S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

SPC8104 Technical Manual

Issue Date: 04/01/97

THIS PAGE INTENTIONALLY BLANK

1.0 CUSTOMER TECHNICAL SUPPORT

S-MOS Systems provides Technical Support Services to their customers for all its graphics products. Follow the technical support guidelines as listed below for the S-MOS graphics product families — VGA LCD Controllers, LCD Controllers, LCD Driver/Controllers, and LCD Drivers.

On-Line

This Technical Manual and new products can be accessed at **www.smos.com** website

Technical Manuals

S-MOS provides detailed Technical Manuals on all its graphics products. Customers should contact their local sales representative or the nearest S-MOS sales office for a unit.

Evaluation Boards

Customers may borrow evaluation boards for the SPC81xx VGA LCD Controller family and the SED13xx LCD Controller family by contacting their local sales representative or the nearest S-MOS sales office.

Technical Support

There are a number of ways to contact Graphics Technical Support personnel at S-MOS. For customers in the Northeast and North Central United States, please contact the Wakefield S-MOS office for technical support on all products. For customers in the Southeast and South Central United States, please contact the Raleigh S-MOS office for technical support on all products. For all other regions contact S-MOS San Jose.

Sales Inquiries

For pricing, delivery, or any other non-technical inquiry, contact your local sales representative or the nearest S-MOS sales office.**:**

Northern California

S-MOS Systems, Inc.
150 River Oaks Parkway
San Jose, CA 95134
Tel: (408) 922-0200
Fax: (408) 922-0238

North Central

S-MOS Systems Inc., North Central
1450 East American Lane, Suite 1550
Schaumburg, IL 60713
Tel: (847) 517-7667
Fax: (847) 517-7601

Northeast

S-MOS Northeast Systems, Inc.
301 Edgewater Place #120
Wakefield, MA 01880
Tel: (617) 246-3600
Fax: (617) 246-5443

Southeast

S-MOS Southeast Systems, Inc.
4300 Six Forks Road #430
Raleigh, NC 27609
Tel: (919) 781-7667
Fax: (919) 781-6778

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY BLANK

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

Data Sheet

THIS PAGE INTENTIONALLY BLANK

■ DESCRIPTION

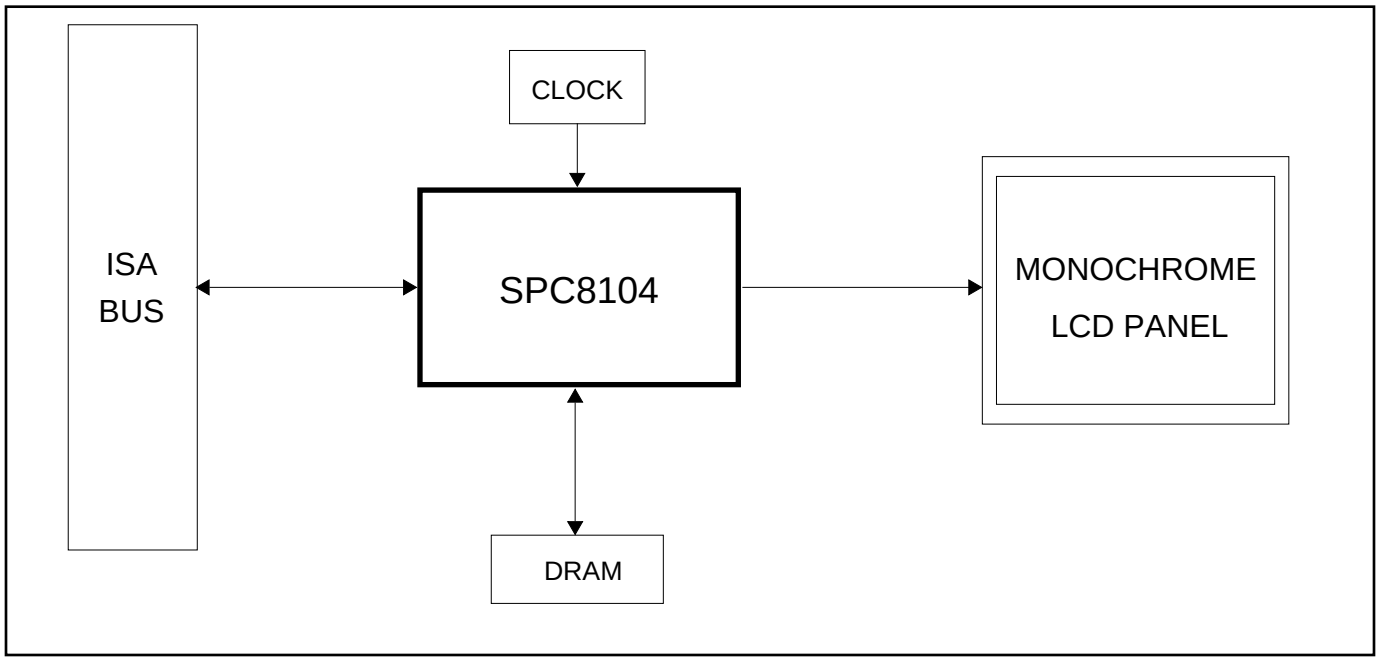
The SPC8104 is a low power, mixed 2.5/3.3 volt graphics controller based on VGA architecture and optimized for driving a 640x480 LCD panel display. VGA standard mode functionality (with the exception of mode 13h) is supported using standard IBM VGA parameters. A proprietary 64x4-bit gray scale lookup table is provided to allow re-mapping of the 16 possible gray shades displayed on an LCD panel.

The target markets for this device are small, cost sensitive mixed 2.5V/3.3V hand-held organizers, or other specialized consumer products where low cost, low power consumption, low component count, and the ability to run most VGA software on a 640x480 LCD panel display are the major design considerations. This chip is intended to operate mainly in planar graphics modes.

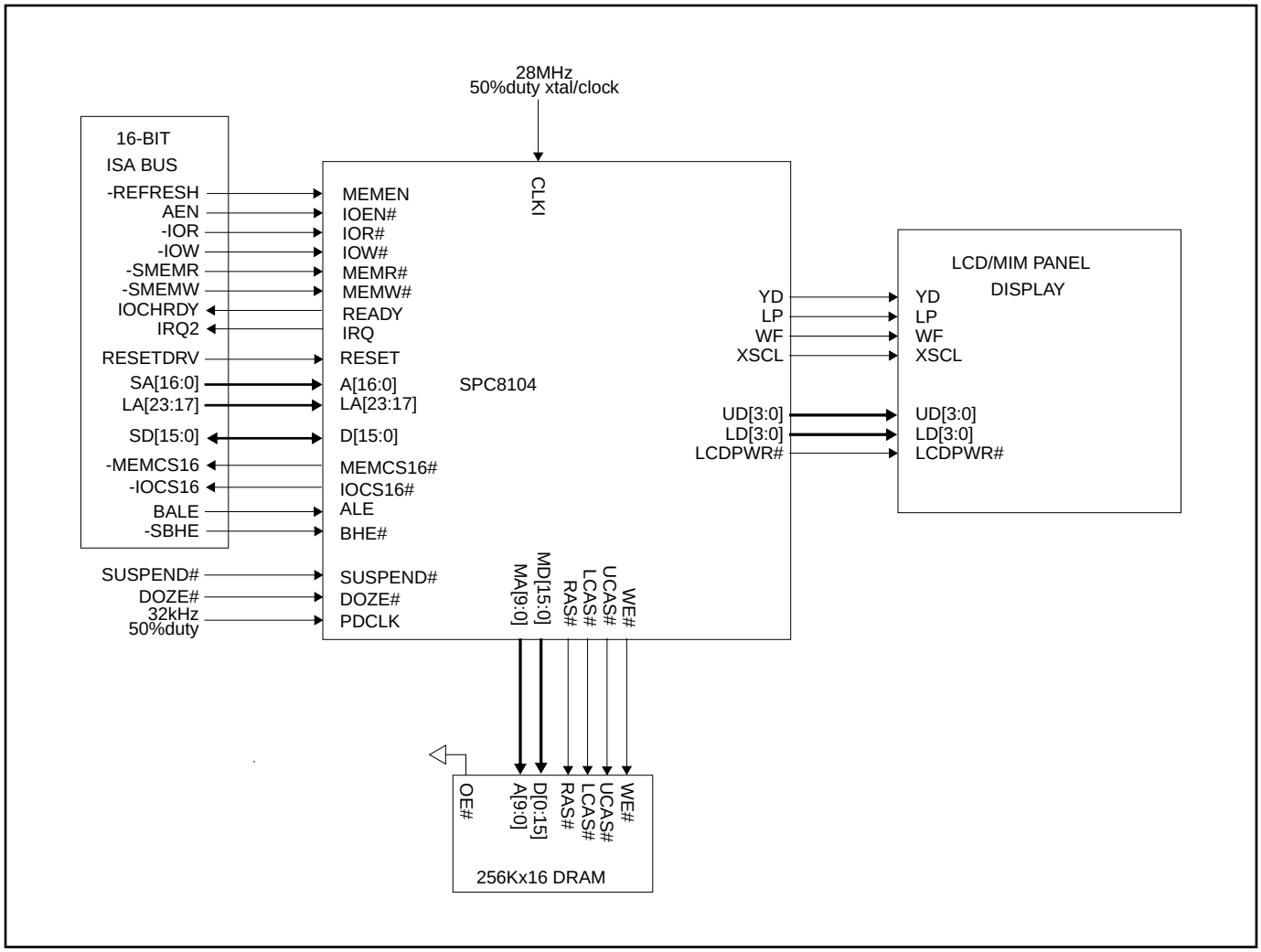
■ FEATURES

- Low power CMOS with 2.5V/3.3V core and 3.3V I/O
- 8/16-bit ISA CPU data bus interface
- Interfaces to a single 256 Kx16 DRAM
- Selectable 256 cycle/4 msec or 256 cycle/32 msec DRAM refresh rate, or low power self-refresh mode
- Three hardware or software initiated power-save modes
- Supports all standard VGA modes except mode 13h.
- Proprietary internal 64x4 gray scale lookup table
- Programmable hardware mapping of VGA palette-style writes to 16 level LCD gray scale values
- Vertical interrupt function on IRQ pin supported
- Optimized for 640x480 single and dual panel monochrome LCD displays
- Flexible support of LCD panels of various sizes
- Supports 0-255 vertical non-display periods
- Supports 640x480 4-bit monochrome MIM panels
- Power consumption of 60 mW in active mode and 0.6 mW in “Power Save” mode when operating at 24 MHz
- F0A - 128 pin QFP15 package

■ SYSTEM BLOCK DIAGRAM



■ INTERFACE OPTIONS



Note: Example implementation, actual may vary.

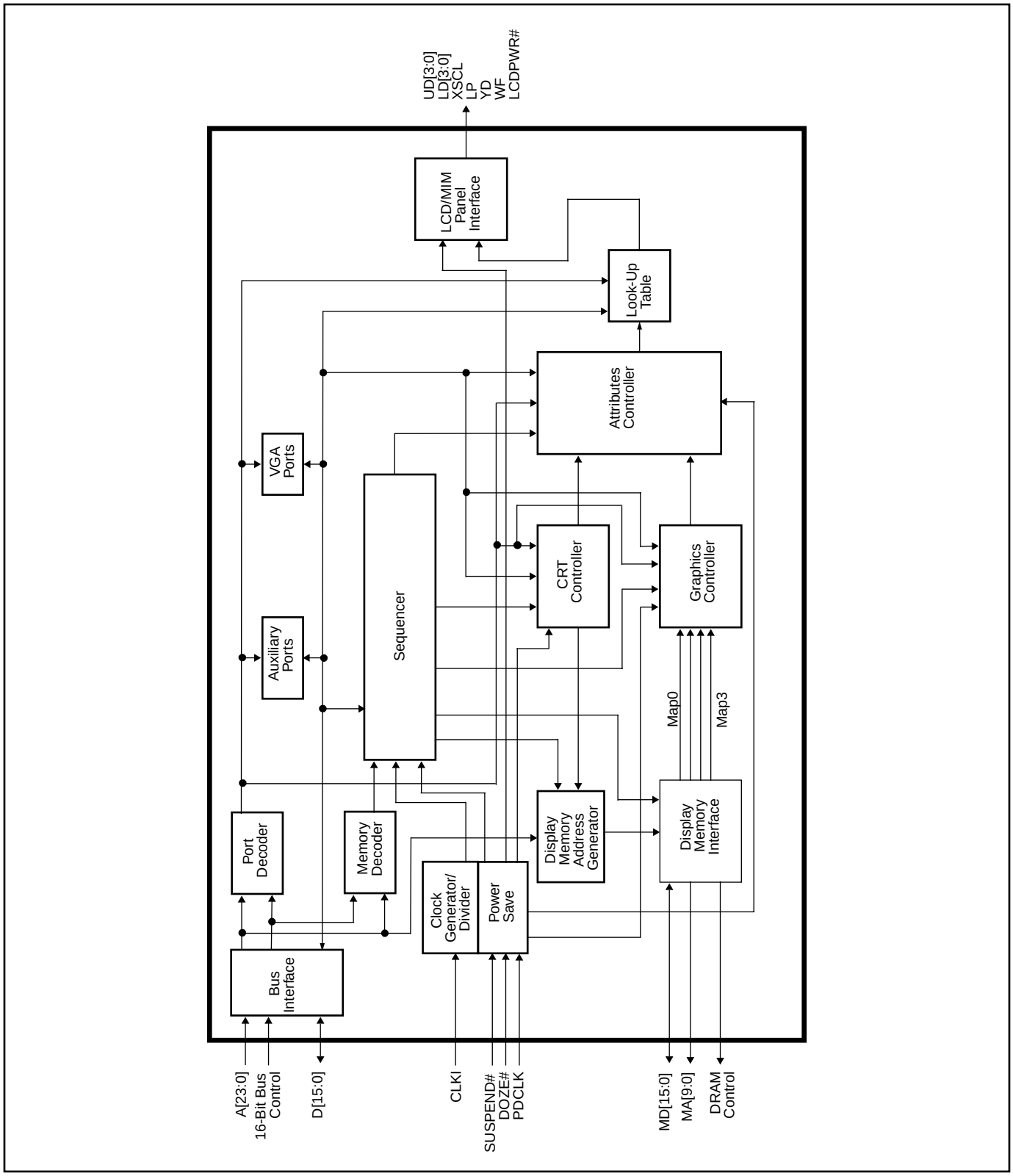
■ SUPPORTED RESOLUTIONS

Mode No.	Mode Type	Font	Characters	Resolution	Displayed Pixels	Gray Shades	Memory Segment
0	T	8 x 8	40 x 25	320 x 200	640x400	16	B800
0+	T	8 x 14	40 x 25	320 x 350	640x350	16	B800
0++	T	8 x 16	40 x 25	320 x 400	640x400	16	B800
1	T	8 x 8	40 x 25	320 x 200	640x400	16	B800
1+	T	8 x 14	40 x 25	320 x 350	640x350	16	B800
1++	T	8 x 16	40 x 25	320 x 400	640x400	16	B800
2	T	8 x 8	80 x 25	640 x 200	640x400	16	B800
2+	T	8 x 14	80 x 25	640 x 350	640x350	16	B800
2++	T	8 x 16	80 x 25	640 x 400	640x400	16	B800
3	T	8 x 8	80 x 25	640 x 200	640x400	16	B800
3+	T	8 x 14	80 x 25	640 x 350	640x350	16	B800
3++	T	8 x 16	80 x 25	640 x 400	640x400	16	B800
4	G	N/A	N/A	320 x 200	640x400	4	B800
5	G	N/A	N/A	320 x 200	640x400	4	B800
6	G	N/A	N/A	640 x 200	640x400	2	B800
7	T	8 x 14	80 x 25	640 x 350	640x350	2	B000
7+	T	8 x 16	80 x 25	640 x 400	640x400	2	B000
0D	G	N/A	N/A	320 x 200	640x400	16	A000
0E	G	N/A	N/A	640 x 200	640x400	16	A000
0F	G	N/A	N/A	640 x 350	640x350	2	A000
10	G	N/A	N/A	640 x 350	640x350	16	A000
11	G	N/A	N/A	640 x 480	640x480	2	A000
12	G	N/A	N/A	640 x 480	640x480	16	A000

■ SUPPORTED LCD INTERFACES

8-Bit Interface				4-Bit Interface	
Dual Panel		Single Panel		Single Panel	
Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical
1 to 640	241 to 480	1 to 640	1 to 480	1 to 640	1 to 480

■ FUNCTIONAL BLOCK DIAGRAM



■ FUNCTIONAL BLOCK DESCRIPTION

Bus Interface

The Bus Interface is a bridge by which the chip communicates with the CPU bus. It supports a 16-bit ISA bus.

Port Decoder

The Port Decoder decodes CPU-bus I/O cycles to provide enable and write strobes for the on-chip I/O registers.

Memory Decoder

The Memory Decoder monitors the CPU-bus activity and decodes cycles for the display DRAM. It supplies memory access control signals to the Sequencer.

Display Memory Address Generator

The Address Generator takes the display and refresh addresses from the CRT Controller and converts them into RAS and CAS addresses for the display DRAM, and multiplexes these display accesses with CPU memory accesses.

Sequencer

The Sequencer generates internal signals to synchronize the operation of the chip as well as the signals to control the timing of the display DRAM. The Sequencer also arbitrates between CPU and video display accesses to the DRAM. It contains registers that allows selection of character font set, control the structure of the video memory and allow write masking of the individual plane of memory.

Display Memory Interface

The Display Memory Interface is a bridge by which the chip communicates with the DRAM. It contains buffers that are used to store recently fetched DRAM data.

Power Save

The Power Save block contains the logic to implement three software and hardware controlled power down modes.

Clock Generator/Divider

The Clock Generation contains clock dividing circuits used by the Power Save block.

Auxiliary Ports

The Auxiliary Ports are I/O registers used to control functions of the chip beyond the basic VGA register set. Registers are included for controlling the LCD interface circuits as well as the power save modes.

CRT Controller

The CRT Controller generates the horizontal and vertical synchronization signals for the single panel or dual panel LCD display and character and/or pixel addresses for display data from DRAM.

Graphics Controller

The Graphics Controller supplies display memory data to the Attributes Controller during display time and provides data translation between the CPU bus and the display memory during CPU read or write access cycles.

VGA Ports

The VGA Ports contain the Miscellaneous Output Status register and the Video Subsystem Enable register used in VGA mode.

Attributes Controller

The Attributes Controller takes in pixel and attribute information from the Graphics Controller and display DRAM and formats the data into pixel information which then passes through the lookup table. It also controls display character attributes such as blink, underline and horizontal pixel panning.

Look-Up Table

The Lookup Table consists of a memory array of 64 locations of 4 bits each and hardware to convert VGA palette writes to gray-scale values.

LCD/MIM Panel Interface

The LCD Interface block converts the display video data from the Lookup Table into LCD display data. It also generates control signals necessary to drive single or dual-panel LCD panels and MIM panels. For monochrome LCD panels, the LCD interface block generates a maximum 16 gray shades through frame rate modulation and dithering techniques.

