EPSON®

SED1352 Graphics LCD Controller

SED1352
TECHNICAL MANUAL

Document Number: X16B-Q-001-06

Copyright © 1997, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

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 imbedded 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.erd.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

October 1998

SED1352 GRAPHICS LCD CONTROLLER

■ DESCRIPTION

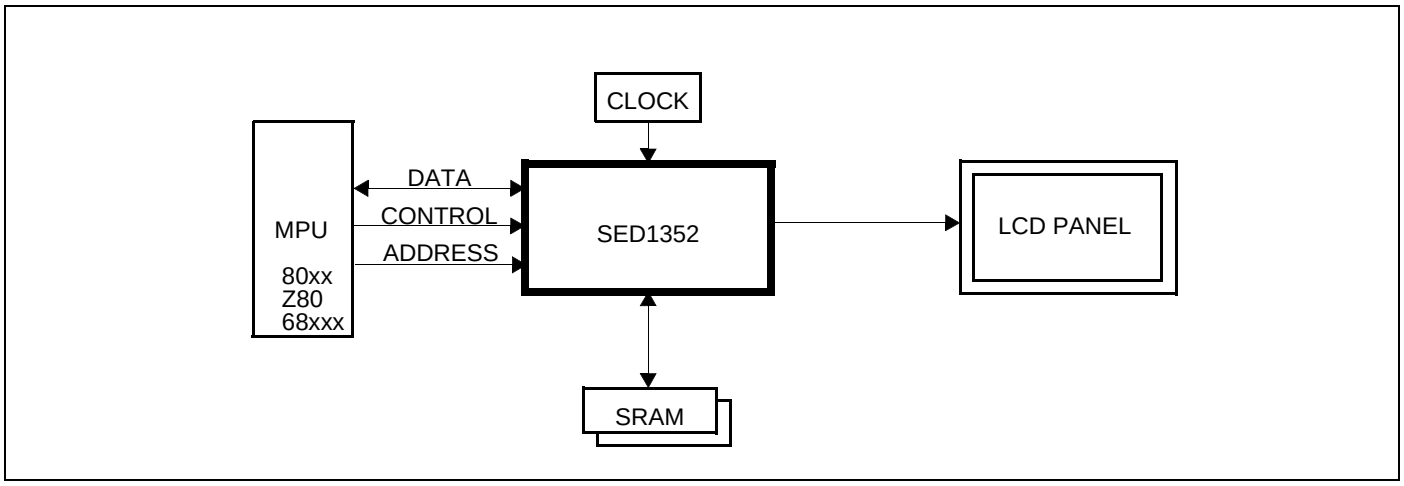
The SED1352 is a graphics display LCD controller capable of displaying a maximum of 16 levels of gray on single and dual scan Liquid Crystal Displays. A 16x4 lookup table is provided to allow remapping of the 16 possible gray shades displayed on the LCD panel. The SED1352 can interface to the MC68000 microprocessor and 8/16 bit MPUs with READY (WAIT#) signal with minimum external "glue" logic. This chip can directly control up to 128 Kbytes of static SRAM.

Optimized for cost and power savings, the SED1352 can operate from 2.7 volts to 5.5 volts and up to 25MHz.

■ FEATURES

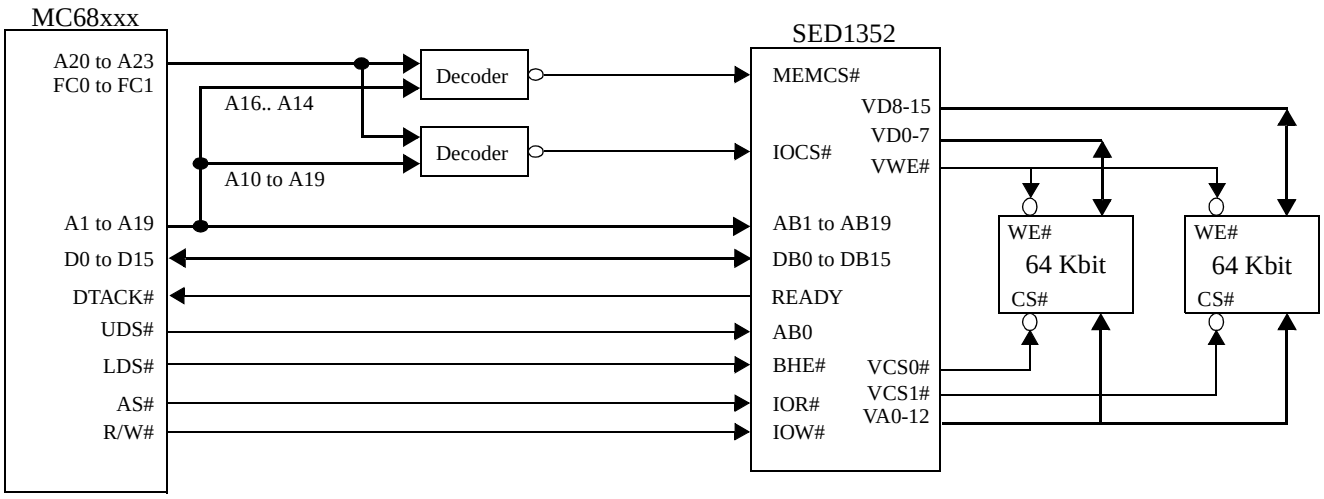
- 16-bit 16 MHz MC68000 MPU interface
- 8/16-bit MPU interface controlled by a READY (or WAIT#) signal
- option to use built-in index register or direct-mapping to access one of fifteen internal registers
- 2-terminal crystal input for internal or external crystal oscillator
- 8/16-bit SRAM interface configurations
- two software power-save modes
- low power consumption
- display modes:
  - 2 bit/pixel, 4-level gray-scale display
  - 4 bit/pixel, 16-level gray-scale display
- virtual display support
- display memory interface:
  - one 1 Mbit SRAM(64Kx16)
  - one or two 32Kbyte SRAM(s)
  - one or two 8Kbyte SRAM(s)
  - one 8Kbyte and one 32Kbyte SRAM
- LCD panel configurations:
  - single-panel, single-drive display
  - dual-panel, dual-drive display
- maximum number of vertical lines:
  - 1,024 lines (single-panel, single-drive display)
  - 2,048 lines (dual-panel, dual-drive display)
- split screen display support at single-panel mode
- package:
  - QFP5-100-S2 package (F0B)
  - or QFP15-100-STD package (F1B)

■ SYSTEM BLOCK DIAGRAM



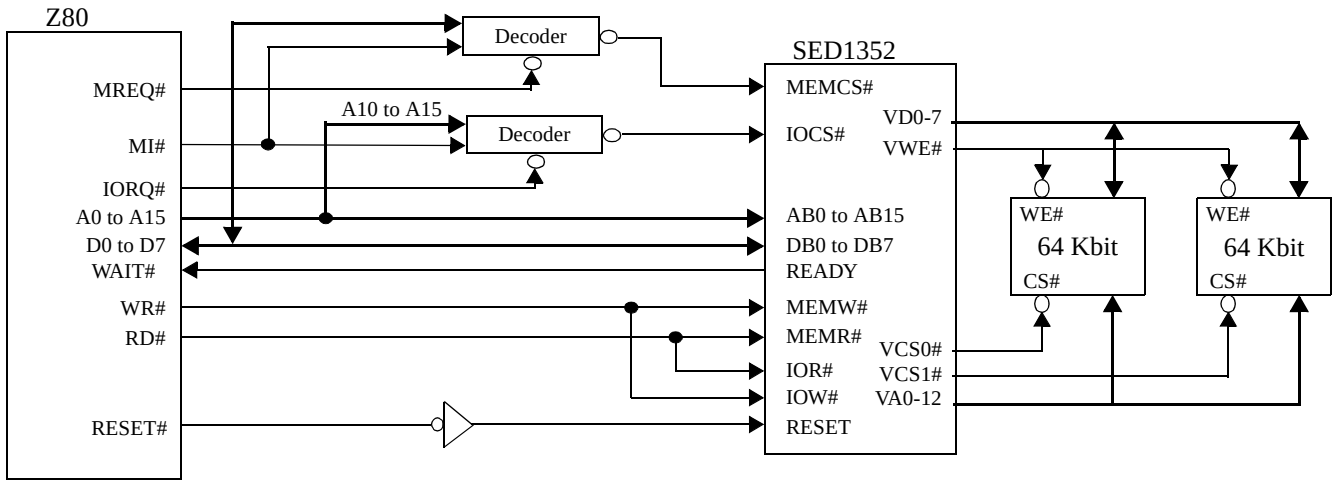
■ INTERFACE OPTIONS

Interface with 16-Bit MC68xxx MPU and 16Kbytes SRAM (2 of 8K x 8)



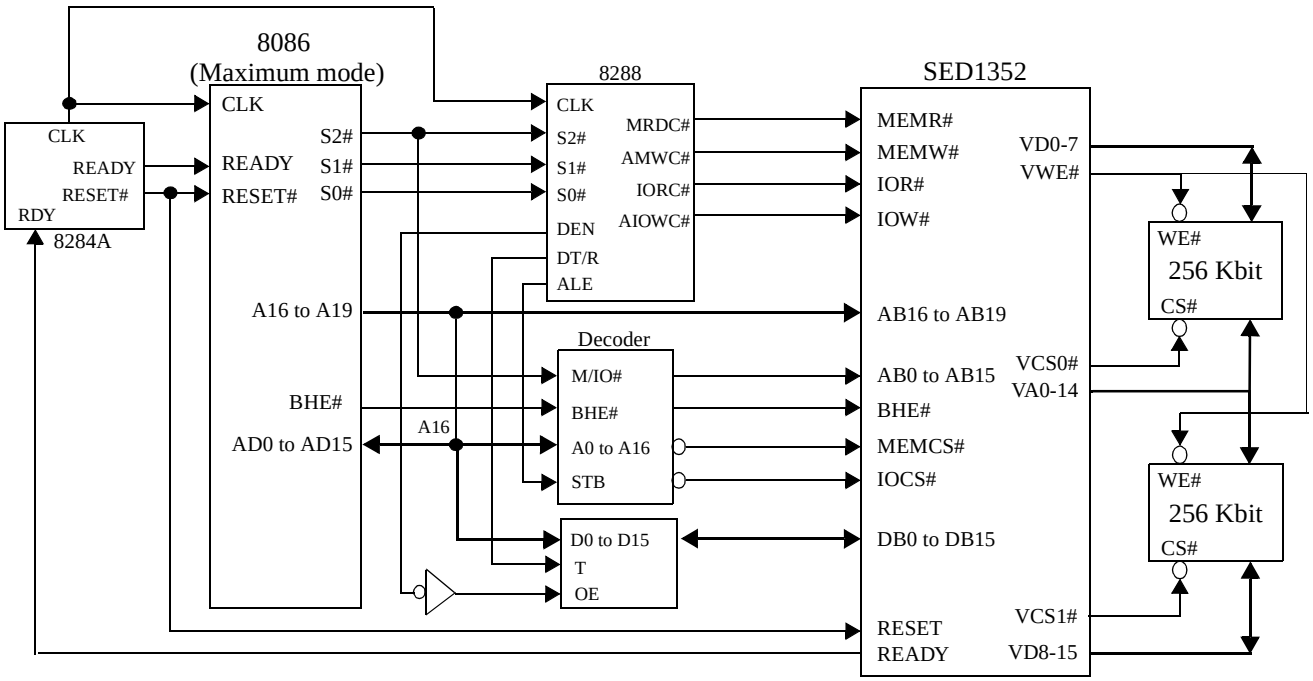
Note: Example implementation, actual may vary.

Interface with 8-Bit Z80 MPU and 16Kbytes SRAM (2 of 8K x 8)



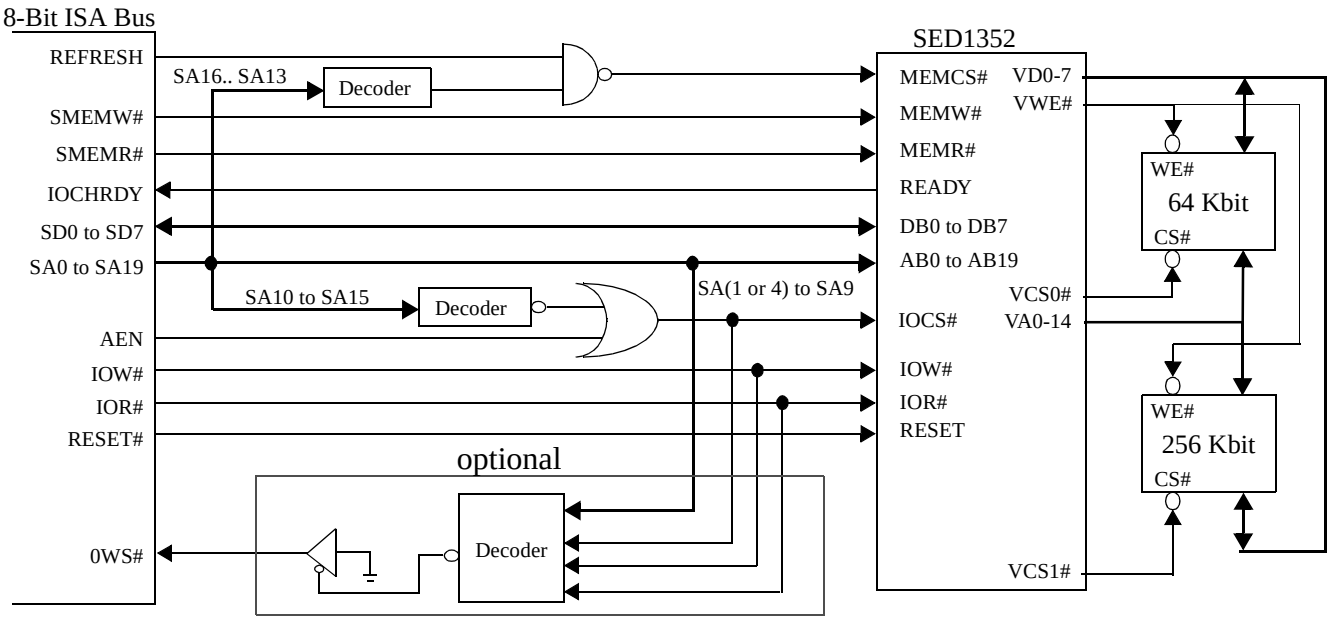
Note: Example implementation, actual may vary.

Interface with 16-Bit 8086 MPU and 64Kbytes SRAM (2 of 32K x 8)



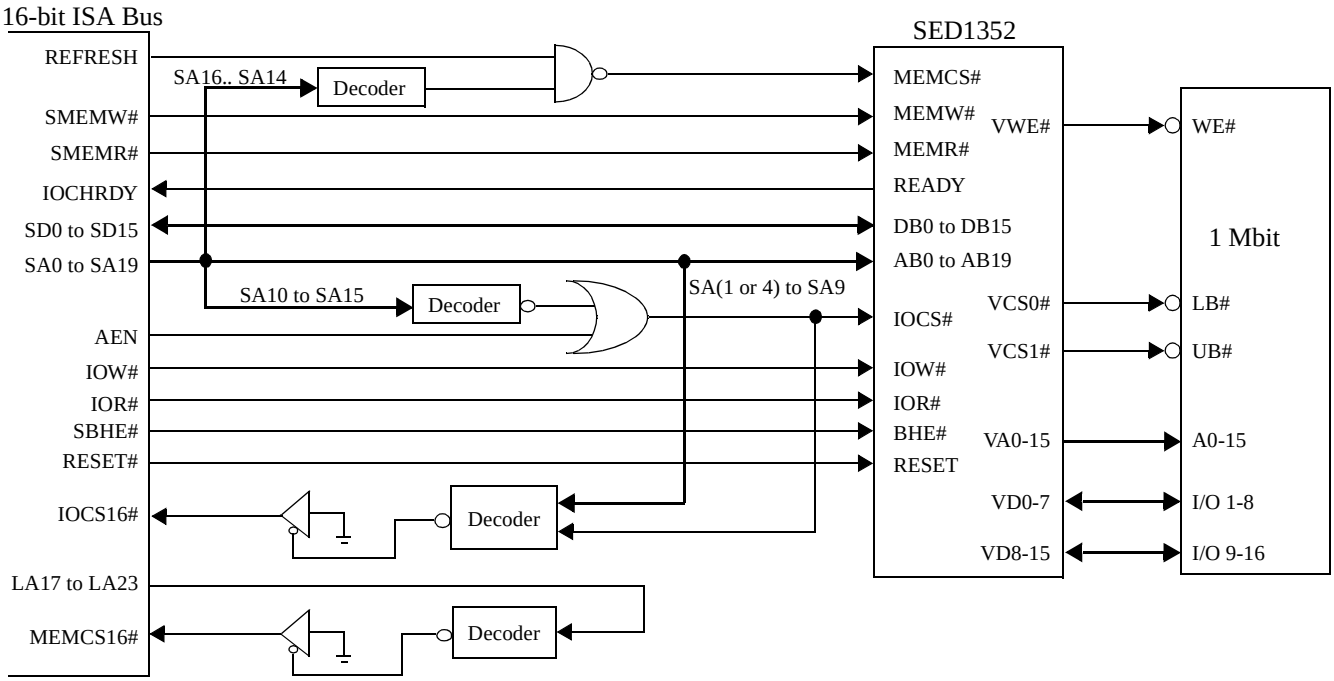
Note: Example implementation, actual may vary.

Interface with 8-Bit ISA Bus and 40Kbytes SRAM (1 of 8K x 8 and 1 of 32K x 8)



Note: Example implementation, actual may vary.

Interface with 16-Bit ISA Bus and 128Kbytes SRAM (1 of 128K x 8)

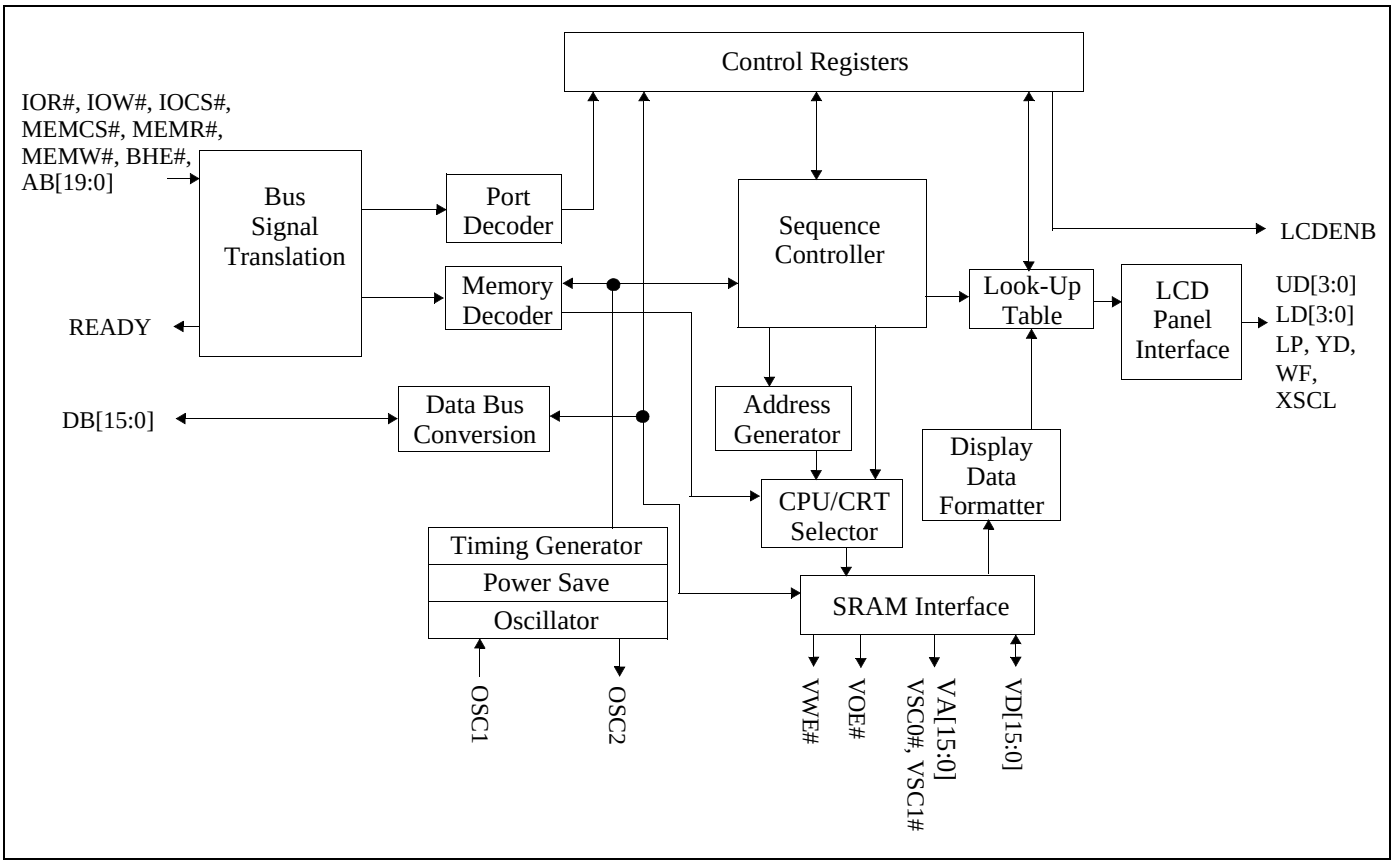


Note: Example implementation, actual may vary.

■ SUPPORTED RESOLUTIONS

Display RAM	Example Display Size				SRAM Type	CPU Interface	SRAM Interface
	4 Grays		16 Grays				
	X	Y	X	Y			
8 Kbytes	256	128	128	128	1 of 8Kx8	8-bit	8-bit
16 Kbytes	320	200	200	160	2 of 8Kx8	8-bit	8-bit/16-bit
						16-bit	16-bit
32 Kbytes	512	256	256	256	1 of 32Kx8	8-bit	8-bit
40 Kbytes	512	320	320	256	1 of 8Kx8 and 1 of 32Kx8	8-bit	8-bit
64 Kbytes	512	512	512	256	2 of 32Kx8	8-bit	8-bit/16-bit
						16-bit	16-bit
128 Kbytes	1024	512	512	512	1 of 64Kx16	16-bit	16-bit

■ BLOCK DIAGRAM



■ FUNCTIONAL BLOCK DESCRIPTIONS

Bus Signal Translation

According to configuration setting VD2, Bus Signal Translation translates MC68000 type CPU signals, or READY type MPU signals, to internal bus interface signals.

Control Registers

The fifteen internal Control and Configuration Registers are accessed by direct-mapping or by using the built-in internal index register.

Sequence Controller

The Sequence Controller generates horizontal and vertical display timings according to the configuration registers settings.

LCD Panel Interface

The LCD Interface performs frame rate modulation for passive monochrome LCD panels.

Look-Up Table

The Look-Up Table contains sixteen 4-bit wide palettes that can be configured as one 16x4 palette or four 4x4 palettes used for the re-mapping of gray-scale outputs.

Port Decoder

According to configuration settings VD1, VD12 - VD4, IOCS# and address lines AB9-1, the Port Decoder validates a given I/O cycle.

Memory Decoder

According to configuration settings VD15 - VD13, MEMCS# and address lines AB19-17, the Memory Decoder validates a given memory cycle.

Data Bus Conversion

According to configuration setting VD0, the Data Bus Conversion maps the external data bus, either 8-bit or 16-bit, into the internal odd and even data bus.

Address Generator

The Address Generator generates display refresh addresses used to access display memory.

CPU / CRT Selector

The CPU / CRT Selector accesses the display memory from the CPU or the display refresh circuitry.

Display Data Formatter

The Display Data Formatter reads the display data from the display memory and outputs the correct format for all supported LCD panel types and gray-scale selections.

Clock Inputs / Timing

Clock Inputs / Timing generates the internal master clock according to the gray-level selected and display memory interface.The master clock (MCLK) can be:

MCLK = input clock
MCLK = 1/2 input clock
MCLK = 1/4 input clock

Pixel clock = input clock.

SRAM Interface

The SRAM Interface generates the necessary signals to interface to the Display memory (SRAM).

■ DC SPECIFICATIONS

Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
V_{DD}	Supply Voltage	V_{SS} - 0.3 to + 6.5	V
V_{IN}	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_{OUT}	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
T_{STG}	Storage Temperature	-65 to 150	° C
T_{SOL}	Solder Temperature/Time	260 for 10 sec. max at lead	° C

Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{DD}	Supply Voltage	V_{SS} = 0V	2.7	3.0/3.3/5.0	5.5	V
V_{IN}	Input Voltage		V_{SS}	--	V_{DD}	V
I_{OPR}	Operating Current	f_{OSC} = 6 MHz, 16 grays		3.0/3.5/7.0		mA
T_{OPR}	Operating Temperature		-40	25	85	° C
P_{TYP}	Typical Active Power Consumption	f_{OSC} = 6 MHz, 16 grays		9.0/11.55/ 35.0		mW

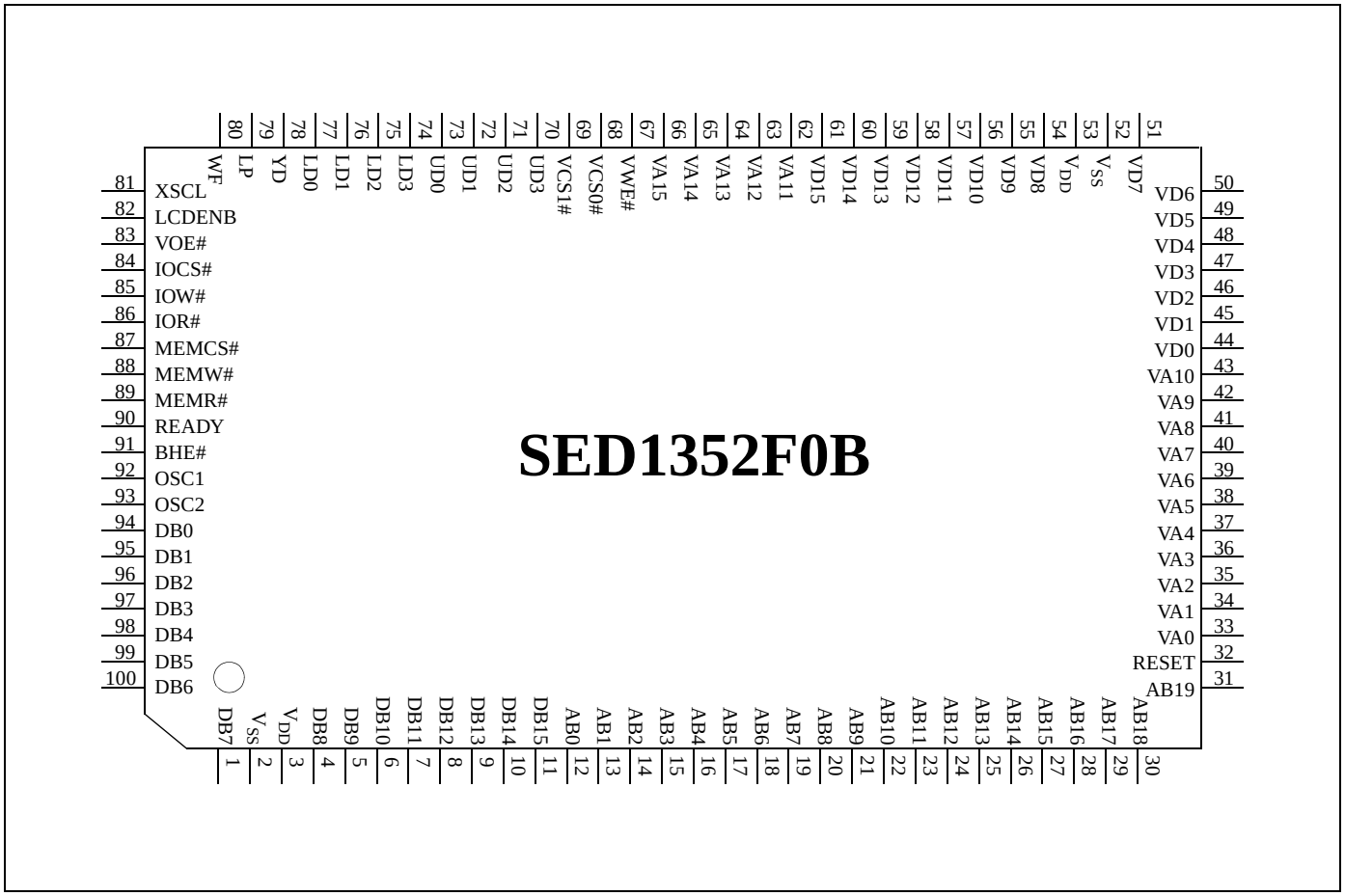
Input Specifications

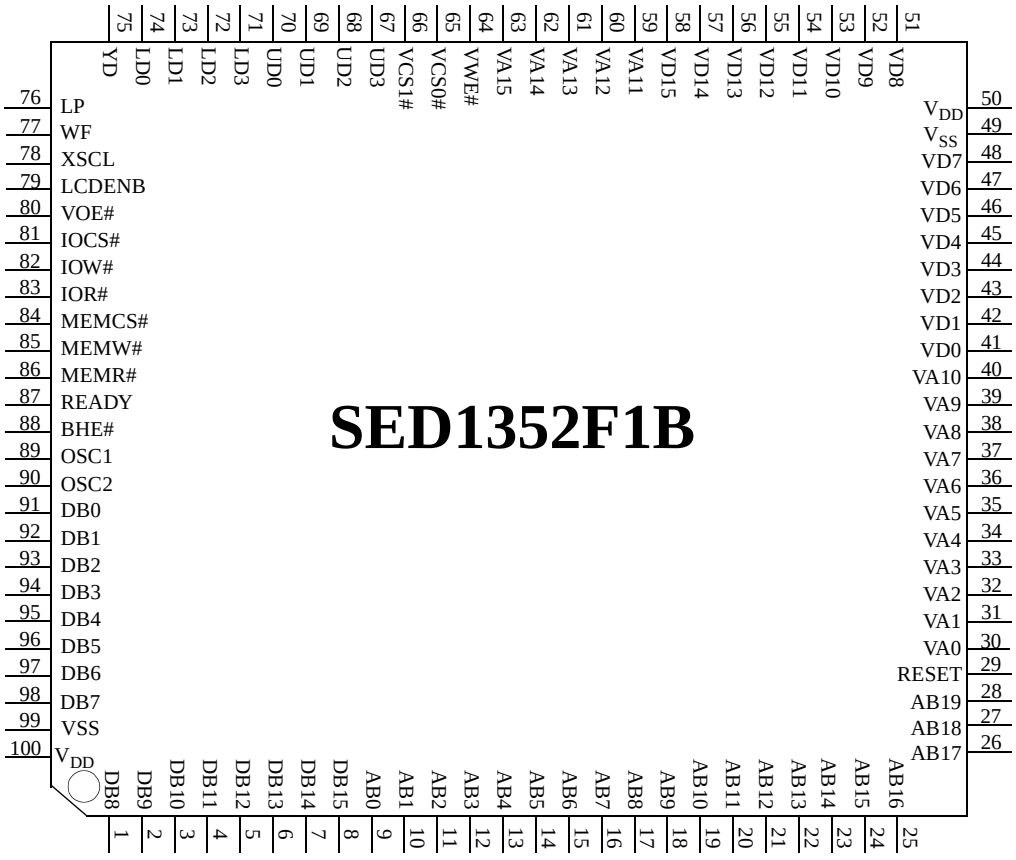
Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage	V_{DD} = 4.5V V_{DD} = 3.0V V_{DD} = 2.7V			0.8 0.6 0.5	V
V_{IH}	High Level Input Voltage	V_{DD} = 5.5V V_{DD} = 3.6V V_{DD} = 3.3V	2.0 2.5 2.3			V
V_{T+}	Positive-going Threshold	V_{DD} = 5.0 V_{DD} = 3.3 V_{DD} = 3.0			2.4 2.4 2.3	V
V_{T-}	Negative-going Threshold	V_{DD} = 5.0 V_{DD} = 3.3 V_{DD} = 3.0	0.6 0.6 0.5			V
V_H	Hysteresis Voltage	V_{DD} = 5.0 V_{DD} = 3.3 V_{DD} = 3.0	0.1 0.1 0.1			V
I_{IZ}	Input Leakage Current	--	-1		1	µA

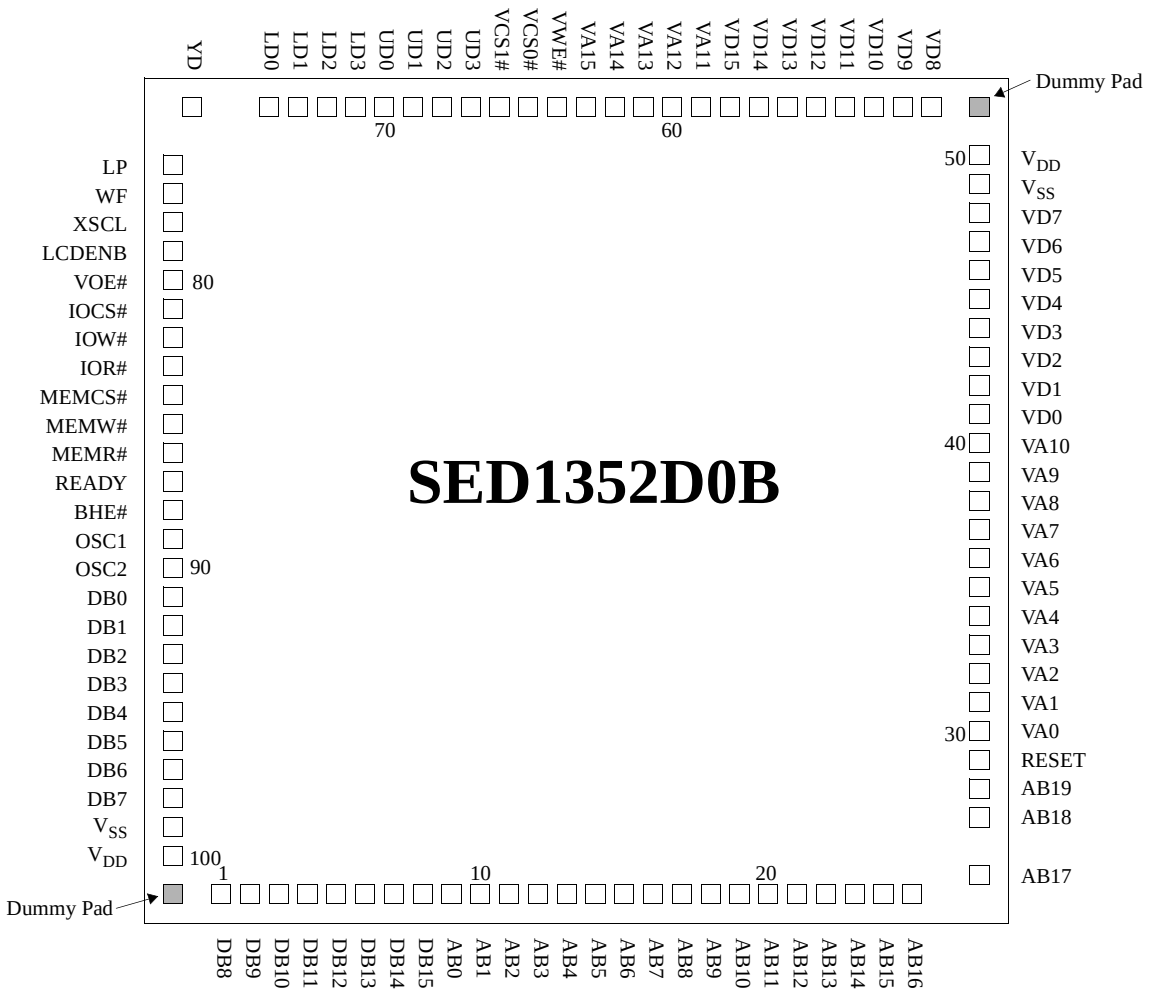
Output Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{OL} (5.0V)	Low Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OL} = 6 mA I_{OL} = 12 mA I_{OL} = 24 mA			V_{SS} + 0.4	V
V_{OL} (3.3V)	Low Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OL} = 3mA I_{OL} = 6mA I_{OL} = 12mA			V_{SS} + 0.3	V
V_{OL} (3.0V)	Low Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OL} = 3mA I_{OL} = 5 mA I_{OL} = 10mA			V_{SS} + 0.3	V
V_{OH} (5.0V)	High Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OH} = -2 mA I_{OH} = -4 mA I_{OH} = -8 mA	V_{DD}-0.4			V
V_{OH} (3.3V)	Low Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OL} = -1 mA I_{OL} = -2 mA I_{OL} = -4 mA	V_{DD}-0.3			V
V_{OH} (3.0V)	High Level Output Voltage Type 2- TS2, CO2, TS2D2 Type 3- TS3 Type 4- TS4, CO4	 I_{OH} = -1 mA I_{OH} = -1.8 mA I_{OH} = -3.5 mA	V_{DD}-0.3			V
I_{OZ}	Output Leakage Current		-1		1	µA
C_{OUT}	Output Pin Capacitance			6		pF
C_{BID}	Bidirectional Pin Capacitance			10		pF

■ SED1352 PIN OUTS




SED1352F1B
1 DB8
2 DB9
3 DB10
4 DB11
5 DB12
6 DB13
7 DB14
8 DB15
9 AB0
10 AB1
11 AB2
12 AB3
13 AB4
14 AB5
15 AB6
16 AB7
17 AB8
18 AB9
19 AB10
20 AB11
21 AB12
22 AB13
23 AB14
24 AB15
25 AB16
26 AB17
27 AB18
28 AB19
29 RESET
30 VA0
31 VA1
32 VA2
33 VA3
34 VA4
35 VA5
36 VA6
37 VA7
38 VA8
39 VA9
40 VA10
41 VD0
42 VD1
43 VD2
44 VD3
45 VD4
46 VD5
47 VD6
48 VD7
49 VSS
50 VDD
51 VD8
52 VD9
53 VD10
54 VD11
55 VD12
56 VD13
57 VD14
58 VD15
59 VA11
60 VA12
61 VA13
62 VA14
63 VA15
64 VWE#
65 VCS0#
66 VCS1#
67 UD3
68 UD2
69 UD1
70 UD0
71 LD3
72 LD2
73 LD1
74 LD0
75 YD
76 LP
77 WF
78 XSCL
79 LCDENB
80 VOE#
81 IOCS#
82 IOW#
83 IOR#
84 MEMCS#
85 MEMW#
86 MEMR#
87 READY
88 BHE#
89 OSC1
90 OSC2
91 DB0
92 DB1
93 DB2
94 DB3
95 DB4
96 DB5
97 DB6
98 DB7
99 VSS
100 VDD



Chip Size = 4.400 mm x 4.400 mm
Chip Thickness = 0.400 mm
Pad Size = 0.090 mm x 0.090 mm
Pad Pitch = 0.140 mm (Min.)

PAD Coordinates

Pad No.	Pin Name	Pad Center Coordinate	
		X	Y
1	DB8	-1.850	-2.071
2	DB9	-1.670	-2.071
3	DB10	-1.496	-2.071
4	DB11	-1.330	-2.071
5	DB12	-1.168	-2.071
6	DB13	-1.012	-2.071
7	DB14	-0.860	-2.071
8	DB15	-0.712	-2.071
9	AB0	-0.566	-2.071
10	AB1	-0.423	-2.071
11	AB2	-0.281	-2.071
12	AB3	-0.140	-2.071
13	AB4	0.000	-2.071
14	AB5	0.140	-2.071
15	AB6	0.281	-2.071
16	AB7	0.423	-2.071
17	AB8	0.566	-2.071
18	AB9	0.712	-2.071
19	AB10	0.860	-2.071
20	AB11	1.012	-2.071
21	AB12	1.168	-2.071
22	AB13	1.330	-2.071
23	AB14	1.496	-2.071
24	AB15	1.670	-2.071
25	AB16	1.850	-2.071
26	AB17	2.071	-2.021
27	AB18	2.071	-1.670
28	AB19	2.071	-1.496
29	RESET	2.071	-1.330
30	VA0	2.071	-1.168
31	VA1	2.071	-1.012
32	VA2	2.071	-0.860
33	VA3	2.071	-0.712
34	VA4	2.071	-0.566
35	VA5	2.071	-0.423
36	VA6	2.071	-0.281

Pad No.	Pin Name	Pad Center Coordinate	
		X	Y
37	VA7	2.071	-0.140
38	VA8	2.071	0.000
39	VA9	2.071	0.140
40	VA10	2.071	0.281
41	VD0	2.071	0.423
42	VD1	2.071	0.566
43	VD2	2.071	0.712
44	VD3	2.071	0.860
45	VD4	2.071	1.012
46	VD5	2.071	1.168
47	VD6	2.071	1.330
48	VD7	2.071	1.496
49	V_{SS}	2.071	1.670
50	V_{DD}	2.071	1.850
51	VD8	1.850	2.071
52	VD9	1.670	2.071
53	VD10	1.496	2.071
54	VD11	1.330	2.071
55	VD12	1.168	2.071
56	VD13	1.012	2.071
57	VD14	0.860	2.071
58	VD15	0.712	2.071
59	VA11	0.566	2.071
60	VA12	0.423	2.071
61	VA13	0.281	2.071
62	VA14	0.140	2.071
63	VA15	0.000	2.071
64	VWE#	-0.140	2.071
65	VCS0#	-0.281	2.071
66	VCS1#	-0.423	2.071
67	UD3	-0.566	2.071
68	UD2	-0.712	2.071
69	UD1	-0.860	2.071
70	UD0	-1.012	2.071
71	LD3	-1.168	2.071
72	LD2	-1.330	2.071

Pad No.	Pin Name	Pad Center Coordinate	
		X	Y
73	LD1	-1.496	2.071
74	LD0	-1.670	2.071
75	YD	-2.021	2.071
76	LP	-2.071	1.850
77	WF	-2.071	1.670
78	XSCL	-2.071	1.496
79	LCDENB	-2.071	1.330
80	VOE#	-2.071	1.168
81	IOCS#	-2.071	1.012
82	IOW#	-2.071	0.860
83	IOR#	-2.071	0.712
84	MEMCS#	-2.071	0.566
85	MEMW#	-2.071	0.423
86	MEMR#	-2.071	0.281
87	READY	-2.071	0.140

Pad No.	Pin Name	Pad Center Coordinate	
		X	Y
88	BHE#	-2.071	0.000
89	OSC1	-2.071	-0.140
90	OSC2	-2.071	-0.281
91	DB0	-2.071	-0.423
92	DB1	-2.071	-0.566
93	DB2	-2.071	-0.712
94	DB3	-2.071	-0.860
95	DB4	-2.071	-1.012
96	DB5	-2.071	-1.168
97	DB6	-2.071	-1.330
98	DB7	-2.071	-1.496
99	V_{SS}	-2.071	-1.670
100	V_{DD}	-2.071	-1.850
101	Dummy Pad	2.071	2.071
102	Dummy Pad	-2.071	-2.071

■ PIN DESCRIPTION

Key

A = Analog
I = Input
O = Output
I/O = Bidirectional
P = Power

Bus Interface

Pin Name	Type	F0B Pin #	F1B Pin # D0B Pad #	Description
DB0-DB15	I/O	94 - 100, 1, 4 -11	91 - 98, 1 - 8	These pins are connected to the system data bus. In 8-bit bus mode, DB8-DB15 must be tied to V_{DD}.
AB0	I	12	9	In MC68000 MPU interface, this pin is connected to the Upper Data Strobe (UDS#) pin of MC68000. In other bus interfaces, this pin is connected to the system address bus.
AB1-AB19	I	13 - 31	10 - 28	These pins are connected to the system address bus.
BHE#	I	91	88	In MC68000 MPU interface, this pin is connected to the Lower Data Strobe (LDS#) pin of MC68000. In other bus interfaces, this pin is the Bus High Enable input for use with 16-bit system. In 8-bit bus mode, tie BHE# input to V_{DD}.
IOCS#	I	84	81	Active low input to select one of fifteen internal registers.
IOW#	I	85	82	In MC68000 MPU interface, this pin is connected to the R/W# pin of MC68000. This input pin will define whether the data transfer is a read (active high) or write (active low) cycle. In other bus interfaces, this is the active low input to write data into an internal register.
IOR#	I	86	83	In MC68000 MPU interface, this pin is connected to the AS# pin of MC68000. This input pin will indicate a valid address is available on the address bus. In other bus interfaces, this is the active low input to read data from an internal register.
MEMCS#	I	87	84	Active low input to indicate the attempt to access the display memory.
MEMW#	I	88	85	Active low input to write data to the display memory. This pin should be tied to V_{DD} in an MC68000 MPU interface.
MEMR#	I	89	86	Active low input to read data from the display memory. This pin should be tied to V_{DD} in an MC68000 MPU interface.
READY	O	90	87	For MC68000 MPU interface, this pin is connected to the DTACK# pin of MC68000 and will be driven low when ever a data transfer is complete. In other bus interfaces, this output is driven low to force the system to insert wait states when needed. READY is placed in a high-impedance (Hi-Z) state after the transfer is completed.
RESET	I	32	29	Active high input to force all signals to their inactive states.

Display Memory Interface

Pin Name	Type	F0B Pin #	F1B Pin # D0B Pad #	Description
VD0-VD15	I/O	44 - 51, 54 - 61	41 - 48, 51 - 58	These pins are connected to the display memory data bus. For 16-bit interface, VD0-VD7 are connected to the display memory data bus of even byte addresses and VD8-VD15 are connected to the display memory data bus of odd byte addresses. The output drivers of these pins are tri-stated when RESET is high. On the falling edge of RESET the values of VD0-VD15 are latched into the chip to configure various hardware options. VD0-VD15 each have an internal pull-down resistor
VA0-VA15	O	33 - 43, 62 - 66	30 - 40 59, 63	These pins are connected to the display memory address bus.
VCS1#	O	69	66	Active low chip-select output to the second or odd byte address SRAM.
VCS0#	O	68	65	Active low chip-select output to the first or even byte address SRAM.
VWE#	O	67	64	Active low output used for writing data to the display memory. This pin is connected to the WE# input of the SRAMs.
VOE#	O	83	80	Active low output to enable reading of data from the display memory. This pin is connected to the OE# input of the SRAMs.

LCD Interface

Pin Name	FPDI-1™ Pin Name[a]	Type	F0B Pin #	F1B Pin # D0B Pad #	Description
UD3-UD0	UD3-UD0	O	70 - 73	67 - 70	Upper panel display data for dual panel mode. For single panel mode, these bits are the most significant 4 bits of the 8 bits output data to the panel (PD[4:7]). For 4-bit single panel mode, these bits are the 4 bits of output data to the panel.
LD3-LD0	LD3-LD0	O	74 - 77	71 - 74	Lower panel display data for dual panel mode. For 8-bit single panel mode, these bits are the least significant 4 bits of the 8 bits output data to the panel (PD[0:3]). For 4-bit single panels, these bits are driven 0 (low state).
XSCL	FPSHIFT	O	81	78	Display data shift clock. Data is shifted into the LCD X-drivers on the falling edge of this signal.
LP	FPLINE	O	79	76	Display data latch clock. The falling edge of this signal is used to latch a row of display data in the LCD X-drivers and to turn on the row driver (Y driver).
WF	MOD	O	80	77	LCD backplane BIAS signal. This output toggles once every n LP periods, as programmed in AUX[5]
YD	FPFRAME	O	78	75	Vertical scanning start pulse. A logic ‘1’ on this signal, sampled by the LCD module on the falling edge of LP, is used by the panel row driver (Y driver) to indicate the start of the vertical frame.
LCDENB		O	82	79	LCD enable signal output. It can be used externally to turn off the panel supply voltage and backlight.

[a] VESA Flat Panel Display Interface Standard (FPDI-1™)

Clock Inputs

Pin Name	Type	F0B Pin #	F1B Pin # D0B Pad #	Description
OSC1	I	92	89	This pin, along with OSC2 is the 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin is the clock input.
OSC2	O	93	90	This pin, along with OSC1 is the 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin should be left unconnected.

Power Supply

Pin Name	Type	F0B Pin #	F1B Pin # D0B Pad #	Description
V_{DD}	P	3, 53	50, 100	Voltage supply.
V_{SS}	P	2, 52	49, 99	Voltage ground.

■ SUMMARY OF CONFIGURATION OPTIONS

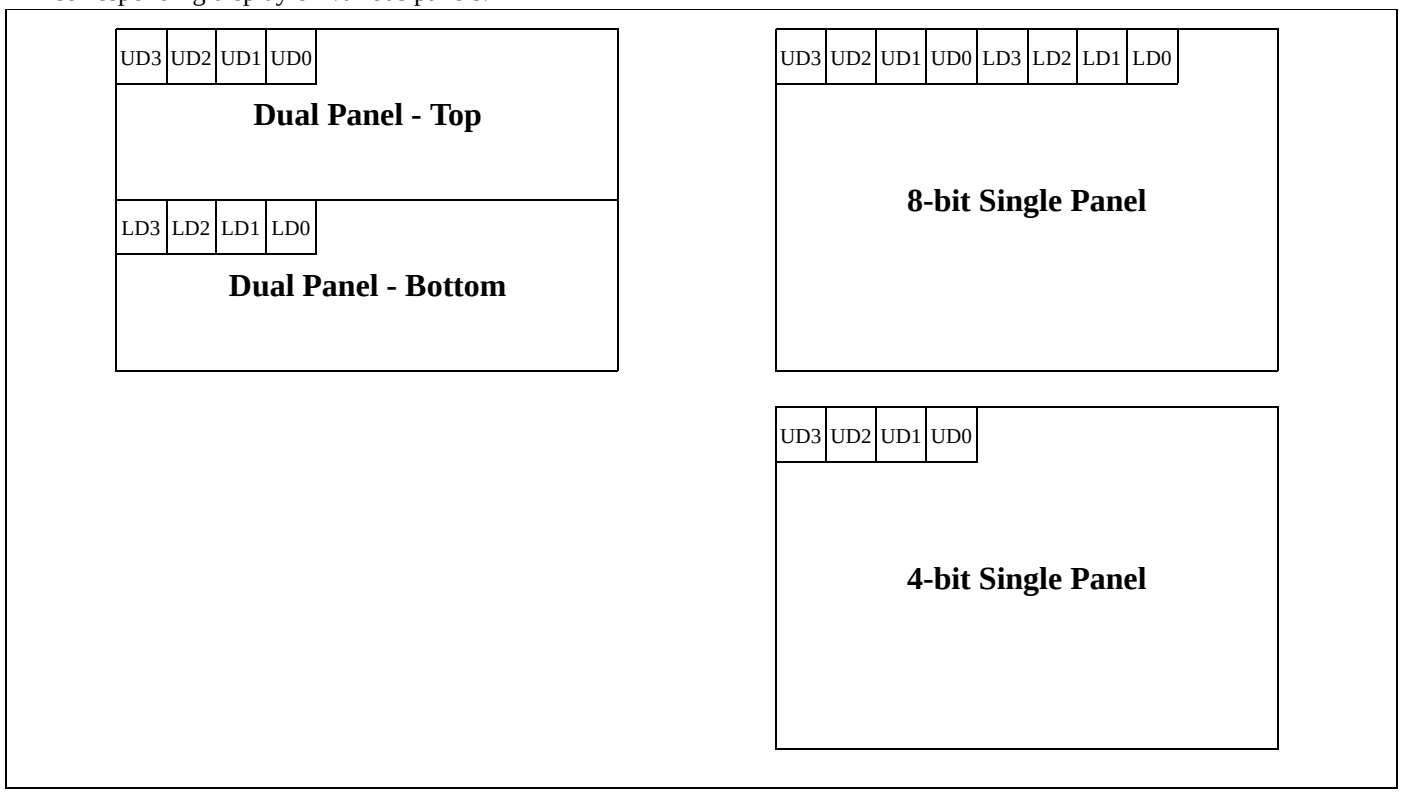
Pin Name	value on this pin at falling edge of RESET is used to configure: (1/0)	
	1	**0**
VD0	16-bit host bus interface	8-bit host bus interface
VD1	Use direct-mapping for I/O accesses	Use internal index register for I/O accesses
VD2	MC68000 MPU interface	MPU / Bus interface with memory accesses controlled by a READY (WAIT#) signal
VD3	Swap of high and low data bytes in 16-bit bus interface	No byte swap of high and low data bytes in 16-bit bus interface
VD4-VD12	Select I/O mapping address bits [1:9]. These nine bits are latched on power-up and are compared to the MPU address bits [1-9]. A valid I/O cycle combined with a valid address will enable the internal I/O decoder. Therefore, both types of I/O mapping are limited to even address boundaries to determine either the absolute or indexed I/O address of the first register. Note that a “valid I/O cycle” includes IOCS# being toggled low. In direct mapping, the base I/O address is selected by VD7-VD12. In indexing, the base I/O address is selected by VD4-VD12.	
VD13-VD15	Select memory mapping address bits [1:3]. These three bits are latched on power-up and are compared to the MPU address bits [17-19]. A valid memory cycle combined with a valid address will enable the internal memory decoder. As only the three most significant bits of the address are compared, the maximum amount of memory supported is 128K bytes. Note that a “valid memory cycle” includes MEMCS# being toggled low. If 128K byte memory is used, it must be mapped at an even address so all 128K bytes is available without a change in state on A17, as this would invalidate the internal compare logic.	

