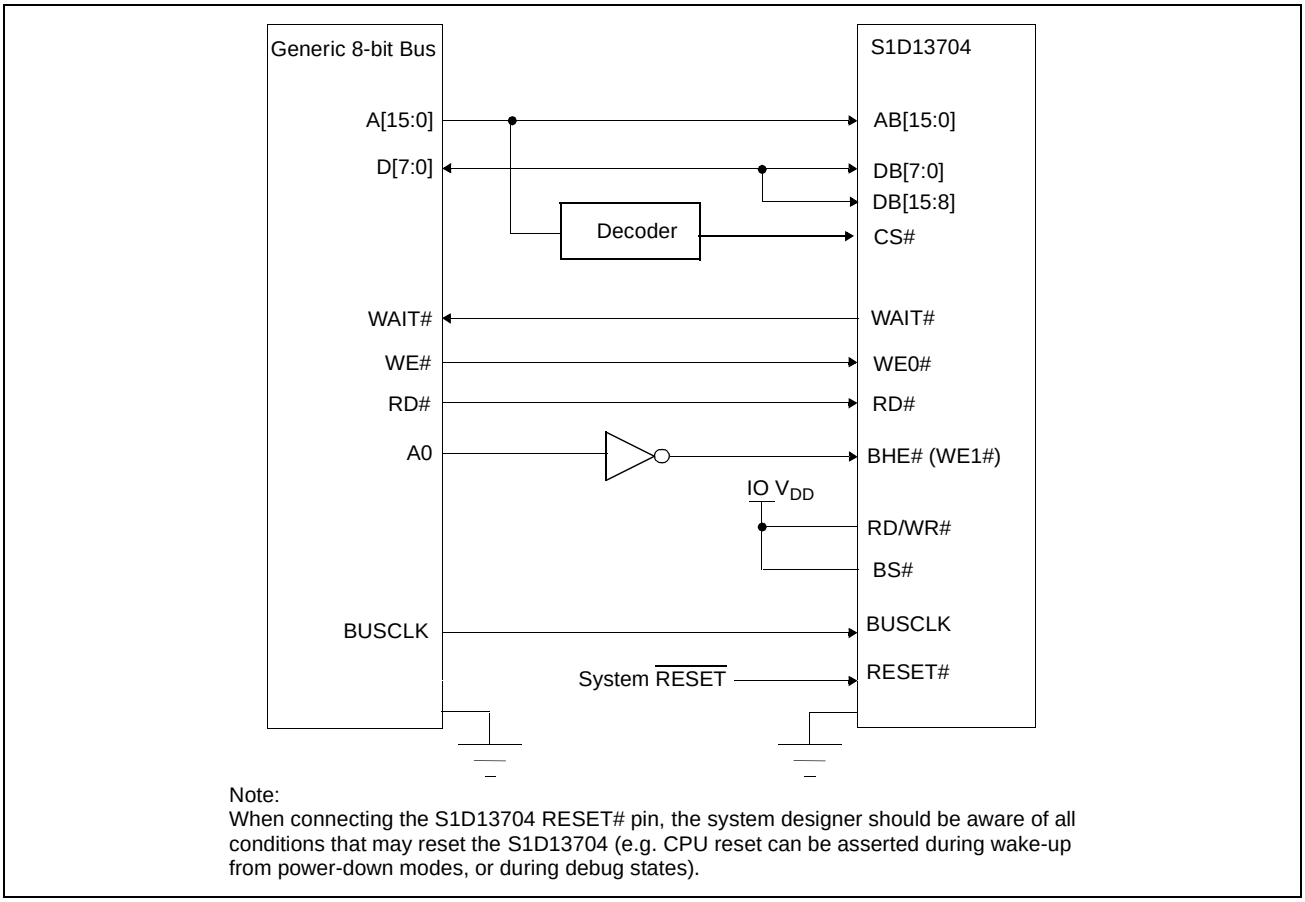
4 8-Bit Processor to S1D13704 Interface

4.1 Hardware Description

The interface between the S1D13704 and an 8-bit processor requires minimal glue logic. A decoder is used to generate the chip select for the S1D13704 based on where the S1D13704 is mapped into memory. Alternatively, if the processor supports a chip select module, it can be programmed to generate a chip select for the S1D13704 without the need of an address decoder.