■ DC SPECIFICATIONS

Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
V_{DD}	Supply Voltage	V_{SS}-0.3 to +7.0	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_{OPR}	Operating Temperature	0 to +70	°C
T_{STG}	Storage Temperature	-65 to +150	°C
T_{SOL}	Soldering Temperature/Time	260 for 10sec max at lead	°C

Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
HV_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	3.6	V
LV_{DD}	Supply Voltage	V_{SS} = 0 V	2.25	2.5	3.6	V
V_{IN}	Input Voltage	V_{SS}	V_{SS}	--	V_{DD}	V
T_{OPR}	Operating Temperature		0	25	70	˚C
I_{OPR}	Average Power Consumption	$V_{DD\ Core}$ = 2.5 V $V_{DD\ IO}$ = 3.3 V		20		mA
IP_{D1}	Doze Mode 1	$V_{DD\ Core}$ = 2.5 V $V_{DD\ IO}$ = 3.3 V		10		mA
IP_{D2}	Doze Mode 2	$V_{DD\ Core}$ = 2.5 V $V_{DD\ IO}$ = 3.3 V	5	5		mA
IP_{SUS}	Suspend	$V_{DD\ Core}$ = 2.5 V $V_{DD\ IO}$ = 3.3 V		0.2		mA

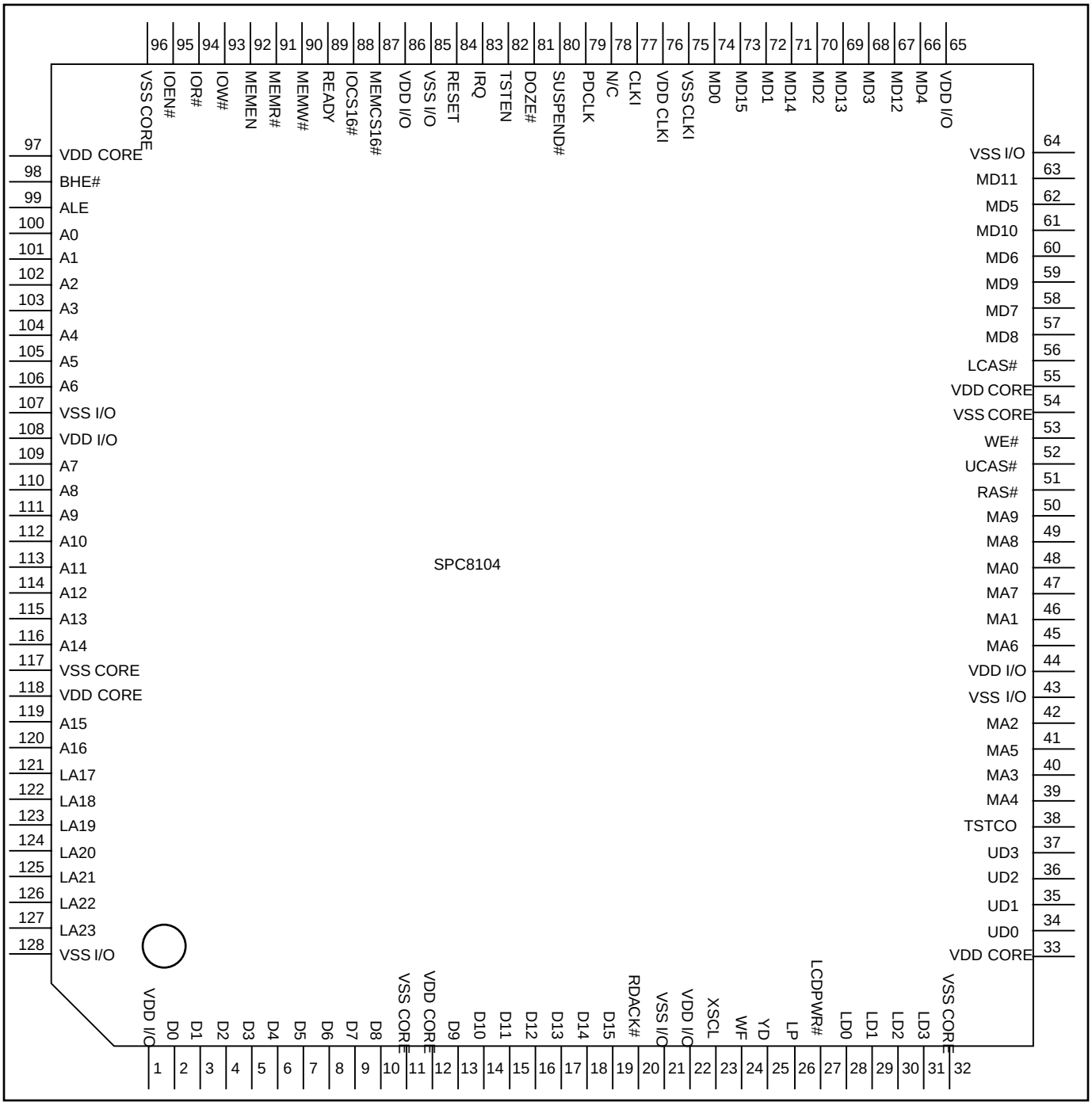
Input Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage	V_{DD} = MIN			0.8	V
V_{IH}	High Level Input Voltage	V_{DD} = MAX	2.0			V
V_{T+}	Positive-going Threshold (CMOS Schmitt inputs)	V_{DD} = 3.3 V			2.4	V
V_{T-}	Negative-going Threshold (CMOS Schmitt inputs)	V_{DD} = 3.3 V	0.6			V
V_H	Hysteresis Voltage (CMOS Schmitt inputs)	V_{DD} = 3.3 V	0.1			V
I_{IZ}	Input Leakage Current	V_{DD} = MAX V_{IH} = V_{DD} V_{IL} = V_{SS}	-1		1	µA
C_{IN}	Input Pin Capacitance			8		pF
R_{PU}	Pull Up Resistance	V_{DD} = 3.3 V		90		kΩ
R_{PD}	Pull Down Resistance	V_{DD} = 3.3 V		90		kΩ

Output Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{OL1}	Low Level Output Current	V_{OL}=V_{SS}+0.4V TS1, TS1U, CO1	3.0			mA
I_{OH1}	High Level Output Current	V_{OH}=V_{DD}-0.4V TS1, TS1U, CO1	-3.0			mA
I_{OL2}	Low Level Output Current	V_{OL}=V_{SS}+0.4V TS2, CO2	6.0			mA
I_{OH2}	High Level Output Current	V_{OH}=V_{DD}-0.4V TS2, CO2	-6.0			mA
I_{OZ}	Output Leakage Current	V_{OH}=V_{DD} or V_{OL}=V_{SS}	-1		1	µA
C_{OUT}	Output Pin Capacitance			8		pF

■ SPC8104 PIN OUTS SOURCE: 8110_PINOUT_08.CAN



■ PIN DESCRIPTION

Key

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

CPU Interface

Pin Name	Type	Pin #	Description
A[0:16], LA[17:23]	I	100~106, 109~116, 119~127	CPU bus unlatched address inputs. For an 8-bit CPU interface configuration, LA[20:23] are ignored and LA[17:19] should be connected to the latched CPU address SA[17:19]. In Suspend Mode, the Address inputs are internally masked off.
D[0:15]	I/O	2~10, 13~19	16 bit ISA-Bus data bus. These lines are driven by the chip only during read cycles, and are in a hi-Z state at all other times. In Suspend Mode, these inputs are internally masked off.
ALE	I	99	ISA Bus Address Latch Enable. ALE is ignored for an 8-bit CPU interface configuration. In Suspend Mode the this input is disabled.
MEMEN	I	92	ISA Bus Memory Enable. This signal should be connected to the -REFRESH signal on the ISA bus. When this signal is low (e.g. during a system memory refresh cycle), memory address decoding is disabled.
IOR#	I	94	ISA Bus I/O Read Strobe. In Suspend Mode the this input is disabled.
IOW#	I	93	ISA Bus I/O Write Strobe. In Suspend Mode the this input is disabled.
MEMR#	I	91	ISA Bus Memory Read Strobe. In Suspend Mode the this input is disabled.
MEMW#	I	90	ISA Bus Memory Write Strobe. In Suspend Mode the this input is disabled.
IOEN#	I	95	ISA Bus I/O Enable. This input should be connected to the ISA bus AEN signal. When this signal is high, I/O address decoding is disabled. In Suspend Mode the this input is disabled.
READY	O	89	ISA Bus READY signal. This output is driven low to force the CPU to insert wait states during memory cycles. READY is released to high-Z after a transfer is complete.
RESET	I	84	The active high Reset signal from the CPU clears all internal registers and forces all signals to their inactive state. During Suspend Mode the RESET input is ignored.
IRQ	O	83	ISA Bus Vertical Interrupt. When enabled, a Vertical Retrace Interrupt will cause this signal to be driven from a logic 0 state to a logic 1 (rising-edge triggered interrupt). Once set, this interrupt must be cleared by a bit in the CRTC registers. A control bit in the Auxiliary Registers allows this output to be optionally disabled (tri-stated).
MEMCS16#	O	87	ISA Bus Memory Chip Select 16. Address inputs LA[23:17] are decoded to drive this output low when a valid memory address (AXXXXh, BXXXXh) appears on the bus.
IOCS16#	O	88	ISA Bus I/O chip Select 16. Address inputs A[15:0] and IOEN# are decoded to drive this output low when a valid SPC8104 I/O register address appears on the bus. Note that I/O addresses 3C6h-3C9h does not result in IOCS16# being driven low (i.e. internal LUT register reads and writes are 8 bit cycles).
BHE#	I	98	ISA Bus Byte High Enable. In Suspend Mode the this input is disabled.
RDACK#	O	20	Read Acknowledge. This pin goes low during valid I/O or memory reads to the chip.

Frame Buffer Memory Interface

Pin Name	Type	Pin #	Description
MA[0:9]	O	48, 46, 42, 40, 39, 41, 45, 47, 49, 50	Multiplexed row/column address bits for video display memory.
MD[0:4] MD[7:15]	I/O	74, 72, 70, 68, 66, 58, 57, 59, 61, 63, 67, 69, 71, 73	Data bits for video display memory. The output drivers of these pins are placed into a high-impedance state when RESET is high. On the falling edge of RESET, the values on MD[3:0] are latched into a read-only Auxiliary Register and are available to be read as configuration inputs. Also, the values on MD[5:6] are used to configure other various hardware options - see Section 5.8, Summary of Configuration Options, for details. Note that there are internal pullup resistors on the inputs of these pins except MD[5:6].
MD[5:6]	I/O	62, 60	
RAS#	O	51	DRAM Row Address Strobe.
LCAS# (LWE#)	O	56	DRAM Column Address Strobe for low byte (LCAS#), or Write Enable Strobe for low byte (LWE#), as determined by logic value on MD[6] during RESET (see pin mapping table).
UCAS# (CAS#)	O	52	DRAM Column Address Strobe for high byte (UCAS#), or single Column Address Strobe (CAS#), as determined by logic value on MD[6] during RESET (see pin mapping table).
WE# (UWE#)	O	53	DRAM Write Enable Strobe (WE#), or Write Enable Strobe for high byte (UWE#), as determined by logic value on MD[6] during RESET (see pin mapping table).

Clock Inputs

Pin Name	Type	Pin #	Description
CLKI	I	77	This is the clock source input and should be connected to an external oscillator.

Power Supply

Pin Name	Type	Pin #	Description
V_{DD} CORE	P	12, 33, 55, 97, 118	V_{DD} supply for core logic.
V_{DD} I/O	P	1, 22, 44, 65, 86, 108	V_{DD} supply for I/O pins.
V_{DD} CLKI	P	76	V_{DD} supply for CLKI pin.
V_{SS} CORE	P	11, 32, 54, 96, 117	V_{SS} supply for core logic.
V_{SS} I/O	P	21, 43, 64, 85, 107, 128	V_{SS} supply for I/O pins.
V_{SS} CLKI	P	75	V_{SS} supply for CLKI pin.

Test Function

Pin Name	Type	Pin #	Drv	Description
TSTCO	I	38	CD	This pin enables the chip's test mode for the core logic. This pin must always be unconnected or tied to ground.
TSTEN	I	82	CD	This pin enables the chip's test mode for the I/O cells. This pin must always be unconnected or tied to ground.
N/C		78		No Connection

LCD Panel Interface

Pin Name	Type	Pin #	Description
YD	O	25	Vertical Scanning Start Pulse output. A logic 1 on this signal, sampled by the LCD/MIM panel module on the falling edge of LP, is used by the panel row drivers (Y drivers) to indicate the start of the vertical frame.
LP	O	26	Latch Pulse output. The falling edge of this signal is used to latch a row of display data in the LCD/MIM panel module's column driver shift registers and to turn on the row driver (Y driver) for that line.
XSCL	O	23	Shift Clock for LCD panel data or Pixel Clock for MIM panel data. Display data is clocked out of the chip on the rising edge of this signal, to be shifted into the LCD/MIM panel module column drivers (X drivers) on each falling edge.
UD[0:3]	O	34~37	Upper panel display data for dual LCD panel mode. For single LCD 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 LCD panel mode, these bits are the 4 bits of output data to the panel. For 4-bit MIM panel mode, these bits are driven 0.
LD[0:3]	O	28~31	Lower panel display data for dual LCD panel mode. For 8-bit single LCD 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 LCD panels, these bits are driven 0. For 4-bit MIM panel mode, these are the 4 bits of output data to the panel.
LCDPWR#	O	27	LCD power control. In normal operation this signal is driven low to enable an external LCD power supply. This signal is driven high when the chip is put into any power save mode, or if the Sequencer is in a reset state. It can be used externally to turn off the panel supply voltage and backlight. After a RESET, this signal is held high until the CRTC is programmed and running.
WF	O	24	LCD Panel Backplane Bias Signal or MIM Panel Data Enable Signal. In LCD panel mode, the WF signal toggles once per vertical frame period. In MIM panel mode, this signal goes high whenever display data are valid.

Power Save Mode Control

Pin Name	Type	Pin #	Description
PDCLK	I	79	Power Down Clock. This input may be used to provide a low frequency clock for generating DRAM refresh in Suspend mode, as an optional alternative to using the pixel clock or MEMEN input as the DRAM refresh clock source. This clock input should be driven by a 32 kHz 50% duty cycle clock. The PDCLK input is used to directly generate the RAS and CAS pulses in Suspend mode.
SUSPEND#	I	80	A low level on this pin puts the chip into the hardware Suspend mode. The SUSPEND# signal overrides any software initiated power save modes as well as the DOZE# input pin, and disables the CPU bus interface inputs. CPU Address and Data inputs are masked when this signal is low. When in Suspend Mode the UD[3:0], LD[3:0], XSCL, LP, YD and WF signals are driven into a low state (or optionally, a high impedance state) and the LCDPWR# signal is driven high.
DOZE#	I	81	A low level on this pin puts the chip into Doze mode. The function of the Doze mode is determined by the Doze Mode Select bits in AUX[03]. This pin is ignored if the SUSPEND# input pin is asserted.

Configuration Options

Pin Name	value on this pin at falling edge of RESET is used to configure: (1/0)
MD[3:0]	values latched into read-only AUX[0C] bits 7-4 for software use
MD[5]	LCD signals' state in Suspend mode: Low (1), or Hi-Z (0)
MD[6]	2 CAS, 1 WE type DRAM (1), or 1 CAS, 2 WE type DRAM (0)

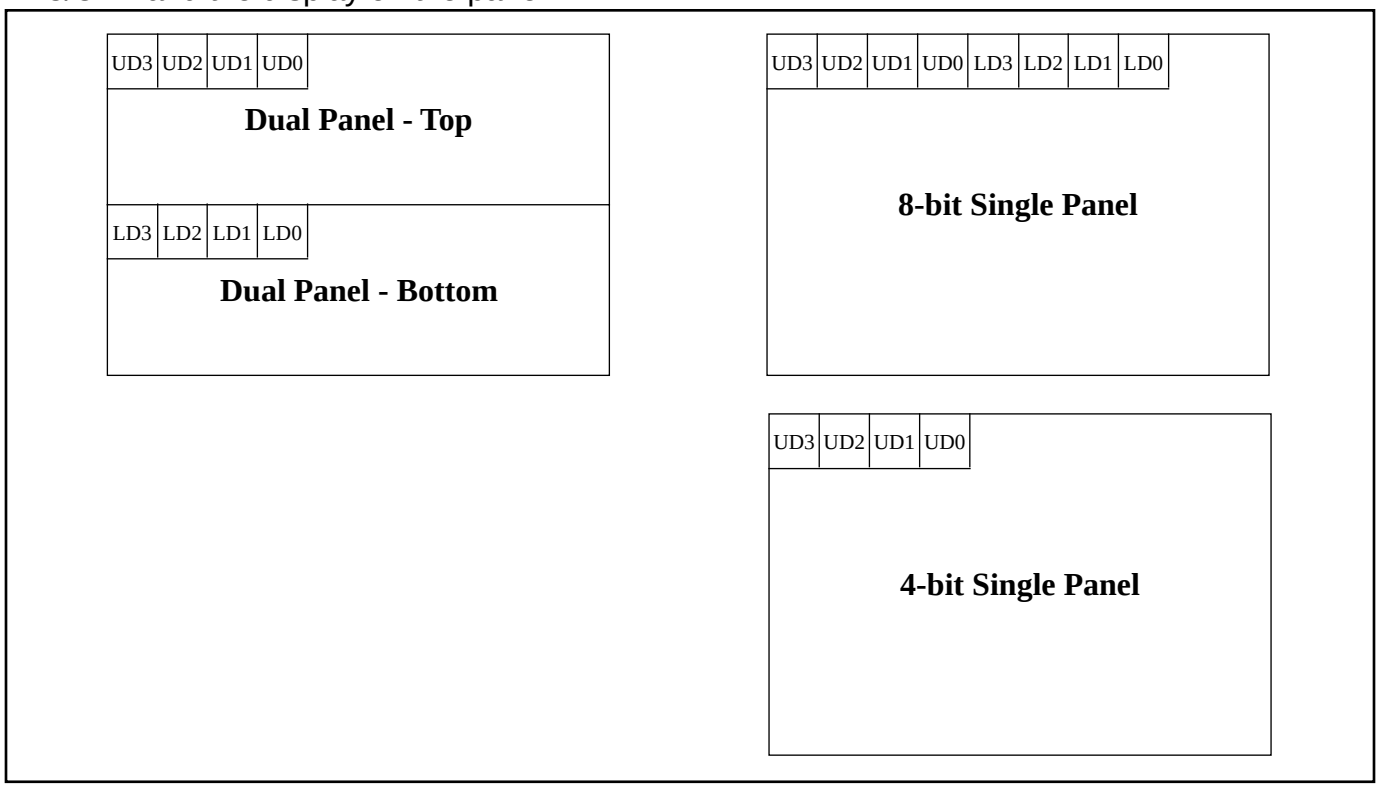
Multiple Function Pin Descriptions

Pin Name	Function	MD Line Status	Functional Description
LCAS#, LWE#	LCAS#	MD[6] = 1	DRAM Column Address Strobe (Low Byte)
	LWE#	MD[6] = 0	DRAM Write Enable Strobe
UCAS#, CAS#	UCAS#	MD[6] = 1	DRAM Column Address Strobe (High Byte)
	CAS#	MD[6] = 0	DRAM Column Address Strobe
WE#, UWE#	WE#	MD[6] = 1	DRAM Write Strobe
	UWE#	MD[6] = 0	DRAM Write Strobe (High Byte)