Example: If an ISA bus (no byte swap) with memory segment "A" and I/O location 300h are used, the corresponding settings of VD15-VD0 would be:

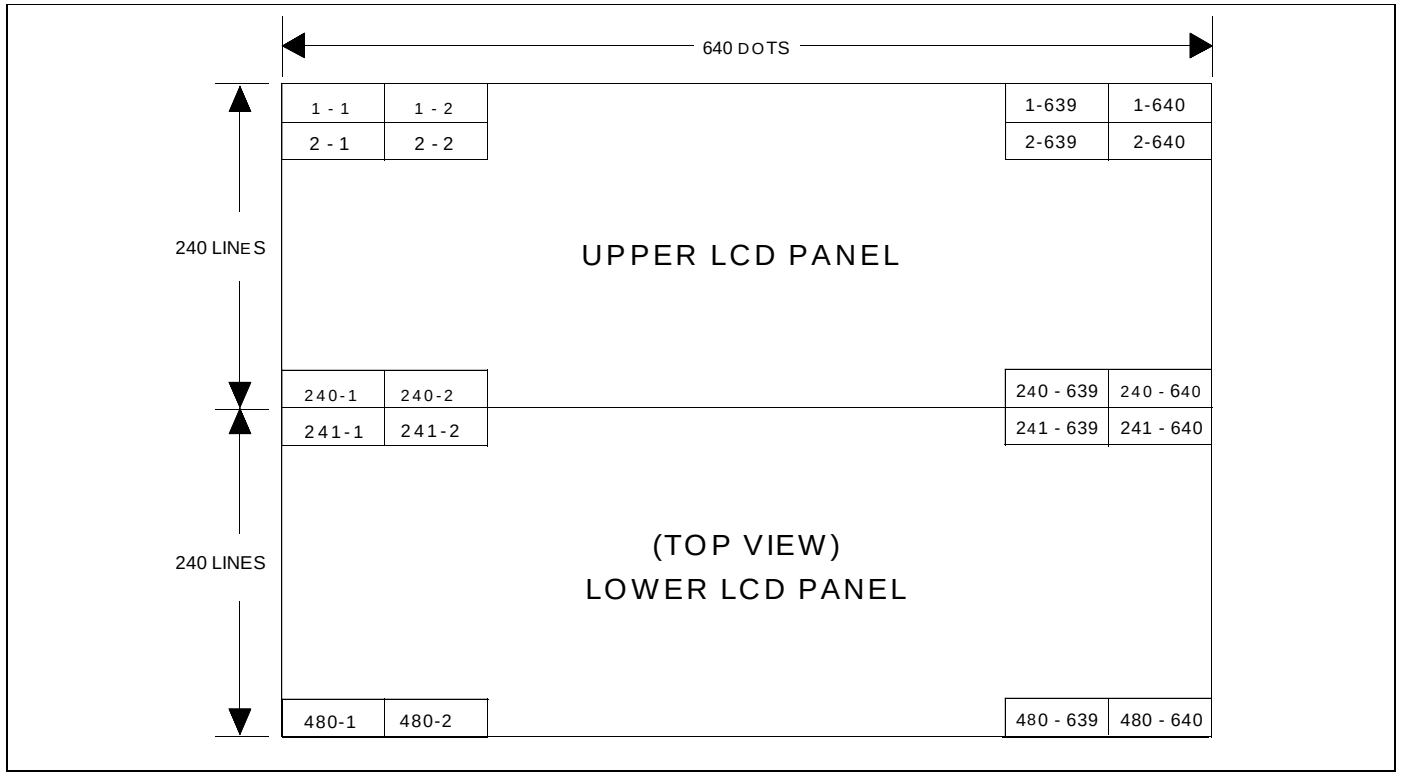
	8-Bit ISA Bus		16-Bit ISA Bus	
Pin Name	**Index Register**	**Direct Mapping**	**Index Register**	**Direct Mapping**
VD0	0	0	1	1
VD1	0	1	0	1
VD2	0	0	0	0
VD3	0	0	0	0
VD12-VD4	11 0000 000	11 0000 xxx	11 0000 000	11 0000 xxx
VD15-VD13	101	101	101	101

Where x = don't care; 1 = connected to pull-up resistor; 0 = no pull-up resistor.

Illustrated below is the display data which is output from the UD0 to UD3 signal pins and the corresponding display on various panels:

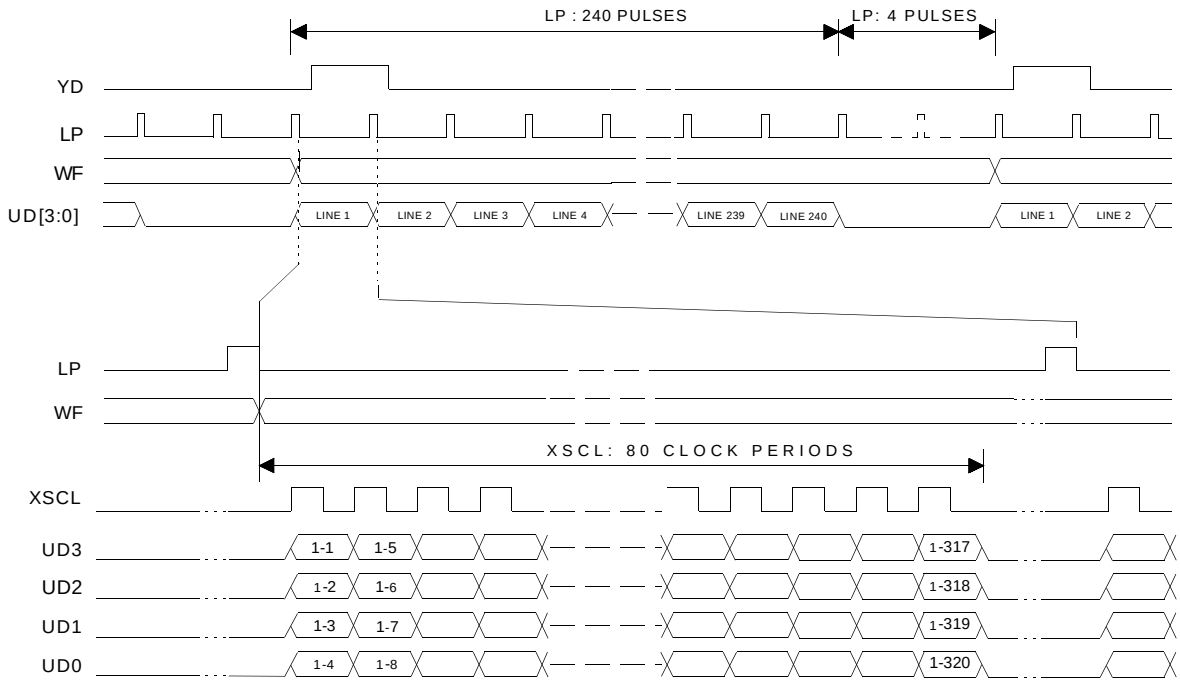


■ LCD PANEL PIXELS



■ MONOCHROME PASSIVE STN LCD PANEL INTERFACE

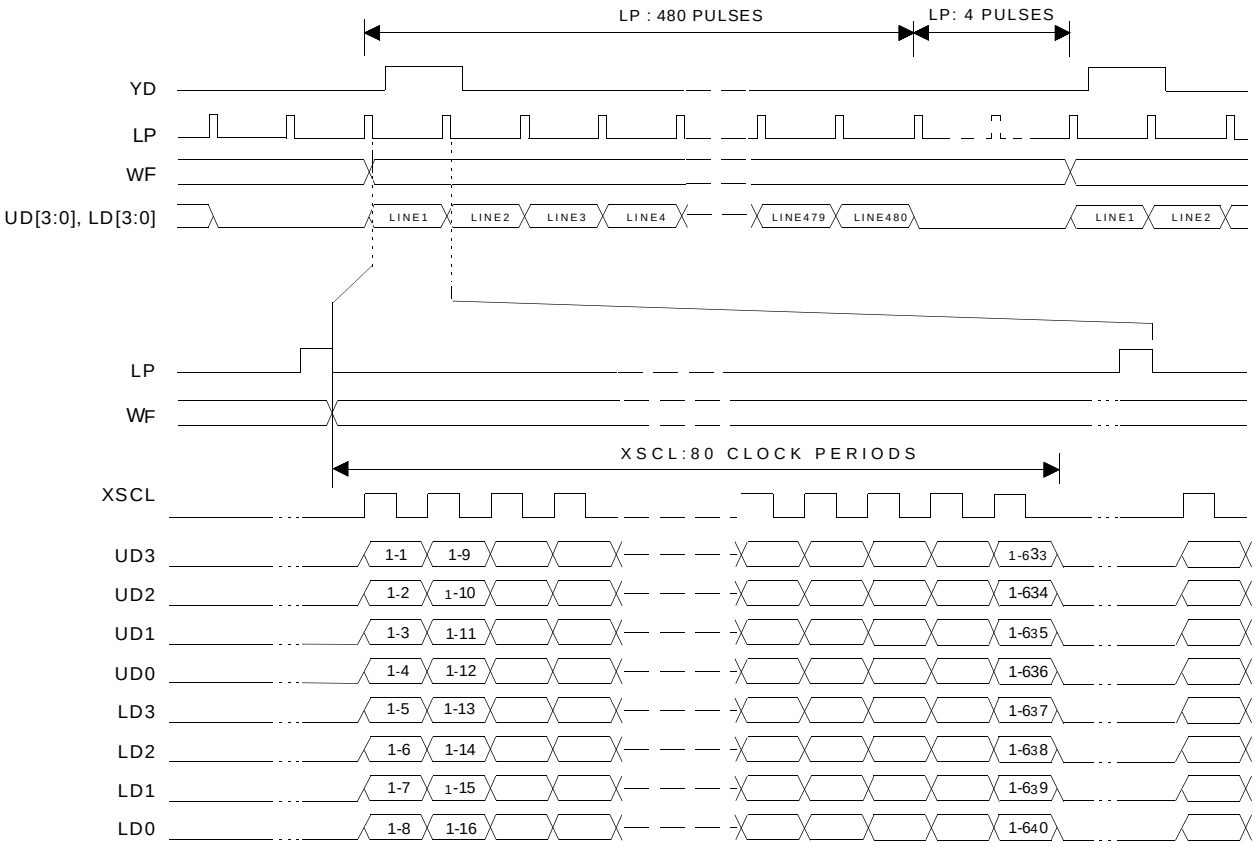
4-BIT SINGLE PANEL



Example Timing for a 320x240 single panel

■ MONOCHROME PASSIVE STN LCD PANEL INTERFACE

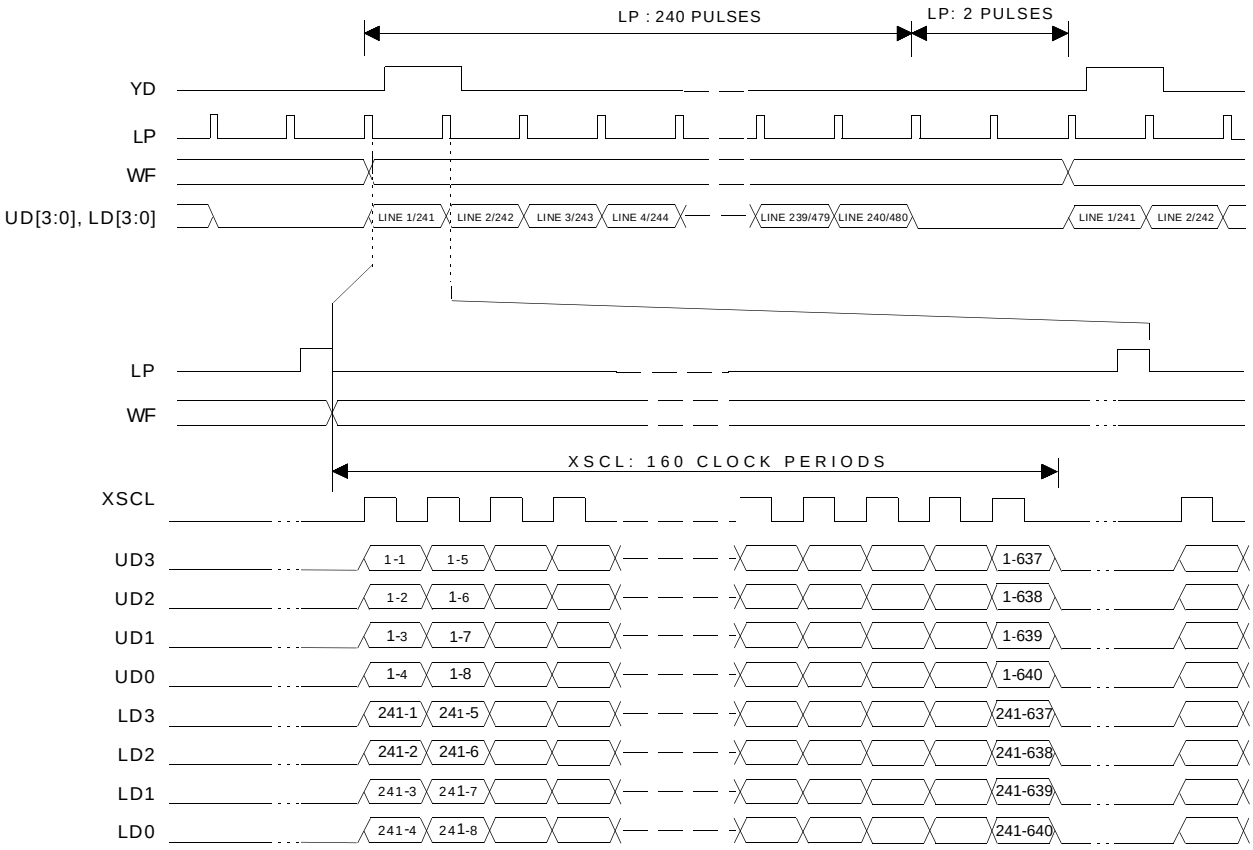
8-BIT SINGLE PANEL



Example timing for a 640x480 panel

■ MONOCHROME PASSIVE STN LCD PANEL INTERFACE

8-BIT DUAL PANEL

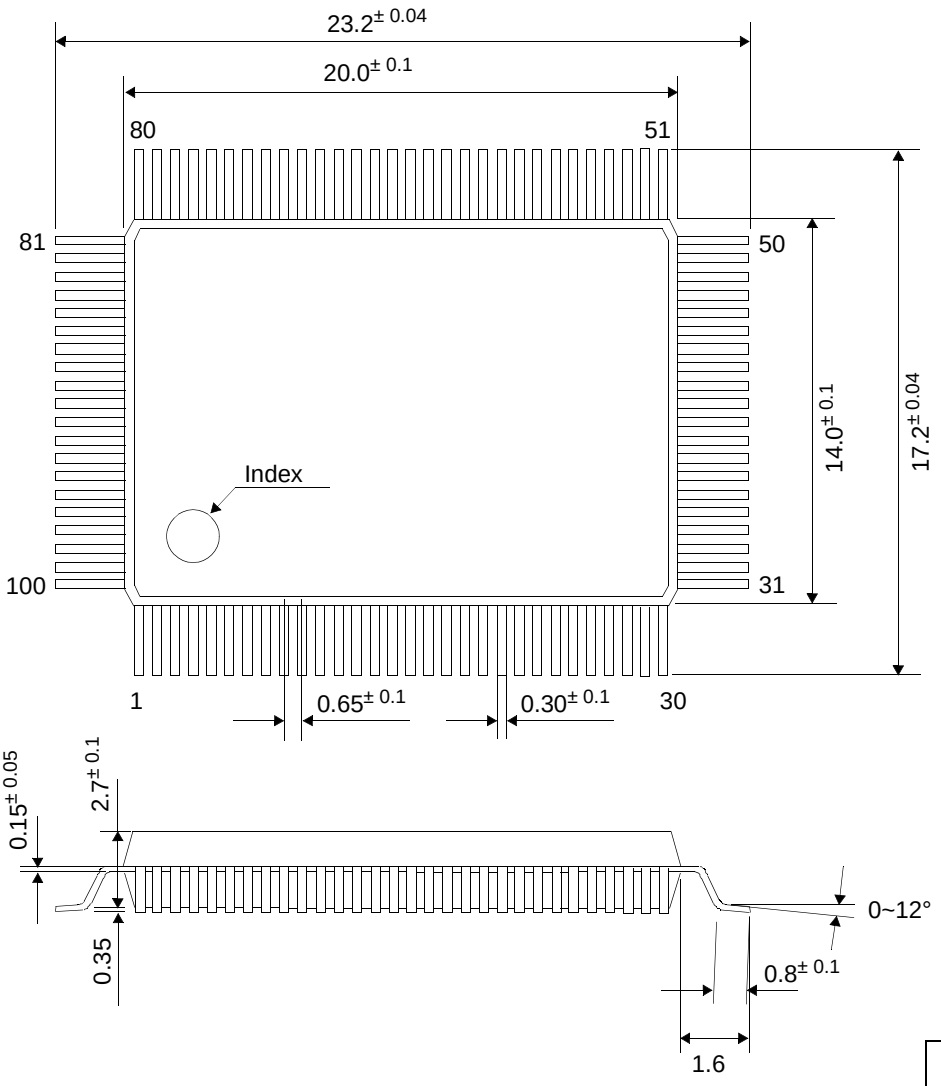


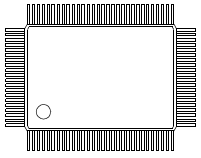
Example timing for a 640x480 panel

■ PACKAGE DIMENSIONS

QFP5-100PIN-S2
(SED1352)

Unit: mm

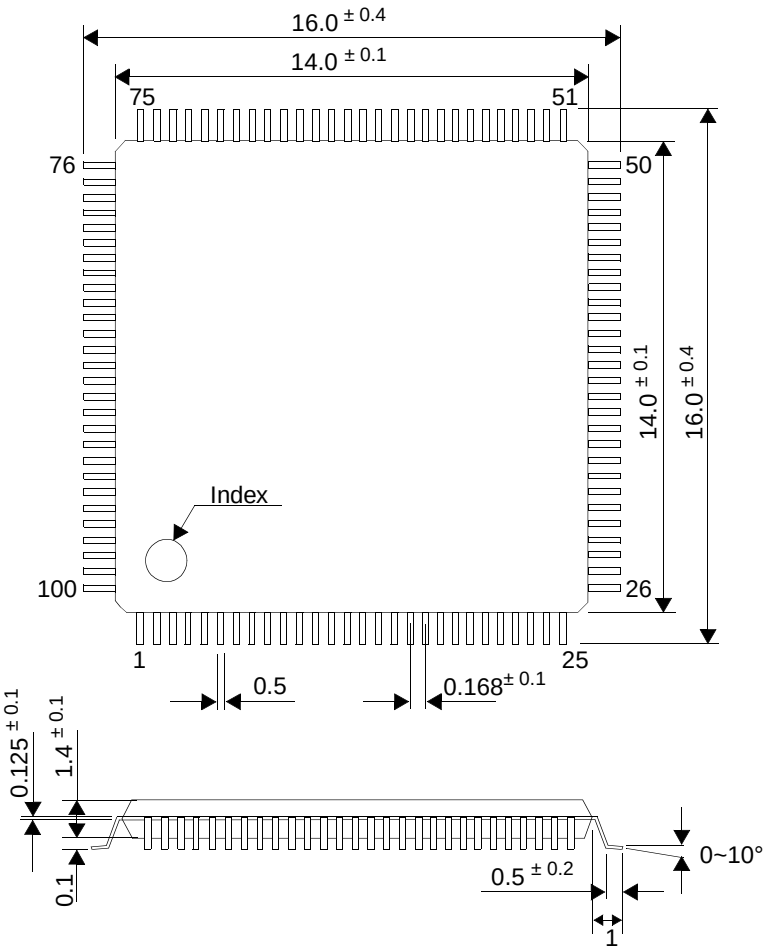


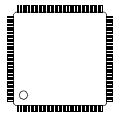


Actual Size

QFP15-100PIN-STD
(SED1352F1B)

Unit: mm





Actual Size

■ COMPREHENSIVE SUPPORT TOOLS

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

Documentation

- Technical manuals
- Evaluation/Demonstration board manual

Evaluation/Demonstration Board

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

■ Application Engineering Support

Seiko Epson offers the following services through their Sales and Marketing Network:

- Sales Technical Support
- Customer Training
- Design Assistance

CONTACT YOUR SALES REPRESENTATIVE FOR THESE COMPREHENSIVE DESIGN TOOLS:

- SED1352 Technical Manual
- SDU1352 Evaluation Boards
- CPU Independent Software Utilities

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

Copyright ©1997, 1998 Epson Research and Development, Inc. All rights reserved. VDC
Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws. EPSON is a registered trademark of Seiko Epson Corporation. Microsoft and Windows are registered trademarks of Microsoft Corporation.

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

Hardware Functional Specification

Document Number: X16-SP-001-16

Copyright © 1995, 1999 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

Table of Contents

List of Tables

THIS PAGE LEFT BLANK

List of Figures

THIS PAGE LEFT BLANK

1 INTRODUCTION

1.1 Scope

This is the Functional Specification for the SED1352 Dot Matrix Graphic Display 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; Graphics Subsystem Designers and Software Developers.

1.2 Overview Description

This device is designed for products where low cost, low power consumption, and low component count are the major design considerations. This chip operates from 2.7 Volts to 5.5 Volts and up to 25MHz to suit different power consumption, speed and cost requirements. The SED1352 offers a flexible microprocessor interface.

The SED1352 is capable of displaying a maximum of 16 levels of gray. A 16x4 Look-Up Table is provided to allow remapping of the 16 possible gray shades displayed on the LCD panel. The SED1352 can interface to an MC68000 family microprocessor or an 8/16-bit MPU/Bus with minimum external “glue” logic. This device can directly control up to 128K bytes of static RAM with a 16-bit data path, or up to 64K bytes with an 8-bit data path.

2 FEATURES

2.1 Technology

- low power CMOS
- 2.7 to 5.5 volt operation
- QFP5-100pin-S2 and QFP15-100 surface mount package

2.2 System

- maximum 25MHz input clock (or pixel clock)
- 2-terminal crystal input for internal oscillator or direct connection to external clock source
- maximum 16MHz, 16-bit MC68000 MPU interface
- 8-bit or 16-bit MPU/Bus interface with memory accesses controlled by a READY (or WAIT#) signal
- option to use built-in index register or direct-mapping to access one of fifteen internal registers
- 8-bit or 16-bit SRAM data bus interface configurations
- display memory configurations:
    - 128K bytes using one 64Kx16 SRAM
    - 128K bytes using two 64Kx8 SRAMs
    - 64K bytes using two 32Kx8 SRAMs
    - 40K bytes using one 8Kx8 and one 32Kx8 SRAM
    - 32K bytes using one 32Kx8 SRAM
    - 16K bytes using two 8Kx8 SRAMs
    - 8K bytes using one 8Kx8 SRAM

2.3 Display Modes

- 2/4 bits-per-pixel, 4/16 level gray shade display modes
- one 16x4 Look-Up Table provided for gray shade display modes
- maximum 16 shades of gray
- split screen display mode (see AUX[0Ah])
- virtual display mode (see AUX[0Dh])

2.4 Display Support

- example resolutions:
  - 640x480 with 4 grays
  - 640x400 with 16 grays
- passive monochrome LCD panels:
  - 4-bit single (4-bit data transfer)
  - 8-bit single (8-bit data transfer)
  - 8-bit dual (4-bit data transfer for each half panel)

2.5 Power Management

- two software power-save modes
- low power consumption
- panel power control switch (see AUX[01h] bit 4)

3 TYPICAL SYSTEM BLOCK DIAGRAMS

The following figures show typical system implementations of the SED1352. All of the following block diagrams are shown without SRAM or LCD display. Refer to interface specific Application Notes for complete details (X16-AN-xxx-xx).

3.1 16-Bit MC68000 MPU

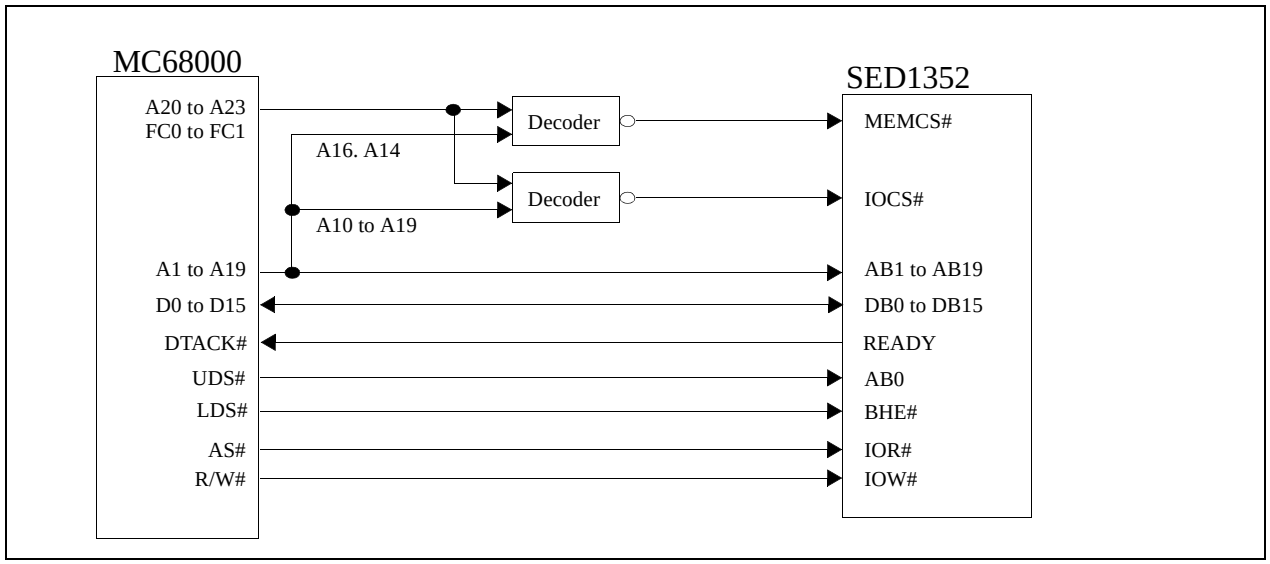


Figure 1: 16-Bit 68000 Series
(example implementation only - actual may vary)

3.1.1 MPU with READY (or WAIT#) signal

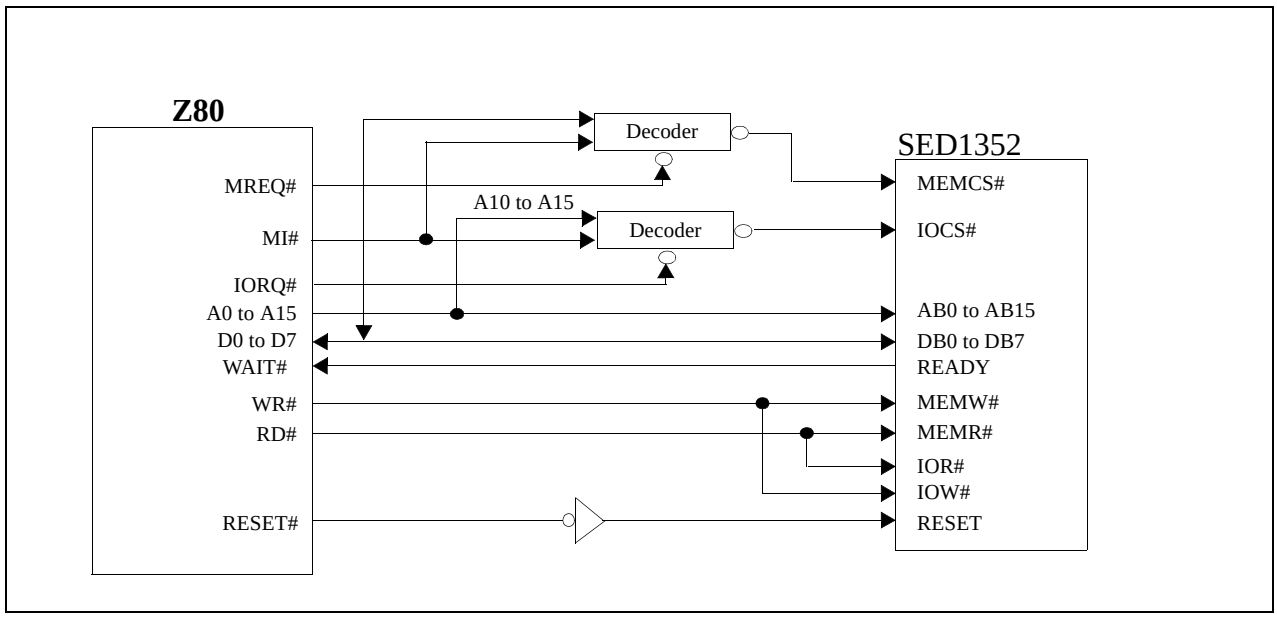


Figure 2: 8-Bit Mode, Example: Z80
(example implementation only - actual may vary)

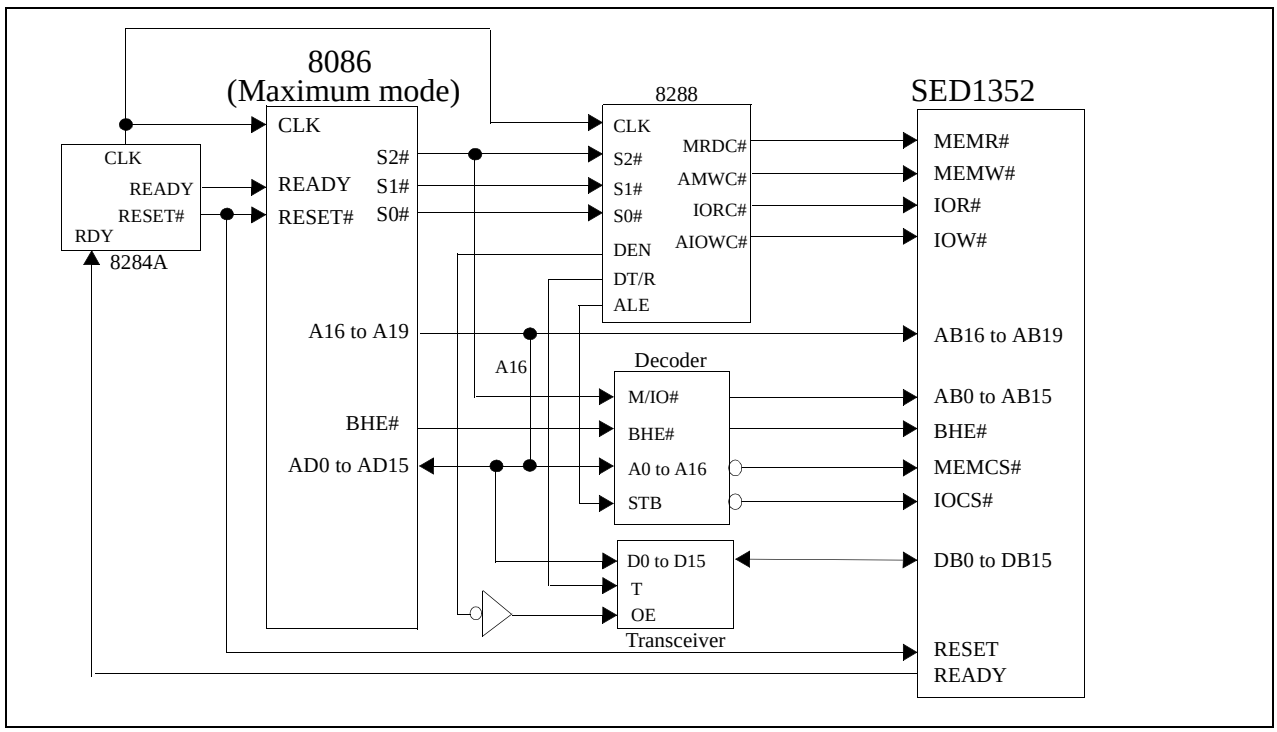


Figure 3: 16-Bit Mode, Example: i8086 (maximum mode)
(example implementation only - actual may vary)

3.1.2 ISA Bus

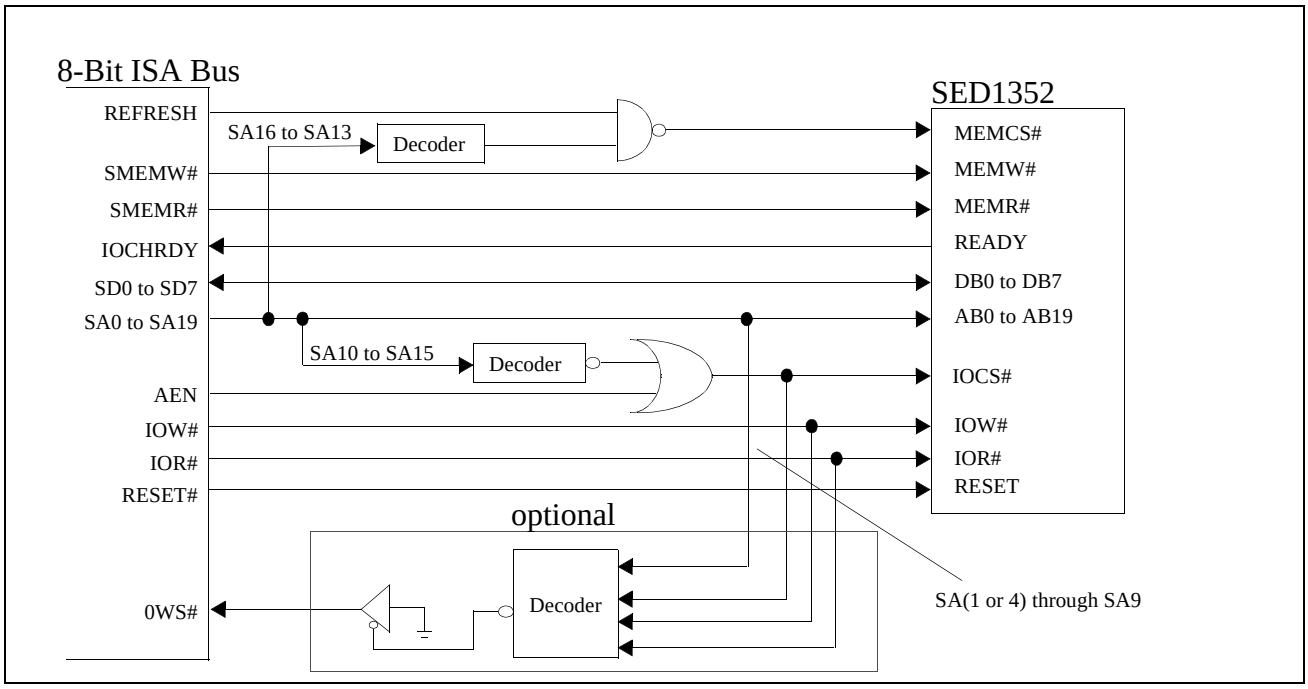


Figure 4: 8-Bit Mode (ISA)
(example implementation only - actual may vary

)

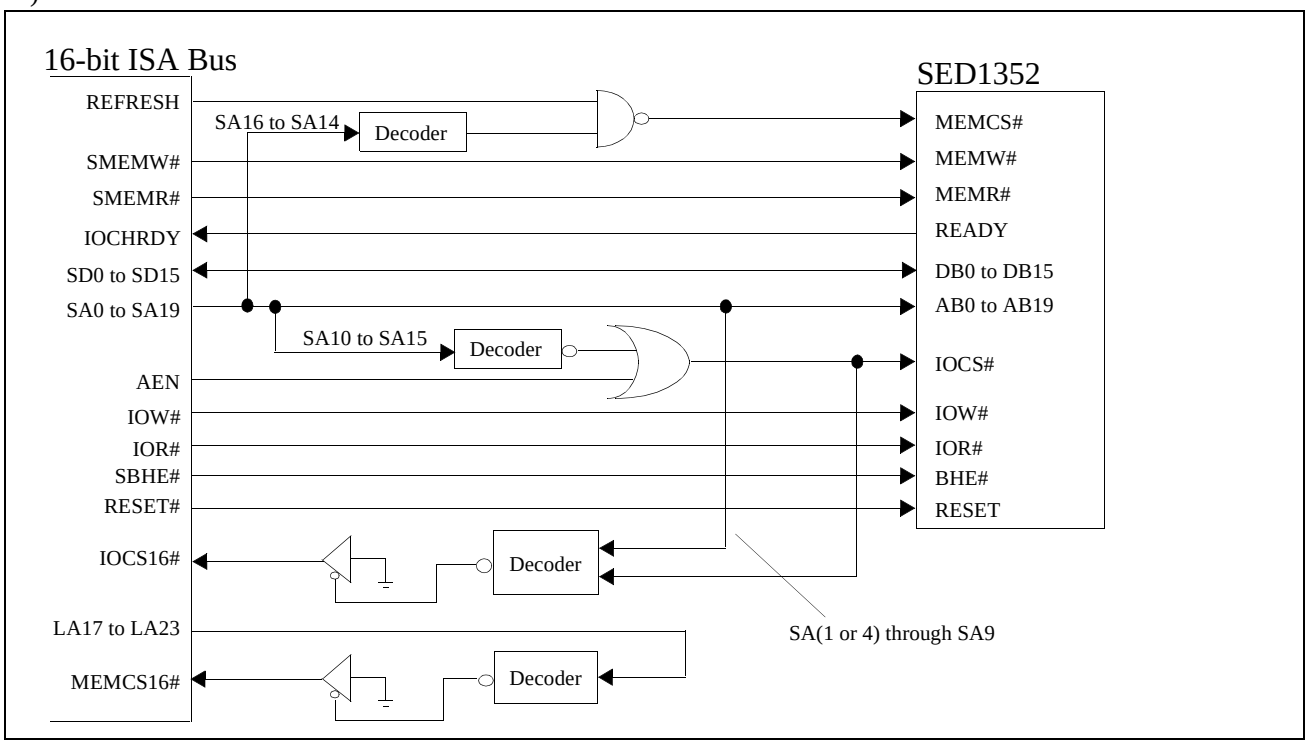


Figure 5: 16-Bit Mode (ISA)
(example implementation only - actual may vary)

3.2 Internal Block Diagram

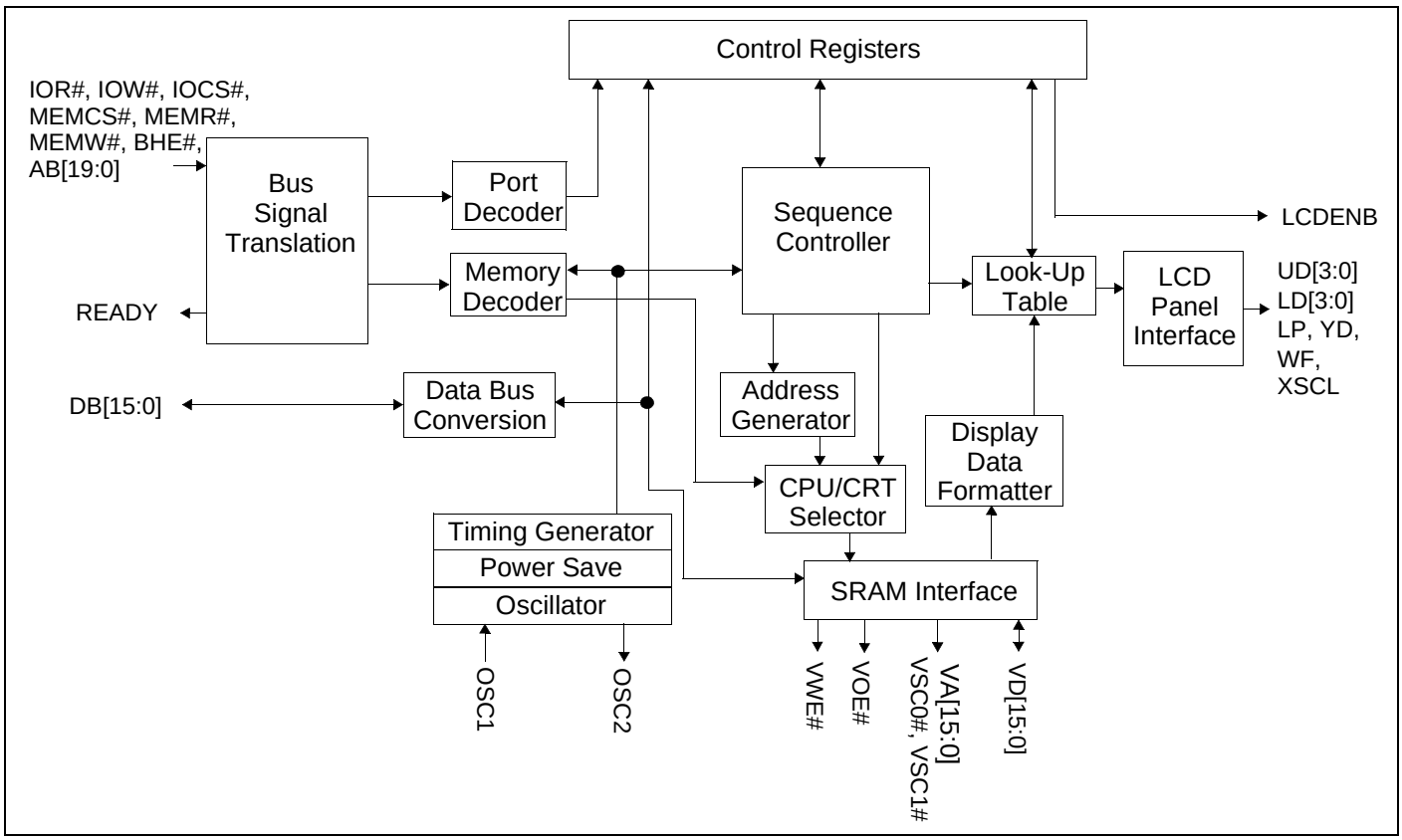


Figure 6: Internal Block Diagram

3.3 Functional Block Descriptions

3.3.1 Bus Signal Translation

According to configuration setting VD2, Bus Signal Translation translates MC68000 type CPU signals or READY type MPU signals to internal bus interface signals.

3.3.2 Control Registers

The fifteen internal Control and Configuration Registers are accessed by direct-mapping or by using the built-in internal index register.

3.3.3 Sequence Controller

The Sequence Controller generates horizontal and vertical display timings according to the configuration registers settings.

3.3.4 LCD Panel Interface

The LCD Interface performs frame rate modulation for passive monochrome LCD panels.

3.3.5 Look-Up Table

The Look-Up Table contains sixteen 4-bit wide palettes that can be configured as one 16x4 palette, or four 4x4 palettes used for the re-mapping of gray-scale outputs. See “Look-Up Table Architecture” on page 54.

3.3.6 Port Decoder

According to configuration settings VD1, VD12 - VD4, IOCS# and address lines AB9-1, the Port Decoder validates a given I/O cycle.

3.3.7 Memory Decoder

According to configuration settings VD15 - VD13, MEMCS# and address lines AB19-17, the Memory Decoder validates a given memory cycle.

3.3.8 Data Bus Conversion

According to configuration setting VD0, the Data Bus Conversion maps the external data bus, either 8-bit or 16-bit, into the internal odd and even data bus.

3.3.9 Address Generator

The Address Generator generates display refresh addresses used to access display memory.

3.3.10 CPU / CRT Selector

The CPU / CRT Selector accesses the display memory from the CPU or the display refresh circuitry.

3.3.11 Display Data Formatter

The Display Data Formatter reads the display data from the display memory and outputs the correct format for all supported LCD panel types and gray scale selections.

3.3.12 Clock Inputs / Timing

Clock Inputs / Timing generates the internal master clock according to the gray-level selected and display memory interface. The master clock (MCLK) can be:
- MCLK = input clock
- MCLK = 1/2 input clock
- MCLK = 1/4 input clock

Refer to section 9.2 SRAM Access Time for further details
Pixel clock = input clock.

3.3.13 SRAM Interface

The SRAM Interface generates the necessary signals to interface to the Display memory (SRAM).

4 PINOUT DIAGRAM

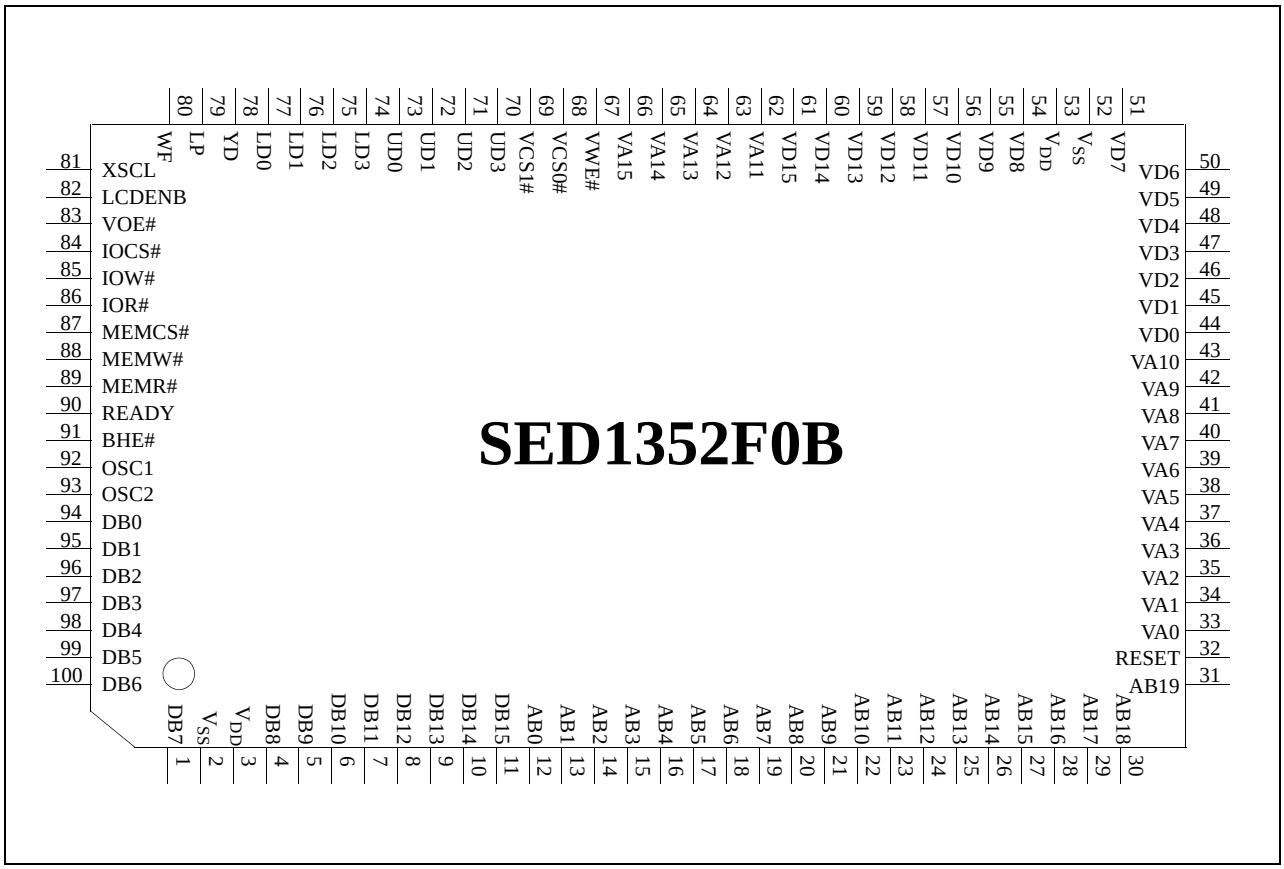


Figure 7: SED1352F0B Pinout Diagram

Note
Package type: surface mount QFP5-100pin-S2.

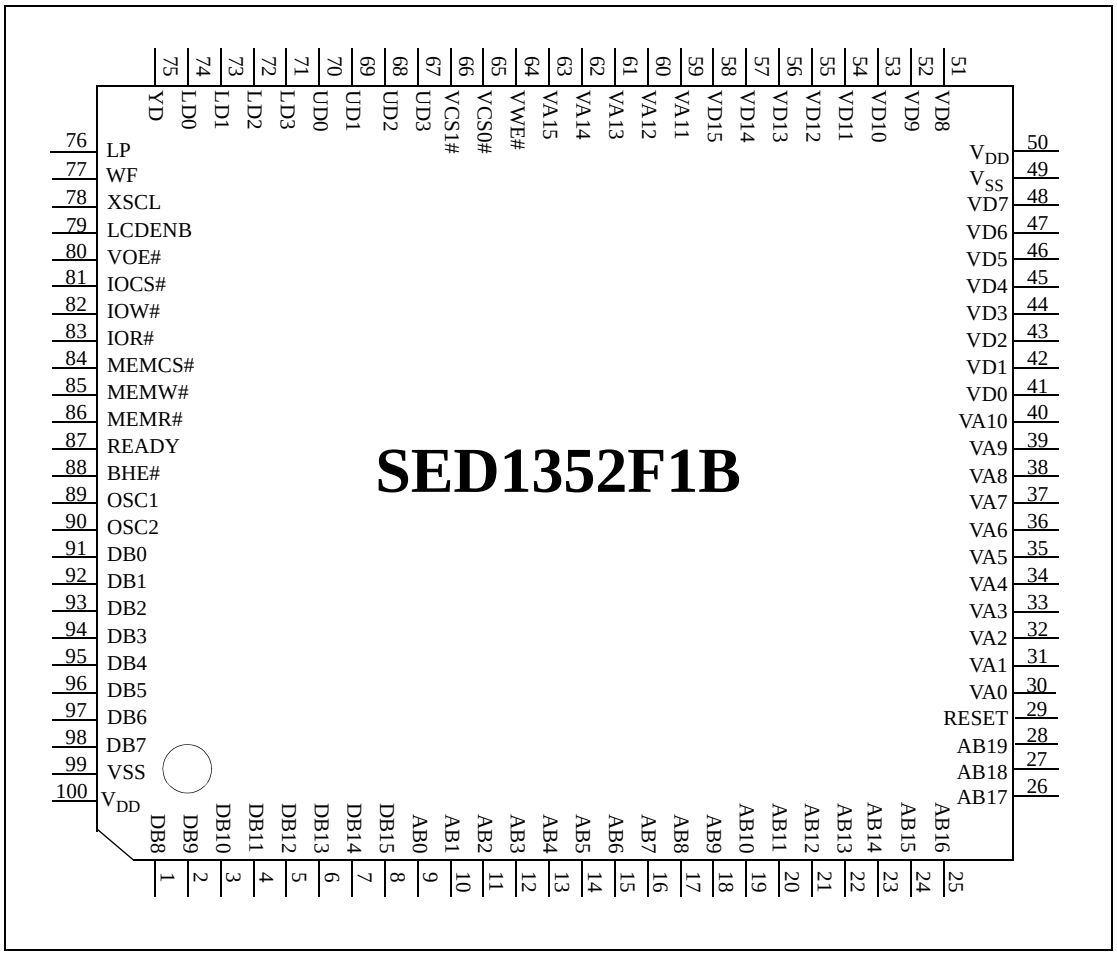
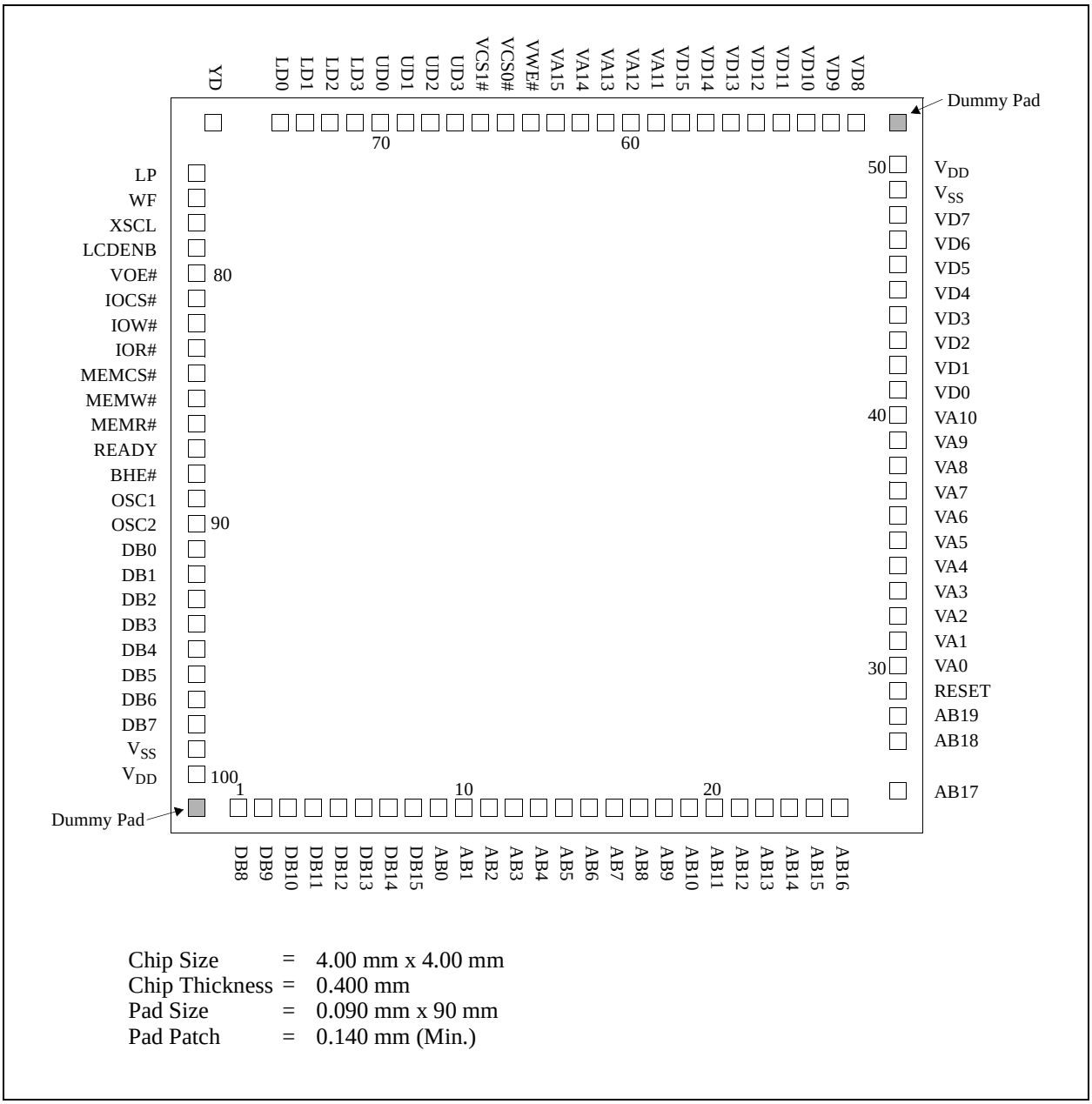


Figure 8: SED1352F1B Pinout Diagram

Note
Package type: surface mount QFP15-100pin.



Chip Size = 4.00 mm x 4.00 mm
Chip Thickness = 0.400 mm
Pad Size = 0.090 mm x 90 mm
Pad Patch = 0.140 mm (Min.)