An inverter inverts A0 to generate the Byte High Enable signal for the S1D13704. If the 8-bit host interface has an active high WAIT signal, it must be inverted as well.

In order to support an 8-bit microprocessor 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 S1D13704 interface.



Note:
When connecting the S1D13704 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13704 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 4-1: Typical Implementation of an 8-bit Processor to the S1D13704 Generic #2 Interface

4.2 S1D13704 Hardware Configuration

The S1D13704 uses CNF4 through CNF0 and BS# to allow selection of the bus mode and other configuration data on the rising edge of RESET#. Refer to the *S1D13704 Hardware Functional Specification*, document number X26A-A-001-xx for details.

The tables below show only those configuration settings important to the 8-bit processor interface. The endian must be selected based on the 8-bit processor used.

Table 4-1: Configuration Settings

Signal	Low	High
CNF0	See “Host Bus Selection” table below	See “Host Bus Selection” table below
CNF1		
CNF2		
CNF3	Little Endian	Big Endian
CNF4	Active low LCDPWR signal	Active high LCDPWR signal

= configuration for 8-bit processor host bus interface

Table 4-2: Host Bus Selection

CNF2	CNF1	CNF0	BS#	Host Bus Interface
1	1	1	1	Generic #2, 16-bit

= required configuration for this application.

4.3 Register/Memory Mapping

The S1D13704 needs a 64K byte block of memory to accommodate its 40K byte display buffer and its 32 byte register set. The starting memory address is located at 0000h of the 64K byte memory block while the internal registers are located in the upper 32 bytes of this memory block. (i.e. REG[0]= FFE0h).

An external decoder can be used to decode the address lines and generate a chip select for the S1D13704 whenever the selected 64K 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 S1D13704 whenever the S1D13704 memory block is accessed.

5 Software

Test utilities and Windows® CE v2.0 display drivers are available for the S1D13704. 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 13704CFG, or by directly modifying the source. The Windows CE v2.0 display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13704 test utilities and Windows CE v2.0 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., *S1D13704 Embedded Memory LCD Controller Hardware Functional Specification*; Document Number X26A-A-002-xx.
- Epson Research and Development, Inc., *S5U13704B00C Rev. 1.0 ISA Bus Evaluation Board User Manual;* Document Number X26A-G-005-xx.
- Epson Research and Development, Inc., *Programming Notes and Examples*; Document Number X26A-G-002-xx.

6.2 Document Sources

- Epson Electronics America Website: http://www.eea.epson.com.

7 Technical Support

7.1 Epson LCD/CRT Controllers (S1D13704)

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, R.O.C.
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan, R.O.C.
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

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Display Drivers & Controllers ***category:***

Click to view products by Epson ***manufacturer:***

Other Similar products are found below :

ICB2FL01G HV5812PJ-G-M904 TW8813-LB2-GR TW8811-PC2-GR MAX1839EEP+ TW9907-TA1-GR LX27901IDW SSD2828QN4 ICB2FL01GXUMA2 DLP2000FQC PAD1000YFFR S1D13746F01A600 FIN324CMLX AD8387JSVZ DLPC6421ZPC HV852K7-G HV859K7-G HV857K7-G DIO2133CT14 S1D13506F00A200 S1D13L03F00A100-40 TW2836-BA1-GR SSD2829QL9 MAX749CSA+T MAX4820EUP+T ICL7135CAI+ ICL7135CMH+D ICL7137CMH+D MAX25221BATJ/V+ S1D13748B00B100 S1D13A05B00B200 MAX3738ETG+T MAX8722CEEG+ MAX749CPA+ MAX8785AETI+ ICL7135CQI+ HV518PJ-G-M903 HV5812P-G HV5812PJ-G HV5812WG-G HV7224PG-G HV853K7-G HV860K7-G HV6810WG-G HV823LG-G HV857MG-G HV833MG-G HV857LMG-G HV850MG-G HV859MG-G