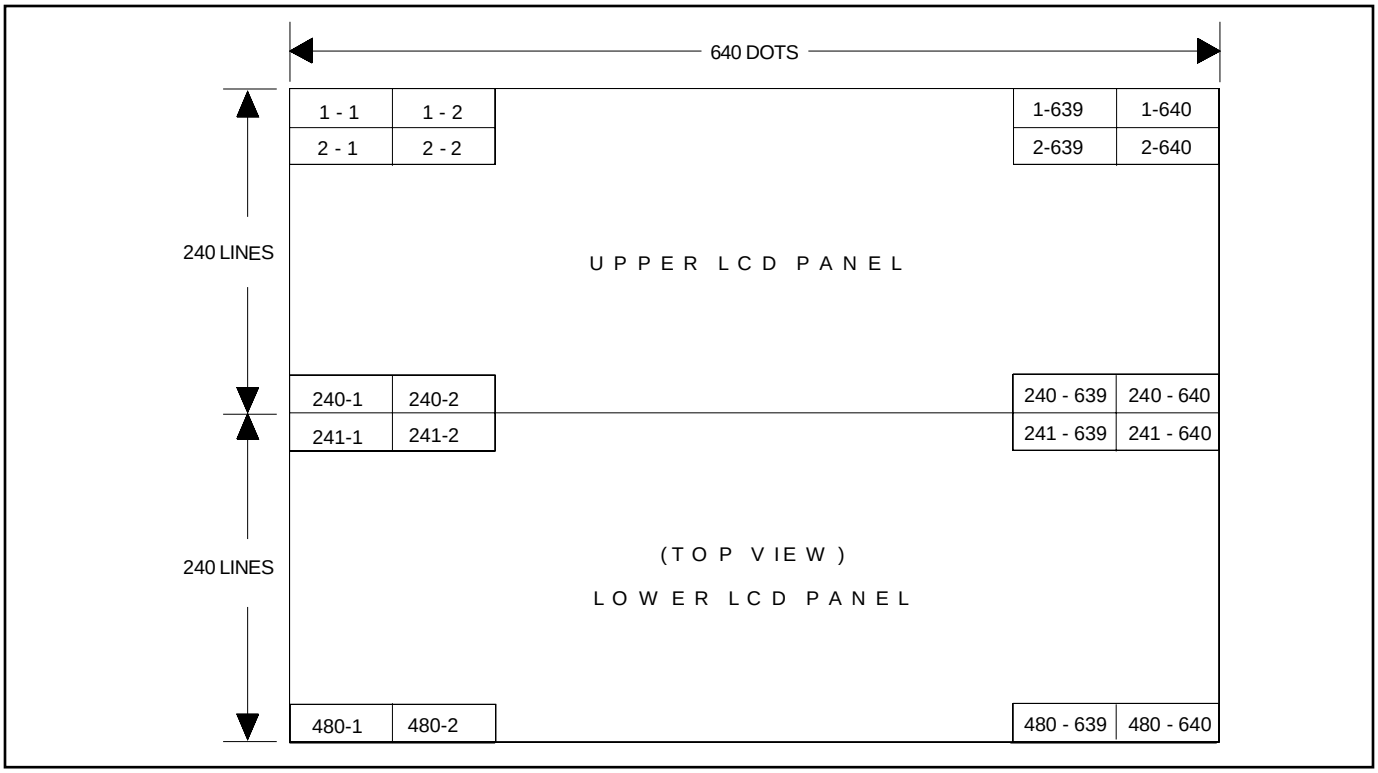
Mixed Voltage Configurations

Core V_{DD}	I/O V_{DD}	
	2.5 V	3.3 V
2.5 V	No	Yes
3.3 V	No	Yes

Illustrated below are the display data output which are output from the UD0 to UD3, LD0/UD4 to LD3/UD7 and the display on the panel:

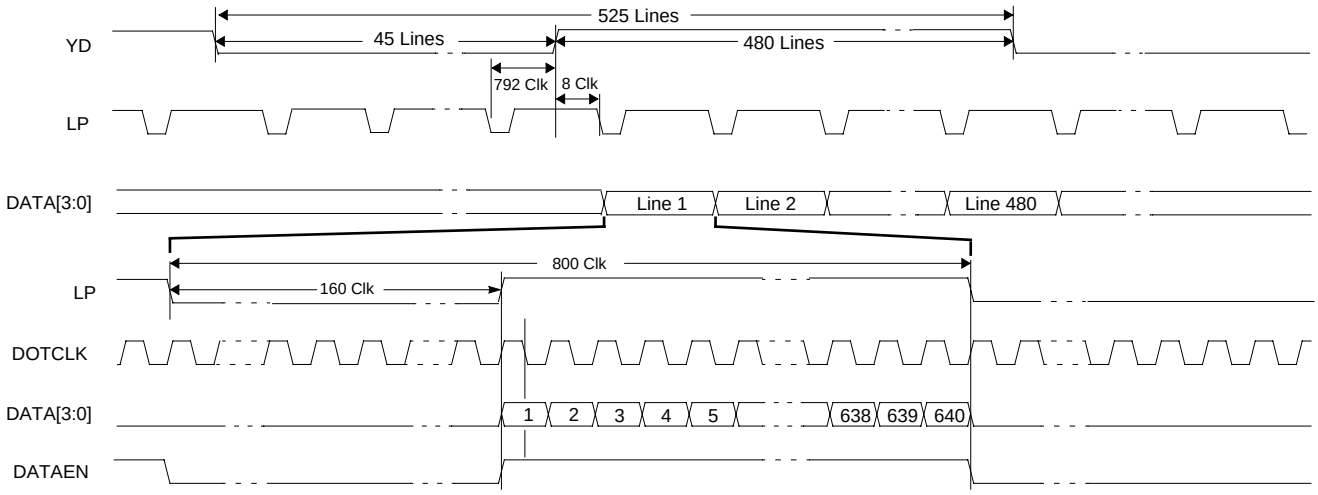


■ LCD PANEL PIXELS



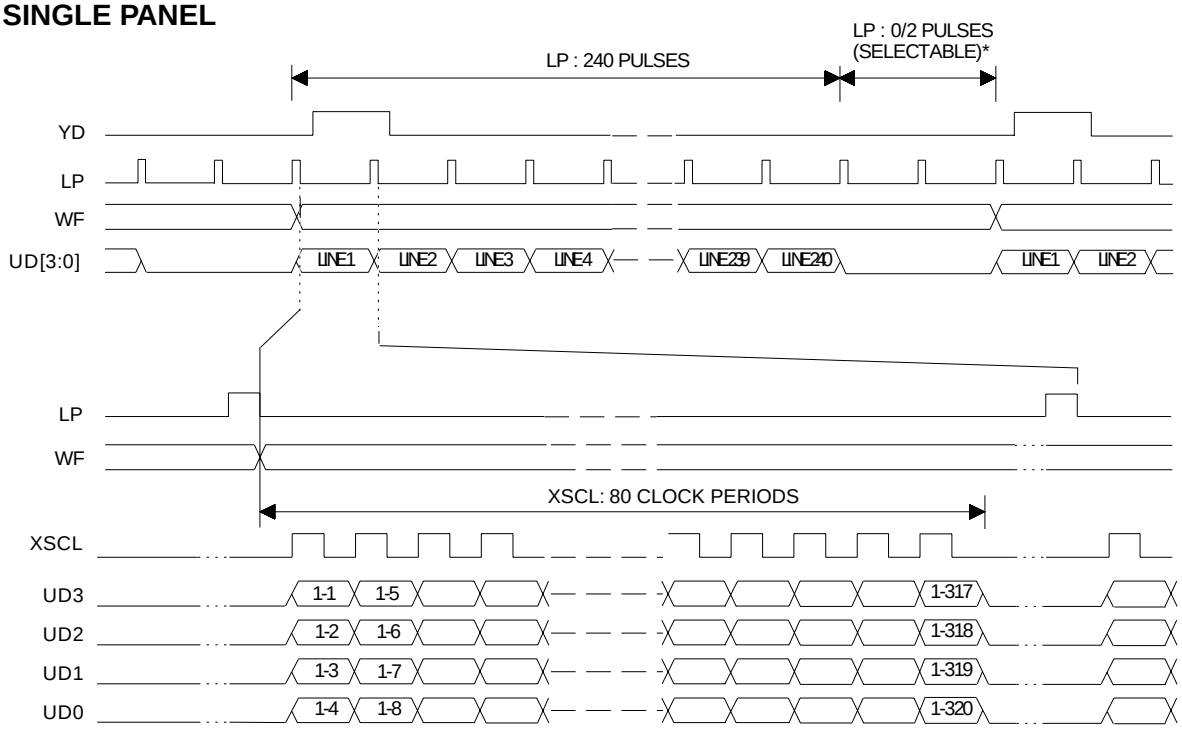
■ MIM PANEL INTERFACE

MIM PANEL



■ MONOCHROME PASSIVE STN LCD PANEL INTERFACE

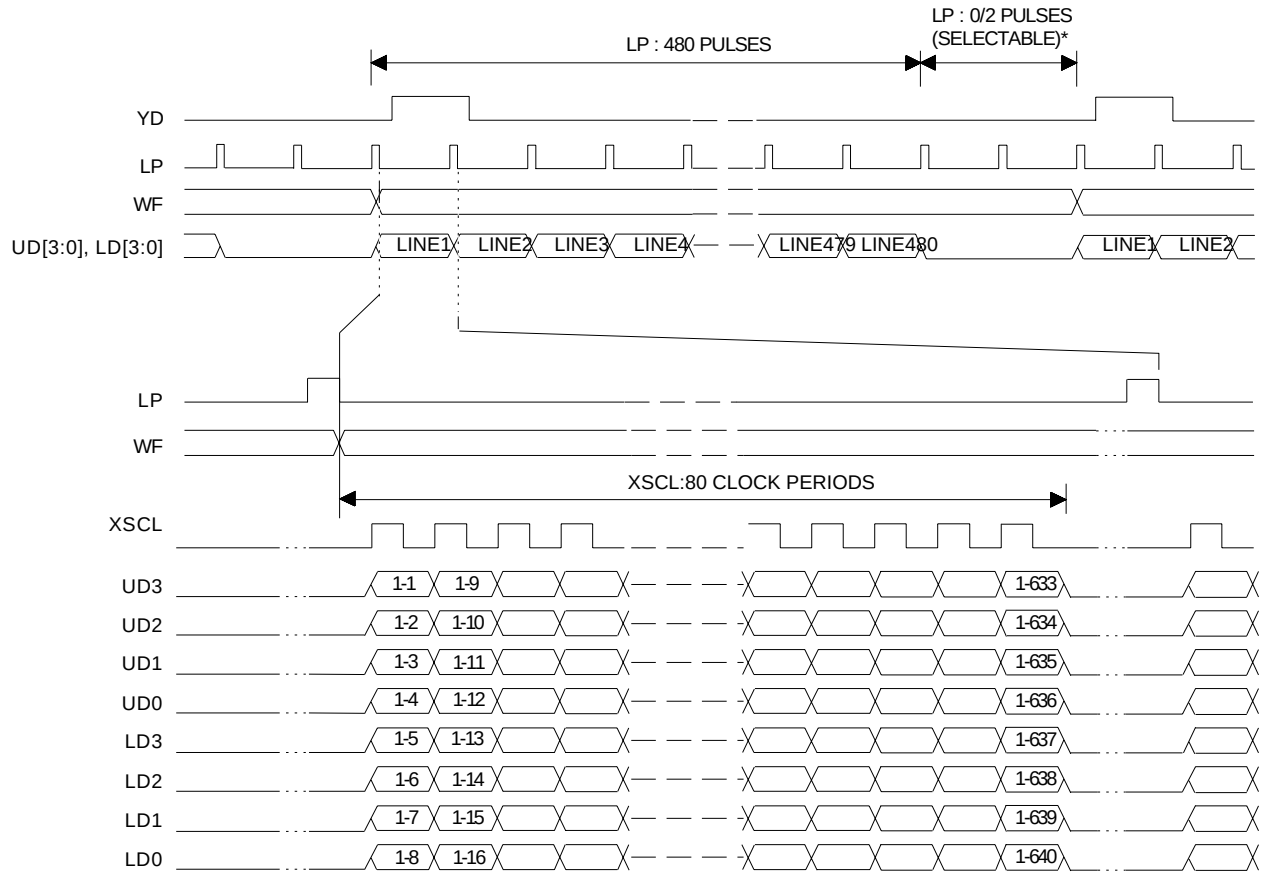
4-BIT SINGLE PANEL



* diagram drawn with 2 LP vertical blank period
Example timing for a 320 x 240 panel

■ MONOCHROME PASSIVE STN LCD PANEL INTERFACE

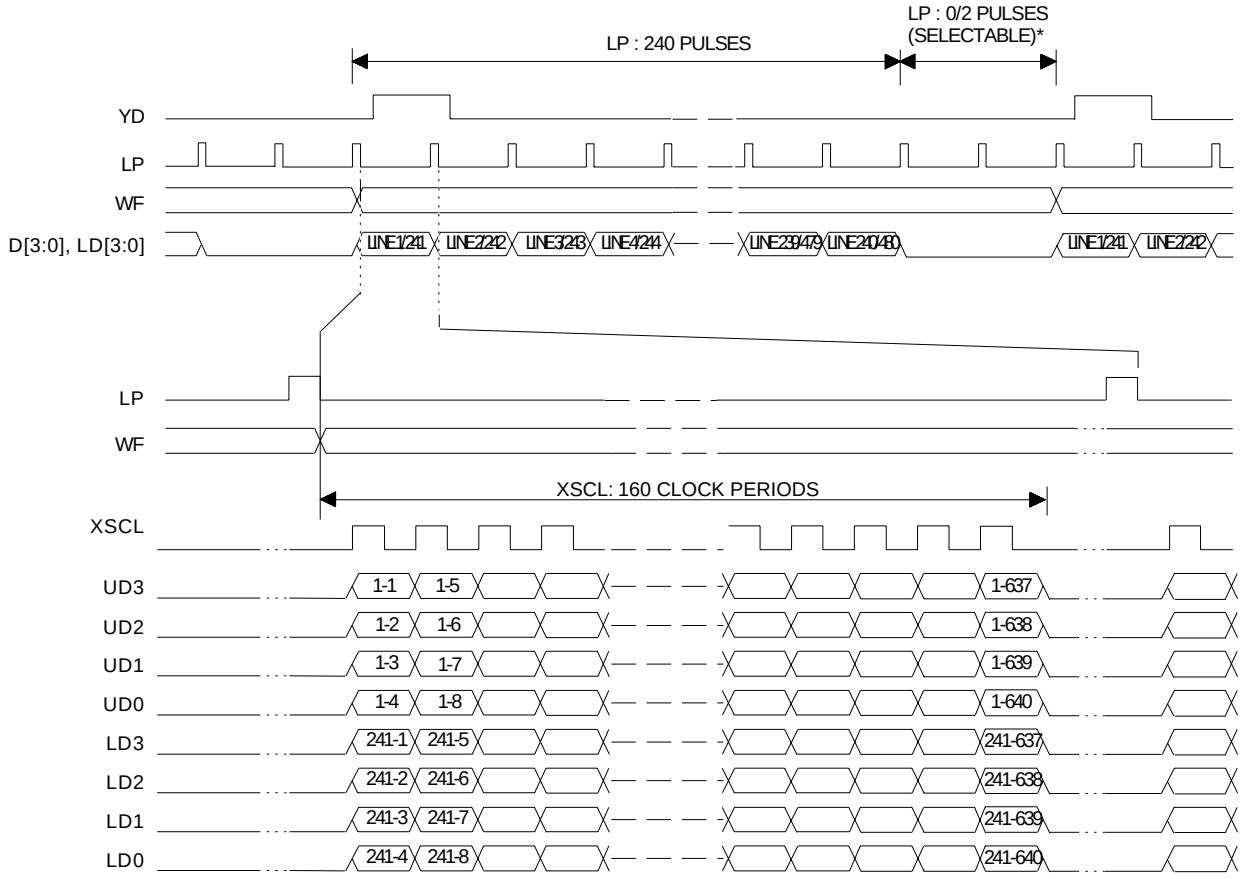
8-BIT SINGLE PANEL



* diagram drawn with 2 LP vertical blank period

■ MONOCHROME PASSIVE STN LCD PANEL INTERFACE

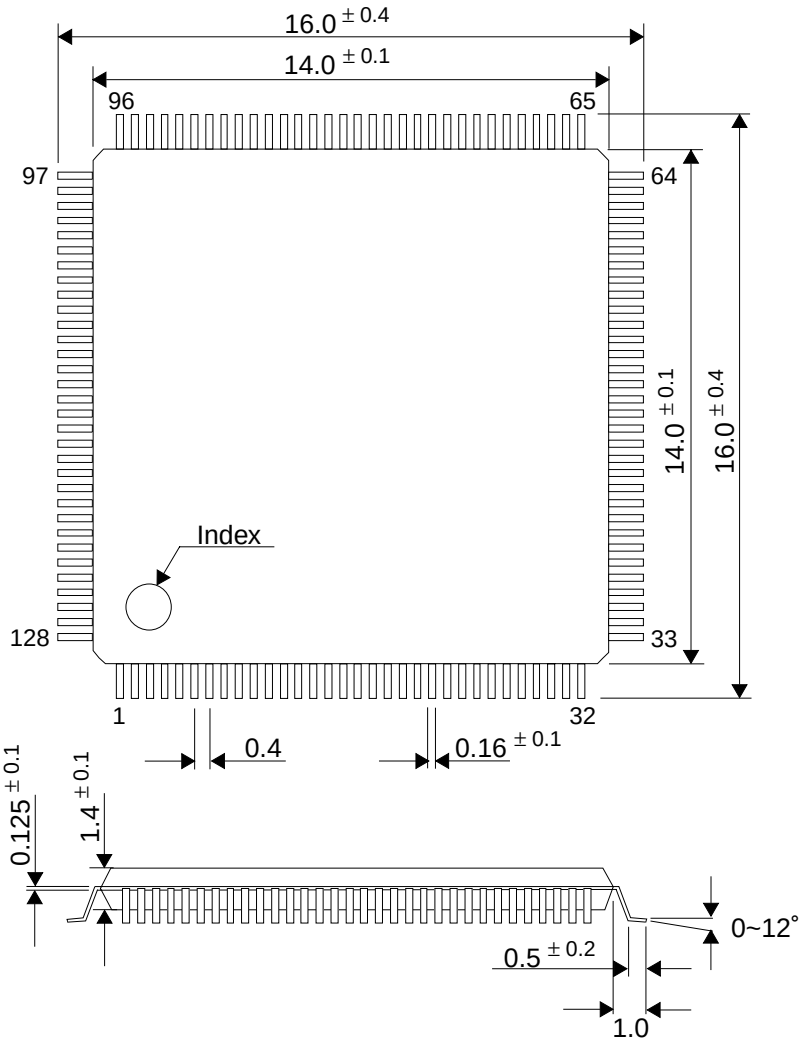
8-BIT DUAL PANEL

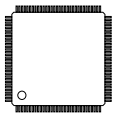


* diagram drawn with 2 LP vertical blank period

■ PACKAGE DIMENSIONS

QFP15 - 128 pin **Unit: mm**





Actual Size

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

Hardware Functional Specification

Drawing Office No. X15-SP-001-08.1

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY BLANK

1.0 INTRODUCTION

1.1 Scope

This is the Hardware Functional Specification for the SPC8104F0A LCD VGA Controller Chip. Included in this documentation are timing diagrams, AC and DC characteristics, register descriptions, and power management descriptions. This document is intendended for two audiences; Video Subsystem Designers and Software Developers.