Figure 9: SED1352D0B Pad Diagram

Table 4-1: SED1352D0A Pad Coordinates

Pad No.	Pin Name	Pad Center Coordinate		Pad No.	Pin Name	Pad Center Coordinate	
		X	Y			X	Y
1	DB8	-1.850	-2.071	37	VA7	2.071	-0.140
2	DB9	-1.670	-2.071	38	VA8	2.071	0.000
3	DB10	-1.496	-2.071	39	VA9	2.071	0.140
4	DB11	-1.330	-2.071	40	VA10	2.071	0.281
5	DB12	-1.168	-2.071	41	VD0	2.071	0.423
6	DB13	-1.012	-2.071	42	VD1	2.071	0.566
7	DB14	-0.860	-2.071	43	VD2	2.071	0.712
8	DB15	-0.712	-2.071	44	VD3	2.071	0.860
9	AB0	-0.566	-2.071	45	VD4	2.071	1.012
10	AB1	-0.423	-2.071	46	VD5	2.071	1.168
11	AB2	-0.281	-2.071	47	VD6	2.071	1.330
12	AB3	-0.140	-2.071	48	VD7	2.071	1.496
13	AB4	0.000	-2.071	49	V_{SS}	2.071	1.670
14	AB5	0.140	-2.071	50	V_{DD}	2.071	1.850
15	AB6	0.281	-2.071	51	VD8	1.850	2.071
16	AB7	0.423	-2.071	52	VD9	1.670	2.071
17	AB8	0.566	-2.071	53	VD10	1.496	2.071
18	AB9	0.712	-2.071	54	VD11	1.330	2.071
19	AB10	0.860	-2.071	55	VD12	1.168	2.071
20	AB11	1.012	-2.071	56	VD13	1.012	2.071
21	AB12	1.168	-2.071	57	VD14	0.860	2.071
22	AB13	1.330	-2.071	58	VD15	0.712	2.071
23	AB14	1.496	-2.071	59	VA11	0.566	2.071
24	AB15	1.670	-2.071	60	VA12	0.423	2.071
25	AB16	1.850	-2.071	61	VA13	0.281	2.071
26	AB17	2.071	-2.021	62	VA14	0.140	2.071
27	AB18	2.071	-1.670	63	VA15	0.000	2.071
28	AB19	2.071	-1.496	64	VWE#	-0.140	2.071
29	RESET	2.071	-1.330	65	VCS0#	-0.281	2.071
30	VA0	2.071	-1.168	66	VCS1#	-0.423	2.071
31	VA1	2.071	-1.012	67	UD3	-0.566	2.071
32	VA2	2.071	-0.860	68	UD2	-0.712	2.071
33	VA3	2.071	-0.712	69	UD1	-0.860	2.071
34	VA4	2.071	-0.566	70	UD0	-1.012	2.071
35	VA5	2.071	-0.423	71	LD3	-1.168	2.071
36	VA6	2.071	-0.281	72	LD2	-1.330	2.071

Table 4-1: SED1352D0A Pad Coordinates (Continued)

Pad No.	Pin Name	Pad Center Coordinate	
		X	Y
73	LD1	-1.496	2.071
74	LD0	-1.670	2.071
75	YD	-2.021	2.071
76	LP	-2.071	1.850
77	WF	-2.071	1.670
78	XSCL	-2.071	1.496
79	LCDENB	-2.071	1.330
80	VOE#	-2.071	1.168
81	IOCS#	-2.071	1.012
82	IOW#	-2.071	0.860
83	IOR#	-2.071	0.712
84	MEMCS#	-2.071	0.566
85	MEMW#	-2.071	0.423
86	MEMR#	-2.071	0.281
87	READY	-2.071	0.140
88	BHE#	-2.071	0.000
89	OSC1	-2.071	-0.140
90	OSC2	-2.071	-0.281
91	DB0	-2.071	-0.423
92	DB1	-2.071	-0.566
93	DB2	-2.071	-0.712
94	DB3	-2.071	-0.860
95	DB4	-2.071	-1.012
96	DB5	-2.071	-1.168
97	DB6	-2.071	-1.330
98	DB7	-2.071	-1.496
99	V_{SS}	-2.071	-1.670
100	V_{DD}	-2.071	-1.850
101	Dummy Pad	2.071	2.071
102	Dummy Pad	-2.071	-2.071

5 PINOUT DESCRIPTION

Key:

I	=	Input
O	=	Output
I/O	=	Bidirectional (Input/Output)
P	=	Power pin
COx	=	CMOS level output driver, x denotes driver type (see Table 6-4, “Output Specifications,” on page 27)
TSx	=	Tri-state CMOS level output driver, x denotes driver type (see Table 6-4, “Output Specifications,” on page 27)
TSxD2	=	Tri-state CMOS level output driver with pull down resistor (typical values of 100KΩ/200KΩ at 5V/3.0V respectively), x denotes driver type (see Table 6-4, “Output Specifications,” on page 27)
TTL	=	TTL level input (for VDD = 5.0V, (see Table 6-3, “Input Specifications,” on page 26 for VDD = 3.0V and 3.3V)
TTLS	=	TTL level input with hysteresis

Table 5-1: Bus Interface

Pin Name	Type	F0B Pin #	F1B/D0B Pin/Pad #	Driver	Description
DB0-DB15	I/O	94 - 100, 1, 4 -11	91 - 98, 1 - 8	TS2	These pins are connected to the system data bus. In 8-bit bus mode, DB8-DB15 must be tied to V_{DD}.
AB0	I	12	9	TTL	In MC68000 MPU interface, this pin is connected to the Upper Data Strobe (UDS#) pin of MC68000. In other bus interfaces, this pin is connected to the system address bus.
AB1-AB19	I	13 - 31	10 - 28	TTL	These pins are connected to the system address bus.
BHE#	I	91	88	TTLS	In MC68000 MPU interface, this pin is connected to the Lower Data Strobe (LDS#) pin of MC68000. In other bus interfaces, this pin is the Bus High Enable input for use with 16-bit system. In 8-bit bus mode, tie BHE# input to V_{DD}.
IOCS#	I	84	81	TTLS	Active low input to select one of fifteen internal registers.
IOW#	I	85	82	TTLS	In MC68000 MPU interface, this pin is connected to the R/W# pin of MC68000. This input pin defines whether the data transfer is a read (active high) or write (active low) cycle. In other bus interfaces, this is the active low input to write data into an internal register.
IOR#	I	86	83	TTLS	In MC68000 MPU interface, this pin is connected to the AS# pin of MC68000. This input pin indicates a valid address is available on the address bus. In other bus interfaces, this is the active low input to read data from an internal register.
MEMCS#	I	87	84	TTLS	Active low input to indicate the attempt to access the display memory.
MEMW#	I	88	85	TTLS	Active low input to write data to the display memory. This pin should be tied to V_{DD} in an MC68000 MPU interface.

Table 5-1: Bus Interface (Continued)

Pin Name	Type	F0B Pin #	F1B/D0B Pin/Pad #	Driver	Description
MEMR#	I	89	86	TTLS	Active low input to read data from the display memory. This pin should be tied to V_{DD} in an MC68000 MPU interface.
READY	O	90	87	TS3	For MC68000 MPU interface, this pin is connected to the DTACK# pin of MC68000 and is driven low when ever a data transfer is complete. In other bus interfaces, this output is driven low to force the system to insert wait states when needed. READY is placed in a high impedance (Hi-Z) state after the transfer is completed.
RESET	I	32	29	TTLS	Active high input to force all signals to their inactive states.

Table 5-2: Display Memory Interface

Pin Name	Type	F0B Pin #	F1B/D0B Pin/Pad #	Driver	Description
VD0-VD15	I/O	44 - 51, 54 - 61	41 - 48, 51 - 58	TS2D2	These pins are connected to the display memory data bus. For 16-bit interface, VD0-VD7 are connected to the display memory data bus of even byte addresses and VD8-VD15 are connected to the display memory data bus of odd byte addresses. The output drivers of these pins are placed in a high impedance state when RESET is high. On the falling edge of RESET the values of VD0-VD15 are latched into the chip to configure various hardware options. VD0-VD15 each have an internal pull-down resistor (see section Table 5-6: on page 25).
VA0-VA15	O	33 - 43, 62 - 66	30 - 40 59, 63	CO2	These pins are connected to the display memory address bus.
VCS1#	O	69	66	CO2	Active low chip-select output to the second or odd byte address SRAM. See Display Memory Interface section for details.
VCS0#	O	68	65	CO2	Active low chip-select output to the first or even byte address SRAM. See Display Memory Interface section for details.
VWE#	O	67	64	CO2	Active low output used for writing data to the display memory. This pin is connected to the WE# input of the SRAMs.
VOE#	O	83	80	CO2	Active low output to enable reading of data from the display memory. This pin is connected to the OE# input of the SRAMs.

Table 5-3: LCD Interface

Pin Name	FPDI-1™ Pin Name[a]	Type	F0B Pin #	F1B/D0B Pin/Pad #	Driver	Description
UD3-UD0	UD3-UD0	O	70 - 73	67 - 70	CO4	Upper panel display data for dual panel mode. For single panel mode, these bits are the most significant 4 bits of the 8-bit output data to the panel (PD[4:7]). For 4-bit single panel mode, these bits are the 4 bits of output data to the panel.
LD3-LD0	LD3-LD0	O	74 - 77	71 - 74	CO4	Lower panel display data for dual panel mode. For 8-bit single panel mode, these bits are the least significant 4 bits of the 8-bit output data to the panel (PD[0:3]). For 4-bit single panels, these bits are driven 0 (low state).
XSCL	FPSHIFT	O	81	78	CO4	Display data shift clock. Data is shifted into the LCD X-drivers on the falling edge of this signal.
LP	FPLINE	O	79	76	CO4	Display data latch clock. The falling edge of this signal is used to latch a row of display data in the LCD X-drivers and to turn on the row driver (Y driver).
WF	MOD	O	80	77	CO4	LCD backplane BIAS signal. This output toggles according to the value programmed in AUX[05h].
YD	FPFRAME	O	78	75	CO4	Vertical scanning start pulse. A logic ‘1’ on this signal, sampled by the LCD module on the falling edge of LP, is used by the panel row driver (Y driver) to indicate the start of the vertical frame.
LCDENB		O	82	79	CO2	LCD enable signal output. It can be used externally to turn off the panel supply voltage and backlight.

[a] VESA Flat Panel Display Interface Standard (FPDI-1™)

Table 5-4: Clock Inputs

Pin Name	Type	F0B Pin #	F1B/D0B Pin/Pad #	Driver	Description
OSC1	I	92	89	*	This pin, along with OSC2, is the 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin is the clock input.
OSC2	O	93	90	*	This pin, along with OSC1, is the 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source this pin should be left unconnected.

Table 5-5: Power Supply

Pin Name	Type	F0B Pin #	F1B/D0B Pin/Pad #	Driver	Description
V_{DD}	P	3, 53	50, 100	P	Voltage supply.
V_{SS}	P	2, 52	49, 99	P	Voltage ground.

5.1 Summary of Configuration Options

The SED1352 requires some configuration information on power-up. This information is provided through the SRAM data lines VD[0...15]. The state of these pins are read on the falling edge of RESET and used to configure the following options:

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

Pin Name	value on this pin at falling edge of RESET is used to configure: (1/0)	
	1	**0**
VD0	16-bit host bus interface	8-bit host bus interface
VD1	Use direct-mapping for I/O accesses	Use internal index register for I/O accesses
VD2	MC68000 MPU interface	MPU / Bus interface with memory accesses controlled by a READY (WAIT#) signal
VD3	Swap of high and low data bytes in 16-bit bus interface	No byte swap of high and low data bytes in 16-bit bus interface
VD4-VD12	Select I/O mapping address bits [1:9]. These nine bits are latched on power-up and are compared to the MPU address bits [1-9]. A valid I/O cycle combined with a valid address will enable the internal I/O decoder. Therefore, both types of I/O mapping are limited to even address boundaries to determine either the absolute or indexed I/O address of the first register. Note that a “valid I/O cycle” includes IOCS# being toggled low. In direct mapping, the base I/O address is selected by VD7-VD12. In indexing, the base I/O address is selected by VD4-VD12.	
VD13-VD15	Select memory mapping address bits [1:3]. These three bits are latched on power-up and are compared to the MPU address bits [17-19]. A valid memory cycle combined with a valid address will enable the internal memory decoder. As only the three most significant bits of the address are compared, the maximum amount of memory supported is 128K bytes. Note that a “valid memory cycle” includes MEMCS# being toggled low. If 128K byte memory is used, it must be mapped at an even address so all 128K bytes is available without a change in state on A17, as this would invalidate the internal compare logic.	

Note

The SED1352 has internal pulldown resistors on these pins and therefore will be pulled down and read on a logic “0” after RESET. If pullup resistors are required refer to Table 6-3, “Input Specifications,” on page 26 for pulldown resistor values.

Example: If an ISA bus (no byte swap) with memory segment A000h and I/O location 300h are used, the corresponding settings of VD15-VD0 would be:

Table 5-7: I/O and Memory Addressing Example

	8-Bit ISA Bus		16-Bit ISA Bus	
Pin Name	**Index Register**	**Direct Mapping**	**Index Register**	**Direct Mapping**
VD0	0	0	1	1
VD1	0	1	0	1
VD2	0	0	0	0
VD3	0	0	0	0
VD12-VD4	11 0000 000	11 0000 xxx	11 0000 000	11 0000 xxx
VD15-VD13	101	101	101	101

Where x = don’t care; 1 = connected to pull-up resistor; 0 = no pull-up resistor.

6 D.C. CHARACTERISTICS

Table 6-1: Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
V_{DD}	Supply Voltage	V_{SS} - 0.3 to + 6.5	V
V_{IN}	Input Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_{OUT}	Output Voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
T_{STG}	Storage Temperature	-65 to 150	° C
T_{SOL}	Solder Temperature/Time	260 for 10 sec. max at lead	° C

Table 6-2: Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{DD}	Supply Voltage	V_{SS} = 0V	2.7	3.0/3.3/5.0	5.5	V
V_{IN}	Input Voltage		V_{SS}	--	V_{DD}	V
I_{OPR}	Operating Current	f_{OSC} = 6 MHz, 16 grays		3.0/3.5/7.0		mA
T_{OPR}	Operating Temperature		-40	25	85	° C
P_{TYP}	Typical Active Power Consumption	f_{OSC} = 6 MHz, 16 grays		9.0/11.55/35.0		mW

Table 6-3: Input Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage	V_{DD} = 4.5V V_{DD} = 3.0V V_{DD} = 2.7V			0.8 0.6 0.5	V
V_{IH}	High Level Input Voltage	V_{DD} = 5.5V V_{DD} = 3.6V V_{DD} = 3.3V	2.0 2.5 2.3			V
V_{T+}	Positive-going Threshold	V_{DD} = 5.0 V_{DD} = 3.3 V_{DD} = 3.0			2.4 2.4 2.3	V
V_{T-}	Negative-going Threshold	V_{DD} = 5.0 V_{DD} = 3.3 V_{DD} = 3.0	0.6 0.6 0.5			V
V_H	Hysteresis Voltage	V_{DD} = 5.0 V_{DD} = 3.3 V_{DD} = 3.0	0.1 0.1 0.1			V
I_{IZ}	Input Leakage Current	--	-1		1	µA

Table 6-3: Input Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
C_{IN}	Input Pin Capacitance			4		pF
R_{PD}	Pull Down Resistance	V_{DD} = 5.0V $V_I = V_{DD}$	50		200	kΩ
R_{PD}	Pull Down Resistance	V_{DD} = 3.3V $V_I = V_{DD}$	90		360	kΩ
R_{PD}	Pull Down Resistance	V_{DD} = 3.0V $V_I = V_{DD}$	100		400	kΩ

Table 6-4: Output Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{OL} (5.0V)	Low Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OL} = 6 mA I_{OL} = 12 mA I_{OL} = 24 mA			V_{SS} + 0.4	V
V_{OL} (3.3V)	Low Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OL} = 3mA I_{OL} = 6mA I_{OL} = 12mA			V_{SS} + 0.3	V
V_{OL} (3.0V)	Low Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OL} = 3mA I_{OL} = 5 mA I_{OL} = 10mA			V_{SS} + 0.3	V
V_{OH} (5.0V)	High Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OH} = -2 mA I_{OH} = -4 mA I_{OH} = -8 mA	V_{DD}-0.4			V
V_{OH} (3.3V)	Low Level Output Voltage Type 2 - TS2, CO2, TS2D2 Type 3 - TS3 Type 4 - TS4, CO4	 I_{OL} = -1 mA I_{OL} = -2 mA I_{OL} = -4 mA	V_{DD}-0.3			V
V_{OH} (3.0V)	High Level Output Voltage Type 2- TS2, CO2, TS2D2 Type 3- TS3 Type 4- TS4, CO4	 I_{OH} = -1 mA I_{OH} = -1.8 mA I_{OH} = -3.5 mA	V_{DD}-0.3			V
I_{OZ}	Output Leakage Current		-1		1	µA
C_{OUT}	Output Pin Capacitance			6		pF
C_{BID}	Bidirectional Pin Capacitance			10		pF

7 A.C. CHARACTERISTICS

Conditions: V_{DD} = 3.0V ± 10%, 3.3V ± 10% or V_{DD} = 5.0V ± 10%

T_A = -40 °C to 85 °C

T_{rise} and T_{fall} for all inputs must be ≤ 5 nsec (10% ~ 90%)

C_L = 80pF (Bus/MPU Interface)

C_L = 100pF (LCD Panel Interface)

C_L = 20pF (Display Memory Interface)

7.1 Bus Interface Timing

7.1.1 MC68000 Interface Timing

Note
All input timing parameters are based on a maximum 16MHz bus clock.

IOW# Timing

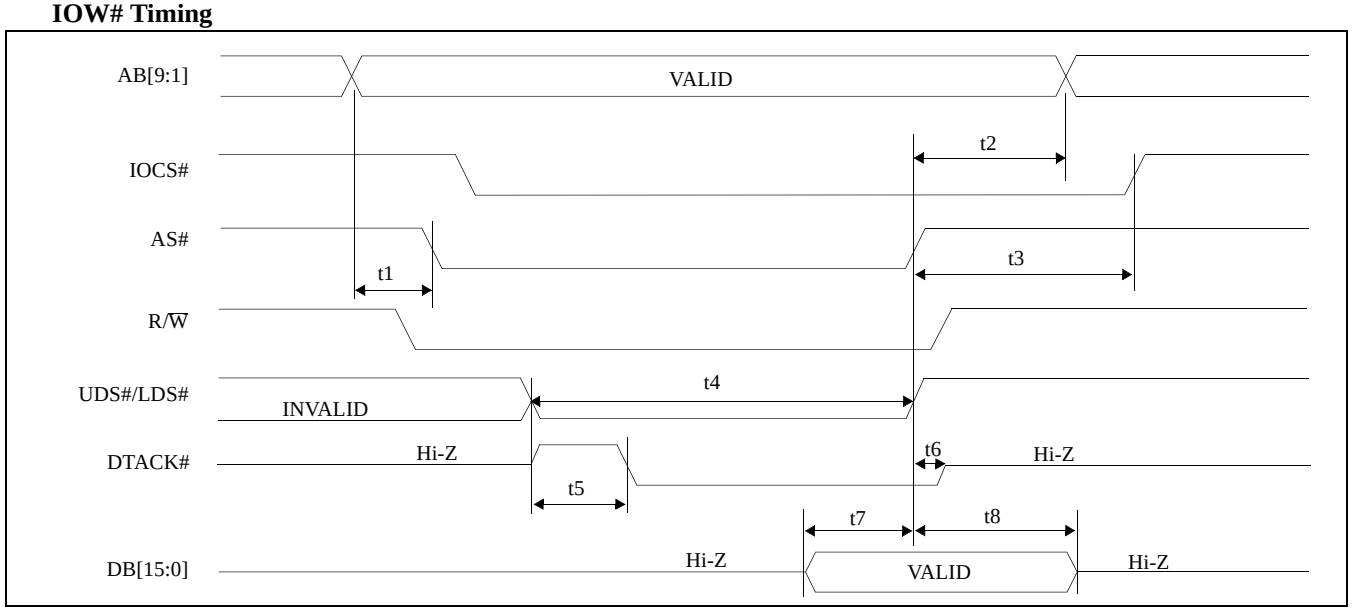


Figure 10: IOW# Timing (68000)

Table 7-1: IOW# Timing (68000)

		3V/3.3V			5V			
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**
t1	AB[9:1] valid before AS# falling edge	10			0			ns
t2	AB[9:1] hold from AS# rising edge	20			10			ns
t3	IOCS# hold from AS# rising edge	0			0			ns
t4	UDS#/LDS# valid before AS# rising edge	30			20			ns
t5	UDS#/LDS# falling edge to DTACK# falling edge			40			25	ns
t6	AS# rising edge to DTACK# hi-z delay			45			25	ns
t7	DB[15:0] setup to AS# rising edge	20			10			ns
t8	DB[15:0] hold from AS# rising edge	20			10			ns

IOR# Timing

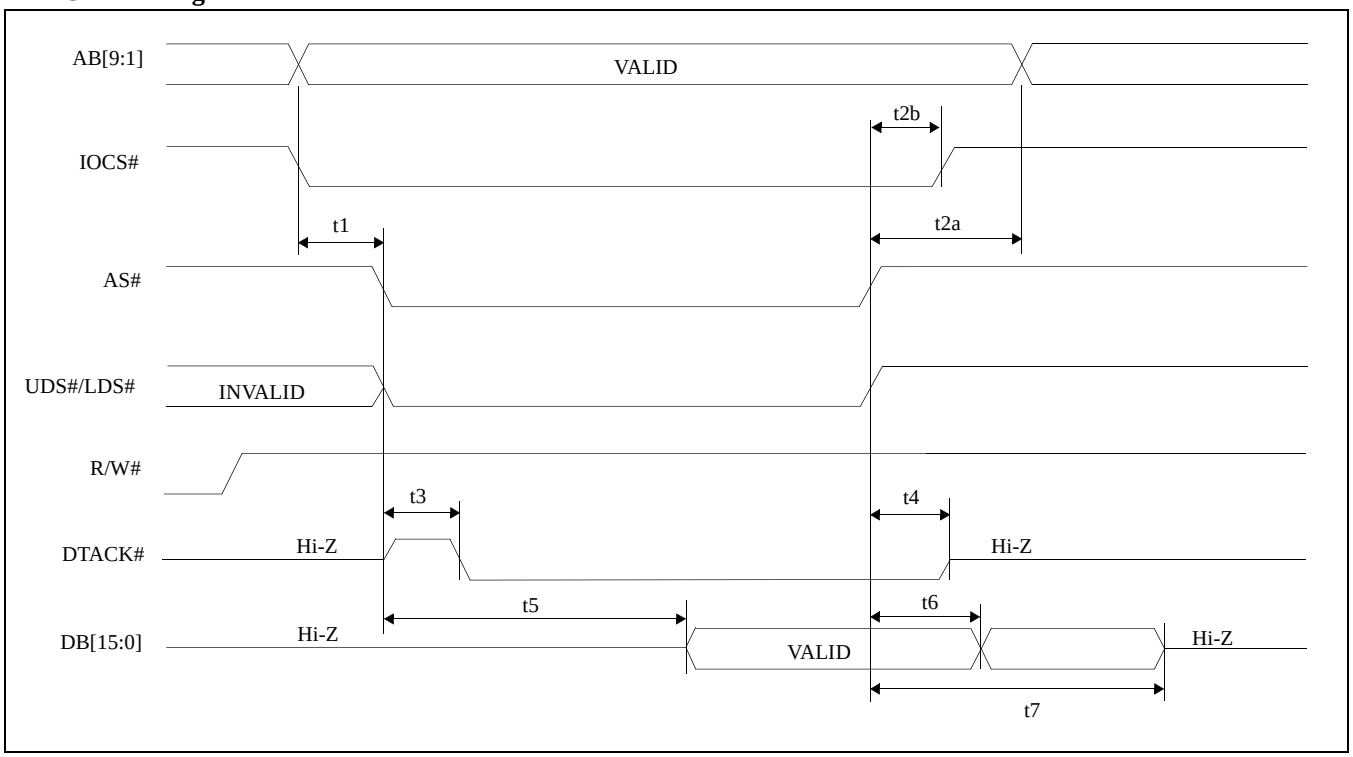


Figure 11: IOR# Timing (68000)

Table 7-2: IOR# Timing (68000)

		3V/3.3V			5V			
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**
t1	AB[9:1] and IOCS# valid before AS# falling edge	10			0			ns
t2a	AB[9:1] hold from AS# rising edge	20			10			ns
t2b	IOCS# hold from AS# rising edge			0			0	ns
t3	AS# falling edge to DTACK# falling edge			35			25	ns
t4	AS# rising edge to DTACK# hi-z delay			45			25	ns
t5	AS# falling edge to DB[15:0] valid			80			60	ns
t6	DB[15:0] hold from AS# rising edge			25			20	ns
t7	AS# rising edge to DB[15:0] hi-z delay			35			30	ns

MEMW# Timing

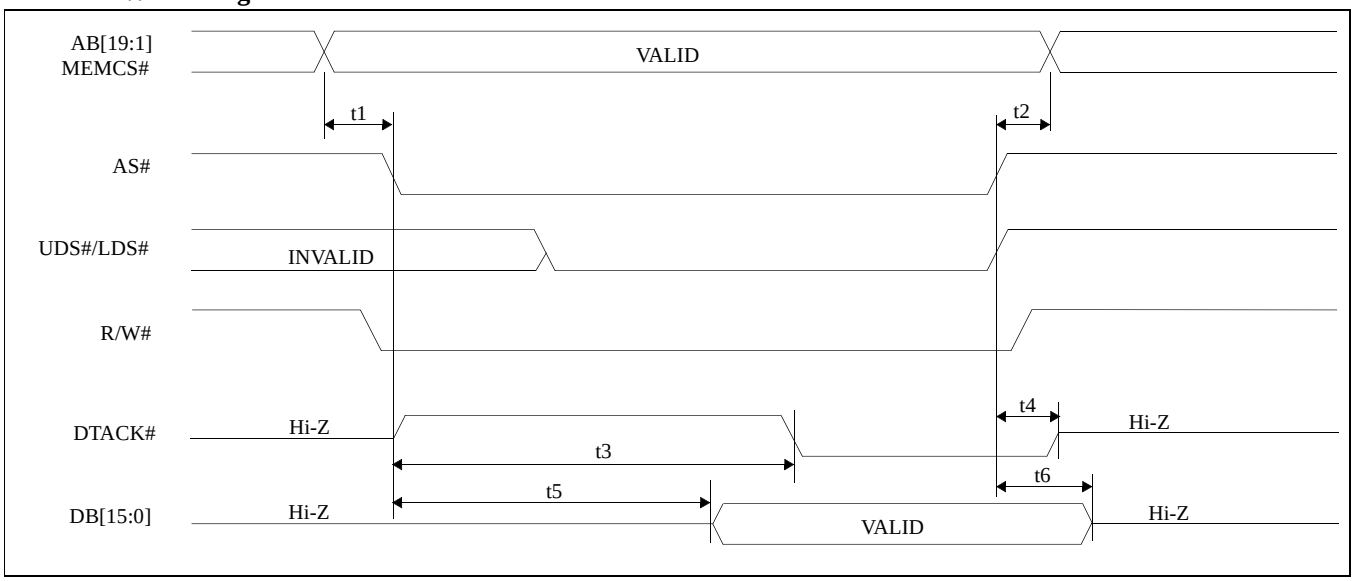


Figure 12: MEMW# Timing (68000)

Table 7-3: MEMW# Timing (68000)

		3V/3.3V			5V			
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**
t1	AB[19:1] and MEMCS# valid before AS# falling edge	0			0			ns
t2	AB[19:1] and MEMCS# hold from AS# rising edge	0			0			ns
t3	AS# falling edge to DTACK# falling edge			3.5 * MCLK + 20			3.5 * MCLK + 10	ns
t4	AS# rising edge to DTACK hi-z delay			45			22	ns
t5	AS# falling edge to DB[15:0] valid			120			140	ns
t6	DB[15:0] hold from AS# rising edge	0			0			ns

Where MCLK period = $1/f_{OSC}$, or $2/f_{OSC}$, or $4/f_{OSC}$ depending on which mode the chip is in. (see section 9.2 and 9.3).

MEMR# Timing

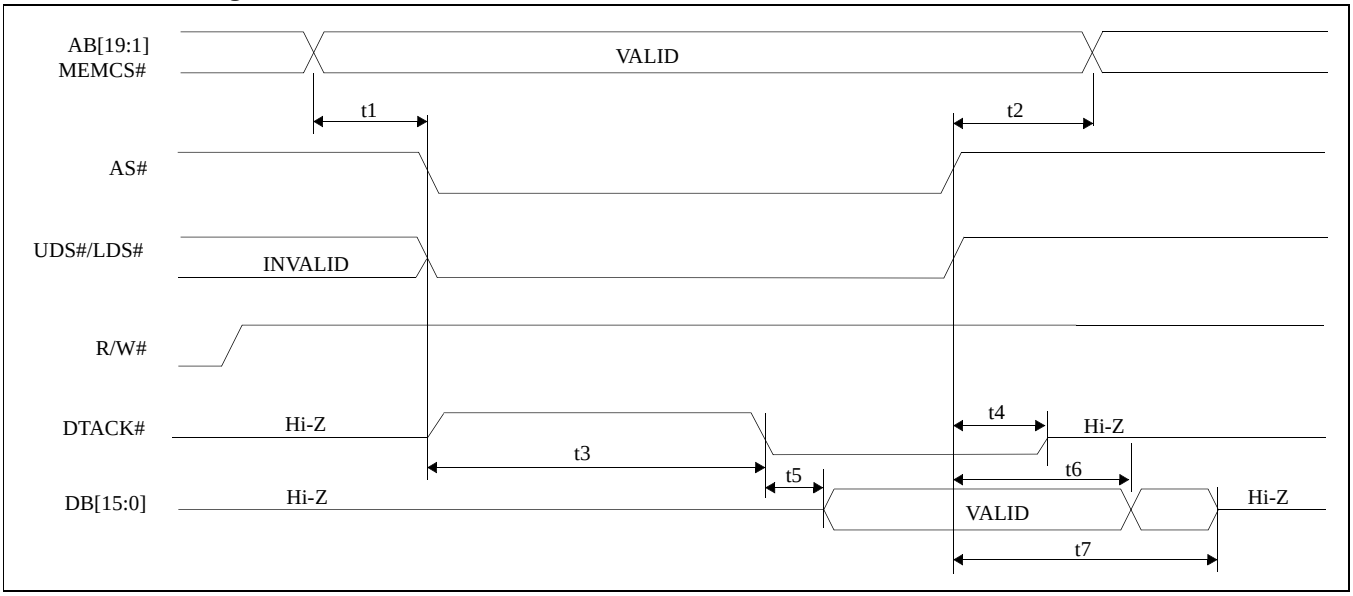


Figure 13: MEMR# Timing (68000)

Table 7-4: MEMR# Timing (68000)

		3V/3.3V			5V			
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**
t1	AB[19:1] and MEMCS# valid before AS# falling edge	0			0			ns
t2	AB[19:1] and MEMCS# hold from AS# rising edge	0			0			ns
t3	AS# falling edge to DTACK# falling edge			3.5 * MCLK + 20			3.5 * MCLK + 10	ns
t4	AS# rising edge to DTACK# hi-z delay			42			20	ns
t5	DTACK# falling edge to DB[15:0] valid			20			20	ns
t6	DB[15:0] hold from AS# rising edge			54			28	ns
t7	AS# rising edge to DB[15:0] hi-z delay			60			30	ns

Where MCLK period = $1/f_{OSC}$, or $2/f_{OSC}$, or $4/f_{OSC}$ depending on which mode the chip is in. (see section 9.2 and 9.3).

7.1.2 Non-68000, MPU/Bus With READY (or WAIT#) Signal

IOW# Timing

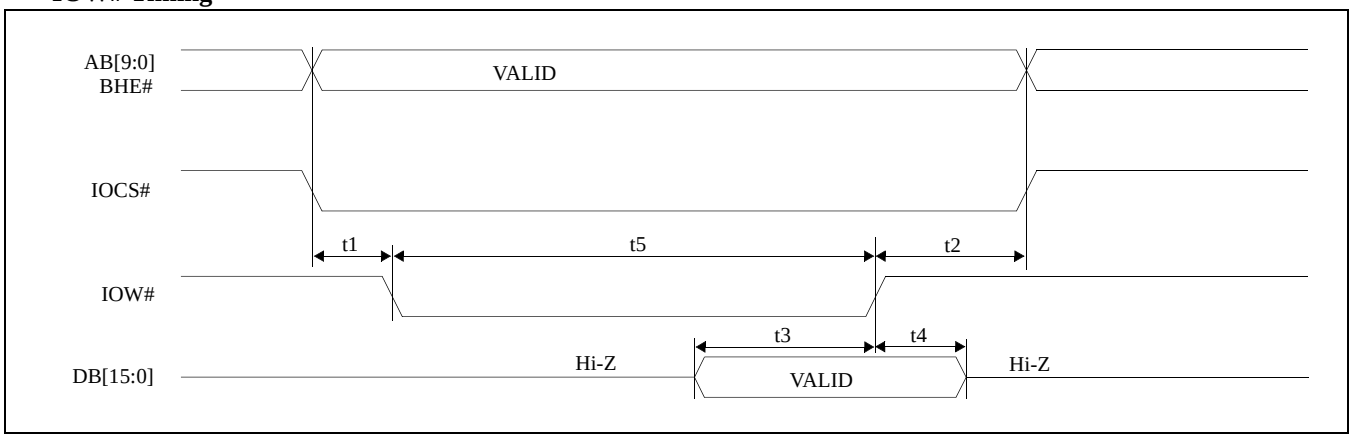


Figure 14: IOW# Timing (Non-68000)

Table 7-5: IOW# Timing (Non-68000)

		3V/3.3V			5V			
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**
t1	AB[9:0], BHE# and IOCS# valid before IOW# falling edge	10			0			ns
t2	AB[9:0], BHE# and IOCS# hold from IOW# rising edge	20			10			ns
t3	DB[15:0] setup to IOW# rising edge	20			10			ns
t4	DB[15:0] hold from IOW# rising edge	20			10			ns
t5	Pulse width of IOW#	30			20			

IOR# Timing

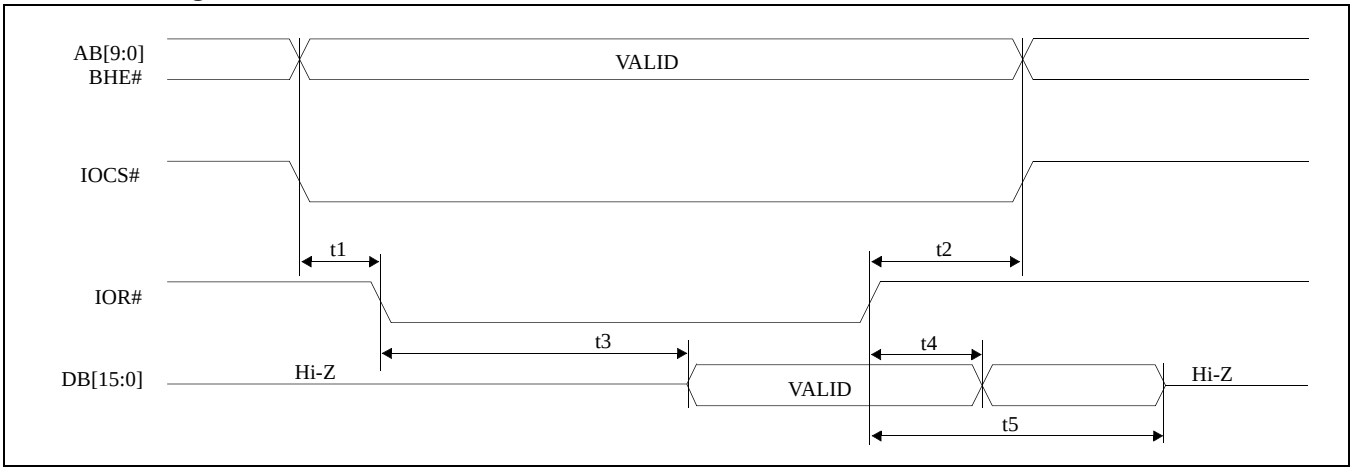


Figure 15: IOR# Timing (Non-68000)

Table 7-6: IOR# Timing (Non-68000)

		3V/3.3V			5V			
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**
t1	AB[9:0], BHE# and IOCS# valid before IOR# falling edge	10			0			ns
t2	AB[9:0], BHE# and IOCS# hold from IOR# rising edge	20			10			ns
t3	IOR# falling edge to DB[15:0] valid			80			60	ns
t4	DB[15:0] hold from IOR# rising edge			25			20	ns
t5	IOR# rising edge to DB[15:0] hi-z delay			30			30	ns

MEMW# Timing

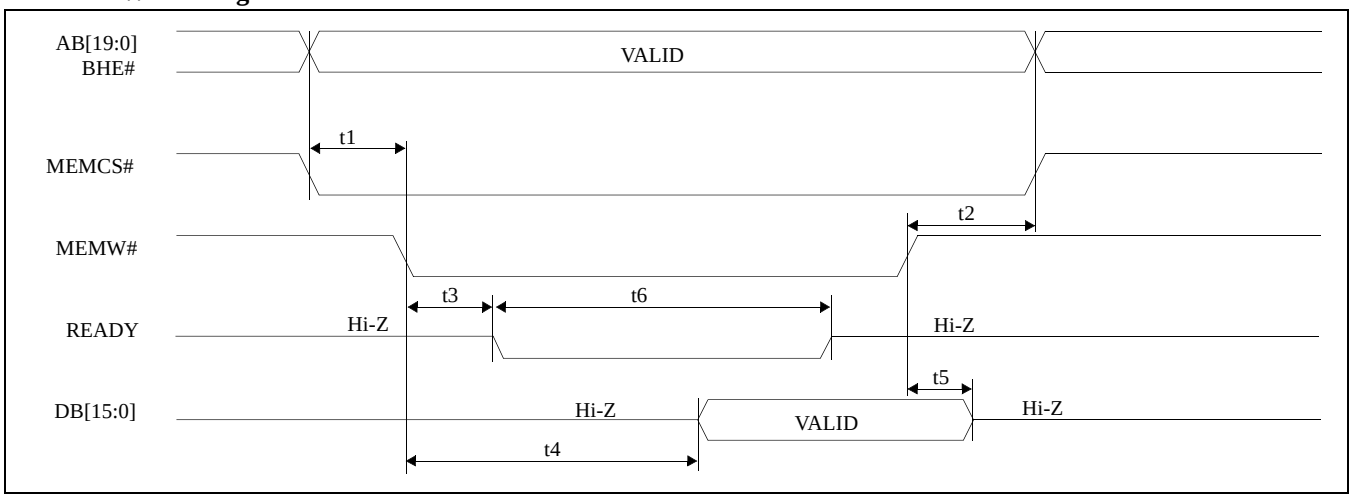


Figure 16: MEMW# Timing (Non-68000)

Table 7-7: MEMW# Timing (Non-68000)

		3V/3.3V			5V			
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**
t1	AB[19:0], BHE# and MEMCS# valid before MEMW# falling edge	0			0			ns
t2	AB[19:0], BHE# and MEMCS# hold from MEMW# rising edge	0			0			ns
t3	MEMW# falling edge to READY falling edge			30			20	ns
t4	MEMW# falling edge to DB[15:0] valid			120			140	ns
t5	DB[15:0] hold from MEMW# rising edge	0			0			ns
t6	READY negated pulse width			3.5* MCLK + 20			3.5* MCLK + 10	ns

Where MCLK period = $1/f_{OSC}$, or $2/f_{OSC}$, or $4/f_{OSC}$ depending on which mode the chip is in. (see section 9.2 and 9.3).

MEMR# Timing

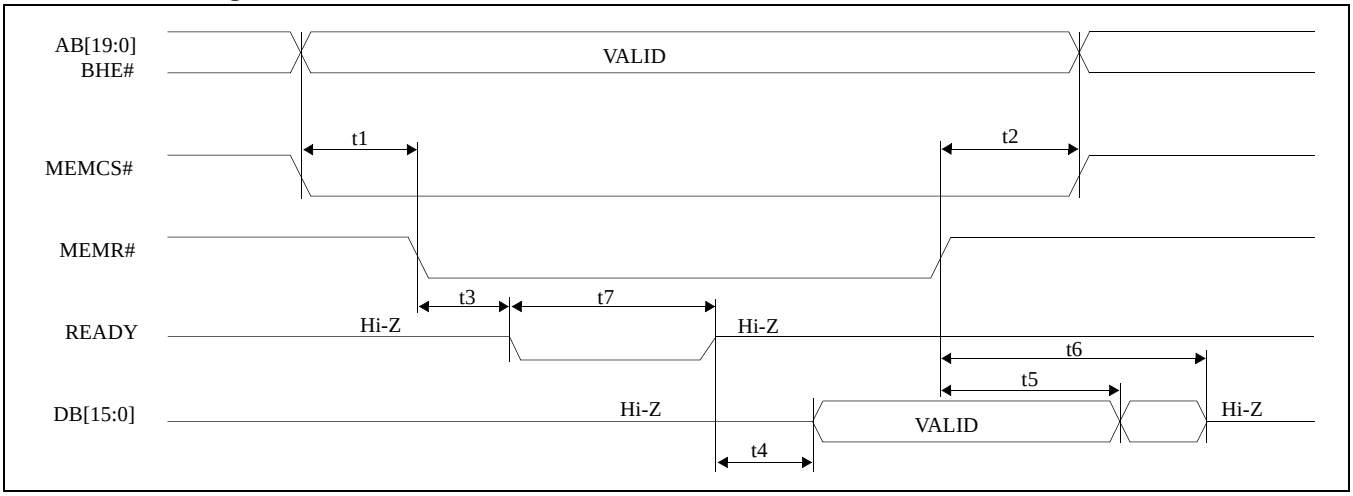


Figure 17: MEMR# Timing (Non-68000)

Table 7-8: MEMR# Timing (Non-68000)

		3V/3.3V			5V			
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**
t1	AB[19:0], BHE# and MEMCS# valid before MEMR# falling edge	0			0			ns
t2	AB[19:0], BHE# and MEMCS# hold from MEMR# rising edge	0			0			ns
t3	MEMR# falling edge to READY falling edge			30			20	ns
t4	READY rising edge to DB[15:0] valid			15			10	ns
t5	DB[15:0] hold from MEMR# rising edge			30			28	ns
t6	MEMR# rising edge to DB[15:0] hi-z delay			30			30	ns
t7	READY negated pulse width			3.5* MCLK + 30			3.5* MCLK + 10	ns

Where MCLK period = $1/f_{OSC}$, or $2/f_{OSC}$, or $4/f_{OSC}$ depending on which mode the chip is in. (see section 9.2 and 9.3)

7.2 Clock Input Requirements

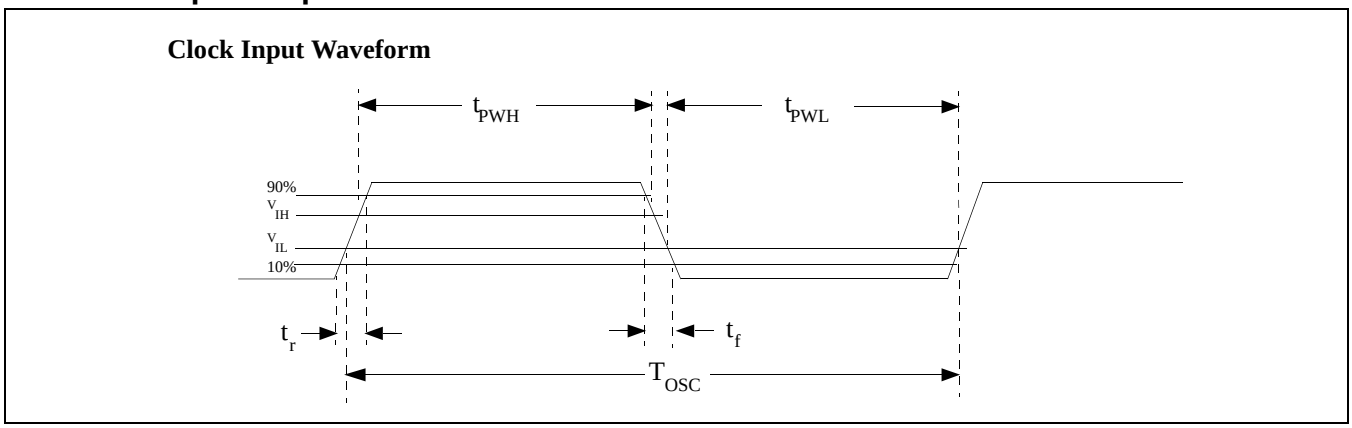


Figure 18: Clock Input Requirements

Table 7-9: Clock Input Requirements

Symbol	Parameter	Min	Typ	Max	Units
T_{OSC}	Input Clock Period (CLKI)	40			ns
t_{PWH}	Input Clock Pulse Width High (CLKI)	40		60	T_{OSC}
t_{PWL}	Input Clock Pulse Width Low (CLKI)	40		60	T_{OSC}
t_f	Input Clock Fall Time (10% - 90%)		5		ns
t_r	Input Clock Rise Time (10% - 90%)		5		ns

7.2.1 Recommended Clock Input

The nominal frequency must be calculated based on the formulas found in Frame Rate Calculation on page 61.

The crystal oscillator must be “fundamental mode” and have the following recommended RC load values:

$R_L = 2M\Omega \pm 5\%$
$C_L = 6.8$ pF

The figure below demonstrates both a crystal interface and an oscillator interface to the SED1352.

.

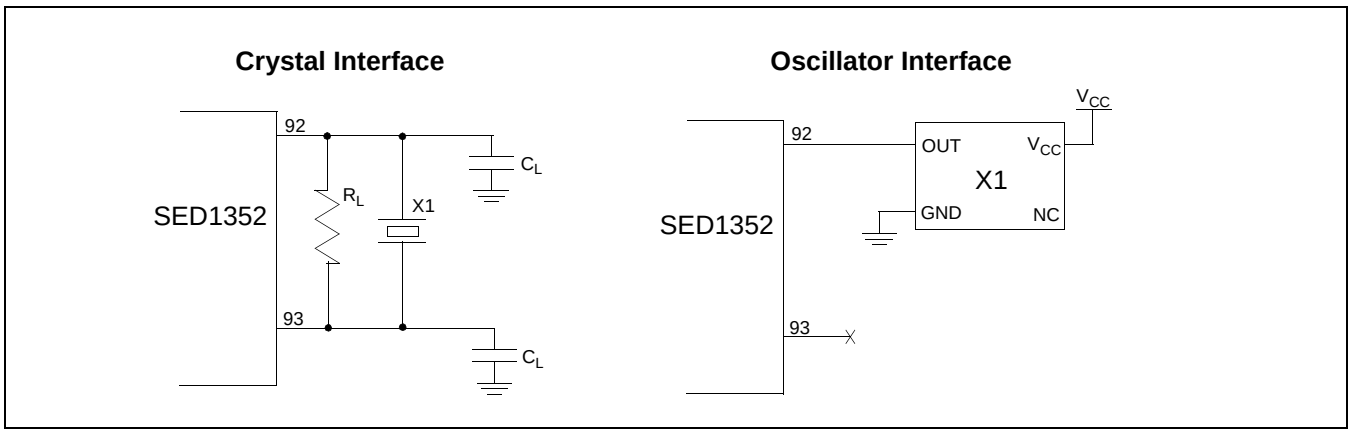


Figure 19: Recommended Clock Interface

7.3 Display Memory Interface Timing

7.3.1 Write Data to Display Memory

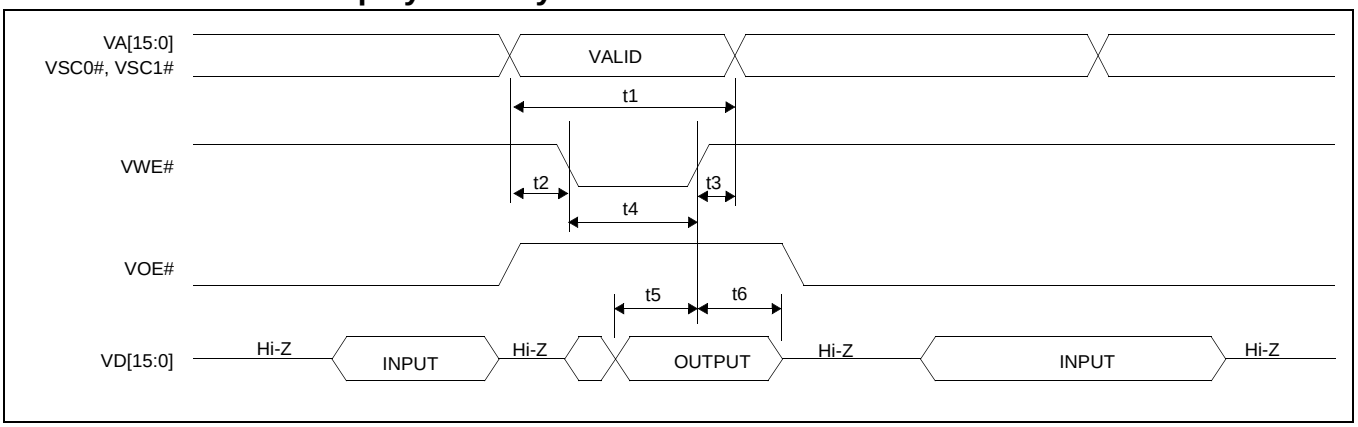


Figure 20: Write Data to Display Memory

Table 7-10: Write Data to Display Memory

		3V/3.3V			5V			
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**
t1	Address cycle time	MCLK - 10			MCLK - 10			ns
t2	VA[15:0], VCS0# and VCS1# valid before VWE# falling edge	MCLK/2 - 20			MCLK/2 - 10			ns
t3	VA[15:0], VCS0# and VCS1# hold from VWE# rising edge	0			0			ns
t4	Pulse width of VWE#	MCLK/2 - 5			MCLK/2 - 5			ns
t5	VD[15:0] setup to VWE# rising edge	MCLK/2 - 20			MCLK/2 - 20			ns
t6	VD[15:0] hold from VWE# rising edge	0			0			ns

Where MCLK period = $1/f_{OSC}$, or $2/f_{OSC}$, or $4/f_{OSC}$ depending on which mode the chip is in. (see section 9.2 and 9.3).

7.3.2 Read Data From Display Memory

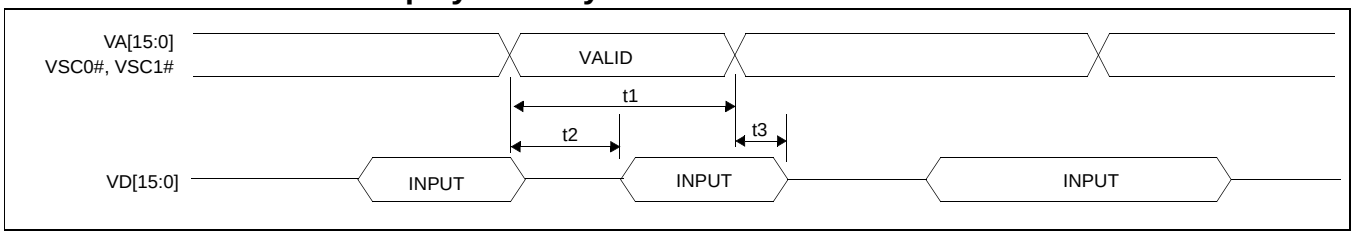


Figure 21: Read Data From Display Memory

Table 7-11: Read Data From Display Memory

		3V/3.3V			5V		
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
t1	Address cycle time	MCLK - 10			MCLK - 10		
t2	VA[15:0], VCS0# and VCS1# access time			MCLK - 50			MCLK - 30
t3	VD[15:0] hold time	0			0		

Where MCLK period = $1/f_{OSC}$, or $2/f_{OSC}$, or $4/f_{OSC}$ depending on which mode the chip is in. (See sections 9.2 and 9.3.)

7.4 LCD Interface Timing

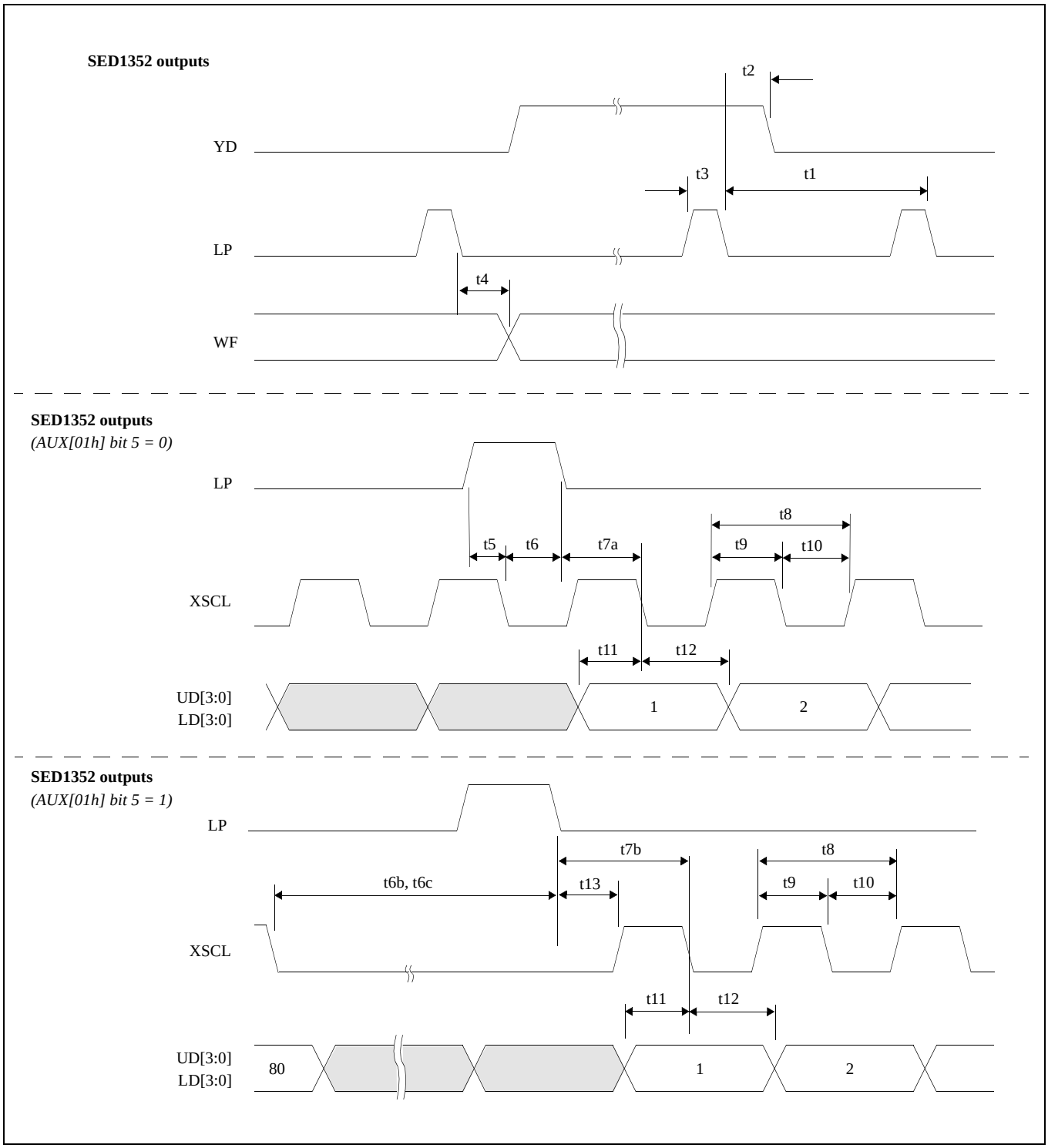


Figure 22: LCD Interface Timing

7.4.1 4-Bit Single LCD Interface Timing

Table 7-12: 4-Bit Single LCD Interface Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	LP period	HT - 24			ns
t2a	YD hold from LP negated (R1 bit 5 = 0)	$8t_{OSC}$ - 24			ns
t2b	YD hold from LP negated (R1 bit 5 = 1)	$13t_{OSC}$ - 24			ns
t3a	LP pulse width (R1 bit 5 = 0)	$6t_{OSC}$ - 24			ns
t3b	LP pulse width (R1 bit 5 = 1)	$5t_{OSC}$ - 24			ns
t4	WF delay from LP falling edge	0		20	ns
t5	LP setup to XSCL falling edge (R1 bit 5 = 0)	$2t_{OSC}$ - 24			ns
t6a	LP hold from XSCL falling edge (R1 bit 5 = 0)	$2t_{OSC}$ - 24			ns
t6b	XSCL falling edge to LP falling edge (R1 bit 5 = 1 only)	$13t_{OSC}$ - 24			ns
t7a	LP negated to XSCL falling edge (R1 bit 5 = 0)	$2t_{OSC}$ - 24			ns
t7b	LP negated to XSCL falling edge (R1 bit 5 = 1)	$7t_{OSC}$ - 24			ns
t8	XSCL period	$4t_{OSC}$ - 24			ns
t9	XSCL high width	$2t_{OSC}$ - 24			ns
t10	XSCL low width	$2t_{OSC}$ - 24			ns
t11	UD[3:0] setup to XSCL falling edge	$2t_{OSC}$ - 24			ns
t12	UD[3:0] hold from XSCL falling edge	$2t_{OSC}$ - 24			ns
t13a	LP negated to XSCL rising edge (R1 bit 5 = 0)	0			ns
t13b	LP negated to XSCL rising edge (R1 bit 5 = 1)	$5t_{OSC}$ - 24			ns

Where HT = (number of horizontal panel pixels + 16) * t_{OSC} , where $t_{OSC} = 1/f_{OSC}$.

7.4.2 8-Bit LCD Interface Timing

Table 7-13: 8-Bit LCD Interface Timing

Symbol	Parameter	Min	Typ	Max	Units
t1a	LP period (single panel mode)	HT - 24			ns
t1b	LP period (dual panel mode)	2*HT - 24			ns
t2a	YD hold from LP negated (R1 bit 5 = 0)	$8t_{OSC}$ - 24			ns
t2b	YD hold from LP negated (R1 bit 5 = 1)	$13t_{OSC}$ - 24			ns
t3a	LP pulse width (R1 bit 5 = 0)	$6t_{OSC}$ - 24			ns
t3b	LP pulse width (R1 bit 5 = 1)	$5t_{OSC}$ - 24			ns
t4	WF delay from LP falling edge	0		20	ns
t5	LP setup to XSCL falling edge (R1 bit 5 = 0)	$2t_{OSC}$ - 24			ns
t6a	LP hold from XSCL falling edge (R1 bit 5 = 0)	$4t_{OSC}$ - 24			ns
t6b	XSCL falling edge to LP falling edge - single panel mode (R1 bit 5 = 1 only)	$15t_{OSC}$ - 24			ns
t6c	XSCL falling edge to LP falling edge - dual panel mode (R1 bit 5 = 1 only)	$31t_{OSC}$ - 24			ns
t7a	LP negated to XSCL falling edge (R1 bit 5 = 0)	$4t_{OSC}$ - 24			ns
t7b	LP negated to XSCL falling edge (R1 bit 5 = 1)	$9t_{OSC}$ - 24			ns
t8	XSCL period	$8t_{OSC}$ - 24			ns
t9	XSCL high width	$4t_{OSC}$ - 24			ns
t10	XSCL low width	$4t_{OSC}$ - 24			ns
t11	UD[3:0], LD[3:0] setup to XSCL falling edge	$4t_{OSC}$ - 24			ns
t12	UD[3:0], LD[3:0] hold from XSCL falling edge	$4t_{OSC}$ - 24			ns
t13a	LP negated to XSCL rising edge (R1 bit 5 = 0)	0			ns
t13b	LP negated to XSCL rising edge (R1 bit 5 = 1)	$5t_{OSC}$ - 24			ns

Where HT = (number of horizontal panel pixels + 16) * t_{OSC} , where $t_{OSC} = 1/f_{OSC}$.

LCD Interface Pixel/Data Position

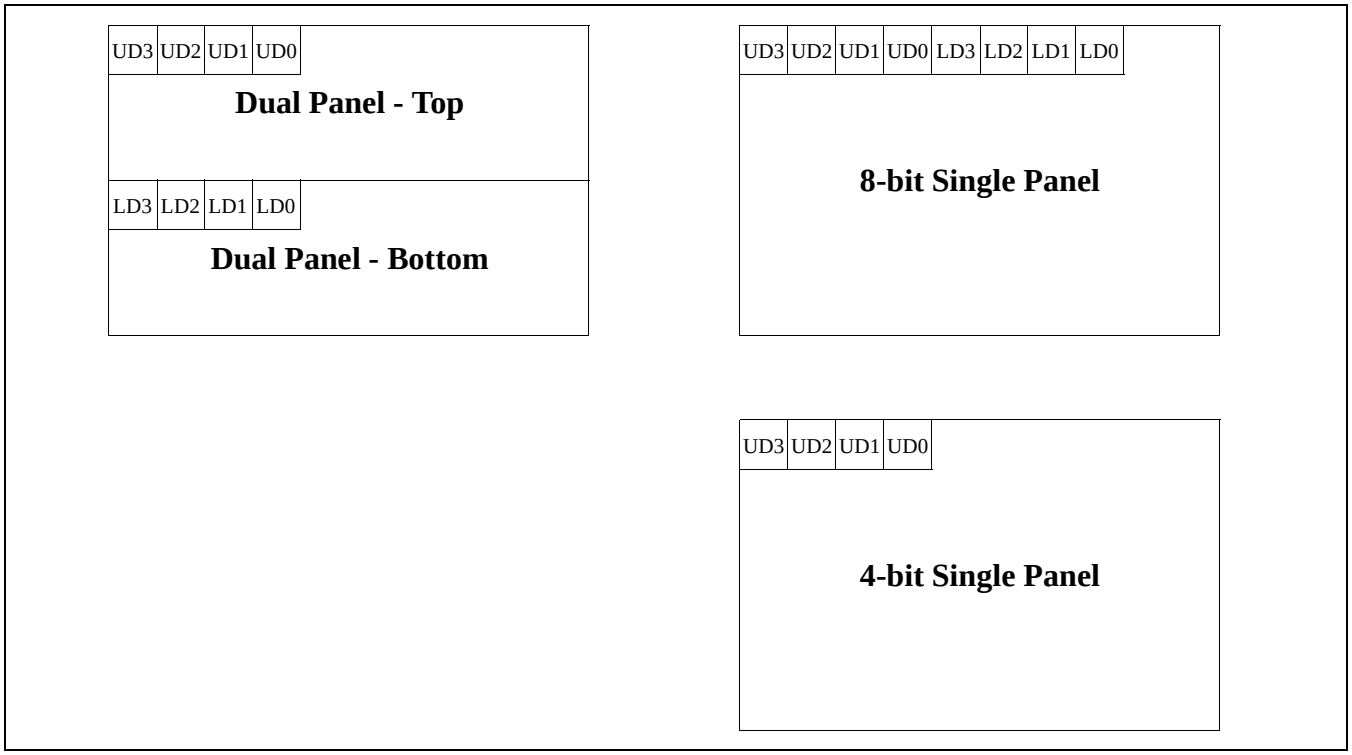


Figure 23: LCD Interface Pixel/Data Position

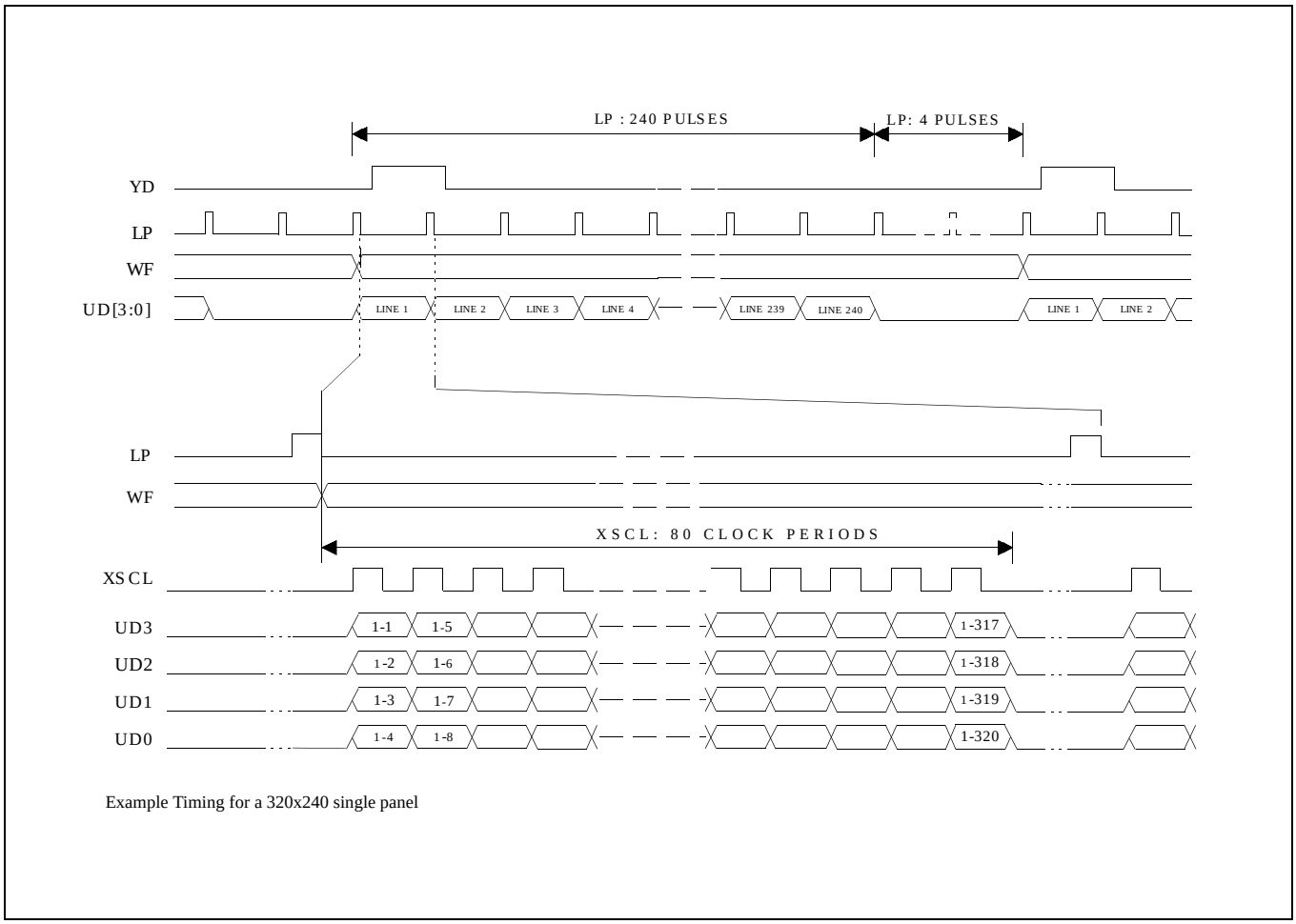
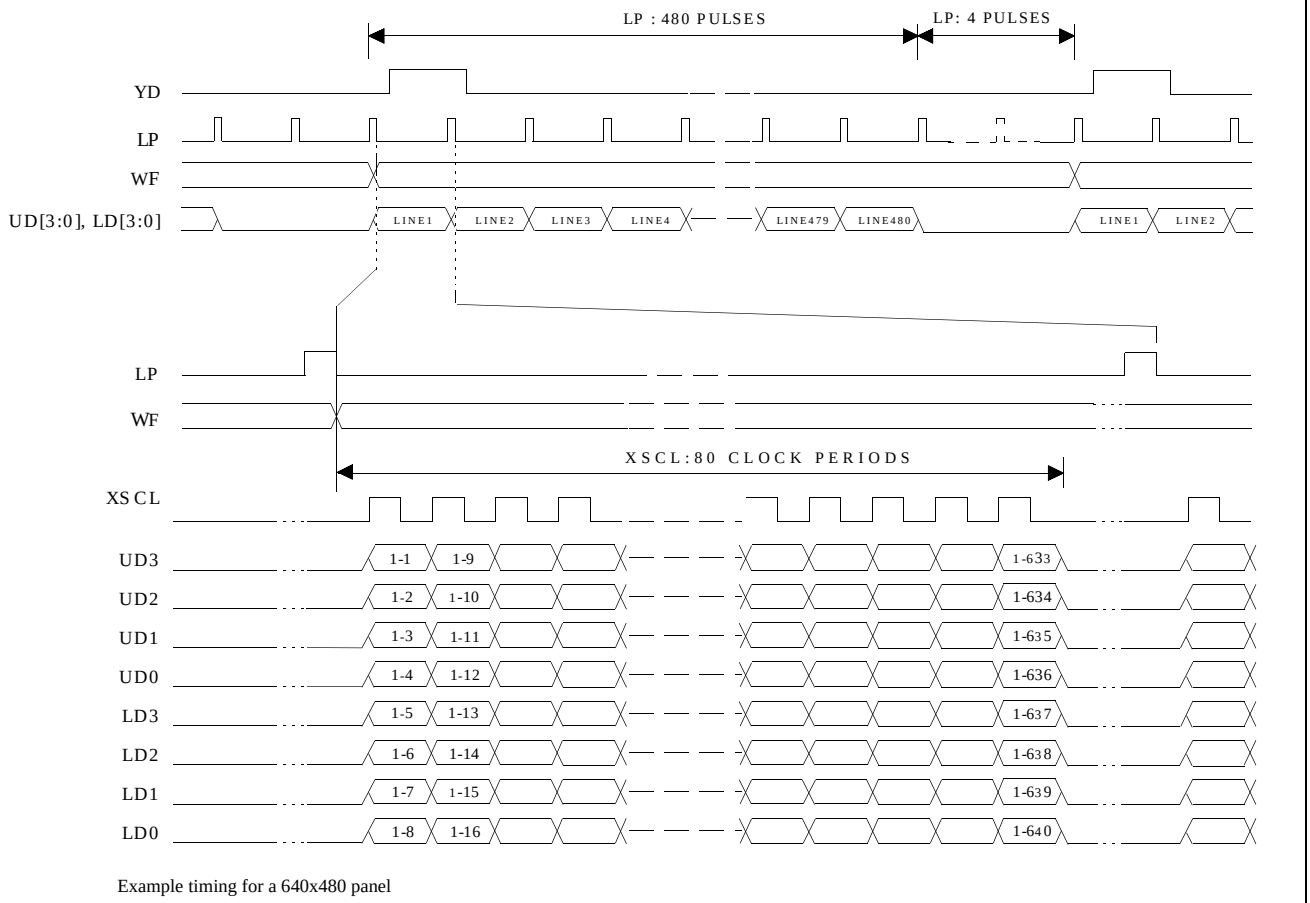
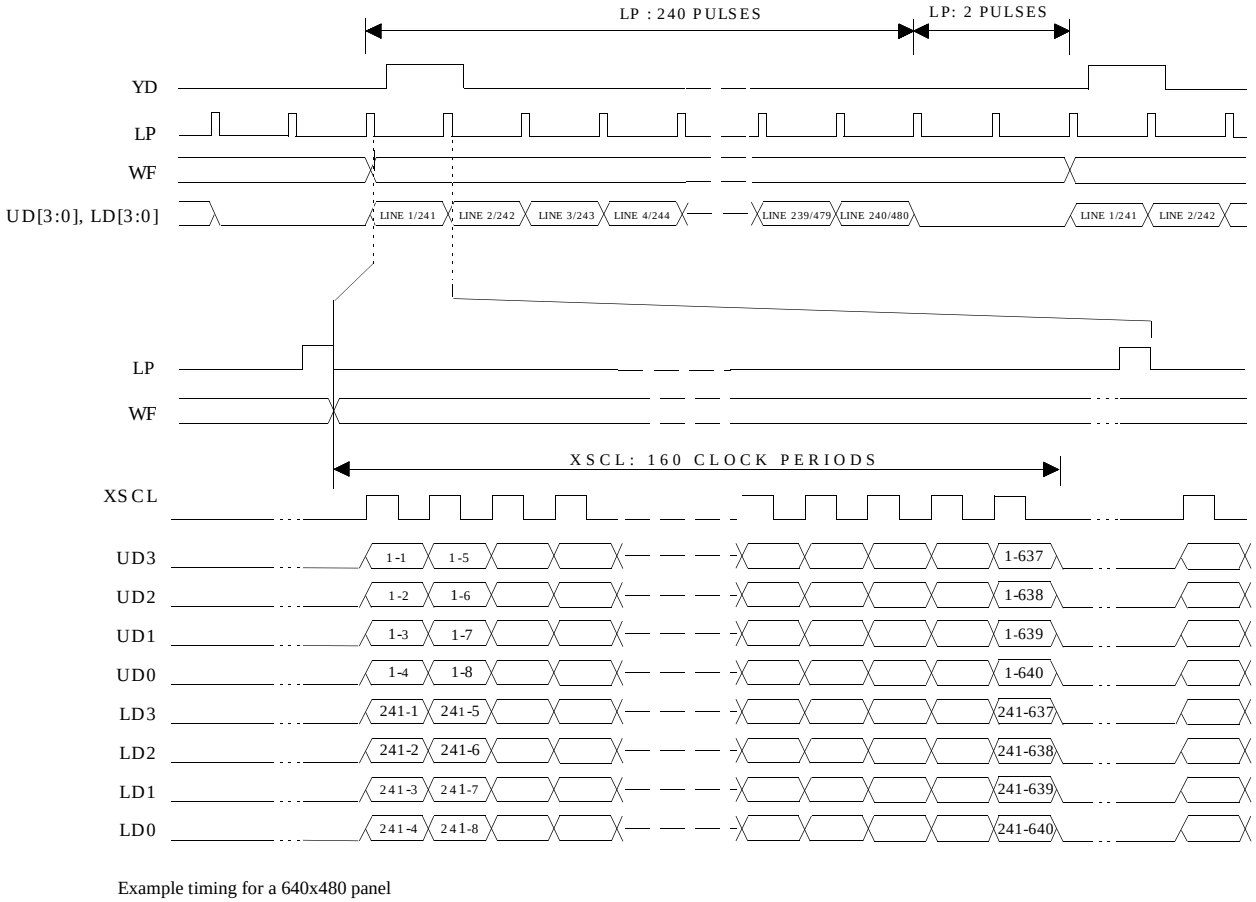


Figure 24: 4-Bit Single Monochrome Panel Timing



Example timing for a 640x480 panel

Figure 25: 8-Bit Single Monochrome Panel Timing



Example timing for a 640x480 panel

Figure 26: 8-Bit Dual Monochrome Panel Timing

8 HARDWARE REGISTER INTERFACE

The SED1352 is configured and controlled via 15 internal 8-bit registers. There are two ways to map these registers into the system I/O space.

1. Direct-mapping: Absolute I/O address = system address lines AB[3:0] + base I/O mapped address
   (where base I/O address is selected by VD7-VD12, see Table 5-6)

   This scheme requires 16 sequential I/O addresses starting from the I/O mapped base address selected by VD7-VD12 (see Table 5-6).

   To perform an I/O access:
   write data IOW {absolute I/O address}, {data}
   read data IOR {absolute I/O address}

2. Indexing: I/O address = internal index register bits [3:0]

   This scheme requires 2 sequential I/O addresses starting from the base address selected by VD4-VD12 (see Table 5-6).

   To perform an 8-bit I/O access:
   write index IOW {I/O mapped address}, {index} ; write the index of the register to be accessed
   then
   write data IOW {I/O mapped address +1}, {data} ; write data to the indexed register
   or
   read data IOR {I/O mapped address +1} ; read the indexed register

   To perform a 16-bit I/O access:
   write data IOW {I/O mapped address}, {index, data} ; write the index and data of the register to be accessed

   read data IOW {I/O mapped address}, {index} ; write to the indexed register
   IOR {I/O mapped address +1} ; read the indexed register

Note
Bits marked “n/o” should be set to 0 in the following registers.

8.1 Register Descriptions

AUX[00h] Test Register I/O address = 0000b, Read/Write							
Test Mode Enable	Reserved	Test Input Select Bit 2	Test Input Select Bit 1	Test Input Select Bit 0	Test Output Select Bit 2	Test Output Select Bit 1	Test Output Select Bit 0

bit 7 Test Mode Enable
When this bit = 0 normal operation is enabled. When this bit = 1 the chip is placed in a special test mode. The test input bits and test output bits (bits 6:0) are used to select various internal test functions.

bit 6 Reserved
During normal operation this bit must = 0.

bits 5-0 Test Mode Input Bits [2:0] and Output Bits [2:0]
When bit 7 = 1 these are the Test Input Select Input and Output bits. When bits 6 and 7 = 0 (normal operation) these bits may be used as read/write scratch registers.

AUX[01h] Mode Register I/O address = 0001b, Read/Write.							
DISP	Panel	Mask XSCL	LCDE	Gray Scale	LCD Data Width	Memory Interface	RAMS

bit 7 DISP
This bit selects display on or off. When this bit = 0, Display OFF is selected (LD0-3 and UD0-3 are forced to 0). When this bit = 1, Display ON is selected. This bit goes low on RESET.

bit 6 Panel
This bit selects the LCD panel configuration (single or dual). When this bit = 0, Single LCD panel drive is selected. When this bit = 1 Dual LCD panel drive is selected. This bit goes low on RESET.

bit 5 Mask XSCL
When this bit = 0 XSCL is not masked off during the horizontal non-display period. When this bit = 1 XSCL is masked off during the horizontal non-display period. This bit goes low on RESET.

bit 4 LCDE
The state of this pin determines the state of output pin 82, LCDENB, and is intended for control of an external LCDBIAS power supply. However, this pin can be used as a General I/O pin if desired. When LCDE = 0, LCDENB is forced low. When LCDE = 1, LCDENB is forced high. This bit goes low on RESET.

bit 3 Gray Scale
Selects between 16-level or 4-level gray scale display. When this bit = 1, 16 gray shades are displayed (4 bits/pixel). When this bit = 0, 4 gray shades are displayed (2 bits/pixel). This bit goes low on RESET.

bit 2 LCD Data Width
Selects between 4-bit and 8-bit display data widths for single LCD mode. When this bit = 1, 8-bit data transfer width is enabled. When this bit = 0, 4-bit data transfer width is enabled. In dual panel mode the data transfer width is forced to 4 bits per panel. This bit goes low on RESET.

bit 1 Memory Interface
This bit selects between the 8-bit or 16-bit memory interface. When this bit = 0, the 16-bit memory interface is selected. When this bit = 1, the 8-bit memory interface is selected. If 16-bit bus interface is selected (VD0 = 1 on RESET), the Memory Interface bit is forced to 0 internally (16-bit). This bit goes low on RESET.

bit 0 RAMS
This bit configures the display memory address lines for an 8-bit memory interface system. When this bit = 0, addressing for 8Kx8 SRAM on an 8-bit display memory data bus interface is selected. When this bit = 1, addressing for 32Kx8 SRAM on an 8-bit display memory data bus interface is selected. This bit goes low on RESET. This bit is ignored for a 16-bit memory interface.

AUX[02h] Line Byte Count Register (LSB) I/O address = 0010b, 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]
These are the 8 LSB of the 9 bit Total Display Line Count and represent the number of scan lines -1, to a maximum value of 3FFh or 1024 scan lines. Line Byte Count Bit 8 is located in register AUX[05h] and is ignored in the 16-bit memory interface. To calculate the Line Byte Count use the following formula:

$$LineByteCount = \left(\frac{BitsPerPixel}{MemoryInterfaceWidth} \times HorizontalResolution\right) - 1$$

Example:
To calculate the Line Byte Count for 640 horizontal pixels with 16 gray shades (4 bits-per-pixel) and 16-bit memory interface:

$$LineByteCount = \frac{4BitsPerPixel}{16Bits} \times 640 - 1 = 159$$

The following two tables summarize the maximum value of the Line Byte Count Register for different display modes and display memory interface.

Table 8-1: Maximum Value of Line Byte Count Register - 8-Bit Display Memory Interface

Display Mode	Maximum Value of Line Byte Count Register	Corresponding Maximum Number of Pixels in One Display Line
4-level gray shades	0FFh	256 x 4 = 1024
16-level gray shades	1FFh	512 x 2 = 1024

Table 8-2: Maximum Value of Line Byte Count Register - 16-Bit Display Memory Interface

Display Mode	Maximum Value of Line Byte Count Register	Corresponding Maximum Number of Pixels in One Display Line
4-level gray shades	0FFh	256 x 8 = 2048
16-level gray shades	0FFh	256 x 4 = 1024

AUX[03h] Line Byte Count Power Save Register (MSB) I/O address = 0011b, Read/Write							
PS Bit 1	PS Bit 0	LCD Signal State	LUT Bypass	n/a	n/a	n/a	Line Byte Count Bit 8

bits 7-6 PS Bits [1:0]
Selects the Power Save Modes as shown in the following table. The PS bits [1:0] go low on RESET.

Table 8-3: Power Save Mode Selection

PS1	PS0	Mode Activated
0	0	Normal Operation
0	1	Power Save Mode 1
1	0	Power Save Mode 2
1	1	Reserved

Refer to Power Save Modes (PSM 1) on page 55 for a complete Power Save Mode description.

bit 5 LCD Signal State
When this bit = 0, all LCD interface signals are forced low during Power Save modes. When this bit = 1, all LCD interface signals are forced to a high impedance (Hi-Z) state during Power Save modes. This bit goes low on RESET.

bit 4 LUT Bypass
When the LUT Bypass bit = 0 the Look-Up Table is used for display data output. When this bit = 1, the Look-Up Table is bypassed for display data output (for power save purposes). The LUT Bypass bit goes low on RESET.

bit 0 Line Byte Count Bit 8
This is the MSB of the number of bytes to be fetched per display line minus 1 (see AUX[02h]). This bit only has effect when in 16 gray shades with 8-bit memory interface. This bit is ignored in the 16-bit memory interface.

.

AUX[04h] Total Display Line Count Register (LSB) (Vertical Total) I/O address = 0100b, Read/Write.							
Total Display Line Count Bit 7	Total Display Line Count Bit 6	Total Display Line Count Bit 5	Total Display Line Count Bit 4	Total Display Line Count Bit 3	Total Display Line Count Bit 2	Total Display Line Count Bit 1	Total Display Line Count Bit 0

bits 7-0 Total Display Line Count Bits [7:0]
These are the 8 LSB of the 10 bit Total Display Line Count and represent the number of scan lines -1, to a maximum value of 3FFh or 1024 scan lines.

In single panel mode:

$$TotalDisplayLineCount = NumberOfDisplayLines - 1$$

In dual panel mode:

$$TotalDisplayLineCount = \left(\frac{NumberOfDisplayLines}{2}\right) - 1$$

Note
Note that the value programmed partially determines the frame period, and hence affects display duty cycle. Bits 8 and 9 are located in the following register (AUX[05h]).

AUX[05h] Total Display Line Count Register (MSB) and WF Count Register I/O address = 0101b, Read/Write							
WF Count Bit 5	WF Count Bit 4	WF Count Bit 3	WF Count Bit 2	WF Count Bit 1	WF Count Bit 0	Total Display Line Count Bit 9	Total Display Line Count Bit 8

bits 7-2 WF Count Bits [5:0]
These bits are used to adjust the WF output signal period. The binary value stored in these bits represents the number of LP pulses -1 between toggles of the WF output. The power up reset value of these bits is 0, which causes the WF output to toggle every frame. When values of 01h to 3Fh are programmed into these bits, the results are WF toggling every 1+n LP pulses, where n is the value programmed.

bits 1-0 Total Display Line Count Bits [9:8]
These are the two MSB of the Total Display Line Count Register, AUX[04h].

AUX[06h] Screen 1 Display Start Address Register (LSB) I/O address = 0110b, Read/Write.							
Screen 1 Display Start Addr Bit 7	Screen 1 Display Start Addr Bit 6	Screen 1 Display Start Addr Bit 5	Screen 1 Display Start Addr Bit 4	Screen 1 Display Start Addr Bit 3	Screen 1 Display Start Addr Bit 2	Screen 1 Display Start Addr Bit 1	Screen 1 Display Start Addr Bit 0

AUX[07h] Screen 1 Display Start Address Register (MSB) I/O address = 0111b, Read/Write.							
Screen 1 Display Start Addr Bit 15	Screen 1 Display Start Addr Bit 14	Screen 1 Display Start Addr Bit 13	Screen 1 Display Start Addr Bit 12	Screen 1 Display Start Addr Bit 11	Screen 1 Display Start Addr Bit 10	Screen 1 Display Start Addr Bit 9	Screen 1 Display Start Addr Bit 8

AUX[06h] bits 7-0 Screen 1 Display Start Address Bits [15:0]
AUX[07h] bits 7-0 These 16 bits determine the Screen 1 Display Start Address. In an 8-bit memory configuration these bits set the 16-bit start address (i.e., byte access). In a 16-bit memory configuration these are the 16 most significant bits of a 17-bit start address (i.e., word access).

The Screen 1 Display Start Address is the memory address corresponding to the first displayed pixel (top left corner). In a dual panel configuration, screen 1 refers to the upper half of the display. While in a single panel configuration, screen 1 refers to the first screen of the Split Screen Display feature where two different images (screen 1 and screen 2) can be displayed at the same time on one display.

Note
The absolute address into display memory is determined by the Memory Mapping Address which is set by VD13 - VD15 (see Table 5-6, “Summary of Power On / Reset Options,” on page 25).

AUX[08h] Screen 2 Display Start Address Register (LSB) I/O address = 1000b, Read/Write.							
Screen 2 Display Start Addr Bit 7	Screen 2 Display Start Addr Bit 6	Screen 2 Display Start Addr Bit 5	Screen 2 Display Start Addr Bit 4	Screen 2 Display Start Addr Bit 3	Screen 2 Display Start Addr Bit 2	Screen 2 Display Start Addr Bit 1	Screen 2 Display Start Addr Bit 0

AUX[09h] Screen 2 Display Start Address Register (MSB) I/O address = 1001b, Read/Write.							
Screen 2 Display Start Addr Bit 15	Screen 2 Display Start Addr Bit 14	Screen 2 Display Start Addr Bit 13	Screen 2 Display Start Addr Bit 12	Screen 2 Display Start Addr Bit 11	Screen 2 Display Start Addr Bit 10	Screen 2 Display Start Addr Bit 9	Screen 2 Display Start Addr Bit 8

AUX[08h] bits 7-0 Screen 2 Display Start Address Bits [15:0]

AUX[09h] bits 7-0 These 16 bits determine the Screen 2 Display Start Address. In an 8-bit memory configuration these bits set the 16-bit start address (i.e., byte access). In a 16-bit memory configuration these are the 16 most significant bits of a 17-bit start address (i.e., word access).

In a dual panel configuration, screen 2 refers to the lower half of the display. The Screen 2 Display Start Address is the memory address corresponding to first displayed pixel in the first line of the lower half of the display. If screen 2 is started right after screen 1, the Screen 2 Display Start Address is calculated with the following formula.

$$Screen2DisplayStartAddress(hex) =$$

$$\frac{(ImageHorizontalResolution) \times (ImageVerticalResolution) \times (BytesPerPixel)}{2 \times \left(\frac{MemoryInterfaceWidth}{8}\right)} + Screen1DisplayStartAddress$$

In a single panel configuration, screen 2 refers to the second screen of the Split Screen Display Feature where two different images (screen 1 and screen 2) can be displayed at the same time on one display. The Screen 2 Display Start Address is the memory address corresponding to the first pixel of the second image stored in display memory. To display screen 2 refer to AUX[0Ah] Screen 1 Display Line Count Register (LSB) below.

AUX[0Ah] Screen 1 Display Line Count Register (LSB) I/O address = 1010b, Read/Write.							
Screen 1 Display Line Count Bit 7	Screen 1 Display Line Count Bit 6	Screen 1 Display Line Count Bit 5	Screen 1 Display Line Count Bit 4	Screen 1 Display Line Count Bit 3	Screen 1 Display Line Count Bit 2	Screen 1 Display Line Count Bit 1	Screen 1 Display Line Count Bit 0

AUX[0Bh] Screen 1 Display Line Count Register (MSB) I/O address = 1011b, Read/Write.							
n/a	n/a	n/a	n/a	n/a	n/a	Screen 1 Display Line Count Bit 9	Screen 1 Display Line Count Bit 8

AUX[0Ah] bits 7-0 Screen 1 Display Line Count Bits [9:0]
AUX[0Bh] bits 1-0 These bits are the eight LSB of a 10-bit value used to determine the number of lines displayed for screen 1. The remaining lines will automatically display from the Screen 2 Display Start Address. The 10-bit value programmed is the number of display lines -1.

This register is used to enable the split screen display feature (single panel only) where two different images can be displayed at the same time on one display.

For example; AUX[0Ah] = 20h for a 320x240 display system. The display will display 20h+1 = 33 lines on the upper part of the screen as dictated by the Screen 1 Display Start Address Registers (AUX[06h] and AUX[07h]), and 240 - 33 = 207 lines will be displayed on the lower part of the screen as dictated by the Screen 2 Display Start Address Registers (AUX[08h] and AUX[09h]).

Two different images can be displayed when using a dual panel configuration by changing the Screen 2 Display Start Address. However, by using this method screen 2 is limited to the lower half of the display.

This register is ignored in dual panel mode.

AUX[0Dh] Address Pitch Adjustment Register I/O address = 1101b, Read/Write.							
Addr Pitch Adjustment Bit 7	Addr Pitch Adjustment Bit 6	Addr Pitch Adjustment Bit 5	Addr Pitch Adjustment Bit 4	Addr Pitch Adjustment Bit 3	Addr Pitch Adjustment Bit 2	Addr Pitch Adjustment Bit 1	Addr Pitch Adjustment Bit 0

bits 7-0 Addr Pitch Adjustment Bits [7:0]
These bits set the numerical difference between the last address of a display line, and the first address in the following line.

If the Address Pitch Adjustment is not equal to zero, then a virtual screen is formed. The size of the virtual screen is only limited by the available display memory. The actual display output is a window that is part of the whole image stored in the display memory. For example, with 128K of display memory, a 640x400 16-gray image can be stored. If the output display size is 320x240, then the whole image can be seen by changing display starting addresses through AUX[06h] and [07h], and AUX[08h] and [09h]. Note that a virtual screen can be produced on either a single or dual panel.

In 8-bit memory interface, if the Address Pitch Adjustment is not equal to zero, then a virtual screen with a line length of (Line Byte Count +AUX[0Dh]+1) bytes is created, with the display reflecting the contents of a window (Line Byte Count+1) bytes wide. The position of the window on the virtual screen is determined by AUX[06h] and [07h], and AUX[08h] and [09h].

In 16-bit memory interface, if the Address Pitch Adjustment is not equal to zero, then a virtual screen with a line length of 2(Line Byte Count +AUX[0Dh]+1) bytes is created, with the display reflecting the contents of a window 2(Line Byte Count+1) bytes wide. The position of the window on the virtual screen is determined by AUX[06h] and [07h], and AUX[08h] and [09h].

.

AUX[0Eh] Look-Up Table Address Register I/O address = 1110b, Read/Write							
Bank Bit 1	Bank Bit 0	ID Bit (Read Only)	ID Bit (Read Only)	Palette Address Bit 3	Palette Address Bit 2	Palette Address Bit 1	Palette Address Bit 0

The SED1352 has one internal 16 position, 4-bit wide Look-Up Table (palette). The 4-bit value programmed into each table position determines the output gray shade/weighting of display data.

The Look-Up Table can be arranged in two different configurations. Refer to Table 27, “4-Level Gray-Shade Mode Look-Up Table Architecture,” on page 54 for formats.

bits 7-6 Bank Bits [1:0]
In 4-level gray mode (2-bits/pixel), the 16 position palette is arranged into four, 4 position “banks”. These two bits control which bank is currently selected. These bits have no effect in 16-level gray mode (4-bits/pixel).

bits 5-4 ID Bits
After power on or hardware reset, these bits can be read to identify the current revision of the SED1352. These same bits are used to identify the pin compatible SED1352F0x and would only be used in system implementations where common software is utilized. As these bits are R/W they must be read before being written in order to be used as ID bits.

Table 8-4: ID Bit Usage

	Chip	**Aux[0Eh]**	
		bit 5	bit 4
Power On or RESET	SED1353	0	0
	F352	0	1
	SED1352F0B/F1B/D0B	1	0
	SED1352F0A	1	1

bits 3-0 Palette Address Bits [3:0]
These 4 bits provide a pointer into the 16 position Look-Up Table currently selected for CPU R/W access.

Note
The Look-Up Table configuration (e.g. 1/2 banks) does not affect the R/W access from the CPU as all 16 positions can be accessed sequentially.

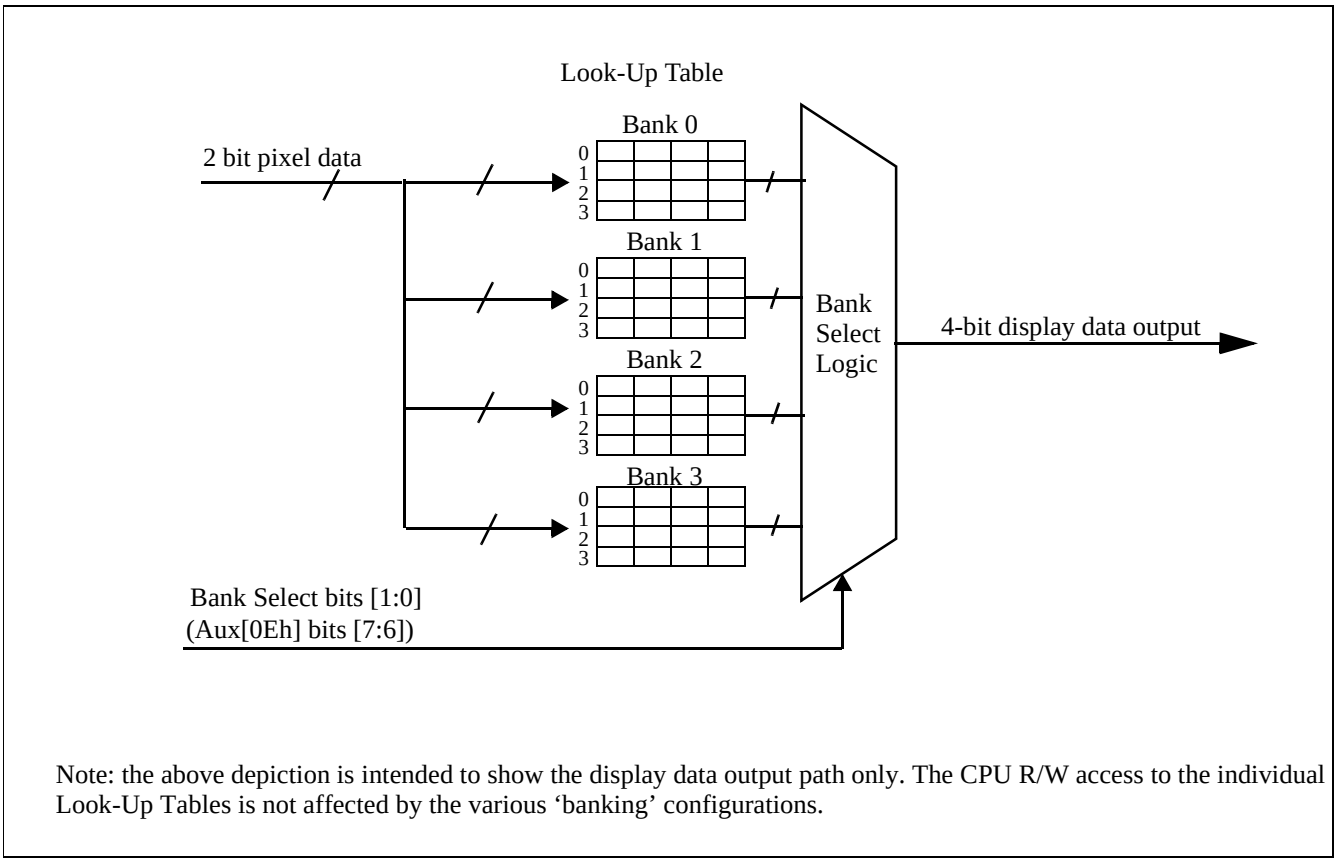
AUX[0Fh] Look-Up Table Data Register I/O address = 1111b, Read/Write.							
n/a	n/a	n/a	n/a	Palette Data Bit 3	Palette Data Bit 2	Palette Data Bit 1	Palette Data Bit 0

bits 3-0 Palette Data Bits [3:0]
These 4-bits are the gray shade values used for display data output. They are programmed into the 4-bit Look-Up Table (palettes) positions pointed to by Palette Address bits [3:0].

For example: in a 16-level gray shade display mode, a data value of 0001b (4-bits / pixel) will point to Look-Up Table position one and display the 4-bit gray shade corresponding to the value programmed into that location.

8.2 Look-Up Table Architecture

8.2.1 4-Level Gray Shade Mode



Note: the above depiction is 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.

Figure 27: 4-Level Gray-Shade Mode Look-Up Table Architecture

8.2.2 16-Level Gray Shade Mode

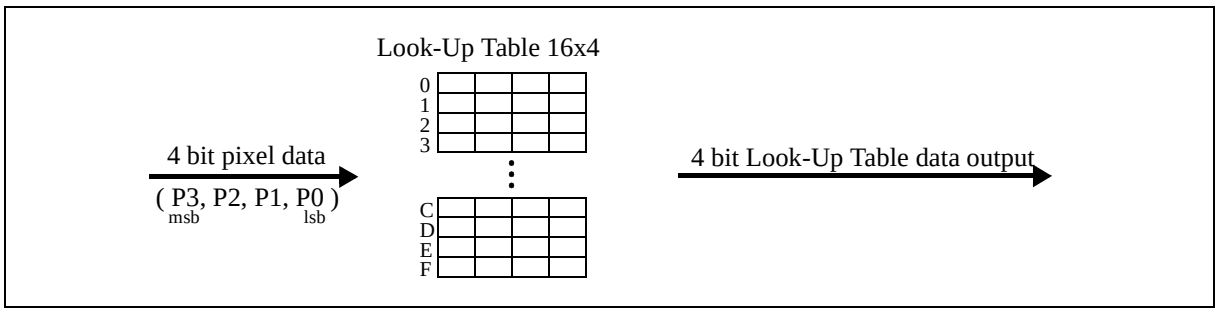


Figure 28: 16-Level Gray-Shade Mode Look-Up Table Architecture

8.3 Power Save Modes (PSM 1)

Two software-controlled Power Save Modes have been incorporated into the SED1352 to accommodate the important need for power reduction in the hand-held devices market. These modes can be enabled by setting the 2 Power Save bits (AUX[03h] bits 7-6).

The various settings are:

Table 8-5: Power Save Mode Selection

Bit 7	Bit 6	Mode Activated
0	0	Normal Operation
0	1	Power Save Mode 1
1	0	Power Save Mode 2
1	1	Reserved

8.3.1 Power Save Mode 1 (PSM1)

Power Save Mode 1 has two states. Initially when set, the SED1352 enters State 1. If no valid memory cycle is detected within 1, 2, or 4 clocks (input clock frequency dependent), the chip will enter State 2. The number of clocks of inactivity before entering State 2 is dependent on the display memory interface and the number of gray shades.

State 1

- I/O read/write of all registers allowed
- Memory read/write allowed
- LCD outputs are either forced low (AUX[03h] bit 5=0), or high impedance (AUX[03h] bit 5=1)

State 2

The same as State 1 as well as:

- Master clock for display memory access is disabled

Once a valid memory read/write cycle is detected, the SED1352 returns to State 1 where the MPU access is serviced. The transition from going from State 2 to State 1 requires 1, 2, or 4 clocks (as described above).

8.3.2 Power Save Mode 2 (PSM2)

- I/O read/write of all registers allowed
- Memory read/write is disabled
- Master clock for display memory access is disabled
- LCD outputs are either forced low (AUX[03h] bit 5=0), or high impedance (AUX[03h] bit 5=1)
- Internal oscillator is disabled.

8.3.3 Power Save Mode Function Summary

Table 8-6: Power Save Mode Function Summary

Function	Power Save Mode (PSM)			
	Normal (Active)	PSM1		PSM2
		State 1	State 2	
Display Active?	Yes	No	No	No
I/O Access Possible?	Yes	Yes	Yes	Yes
Memory Access Possible?	Yes	Yes	No	No
Sequence Controller Running?	Yes	No	No	No
Internal Oscillator Disabled?	No	No	No	Yes

8.3.4 Pin States in Power Save Modes

Table 8-7: Pin States in Power Save Modes

Pin	Pin State			
	Normal (Active)	PSM1		PSM2
		State 1	State 2	
UD[3:0], LD[3:0], LP, XSCL, YD, WF *(Note 1)*	Active	High Impedance	High Impedance	High Impedance
UD[3:0], LD[3:0], LP, XSCL, YD, WF *(Note 2)*	Active	Forced Low	Forced Low	Forced Low
AB[19:0], DB[15:0]	Active	Active	Active	Active
IOR#, IOW#	Active	Active	Active	Active
MEMR#, MEMW#	Active	Active	Active	Active
RESET	Active	Active	Active	Active

Note
Internal Register AUX[03h], bit 5 = 1.
Internal Register AUX[03h], bit 5 = 0.

9 DISPLAY MEMORY INTERFACE

9.1 SRAM Configurations Supported

9.1.1 8-Bit Mode

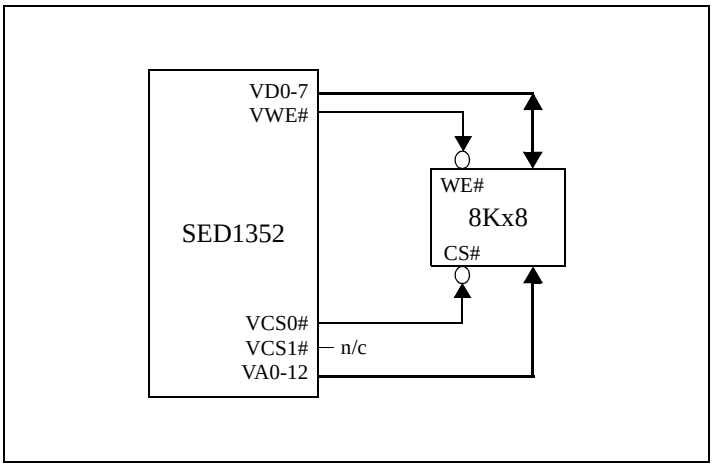


Figure 29: 8-Bit Mode - 8K bytes SRAM

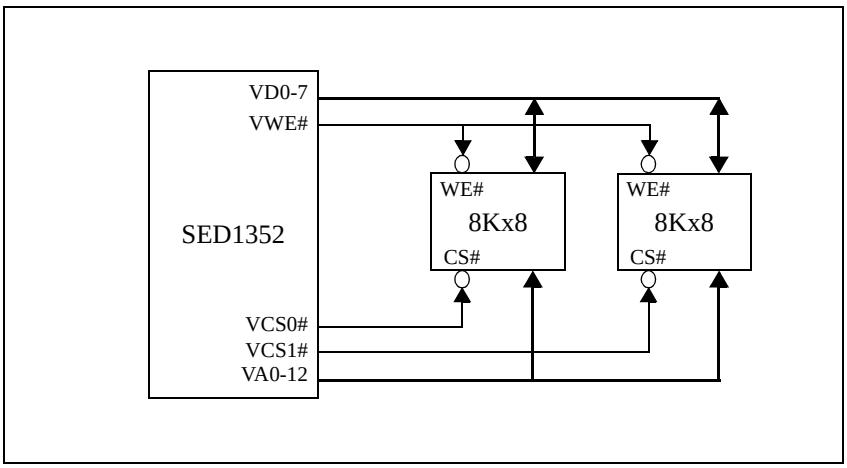


Figure 30: 8-Bit Mode - 16K bytes SRAM
(Requires AUX[01h] bit 0 = 0)

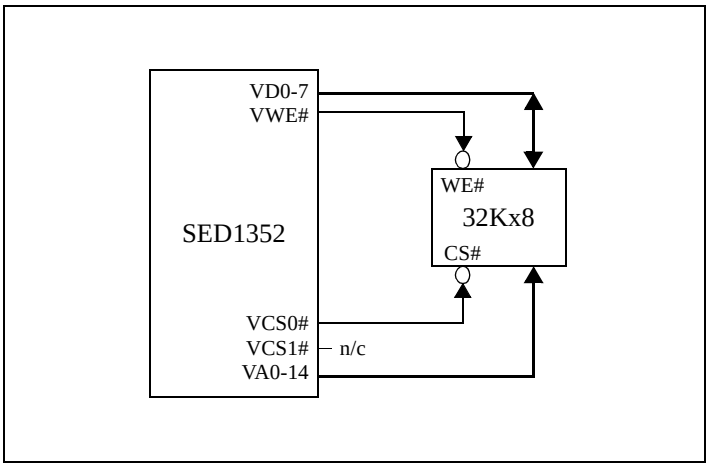


Figure 31: 8-Bit Mode - 32K bytes SRAM
(Requires AUX[01h] bit 0 = 1)

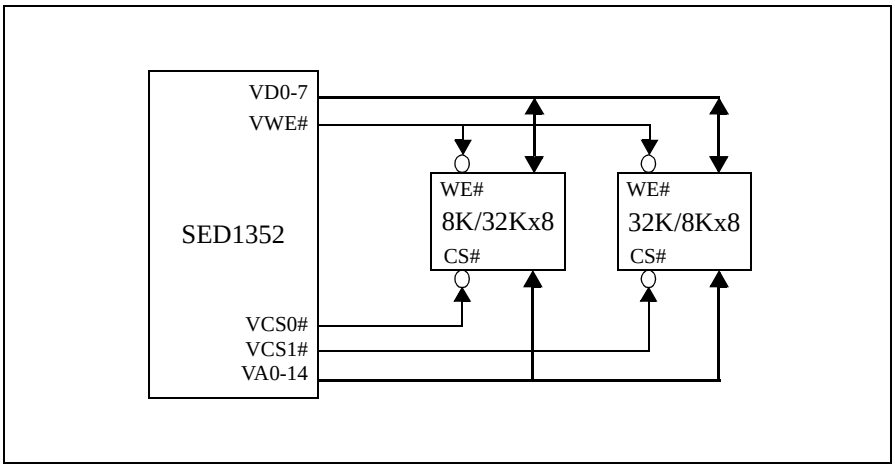


Figure 32: 8-Bit Mode - 40K bytes SRAM
[either (8Kx8 + 32Kx8) requiring AUX[01h] bit 0 = 0 or (32Kx8 + 8Kx8) requiring AUX[01h] bit 0 = 1]

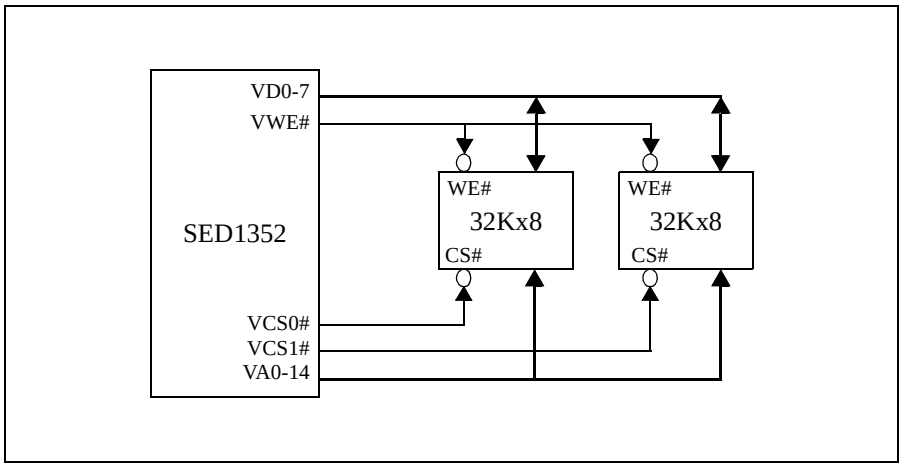


Figure 33: 8-Bit Mode - 64K bytes SRAM
(Requires AUX[01h] bit 0 = 1)

9.1.2 16-Bit Mode

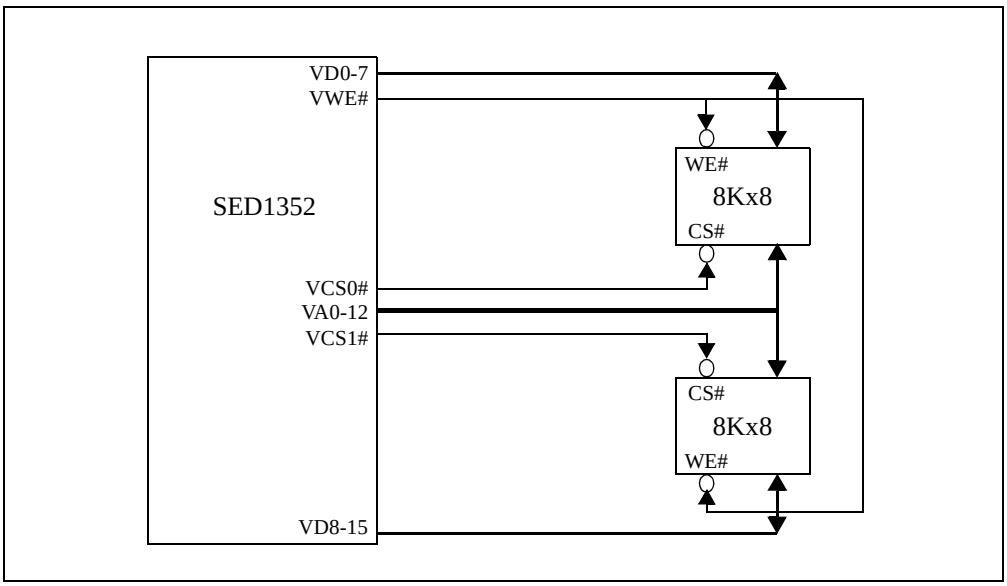


Figure 34: 16-Bit Mode - 16K bytes SRAM

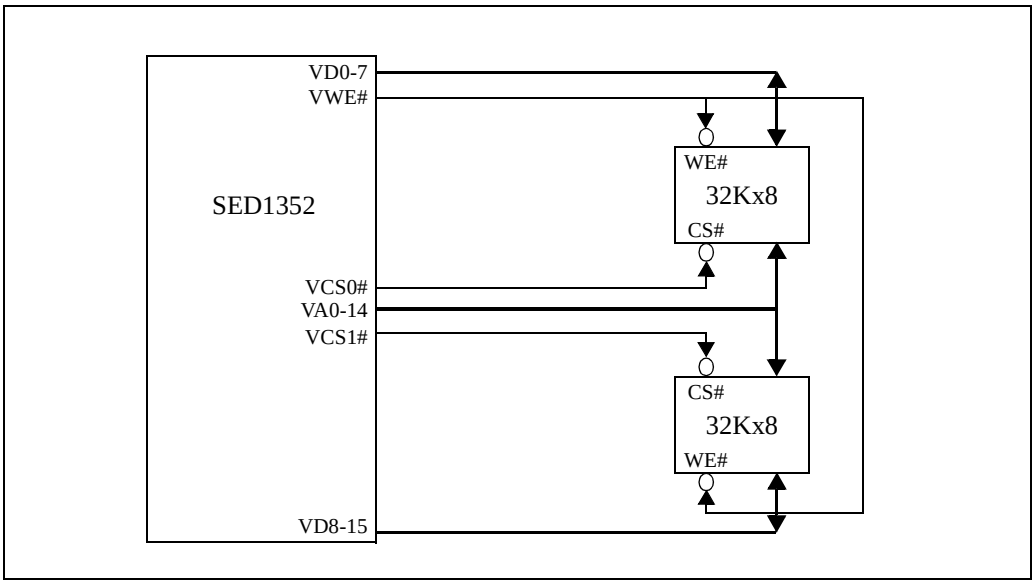


Figure 35: 16-Bit Mode - 64K bytes SRAM

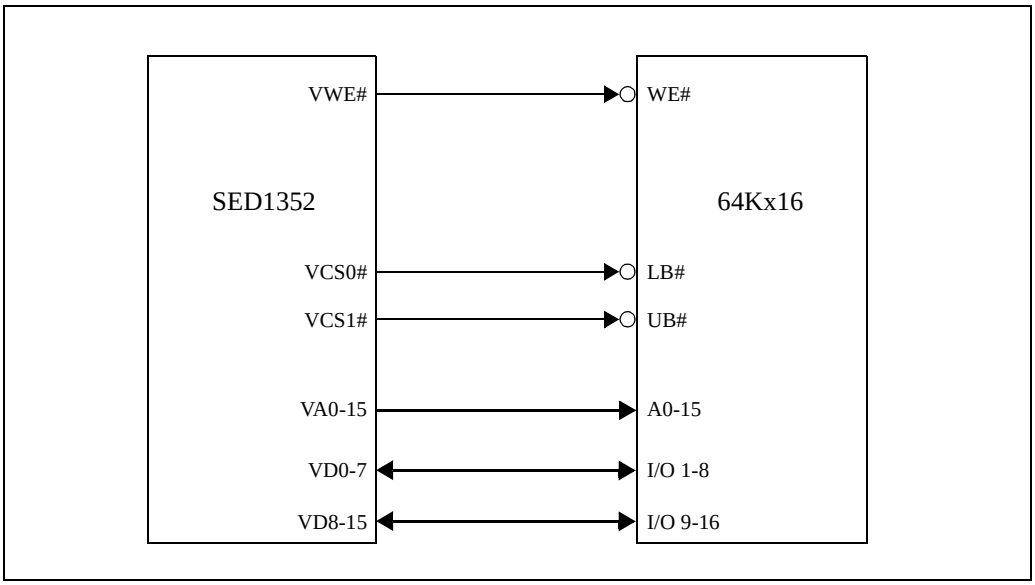


Figure 36: 16-Bit Mode - 128K bytes SRAM

9.2 SRAM Access Time

9.2.1 8-Bit Display Memory Interface:

Table 9-1: 8-Bit Display Memory Interface SRAM Access Time

Display Mode	3V/3.3V	5V
16-level gray shades	Access time $\leq$ 1 / f_{OSC} - 50ns	Access time $\leq$ 1 / f_{OSC} - 30ns
4-level gray shades	Access time $\leq$ 2 / f_{OSC} - 50ns	Access time $\leq$ 2 / f_{OSC} - 30ns

9.2.2 16-Bit Display Memory Interface:

Table 9-2: 16-Bit Display Memory Interface SRAM Access Time

Display Mode	3V/3.3V	5V
16-level gray shades	Access time $\leq$ 2 / f_{OSC} - 50ns	Access time $\leq$ 2 / f_{OSC} - 30ns
4-level gray shades	Access time $\leq$ 4 / f_{OSC} - 50ns	Access time $\leq$ 4 / f_{OSC} - 30ns

9.3 Frame Rate Calculation

9.3.1 For Single Panel

:

$$FrameRate = \frac{f_{osc}}{(HorizontalPixels + DHNDP) \times (VerticalLines + 4)}$$

9.3.2 For Dual Panel

:

$$FrameRate = \frac{f_{osc}}{(HorizontalPixels + DHNDP) \times 2 \times \left(\frac{VerticalLines}{2} + 2\right)}$$

Where DHNDP is Default Horizontal Non-Display Period in term of pixels:
DHNDP = 16 pixels per panel.