2.0 FEATURES

2.1 Technology

- low power CMOS
- 2.5/3.3 volt core and 3.3 volt I/O operation
- 128 pin QFP15 surface mount package

2.2 System

- 16 bit ISA CPU data bus interface bus.
- external crystal oscillator supported only, maximum 28 MHz frequency.
- interfaces to a single 256 Kx16 DRAM 80ns DRAM (1024x256x16 or 512x512x16).
  access to full 512 Kbytes of video memory allowed.
- selectable DRAM interface configurations: 2 CAS/1 WE, or 1 CAS/2 WE.
- selectable 256 cycle/4 msec or 256 cycle/32 msec DRAM refresh rate, or low power self-refresh mode (for DRAMs supporting self-refresh).
- 32 kHz 50% duty cycle power down clock support during Suspend mode.
- three hardware or software initiated power-save modes.
- low power consumption.

2.3 Compatibility

- support for all standard VGA modes except mode 13h (does not support packed pixel modes).
- 3C3h video enable register supported.
- proprietary internal 64x4 gray scale lookup table provided.
- programmable hardware mapping of VGA palette-style writes to 16 level LCD gray scale values.
- vertical interrupt function on IRQ pin supported.
- VGA RAMDAC NOT supported.
- Shift 2/4, Count-by-2/4, HRTC/2 NOT supported.
- 9-dot Character clock NOT supported (no 9-dot character modes).

2.4 Display Support

- optimized for 640x480 single panel-single drive monochrome LCD displays, or 640x480 dual panel-dual drive monochrome LCD displays.
- flexible support of LCD panels of various sizes via programmable horizontal and vertical panel size configuration registers.
- supports 0-255 vertical non-display periods.
- 8-bit or 4-bit single panel interface.
- support 640x480 4-bit monochrome MIM panels.
- adjustable frame rate from 7/8 down to 1/2 of the programmed value.
- support the slower frame rate of small panels via on-chip selectable clock input divide circuitry or via direct low frequency clock input as low as 6 MHz.

3.0 OVERVIEW DESCRIPTION

The SPC8104 is a low power, mixed voltage video controller based on VGA architecture and optimized for driving a 640x480 LCD panel display. VGA standard mode functionality is supported using standard IBM VGA parameters. A proprietary 64 x 4 bit gray scale lookup table is provided to allow re-mapping of the 16 possible gray shades displayed on an LCD panel.

The target markets for this device are small, cost sensitive mixed 2.5V/3.3V sub-notebook computers, or other specialized consumer products where low cost, low power consumption, low component count, and the ability to run most VGA software on a 640x480 LCD panel display are the major design considerations. This chip is intended to operate mainly in planar graphics modes (e.g. mode 12h), and will display 16 levels of gray.

3.1 Typical System Block Diagram

The following figure shows a typical system implementation with SPC8104.

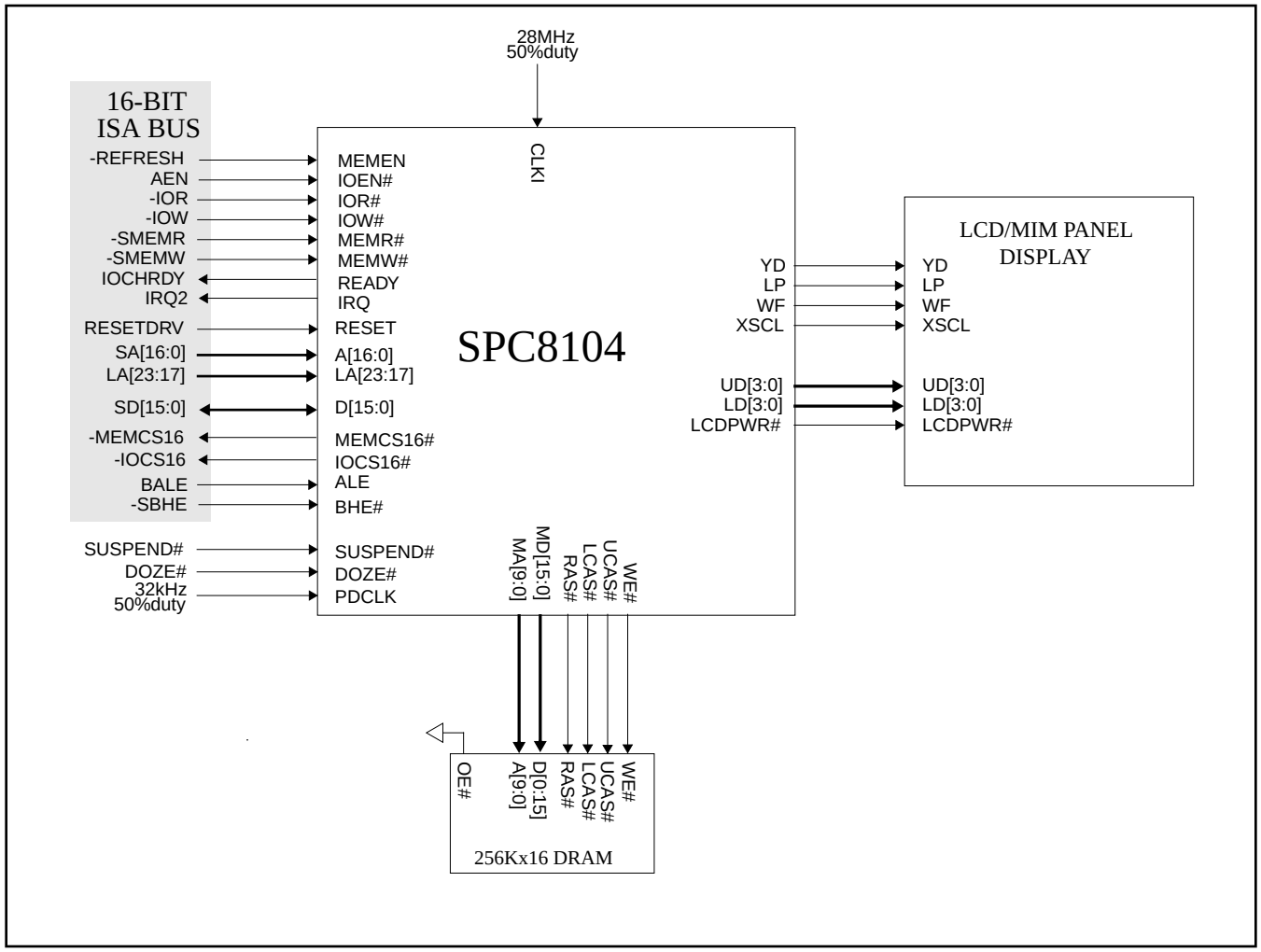


*Figure 1 : Typical System Block Diagram*Source: a000067.cdr

3.2 Internal Block Diagram

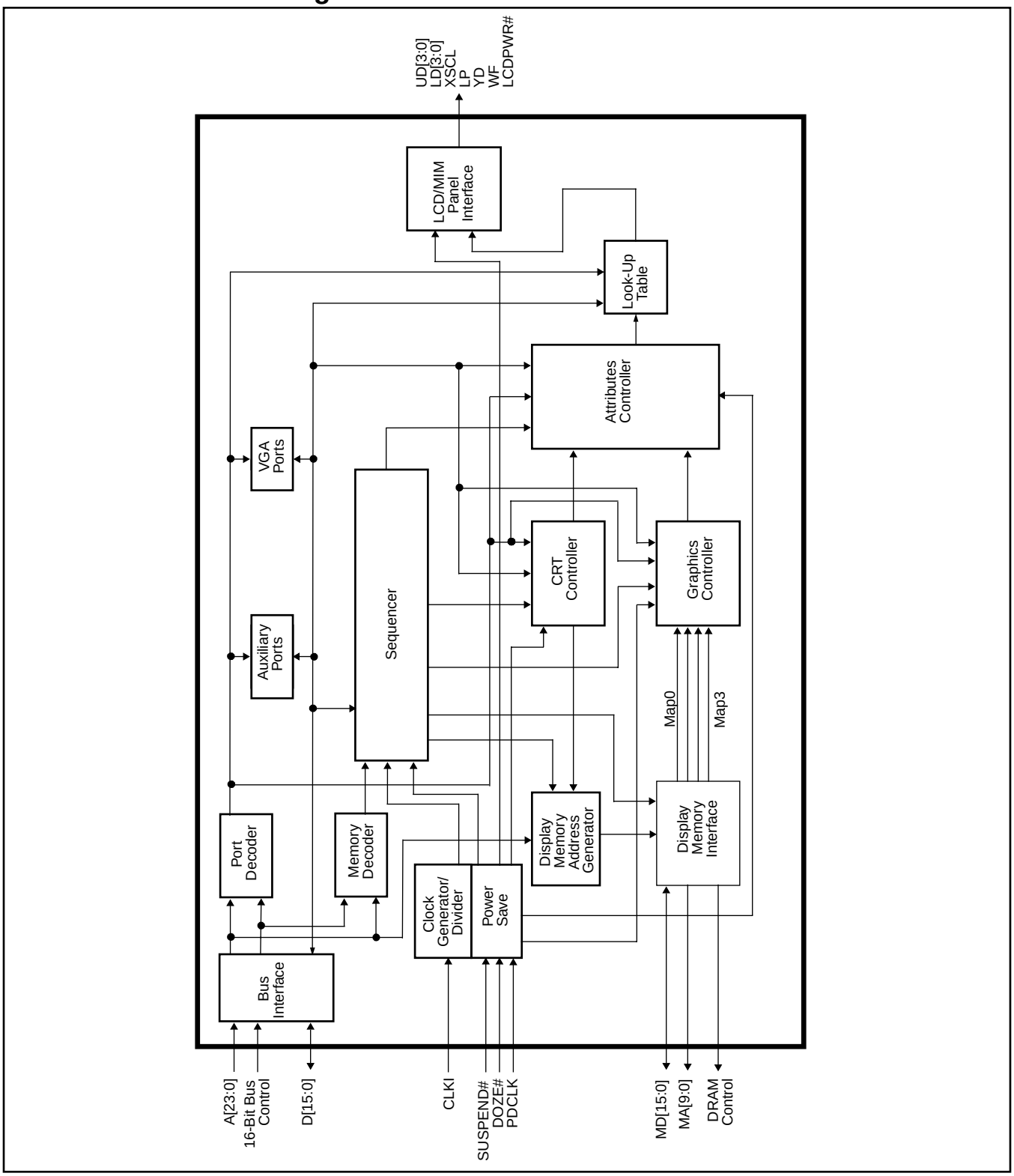


*Figure 2 : Internal Block Diagram*Source: a000068.cdr

4.0 PINOUT DIAGRAM

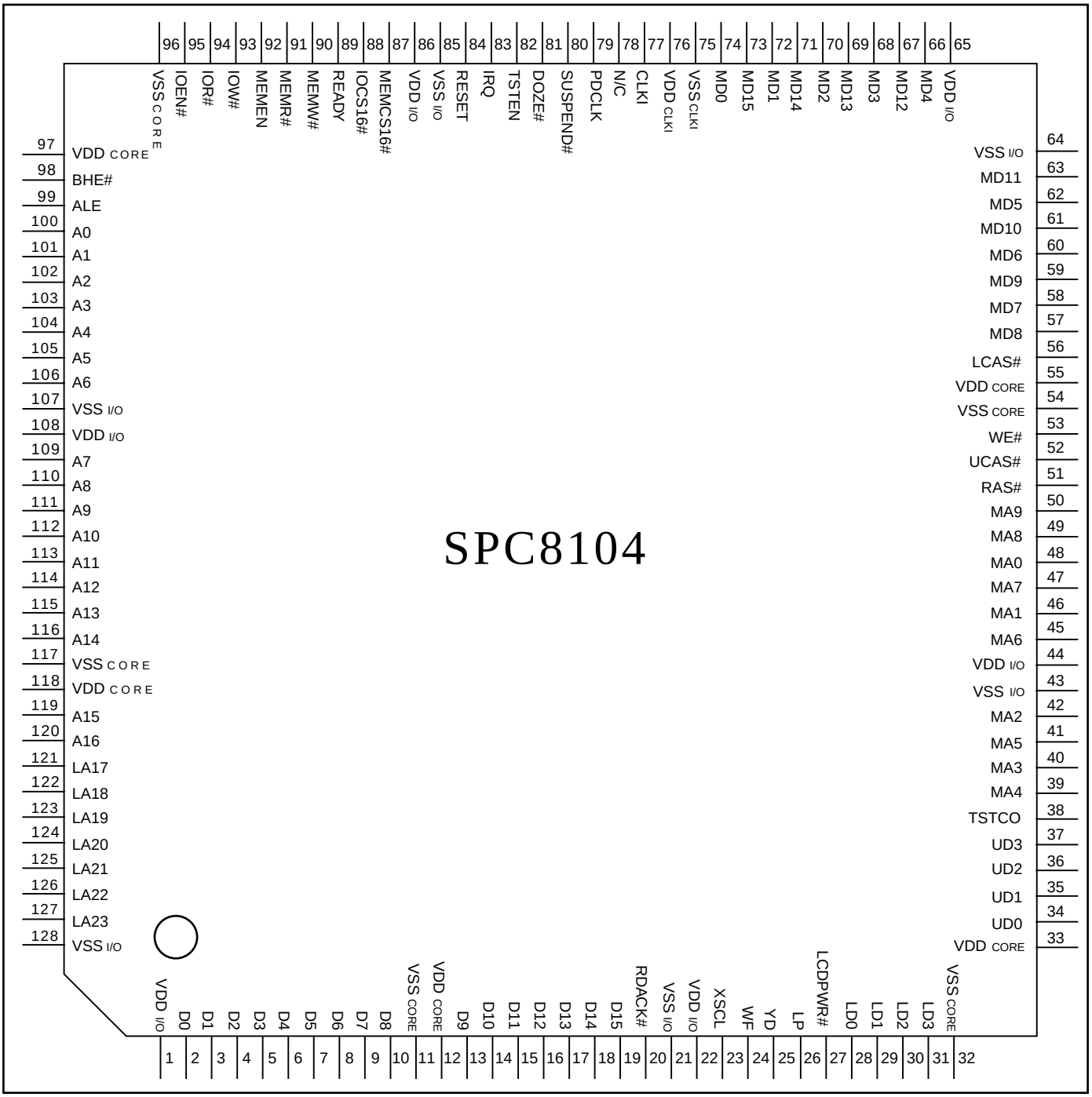


Figure 3 : Pinout Diagram[Source: a000069.cdr]

Package type: 128 pin surface mount QFP15.

5.0 PIN DESCRIPTIONS

Key

C = CMOS level input

CD = CMOS level input with 90 k0 kΩ pull down resistor.

CS = CMOS level input with hysteresis

COx = CMOS level output, x denotes output driver type - see DC characteristics for rating.

TSx = Tri-state CMOS level driver, x denotes driver type - see DC characteristics for rating.

TSxU = Tri-state CMOS level driver with 90 kΩ pull up resistor, x denotes driver type - see DC characteristics for rating.

All signal and pin names followed by # are active low (e.g., MEMR# would be read as $\overline{\text{MEMR}}$).

Table 0-1 CPU Interface

Pin Name	Type	Pin #	Drive	Description
A[0:16], LA[17:23]	I	100~106, 109~116, 119~127	C	CPU bus unlatched address inputs. For an 8-bit CPU interface configuration, LA[20:23] should be connected to ground and LA[17:19] should be connected to the latched CPU address SA[17:19]. In Hardware Suspend Mode, the Address inputs are internally masked off.
D[0:15]	I/O	2~10, 13~19	C/TS1	16 bit ISA-Bus data bus. These lines are driven by the chip only during read cycles, and are in a hi-Z state at all other times. In Hardware Suspend Mode, these inputs are internally masked off.
ALE	I	99	C	ISA Bus Address Latch Enable. ALE should be connected to V_{DD} I/O for an 8-bit CPU interface configuration.
MEMEN	I	92	CS	ISA Bus Memory Enable. This signal should be connected to the -REFRESH signal on the ISA bus. When this signal is low (e.g. during a system memory refresh cycle), memory address decoding is disabled.
IOR#	I	94	CS	ISA Bus I/O Read Strobe. In Hardware Suspend Mode the this input is disabled.
IOW#	I	93	CS	ISA Bus I/O Write Strobe. In Hardware Suspend Mode the this input is disabled.
MEMR#	I	91	CS	ISA Bus Memory Read Strobe. In Suspend Mode the this input is disabled.
MEMW#	I	90	CS	ISA Bus Memory Write Strobe. In Suspend Mode the this input is disabled.
IOEN#	I	95	CS	ISA Bus I/O Enable. This input should be connected to the ISA bus AEN signal. When this signal is high, I/O address decoding is disabled. In Hardware Suspend Mode the this input is disabled.
READY	O	89	TS2	ISA Bus READY signal. This output is driven low to force the CPU to insert wait states during memory cycles. READY is released to high-Z after a transfer is complete.

Table 0-1 CPU Interface

Pin Name	Type	Pin #	Drive	Description
RESET	I	84	CS	The active high Reset signal from the CPU clears all internal registers and forces all signals to their inactive state. During Hardware Suspend Mode the RESET input is ignored.
IRQ	O	83	TS1	ISA Bus Vertical Interrupt. When enabled, a Vertical Retrace Interrupt will cause this signal to be driven from a logic 0 state to a logic 1 (rising-edge triggered interrupt). Once set, this interrupt must be cleared by a bit in the CRTC registers. A control bit in the Auxiliary Registers allows this output to be optionally disabled (tri-stated).
MEMCS16#	O	87	TS2	ISA Bus Memory Chip Select 16. Address inputs LA[23:17] are decoded to drive this output low when a valid memory address (AXXXXh, BXXXXh) appears on the bus.
IOCS16#	O	88	TS2	ISA Bus I/O chip Select 16. Address inputs A[15:0] and IOEN# are decoded to drive this output low when a valid SPC8104 I/O register address appears on the bus. Note that I/O addresses 3C6h-3C9h does not result in IOCS16# being driven low (i.e. internal LUT register reads and writes are 8 bit cycles).
BHE#	I	98	C	ISA Bus Byte High Enable.
RDACK#	O	20	CO1	Read Acknowledge. This pin goes low during valid I/O or memory reads to the chip.

Table 0-2 Video Memory Interface

Pin Name	Type	Pin #	Drive	Description
MA[0:9]	O	48, 46, 42, 40, 39, 41, 45, 47, 49, 50	CO1	Multiplexed row/column address bits for video display memory.
MD[0:4] MD[7:15]	I/O	74, 72, 70, 68, 66, 58, 57, 59, 61, 63, 67, 69, 71, 73	C/ TS1U	Data bits for video display memory. The output drivers of these pins are placed into a high-impedance state when RESET is high. On the falling edge of RESET, the values on MD[3:0] are latched into a read-only Auxiliary Register and are available to be read as configuration inputs. Also, the values on MD[5:6] are used to configure other various hardware options - see section 5.4 on page 21, for details. Note that there are internal pullup resistors on the inputs of these pins except MD[5:6].
MD[5:6]	I/O	62, 60	C/TS1	
RAS#	O	51	CO2	DRAM Row Address Strobe.
LCAS# (LWE#)	O	56	CO2	DRAM Column Address Strobe for low byte (LCAS#), or Write Enable Strobe for low byte (LWE#), as determined by logic value on MD[6] during RESET (see pin mapping table).
UCAS# (CAS#)	O	52	CO2	DRAM Column Address Strobe for high byte (UCAS#), or single Column Address Strobe (CAS#), as determined by logic value on MD[6] during RESET (see pin mapping table).
WE# (UWE#)	O	53	CO2	DRAM Write Enable Strobe (WE#), or Write Enable Strobe for high byte (UWE#), as determined by logic value on MD[6] during RESET (see pin mapping table).