9.4 Memory Size Calculation

Memory Size (bytes) = $\frac{(HorizontalPixels) \times (VerticalLines) \times (BitsPerPixel)}{8}$

Example: For a 640x480, 4 gray shades (2 bits-per-pixel) system:

Memory Size (bytes) = $\frac{(640) \times (480) \times (2)}{8} = 76800bytes = 75Kbyte$

9.5 Memory Size Requirement

The following tables summarize the preceding information (formulae).

Input clock (f_{OSC}) is limited by SRAM access time depending on the display mode and display memory interface that is being used. As a result, different resolutions will have different input clock and memory requirements for a particular frame rate. Tables 9-3 through 9-5 summarize the minimum memory size and access time requirements for various resolutions at a particular input clock along with the corresponding frame rates.

Table 9-3: Memory Size Requirement: Number of Horizontal Pixels = 640

		Number of Horizontal Pixels = 640								
			4 Grays (2 bits-per-pixel)			**16 Grays (4 bits-per-pixel)**			**Example**	
		Display Memory Interface	**Size (KB)**	**Access Time**		**Size (KB)**	**Access Time**		**Input Clock (f_{OSC})**	**Frame Rate**
				3V/3.3V	**5V**		**3V/3.3V**	**5V**		
Number of Vertical Lines	**480**	8-bit 16-bit	75	(2) 115 ns	(2) 135 ns	150	(1)	(1)	24 MHz	76 Hz
	400	8-bit 16-bit	62.5	50 ns 150 ns	70 ns 170 ns	125	(2) 50 ns	(2) 70 ns	20 MHz	75 Hz
	320	8-bit 16-bit	50	75 ns 200 ns	95 ns 220 ns	100	(2) 75 ns	(2) 95 ns	16 MHz	75 Hz
	256	8-bit 16-bit	40	115 ns 280 ns	135 ns 300 ns	80	(2) 115 ns	(2) 135 ns	12 MHz	70 Hz
	240	8-bit 16-bit	37.5	115 ns 280 ns	135 ns 300 ns	75	(2) 115 ns	(2) 135 ns	12 MHz	75 Hz
	200	8-bit 16-bit	32	150 ns 350 ns	170 ns 370 ns	62.5	50 ns 150 ns	70 ns 170 ns	10 MHz	75 Hz

(1) Memory more than 128KB cannot be supported by SED1352.

(2) Memory more than 64KB can only be supported through 16-bit display memory interface.

* KB = K byte = 1024 bytes

Table 9-4: Memory Size Requirement: Number of Horizontal Pixels = 480

			Number of Horizontal Pixels = 480							
			4 Grays (2 bits-per-pixel)			16 Grays (4 bits-per-pixel)			Example	
		Display Memory Interface	Size (KB)	Access Time		Size (KB)	Access Time		Input Clock (f_{OSC})	Frame Rate
				3V/3.3V	5V		3V/3.3V	5V		
Number of Vertical Lines	**480**	8-bit 16-bit	57	60 ns 170 ns	80 ns 190 ns	113	(2) 60 ns	(2) 80 ns	18 MHz	75 Hz
	400	8-bit 16-bit	47	90 ns 230 ns	110 ns 250 ns	94	(2) 90 ns	(2) 110 ns	14 MHz	70 Hz
	320	8-bit 16-bit	37.5	115 ns 280 ns	135 ns 300 ns	75	(2) 115 ns	(2) 135 ns	12 MHz	75 Hz
	256	8-bit 16-bit	30	150 ns 350 ns	170 ns 370 ns	60	50 ns 150 ns	70 ns 170 ns	10 MHz	77 Hz
	240	8-bit 16-bit	29	200 ns 450 ns	220 ns 470 ns	57	75 ns 200 ns	95 ns 220 ns	8 MHz	66 Hz
	200	8-bit 16-bit	23.5	200 ns 450 ns	220 ns 470 ns	47	75 ns 200 ns	95 ns 220 ns	8 MHz	73 Hz

Table 9-5: Memory Size Requirement: Number of Horizontal Pixels = 320

			Number of Horizontal Pixels = 320							
			4 Grays (2 bits-per-pixel)			16 Grays (4 bits-per-pixel)			Example	
		Display Memory Interface	Size (KB)	Access Time		Size (KB)	Access Time		Input Clock (f_{OSC})	Frame Rate
				3V/3.3V	5V		3V/3.3V	5V		
Number of Vertical Lines	**480**	8-bit 16-bit	37.5	115 ns 280 ns	135 ns 300 ns	75	(2) 115 ns	(2) 135 ns	12 MHz	74 Hz
	400	8-bit 16-bit	32	150 ns 350 ns	170 ns 370 ns	62.5	50 ns 150 ns	70 ns 170 ns	10 MHz	74 Hz
	320	8-bit 16-bit	25	200 ns 450 ns	220 ns 470 ns	50	75 ns 200 ns	95 ns 220 ns	8 MHz	73 Hz
	256	8-bit 16-bit	20	280 ns 615 ns	300 ns 630 ns	40	115 ns 280 ns	135 ns 300 ns	6 MHz	69 Hz
	240	8-bit 16-bit	19	280 ns 615 ns	300 ns 635 ns	37.5	115 ns 280 ns	135 ns 300 ns	6 MHz	73 Hz
	200	8-bit 16-bit	16	350 ns 750 ns	370 ns 770 ns	32	150 ns 350 ns	170 ns 370 ns	5 MHz	73 Hz

(1) Memory more than 128KB cannot be supported by SED1352.

(2) Memory more than 64KB can only be supported through 16-bit display memory interface.

* KB = K byte = 1024 bytes

10 MECHANICAL DATA

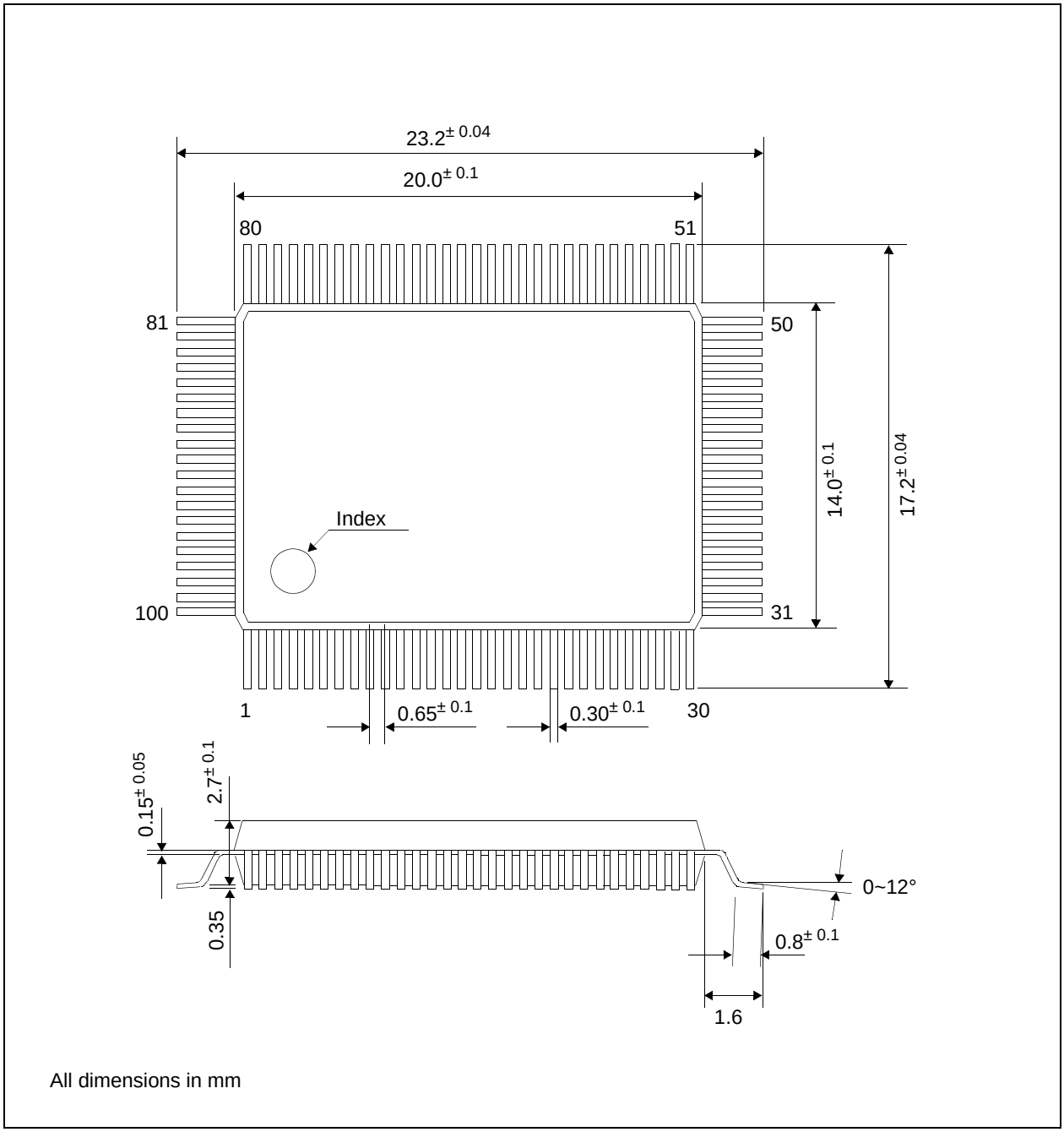


Figure 37: Mechanical Drawing QFP5-100pin-S2

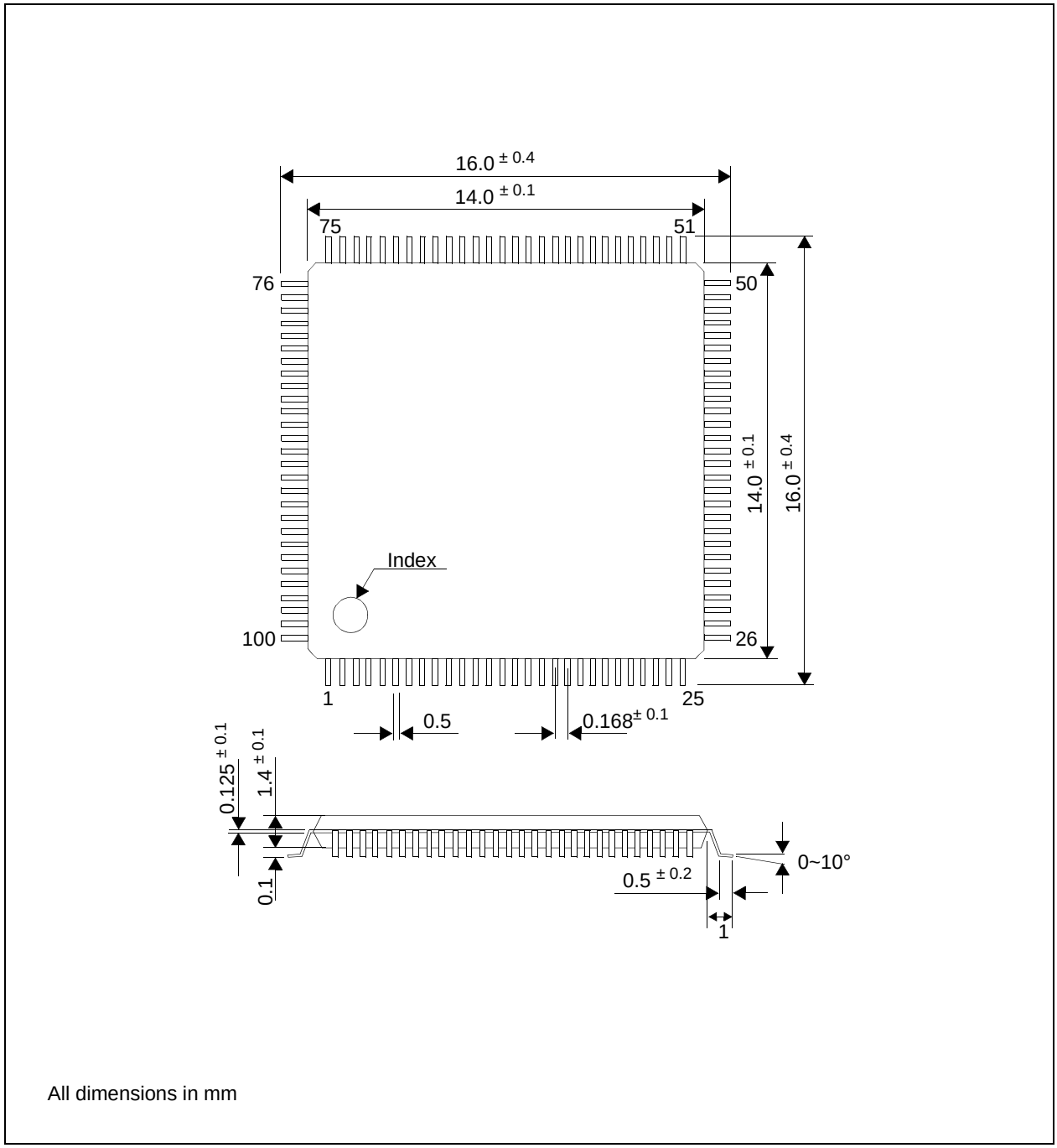


Figure 38: Mechanical Drawing QFP15-100pin

THIS PAGE LEFT BLANK

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

Programming Notes and Examples

Document Number: X16-BG-007-04

Copyright © 1996, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

Table of Contents

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 INTRODUCTION

The purpose of this guide is to demonstrate how to program the SED1352 LCD controller, with reference made to the SDU1352B0x evaluation board. The first half of this guide presents the basic concepts of LCD controllers, which describe the following:

- Initializing the SED1352
- Gray Shades and Look-Up Tables
- Display Memory Models
- Virtual Displays
- Bitmaps and Text Displays
- Registers
- Split Screen
- Panning and Scrolling
- Power Saving

The second half of this guide presents programming examples for the following:

- Initialization
- Read Registers
- Gray Shades and Look-Up Tables
- Text
- Split Screen
- Panning and Scrolling
- Power Saving

These programming examples are combined in a simple menu-driven program. Most of the program is written in the ‘C’ programming language, with some parts written in 8086 assembly.

2 INITIALIZING THE SED1352

This section presents two examples to show how to initialize the SED1352 registers and write a pixel to the display. Code to initialize the SED1352 is provided in Section 6.2, *"Initialization Code"* on page 48.

The following examples describe values written to registers.

- A "normal" value is one which must not change after initialization of all registers.
- A "panel specific" value is one required for the given type of panel. Such a value must never change after initialization of all registers.
- An "implementation specific" value is one required for the hardware implementation of the SED1352. Such a value must never change after initialization of all registers. Refer to the *SED1352F0x Hardware Functional Specification* and *SDU1352B0x Evaluation Board User's Manual* for more information on hardware implementation issues.
- An "application specific" value is one that can be changed by the program after initialization of all registers.

Example 1: Initialize the registers for a 16 gray shade 320x240 single panel LCD with 64k of display memory. Afterwards write one pixel to the top left corner of the display.

Program SED1352 Registers 00h-0Dh:

AUX Register	Data (in Binary)	Notes	See Also
AUX[00h]	0000 0000	• must be zero	
AUX[01h]	1000 1000	• b7 = display on (normal) • b6 = single panel (panel specific) • b5 = XSCL not masked (panel specific) • b4 = LCDE LCDENB pin = 0 (implementation specific; the recommended procedure is to turn this bit off during register initialization and afterwards turn this bit on) • b3 = 16 grays (application specific) • b2 = 4 bit LCD data width (panel specific) • b1 = 16 bit Memory Interface (implementation specific) • b0 = RAMS ignored (implementation specific)	
AUX[02h]	0100 1111	• bits 7-0 = bits 7-0 of Line Byte Count (panel specific) • bit 8 of Line Byte Count in bit 0 of AUX[03h] (panel specific)	see Note A at end of Table for calculation
AUX[03h]	0000 0110	• bits 7-6 = Power Save Mode 0 (application specific) • bit 5 = LCD interface signals forced to 0 during Power Save (implementation specific) • bit 4 = no LUT bypass (application specific) • bits 3-1 = not used • bit 0 = bit 8 of Line Byte Count (panel specific, see AUX[02h])	see Section 5.6, *"Power Saving"* on page 44
AUX[04h]	1110 1111	• bits 7-0 = bits 7-0 of Total Display Line Count (panel specific) • bits 9-8 of Total Display Line Count in bits 1-0 of AUX[05h] (panel specific)	see Note B and C at end of Table for calculation
AUX[05h]	0000 0000	• bits 7-2 = WF not required (panel specific) • bits 1-0 = bits 9-8 of Total Display Line Count (panel specific, see AUX[04h])	

AUX Register	Data (in Binary)	Notes	See Also
AUX[06h] AUX[07h]	0000 0000 1000 0000	• bits 7-0 = bits 7-0 of Screen 1 Display Start Address (application specific) • bits 15-8 of Screen 1 Display Start Address in AUX[07h] (application specific) • Screen 1 Display Start Address points to D000:0000h • when 0000h, Screen 1 Display Start Address is located at D000:0000h, bank 0, on the SDU1353B0C • bits 7-0 = bits 15-8 of Screen 1 Display Start Address (application specific, see AUX[06h])	see Section 4.2.1, *"SDU1352B0x Evaluation Board Display Memory"* on page 24 and Section 4.1, *"Registers"* on page 22
AUX[08h] AUX[09h]	0000 0000 1000 0000	• bits 7-0 = bits 7-0 of Screen 2 Display Start Address (application specific) • bits 15-8 of Screen 2 Display Start Address in AUX[09h] (application specific) • Screen 2 Display Start Address points to D000:0000h • bits 7-0 = bits 15-8 of Screen 2 Display Start Address (application specific, see AUX[08h])	see Section 4.2.1, *"SDU1352B0x Evaluation Board Display Memory"* on page 24 and Section 4.1, *"Registers"* on page 22
AUX[0Ah]	1110 1111	• bits 7-0 = bits 7-0 of Screen 1 Display Line Count (application specific) • bits 9-8 of Screen 1 Display Line Count in bits 1-0 of AUX[0Bh] (application specific) • Screen 1 Display Line Count is typically the same as Total Display Line Count • (AUX[0Ah] = AUX[04h], bits 1-0 of AUX[0Bh] = bits 1-0 of AUX[05h])	see Section 5.4, *"Split Screen"* on page 34
AUX[0Bh]	1111 1100	• bits 7-2 = not used • bits 1-0 = bits 9-8 of Screen 1 Display Line Count (application specific, see AUX[0Ah])	
AUX[0Dh]	0000 0000	• bits 7-0 = no address pitch adjustment	see Section 5.1, *"Virtual Displays"* on page 28
AUX[0Eh]	0000 0000	select palette address • bits 7-6 = bank 0 (application specific) • bits 5-4 = ID bits (read only; application specific) • bits 3-0 = palette address (application specific)	
AUX[0Fh]	0000 0000	write monochrome LUT data • bits 7-4 = N/A • bits 3-0 = palette data (application specific)	
AUX[0Eh]	0000 0001	increment palette address	
AUX[0Fh]	0000 0001	write monochrome LUT data	
AUX[0Eh]	0000 0010	increment palette address	
AUX[0Fh]	0000 0010	write monochrome LUT data	
AUX[0Eh]	0000 0011	increment palette address	
AUX[0Fh]	0000 0011	write monochrome LUT data	
AUX[0Eh]	0000 0100	increment palette address	

AUX Register	Data (in Binary)	Notes	See Also
AUX[0Fh]	0000 0100	write monochrome LUT data	
AUX[0Eh]	0000 0101	increment palette address	
AUX[0Fh]	0000 0101	write monochrome LUT data	
AUX[0Eh]	0000 0110	increment palette address	
AUX[0Fh]	0000 0110	write monochrome LUT data	
AUX[0Eh]	0000 0111	increment palette address	
AUX[0Fh]	0000 0111	write monochrome LUT data	
AUX[0Eh]	0000 1000	increment palette address	
AUX[0Fh]	0000 1000	write monochrome LUT data	
AUX[0Eh]	0000 1001	increment palette address	
AUX[0Fh]	0000 1001	write monochrome LUT data	
AUX[0Eh]	0000 1010	increment palette address	
AUX[0Fh]	0000 1010	write monochrome LUT data	
AUX[0Eh]	0000 1100	increment palette address	
AUX[0Fh]	0000 1100	write monochrome LUT data	
AUX[0Eh]	0000 1101	increment palette address	
AUX[0Fh]	0000 1101	write monochrome LUT data	
AUX[0Eh]	0000 1110	increment palette address	
AUX[0Fh]	0000 1110	write monochrome LUT data	
AUX[0Eh]	0000 1111	increment palette address	
AUX[0Fh]	0000 1111	write monochrome LUT data	
AUX[01h]	1001 1000	Program Mode Register bit DISP to 1, and set LCDE to enable power supply. 1001 0000b 'OR' {original value for AUX[01h]} • b7 = display on (application specific) • b4 = LCDE = LCDENB pin = set to enable specific power supply design (for SDU1353B0C, set bit to 1 to enable power supply) (application specific)	

Write one pixel to the top left corner of display memory.
If the SDU1352B0x evaluation board is used, video memory begins at D000:0000h; in this case write 0F0h to location D000:0000h.

Note
A

$$\text{Line Byte Count} = \left(\frac{\text{Panel Width in Pixels}}{\text{Memory Interface Width (8 or 16 bits)}} \times [\text{bits per pixel (2 or 4 bits)}]\right) - 1$$

$$= \left(\frac{320}{16} \times 4\right) - 1 = 79 = 4\text{Fh}$$

B Single Panel

$$\text{Total Display Line Count} = \text{number of display lines} - 1 = 240 - 1 = 239 = 0\text{EFh}$$

C Dual Panel

$$\text{Total Display Line Count} = \frac{\text{number of display lines}}{2} - 1$$

Example 2: Initialize the registers for a 4 gray shade 640x480 dual panel LCD with 128k of display memory. Afterwards write one pixel to the top left corner of the display.

Program SED1352 Registers 00h-0Dh:

AUX Register	Data (in Binary)	Notes	See Also
AUX[00h]	0000 0000	• must be zero	
AUX[01h]	1100 1000	• b7 = display on (normal) • b6 = dual panel (panel specific) • b5 = XSCL not masked (panel specific) • b4 = LCDE LCDENB pin = 0 (implementation specific; the recommended procedure is to turn this bit off during register initialization and afterwards turn this bit on) • b3 = 16 grays (application specific) • b2 = 4 bit LCD data width (panel specific) • b1 = 16 bit Memory Interface (implementation specific) • b0 = RAMS ignored (implementation specific)	
AUX[02h]	0100 1111	• bits 7-0 = bits 7-0 of Line Byte Count (panel specific) • bit 8 of Line Byte Count in bit 0 of AUX[03h] (panel specific)	see Note A at end of Table for calculation
AUX[03h]	0000 0110	• bits 7-6 = Power Save Mode 0 (application specific) • bit 5 = LCD interface signals forced to 0 during Power Save (implementation specific) • bit 4 = no LUT bypass (application specific) • bits 3-1 = not used • bit 0 = bit 8 of Line Byte Count (panel specific, see AUX[02h])	see Section 5.6, *"Power Saving"* on page 44
AUX[04h]	1110 1111	• bits 7-0 = bits 7-0 of Total Display Line Count (panel specific) • bits 9-8 of Total Display Line Count in bits 1-0 of AUX[05h] (panel specific)	see Note B and C at end of Table for calculation
AUX[05h]	0000 0000	• bits 7-2 = WF not required (panel specific) • bits 1-0 = bits 9-8 of Total Display Line Count (panel specific, see AUX[04h])	

AUX Register	Data (in Binary)	Notes	See Also
AUX[06h] AUX[07h]	0000 0000 0000 0000	• bits 7-0 = bits 7-0 of Screen 1 Display Start Address (application specific) • bits 15-8 of Screen 1 Display Start Address in AUX[07h] (application specific) • Screen 1 Display Start Address points to C000:0000h • when 0000h, Screen 1 Display Start Address is located at D000:0000h, bank 0, on the SDU1353B0C • bits 7-0 = bits 15-8 of Screen 1 Display Start Address (application specific, see AUX[06h])	see Section 4.2.1, *"SDU1352B0x Evaluation Board Display Memory"* on page 24 and Section 4.1, *"Registers"* on page 22
AUX[08h] AUX[09h]	0000 0000 0100 1011	• bits 7-0 = bits 7-0 of Screen 2 Display Start Address (application specific) • bits 15-8 of Screen 2 Display Start Address in AUX[09h] (application specific) • Screen 2 Display Start Address points to C000:9600h • bits 7-0 = bits 15-8 of Screen 2 Display Start Address (application specific, see AUX[08h])	see Section 4.2.1, *"SDU1352B0x Evaluation Board Display Memory"* on page 24 and Section 4.1, *"Registers"* on page 22
AUX[0Ah]	1110 1111	• bits 7-0 = bits 7-0 of Screen 1 Display Line Count (application specific) • bits 9-8 of Screen 1 Display Line Count in bits 1-0 of AUX[0Bh] (application specific) • Screen 1 Display Line Count is typically the same as Total Display Line Count • (AUX[0Ah] = AUX[04h], bits 1-0 of AUX[0Bh] = bits 1-0 of AUX[05h])	see Section 5.4, *"Split Screen"* on page 34
AUX[0Bh]	1111 1100	• bits 7-2 = not used • bits 1-0 = bits 9-8 of Screen 1 Display Line Count (application specific, see AUX[0Ah])	
AUX[0Dh]	0000 0000	• bits 7-0 = no address pitch adjustment	see Section 5.1, *"Virtual Displays"* on page 28
AUX[0Eh]	0000 0000	select palette address • bits 7-6 = bank 0 (application specific) • bits 5-4 = ID bits (read only; application specific) • bits 3-0 = palette address (application specific)	
AUX[0Fh]	0000 0000	write monochrome LUT data • bits 7-4 = N/A • bits 5-4 = bank 0 (application specific) • bits 3-0 = palette data (application specific)	
AUX[0Eh]	0000 0001	increment palette address	
AUX[0Fh]	0000 0101	write monochrome LUT data	
AUX[0Eh]	0000 0010	increment palette address	
AUX[0Fh]	0000 1010	write monochrome LUT data	
AUX[0Eh]	0000 0011	increment palette address	
AUX[0Fh]	0000 1111	write monochrome LUT data	

AUX Register	Data (in Binary)	Notes	See Also
AUX[0Eh]	0000 0100	increment palette address	
AUX[0Fh]	0000 0000	write monochrome LUT data	
AUX[0Eh]	0000 0101	increment palette address	
AUX[0Fh]	0000 0101	write monochrome LUT data	
AUX[0Eh]	0000 0110	increment palette address	
AUX[0Fh]	0000 1010	write monochrome LUT data	
AUX[0Eh]	0000 0111	increment palette address	
AUX[0Fh]	0000 1111	write monochrome LUT data	
AUX[0Eh]	0000 1000	increment palette address	
AUX[0Fh]	0000 0000	write monochrome LUT data	
AUX[0Eh]	0000 1001	increment palette address	
AUX[0Fh]	0000 0101	write monochrome LUT data	
AUX[0Eh]	0000 1010	increment palette address	
AUX[0Fh]	0000 1010	write monochrome LUT data	
AUX[0Eh]	0000 1011	increment palette address	
AUX[0Eh]	0000 1111	increment palette address	
AUX[0Fh]	0000 1100	write monochrome LUT data	
AUX[0Fh]	0000 0000	write monochrome LUT data	
AUX[0Eh]	0000 1101	increment palette address	
AUX[0Fh]	0000 0101	write monochrome LUT data	
AUX[0Eh]	0000 1110	increment palette address	
AUX[0Fh]	0000 1010	write monochrome LUT data	
AUX[0Eh]	0000 1111	select palette address	
AUX[0Fh]	0000 1111	write monochrome LUT data	
AUX[01h]	1001 1000	Program Mode Register bit DISP to 1, and set LCDE to enable power supply. 1001 0000b 'OR' {original value for AUX[01h]} • b7 = display on (application specific) • b4 = LCDE = LCDENB pin = set to enable specific power supply design (for SDU1353B0C, set bit to 1 to enable power supply) (application specific)	

Write one pixel to the top left corner of display memory.
If the SDU1352B0x evaluation board is used, the first panel's memory addresses begin at C000:0000h (see Section 5.4.4.1, *"Displaying a Single Image on a Dual Panel"* on page 40). Consequently write 0C0h to location C000:0000h for the SDU1352B0x.

Note
A

$$\text{Line Byte Count} = \left(\frac{\text{Panel Width in Pixels}}{\text{Memory Interface Width (8 or 16 bits)}} \times [\text{bits per pixel (2 or 4 bits)}]\right) - 1$$

$$= \left(\frac{640}{16} \times 2\right) - 1 = 79 = 4\text{Fh}$$

B Single Panel

$$\text{Total Display Line Count} = \text{number of display lines} - 1$$

C Dual Panel

$$\text{Total Display Line Count} = \frac{\text{number of display lines}}{2} - 1 = \frac{480}{2} - 1 = 239 = 0\text{EFh}$$

3 GRAY SHADES AND LOOK-UP TABLES

This section discusses memory formats and Look-Up Table formats for the SED1352.

3.1 Pixels

A pixel is physically stored in display memory as a series of bits. The more bits, the more gray shades the pixel can show. With only one bit, the pixel can only show two different combinations of gray shades (0 or 1). With two bits, the pixel can show four different combinations of gray shades (00b, 01b, 10b, or 11b). Similarly, four bits allow 16 different combinations of gray shades (0000b, 0001b, 0010b, ... 1111b).

The SED1352 can be programmed to use either two bit or four bit pixels. The following sections show how these pixels are stored in display memory.

3.1.1 Two Bit Pixels

To store two bit pixels, four pixels are grouped into one byte of display memory as shown below:

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 1: Pixel Storage for 2 Bits (4 Gray Shades) In One Byte of Display Memory

When these pixels are shown, Pixel 0 is seen to be left of Pixel 1, Pixel 1 is seen to be left of Pixel 2, and so on.

3.1.2 Four Bit Pixels

To store four bit pixels, two pixels are grouped into one byte of display memory as shown below:

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 2: Pixel Storage for 4 Bits (16 gray shades) in One Byte of Display Memory

When these pixels are shown, Pixel 0 is seen to be left of Pixel 1.

3.2 Look-Up Table (LUT)

This section provides a concise description of the LUT registers, followed by a description of a LUT. Next is a series of examples which show how to initialize a LUT, create an inverted LUT, and how to select one of four banks in the 4 gray shade mode.

3.2.1 LUT Registers

Note
Register bits discussed in this section are highlighted.

AUX[0E] Look-Up Table Address Register I/O address = 1110b, Read/Write							
Bank Bit 1	Bank Bit 0	ID Bit (Read Only)	ID Bit (Read Only)	Palette Address Bit 3	Palette Address Bit 2	Palette Address Bit 1	Palette Address Bit 0

The SED1352 has one internal 16 position, 4-bit wide Look-Up Table (palette). The 4-bit value programmed into each table position determines the output gray shade of display data.

For example, in 16-level gray shade mode, a data value of 0001h (4 bits per pixel) will point to Look-Up Table positions one and display the 4-bit gray shade that was previously programmed into that location.

bits 7-6 Bank Bits [1:0]
In 4-level gray mode (2-bits/pixel), the 16 position palette is arranged into four, 4 position “banks”. These two bits control which bank is currently selected. These bits have no effect in 16-level gray mode (4-bits/pixel).

bits 3-0 Palette Address Bits [3:0]
These 4 bits provide a pointer into the 16 position Look-Up Table currently selected for CPU R/W access.

Note
The Look-Up Table configuration (e.g. 1/2/4 banks) does not affect the R/W access from the CPU as all 16 positions can be accessed sequentially.

AUX[0F] Look-Up Table Data Register I/O address = 1111b, Read/Write.							
n/a	n/a	n/a	n/a	Palette Data Bit 3	Palette Data Bit 2	Palette Data Bit 1	Palette Data Bit 0

bits 3-0 Palette Data Bits [3:0]
These 4-bits are the gray shade values used for display data output. They are programmed into the 4-bit Look-Up Table (palettes) positions pointed to by Palette Address bits [3:0].

For example; in a 16-level gray shade display mode, a data value of 0001b (4-bits / pixel) will point to Look-Up Table position one and display the 4-bit gray shade corresponding to the value programmed into that location.

3.2.2 Look-Up Table Description

The Look-Up Table (LUT, or palette) treats the value of a pixel as an index of an array of 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.

Because LUT entries represent the actual gray shades shown on the LCD panel, pixel values indirectly select which gray shade displays.

The SED1352 supports two different data formats; 4 bits-per-pixel (16 gray shades) and 2 bits-per-pixel (4 gray shades). In 4 bits-per-pixel mode the SED1352 provides a 16 position, 4 bit wide LUT. In 2 bits-per-pixel mode, the SED1352 provides 4 “banks” of 4 position, 4 bit wide LUTs.

The value inside each LUT entry represents the gray shade. This value ranges between 0 and 15.

The SED1352FOB Look-Up Table is linear; increasing the LUT entry number results in a lighter gray shade. For example, a LUT entry of 0Fh into a look-up entry will always result in a bright white output. An entry of 00h into a look-up entry will always result in a black output.

Example 3: Initialize the Look-Up Table

The following describes how to initialize the Look-Up Table for 16 gray shades. Table 3-1 shows a LUT with gray shades starting from black (index 0) and finishing in white (index 15, or 0Fh).

1. Write LUT index to Look-Up Table Address Register AUX[0Eh].
2. Write LUT entry value to Look-Up Table Data Register AUX[0Fh].
3. Repeat steps 1 and 2 until all 16 LUT entries have been written.

Table 3-1: SED1352F0B Black-To-White Look-Up Table for 16 Gray Shades

Index (hex)	**Look-Up Table** (hex)	**Index** (hex)	**Look-Up Table** (hex)
0	0	8	8
1	1	9	9
2	2	A	A
3	3	B	B
4	4	C	C
5	5	D	D
6	6	E	E
7	7	F	F

Example 4: Initialize an Inverted Look-Up Table

This example shows how to invert an image by changing only the LUT. Inverting means that pixels formally shown as light gray shades are now shown as dark gray shades, and vise versa. It does not matter whether the SED1352 is in 4 gray shade or 16 gray shade mode.

1. Read LUT entry:
   Write LUT index to Look-Up Table Address Register AUX[0Eh]
   Read "Old LUT Entry" from Look-Up Table Data Register AUX[0Fh]
2. Calculate "New LUT Entry" according to the following formula:

   New LUT Entry = 15 – Old LUT Entry

3. Write LUT entry back:
   Write LUT index to Look-Up Table Address Register AUX[0Eh]
   Write "New LUT Entry" to Look-Up Table Data Register AUX[0Fh]
4. Repeat steps 1 to 3 until all 16 LUT entries have been changed.

If Table 3-1 was previously programmed into the SED1352, the new inverted LUT would be the following:

Table 3-2: SED1352F0B Inverted Look-Up Table (White-To-Black)

Index (hex)	**Look-Up Table** (hex)	**Index** (hex)	**Look-Up Table** (hex)
0	F	8	7
1	E	9	6
2	D	A	5
3	C	B	4
4	B	C	3
5	A	D	2
6	9	E	1
7	8	F	0

3.2.3 Four Gray Shades (Two Bits/Pixel)

When the SED1352 is configured for two bit pixels, each pixel can index one of four LUT entries. In this 4 gray shade mode, the SED1352 treats the 16 entries in the LUT as four separate look-up tables or *banks*, each having four entries (see Figure 3). The following examples show how to program and select these banks.

Example 5: In 4 gray shade mode, program bank 2 LUT entries and select for use.

1. Determine location of bank 2 in LUT.
   The first four entries in the 16 entry LUT represent the first bank (bank 0). The following four entries in the LUT represent the second bank (bank 1), etc. Consequently bank 2 starts at LUT index 8 as shown below:

   $$\text{start of bank index} = \text{bank number} \times 4$$
   $$\text{start of bank 2} = 2 \times 4 = 8$$

   Bank 2 is shown in Figure 3, palette 2.
2. Write LUT index to Look-Up Table Address Register AUX[0Eh].
   For bank 2, the index will one of the following values: 08h, 09h, 0Ah, or 0Bh
3. Write LUT entry value to Look-Up Table Data Register AUX[0Fh].
   For a linear LUT, use the look-up table entries in Table 3-1, “SED1352F0B Black-To-White Look-Up Table for 16 Gray Shades,” on page 17.
4. Repeat steps 2 and 3 until all 4 LUT entries have been written.
5. To display data using Bank 2 write 10b to AUX[0E] bits 7,6.

Table 3-3: SED1352F0B Black-To-White Look-Up Table for 4 Gray Shades

Index (hex)	**Look-Up Table** (hex)
0	0
1	5
2	A
3	F

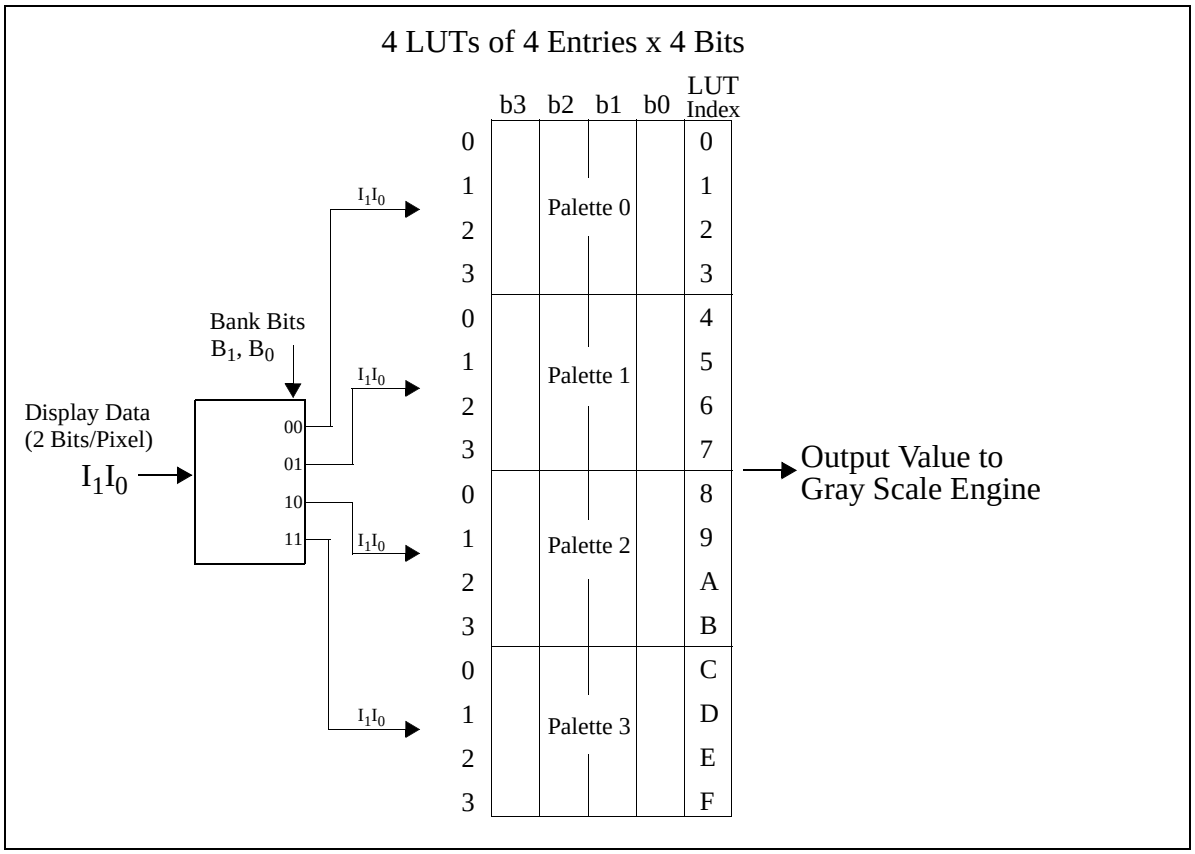


Figure 3: 4-Level Gray Shade Mode Look-Up Table Architecture

3.2.4 Sixteen Gray Shades (Four Bits/Pixel)

When the SED1352 has 4 bit pixels, each pixel can index into one of 16 LUT entries. The LUT bank bits are ignored in this mode.

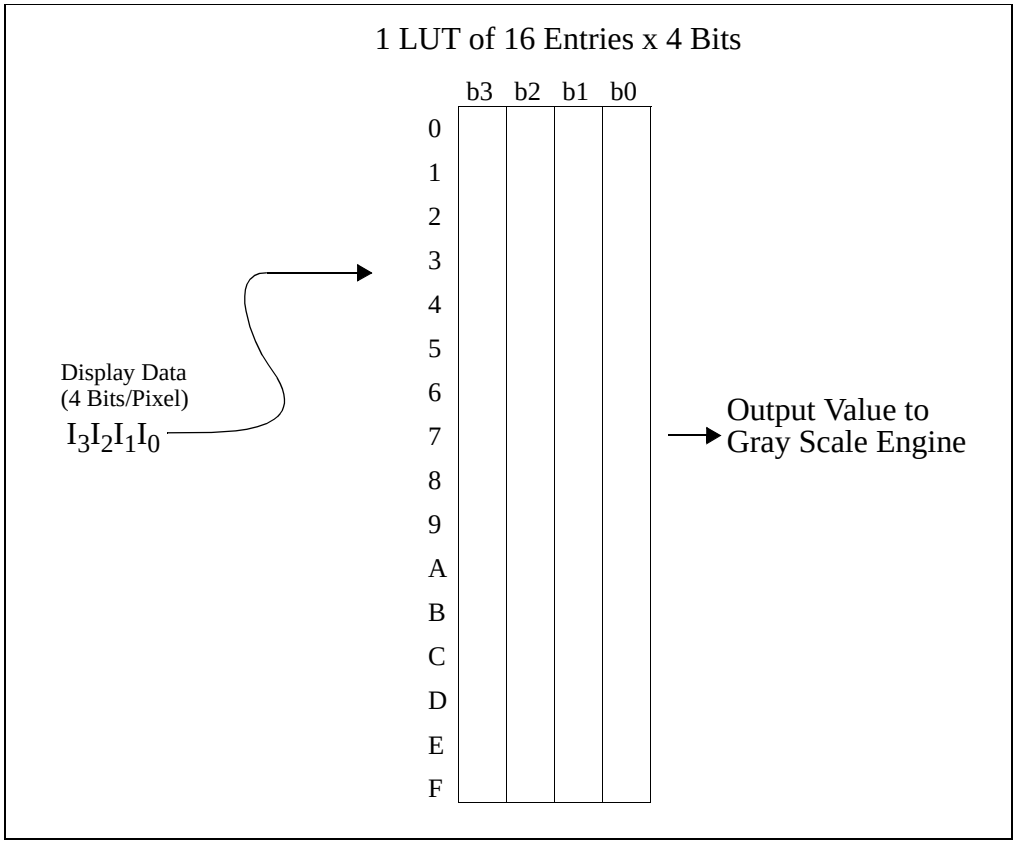


Figure 4: 16-Level Gray Shade Mode Look-Up Table Architecture

4 DISPLAY MEMORY MODELS

This section introduces display memory models. A concise description of the Display Start Address Registers is provided, followed by a description of display memory. Afterwards examples are provided, illustrating how to calculate the display memory model for a given display resolution and gray level mode. Once this model is calculated, examples on programming the Display Start Address Registers are provided.

4.1 Registers

Register bits discussed in this section are highlighted.

AUX[01] Mode Register I/O address = 0001b, Read/Write.							
DISP	Panel	Mask XSCL	LCDE	Gray Scale	LCD Data Width	Memory Interface	RAMS

bit 1 Memory Interface
This bit selects between the 8-bit or 16-bit memory interface. When this bit = 0, the 16-bit memory interface is selected. When this bit = 1, the 8-bit memory interface is selected. If 16-bit bus interface is selected (VD0 = 1 on RESET), the Memory Interface bit is forced to 0 internally (16-bit). This bit goes low on RESET.

AUX[06] Screen 1 Display Start Address Register (LSB) I/O address = 0110b, Read/Write.							
Screen 1 Display Start Addr Bit 7	Screen 1 Display Start Addr Bit 6	Screen 1 Display Start Addr Bit 5	Screen 1 Display Start Addr Bit 4	Screen 1 Display Start Addr Bit 3	Screen 1 Display Start Addr Bit 2	Screen 1 Display Start Addr Bit 1	Screen 1 Display Start Addr Bit 0

AUX[07] Screen 1 Display Start Address Register (MSB) I/O address = 0111b, Read/Write.							
Screen 1 Display Start Addr Bit 15	Screen 1 Display Start Addr Bit 14	Screen 1 Display Start Addr Bit 13	Screen 1 Display Start Addr Bit 12	Screen 1 Display Start Addr Bit 11	Screen 1 Display Start Addr Bit 10	Screen 1 Display Start Addr Bit 9	Screen 1 Display Start Addr Bit 8

AUX[06] bits 7-0 Screen 1 Display Start Address Bits [15:0]
AUX[07] bits 7-0 These 16 bits determine the Screen 1 Display Start Address. In an 8-bit memory configuration these bits set the 16-bit start address (i.e., byte access). In a 16-bit memory configuration these are the 16 most significant bits of a 17-bit start address (i.e., word access).

The Screen 1 Display Start Address is the memory address corresponding to the first displayed pixel (top left corner). In a dual panel configuration, screen 1 refers to the upper half of the display. While in a single panel configuration, screen 1 refers to the first screen of the Split Screen Display feature where two different images (screen 1 and screen 2) can be displayed at the same time on one display.

Note
The absolute address into display memory is determined by the Memory Mapping Address which is set by VD13 - VD15.

AUX[08] Screen 2 Display Start Address Register (LSB) I/O address = 1000b, Read/Write.							
Screen 2 Display Start Addr Bit 7	Screen 2 Display Start Addr Bit 6	Screen 2 Display Start Addr Bit 5	Screen 2 Display Start Addr Bit 4	Screen 2 Display Start Addr Bit 3	Screen 2 Display Start Addr Bit 2	Screen 2 Display Start Addr Bit 1	Screen 2 Display Start Addr Bit 0

AUX[09] Screen 2 Display Start Address Register (MSB) I/O address = 1001b, Read/Write.							
Screen 2 Display Start Addr Bit 15	Screen 2 Display Start Addr Bit 14	Screen 2 Display Start Addr Bit 13	Screen 2 Display Start Addr Bit 12	Screen 2 Display Start Addr Bit 11	Screen 2 Display Start Addr Bit 10	Screen 2 Display Start Addr Bit 9	Screen 2 Display Start Addr Bit 8

AUX[08] bits 7-0 Screen 2 Display Start Address Bits [15:0]

AUX[09] bits 7-0 These 16 bits determine the Screen 2 Display Start Address. In an 8-bit memory configuration these bits set the 16-bit start address (i.e., byte access). In a 16-bit memory configuration these are the 16 most significant bits of a 17-bit start address (i.e., word access).

In a dual panel configuration, screen 2 refers to the lower half of the display. The Screen 2 Display Start Address is the memory address corresponding to first displayed pixel in the first line of the lower half of the display. If Screen 2 is started right after Screen 1, the Screen 2 Display Start Address is calculated with the following formula.

$$Screen2DisplayStartAddress(hex) =$$

$$\frac{(ImageHorizontalResolution) \times (ImageVerticalResolution) \times (BytesPerPixel)}{2 \times \left(\frac{MemoryInterfaceWidth}{8}\right)} + Screen1DisplayStartAddress$$

In a single panel configuration, screen 2 refers to the second screen of the Split Screen Display Feature where two different images (screen 1 and screen 2) can be displayed at the same time on one display. The Screen 2 Display Start Address is the memory address corresponding to the first pixel of the second image stored in display memory. To display screen 2 refer to AUX[0A] Screen 1 Display Line Count Register (LSB) below.

.

4.2 Description

When displaying an image, the SED1352 must read pixel data from display memory. This memory is organized to match the display resolution of the given LCD panel. To organize display memory, the following registers must be programmed:

1. Screen 1 Display Start Address Registers.
2. Screen 2 Display Start Address Registers.
3. Address Pitch Adjustment Register.

For the first example, the Address Pitch Adjustment Register is programmed to zero. This means that no virtual display is available; for information on virtual displays see Section 5.1, *"Virtual Displays"* on page 28.

4.2.1 SDU1352B0x Evaluation Board Display Memory

There are several issues to consider when programming the Screen Display Start Address Registers for the SDU1352B0x evaluation board:

- When the SDU1352B0x is set for 64k of display memory, display memory exists from address D000:0000h to address D000:FFFFh. When the SDU1352B0x is set for 128k of display memory, display memory exists from address C000:0000h to address D000:FFFFh.
- For the SDU1352B0x, the Screen Display Start Address Registers are always in reference to the display memory address C000:0000h. Writing 0 to a Display Start Address Register will always refer to C000:0000h, even if display memory only exists from D000:0000h to D000:FFFFh. Consequently if only 64k of display memory is present, 64k must be added to the display address in order to point to D000:0000h. This is a limitation of the evaluation board only.
- Although the SED1352 can set the Memory Interface to 8 or 16 bits, the SDU1352B0x evaluation board should be set up for 16 bits. As a result, the Display Start Address Registers are word pointers, not byte pointers. To illustrate how to use a word pointer, refer to Example 6. In general, any system which uses more than 64k of display memory must ***always*** have the Memory Interface set to 16 bits.

Example 6: For the SDU1352B0x, calculate the required start address register value which refers to location D000:0000h.

Since a value of 0 refers to location C000:0000h, the start address register must be programmed with an offset address of 1000:0000h = 10000h bytes, or 8000h words.
START ADDRESS[LSB] = 00h
START ADDRESS[MSB] = 80h

4.2.2 Display Start Address Registers

This section illustrates how to properly calculate the values for the Screen Start Address Registers for a given LCD panel resolution. However, this section is limited to single panel displays; refer to Section 5.4.4, *"Dual Panel LCD"* on page 38 to program the Screen Start Address Registers for a dual panel display.