5.1 Video Memory Interface Options

The logic value at pin MD[6] during RESET is used to determine memory configuration (1 - 256Kx16 with “2 CAS with 1 WE” or “1 CAS with 2 WE” type DRAMs). The pinout diagram is labelled with the pin names for the default configuration of single 256Kx16 DRAM with 2 CAS and 1 WE signals. The MD[6] pin does not have an internal pullup resistor, so an external resistor is required for this default case. Refer to the following table for details.

Table 0-3 Video Memory Interface Options

MD[6]	# DRAM / size	CAS/WE Configuration
0	1 / 256Kx16	1 CAS, 2 WE
1	1 / 256Kx16	2 CAS, 1 WE

5.2 Pin mapping for various DRAM configurations

Non-italicized pin names are the default configuration and correspond to the names on the pinout diagram.

Table 0-4 DRAM Configuration Pin Mapping

Pin #	MD[6] = 1	MD[6] = 0
52	UCAS#	CAS#
53	WE#	UWE#
56	LCAS#	LWE#

5.3 Address Mapping for 256Kx16 DRAMs

Two addressing configurations of 256Kx16 DRAMs are supported by the SPC8104:

1024 x 256 x 16 10 row address bits X 8 column address bits
512 x 512 x 16 9 row address bits X 9 column address bits

The SPC8104 is designed to accommodate both types of 256Kx16 DRAMs directly without any special configuration options required. The full addressing space is available for either memory configuration. The only difference is that in the installation of the 10x8 type of DRAM, an extra row/column address output pin of the SPC8104 (MA[9]) must be connected to the DRAM. For the 9x9 DRAM, only 9 row/column address pins exist, so MA[9] from the SPC8104 is left unconnected.

The following tables summarize the mapping of the SPC8104's internal memory address bits a[17:0] to the multiplexed row/column address outputs MA[9:0] for the two types of DRAM.

1. For the 10x8 type DRAMs:

Table 0-5 10x8 DRAM Address Mapping

	MA[9]	MA[8]	MA[7:0]
row addr	a[17]	a[16]	a[15:8]
column addr	-	***a[17]***	a[7:0]

2. For the 9x9 type DRAMs:

Table 0-6 9x9 DRAM Address Mapping

	MA[9]	MA[8]	MA[7:0]
row addr	n/c	a[16]	a[15:8]
column addr	n/c	a[17]	a[7:0]

1. The address pin mapping for both types of DRAM are actually the same.

2. For the standard 256K byte VGA address space, a[17] is normally = 0, and for the upper 256K bytes, a[17] = 1.

3. The 10x8 type DRAM does not use the 9th column bit MA[8] - it is ignored.

4. a[17] is output on the 9th column bit MA[8], so the same physical addressing scheme works for both 10x8 and 9x9 type DRAMs. In the case of the 10x8 type DRAM, this 9th column address bit is not used by the DRAM (***bold-italicized*** in the above table).

5. The 9x9 configuration does not have a 10th row/column address bit.

Table 0-7 Clock Inputs

Pin Name	Type	Pin #	Drive	Description
CLKI	I	77	C	This is the clock source input and should be connected to an external oscillator.

Table 0-8 LCD Panel Interface

Pin Name	Type	Pin #	Drive	Description
YD	O	25	TS2	Vertical Scanning Start Pulse output. A logic 1 on this signal, sampled by the LCD/MIM panel module on the falling edge of LP, is used by the panel row drivers (Y drivers) to indicate the start of the vertical frame.
LP	O	26	TS2	Latch Pulse output. The falling edge of this signal is used to latch a row of display data in the LCD/MIM panel module's column driver shift registers and to turn on the row driver (Y driver) for that line.
XSCL	O	23	TS2	Shift Clock for LCD panel data or Pixel Clock for MIM panel data. Display data is clocked out of the chip on the rising edge of this signal, to be shifted into the LCD/MIM panel module column drivers (X drivers) on each falling edge.
UD[0:3]	O	34~37	TS2	Upper panel display data for dual LCD panel mode. For single LCD 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 LCD panel mode, these bits are the 4 bits of output data to the panel. For 4-bit MIM panel mode, these bits are driven 0.
LD[0:3]	O	28~31	TS2	Lower panel display data for dual LCD panel mode. For 8-bit single LCD 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 LCD panels, these bits are driven 0. For 4-bit MIM panel mode, these are the 4 bits of output data to the panel.
LCDPWR#	O	27	CO2	LCD power control. In normal operation this signal is driven low to enable an external LCD power supply. This signal is driven high when the chip is put into any Suspend mode, or if the Sequencer is in a reset state. It can be used externally to turn off the panel supply voltage and backlight. After a RESET, this signal is held high until the CRTC is programmed and running.
WF	O	24	TS2	LCD Panel Backplane Bias Signal or MIM Panel Data Enable Signal. In LCD panel mode, the WF signal toggles once per vertical frame period. In MIM panel mode, this signal goes high whenever display data are valid.

Table 0-9 Power Save Mode Control

Pin Name	Type	Pin #	Drv	Description
PDCLK	I	79	C	Power Down Clock. This input may be used to provide a low frequency clock for generating DRAM refresh in Suspend mode, as an optional alternative to using the CLKI or MEMEN input as the DRAM refresh clock source. This clock input should be driven by a 32 kHz 50% duty cycle clock. The PDCLK input is used to directly generate the RAS and CAS pulses in Suspend mode. Refer to section 10.3 on page 52, for details.
SUSPEND#	I	80	CS	A low level on this pin puts the chip into the Hardware Suspend mode. The SUSPEND# signal overrides any software initiated power save modes as well as the DOZE# input pin, and disables the CPU bus interface inputs. CPU Address and Data inputs are masked when this signal is low. When in Suspend Mode the UD[3:0], LD[3:0], XSCL, LP, YD and WF signals are driven into a low state (or optionally, a high impedance state) and the LCDPWR# signal is driven high.
DOZE#	I	81	CS	A low level on this pin puts the chip into Doze mode. The function of the Doze mode is determined by the Doze Mode Select bits in AUX[03]. This pin is ignored if the SUSPEND# input pin is asserted.

Table 0-10 Power Supply

Pin Name	Type	Pin #	Description
VDD CORE	P	12, 33, 55, 97, 118	VDD supply for core logic.
VDD I/O	P	1, 22, 44, 65, 86, 108	VDD supply for I/O pins.
VDD CLKI	P	76	VDD supply for CLKI pin.
VSS CORE	P	11, 32, 54, 96, 117	VSS supply for core logic.
VSS I/O	P	21, 43, 64, 85, 107, 128	VSS supply for I/O pins.
VSS CLKI	P	75	VSS supply for CLKI pin.

Table 0-11 Test Function

Pin Name	Type	Pin #	Drv	Description
TSTCO	I	38	CD	This pin enables the chip's test mode for the core logic. This pin must always be unconnected or tied to ground.
TSTEN	I	82	CD	This pin enables the chip's test mode for the I/O cells. This pin must always be unconnected or tied to ground.
N/C		78		No Connection

5.4 Power On / Reset Options

Table 0-12 Summary of Power On / Reset Options

Pin Name	value on this pin at falling edge of RESET is used to configure: (1/0)
MD[3:0]	values latched into read-only AUX[0C] bits 3-0 for software use
MD[5]	LCD signals' state in Suspend mode: Low (1), or Hi-Z (0)
MD[6]	2 CAS, 1 WE type DRAM (1), or 1 CAS, 2 WE type DRAM (0)

Only MD[3:0] inputs have internal pullup resistors; MD[6:5] inputs do not have internal pullup resistors. Based on the value of the internal pull-ups, the external pull-down resistors if necessary, should be approximately 15K ohm. This value will provide the correct voltage levels on power-up without loading the DRAM Data lines (VDD = 3.3V).

5.5 Multiple Function Pin Descriptions

Table 0-13 Multiple Function Pin Descriptions

Pin Name	Function	MD Line Status	Functional Description
LCAS#, LWE#	LCAS#	MD[6] = 1	DRAM Column Address Strobe (Low Byte)
	LWE#	MD[6] = 0	DRAM Write Enable Strobe
UCAS#, CAS#	UCAS#	MD[6] = 1	DRAM Column Address Strobe (High Byte)
	CAS#	MD[6] = 0	DRAM Column Address Strobe
WE#, UWE#	WE#	MD[6] = 1	DRAM Write Strobe
	UWE#	MD[6] = 0	DRAM Write Strobe (High Byte)

5.6 Mixed Voltage Configurations

Table 0-14 Mixed Voltage Configuration

Core V_{DD}	I/O V_{DD}	
	2.5 V	3.3 V
2.5 V	No	Yes
3.3 V	No	Yes

6.0 D.C. CHARACTERISTICS

D.C. Characteristics is a table of interface thresholds, static and dynamic current consumption.

Conditions: V_{DD} = 3.3V ±10% T_a = 0°C to 70°C

Table 0-15 Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
V_{DD}	Supply Voltage	V_{SS}-0.3 to +7.0	Volts
V_{IN}	Input Voltage	V_{SS}-0.3 to V_{DD}+0.3	Volts
V_{OUT}	Output Voltage	V_{SS}-0.3 to V_{DD}+0.3	Volts
T_{OPR}	Operating Temperature	0 to +70	°C
T_{STG}	Storage Temperature	-65 to +150	°C
T_{SOL}	Soldering Temperature/Time	260 for 10 sec max at lead	°C

Table 0-16 Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
HV_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	3.6	V
LV_{DD}	Supply Voltage	V_{SS} = 0 V	2.25	2.5	3.6	V
V_{IN}	Input Voltage	V_{SS}	V_{SS}	--	V_{DD}	V
T_{OPR}	Operating Temperature		0	25	70	˚C
I_{OPR}	Average Power Consumption	$V_{DD\ Core}$ = 2.5 V $V_{DD\ IO}$ = 3.3 V		20		mA
IP_{D1}	Doze Mode 1	$V_{DD\ Core}$ = 2.5 V $V_{DD\ IO}$ = 3.3 V		10		mA
IP_{D2}	Doze Mode 2	$V_{DD\ Core}$ = 2.5 V $V_{DD\ IO}$ = 3.3 V	5	5		mA
IP_{SUS}	Suspend	$V_{DD\ Core}$ = 2.5 V $V_{DD\ IO}$ = 3.3 V		0.2		mA

Table 0-17 Input Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage	V_{DD} = MIN			0.8	V
V_{IH}	High Level Input Voltage	V_{DD} = MAX	2.0			V
V_{T+}	Positive-going Threshold (CMOS Schmitt inputs)	V_{DD} = 3.3 V			2.4	V
V_{T-}	Negative-going Threshold (CMOS Schmitt inputs)	V_{DD} = 3.3 V	0.6			V
V_H	Hysteresis Voltage (CMOS Schmitt inputs)	V_{DD} = 3.3 V	0.1			V
I_{IZ}	Input Leakage Current	V_{DD} = MAX $V_{IH} = V_{DD}$ $V_{IL} = V_{SS}$	-1		1	µA
C_{IN}	Input Pin Capacitance			8		pF
R_{PU}	Pull Up Resistance	V_{DD} = 3.3 V		90		kΩ
R_{PD}	Pull Down Resistance	V_{DD} = 3.3 V		90		kΩ

Table 0-18 Output Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{OL1}	Low Level Output Current	$V_{OL} = V_{SS}$+0.4V TS1, TS1U, CO1	3.0			mA
I_{OH1}	High Level Output Current	$V_{OH} = V_{DD}$-0.4V TS1, TS1U, CO1	-3.0			mA
I_{OL2}	Low Level Output Current	$V_{OL} = V_{SS}$+0.4V TS2, CO2	6.0			mA
I_{OH2}	High Level Output Current	$V_{OH} = V_{DD}$-0.4V TS2, CO2	-6.0			mA
I_{OZ}	Output Leakage Current	$V_{OH} = V_{DD}$ or $V_{OL} = V_{SS}$	-1		1	µA
C_{OUT}	Output Pin Capacitance			8		pF

7.0 A.C. CHARACTERISTICS

Conditions: HV_{DD} = 3.3V ±10% LV_{DD} = 2.5V ±10% T_a = 0°C to 70°C
T_r, T_f for all CMOS inputs must be < 5 nsec (V_{IN} = 0.1 V_{DD} ~ 0.9 V_{DD})
C_L = 50 pF (LCD Panel Interface)
C_L = 80 pF (CPU Bus Interface)
C_L = 20 pF (Video Memory Interface)

CLK Signal Dependant Input Timing

Pixel Clock:

$$T_S = \left[\frac{1}{f_{CLKI}}\right]$$

MEMEN input:

$$T_m = \left[\frac{1}{f_{MEMEN}}\right]$$

T_{ml} = low pulse width of MEMEN

PDCLK input:

$$T_p = \left[\frac{1}{f_{PDCLK}}\right]$$

If MEMEN is to be used as a DRAM refresh clock source in Suspend mode, then T_{ml} should be as short as possible, but 50 ns greater than the minimum RAS pulse width required by the DRAM.

Propagation Delay Time

The following are tables of timing parameters and min/max values. These tables are followed by waveforms defining these parameters.

7.1 CPU Bus Cycle Timing - 8-bit Memory and I/O

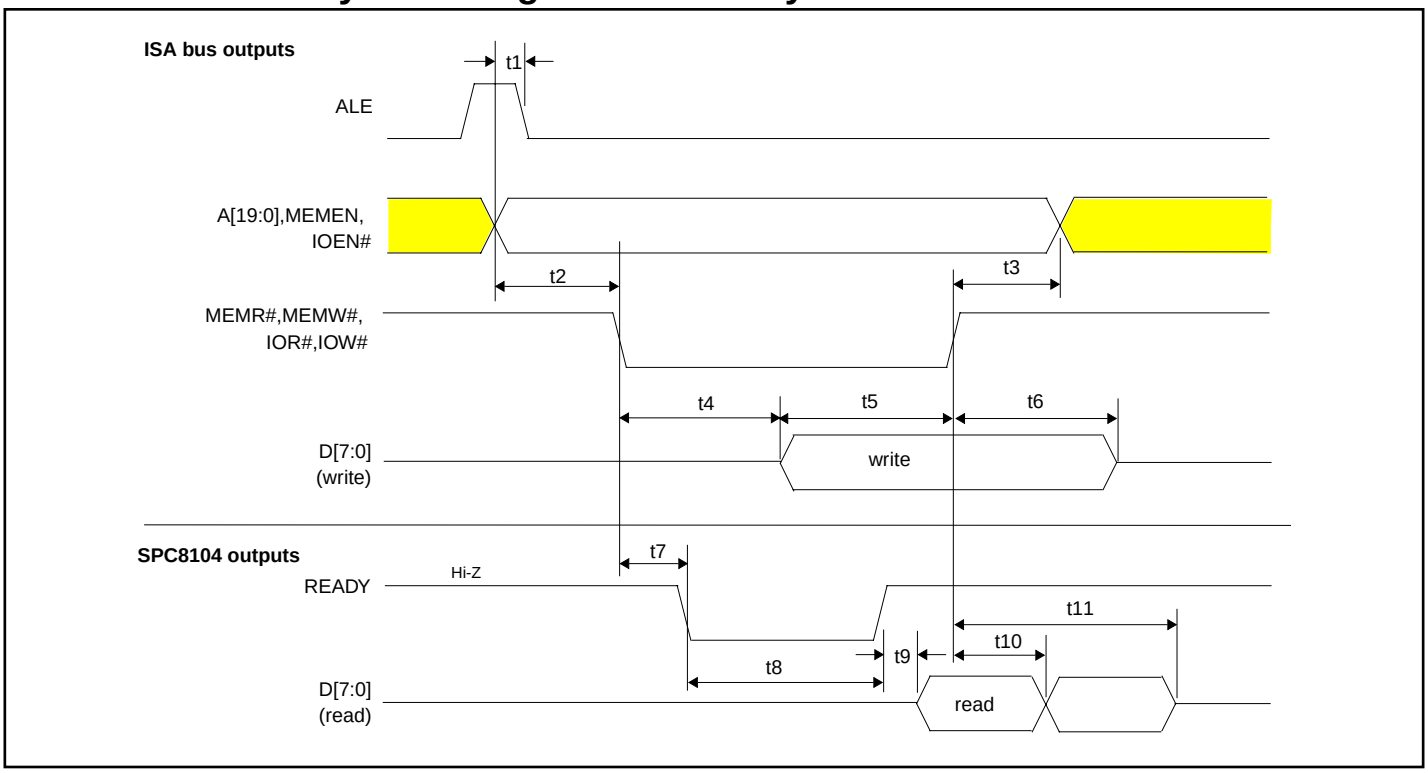


*Figure 4 : CPU Bus Cycle Timing - 8-bit Memory and I/O*Source: a000070.cdr

Table 0-19 CPU Bus Cycle Timing - 8-bit Memory and I/O

Symbol	Parameter	Min	Typ	Max	Units
t1	A[19:17] valid setup to ALE negated (falling edge)	10			ns
t2a	A[19:0], MEMEN valid before MEMR#, MEMW# asserted	0			ns
t2b	A[15:0], IOEN# valid before IOR#, IOW# asserted	0			ns
t3a	A[19:0], MEMEN hold from MEMR#, MEMW# negated	0			ns
t3b	A[15:0], IOEN# hold from IOR#, IOW negated	24			ns
t4	MEMW# asserted to D[7:0] valid			3Ts	ns
t5	D[7:0] setup to IOW# negated	0			ns
t6a	D[7:0] hold from MEMW# negated	0			ns
t6b	D[7:0] hold from IOW# negated	10			ns
t7	MEMR#, MEMW# asserted to READY negated			50	ns
t8a	READY negated pulse width (dual panel)			36Ts	ns
t8b	READY negated pulse width (single panel)			69Ts	ns
t8c	IOR# asserted to D[7:0] valid			80	ns
t9	READY asserted to D[7:0] valid (read)			70	ns
t10a	D[7:0] hold from MEMR# negated	10			ns
t10b	D[7:0] hold from IOR# negated	10			ns
t11a	MEMR# negated to D[7:0] hi-Z delay			30	ns
t11b	IOR# negated to D[7:0] hi-Z delay			30	ns