In the following example, the Display Start Address Registers are programmed for a 16 gray shade 320 x 240 single monochrome display. The technique shown, however, can also be used to calculate the memory map of other resolutions. In addition, reference is made to the SDU1352B0x evaluation board; other hardware implementations of the SED1352 may assign different display and port addresses from those of the SDU1352B0x. Refer to the *SDU1352B0x Evaluation Board User's Manual* for more information on these hardware issues.

Example 7: Program the Display Start Address Registers for a 16 gray shade 320 x 240 single monochrome LCD panel; the display is attached to the SDU1352B0x evaluation board with 64k of display memory.

1. Calculate the number of bytes per scan line.

   16 gray shades => 4 bits per pixel

   4 bits per pixel => 2 pixels per byte

   $$\text{number of bytes per scan line} = \frac{\text{pixels per scan line}}{\text{pixels per byte}} = \frac{320}{2} = 160 \text{ bytes per scan line} = 00A0h \text{ bytes per scan line}$$

2. Calculate the total number of bytes required for display memory.

   $$(\text{bytes per scan line}) \times (\text{number of scan lines}) = 160 \times 240 = 38400 \text{ bytes} = 9600h \text{ bytes}$$

3. Create the memory map.
   Each scan line is 00A0h bytes long, there are 240 scan lines, and the last memory address is 9600h - 1.

Offset (hex)		Offset (hex)
0000	Scan Line 0	009F
00A0	Scan Line 1	013F
	⋮	
94C0	Scan Line 238	955F
9560	Scan Line 239	95FF

Figure 5: Memory Map for 320 x 240 LCD Panel with 16 Gray Shades

4. Program the Screen 1 Display Start Address Registers.
   Assume that the image starts at the beginning of display memory, which for 64k is D000:0000h. As shown in Example 6, the Screen 1 Display Start Address Registers must be programmed to 8000h words.
   AUX[06h] = 00h
   AUX[07h] = 80h

5. Program the Screen 2 Display Start Address Registers.
   Under normal programming conditions, the Screen 2 Display Start Address should be set to the same value as the Screen 1 Display Start Address. In the event that a split screen is required, refer to Section 5.4, *"Split Screen"* on page 34.
   AUX[08h] = 00h
   AUX[09h] = 80h

Example 8: Program the Display Start Address Registers for a dual panel LCD.

Refer to Section 5.4.4.1, *"Displaying a Single Image on a Dual Panel"* on page 40.

Example 9: Determine if the SED1352 implementation can support a 640x480 LCD with 4 gray shades.

1. Calculate the number of bytes per scan line:

$$\frac{\text{pixels per scan line}}{\text{pixels per byte}} = \frac{640}{4} = 160 \text{ bytes per scan line}$$

2. Calculate the total number of bytes required for display memory:

$$(160 \text{ bytes per scan line})(480 \text{ scan lines}) = 76800 \text{ bytes}$$

3. Compare the required number of bytes with the amount of memory available to the SED1352.
   - If the SED1352 has 128k available, there is 131,072 bytes available, which is greater than the 76,800 bytes required for 640 x 480 with 4 gray shades.
   - If the SED1352 has 64k available, there is 65,536 bytes available, which is less than the 76,800 bytes required for 640 x 480 with 4 gray shades.

4.3 Common Display Memory Requirements for LCD Panel Sizes

The following is a list of memory requirements and memory maps for common LCD resolutions. Note that the memory required for 640x480 with 16 gray shades exceeds 128k and is therefore not supported on the SED1352.

Table 4-1: Memory Size Requirements

Display Resolution	Pixel Storage		Memory Requirements	
	Bits/Pixel	**Gray Shades**	**Bytes**	**Hex**
320x240	2	4	19,200	0000 4B00
	4	16	38,400	0000 9600
640x200	2	4	32,000	0000 7D00
	4	16	64,000	0000 FA00
640x480	2	4	76,800	0001 2C00
	4	16	N/A	N/A

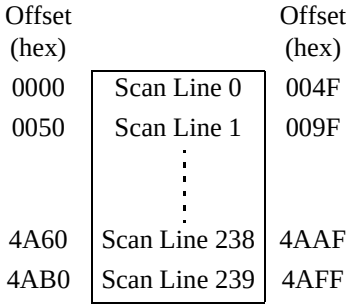


Figure 6: Memory Map Example for 320 x 240 LCD Panel with 4 Gray Shades

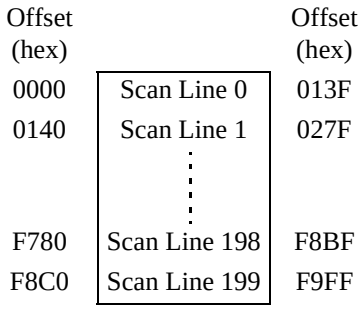


Figure 7: Memory Map Example for 640 x 200 LCD Panel with 16 Gray Shades

5 ADVANCED TECHNIQUES

This section presents information on the following:

- virtual displays
- bitmaps and text displays
- reading and writing to the SED1352 registers
- split screen displays
- panning and scrolling.
- power saving.

5.1 Virtual Displays

This section presents a detailed description of the Address Pitch Adjustment Register, followed by a description of a virtual display. Afterwards an example is given, showing how to create a virtual display.

5.1.1 Registers

Note
Register bits discussed in this section are highlighted.

AUX[0D] Address Pitch Adjustment Register I/O address = 1101b, Read/Write.							
Addr Pitch Adjustment Bit 7	Addr Pitch Adjustment Bit 6	Addr Pitch Adjustment Bit 5	Addr Pitch Adjustment Bit 4	Addr Pitch Adjustment Bit 3	Addr Pitch Adjustment Bit 2	Addr Pitch Adjustment Bit 1	Addr Pitch Adjustment Bit 0

bits 7-0 Addr Pitch Adjustment Bits [7:0]
These bits set the numerical difference between the last address of a display line, and the first address in the following line.

If the Address Pitch Adjustment is not equal to zero, then a virtual screen is formed. The size of the virtual screen is only limited by the available display memory. The actual display output is a window that is part of the whole image stored in the display memory. For example, with 128K of display memory, a 640x400 16-gray image can be stored. If the output display size is 320x240, then the whole image can be seen by changing display starting addresses through AUX[06] and [07], and AUX[08] and [09]. Note that a virtual screen can be produced on either a single or dual panel.

In 8-bit memory interface, if the Address Pitch Adjustment is not equal to zero, then a virtual screen with a line length of (Line Byte Count +AUX[0D]) bytes is created, with the display reflecting the contents of a window (Line Byte Count+1) bytes wide. The position of the window on the virtual screen is determined by AUX[06] and [07], and AUX[08] and [09].

In 16-bit memory interface, if the Address Pitch Adjustment is not equal to zero, then a virtual screen with a line length of 2*(Line Byte Count +AUX[0D]) bytes is created, with the display reflecting the contents of a window 2*(Line Byte Count+1) bytes wide. The position of the window on the virtual screen is determined by AUX[06] and [07], and AUX[08] and [09].

5.1.2 Description

The SED1352 can be programmed to use memory offsets in such a way that the physical display behaves as a viewport into a much larger "virtual" memory space. This viewport can be panned and/or scrolled to display this larger memory space.

Referring to the figure below, a virtual image of 640x480 can be viewed by navigating the 320x240 viewport around the image by panning and scrolling

.

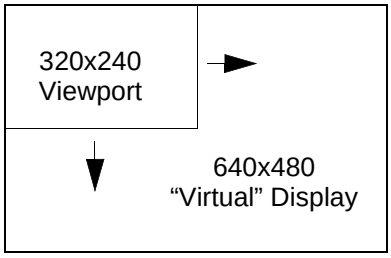


Figure 8: 640 x 480 Virtual Display

To create a virtual display, the Address Pitch Adjustment Register must be programmed to indicate the horizontal size of the larger, "virtual" image stored in display memory. The Address Pitch Adjustment Register tells the SED1352 how many bytes or words of display memory are part of the nonvisible region of display memory (see Example 10).

Example 10: Program the Address Pitch Adjustment Register to support a 16 gray shade 640x480 virtual display on a 320x240 LCD panel; the Memory Interface is 16 bits.

1. Initialize the SED1352 registers for a 320x240 panel.
2. Determine whether the Address Pitch Adjustment Register refers to bytes or words.
   Since the Memory Interface is set to 16 bits, the Address Pitch Adjustment Register refers to words.
3. Determine the number of pixels per unit referred to by the Address Pitch Adjustment Register.
   The Address Pitch Adjustment Register refers to units of words, so find the number of pixels per word.

   16 gray shades => 4 bits per pixel

   4 bits per pixel => 2 pixels per byte

   $$\text{pixels per word} = (\text{pixels per byte}) \times 2 = 2 \times 2 = 4 \text{ pixels per word}$$

4. Calculate the number of pixels on a horizontal scan line not visible.

   $$(\text{virtual display width in pixels}) - (\text{panel width in pixels}) = 640 - 320 = 320 \text{ hidden pixels}$$

   Consequently on a screen update the SED1352 will show the first 320 of 640 pixels, and then ignore the remaining 320 pixels in order to reach the next scan line.

5. Program the Address Pitch Adjustment Register:

   $$\frac{\text{number of hidden horizontal pixels}}{\text{pixels per word}} = \frac{320}{4} = 80 \text{ words} = 50\text{h words}$$

   Therefore AUX[0Dh] = 50h

6. To view the rest of the image refer to Section 5.5, *"Panning and Scrolling"* on page 42, keeping in mind that the horizontal width is 640 pixels, not 320.

5.2 Bitmaps and Text Displays

For the scope of this guide, a bitmap is a data structure which represents the image shown on the LCD. The bitmap includes the dimensions of the image, and the gray shade palette used to program the look-up table. Text is shown by creating a font, which in this example is a series of bitmaps, one bitmap per alphanumeric character.

Example 11: Display the word "TEXT" on a 16 gray shade 320x240 LCD panel; the Memory Interface is 16 bits.

1. Define the font for the letters 'T', 'E', and 'X'.
   Each character is 8x8 pixels, with at least one horizontal and vertical side left blank for spacing.

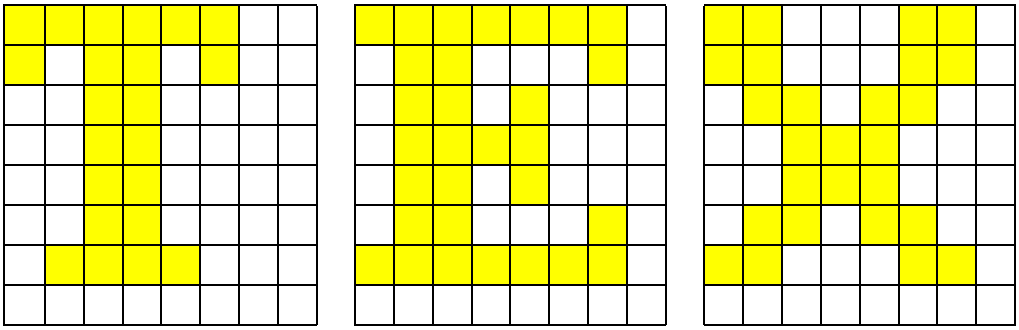


Figure 9: Font for the Message "TEXT"

2. Program the Look-up Table.
   See Example 3, "Initialize the Look-Up Table," on page 17.
3. Calculate the display memory map.
   See Figure 5, "Memory Map for 320 x 240 LCD Panel with 16 Gray Shades," on page 25.
4. Write font to display memory.
   In a general purpose program the entire bitmapped font would be placed in an array. As characters are to be displayed, the program would choose the appropriate bitmap, select the proper position on the screen, and write to display memory. For this example assume that the program has already selected the proper bitmaps and the correct positions in display memory (there is a detailed programming example later in this guide; see Section 6.3, *"Advanced Functions"* on page 52).

   Each highlighted pixel in the text bitmap will be shown at maximum intensity, which is pixel value 15. The text, for simplicity, will be shown in the upper left corner of the screen. When the program has completed writing the pixels for the word "TEXT," the display memory will have the data shown in Figure 10. In this figure the bytes are grouped within vertical lines.

Offset (hex)																													Offset (hex)
0000	F	F	F	F	F	F	0	0	F	F	F	F	F	F	F	0	F	F	0	0	0	F	F	0	F	F	F	F	F	F	0	0	000F
00A0	F	0	F	F	0	F	0	0	0	F	F	0	0	0	F	0	F	F	0	0	0	F	F	0	F	0	F	F	0	F	0	0	00AF
0140	0	0	F	F	0	0	0	0	0	F	F	0	F	0	0	0	0	F	F	0	F	F	0	0	0	0	F	F	0	0	0	0	014F
01E0	0	0	F	F	0	0	0	0	0	F	F	F	F	0	0	0	0	0	F	F	F	0	0	0	0	0	F	F	0	0	0	0	01EF
0280	0	0	F	F	0	0	0	0	0	F	F	0	F	0	0	0	0	0	F	F	F	0	0	0	0	0	F	F	0	0	0	0	028F
0320	0	0	F	F	0	0	0	0	0	F	F	0	0	0	F	0	0	F	F	0	F	F	0	0	0	0	F	F	0	0	0	0	032F
03C0	0	F	F	F	F	0	0	0	F	F	F	F	F	F	F	0	F	F	0	0	0	F	F	0	0	F	F	F	F	0	0	0	03CF
0460	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	046F

Figure 10: Display Memory Contents for Message "TEXT"

5.3 Registers

The SED1352 has an internal set of sixteen 8-bit read/write registers which configure it for various modes of operation. The registers can be accessed in two ways: Indexed Addressing and Direct Addressing.

Note
Refer to the *SED1352 Hardware Functional Specification* and *SDU1352B0x Evaluation Board User's Manual* for more information on the SED1352 registers.

5.3.1 Indexed Addressing

This method requires only two sequential I/O address locations starting from the base I/O address. The base I/O address is determined by the power-on state of the SRAM data lines VD[4-12]. See Table 5-6 in the *SED1352 Hardware Functional Specification*, X16-SP-001-xx.

The two sequential I/O addresses are defined as Index Address and Data. To access registers using this method, an Index Address must be written to the first I/O address location allowing data to be written/read to/from the second I/O address.

Example 12: Write 12h to register 08h on the SDU1352B0x evaluation board; the base port address is 310h, and indexed port mapping is used.

1. Write 08h to the index register.
   The index register is at base port address + 0 = 310h.

```
MOV DX,310h
MOV AL,08h
OUT DX,AL
```

2. Write 12h to the data register.
   The data register is at base port address + 1 = 311h.

```
MOV DX,311h
MOV AL,12h
OUT DX,AL
```

5.3.2 Direct Addressing

This method of addressing requires 16 sequential I/O addresses starting from the base I/O address. The base I/O address is determined by the power-on state of the SRAM data lines VD[7-12]. See Table 5-6 in the *SED1352 Hardware Functional Specification*, X16-SP-001-xx.

To access the internal 16 registers of the SED1352, simply perform I/O read/write functions to the absolute address as defined in the previous paragraph.

Example 13: Write 12h to register 08h on the SDU1352B0x evaluation board; the base port address is 310h, and direct port mapping is used.

1. Calculate the port address for register 08h.

   port address = 310h + 8h = 318h

2. Write the value 12h to port address 318h.

```
MOV DX,318h
MOV AL,12h
OUT DX,AL
```

Note
The SDU1352B0x is normally configured for register indexing, not direct mapping. Refer to the *SDU1352B0x Evaluation Board User's Manual* for more information configuring the SDU1352B0x board for register indexing or register direct mapping.

5.4 Split Screen

This section describes how to create a split screen for both single and dual LCD panels. For single panel displays, the Screen 1 Display Line Count Registers are used. For dual panel displays, the Screen 2 Display Start Address Registers are used.

5.4.1 Registers

AUX[0A] Screen 1 Display Line Count Register (LSB) I/O address = 1010b, Read/Write.							
Screen 1 Display Line Count Bit 7	Screen 1 Display Line Count Bit 6	Screen 1 Display Line Count Bit 5	Screen 1 Display Line Count Bit 4	Screen 1 Display Line Count Bit 3	Screen 1 Display Line Count Bit 2	Screen 1 Display Line Count Bit 1	Screen 1 Display Line Count Bit 0

This register is used to enable the split screen display feature (single panel only) where two different images can be displayed at the same time on one display. This register has no effect when using a dual panel configuration.

bits 7-0 Screen 1 Display Line Count Bits [7:0]
These bits are the seven LSB of a 9-bit value used to determine the number of lines displayed for screen 1. The remaining lines will automatically display from the screen 2 display start address. The 9-bit value programmed is the number of display lines -1.

For example, if AUX[0A] = 20h for a 320x240 display system. The display will show 20h+1 = 33 lines on the upper part of the screen according to display starting address AUX[06] and AUX[07] and 240 - 33 = 207 lines on the lower part of the screen according to display starting address AUX[08] and AUX[09].

Two different images can be displayed when using a dual panel configuration by changing the screen 2 display start address. However, by using this method screen 2 is limited to the lower half of the display.

AUX[0B] Screen 1 Display Line Count Register (MSB) I/O address = 1011b, Read/Write.							
n/a	n/a	n/a	n/a	n/a	n/a	Screen 1 Display Line Count Bit 9	Screen 1 Display Line Count Bit 8

bits 1-0 Screen 1 Display Line Count Bits [9:8]
These are the two MSB of the Screen 1 Display Line Count Register.

5.4.2 Description

A split screen is generally considered as the presentation of two different images on the screen. Image 1 is shown on the top half and image 2 is shown on the bottom half of the screen. Due to the design the SED1352, the system is *always* in split screen mode. If only one image is to be shown, either image 2 is hidden or image 2 appears as part of image 1; this depends on whether a single or dual panel LCD is in use.

5.4.3 Single Panel LCD

The following is the procedure to show a split screen image on a 16 gray shade 320 x 240 single panel LCD. For this example the SDU1352B0x is used; the Memory Interface is set to 16 bits, and 128k of display memory is available. In addition, the two images shown on the split screen are each 320x240; only a portion of each image is shown.

1. Determine whether the Display Start Address Registers refer to bytes or words.
   Since the Memory Interface is set to 16 bits, the Display Start Address Registers refer to words. Note that when addresses refer to words, the image must be aligned in memory such that the beginning is found on a word boundary (the least significant bit of the memory address must be 0).

2. Calculate the number of bytes per scan line.

   16 gray shades => 4 bits per pixel

   4 bits per pixel => 2 pixels per byte

   $$\text{number of bytes per scan line} = \frac{\text{pixels per scan line}}{\text{pixels per byte}} = \frac{320}{2} = 160 \text{ bytes per scan line} = 00A0h \text{ bytes per scan line}$$

3. Determine the display memory location for image 1.
   For simplicity, assign the beginning of display memory as the starting address of image 1 (see Figure 11). For the SDU1352B0x, this address is C000:0000h.

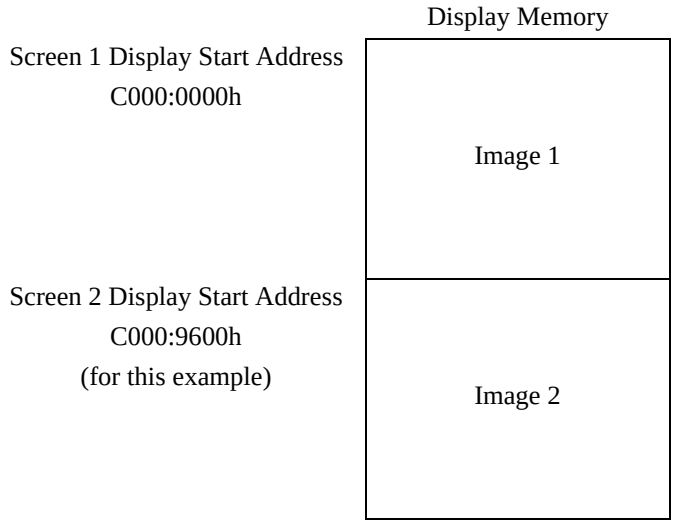


Figure 11: Memory Map for Split Screen

4. Program the Screen 1 Display Start Address Register to point to the beginning of image 1.
   Since image 1 is at the beginning of display memory for a 128k system, program the Screen 1 Display Start Address Register to 0000h.
   AUX[06h] = 00h
   AUX[07h] = 00h

5. Calculate the total number of bytes required for image 1.

$$(\text{bytes per scan line}) \times (\text{number of scan lines for image 1}) = 160 \times 240 = 38400 \text{ bytes} = 9600\text{h bytes}$$

6. Determine the display memory location for image 2.
Place image 2 immediately after image 1 (see Figure 11). Assign the starting address for image 2 as follows:

$$\begin{aligned}\text{image 2 address} &= (\text{base display memory address}) + (\text{size of image 1}) \\ &= \text{C000:0000h} + \text{0000:9600h} \\ &= \text{C000:9600h}\end{aligned}$$

7. Program the Screen 2 Display Start Address Register to point to the beginning of image 2.
Image 2 is placed right after image 1, as shown below:

$$\text{Screen 2 Display Start Address} = \text{Screen 1 Display Start Address} + \frac{\text{size of image 1 in bytes}}{\text{2 bytes per word}}$$

$$= 0000\text{h} + \frac{9600\text{h}}{2} = 4\text{B00h}$$

AUX[08h] = 00h
AUX[09h] = 4Bh

8. Program the Screen 1 Display Line Count Register.
The Display Line Count Register indicates how many lines of the first screen should be shown *minus 1*. By changing the line count, image 2 appears to move up or down the display.

   - If the line count is set to the maximum number of visible scan lines - 1, only image 1 is shown.

$$\text{visible scan lines} - 1 = 240 - 1 = 239 = \text{00EFh}$$

     AUX[0Ah] = LSB of (visible scan lines - 1) = 0EFh
     AUX[0Bh] = MSB of (visible scan lines - 1) = 00h

   - If the line count is set to 0, then the first scan line of image 1 is shown followed by the first part of image 2.
     AUX[0Ah] = 00h
     AUX[0Bh] = 00h

     It is not possible to show only image 2 by changing the line count. If only image 2 needs to be shown, reprogram the Screen 1 Display Start Address Registers to point to the beginning of image 2, and set the line count to the maximum number of visible scan lines - 1.

- If the line count is set to 99, then the first 100 scan lines of image 1 are shown, following by the first part of image 2 (see Figure 12).
  AUX[0Ah] = 63h (99 decimal)
  AUX[0Bh] = 00h

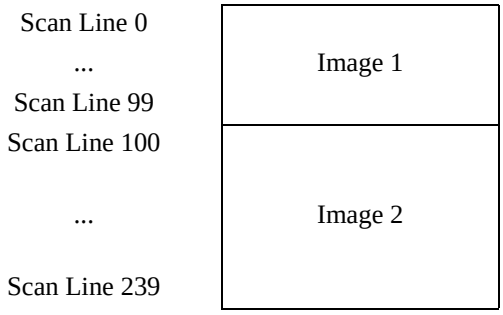


Figure 12: 320 x 240 Single Panel for Split Screen

9. Write both image 1 and image 2 to their respective locations in display memory.

5.4.4 Dual Panel LCD

The following is the procedure to show a split screen image on a 4 gray shade 640x480 dual panel LCD. For this example the SDU1352B0x is used; the Memory Interface is set to 16 bits, and 128k of display memory is available. In addition, the two images shown on the split screen are each 640x240.

1. Determine whether the Display Start Address Registers refer to bytes or words.
   Since the Memory Interface is set to 16 bits, the Display Start Address Registers refer to words. Note that when addresses refer to words, the image must be aligned in memory such that the beginning is found on a word boundary (the least significant bit of the memory address must be 0).

2. Calculate the number of bytes per scan line.

   4 gray shades => 2 bits per pixel

   2 bits per pixel => 4 pixels per byte

   $$\text{number of bytes per scan line} = \frac{\text{pixels per scan line}}{\text{pixels per byte}} = \frac{640}{4} = 160 \text{ bytes per scan line} = 00A0h \text{ bytes per scan line}$$

3. Determine the display memory location for image 1.
   For simplicity, assign the beginning of display memory as the starting address of image 1 (see Figure 11). For the SDU1352B0x, this address is C000:0000h.

4. Program the Screen 1 Display Start Address Register to point to the beginning of image 1.
   Since image 1 is at the beginning of display memory for a 128k system, program the Screen 1 Display Start Address Register to 0000h.
   AUX[06h] = 00h
   AUX[07h] = 00h

5. Calculate the total number of bytes required for image 1.

   $$(\text{bytes per scan line}) \times (\text{number of scan lines for image 1}) = 160 \times 240 = 38400 \text{ bytes} = 9600h \text{ bytes}$$

6. Determine the display memory location for image 2.
   Place image 2 immediately after image 1 (see Figure 11). Assign the starting address for image 2 as follows:

   $$\begin{aligned}\text{image 2 address} &= (\text{base display memory address}) + (\text{size of image 1}) \\ &= C000:0000h + 0000:9600h \\ &= C000:9600h\end{aligned}$$

7. Program the Screen 2 Display Start Address Register to point to the beginning of image 2.
   Image 2 is placed right after image 1, as shown below:

   $$\text{Screen 2 Display Start Address} = \text{Screen 1 Display Start Address} + \frac{\text{size of image 1 in bytes}}{\text{2 bytes per word}}$$

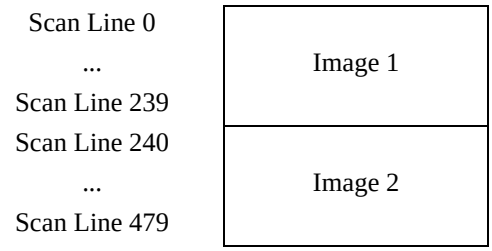
   $$= 0000h + \frac{9600h}{2} = 4B00h$$

   AUX[08h] = 00h
   AUX[09h] = 4Bh

8. Write both image 1 and image 2 to their respective locations in display memory.

Notes

When using a dual panel, the Screen 1 Display Line Count Register is ignored by the SED1352. Once the two Display Start Address Registers are programmed, the top panel will show the beginning of image 1, and the bottom panel will show the beginning of image 2 (see Figure 13).



Screen 1 Display Line Count is ignored; Image 1 always has half the total number of scan lines (240 in this example).

Figure 13: 640 x 480 Dual Panel for Split Screen

Each image can be scrolled or panned by appropriate programming of the respective Display Start Address Registers. The following are some examples:

- To scroll image 1 up, the Screen 1 Start Address Register must point to the following scan line.

$$\text{Screen 1 Display Start Address} = \text{Screen 1 Display Start Address} + \frac{\text{number of bytes per scan line}}{\text{2 bytes per word}}$$

  AUX[06h] = LSB of Screen 1 Display Start Address
  AUX[07h] = MSB of Screen 1 Display Start Address

- To scroll image 2 down, the Screen 2 Start Address Register must point to the previous scan line.

$$\text{Screen 2 Display Start Address} = \text{Screen 2 Display Start Address} - \frac{\text{number of bytes per scan line}}{\text{2 bytes per word}}$$

  AUX[08h] = LSB of Screen 2 Display Start Address
  AUX[09h] = MSB of Screen 2 Display Start Address

- To pan image 1 to the right by a group of pixels, the Screen 1 Start Address Register must be increased by 1.

$$\text{Screen 1 Display Start Address} = \text{Screen 1 Display Start Address} + 1$$

  AUX[06h] = LSB of Screen 1 Display Start Address
  AUX[07h] = MSB of Screen 1 Display Start Address
  See Section 5.5.2, *"Panning Right and Left"* on page 42 for more information.

- To pan image 2 to the left by a group of pixels, the Screen 2 Start Address Register must be decreased by 1.

$$\text{Screen 2 Display Start Address} = \text{Screen 2 Display Start Address} - 1$$

  AUX[08h] = LSB of Screen 2 Display Start Address
  AUX[09h] = MSB of Screen 2 Display Start Address
  See Section 5.5.2, *"Panning Right and Left"* on page 42 for more information.

5.4.4.1 Displaying a Single Image on a Dual Panel

The following is the procedure to show a single image on a dual panel LCD. In this procedure the single image is broken into two smaller images; image 1 is placed on the top panel and image 2 is placed on the bottom panel. For this example the SDU1352B0x is used with a 4 gray shade 640x480 dual panel LCD; the Memory Interface is set to 16 bits, and 128k of display memory is available.

1. Determine whether the Display Start Address Registers refer to bytes or words.
   Since the Memory Interface is set to 16 bits, the Display Start Address Registers refer to words. Note that when addresses refer to words, the image must be aligned in memory such that the beginning is found on a word boundary (the least significant bit of the memory address must be 0).

2. Calculate the number of bytes per scan line.

   4 gray shades => 2 bits per pixel

   2 bits per pixel => 4 pixels per byte

   $$\text{number of bytes per scan line} = \frac{\text{pixels per scan line}}{\text{pixels per byte}} = \frac{640}{4} = 160 \text{ bytes per scan line} = 00A0h \text{ bytes per scan line}$$

3. Determine the display memory location for image 1.
   For simplicity, assign the beginning of display memory as the starting address of image 1 (see Figure 14). For the SDU1352B0x, this address is C000:0000h.

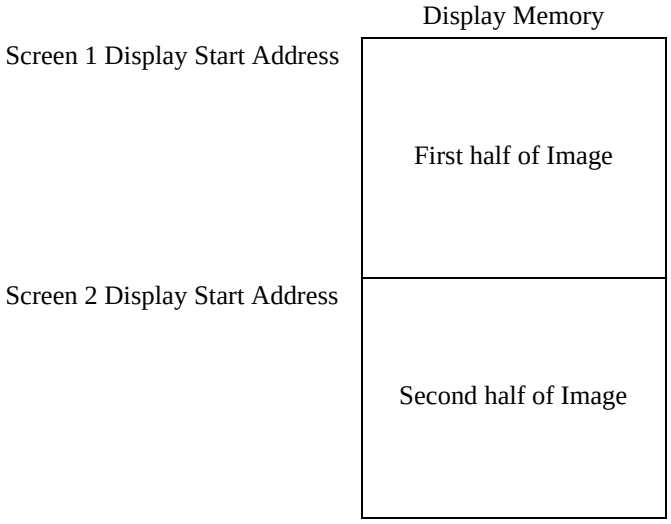


Figure 14: Memory Map for a Dual Panel showing a Single Image

4. Program the Screen 1 Display Start Address Register to point to the beginning of image 1.
   Since image 1 is at the beginning of display memory for a 128k system, program the Screen 1 Display Start Address Register to 0000h.
   AUX[06h] = 00h
   AUX[07h] = 00h

5. Determine the size of image 1.

   $$\text{vertical size of image 1} = \text{vertical size of panel 1} = \frac{\text{number of scan lines in display}}{2} = \frac{480}{2} = 240 \text{ scan lines}$$

   $$\text{size} = \frac{\text{display width in pixels}}{\text{pixels per byte}} \times (\text{number of scan lines in image 1}) = \frac{640}{4} \times 240 = 38400 \text{ bytes} = 9600h \text{ bytes}$$

6. Determine the display memory location for image 2.
   Place image 2 immediately after image 1 (see Figure 14). Assign the starting address for image 2 as follows:

$$\begin{aligned}\text{image 2 address} &= (\text{base display memory address}) + (\text{size of image 1}) \\ &= \text{C000:0000h} + \text{0000:9600h} \\ &= \text{C000:9600h}\end{aligned}$$

7. Program the Screen 2 Display Start Address Register to point to the beginning of image 2.
   Image 2 is placed right after image 1, as shown below:

$$\text{Screen 2 Display Start Address Register} = \text{Screen 1 Display Start Address Register} + \frac{\text{size of image 1 in bytes}}{\text{2 bytes per word}}$$

$$= 0000\text{h} + \frac{9600\text{h}}{2} = 4\text{B00h}$$

   AUX[08h] = 00h
   AUX[09h] = 4Bh

8. Write both image 1 and image 2 to their respective locations in display memory.

5.5 Panning and Scrolling

Panning and scrolling are typically used to show an image which is too large to be shown completely on an LCD panel. Although the image is stored entirely in display memory, only a small portion is actually visible on the LCD panel. This visible portion is called the *viewport*; the user moves this viewport over different portions of the image by panning and scrolling. *Panning* moves the viewport right or left. *Scrolling* moves the viewport up or down.

5.5.1 Initialization

To pan and scroll over a large image, the SED1352 registers must first be initialized and the image written to display memory. To do so, initialize the registers as described in Section 2, *"INITIALIZING THE SED1352"* on page 8, but with the following exception: the Address Pitch Adjustment Register in the SED1352 must be set to create a virtual display; see Section 5.1, *"Virtual Displays"* on page 28 for more information.

5.5.2 Panning Right and Left

To pan to the right, increase the value in the Screen 1 Display Start Address Register. To pan to the left, decrease the value in the Screen 1 Display Start Address Register.

Note that the SED1352 can pan right or left by either 2, 4, or 8 pixels. This is because the Screen 1 Display Start Address Register refers to either bytes or words (see Section 4.2.1, *"SDU1352B0x Evaluation Board Display Memory"* on page 24), and a byte can represent either 2 or 4 pixels, and so a word can represent 4 or 8 pixels; see Table 5-1 below:

Table 5-1: Smallest Number of Pixels for Panning

Memory Interface	Gray Levels	Pixels per Byte	Smallest Number of Pixels for Panning
8 bits	4	4	4
	16	2	2
16 bits	4	4	8
	16	2	4

5.5.3 Scrolling Up and Down

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

A virtual scan line is in reference to a virtual display, in which an image larger than the physical size of the LCD is stored. The number of bytes in a virtual scan line is the number of bytes required to store one horizontal line of pixels in the virtual image.

Example 14: Scroll down one line for a 16 gray shade 640x200 virtual image using a 320x240 single panel LCD. The Memory Interface is 16 bits, and 64k of display memory is available. Also describe how to scroll in a dual panel LCD.

1. Calculate the number of bytes in a virtual scan line.

$$\frac{\text{number of horizontal pixels in virtual image}}{\text{number of pixels per word}} = \frac{640 \text{ pixels per scan line}}{2 \text{ pixels per byte}} = 320 \text{ bytes per scan line}$$

2. Add the number of *words* in a virtual scan line to the Screen 1 Display Start Address Register.
   In this example the Screen 1 Display Start Address points to the beginning of the image.

$$\text{Screen 1 Display Start Address} = \text{Screen 1 Display Start Address} + \frac{\text{number of bytes in a virtual scan line}}{\text{2 bytes per word}}$$

$$= 8000\text{h} + \frac{320}{2}$$

$$= 80\text{A0h}$$

3. Program the Screen 1 Display Start Address.
   AUX[06h] = A0h
   AUX[07h] = 80h

4. **FOR DUAL PANELS ONLY**
   Add the number of *words* in a virtual scan line to the Screen 2 Display Start Address Register.
   In this example the Screen 2 Display Start Address has previously been initialized as described in Section 5.4.4.1, *"Displaying a Single Image on a Dual Panel"* on page 40.

$$\text{Screen 2 Display Start Address} = \text{Screen 2 Display Start Address} + \frac{\text{number of bytes in a virtual scan line}}{\text{2 bytes per word}}$$

5. **FOR DUAL PANELS ONLY**
   Program the Screen 2 Display Start Address.
   AUX[08h] = least significant byte of "Screen 2 Display Start Address"
   AUX[09h] = most significant byte of "Screen 2 Display Start Address"

5.6 Power Saving

The following section introduces the power saving capabilities of the SED1352. A detailed description of the Power Save Register is provided, followed by a description of the power save modes.

5.6.1 Registers

Note
Register bits discussed in this section are highlighted.

AUX[03] Line Byte Count (MSB [bit 8] for 16-level gray scale mode only) / Power Save Register I/O address = 0011b, Read/Write							
PS Bit 1	PS Bit 0	LCD Signal State	LUT Bypass	n/a	n/a	n/a	Line Byte Count Bit 8

bits 7-6 PS Bits [1:0]
Selects the Power Save Modes as shown in the following table. The PS bits [1:0] go to 0 on RESET.

Table 5-2: Power Save Mode Selection

PS1	PS0	Mode Activated
0	0	Normal Operation
0	1	Power Save Mode 1
1	0	Power Save Mode 2
1	1	Reserved

Refer to Section 5.6.2, *"Power Save Modes"* on page 44 for a complete Power Save Mode description.

5.6.2 Power Save Modes

Two software-controlled Power Save Modes have been incorporated into the SED1352 to accommodate the important need for power reduction in hand-held devices market. These modes can be enabled by setting the 2 Power Save bits (AUX[03h] bits 7-6).

The various settings are:

Table 5-3: Power Save Mode Selection

Bit 7	Bit 6	Mode Activated
0	0	Normal Operation
0	1	Power Save Mode 1
1	0	Power Save Mode 2
1	1	Reserved

5.6.2.1 Power Save Mode 1

Power Save Mode 1 would typically be used when power savings are required and memory accesses may occur. The disadvantage is that since the oscillator is running, this mode consumes more power that Power Save Mode 2.

5.6.2.2 Power Save Mode 2

Power Save Mode 2 is typically used when memory accesses would not occur.

5.6.2.3 Power Save Mode Function Summary

Table 5-4: Power Save Mode Function Summary

Function	Power Save Mode (PSM)			
	Normal (Active)	PSM1		PSM2
		State 1	State 2	
Display Active?	Yes	No	No	No
I/O Access Possible?	Yes	Yes	Yes	Yes
Memory Access Possible?	Yes	Yes	No	No
Sequence Controller Running?	Yes	No	No	No
Internal Oscillator Disabled?	No	No	No	Yes

Note

1. When programming the PS bits do a read/modify/write operation so as not to destroy any other data in the register.
2. Refer to the programming example in Advanced Functions on page 52.

6 PROGRAMMING THE SED1352

The purpose of this section is to show how to program the SED1352 exercising the specific capabilities of this chip. A series of functions written in 'C' will be presented, each illustrating a basic feature of the SED1352. These functions are written for the SDU1352B0x evaluation board, and are combined under a menu-driven program called DEMO.EXE.

Note
The sample code will not run on a display larger than 320x240, and will use 16 gray shades in most of the examples.

This program accepts the following command line options:

DEMO type x=n y=n p=n

where: **type** = **SINGLE** | **DUAL**

x = horizontal panel size in pixels from 1 to 320 (decimal)

y = vertical panel size in pixels from 1 to 240 (decimal)

p = **300** | **310**...**360** | **370** (port address in hex) (I/O indexed addressing selected by default)

For example, if there is a 320x240 single panel LCD with a port address of 310h, type

DEMO SINGLE x=320 y=240 p=310

When DEMO is started, output will be sent to the standard output device. This output will present a menu of numbered options:

```
SDU1352B0x DEMO PROGRAM
Press 1 to read registers
Press 2 to show gray shade bar
Press 3 to show split screen
Press 4 to show panning and scrolling
Press 5 to start power saving
Press ESC to quit
```

Figure 15: Display for DEMO.EXE

6.1 Main Loop Code

```
//-------------------------------------------------------------------------
//
// FUNCTION: main()
//
// DESCRIPTION: Start of demo program.
//
// INPUTS: Command line arguments.
// RETURN VALUE: None.
//
//-------------------------------------------------------------------------

void main(char argc, char **argv)
{
int ch;

CheckArguments(argc, argv);
printf("Initializing\n");
Initialize();
ClearLCDScreen();
ShowMenu();

while ((ch = getch()) != ESC)
   {
   switch (ch)
      {
      case '1':
         ShowRegisters();
         break;

      case '2':
         GrayShadeBars();
         break;

      case '3':
         SplitScreen();
         break;

      case '4':
         PanScroll();
         break;

      case '5':
         PowerSaving();
         break;

      case ESC:
         exit(0);
      }
   }
}
```

6.2 Initialization Code

```
//-----------------------------------------------------------------------
//
// FUNCTION: Initialize()
//
// DESCRIPTION: Intialize SED1352 registers.
//
// INPUTS: This function looks at the followingl global variables to
//         determine the appropriate register settings:
//            PanelX, PanelY, PanelType
//
// OUTPUTS: The following global variables are changed:
//            PanelGrayLevel, BytesPerScanLine
//
//-----------------------------------------------------------------------

void Initialize(void)
{
static unsigned int val;
static unsigned int x;

PanelGrayLevel = 16;

//--------------------------------------

//
// Mode Register:
//   Display = ON
//   Panel = SINGLE
//   Mask XSCL = NOT MASKED
//   LCDE = NOT ENABLED
//   Gray Scale = 16 Gray Shades (4 bits/pixel)
//   LCD Data Width = 8 bit data transfer
//   Memory Interface = 16 bits
//   RAMS = Addressing for 8Kx8 SRAM
//
val = 0x8C;


if (PanelType == TYPE_DUAL)
   {
   val |= 0x40;   // Set panel type to DUAL
   val &= ~0x04;  // Set LCD Data Width to 4 bit data transfer
   }

WriteRegister(1, val);   // Write to Mode Register

//--------------------------------------

//
// Line Byte/Word Count Register
//
```

```
// Bits 0-7 are in AUX[2], Bit 8 is in AUX[3].
//
// Because the Memory Interface is set to 16 bits, the
// Line Byte/Word Count Register counts in words.  In addition,
// there are 2 pixels/byte since there are 16 gray levels.
// To calculate the number of words in a scan line, use the following
// formula:
//
//        number of pixels per scan line
//       ------------------------------- - 1
//       (2 pixels/byte) * (2 bytes/word)
//
val = (PanelX / 4) - 1;              // For 16 gray shades only

WriteRegister(2, val & 0xff);        // Line Byte/Word Count Register
WriteRegister(3, (val >> 8) & 0x01); // Line Byte/Word Count/Power Save Reg


//
// BytesPerScanLine is a global variable
//
BytesPerScanLine = (PanelX / 2);     // For 16 gray shades only

//-------------------------------------

//
// Total Display Line Count Register
// Screen 1 Display Line Count Register
//
// To show a full image on Screen 1, copy the Total Display Line Count
// into the Screen 1 Display Line Count.
//

//
// Assume that all panels smaller than 400 lines are in 4 bit mode
//
if (PanelY < 400)
   {
   val = ReadRegister(1);
   val &= ~0x04;
   WriteRegister(1, val);  // Write to Mode Register; LCD Data Width = 4 bits
   }

val = PanelY;

//
// A dual panel LCD will, of course, have two panels.  Each panel will
// show either the top or bottom half of the image, which is half of the
// vertical resolution.
//
if (PanelType == TYPE_DUAL)
   val /= 2;
```

```
--val;

WriteRegister(4, val & 0xff);    // Write to Total Display Line Count Reg
WriteRegister(0x0a, val & 0xff); // Write to Screen 1 Display Line Count Reg
WriteRegister(5, (val >> 8) & 0x03); // Total Disp Line Cnt (MSB)/WF Count Reg
WriteRegister(0x0b, (val >> 8) & 0x03);  // Scrn 1 Disp Line Count Reg (MSB)

//--------------------------------------

//
// Set Screen 1 Display Start Address to beginning of video memory
//
WriteRegister(6, 0);  // Write to Screen 1 Display Start Address Register
WriteRegister(7, 0);

//--------------------------------------

//
// Screen 2 Display Start Address Register
//
// If using a dual panel, the Screen 2 Display Start Address must point
// to the second half of the image in video memory.
//
if (PanelType == TYPE_DUAL)
   {
   val = (unsigned int) ((ReadRegister(3) & 0x01) << 8) | ReadRegister(2);
   ++val;

   val *= (PanelY / 2);
   WriteRegister(8, val & 0xff);
   WriteRegister(9, val >> 8);
   }
else
   {
   //
   // On a single panel, Screen 1 was programmed to show all of its
   // lines.  Consequently Screen 2 will not be seen, and so the
   // Screen 2 Display Start Address will have no observable effect.
   // For convenience, set the screen 2 address to 0.
   //
   WriteRegister(8, 0);
   WriteRegister(9, 0);
   }

//--------------------------------------

//
// When the SDU1352B0x is set to 64k, video memory exists from
// D000:0000 to D000:FFFF.  When the SDU1352B0x is set to 128k, video
// memory exists from C000:0000 to D000:FFFF.  As far as the SED1352
// is concerned, video memory ALWAYS begins at C000:0000, even if
// there is no physical memory present.
//
```

```
// Since this demo program uses only 64k, the Display Start Address
// Registers must be adjusted to point to the D000 segment.  To do so,
// note that these registers refer to words of data, not bytes,
// since the Memory Interface is set to 16 bits.  Consequently adding
// the value 8000h (words) to the address registers will effectively
// add 10000h (bytes) to the address.  Adding 10000h to C000:0000 will
// point to D000:0000, which is why this address correction works.
//
WriteRegister(7, 0x80); // MSB of Screen 1 Display Start Address

val = ReadRegister(9);  // MSB of Screen 2 Display Start Address
val += 0x80;
WriteRegister(9, val);

//--------------------------------------

//
// Set Address Pitch Adjustment to 0
//
WriteRegister(0x0d, 0);  // Write to Address Pitch Adjustment Register

//--------------------------------------

//
// Update Look-Up Table for 16 gray shades
//
for (x = 0; x < 16; ++x)
   {
   WriteRegister(0x0e, x);
   WriteRegister(0x0f, x);
   }

//--------------------------------------

//
// Now that system is initialized, enable LCDE
//
val = ReadRegister(1);
val |= 0x10;   // LCDE enabled
WriteRegister(1, val);
}
```