7.2 CPU Bus Cycle Timing - 16-bit Memory

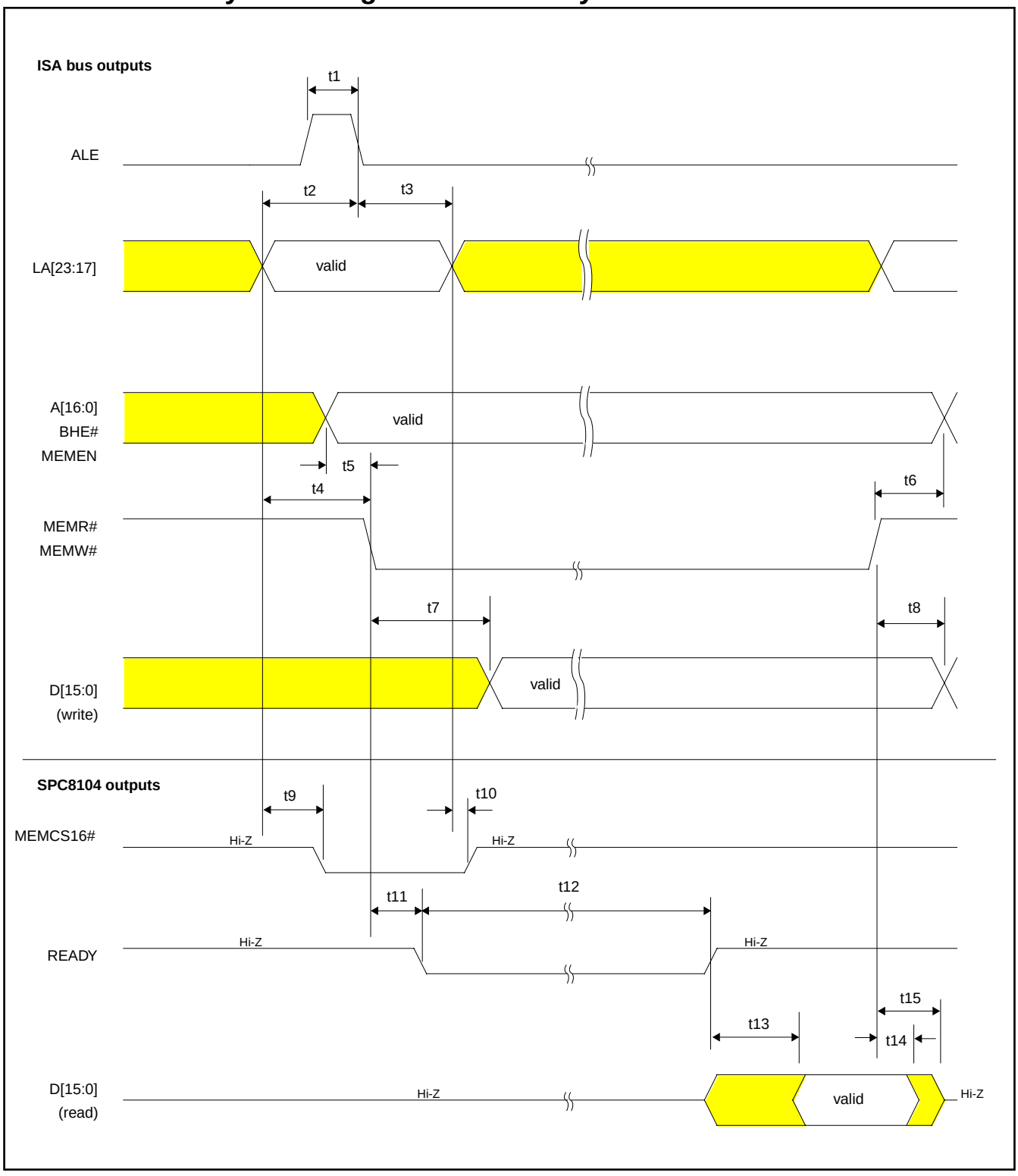


*Figure 5 : CPU Bus Cycle Timing - 16-bit Memory*Source: a000071.cdr

Table 0-20 CPU Bus Cycle Timing - 16-bit Memory

Symbol	Parameter	Min	Typ	Max	Units
t1	ALE pulse width	30			ns
t2	LA[23:17] valid setup to ALE asserted	20			ns
t3	LA[23:17] valid hold from ALE negated	10			ns
t4	LA[23:17] valid setup to MEMR#, MEMW# asserted	10			ns
t5	A[16:0], BHE#, MEMEN valid setup to MEMR#, MEMW# asserted	10			ns
t6	A[16:0], BHE#, MEMEN hold from MEMR#, MEMW# negated	10			ns
t7	valid write data delay from MEMW# asserted			3Ts	ns
t8	valid write data hold from MEMW# negated	0			ns
t9	MEMCS16# asserted from valid LA[23:17]			34	ns
t10	MEMCS16# hold from LA[23:17] invalid	0			ns
t11	READY negated from MEMR#, MEMW# asserted			50	ns
t12a	READY negated pulse width (dual panel)			52Ts	ns
t12b	READY negated pulse width (single panel)			101Ts	ns
t13	valid read data from READY released			80	ns
t14	read data hold from MEMR# negated	10			ns
t15	MEMR# negated to D[15:0] high-impedance			30	ns

Parameter t13 maximum only occurs when a refresh cycle is pending during fast dot, text modes. Typical values are much shorter.

7.3 CPU Bus Cycle Timing - 16-bit I/O

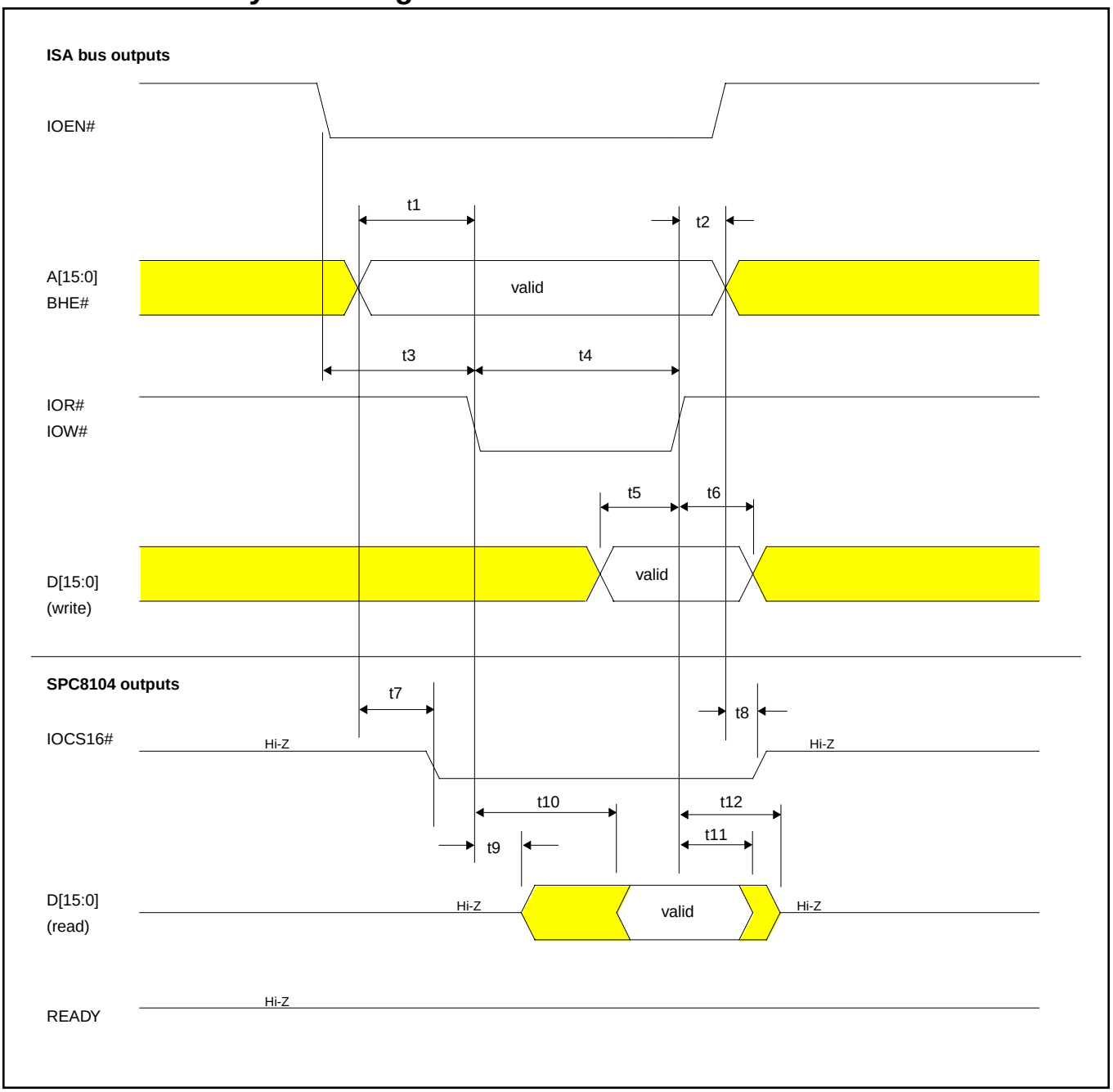


Figure 6 : CPU Bus Cycle Timing - 16-bit I/OSource: a000072.cdr

Table 0-21 CPU Bus Cycle Timing - 16-bit I/O

Symbol	Parameter	Min	Typ	Max	Units
t1	A[15:0], BHE# valid setup to IOR#, IOW# asserted	10			ns
t2	A[15:0], BHE# valid hold from IOR#, IOW# negated	24			ns
t3	IOEN# setup to IOR#, IOW# asserted	50			ns
t4a	IOW# command pulse width	120			ns
t4b	IOR# command pulse width	180			ns
t5	valid write data setup to IOW# negated	10			ns
t6	valid write data hold from IOW# negated	10			ns
t7	IOCS16# asserted from A[15:0] valid			50	ns
t8	IOCS16# hold from A[15:0] invalid	0			ns
t9	read data driven delay from IOR# asserted	5			ns
t10	valid read data from IOR# asserted			80	ns
t11	read data hold from IOR# negated	10			ns
t12	IOR# negated to D[15:0] high-impedance			30	ns

7.4 DRAM Read Cycle Timing - Non-Page Mode

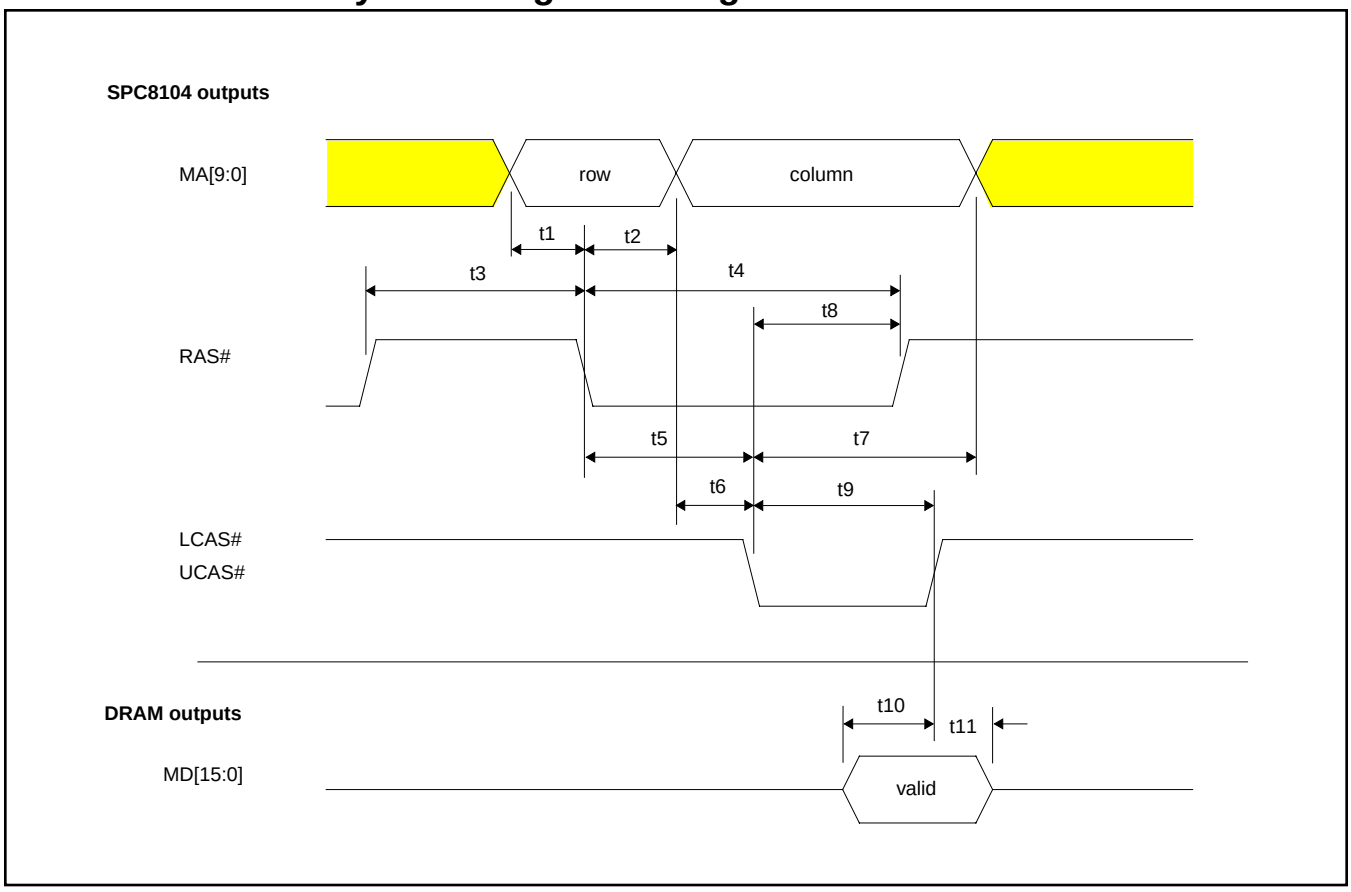


*Figure 7 : DRAM Read Cycle Timing - Non-Page Mode*Source: a000073.cdr

Table 0-22 DRAM Read Cycle Timing - Non-Page Mode

Symbol	Parameter	Min	Typ	Max	Units
t1	MA[9:0] row address setup to RAS# asserted	Ts-22			ns
t2	MA[9:0] row address hold from RAS# asserted	0.5Ts-5			ns
t3	RAS# precharge	2Ts-12			ns
t4	RAS# low pulse width	2.5Ts-10		2.5Ts+12	ns
t5	RAS# asserted to LCAS#, UCAS# asserted	1.5Ts-5			ns
t6	MA[9:0] column address setup to LCAS#, UCAS# asserted	Ts-12			ns
t7	MA[9:0] column address hold from LCAS#, UCAS# asserted	Ts-10			ns
t8	LCAS#, UCAS# asserted to RAS# negated	Ts-10			ns
t9	LCAS#, UCAS# low pulse width	Ts-10			ns
t10	MD[15:0] read data setup to LCAS#, UCAS# negated	10			ns
t11	MD[15:0] read data hold from LCAS#, UCAS# negated	0			ns

7.5 DRAM Read Cycle Timing - Page Mode

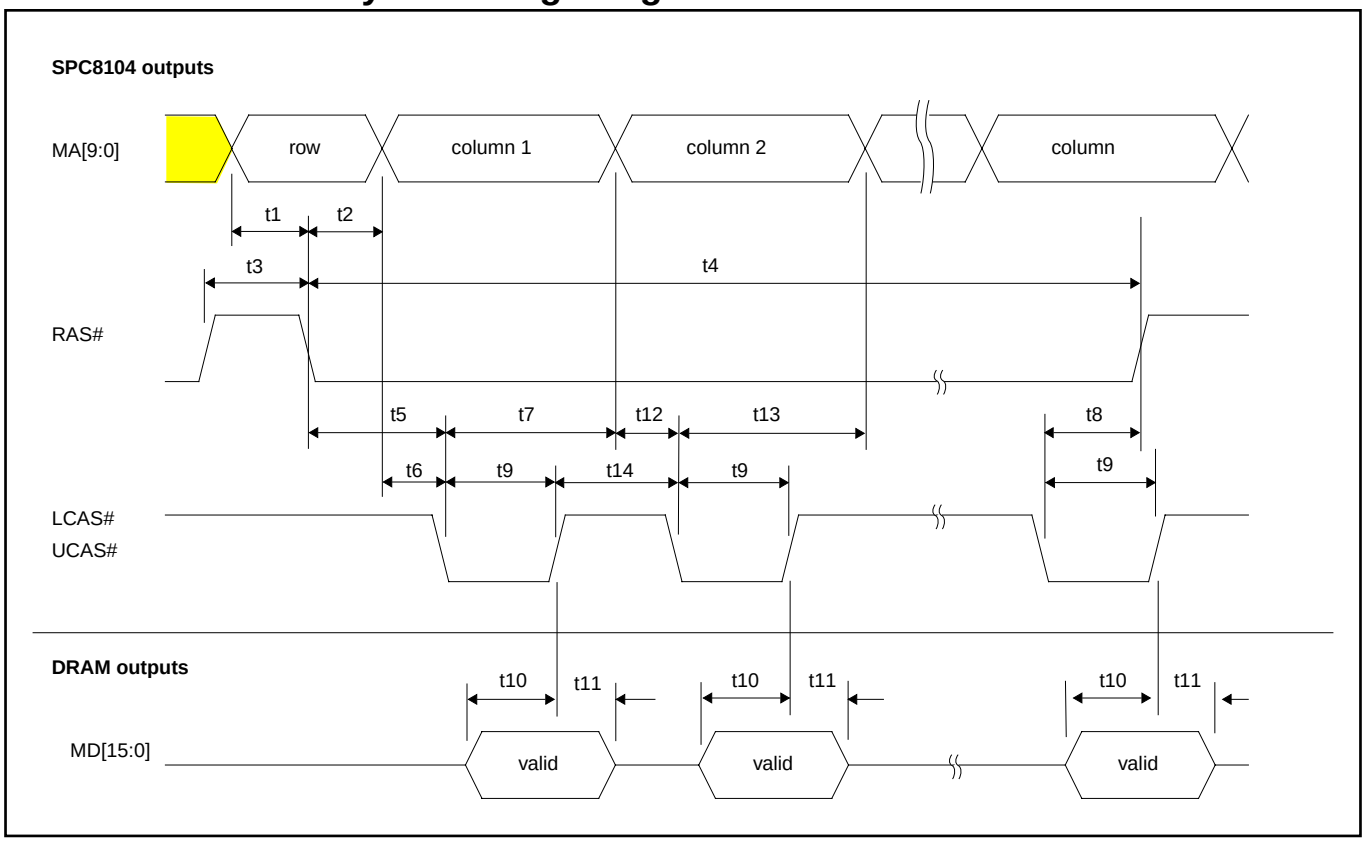


*Figure 8 : DRAM Read Cycle Timing - Page Mode*Source: a000074.cdr
Table 0-23 DRAM Read Cycle Timing - Page Mode

Symbol	Parameter	Min	Typ	Max	Units
t1	MA[9:0] row address setup to RAS# asserted	Ts-15			ns
t2	MA[9:0] row address hold from RAS# asserted	0.5Ts-2			ns
t3	RAS# precharge	2Ts-12			ns
t4	RAS# pulse width	3.5Ts-5 (n=2)		6.5Ts-5 (n=4)	ns
t5	RAS# asserted to LCAS#, UCAS# asserted	1.5Ts-5			ns
t6	MA[9:0] col. address setup to LCAS#, UCAS# asserted(1st)	Ts-13			ns
t7	MA[9:0] col. address hold from LCAS#, UCAS# asserted(1st)	0.5Ts-2			ns
t8	LCAS#, UCAS# asserted to RAS# negated	0.5Ts-2			ns
t9	LCAS#, UCAS# pulse width	Ts-5			ns
t10	MD[15:0] read data setup to LCAS#, UCAS# negated	10			ns
t11	MD[15:0] read data hold from LCAS#, UCAS# negated	0			ns
t12	MA[9:0] column address setup to LCAS#, UCAS# asserted (2nd-nth page access)	Ts-20			ns
t13	MA[9:0] column address hold from LCAS#, UCAS# asserted (2nd-nth page access)	0.5Ts			ns
t14	LCAS#, UCAS# precharge	0.5Ts-8			ns