6.3 Advanced Functions

```
#define VIRTUAL_X    (360)
#define VIRTUAL_Y    (360)

//-------------------------------------------------------------------------
//
// FUNCTION: ShowRegisters()
//
// DESCRIPTION: Shows the contents of the SED1352 registers.
//
// INPUTS: None.
// RETURN VALUE: None.
//
//-------------------------------------------------------------------------

void ShowRegisters(void)
{
static unsigned char x;

printf("SED1352 Registers: ");

for (x = 0; x < 16; ++x)
   printf("%02X ", ReadRegister(x));

printf("\nSED1352 Look-Up Table: ");

for (x = 0; x < 16; ++x)
   {
   WriteRegister(0x0e, x);
   printf("%02X ", ReadRegister(0x0f));
   }

ShowMenu();
}

//-------------------------------------------------------------------------
//
// FUNCTION: GrayShadeBars()
//
// DESCRIPTION: Displays one set of vertical bars, each with a
//              different gray shade.
//
// INPUTS: None.
//
// RETURN VALUE: None.
//
//-------------------------------------------------------------------------

void GrayShadeBars(void)
{
static unsigned int val, x;
static unsigned char _far *pVideo;
```

```
Initialize();
ClearLCDScreen();


//
// For 64k only
//
FP_SEG(pVideo) = 0xd000;
FP_OFF(pVideo) = 0x0000;


//
// Update Look-Up Table for 16 gray shades
//
for (x = 0; x < 16; ++x)
   {
   WriteRegister(0x0e, x);
   WriteRegister(0x0f, x);
   }

//
// Change Mode Register for 16 gray shades
//
val = ReadRegister(1);
val |= 0x08;
WriteRegister(1, val);

//
// Update Line Byte Count register for 16 gray shades
//
// Since 16 gray shades corresponds to 2 pixels per byte, there
// are ((x horizontal pixels)/2) bytes per scan line.  This means that
// there are ((x horizontal pixels)/4) words per scan line.
//
// Since the Memory Interface is set to 16 bits, the Line Byte/Word Count
// refers to words.
//
val = (PanelX / 4) - 1;
BytesPerScanLine = (PanelX / 2);

WriteRegister(2, val & 0xff);         // Line Byte Count Register
WriteRegister(3, (val >> 8) & 0x01);  // Line Byte Count/Power Save Reg

PanelGrayLevel = 16;
ShowVerticalBars(pVideo);

//
// Show text.  The lightest gray shade is set to PanelGrayLevel-1.
//
ShowText(pVideo, "VERTICAL BARS AT SIXTEEN GRAY SHADES", PanelGrayLevel-1);
}
```

```
//-------------------------------------------------------------------------
//
// FUNCTION: ShowVerticalBars()
//
// DESCRIPTION: Displays a series of vertical bars, each with a
//              different gray shade.  For 4 gray levels, each
//              vertical bar is 40 pixels wide.  For 16 gray levels,
//              each vertical bar is 20 pixels wide.
//
// INPUTS: Video address which points to beginning of vertical bars.
//         This address must be at the leftmost column of the display.
//
// RETURN VALUE: None.
//
//-------------------------------------------------------------------------

void ShowVerticalBars(unsigned char _far *pVideo)
{
static unsigned int y;
static unsigned int Bar, BarWidth, val;
static unsigned char _far *pVideoStart;

//
// To display vertical bars, this routine assumes that pVideo points
// to the beginning of a scan line.  In addition, this routine assumes that
// the Address Pitch Adjustment Register is 0 (no virtual display).
// To write one vertical line, first write one pixel to the first byte
// pointed to by pVideo.  Write the next pixel to the byte on the next scan
// line pointed to by pVideo+BytesPerScanLine (this only works if the
// Address Pitch Adjustment Register is 0).  Continue writing pixels by
// going down each scan line.
//
pVideoStart = pVideo;

for (y = 0; y < PanelY; ++y)
   {
   for (Bar = 0; Bar < PanelGrayLevel; ++Bar)
      {
      for (BarWidth = 0; BarWidth < 10; ++BarWidth)
         {
         if (PanelGrayLevel == 4)
            {
            //
            // In the 4 gray level mode, each pixel is stored as two bits.
            // Since a byte holds 8 bits, there are 4 pixels per byte.
            // The variable "val" represents the pixel value.
            //
            val = Bar % 4;
            *pVideo++ = (unsigned char) ((val << 6) | (val << 4) |
                                         (val << 2) | val);
            }
         else
            {
```

```
            //
            // In the 16 gray level mode, each pixel is stored as four bits.
            // Since a byte holds 8 bits, there are 2 pixels per byte.
            // The variable "val" represents the pixel value.
            //
            val = Bar % 16;
            *pVideo++ = (unsigned char) ((val << 4) | val);
            }
         }
      }

   //
   // Point to the beginning of the next scan line
   //
   pVideoStart += BytesPerScanLine;
   pVideo = pVideoStart;
   }
}

//-------------------------------------------------------------------------
//
// ShowText()
//
// DESCRIPTION: Writes text to the LCD panel.  Text must only contain
//              the letters A-Z, and the space character.  All other
//              characters are replaced by spaces.
//
// NOTES: It is assumed that a pixel set to a value of 0 represents the
//        background color (black).
//        The character "!" is translated to a block character.
//
//-------------------------------------------------------------------------

void ShowText(unsigned char _far *pdisplayStart, char *str, int color)
{
static const unsigned char *pFont;
static unsigned char _far *pdisplayFirstColumn;
static unsigned char _far *pDisplay;
static unsigned char ch;
static unsigned int y, val, Display;

//
// Each letter in the font is 8 x 8 bits
//
static const unsigned char font[28][8] =
   { { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },   // blank
     { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF },   // block char
     { 0x30, 0x78, 0xCC, 0xCC, 0xFC, 0xCC, 0xCC, 0x00 },   // A
     { 0xFC, 0x66, 0x66, 0x7C, 0x66, 0x66, 0xFC, 0x00 },   // B
     { 0x3C, 0x66, 0xC0, 0xC0, 0xC0, 0x66, 0x3C, 0x00 },   // C
```

```
     { 0xF8, 0x6C, 0x66, 0x66, 0x66, 0x6C, 0xF8, 0x00 },   // D
     { 0xFE, 0x62, 0x68, 0x78, 0x68, 0x62, 0xFE, 0x00 },   // E
     { 0xFE, 0x62, 0x68, 0x78, 0x68, 0x60, 0xF0, 0x00 },   // F
     { 0x3C, 0x66, 0xC0, 0xC0, 0xCE, 0x66, 0x3E, 0x00 },   // G
     { 0xCC, 0xCC, 0xCC, 0xFC, 0xCC, 0xCC, 0xCC, 0x00 },   // H
     { 0x78, 0x30, 0x30, 0x30, 0x30, 0x30, 0x78, 0x00 },   // I
     { 0x1E, 0x0C, 0x0C, 0x0C, 0xCC, 0xCC, 0x78, 0x00 },   // J
     { 0xE6, 0x66, 0x6C, 0x78, 0x6C, 0x66, 0xE6, 0x00 },   // K
     { 0xF0, 0x60, 0x60, 0x60, 0x62, 0x66, 0xFE, 0x00 },   // L
     { 0xC6, 0xEE, 0xFE, 0xFE, 0xD6, 0xC6, 0xC6, 0x00 },   // M
     { 0xC6, 0xE6, 0xF6, 0xDE, 0xCE, 0xC6, 0xC6, 0x00 },   // N
     { 0x38, 0x6C, 0xC6, 0xC6, 0xC6, 0x6C, 0x38, 0x00 },   // O
     { 0xFC, 0x66, 0x66, 0x7C, 0x60, 0x60, 0xF0, 0x00 },   // P
     { 0x78, 0xCC, 0xCC, 0xCC, 0xDC, 0x78, 0x1C, 0x00 },   // Q
     { 0xFC, 0x66, 0x66, 0x7C, 0x6C, 0x66, 0xE6, 0x00 },   // R
     { 0x78, 0xCC, 0xE0, 0x70, 0x1C, 0xCC, 0x78, 0x00 },   // S
     { 0xFC, 0xB4, 0x30, 0x30, 0x30, 0x30, 0x78, 0x00 },   // T
     { 0xCC, 0xCC, 0xCC, 0xCC, 0xCC, 0xCC, 0xFC, 0x00 },   // U
     { 0xCC, 0xCC, 0xCC, 0xCC, 0xCC, 0x78, 0x30, 0x00 },   // V
     { 0xC6, 0xC6, 0xC6, 0xD6, 0xFE, 0xEE, 0xC6, 0x00 },   // W
     { 0xC6, 0xC6, 0x6C, 0x38, 0x38, 0x6C, 0xC6, 0x00 },   // X
     { 0xCC, 0xCC, 0xCC, 0x78, 0x30, 0x30, 0x78, 0x00 },   // Y
     { 0xFE, 0xC6, 0x8C, 0x18, 0x32, 0x66, 0xFE, 0x00 } }; // Z

pdisplayFirstColumn = pdisplayStart;
pDisplay = pdisplayFirstColumn;
//
// If there are 4 gray levels, there are 4 pixels/byte
//
if (PanelGrayLevel == 4)
   {
   color &= 0x03;
   while (*str != 0)
      {
      ch = *str++;
      if (ch == '!')
         pFont = &font[1][0];   // "Block" character
      else if ((ch < 'A') || (ch > 'Z'))
         pFont = &font[0][0];   // blank character
      else
         pFont = &font[ch - 'A' + 2][0];

      for (y = 0; y < 8; ++y)
         {
         pDisplay = pdisplayFirstColumn;
         val = *pFont++;
         //
         // Since there are 4 gray shades, each bit in the font will be
         // represented in display memory as a two bit gray shade.
```

```
         //
         if (val & 0x80)
            Display = color << 6;
         else
            Display = 0;
         if (val & 0x40)
            Display |= (color << 4);
         if (val & 0x20)
            Display |= (color << 2);
         if (val & 0x10)
            Display |= color;
         *pDisplay++ = (unsigned char) Display;
         if (val & 0x08)
            Display = color << 6;
         else
            Display = 0;
         if (val & 0x04)
            Display |= (color << 4);
         if (val & 0x02)
            Display |= (color << 2);
         if (val & 0x01)
            Display |= color;
         *pDisplay++ = (unsigned char) Display;
         pdisplayFirstColumn += BytesPerScanLine;
         }
      pdisplayStart += 2;   // Point to next character
      pdisplayFirstColumn = pdisplayStart;
      }
   }
else  // 16 Gray Shades
   {
   color &= 0x0f;
   while (*str != 0)
      {
      ch = *str++;
      if (ch == '!')  // "Block" character
         pFont = &font[1][0];
      else if ((ch < 'A') || (ch > 'Z'))
         pFont = &font[0][0];
      else
         pFont = &font[ch - 'A' + 2][0];

      for (y = 0; y < 8; ++y)
         {
         pDisplay = pdisplayFirstColumn;
         val = *pFont++;
         //
         // Since there are 16 gray shades, each bit in the font will be
         // represented in display memory as a four bit gray shade.
```

```
         //
         if (val & 0x80)
            Display = color << 4;
         else
            Display = 0;
         if (val & 0x40)
            Display |= color;
         *pDisplay++ = (unsigned char) Display;
         if (val & 0x20)
            Display = color << 4;
         else
            Display = 0;
         if (val & 0x10)
            Display |= color;
         *pDisplay++ = (unsigned char) Display;
         if (val & 0x08)
            Display = color << 4;
         else
            Display = 0;
         if (val & 0x04)
            Display |= color;
         *pDisplay++ = (unsigned char) Display;
         if (val & 0x02)
            Display = color << 4;
         else
            Display = 0;
         if (val & 0x01)
            Display |= color;
         *pDisplay++ = (unsigned char) Display;
         pdisplayFirstColumn += BytesPerScanLine;
         }
      pdisplayStart += 4;  // Point to next character
      pdisplayFirstColumn = pdisplayStart;
      }
   }
}

//-------------------------------------------------------------------------
//
// FUNCTION: SplitScreen()
//
// DESCRIPTION: Show split screen.
//
// INPUTS: None.
// RETURN VALUE: None.
//
//-------------------------------------------------------------------------

void SplitScreen(void)
{
```

```
static unsigned char _far *pVideoImage1;
static unsigned char _far *pVideoImage2;
static unsigned long ImageSize;
static unsigned int OriginalLineCount;
static unsigned int val;
static unsigned int MinLineCount;
static unsigned int MaxVirtualScanLines;

Initialize();
ClearLCDScreen();


//
// For 64k only
//
FP_SEG(pVideoImage1) = 0xd000;
FP_OFF(pVideoImage1) = 0x0000;


//
// Calculate starting video memory location for image 2 by finding the
// last location of image 1
//
ImageSize = BytesPerScanLine * PanelY;


//
// Because the image size is limited to a maximum of 320 x 240, and there
// is 64k of video memory, there is enough memory available.
//
FP_SEG(pVideoImage2) = 0xd000;
FP_OFF(pVideoImage2) = (unsigned int) ImageSize;


ShowVerticalBars(pVideoImage1);
ShowHorizontalBars(pVideoImage2);

//
// Show text.  The lightest gray shade is set to PanelGrayLevel-1.
//
ShowText(pVideoImage1, "SPLIT SCREEN IMAGE ONE", PanelGrayLevel-1);
ShowText(pVideoImage2, "SPLIT SCREEN IMAGE TWO", PanelGrayLevel-1);


//
// Set Screen 2 Display Start Address register to point to Image 2
//
// Adjust ImageSize to represent the size in words, not bytes.
// This is because the Memory Interface is set to 16 bits.
//
val = (unsigned int) ImageSize / 2;
val += 0x8000;   // Point to D000 segment instead of C000 segment
```

```
WriteRegister(8, (unsigned int) val & 0xff);
WriteRegister(9, (unsigned int) val >> 8);


//
// If this is a dual panel, then the split screen has just been shown.
// Otherwise, set up the Screen 1 Display Line Count register for single
// panels.
//
if (PanelType == TYPE_SINGLE)
   {
   OriginalLineCount =
      (unsigned int) ((ReadRegister(0x0b) & 0x03) << 8) | ReadRegister(0x0a);


   // Only for 64k of memory
   MaxVirtualScanLines = (unsigned int)
      ((unsigned long) 0x10000 / BytesPerScanLine);

   MinLineCount = OriginalLineCount -
                  (MaxVirtualScanLines - OriginalLineCount) + 1;

   Delay(0, 5);

   //
   // Scroll image 2 down
   //
   for (val = MinLineCount; val < OriginalLineCount; val += 1)
      {
      WriteRegister(0x0a, val & 0xff);         // Total Display Line Count
      WriteRegister(0x0b, (val >> 8) & 0x03);  // Total Disp Line Cnt/WF Count

      Delay(0, 1);
      }

   //
   // Scroll image 2 up
   //
   for (val = OriginalLineCount; val > MinLineCount; val -= 1)
      {
      WriteRegister(0x0a, val & 0xff);         // Total Display Line Count
      WriteRegister(0x0b, (val >> 8) & 0x03);  // Total Disp Line Cnt/WF Count

      Delay(0, 1);
      }

   val = MinLineCount;
   WriteRegister(0x0a, val & 0xff);            // Total Display Line Count Reg
   WriteRegister(0x0b, (val >> 8) & 0x03);     // Total Disp Line Cnt/WF Count

   Delay(0, 5);
   }
}
```

```
void SetStartAddress(int x, int y)
{
int addr;

//
// Assume 16 gray shades
//
addr = 0x8000 + (x/2 + (VIRTUAL_X/2) * y)/2;

WriteRegister(6, addr & 0xff);
WriteRegister(7, addr >> 8);
}
```

```
void PanScroll(void)
{
static unsigned int x, y;
static unsigned int MaxX, MaxY;
static unsigned int val, pitch;
static unsigned char _far *pVideo;


printf("Showing Panning and Scrolling\n");

Initialize();
ClearLCDScreen();

//
// This pitch is calculated for 16 gray shades
//
pitch = ((VIRTUAL_X / 2) - BytesPerScanLine) / 2;
WriteRegister(0x0d, pitch);
BytesPerScanLine = (VIRTUAL_X / 2);

//
// For 64k only
//
FP_SEG(pVideo) = 0xd000;
FP_OFF(pVideo) = 0x0000;


//
// Display random blocks of data.  To do so, a text character will be used.
// This character sets all pixels in a character region, so a block is
// shown at the specified gray shade.
//

// Seed the random number generator with current time
srand((unsigned) time(NULL));
```

```
for (x = 0; x < 300; ++x)
   {
   FP_OFF(pVideo) = (unsigned int) ((rand() * 0xffffL) / RAND_MAX);
   val = rand() % 50;

   ShowText(pVideo, "!", rand() % 16);
   }

ShowBorders();

//
// Move virtual display from (0, 0) to (MaxX, 0)
//
MaxX = VIRTUAL_X - PanelX;
MaxY = VIRTUAL_Y - PanelY;

for (x = 0; x <= MaxX; ++x)
   {
   SetStartAddress(x, 0);
   Delay(0, 1);
   }

for (y = 0; y <= MaxY; ++y)
   {
   SetStartAddress(MaxX, y);
   Delay(0, 1);
   }

for (x = MaxX; x > 0; --x)
   {
   SetStartAddress(x, MaxY);
   Delay(0, 1);
   }

for (y = MaxY; y > 0; --y)
   {
   SetStartAddress(0, y);
   Delay(0, 1);
   }

SetStartAddress(0, 0);
}

//-----------------------------------------------------------------------------
//
// FUNCTION: PowerSaving()
//
// DESCRIPTION: Starts power saving mode 2.
//
// INPUTS: None.
// RETURN VALUE: None.
//
```

```
//-------------------------------------------------------------------------

void PowerSaving(void)
{
static unsigned int val;

printf("Starting Power Saving\n");

val = ReadRegister(3);
val &= 0x3f;
val |= 0x80;
WriteRegister(3, val);  // Set power saving mode 2

printf("Press any key to cancel power saving\n");
getch();

val &= 0x38;
WriteRegister(3, val);  // Cancel power saving mode 2
}

//-------------------------------------------------------------------------
//
// FUNCTION: PowerSaving()
//
// DESCRIPTION: Starts power saving mode 2.
//
// INPUTS: None.
// RETURN VALUE: None.
//
//This is an optional method of power saving.
//
//-------------------------------------------------------------------------
void PowerSaving(void)
{
static unsigned int val;
printf("Starting Power Saving\n");
//
// The following are the steps to enter a power save mode.
//
//
// Step 1: Turn off display
//
val = ReadRegister(1);
val &= 0x7f;
WriteRegister(1, val);
//
// Step 2: Disable LCDE (turn off LCD power supply).
//         For the SDU1353B0C, set LCDE bit to 0.
//
val = ReadRegister(1);
val &= 0xef;
WriteRegister(1, val);
//
```

```
// Step 2: Wait for LCD power supply to drop to zero volts
//         For the SDU1353B0C, wait about a half second.
//
Delay(500);

//
// Step 3: Enter Power Save Mode
//
val = ReadRegister(3);
val &= 0x3f;
val |= 0x80;
WriteRegister(3, val);  // Set power saving mode 2

printf("Press any key to cancel power saving\n");
getch();

//
// The following are the steps to exit a power save mode.
//

//
// Step 1: Exit Power Save Mode
//
val = ReadRegister(3);
val &= 0x3f;

WriteRegister(3, val);  // Cancel power saving mode 2
//
// Step 2: Enable LCDE (turn on LCD power supply).
//         For the SDU1353B0C, set LCDE bit to 1.
//
val = ReadRegister(1);
val |= 0x10;
WriteRegister(1, val);

//
// Step 3: Turn on display.
//
val = ReadRegister(1);
val |= 0x80;
WriteRegister(1, val);
ShowMenu();
}
```

7 GLOSSARY

1352	The SED1352 LCD controller chip.
display memory	Memory in which an image is stored for display by the SED1352.
gray shade	A specific combination of white and black colors. For example, a lighter gray shade has more white than black.
LCD	Liquid Crystal Display. The display device used by the SED1352.
LCD controller	The device used to control the LCD display. The SED1352 is an LCD controller.
LUT	Look-Up Table, or palette. The LUT treats the value of a pixel as an index into an array of gray shades.
panel	The circuitry and viewable area of an LCD display which supports a single image. LCD displays may have one or two panels.
panning	The right or left movement of the viewport in a virtual display.
pixel	Picture Element. A pixel is seen as a dot on the display, and can be shown using one of several different gray shades. Combining pixels in a group creates an image.
power saving	A means of reducing the power consumption of the SED1352.
register	A memory storage location to control a peripheral, such as the SED1352.
scrolling	The up and down movement of the viewport in a virtual display.
SED1352	The 1352 chip.
SDU1352B0x	The evaluation board for the SED1352. The SDU1352B0x is an ISA board for a PC-compatible computer.
viewport	The visible portion of a virtual display.
virtual display	An image stored in display memory that is larger than what the LCD display can show. A virtual display supports panning and scrolling.

THIS PAGE LEFT BLANK

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

1352SHOW.EXE Display Utility

Document Number: X16-UI-001-08

Copyright © 1995, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

1352SHOW.EXE DISPLAY UTILITY

1352SHOW is an OEM demonstration utility used to load and display GIF images. It can also be used to demonstrate the split screen capabilities of the SED1352 by loading two images and vertically scrolling one image.

Program Requirements

Video Controller	**:**	SED1352
Display Type	**:**	Up to 640x480 LCD
BIOS	**:**	Seiko Epson BIOS1352 version 1.11 or later
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes
Windows DOS Full Screen	**:**	Yes
OS/2 DOS Full Screen	**:**	Yes

Installation

Copy the file **1352show.exe** to a directory that is in the DOS path on your hard drive.

Usage

1352SHOW is invoked from the DOS command line as follows.

```
1352show [image1] [image2] [/i] [/?]
```

Where**:**

`image1`	is the first screen image to be displayed.
`image2`	is the second screen image to be displayed.
`/i`	will invert all displayed images (show as negative).
`/?`	produces the usage message.

Examples:

`1352show`	with no arguments will run the program in split screen mode. This will display two predefined images, with screen one displaying horizontal bars and screen two displaying vertical bars. Screen two may be scrolled up and down using the arrow, page up, page down, home and end keys.
`1352show picture1.gif`	displays the named GIF image.
`1352show dog.gif cat.gif`	displays the two named GIF images in a split screen. Screen two may be scrolled up and down using the arrow, page up, page down, home and end keys.

Pressing the ESC key will terminate the program.

Comments

- 1352SHOW requires BIOS1352.COM to be loaded prior to running.
- Split screen viewing is only allowed on single panels.
- The size of screen two is determined by available memory and number of gray shades. If there is insufficient memory for screen two 1352SHOW will not accept the two image files and will generate an error message.
- When loading two GIF images, it may take several seconds of apparent inactivity to load the second image into memory.
- The GIF format must be 16 color, non-interlaced GIF89a format.
- 1352SHOW will clear the screen when the Esc key is pressed.

Program Messages

`ERROR: Split screen available for single panel only.`

Split screen viewing is only allowed on single panels.

`ERROR: This program requires BIOS1352 to be loaded!`

The program BIOS1352.COM must be run before 1352SHOW. Load BIOS1352.COM and re-run 1352SHOW.EXE.

`File "`*`filename`*`" not found or cannot be opened for reading.`

The GIF file you are trying to display is not in your DOS path or not on your system.

`File is not GIF89a format.`

The GIF file contains an invalid format. 1352SHOW only supports GIF89a format.

`Insufficient video memory for second image.`

There is not enough video memory available to store both images.

`Invalid format in the GIF file.`

Use non-interlaced GIF89a format.

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

VIRTUAL.EXE Display Utility

Document Number: X16-UI-002-08

Copyright © 1995, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

VIRTUAL.EXE DISPLAY UTILITY

VIRTUAL.EXE demonstrates the virtual panning capabilities of the SED1352. An image larger than the display resolution is loaded in display memory. VIRTUAL.EXE will then display a portion of the complete image while providing panning capabilities using the arrow keys for navigation.

Program Requirements

Video Controller	**:**	SED1352
Display Type	**:**	Up to 640x480 LCD
BIOS	**:**	Seiko Epson BIOS1352 version 1.11 or later
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes
Windows DOS Full Screen	**:**	Yes
OS/2 DOS Full Screen	**:**	Yes

Installation

Copy the file **virtual.exe** to a directory that is in the DOS path on your hard drive.

Usage

VIRTUAL is invoked from the DOS command line as follows.

```
virtual [x=n] [y=n] [/?]
```

Where**:**	x	is the horizontal resolution (in multiples of 8).
	y	is the vertical resolution.
	/?	produces a usage message.

If the user does not provide the virtual size, the program will automatically select the size based on memory and panel size. The user can then navigate throughout the image using the arrow keys to pan and scroll the screen. Pressing the ESC key terminates the program.

Comments

- VIRTUAL requires BIOS1352.COM to be loaded prior to running.
- VIRTUAL forces four gray shade mode regardless of original BIOS1352 settings. The original BIOS1352 settings are restored on exiting VIRTUAL.

Program Messages

ERROR: This program requires BIOS1352 to be loaded!

The program BIOS1352.COM must be run before VIRTUAL.EXE. Load BIOS1352.COM and then re-run VIRTUAL.EXE.

ERROR: Insufficient memory for virtual display.

The virtual display is too large to fit in memory. Choose a smaller x or y value.

ERROR: Horizontal resolution must be a multiple of 8.

Panning moves in multiples of pixels. Choose a horizontal resolution which is ≥ a multiple of 8, so panning will not suffer from screen wrap-around.

ERROR: Specified horizontal resolution is smaller than panel resolution.

The virtual display must always be larger than the panel size.

ERROR: Specified vertical resolution is smaller than panel resolution.

The virtual display must always be larger than the panel size.

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

BIOS1352.COM Utility

Document Number: X16-UI-003-08

Copyright © 1995, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

BIOS1352.COM UTILITY

BIOS1352 is a DOS Terminate and Stay Resident (TSR) program which replaces and/or supplements the PC video interrupt INT 10h. This program provides text, scroll, and cursor functionality when no VGA BIOS is present. Although the SED1352 is not a VGA or EGA compatible controller, this program is supplied to give the user a familiar prompt. Within limits BIOS1352 simulates a VGA BIOS and will allow standard output functions to work. DOS programs such as Edlin, Format, Debug, and internal commands such as Copy, Ren, Mkdir, etc., should work; however, complex programs such as Edit, Qbasic, and Scandisk will not work. The standard output functions are handled by the VGA BIOS, if present.

Program Requirements

Video Controller	:	SED1352
Display Type	:	Up to 640x480 LCD
BIOS	:	None or any VGA
DOS Program	:	Yes
DOS Version	:	3.0 or greater
Windows Program	:	No
Windows DOS Box	:	Yes
Windows DOS Full Screen	:	Yes
OS/2 DOS Full Screen	:	Yes

Installation

Copy the file **bios1352.com** to a directory that is in the DOS path on your hard drive.

Usage

BIOS1352.COM is run from the DOS command line as follows:

```
bios1352 type x=n y=n g=n p=n m=n [/?]
```

Where:	`type`	is the panel type: **single** for single panel or **dual** for dual panel
	`x`	is the horizontal panel size in pixels (decimal)
	`y`	is the vertical panel size in lines (decimal)
	`g`	is the number of gray shades: 4 or 16
	`p`	is the port address in hex: 300|310...360|370
	`m`	is the memory size in K bytes: 64 or 128
	/?	produces a usage message

The order and case of arguments is arbitrary. Any invalid or missing argument will result in an error message. Note that the port address must be the same as the physical address set on the SDU1352 evaluation board.

Example:

```
BIOS1352 SINGLE x=320 y=240 g=16 p=320 m=128
```

Comments

- BIOS1352 can be used in conjunction with a Monochrome Display Adapter (mono) card. The standard DOS command MODE MONO will switch to the mono card and the DOS command MODE CO80 will switch to the LCD panel.
- BIOS1352 emulates mode 3, but any program that attempts to write directly to video memory, bypassing the video BIOS, will not display correctly.
- BIOS1352 can be used in conjunction with a VGA BIOS. In this case all TTY output will be displayed on the VGA monitor.
- When the SED1352 video memory is specified as 64K bytes, the SED1352 video memory will reside at D000h to DFFFh. For 128K bytes of SED1352 video memory, the memory will reside at C000h to DFFFh.

Program Messages

`ERROR: Panels greater than 640 pixels not supported.`

More than 640 horizontal pixels has been specified for the panel in the command line.

`ERROR: Panels greater than 480 lines not supported.`

More than 480 vertical lines has been specified for the panel in the command line.

`ERROR: Invalid port specified.`

The port address (**`p`**) must be specified in the format 3x0 in the command line. The range is 300h to 370h in 10h increments.

`ERROR: Only 4 or 16 gray shades allowed.`

A number other than 4 or 16 has been specified for the variable **`g`** in the command line.

`ERROR: Not enough video memory for the panel.`

The panel specified is too large to run in 16 gray shades mode. Select 4 gray shades instead.

`ERROR: Video memory and VGA BIOS memory conflict.`

Both the SED1352 video memory and the VGA BIOS are trying to use the memory at location C000h to CFFFh.

`ERROR: Only 8k, 16k, 32k, 40k, 64k or 128k memory allowed.`

An invalid value has been specified for memory size (**`m`**) on the command line.

`ERROR: Only 8 or 16 bits allowed for width.`

The SED1352 only supports 8 or 16 bit memory width.

`ERROR: Memory size cannot support memory width.`

Choose one of the following combinations:

Memory Size (m)	8	16	32	40	64	128
Memory Width (w)	8	16	8	8	16	16

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

1352GRAY.EXE Display Utility

Document Number: X16-UI-004-08

Copyright © 1995, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

1352GRAY.EXE DISPLAY UTILITY

1352GRAY is a menu driven display utility for the SED1352 which demonstrates the gray shades and available palettes. For 128K bytes of display memory and a panel size of 640x400 or less, either 4 or 16 gray shades are available. If the panel size is greater than 640x400 only 4 shades of gray are available. For 64K bytes of display memory and a panel size of 640x200, 320x240 or less, either 4 or 16 shades are available. In 4 gray shade mode it is possible to select 1 of 4 palettes.

Program Requirements

Video Controller	**:**	SED1352
Display Type	**:**	Up to 640x480 LCD
BIOS	**:**	Seiko Epson BIOS1352 version 1.11 or later
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes
Windows DOS Full Screen	**:**	Yes
OS/2 DOS Full Screen	**:**	Yes

Installation

Copy the file **1352gray.exe** to a directory that is in the DOS path on your hard drive.

Usage

1352GRAY is invoked from the DOS command line as follows.

```
1352gray [/?]
```

Where**:** `/?` produces a usage message.

1352GRAY displays a default gray shade pattern as a series of vertical or horizontal bars. The pattern, number of gray shades, and current palette may be modified by the user when possible. Instructions to modify these options will appear when available.

Pressing the ESC key terminates the program and restores the original BIOS1352 settings.

Comments

- 1352GRAY requires BIOS1352.COM to be loaded prior to running.
- Four gray shades is always possible. Switching to 16 gray shades may not be possible if the panel size exceeds 640x400.

Program Messages

`ERROR: This program requires BIOS1352 to be loaded!`

The program BIOS1352.COM must be run before 1352GRAY. Load BIOS1352.COM and then re-run 1352GRAY.EXE.

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

1352PD.EXE Power Down Utility

Document Number: X16-UI-005-07

Copyright © 1996, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

1352PD.EXE POWER DOWN UTILITY

1352PD is an OEM utility program for setting power down modes in the SED1352 LCD Display Controller. It provides a simple method for setting power modes during power consumption testing.

Program Requirements

Video Controller	:	SED1352
Display Type	:	Up to 640x480 LCD
BIOS	:	Seiko Epson BIOS1352 version 1.11 or later
DOS Program	:	Yes
DOS Version	:	3.0 or greater
Windows Program	:	No
Windows DOS Box	:	Yes
Windows DOS Full Screen	:	Yes
OS/2 DOS Full Screen	:	Yes

Installation

Copy the file **1352pd.exe** to a directory that is in the DOS path on your hard drive.

Usage

1352PD is run from the DOS command line as follows:

```
1352pd ModeNumber
```

Where: `ModeNumber` is a decimal number (0, 1, or 2) for the desired power down mode.

Example: typing the following command line activates power down mode 2:

```
1352pd 2 <ENTER>
```

Output from the program can be redirected to an external DOS device such as a terminal attached to the serial port such as COM1 as shown below:

```
1352pd 2 > com1 <ENTER>
```

Striking any key will set mode state 0 (no power down).

Comments

- 1352PD.EXE requires BIOS1352.COM to be loaded prior to running.
- The following power modes are supported:

Mode 0	Mode 0 operates at full power.
Mode 1 or 2	SED1352 will engage power down mode 1 or 2. SED1352 LUT will be disabled and all LCD signals are forced low.

Program Messages

`Power Down Mode xx is set.`

The power down mode xx has been set. This message may not be visible if the active display controller is the SED1352.

`ERROR: Cannot set power mode xx!`

1352PD.EXE cannot set the power down mode requested . The power down mode must be 0, 1, or 2.

`ERROR: This program requires BIOS1352 to be loaded!`

The program BIOS1352.COM must be run before 1352PD. Load BIOS1352 and re-run 1352PD.EXE.

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

1352READ.EXE Diagnostic Utility

Document Number: X16-UI-006-06

Copyright © 1996, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

1352READ.EXE DIAGNOSTIC UTILITY

1352READ is an OEM utility program which enables the user to read the SED1352 register contents. It is a useful utility for OEMs wishing to submit a problem report for the video controller. If run with BIOS1352 loaded, it will try to interpret the BIOS settings.

Program Requirements

Video Controller	**:**	SED1352
Display Type	**:**	Up to 640x480 LCD
BIOS	**:**	Seiko Epson BIOS1352.COM (optional)
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes
Windows DOS Full Screen	**:**	Yes
OS/2 DOS Full Screen	**:**	Yes

Note
1352READ uses “stdout” calls and may be redirected to a file or piped to a DOS filter such as MORE.COM.

Installation

Copy the file **1352read.exe** to a directory that is in the DOS path on your hard drive.

Usage

From DOS prompt, type the following:

```
1352read [port] [/?]
```

Where:

`1352read`	without any argument will read the SED1352 registers, including the gray shade lookup table.
`port`	is the SED1352 port address in hex (e.g., 310).
`/?`	produces a usage message.

Example: to generate a report, simply type

```
1352read [port] > report.txt
```

and the information which 1352READ obtains will be stored in the file report.txt.

Comments

- It is not necessary to specfy a port address if BIOS1352 has previously been loaded.
- 1352READ will search for BIOS1352.COM. If this program is found the port address reported by BIOS1352 will be used. If the port address is specified on the 1352READ command line the two port addresses are compared and if different an error message is generated.
- 1352READ will accept any port address, however, the SDU1352 can only be configured to an address in the range of 300h through 370h.

Program Messages

`ERROR: 1352 registers not responding at port address [port].`

1352READ has not found an SED1352 at the port address specified. Check the command line port setting for BIOS1352 and/or 1352READ to ensure it is correct and re-run the program.

`ERROR: 1352READ requires a port address.`

1352READ has not detected BIOS1352.COM to obtain the port address and no port address was specified on the command line. Either specify a port address on the 1352READ command line or run BIOS1352.COM prior to running 1352READ.

`ERROR: BIOS1352 reports a port address of [port], which is different from the specified port address of [port].`

The poert address entered for 1352READ is different that the one entered for BIOS1352.COM. Specify the same port address on the 1352READ command line as the one in BIOS1352.COM and the physical address of the SDU1352 evaluation board and re-run the program.

`WARNING: BIOS1352 state is out of sync with SED1352 registers.`

One or more of the following command line items reported by BIOS1352 does not match the values found in the SED1352 registers; horizontal panel size, vertical panel size, number of gray shades, or panel type (single or dual).

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

SDU1352B0C Rev. 1.0 Evaluation Board User Manual

Document Number: X16-AN-002-09

Copyright © 1995, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 SDU1352B0C REV 1.0 EVALUATION BOARD

This manual reflects the use of the SDU1352B0C Rev 1.0 evaluation board in conjunction with the SED1352 LCD Controller. All appropriate components are surface-mount to reduce cost and minimize board space.

1.1 Features

- 100 pin QFP5 package
- SMD technology for all appropriate devices
- Monochrome STN LCD support
- 8-bit and 16-bit ISA Bus support
- 5V operation
- Two terminal crystal support (up to 25.175MHz)
- 16-bit wide, 128K bytes SRAM support
- Configuration Options
- Support for Software Power Save Modes
- On-board adjustable LCD BIAS negative power supply
- On-board adjustable LCD BIAS positive power supply
- CPU/Bus Interface Header strips

1.2 Installation and Configuration

The SED1352 has 16 configuration inputs (VD[15:0]) which are read on power-up. For the purpose of this design, most of these configuration inputs have been factory set and therefore are not configurable. A four position DIP switch block is provided for the selection of 8- or 16-bit bus interface, and setting I/O address bits 4 through 6.

Table 1-1: Configuration DIP Switch Settings

Switch	Signal	Closed	Open
SW1-1	VD0	16-bit ISA Bus interface	8-bit ISA Bus interface
SW1-2	VD1	Direct-mapping I/O	Indexing I/O
SW1-3	VD2	M68K CPU Interface	ISA Bus / other MPU other
SW1-4	VD3	Byte-swap high and low data bytes	No byte-swap
SW1-5	VD7	I/O mapping address bit 4	See Table 1-2, “I/O Mapping Example”
SW1-6	VD8	I/O mapping address bit 5	
SW1-7	VD9	I/O mapping address bit 6	
SW1-8	-	64K bytes of SRAM available at segment D000h	128K bytes of SRAM available at segment C000h-D000h

Note
The polarity of the Configuration Dip Switches is Closed = ’1’ or ’high’, Open = ’0’ or ’low’.

Factory set fixed options on this board are:

- 16-bit display memory interface (either 64K bytes or 128K bytes)
- 128K bytes available at C000h memory segment

This board is also pre-set to use indexing I/O with address 0000 0011 0??? 000x, where x is don’t care and ??? can be configured with dip-switch SW1-5 through SW1-7. The factory setting of ??? = 001, i.e., I/O address = 0310h and 0311h. When using direct-mapping I/O, the I/O address is 0000 0011 0??? xxxx, where x is don’t care and ??? can be configured with dip-switch SW1-5 through SW1-7. If ??? = 001, then the I/O address for AUX[00h] = 0310h, I/O address for AUX[01h] = 0311h, I/O address for AUX[02h] = 0312h and so on.

Table 1-2: I/O Mapping Example

	bit 6	bit 5	bit 4
I/O Mapping Address (Hex)	0	0	1

Table 1-3: Decoding Jumper Setting

	Description	1-2	2-3
JP1	Set to the same polarity as SW1-1 (VD0)	1	0
JP2	Set to the same polarity as SW1-5 (VD7)	1	0
JP3	Set to the same polarity as SW1-6 (VD8)	1	0
JP4	Set to the same polarity as SW1-7 (VD9)	1	0

Note
These jumpers are necessary for the external ISA Bus decode logic.

LCD Signal Connector Pinout

Table 1-4: LCD Signal Connector J1 Pinout

SED1352 Pin Name	LCD Connector Pin No.	Mono STN LCD		Comments
		8-bit	4-bit	
LD0	1	LD0		Lower panel display data for dual panel-dual drive mode. In 8-bit single panel-single drive mode, these are the least significant 4 bits of the 8-bit output data to the panel (data[3:0]). In 4-bit single panel mode, these outputs are low.
LD1	3	LD1		
LD2	5	LD2		
LD3	7	LD3		
UD0	9	UD0	UD0	Upper panel display data for dual panel-dual drive mode. In 8-bit single panel-single drive mode, these are the most significant 4 bits of the 8-bit output data to the panel (data[7:4]). In 4-bit single panel mode, these are the 4 data bits output to the panel.
UD1	11	UD1	UD1	
UD2	13	UD2	UD2	
UD3	15	UD3	UD3	
N/C	17-31 (odd pins)			
XSCL	33	XSCL	XSCL	Shift Clock for LCD data
NC	35			
LP	37	LP	LP	Latch Pulse output
YD	39	YD	YD	Vertical Scanning Start Pulse
GRND	2-26 (even pins)	GRND	GRND	Logic Ground
N/C	28			
VLCD	30	VLCD	VLCD	Negative power supply output (-18V to -23V)
VCC	32	+5V	+5V	
+12V	34	+12V	+12V	
VDDH	36	VDDH	VDDH	Positive power supply output (+23V to +40V)
WF	38	WF	WF	LCD backplane Bias signal
LCDENB	40	/LCDPWR	/LCDPWR	LCD power control to external supply

CPU / BUS Interface Connector Pinouts

Table 1-5: CPU/BUS Connector H1 Pinout

Connector Pin No.	CPU/BUS Pin Name	Comments
1	SD0	Connected to DB0 of the SED1352
2	SD1	Connected to DB1 of the SED1352
3	SD2	Connected to DB2 of the SED1352
4	SD3	Connected to DB3 of the SED1352
5	GND	Ground
6	GND	Ground
7	SD4	Connected to DB4 of the SED1352
8	SD5	Connected to DB5 of the SED1352
9	SD6	Connected to DB6 of the SED1352
10	SD7	Connected to DB7 of the SED1352
11	GND	Ground
12	GND	Ground
13	SD8	Connected to DB8 of the SED1352
14	SD9	Connected to DB9 of the SED1352
15	SD10	Connected to DB10 of the SED1352
16	SD11	Connected to DB11 of the SED1352
17	GND	Ground
18	GND	Ground
19	SD12	Connected to DB12 of the SED1352
20	SD13	Connected to DB13 of the SED1352
21	SD14	Connected to DB14 of the SED1352
22	SD15	Connected to DB15 of the SED1352
23	RESET	Connected to the RESET signal of the SED1352
24	GND	Ground
25	GND	Ground
26	GND	Ground
27	+12V	12 volt supply
28	+12V	12 volt supply
29	/SBHE	Connected to the BHE# signal of the SED1352
30	IOCHRDY	Connected to the READY signal of the SED1352
31	/IOSC	Connected to the IOCS# signal of the SED1352
32	/MEMCS	Connected to the MEMCS# signal of the SED1352

Table 1-6: CPU/BUS Connector H2 Pinout

Connector Pin No.	CPU/BUS Pin Name	Comments
1	SA0	Connected to AB0 of the SED1352
2	SA1	Connected to AB1 of the SED1352
3	SA2	Connected to AB2 of the SED1352
4	SA3	Connected to AB3 of the SED1352
5	SA4	Connected to AB4 of the SED1352
6	SA5	Connected to AB5 of the SED1352
7	SA6	Connected to AB6 of the SED1352
8	SA7	Connected to AB7 of the SED1352
9	GND	Ground
10	GND	Ground
11	SA8	Connected to AB8 of the SED1352
12	SA9	Connected to AB9 of the SED1352
13	SA10	Connected to AB10 of the SED1352
14	SA11	Connected to AB11 of the SED1352
15	SA12	Connected to AB12 of the SED1352
16	SA13	Connected to AB13 of the SED1352
17	GND	Ground
18	GND	Ground
19	SA14	Connected to AB14 of the SED1352
20	SA15	Connected to AB14 of the SED1352
21	SA16	Connected to AB16 of the SED1352
22	SA17	Connected to AB17 of the SED1352
23	SA18	Connected to AB18 of the SED1352
24	SA19	Connected to AB19 of the SED1352
25	GND	Ground
26	GND	Ground
27	+5V	5 volt supply
28	+5V	5 volt supply
29	/IOW	Connected to the IOW# signal of the SED1352
30	/IOR	Connected to the IOR# signal of the SED1352
31	/SMEMW	Connected to the MEMW# signal of the SED1352
32	/SMEMR	Connected to the MEMR# signal of the SED1352

1.3 Technical Description

1.3.1 ISA Bus Support

This board directly supports the 16-bit and 8-bit ISA Bus with indexing I/O via a standard AT edge connector. External logic has been added to provide signals which the SED1352 does not support directly. See Application Note X16-AN-003-xx.

Note

1. This board has been designed to operate in conjunction with either a VGA card or monochrome card, or as a stand-alone card.

If using the SDU1352B0C in conjunction with a VGA display adapter the following limitations apply:

a. Only 64K bytes of memory is available, residing at the D000h segment.
b. Given the memory limitation certain panel size and gray shade capabilities are reduced.
c. The VGA card video BIOS must be 8-bit only.

The SDU1352B0C must be configured as follows:

SW1-1 open	: 8-bit operation, necessary to prevent MEMCS16# conflict when reading VGA BIOS
SW1-2 to 7	: set as desired
SW1-8 closed	: 64K bytes available at D000h segment
JP1 2-3 shorted	: to reflect SW1-8 polarity

If using the SDU1352B0C in conjunction with a monochrome display adapter all 128K bytes of memory is available residing at segment C000h - D000h.

The SDU1352B0C can be used as a stand-alone video adapter with 128K bytes memory available. If used as a stand-alone video adapter the BIOS setup program for the computer must support and have “No Video” selected as the video adapter. The BIOS1352.COM utility program can be used with the evaluation board to simulate a standard video BIOS, thus providing text and cursor functionality. See the BIOS1352.COM Utility manual, X16-UI-003-xx for details.

2. This board is pre-set to use indexing I/O with address 000 0011 0??? 000x, where x is don’t care and ??? can be configured through dip-switch SW1-7 to SW1-5. The factory setting of ??? = 001, i.e., I/O address = 0310h and 0311h.

3. In indexing I/O, only two I/O address spaces are needed. For example, if I/O address 310h is used, 310h will be the index register and 311h will be the data register.

Example:

```
I/O write 310h 01      :set index = 1
I/O read 311h          :read contents of AUX[01h]
I/O write 310h 05      :set index = 5
I/O write 311h 07      :write 07 to AUX[05h]
```

1.3.2 Non-ISA Bus Support

This evaluation board was specifically designed to support the standard 8-/16-bit ISA bus. However, as the SED1352 does support other bus interfaces, header strips have been provided containing all necessary I/O pins (see section 1.3.9 on page 14).

When using the header strips to provide the bus interface observe the following:

1. All I/O 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.
2. U2, a TIBPAL22V10, is currently used to provide the SED1352 IOCS# (pin 23) and MEMCS# (pin 22) input signals for ISA bus use. This functionality must now be provided externally and these two pins need to disconnected as there may be conflict problems associated with two different outputs driving the same input.

1.3.3 SRAM Support

The SDU1352B0C board supports 16-bit wide, 64K byte - 128K byte SRAM only. DIP switch SW1-8 selects between the two options.

1.3.4 Monochrome LCD Support

The SED1352 supports 4- and 8-bit Dual and Single monochrome STN LCD panels. All the necessary signals are provided on the 40-pin ribbon cable header. The interface signals are alternated with grounds on the cable to reduce cross talk and noise related problems.

Refer to Table 1-4, “LCD Signal Connector J1 Pinout,” on page 9 for specific settings.

1.3.5 Power Save Modes

The SED1352 supports 2 software Power Save Modes. The utility program 1352PD.EXE is supplied to control the software modes. The software modes are controlled by directly writing the SED1352 associated internal registers.

1.3.6 Adjustable LCD Panel Negative Power Supply

The majority of Monochrome LCD panels require a negative power supply to provide between -18 V and -23 V (I_{out}=45mA). For ease of implementation, such a power supply has been provided as an integral part of this design. The signal VLCD can be adjusted by R11 (100K potentiometer) to provide an output voltage from -14 V to -23 V and is enabled/disabled by the control signal LCDENB.

Note
LCDENB is directly controlled by register AUX[01], bit 4, of the SED1352. The VLCD power supply used on the SDU1352 requires a logic “1” to disable it. As the signal LCDENB is a logic “0” at power-up, it is inverted by external logic to disable VLCD and prevent damaging the panel connected to the SDU1352.

Determine the panel’s specific power requirements and set the potentiometer accordingly before connecting the panel.

1.3.7 Adjustable LCD Panel Positive Power Supply

Most single Monochrome 640x480 STN 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 signal VDDH can be adjusted by R8 (100K potentiometer) to provide an output voltage from +23 V to +40 V and is enabled/disabled by the control signal LCDENB.

Note
LCDENB is directly controlled by register AUX[01], bit 4, of the SED1352. The VDDH power supply used on the SDU1352 requires a logic "1" to disable it. As the signal LCDENB is a logic "0" at power-up, it is inverted by external logic to disable VLCD and prevent damaging the panel connected to the SDU1352.

Determine the panel's specific power requirements and set the potentiometer accordingly before connecting the panel.

1.3.8 Crystal Support

The input crystal frequency may be up to 25.175MHz depending on the specific panel size and frame rate desired.

Refer to *Section 9.3 of the SED1352 Functional Specification*, Drawing Office No. X16-SP-001-xx for further details.

1.3.9 CPU/Bus Interface Header Strips

All of the CPU/Bus interface pins of SED1352 are connected to the header strips H1 and H2 for easy interface to a CPU/Bus other than the ISA bus.

Refer to Table 1-5, "CPU/BUS Connector H1 Pinout," on page 10 and Table 1-6, "CPU/BUS Connector H2 Pinout," on page 11 for specific settings.

Note
These headers only provide the CPU/Bus interface signals from SED1352, when MC68K interface is selected (SW1-3 closed), external decoding logic MUST be used to access the SED1352.

1.3.10 Schematic Notes

The evaluation boards may have been modified and therefore the following schematics may not reflect the actual implementation. Please request updated information before starting any hardware design.

Appendix A PARTS LIST

Item #	Qty/ Board	Designation	Part Value	Description
1	33	C1, C8-C10, C14-C42	0.1uF	1206 pckg
2	1	C2	1.0uF/35V	Tantalum .1 spacing radical
3	2	C3, C4	56uF/35V	LXF35VB56RM6X11LL
4	4	C7, C11 - C13	10uF/15V	Tantalum D-SIZE
5	2	H1, H3	Con32A	0.1" 2x16 Male Header (PTH)
6	1	H2	Con36A	0.1" 2x18 Male Header (PTH)
7	19	JP1 - JP19	Header 3	0.1" 1x3 Male Header (PTH)
8	1	J1	Con40A	40 pin strouded header dual-row-center key
9	2	J2, J3	M68340EVSP-64A	Socket strip/wire wray 64 pin #100-064-451
10	1	Q1	2N3905	PNP Signal Transistor (TO-92 PTH)
11	1	Q2	2N3903	NPN Signal Translator (TO-92 PTH)
12	7	R1, R3-R8	1 ohm	1206 pckg /1%
13	4	R2, R15, R18, R21	1K	1206 pckg /5%
14	2	R9 , R10	10K	9 resistors resistor-network : Bourne 4610-101-103 (or equivalent)
15	4	R11 - R14	10K	1206 pckg /5%
16	1	R16	100 ohm	1206 pckg /5%
17	1	R17	500 ohm Trim Pot	Bourns 3386W-1-501 (or equivalent)
18	1	R19	100K Trim Pot	Bourns 3386W-1-104 (or equivalent)
19	2	R20, R22	100K	1206 pckg / 5%
20	1	R24	240 ohm	1206 pckg / 5%
21	2	S1, S2	SW DIP-8	Dip Switch 8 position
22	1	S3	SW DIP-4	Dip Switch 4 position
23	1	U1	SED1352F	QFP5-100 / 100 pin SOCKET Supplied by SMOS
24	2	U4, U5	SRM20256LM10	100ns 32K byte Static RAM - SMOS part number (SOP2 SMT)
25	2	U8, U9	74LS688	DW020 SMT
26	2	U10, U11	TIBPAL22V10-15BCNT	SOCKET + Component programmed by SMOS
27	1	U12	74LS09	D014 SMT
28	2	U13, U14	SN74LVT16244	SN74LVT16244 (SSOP)
29	2	U15, U16	SN74LVT16245	SN74LVT16244 (SSOP)
30	2	U17, U18	74HCT244	DW020 SMT
31	1	U19	LM317T	3-pin TO-220 regulator
32	1	U20	EPN001	XENTECK - Negative Power Supply Supplied by SMOS
33	1	U21	OSC-14	SOCKET Only

Appendix B SDU1352B0C REV. 1.0 SCHEMATIC DIAGRAMS

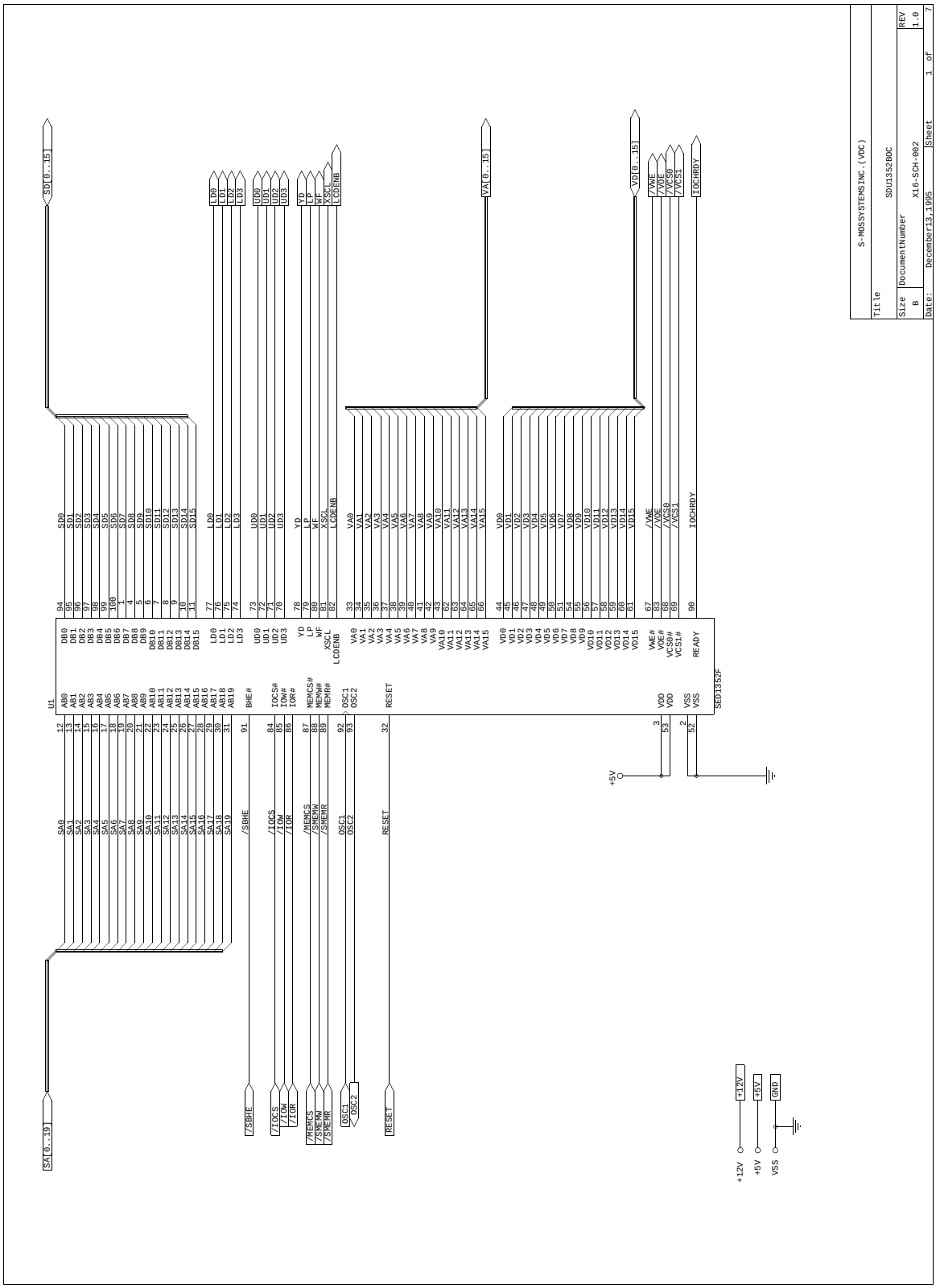


Figure 1: SDU1352B0C Rev. 1.0 Schematic Diagram (1 of 7)

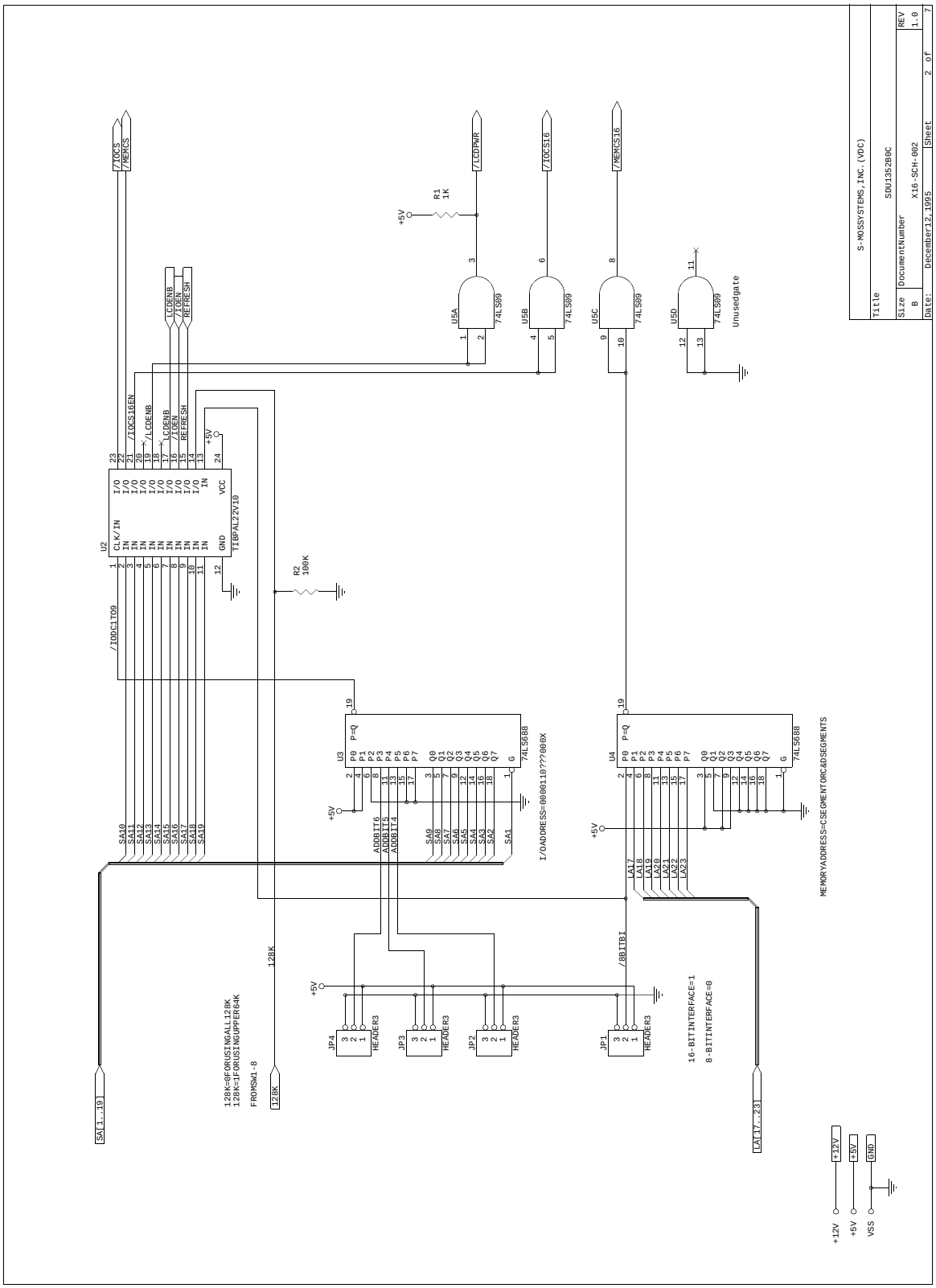


Figure 2: SDU1352B0C Rev. 1.0 Schematic Diagram (2 of 7)

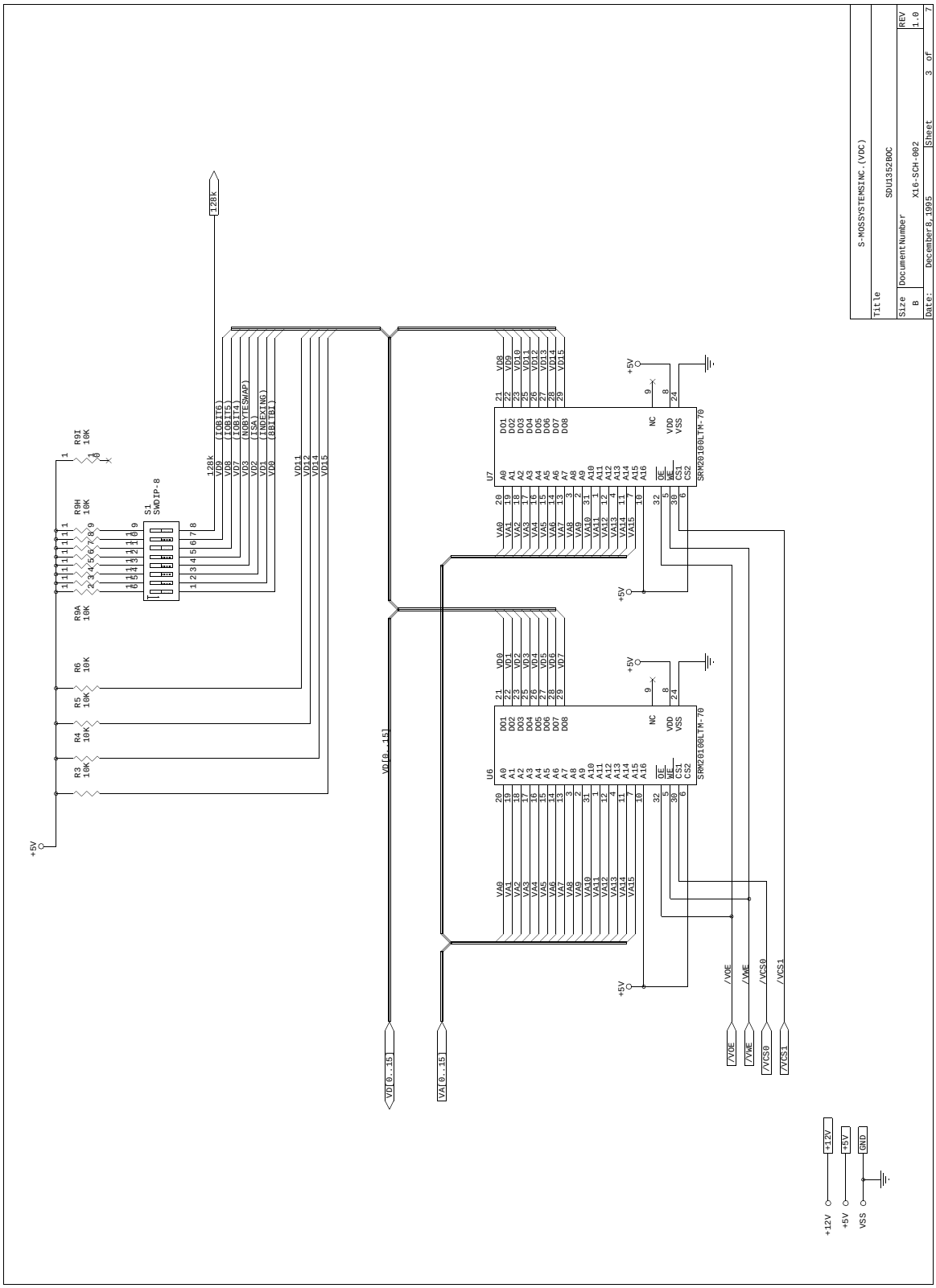


Figure 3: SDU1352B0C Rev. 1.0 Schematic Diagram (3 of 7)