7.6 DRAM Write Cycle Timing

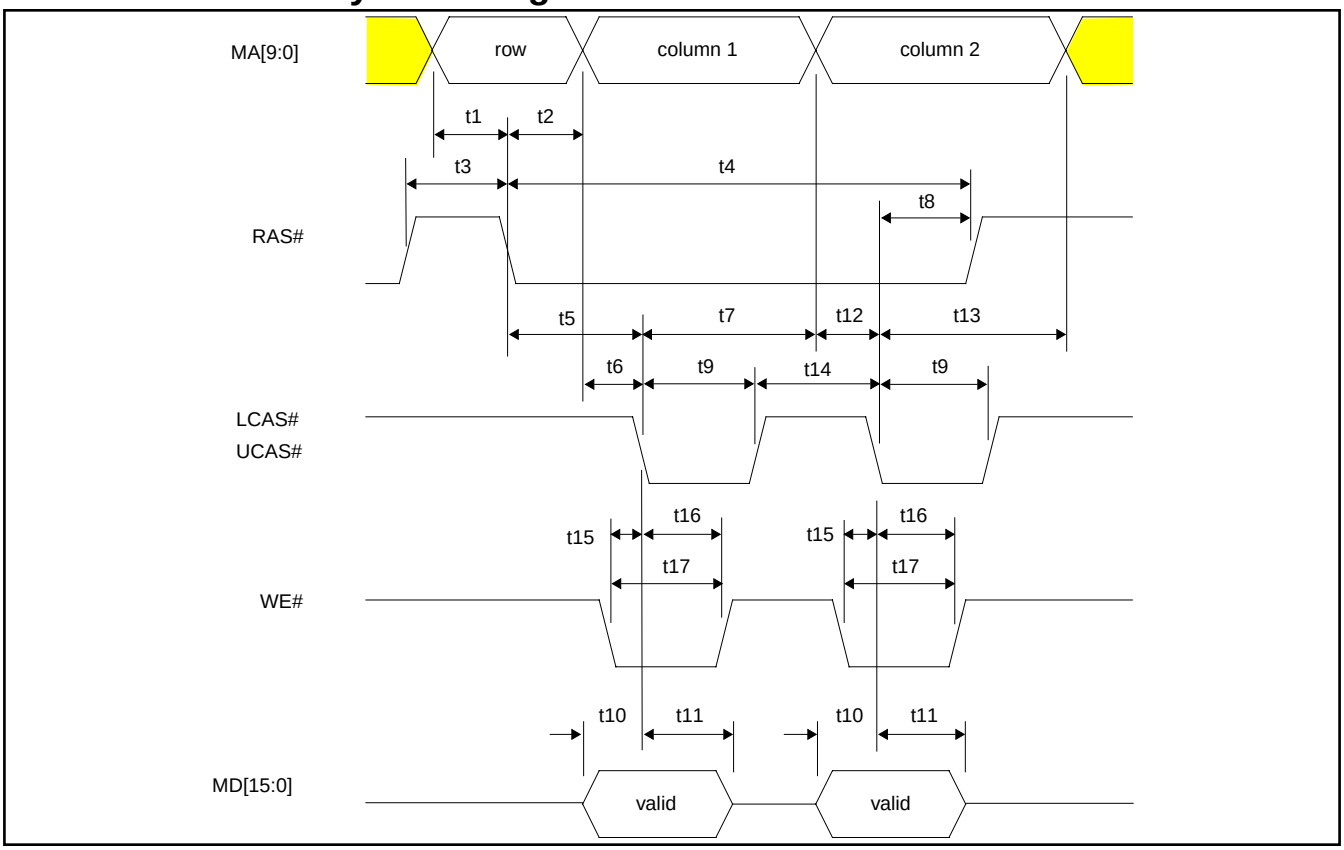


Figure 9 : DRAM Write Cycle Timing Source: a000075.cdr

Table 0-24 DRAM Write Cycle Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	MA[9:0] row address setup to RAS# asserted	Ts-15			ns
t2	MA[9:0] row address hold from RAS# asserted	0.5Ts-5			ns
t3	RAS# precharge	2Ts-12			ns
t4	RAS# pulse width	3.5Ts-10			ns
t5	RAS# asserted to LCAS#, UCAS# asserted	1.5Ts-10			ns
t6	MA[9:0] col. address setup to LCAS#, UCAS# asserted (1st)	Ts-13			ns
t7	MA[9:0] col. address hold from LCAS#, UCAS# asserted (1st)	0.5Ts-2			ns
t8	LCAS#, UCAS# asserted to RAS# negated	0.5Ts-2			ns
t9	LCAS#, UCAS# pulse width	Ts-10			ns
t10	MD[15:0] data setup to LCAS#, UCAS# asserted	0.5Ts-10			ns
t11	MD[15:0] data hold from LCAS#, UCAS# asserted	0.5Ts-2			ns
t12	MA[9:0] col. address setup to LCAS#, UCAS# asserted (2nd)	Ts-10			ns
t13	MA[9:0] col. address hold from LCAS#, UCAS# asserted (2nd)	0.5Ts-5			ns
t14	LCAS#, UCAS# precharge	0.5Ts-8			ns
t15	WE# setup to LCAS#, UCAS# asserted	0.5Ts-5			ns
t16	WE# hold from LCAS#, UCAS# asserted	0.5Ts-5			ns
t17	WE# low pulse width	Ts-10			ns

7.7 DRAM Frame Buffer Cycle Timing

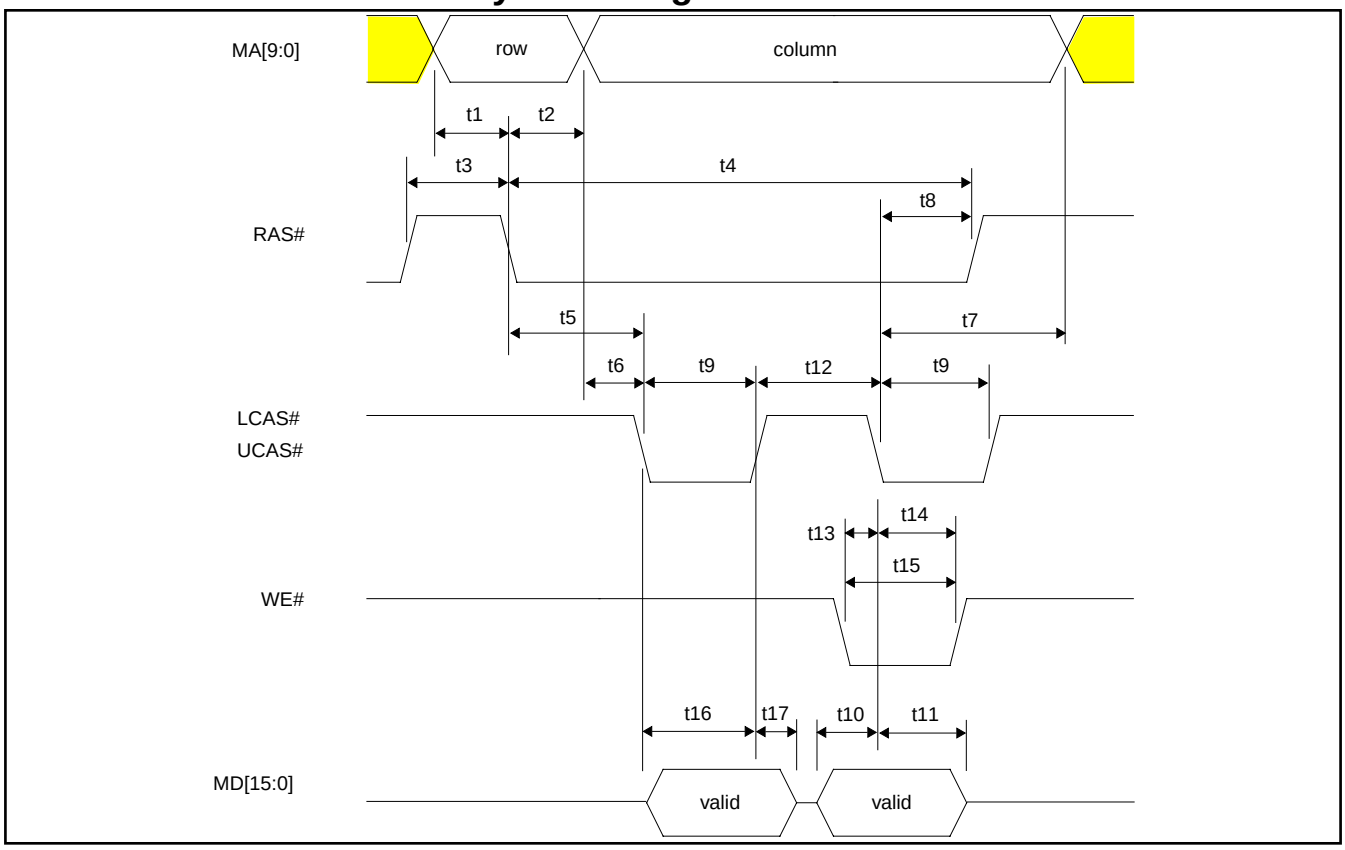


Figure 10 : DRAM Write Cycle Timing Source: aXXXXXX.cdr

Table 0-25 DRAM Write Cycle Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	MA[9:0] row address setup to RAS# asserted	Ts-16			ns
t2	MA[9:0] row address hold from RAS# asserted	0.5Ts-5			ns
t3	RAS# precharge	2Ts-12			ns
t4	RAS# pulse width	4Ts-10		4Ts+12	ns
t5	RAS# asserted to LCAS#, UCAS# asserted	1.5Ts-5			ns
t6	MA[9:0] col. address setup to LCAS#, UCAS# asserted (1st)	Ts-12			ns
t7	MA[9:0] col. address hold from LCAS#, UCAS# asserted (2nd)	0.5Ts-2			ns
t8	LCAS#, UCAS# asserted to RAS# negated	0.5Ts-2			ns
t9	LCAS#, UCAS# pulse width	Ts-10			ns
t10	MD[15:0] write data setup to LCAS#, UCAS# asserted	0.5Ts-11			ns
t11	MD[15:0] write data hold from LCAS#, UCAS# asserted	0.5Ts-2			ns
t12	LCAS#, UCAS# precharge	Ts-7			ns
t13	WE# setup to LCAS#, UCAS# asserted	0.5Ts-5			ns
t14	WE# hold from LCAS#, UCAS# asserted	0.5Ts-3			ns
t15	WE# low pulse width	Ts-10			ns
t16	MD[15:0] read data setup to LCAS#, UCAS# negated	10			ns
t17	MD[15:0] read data hold from LCAS#, UCAS# negated	0			ns

7.8 DRAM Refresh Timing I

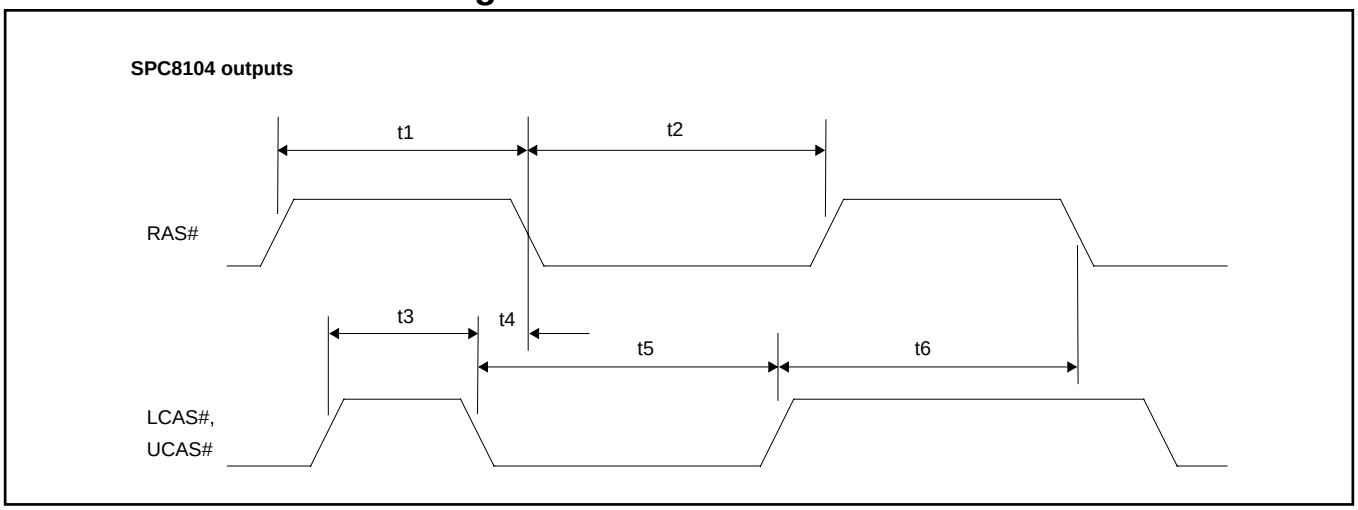


Figure 11 : DRAM Refresh Timing I (generated from CLKI, Sequencer running)Source: a000076.cdr

Table 0-26 DRAM Refresh Timing I (generated from CLKI, Sequencer running)

Symbol	Parameter	Min	Typ	Max	Units
t1	RAS# high pulse width	2Ts-12			ns
t2	RAS# low pulse width	3.5Ts-10			ns
t3	LCAS#, UCAS# high pulse width	0.5Ts-7			ns
t4	LCAS#, UCAS# setup to RAS# asserted (CAS-before-RAS refresh)	Ts-5		Ts	ns
t5	LCAS#, UCAS# low pulse width	4Ts-10			ns
t6	LCAS#, UCAS# negated to RAS# asserted	2.5Ts-9			ns

These timing values apply to CAS-before-RAS refresh cycles occurring during all active modes and Doze modes

7.9 DRAM Refresh Timing II

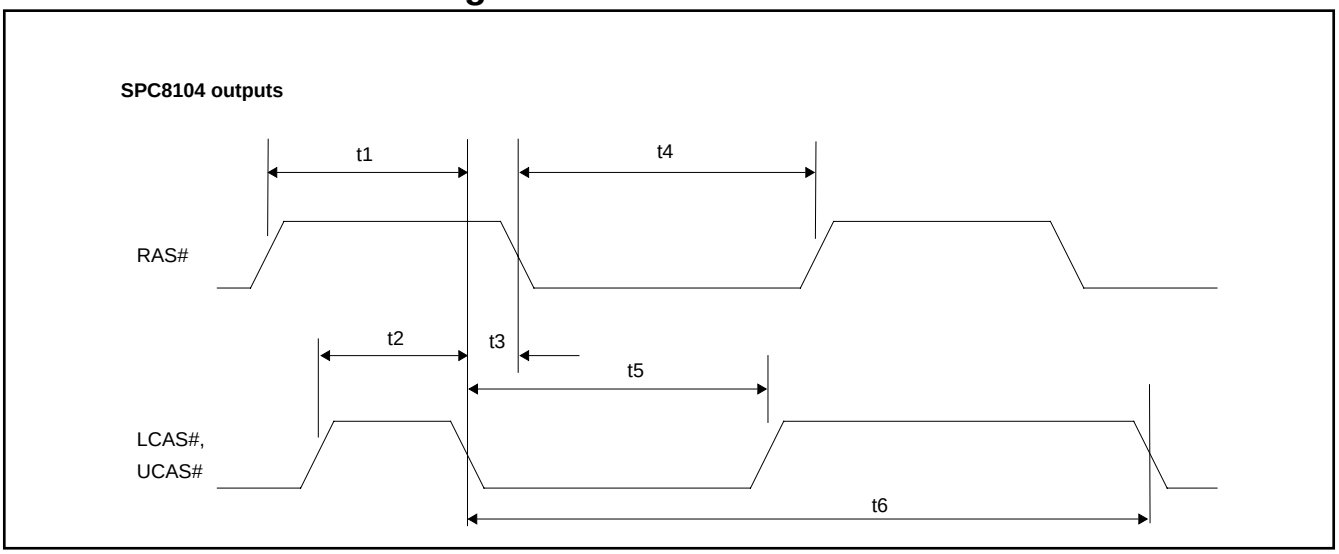


Figure 12 : DRAM Refresh Timing II (Sequencer stopped)Source:a000077.cdr

Table 0-27 DRAM Refresh Timing II (Sequencer stopped)

Symbol	Parameter	Min	Typ	Max	Units
t1	RAS# negated to LCAS#, UCAS# asserted	1.5Ts			ns
t2	LCAS#, UCAS# high pulse width	Ts			ns
t3	LCAS#, UCAS# asserted to RAS# asserted	12			ns
t4a	RAS# low pulse width (Sequencer stopped, selected CLKI used as refresh clock source)	3Ts-10			ns
t4b	RAS# low pulse width (Sequencer stopped, MEMEN input used as refresh clock source)	Tml-44			ns
t4c	RAS# low pulse width (Sequencer stopped, 32kHz PDCLK input used as refresh clock source)	92			ns
t5a	LCAS#, UCAS# low pulse width (Sequencer stopped, selected CLKI used as refresh source)	4Ts-10			ns
t5b	LCAS#, UCAS# low pulse width (Sequencer stopped, MEMEN input used as refresh clock source)	Tml-10			ns
t5c	LCAS#, UCAS# low pulse width (Sequencer stopped, 32kHz PDCLK input used as refresh clock source)	107			ns
t6a	Suspend Mode refresh period (CLKI used as refresh source)	320Ts-20		2560Ts-20	ns
t6b	Suspend Mode refresh period (MEMEN used as refresh source)	Tm-10		8Tm-10	ns
t6c	Suspend Mode refresh period (Sequencer stopped, 32kHz PDCLK input used as refresh clock source)	0.5Tp-5		4Tp-5	ns

These timing values apply to CAS-before-RAS refresh cycles occurring only during Suspend mode where the Sequencer is stopped and the refresh timing is being generated from the selected CLKI input, from MEMEN input, or from PDCLK input. These values also apply to self-refresh mode.