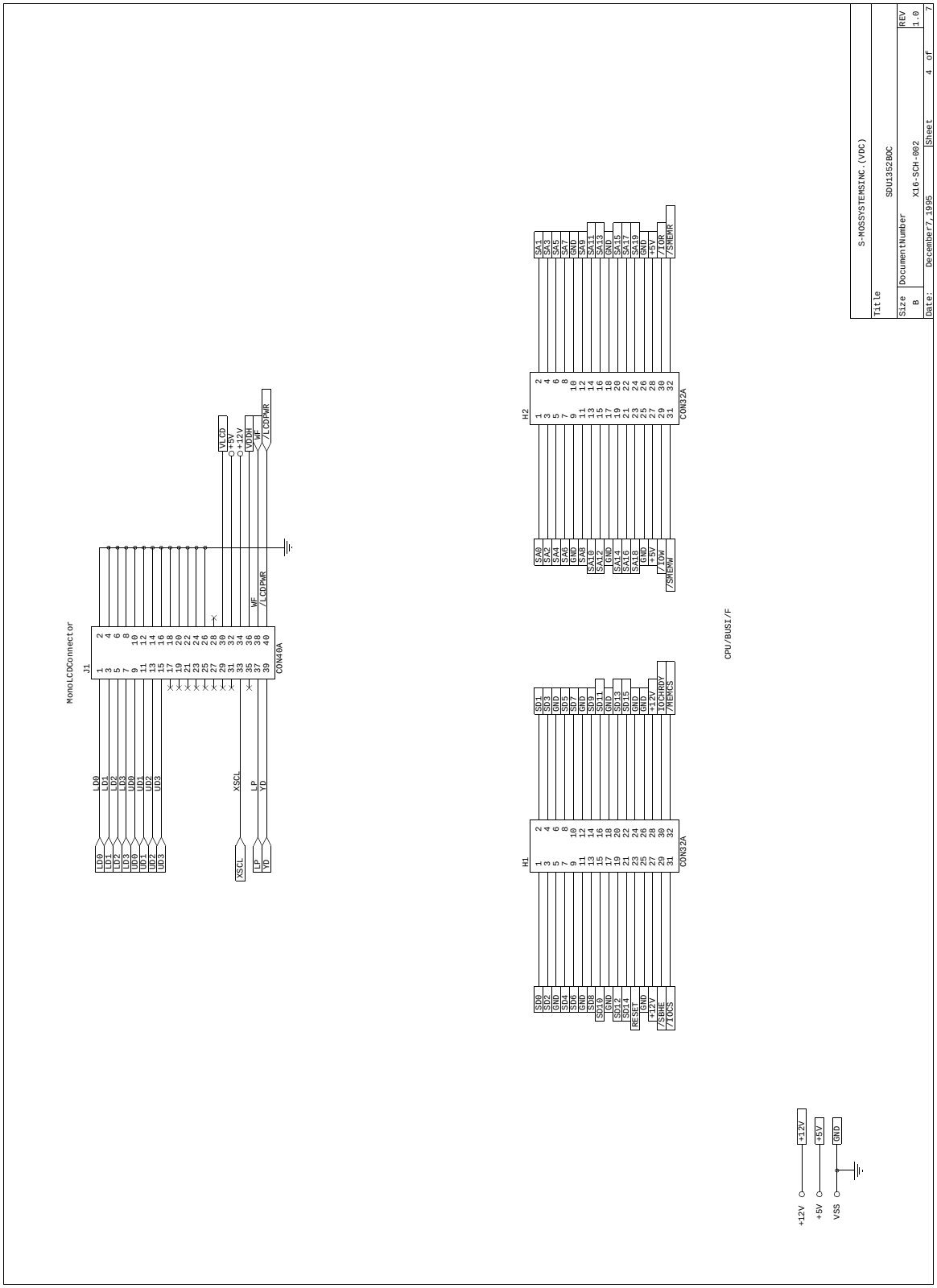


Figure 4: SDU1352B0C Rev. 1.0 Schematic Diagram (4 of 7)

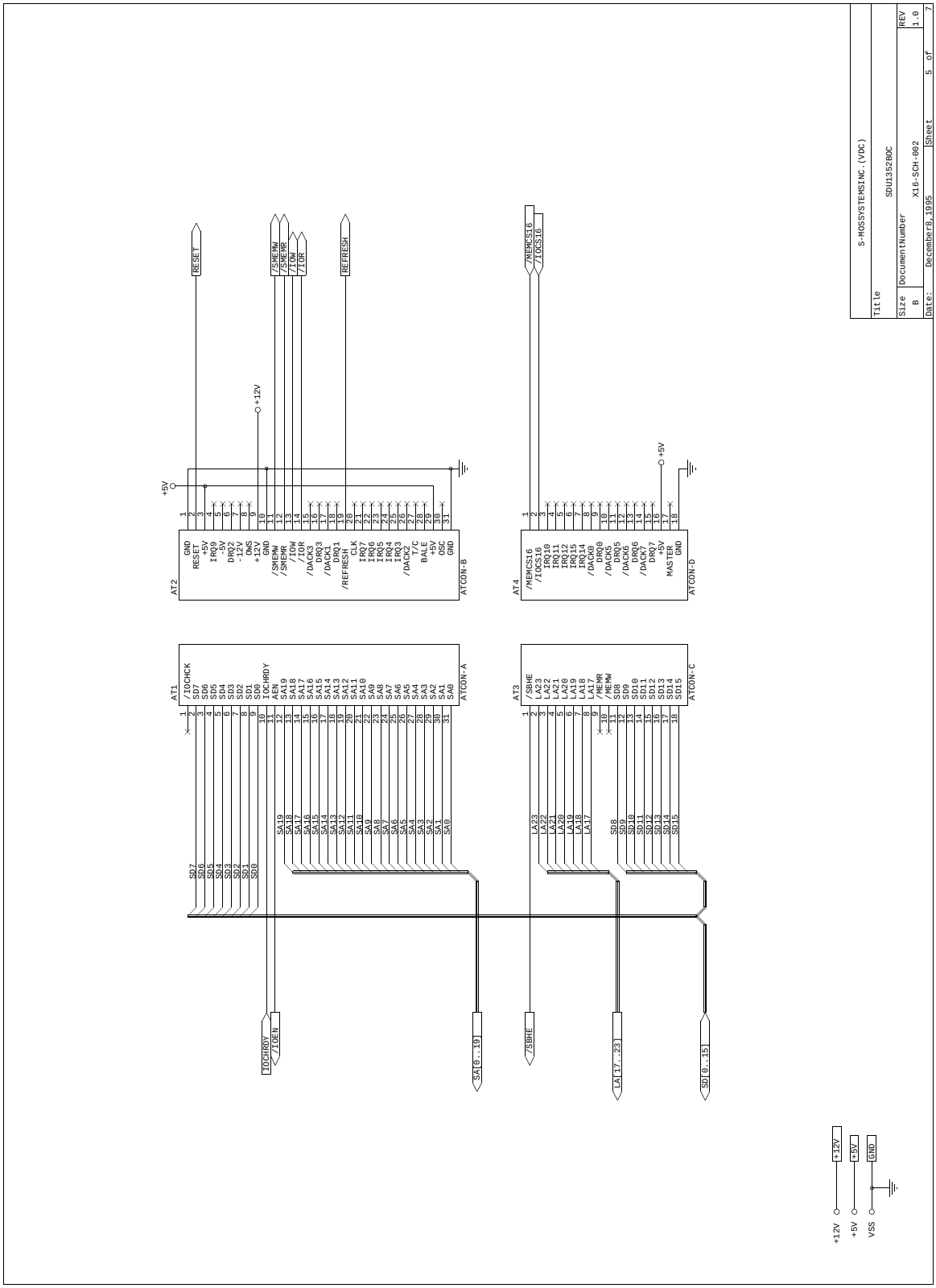


Figure 5: SDU1352B0C Rev. 1.0 Schematic Diagram (5 of 7)

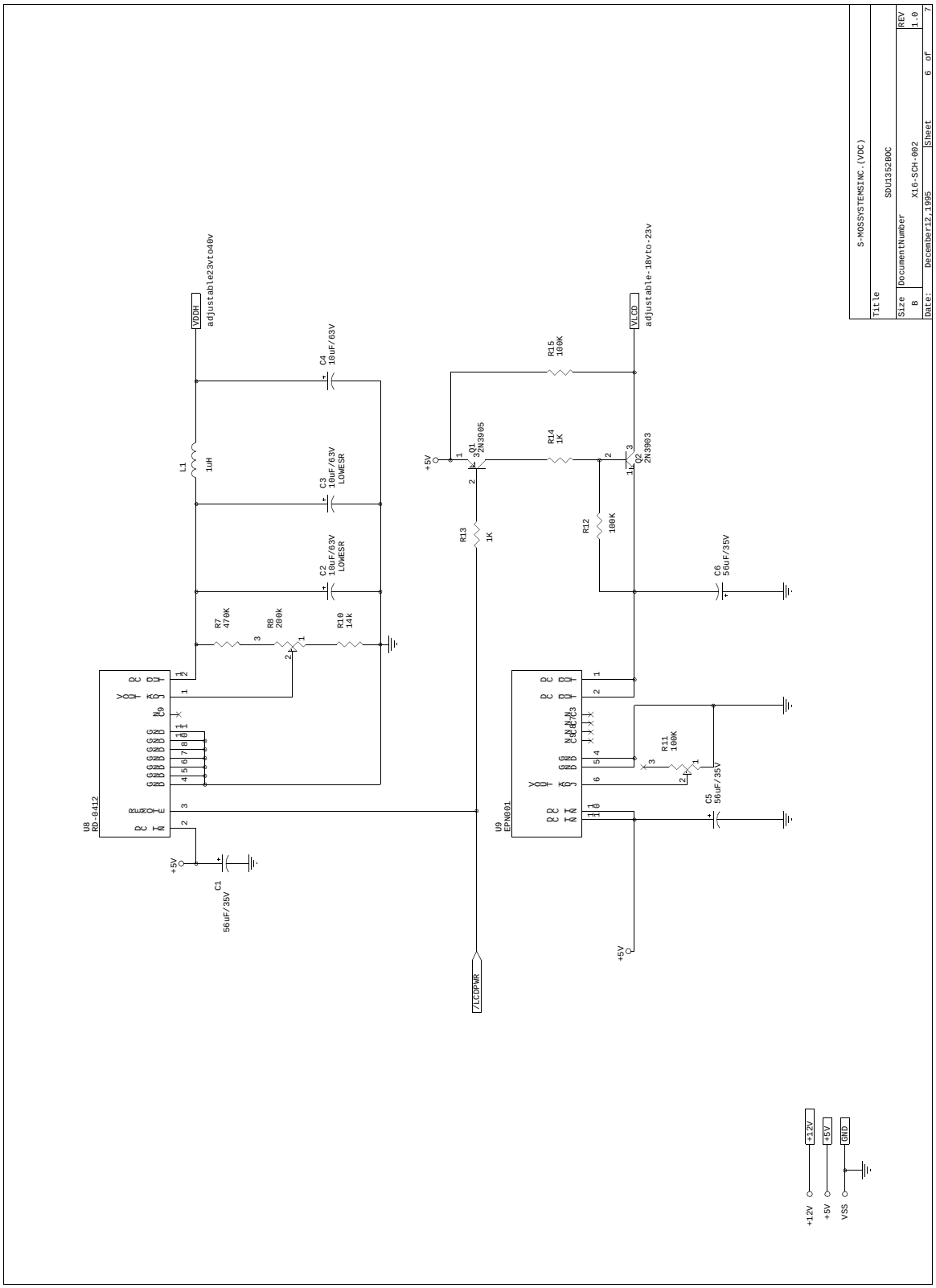


Figure 6: SDU1352B0C Rev. 1.0 Schematic Diagram (6 of 7)

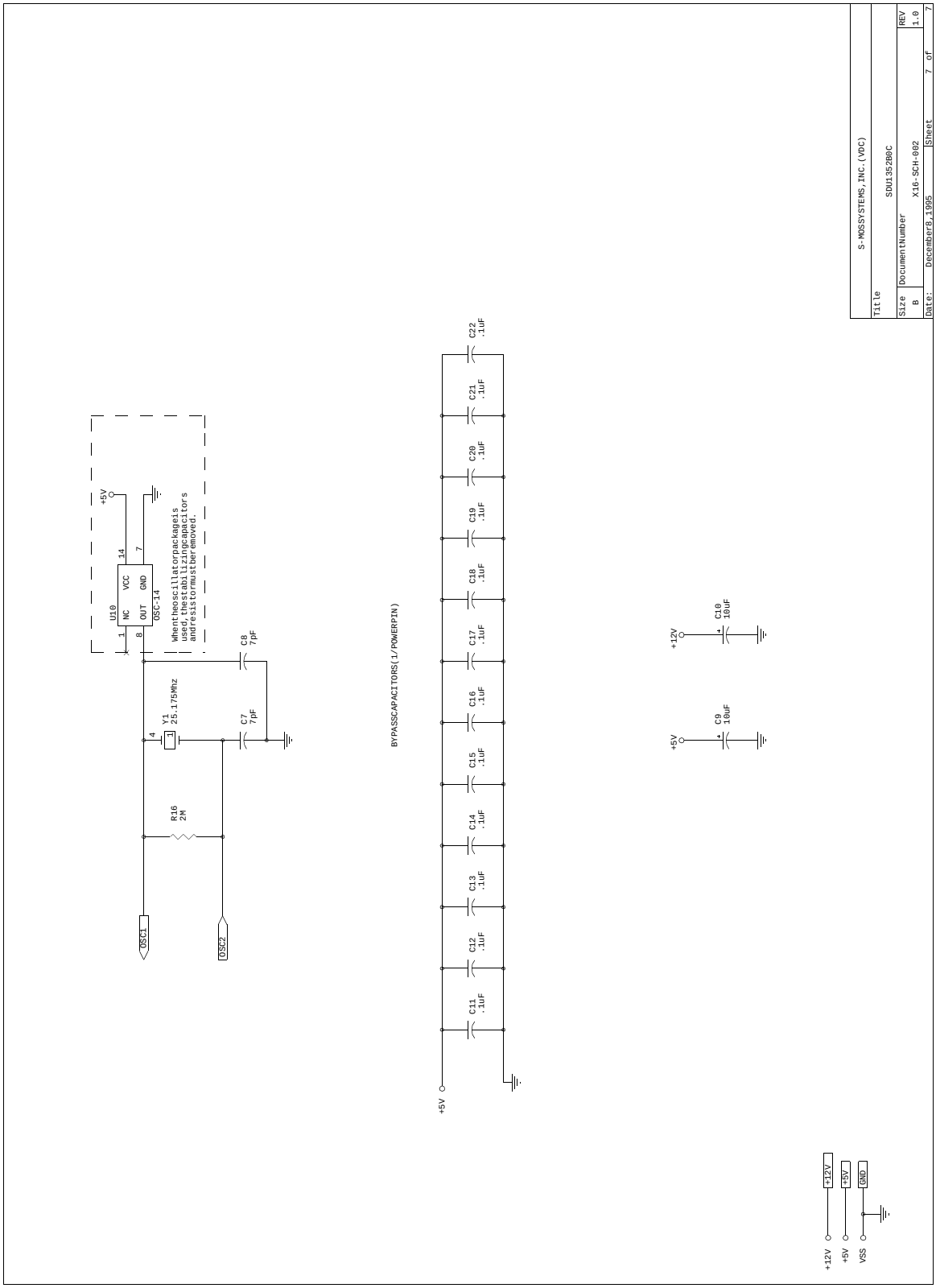


Figure 7: SDU1352B0C Rev. 1.0 Schematic Diagram (7 of 7)

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

Power Consumption

Document Number: X16-AN-006-06

Copyright © 1995, 1998 Epson Research and Development, Inc. All Rights Reserved.

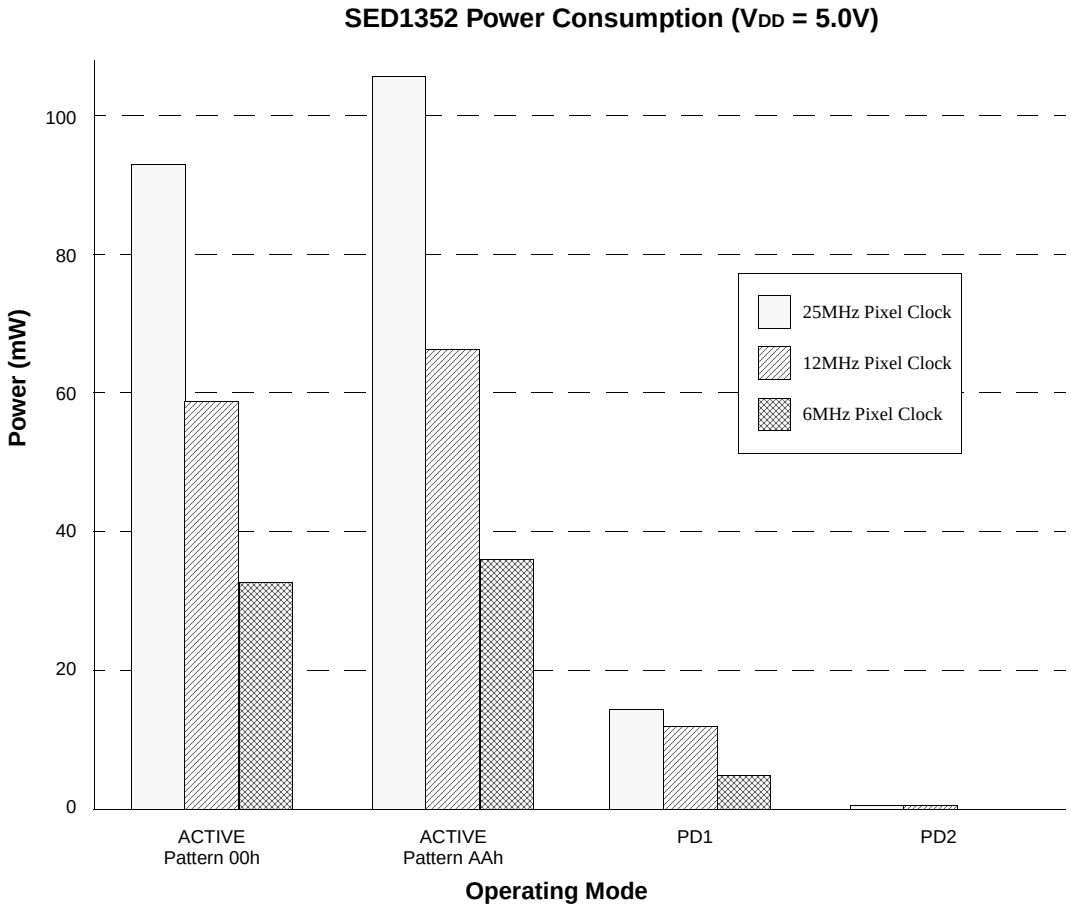
Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

1 SED1352 POWER CONSUMPTION

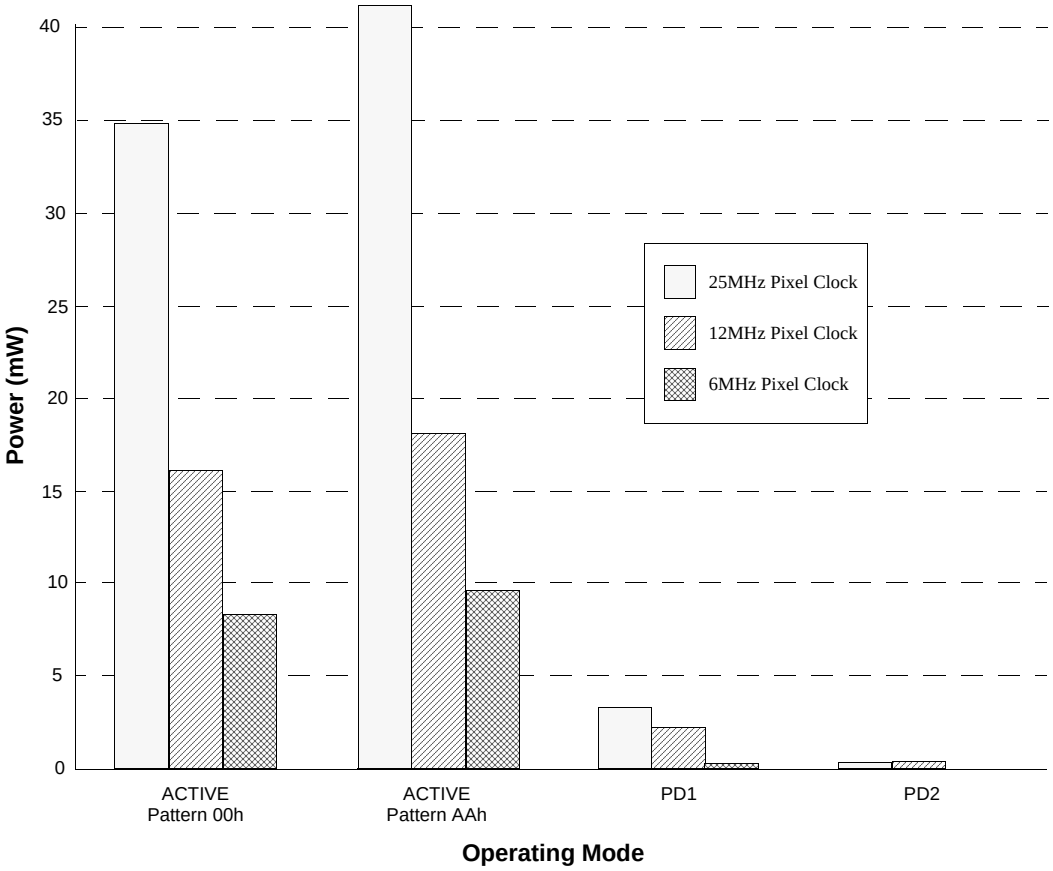
1.1 Conditions

1. Pixel clock = 25MHz: screen pattern = 00h and AAh on 640x480 single panel.
2. Pixel clock = 12MHz: screen pattern = 00h and AAh on 480x320 single panel.
3. Pixel clock = 6MHz: screen pattern = 00h and AAh on 320x240 single panel.
4. No display connected.



	Active Pattern 00h	Active Pattern AAh	PD1	PD2	Units
25MHz	93.0	105.5	14.3	0.1	mW
12MHz	58.7	66.1	11.8	0.1	mW
6MHz	32.7	36.1	4.7	0.0	mW

SED1352 Power Consumption (V_{DD} = 3.0V)



	Active Pattern 00h	**Active** Pattern AAh	**PD1**	**PD2**	**Units**
25MHz	35.0	41.2	3.3	0.2	mW
12MHz	16.1	18.1	2.2	0.3	mW
6MHz	8.3	9.5	0.2	0.0	mW

EPSON®

SED1352F0B Dot Matrix Graphics LCD Controller

ISA Bus Interface Considerations

Document Number: X16-AN-003-05

Copyright © 1995, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

Table of Contents

List of Figures

THIS PAGE LEFT BLANK

1 INTRODUCTION

The SED1352F0B is a general purpose LCD controller capable of interfacing to a variety of microprocessors. This interface is accomplished through the use of minimal external circuitry. This application note describes the interface between the SED1352F0B and the ISA Bus.

1.1 Reference Material

Refer to the SED1352F0B Hardware Functional Specification (X16-SP-001-xx) for complete AC timing details.

This document makes no attempts to describe the operation of the ISA Bus, please refer to the appropriate ISA Bus documentation for complete information.

2 16-BIT ISA BUS INTERFACE

For the purpose of the example shown below, the following conditions are set by default:

1. Indexing I/O with addresses 0310h and 0311h (see Configuration Options)
2. 128Kbytes of display memory occupying $C and $D segments (see Configuration Options)

Note
This memory configuration will conflict with a VGA card installed on the same bus, therefore either a serial terminal or monochrome display adapter is recommended as the primary console.

This section provides the necessary equations and settings to complete the interface between the SED1352F0B and the 16-bit ISA Bus.

Note
A PAL was used instead of discrete logic to reduce external component count.

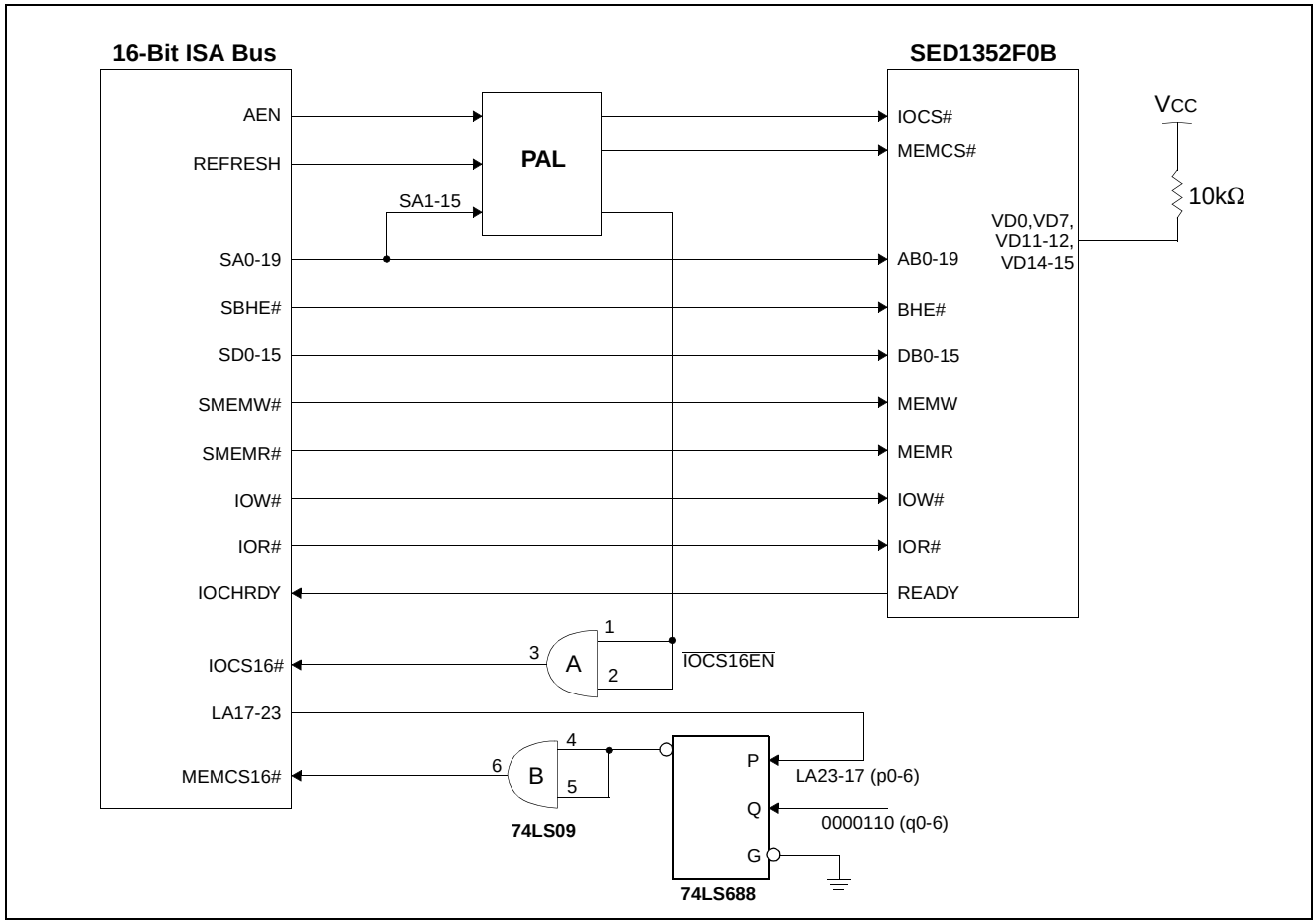


Figure 8: 16-Bit ISA Bus Implementation

1.2 PAL Equations

The PAL is programmed with the following equations:

1. As stated above, the default I/O address is from 0310h to 0311h. The SED1352F0B provides internal decoding of address bits A0 to A9, therefore minimal external circuitry is necessary to provide signals IOCS# and IOCS16#

   IOCS# is required by the SED1352 to indicate a valid IO cycle. In an ISA bus environment, valid IO decoding must include addresses A15-A0. Given this example, addresses A10-15 must all be '0' and AEN must also be '0'.
   IOCS# = !(!AEN & !A15 & !A14 & !A13 & !A12 & !A11 & !A10)

2. As the SED1352 is capable of 16-bit IO access, the IOCS16# bus signal must be driven externally to indicate such a cycle. As stated in the ISA specification, the IOCS16# is a straight address decode without qualification.

   IOCS16EN# = !(!IOCS# & A9 & A8 & !A7 & !A6 & !A5 & A4 & !A3 & !A2 & !A1)

3. With 128Kbytes of display memory and A17 to A19 decoded internally to SED1352F0B;
   MEMCS# = !REFRESH

1.3 Additional Discrete Logic Description

1. As shown in Figure 1, the 74LS688 is configured as a memory decoder with valid addresses between 0Cxxxxh and 0Dxxxxh.
2. The 74LS09 is used simply to provide the Open-Collector outputs necessary for the IOCS16# and MEMCS16# signals.

1.4 SED1352F0B Default Setup

1.4.1 Configuration Options

1.	VD15 - VD13 = 110	memory decoding for locations $C and $D segments
2.	VD12 - VD4 = 110001000	I/O decoding for locations 1100010000b - 1100010001b
3.	VD3 = 0	no byte swap of high and low bytes
4.	VD2 = 0	ISA Bus interface, i.e. non- MC68K interface
5.	VD1 = 0	indexing I/O
6.	VD0 = 1	16-bit bus interface

Where 1 = pull-up with a 10K resistor; 0 = no pull-up resistor

Note
The states of these data pins are internally latched during RESET.

1.4.2 Register Setting

AUX[1] bit 1 = 0 for 16-bit memory interface (must be 16-bit with a 16-bit bus).

3 8-BIT ISA BUS INTERFACE

For the purpose of the example shown below, the following conditions are set by default:

1. Indexing I/O with partial decoding, i.e. address lines A10 to A15 are not decoded for I/O cycles

Note
Partial decoding is quite safe on most ISA Bus systems as I/O addresses above 03FFh are rarely used.

2. I/O addresses are xxxxxx1100000000b and xxxxxx1100000001b
3. 64Kbytes of display memory occupying $A segment

Note
The 74LS00 is simply used to detect the $B segment and invalidate the MEMCS# input.

Note
This memory configuration will conflict with a VGA card installed on the same bus, therefore either a serial terminal or monochrome display adapter is recommended as the primary console.

This section provides the necessary settings to complete the interface between the SED1352F0B and the 8-bit ISA Bus. Since I/O addresses are partially decoded, there is no need to use a PAL for decoding.

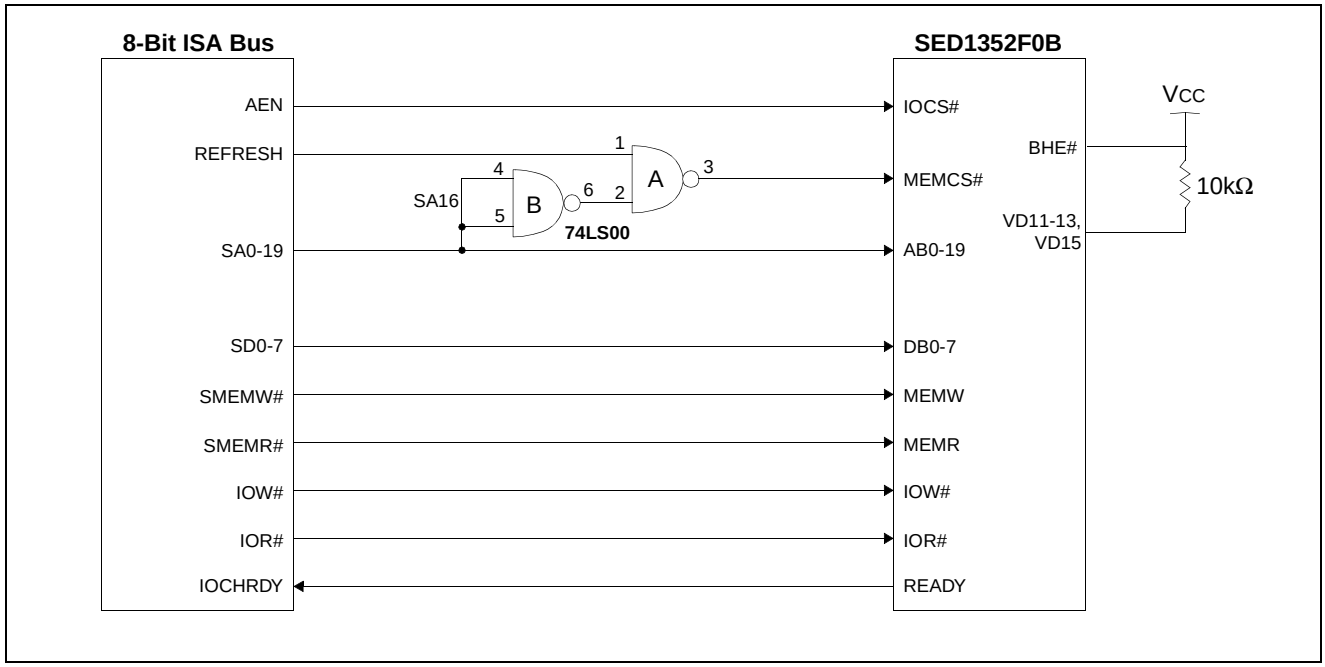


Figure 9: 8-Bit ISA Bus Implementation

1.5 SED1352F0B Default Setup

1.5.1 Configuration Options

1. VD15 - VD13 = 101 memory decoding for locations $A segment
2. VD12 - VD4 = 110000000 I/O decoding for locations 1100000000b - 1100000001b
3. VD3 = 0 No byte swap of high and low bytes
4. VD2 = 0 ISA Bus interface, i.e. non- MC68K interface
5. VD1 = 0 Indexing I/O
6. VD0 = 0 8-bit bus interface

Where 1 = pull-up with a 10K resistor; 0 = no pull-up resistor

Note
The states of these data pins are internally latched during RESET.

1.5.2 Register Setting

AUX[1] bit 1 = 0 for 16-bit memory interface or

AUX[1] bit 1 = 1 for 8-bit memory interface.

THIS PAGE LEFT BLANK

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

MC68340 Interface Considerations

Document Number: X16-AN-004-06

Copyright © 1996, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

Table of Contents

List of Figures

THIS PAGE LEFT BLANK

1 INTRODUCTION

The SED1352 is a general purpose LCD controller capable of interfacing to a variety of microprocessors. This interface is accomplished through the use of minimal external circuitry. This application note describes the interface between the SED1352 and the 16-bit MC68340 microcontroller.

1.1 Reference Material

Refer to the SED1352 Hardware Functional Specification (X16-SP-001-xx) for complete AC timing details.

This document makes no attempts to describe the operation of the MC68340 microcontroller, please refer to the appropriate MC68340 documentation for this information.

2 MC68340 MPU INTERFACE

The following sections provide the necessary settings and equations to complete the interface between the SED1352 and the MC68340 microcontroller.

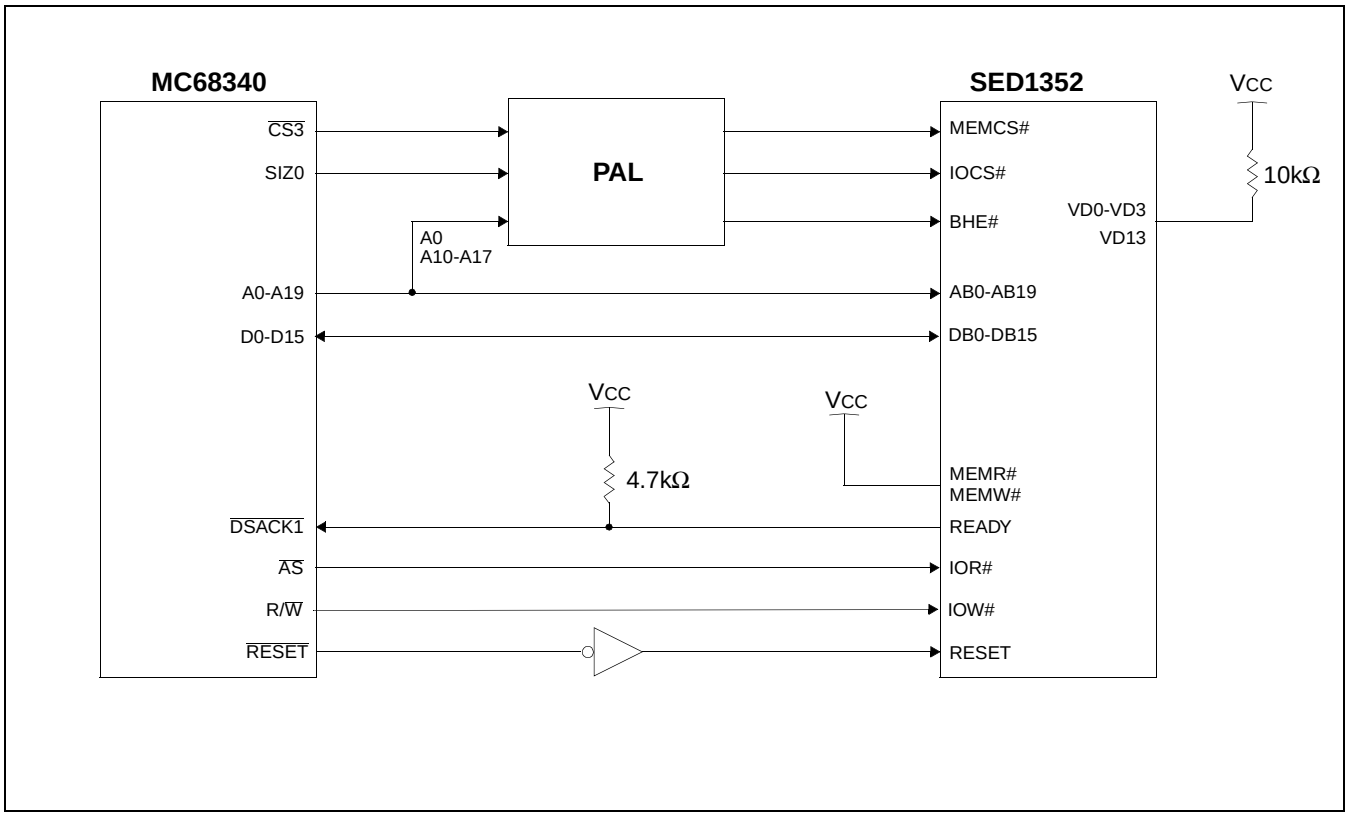


Figure 1: MC68340 MPU Interface Block Diagram

2.1 MC68340 Setup

For the purpose of this example, the following conditions apply:

The internal chip select signal $\overline{CS3}$ of the MC68340, along with external $\overline{DSACK1}$ response, is employed to access the SED1352. Direct mapping of the I/O with starting address at 00000000h, and 128Kbytes of display memory with starting address 00020000h are also used.

1. $\overline{CS3}$ with 256kbyte block size - starting address at 00000000h and ending address at 0003FFFFh
2. External $\overline{DSACK1}$ response - 16-bit port
3. Don't care Function Codes and with CPU space access
4. Both read and write accesses are allowed

Settings for the Address Mask register and Base Address register for the above conditions are:

058h - 05Bh	= 0003FFFFh	Address Mask register
05Ch - 05Fh	= 000000F5h	Base Address register

2.2 PAL Equations

The PAL is programmed with the following equations:

1. With direct-mapping I/O occupying locations from 00000000h to 0000000Fh and A4 to A9 decoded internally to SED1352;
   IOCS# = !(!$\overline{CS3}$ & !A17 & !A16 & !A15 & !A14 & !A13 & !A12 & !A11 & !A10)
2. With memory locations from 00020000h to 003FFFFh and A17 to A19 decoded internally to SED1352;
   MEMCS# = $\overline{CS3}$
3. BHE# becomes valid for two conditions:
   1. 16-bit or 32-bit cycle, i.e., SIZ0=0
   2. 8-bit cycle with odd byte access, i.e., SIZ0=1 and A0=1;

   BHE# = SIZ0 & !A0

2.3 SED1352 Default Setup

Configuration Options

1.	VD15 - VD13 = 001	memory decoding for locations 20000h - 3FFFFh
2.	VD12 - VD4 = 000000xxx	I/O decoding for locations 0000000000b - 0000001111b
3.	VD3 = 1	byte swap of high and low bytes
4.	VD2 = 1	MC68K interface
5.	VD1 = 1	direct-mapping I/O
6.	VD0 = 1	16-bit bus interface

Where x = don't care; 1 = pull-up with a 10K resistor; 0 = no pull-up resistor

Note
The states of these data pins are internally latched during RESET.

Register Setting

AUX[01h] bit 1 = 0 for 16-bit memory interface (must be 16-bit with a 16-bit bus).

THIS PAGE LEFT BLANK

EPSON®

SED1352 Dot Matrix Graphics LCD Controller

LCD Panel Options / Memory Requirements

Document Number: X16-AN-005-07

Copyright © 1995, 1998 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice. You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.
EPSON is a registered trademark of Seiko Epson Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

Table of Contents

List of Figures

THIS PAGE LEFT BLANK

1 INTRODUCTION

The SED1352 is a highly configurable general purpose LCD controller. The LCD panel frame-rate, resolution, and gray shades all determine the memory and input clock requirements. This application note will describe the equations used to determine the various parameters. An example resolution and desired frame-rate will be selected and used to determine the remaining variables.

1.1 Reference Material

Refer to the SED1352 Hardware Functional Specification (X16-SP-001-xx) for complete AC timing details.

2 CONFIGURATION EQUATIONS

2.1 Example:

LCD panel resolution:	640x240
LCD panel configuration	4 bit, Single drive panel
LCD Gray Shades	4
Desired Frame-rate:	~70Hz

2.1.1 Input Clock Requirement

For a frame rate of 70Hz, the input clock (or pixel clock) frequency can be calculated as following,

f_{OSC} = input clock

f_{OSC} = Frame Rate * (# of horizontal pixels + 16) * (# of vertical lines + 4)

Therefore;

f_{OSC} = 70 * (640 + 16) * (240 + 4)

f_{OSC} = 11.2MHz

Note
1. Due to oscillator frequency availability, a 12MHz oscillator is selected thus producing a slightly higher frame-rate (~75Hz).

2. For a detailed description of the frame rate formula, see section 9.3 of the SED1352 Hardware Functional Specification, drawing office number X16-SP-001-xx.

2.2 SRAM Size and Access Time Requirements

2.2.1 SRAM Size

$$\text{Memory Size (bytes)} = \frac{(\text{\# of Horizontal pixels}) * (\text{\# of Vertical pixels})}{8 / (\text{\# of bits/pixel})}$$

i.e., 4 gray shades = 2 bits / pixel, therefore 1 byte (8 bits) = 4 pixels

Therefore:

Memory size (bytes) = (640 * 240) / 4

Memory size (bytes) = 37.5 K bytes.

Note
For a detailed description of the memory size requirement, see section 9.4 of the SED1352 Hardware Functional Specification, drawing office number X16-SP-001-xx.

2.2.2 SRAM Access Time

For 8-bit display memory interface;
Access time $\leq 2/f_{OSC} - 50$.

With 12MHz input clock;
Access time $\leq$ 116ns.

For 16-bit display memory interface;
Access time $\leq 4/f_{OSC} - 50$.

With 12MHz input clock, access time $\leq$ 283ns.

Note
For detail description of the SRAM access time, see section 9.2 of the SED1352 Hardware Functional Specification, drawing office number X16-SP-001-xx.

3 IMPLEMENTATION

3.1 8-Bit Display Memory Interface

Since 35.7K bytes with at least 116ns access time SRAM is required, one 8K bytes SRAM with 100ns access time, and one 32K bytes SRAM with 100ns access time are used in this example.

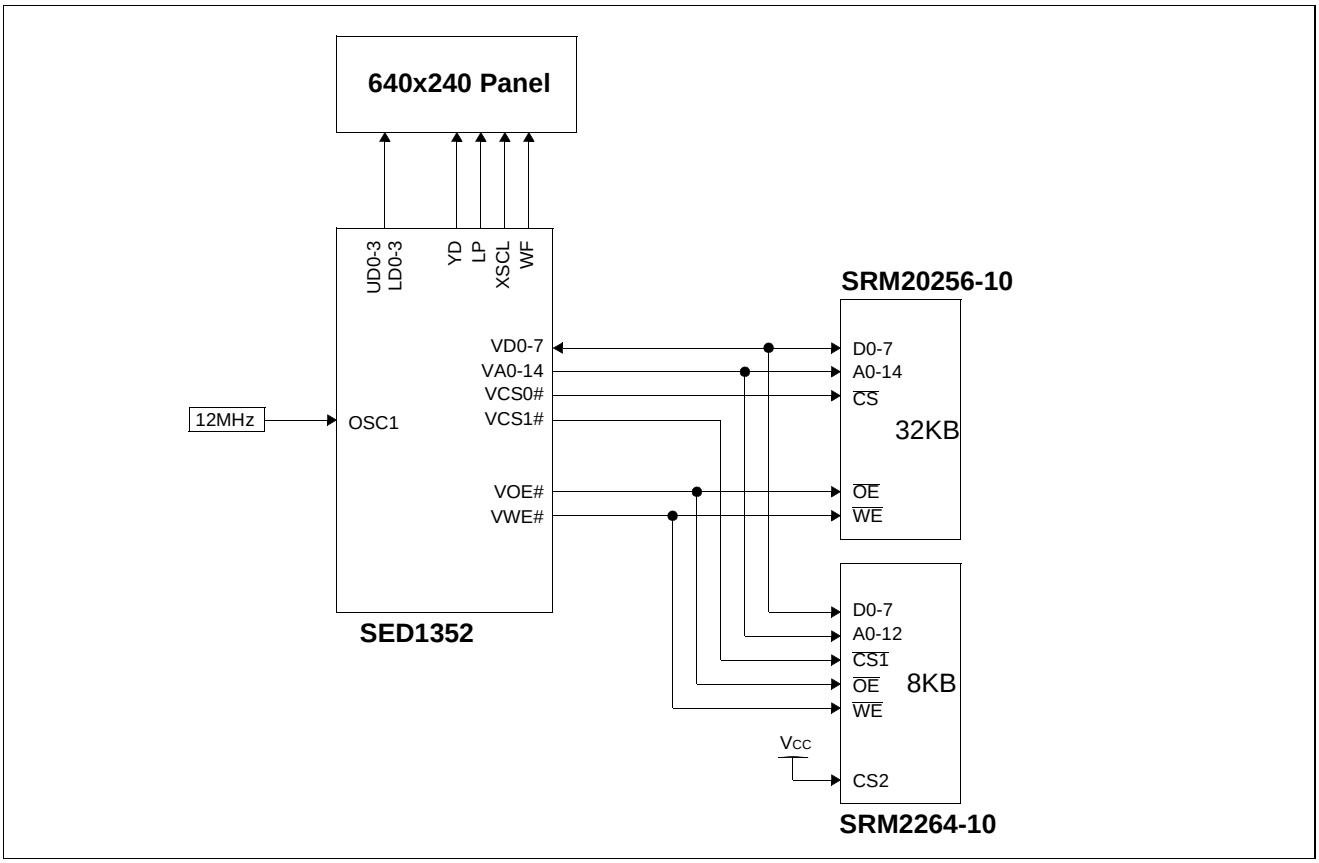


Figure 1: 8-Bit Memory Configuration Example

3.1.1 Configuration Options

VD0 selects 16/8-bit Bus interface. When using a 8-bit memory interface, the 8-bit Bus interface must also be selected. The state of VD0 is internally latched during RESET. In this example VD0 has no external pull-up resistor and is therefore latched as a ’0’ during RESET (due to the internal pull-down resistors) thus selecting the 8-bit Bus interface.

Other option settings are not related to this implementation.

3.1.2 Register Settings

AUX[00h] = 0000 0000 not in test mode
AUX[01h] = 1001 0011 4-bit single panel, 4 gray shades, 8-bit display memory interface with 32K bytes is the first chip
AUX[02h] = 1001 1111 horizontal resolution = 640 ; 4 gray shades = 4 pixels per byte ; 4 pixels per fetch
AUX[03h] = 0000 0000 not in power save modes
AUX[04h] = 1110 1111 total 240 scan lines
AUX[05h] = 0000 0000 WF = 0
AUX[06h] = 0000 0000
AUX[07h] = 0000 0000 default starting address at 0000h (with AUX[06h])
AUX[08h] = xxxx xxxx don't care when not using split screen
AUX[09h] = xxxx xxxx don't care when not using split screen
AUX[0Ah] = 1110 1111 together with AUX[0Bh] bit1-0, should be the same as or larger than AUX[05h] bit1-0 and
AUX[0Bh] = xxxx xx00 AUX[04h] when not using split screen
AUX[0Dh] = 0000 0000 no virtual screen

Example setting of Look-up Table when using bank# 0 for display:

AUX[0Eh] = 00xx 0000 index = 0
AUX[0Fh] = xxxx 0000 gray = 0
AUX[0Eh] = 00xx 0001 index = 1
AUX[0Fh] = xxxx 0101 gray = 5
AUX[0Eh] = 00xx 0010 index = 2
AUX[0Fh] = xxxx 1010 gray = A
AUX[0Eh] = 00xx 0011 index = 3
AUX[0Fh] = xxxx 1111 gray = F

x = don't care

3.2 16-bit Display Memory Interface

Since 35.7K bytes with at least 283ns access time SRAM is required, two 32K bytes SRAM with 120ns access time are used for this example.

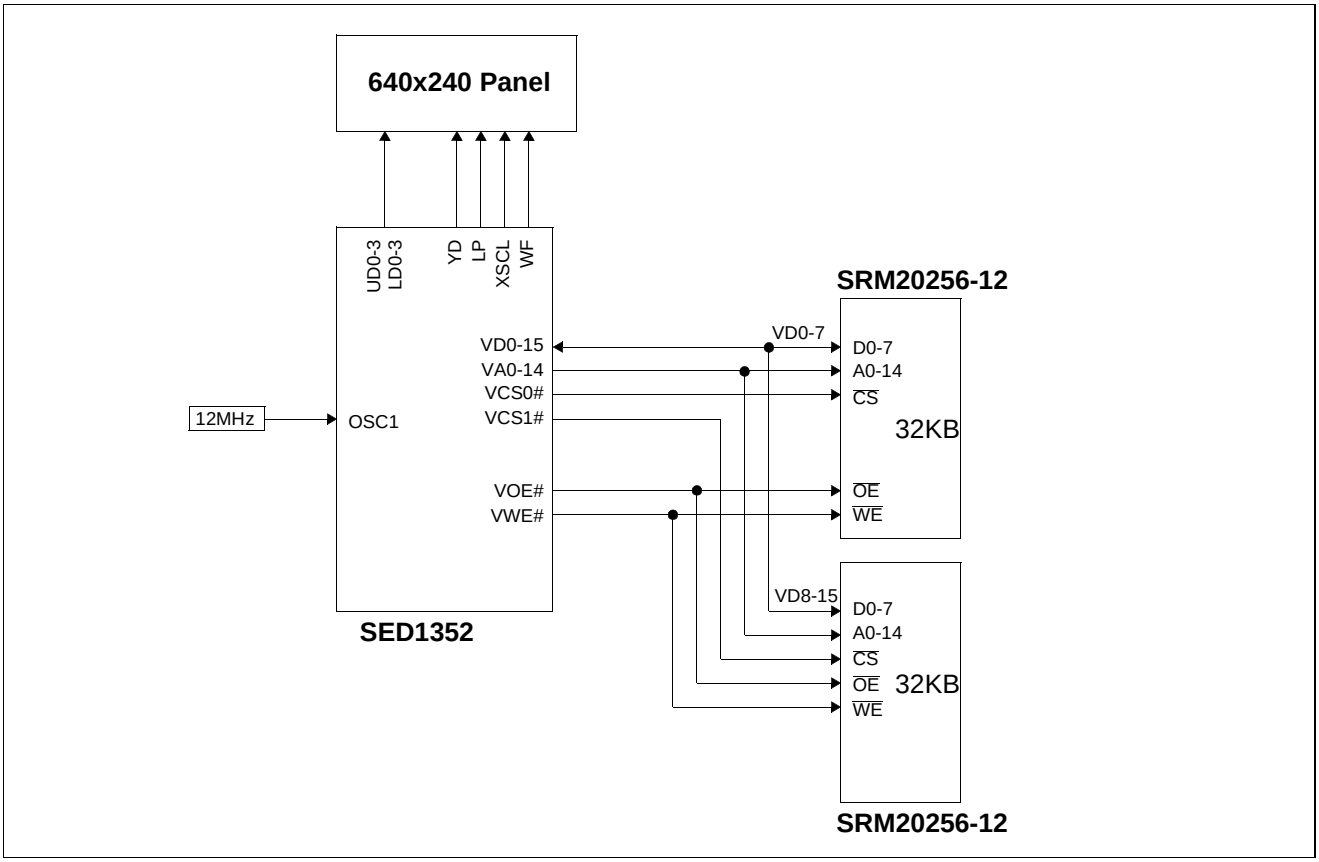


Figure 2: 16-Bit Memory Configuration Example

3.2.1 Configuration options

VD0 = no pull-up resistor for 8-bit bus interface or
VD0 = pull-up (with a 10K resistor) for 16-bit bus interface.

Other option settings are not related to this implementation.

3.2.2 Register settings

AUX[00h] = 0000 0000	not in test mode
AUX[01h] = 1001 000x	4-bit single panel, 4 gray shades, 16-bit display memory interface
AUX[02h] = 0100 1111	horizontal resolution = 640 ; 4 gray shades = 4 pixels per byte ; 8 pixels per fetch
AUX[03h] = 0000 0000	not in power save modes
AUX[04h] = 1110 1111	total 240 scan lines
AUX[05h] = 0000 0000	WF = 0
AUX[06h] = 0000 0000	
AUX[07h] = 0000 0000	default starting address at 0000h (with AUX[06h])
AUX[08h] = xxxx xxxx	don,t care when not using split screen
AUX[09h] = xxxx xxxx	don,t care when not using split screen
AUX[0Ah] = 1110 1111	together with AUX[0Bh] bit1-0, should be the same as or larger than AUX[05h] bit1-0 and
AUX[0Bh] = xxxx xx00	AUX[04h] when not using split screen
AUX[0Dh] = 0000 0000	no virtual screen

Example setting of Look-up Table when using bank# 2 for display:

AUX[0Eh] = 10xx 1000	index = 8
AUX[0Fh] = xxxx 0000	gray = 0
AUX[0Eh] = 10xx 1001	index = 9
AUX[0Fh] = xxxx 0101	gray = 5
AUX[0Eh] = 10xx 1010	index = A
AUX[0Fh] = xxxx 1010	gray = A
AUX[0Eh] = 10xx 1011	index = B
AUX[0Fh] = xxxx 1111	gray = F

x = don’t care

Note
When LCDENB (bit 4 of AUX[01h]) is used to control the LCD power, the following sequence is recommeded to setup the AUX registers of the SED1352:

1. Write to bit 4 of AUX[01h] with value ’0’.
2. Setup the AUX registers accordingly.
3. Delay at least half a second (depend on panel type, it may be required more time delay).
4. Write to bit 4 of AUX[01h] with value ’1’.

THIS PAGE LEFT BLANK