7.10 LCD Interface Timing

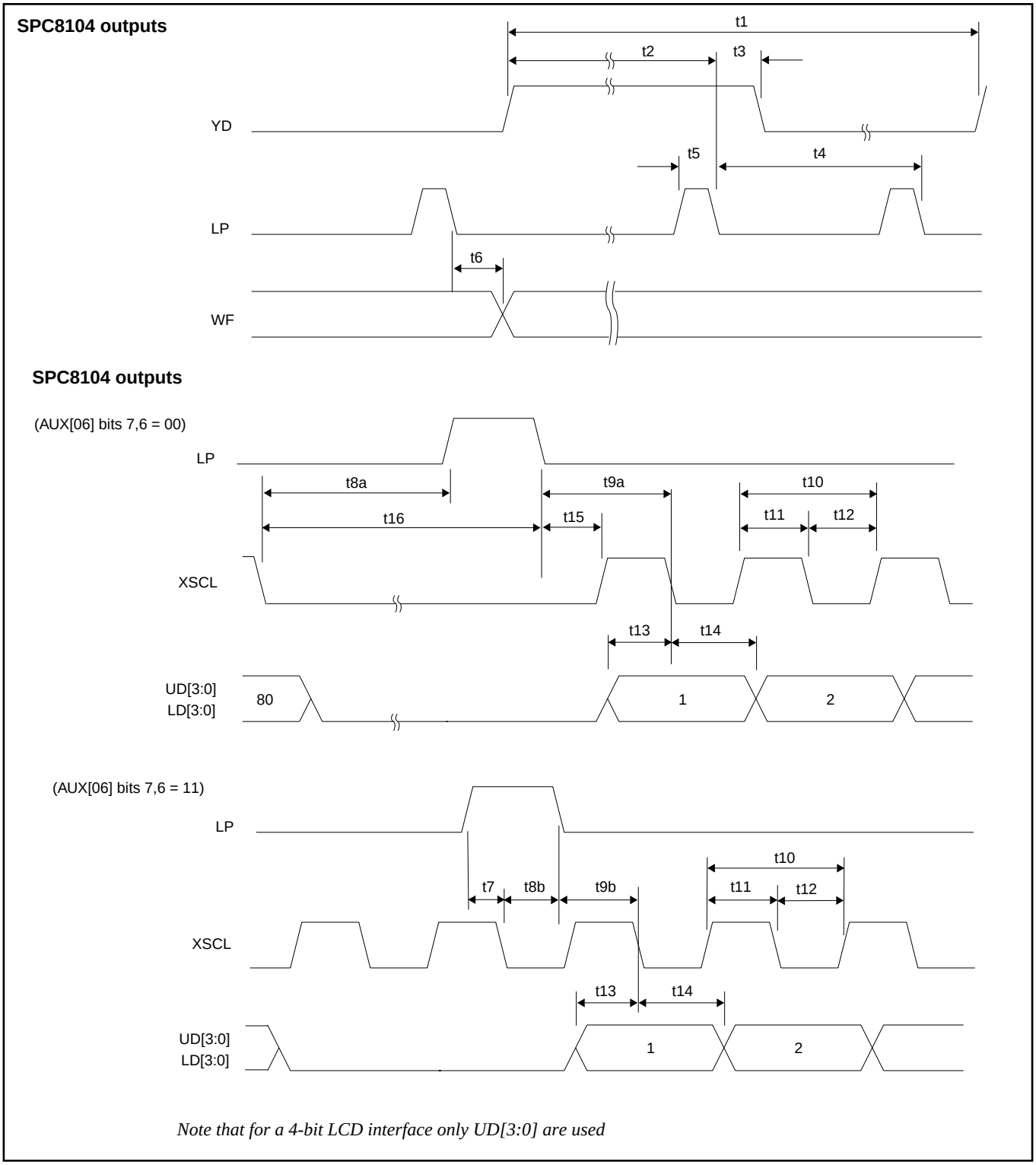


*Figure 13 : LCD Interface Timing*source: aXXXXXX.cdr

Table 0-28 LCD Interface Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	YD period				
t2	YD setup to LP negated (single panel mode)	728Ts-24			ns
	YD setup to LP negated (dual panel mode)	1464Ts -24			ns
t3	YD hold from LP negated (single panel mode)	8Ts-24			ns
	YD hold from LP negated (dual panel mode)	8Ts-24			ns
t4	LP period (single panel mode)	736Ts	752Ts	4080Ts	ns
	LP period (dual panel mode)	1472Ts	1504Ts	8160Ts	ns
t5	LP pulse width (AUX[06] bit 6 = 0)	8Ts-24			ns
	LP pulse width (AUX[06] bit 6 = 1)	6Ts-24			ns
t6	WF delay from LP negated	0		20	ns
t7	LP setup to XSCL falling edge (AUX[06] bit 7, 6 = 11)	2Ts-24			ns
t8a	LP hold from XSCL falling edge (AUX[06] bit 7, 6 = 00)	4Ts-24			ns
t8b	LP hold from XSCL falling edge (AUX[06] bit 7, 6 = 11)	n/a	n/a	n/a	ns
t9a	LP negated to XSCL falling edge (AUX[06] bits 7,6 = 00)	12Ts-24			ns
t9b	LP negated to XSCL falling edge (AUX[06] bits 7,6 =11)	4Ts-24			ns
t10	XSCL period	8Ts-24			ns
t11	XSCL high width	4Ts-24			ns
t12	XSCL low width	4Ts-24			ns
t13	UD[3:0], LD[3:0] setup to XSCL falling edge	4Ts-40			ns
t14	UD[3:0], LD[3:0] hold from XSCL falling edge	4Ts-40			ns
t15	LP negated to XSCL rising edge (AUX[06] bits 7,6 = 00)	8Ts-24			ns
t16	XSCL falling edge to LP falling edge - single panel mode (AUX[06] bit 7,6 = 00 only)	96Ts			ns
	XSCL falling edge to LP falling edge - dual panel mode (AUX[06] bit 7, 6 = 00 only)	192Ts			ns

7.11 MIM Interface Timing

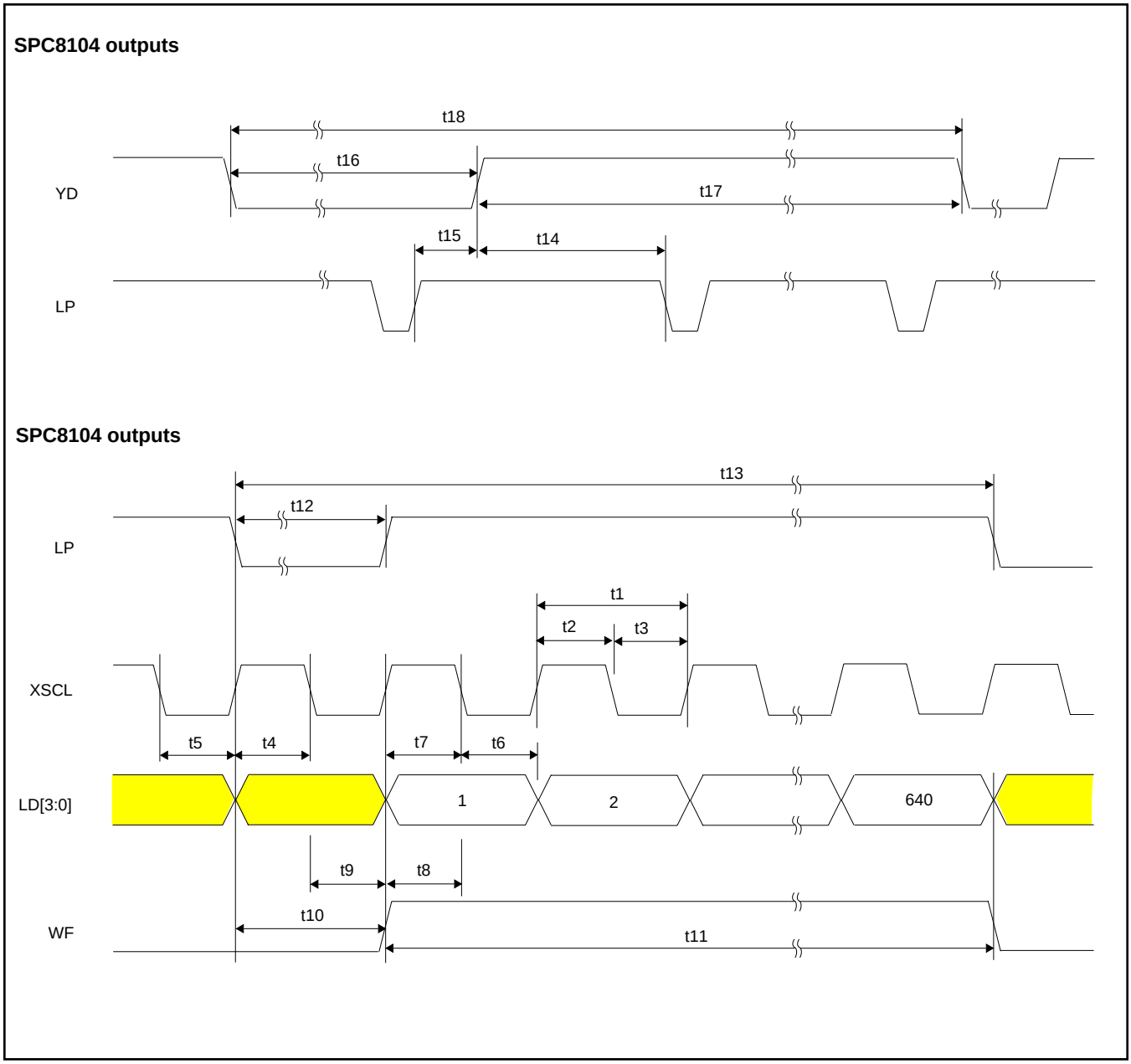


Figure 14 : MIM Panel Interface Timing

Table 0-29 MIM Panel Interface Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	XSCL frequency		24		MHz
t2	XSCL high	16			ns
t3	XSCL low	16			ns
t4	LP setup to XSCL low	10			ns
t5	LP hold from XSCL low	10			ns
t6	LD [3:0] setup to XSCL low	10			ns
t7	LD [3:0] hold from XSCL low	10			ns
t8	WF/DATAEN setup to XSCL low	10			ns
t9	WF/DATAEN hold from XSCL low	10			ns
t10	WF/DATAEN, LD [3:0] valid from LP low	8			Ts
t11	WF/DATAEN, LD [3:0] valid period	26.7			ms
t12	LP active width	3			Ts
t13	LP period	648			Ts
t14	YD setup to LP active	40			ns
t15	YD hold from LP active	40			ns
t16	YD active width	2			lines
t17	YD inactive width	480			lines
t18	YD period	482			lines

7.12 Software Initiated Suspend Mode Signal Timing

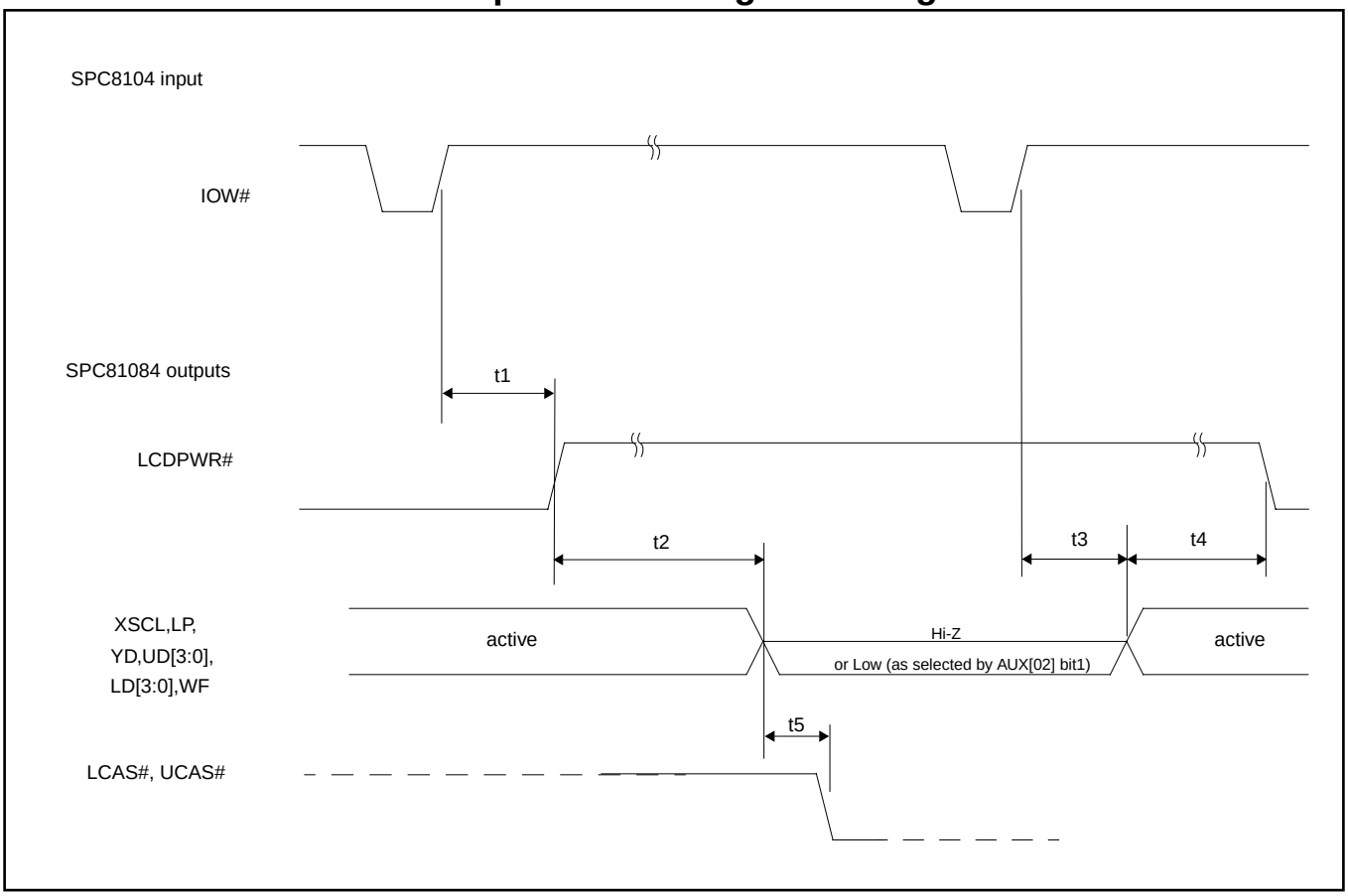


*Figure 15 : Software Initiated Suspend Mode Signal Timing*Source: a000079.cdr

Table 0-30 Software Initiated Suspend Mode Signal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Power Save Register IOW# command negated (valid Power Save Mode set) to LCDPWR# negated			4Ts+30	ns
t2	Power Save Register IOW# command negated (valid Power Save Mode set) to LCD interface signals high-impedance, or driven low (as selected by AUX[02] bit 1)	2Ty		3Ty	ns
t3	Power Save Register IOW# command negated (return to active mode) to LCD interface signals active			4Ts+30	ns
t4	LCD interface signals active to LCDPWR# asserted	2Ty			ns
t5	LCD interface signals high-impedance or low to LCAS#, UCAS# asserted (to initialize DRAM self-refresh mode)	Ts			ns

Tv = internal vertical retrace period
= 327680Ts if CLKI or self-refresh mode (approximately 13.6ms for 24MHz)
= 1024Tm if MEMEN refresh clock source is selected
= 512Tp if PDCLK refresh clock source is selected

Ty = YD period, typically 14.8ms

7.13 Hardware Initiated Suspend Mode Signal Timing

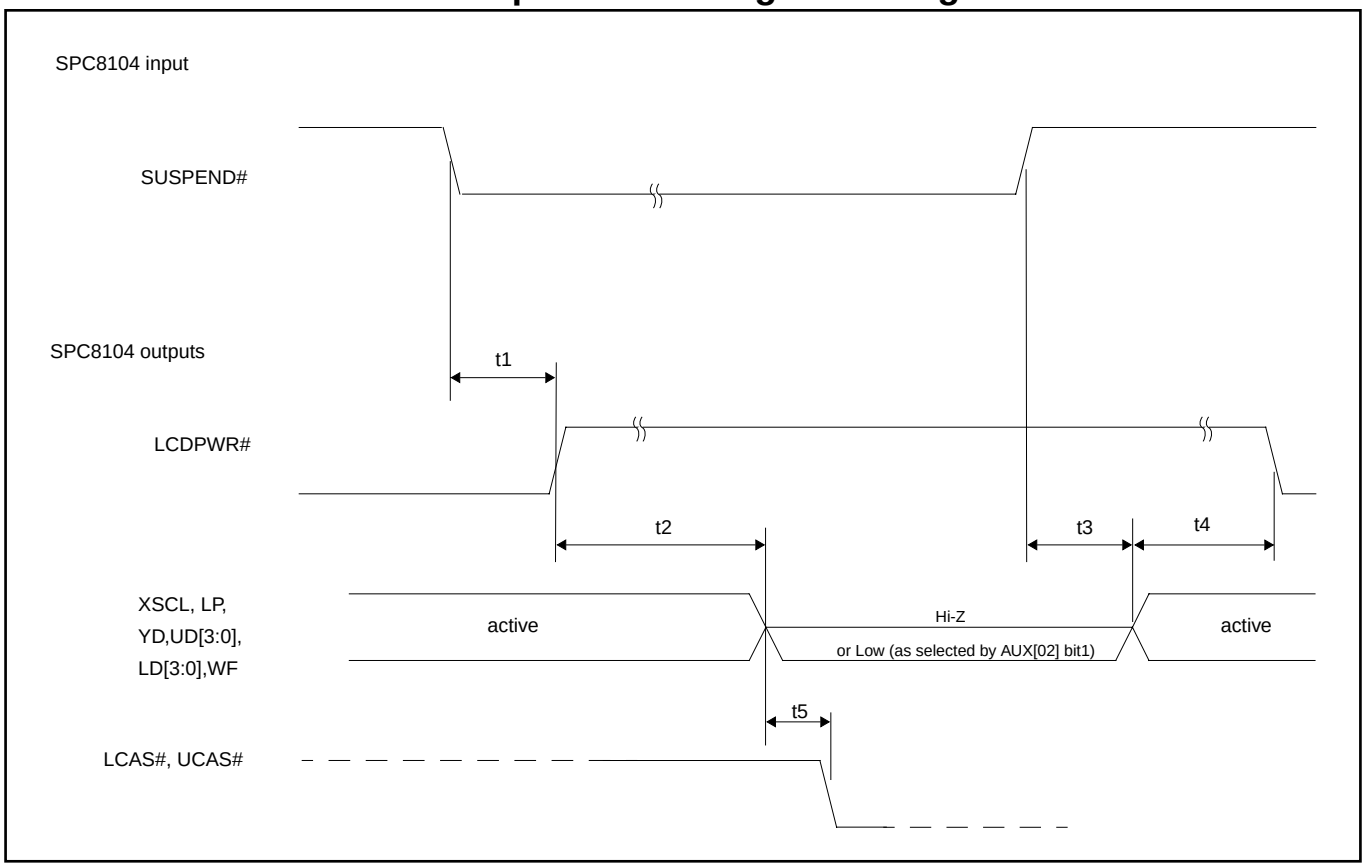


*Figure 16 : Hardware Initiated Suspend Mode Signal Timing*Source: a000080.cdr

Table 0-31 Hardware Initiated Suspend Mode Signal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	SUSPEND# asserted to LCDPWR# negated			4Ts	ns
t2	SUSPEND# asserted to LCD interface signals high-impedance, or driven low (as selected by AUX[02] bit 1)	2Ty		3Ty	ns
t3a	SUSPEND# negated to LCD interface signals driven (CLKI selected as refresh clock source)			4Ts	ns
t3b	SUSPEND# negated to LCD interface signals driven (self-refresh mode, MEMEN, or PDCLK selected as refresh clock source)	Tv+3Ts		2Tv +4Ts	ns
t4	LCD interface signals driven to LCDPWR# asserted	2Ty			ns
t5	LCD interface signals high-impedance or low to LCAS#, UCAS# asserted (to initialize DRAM self-refresh mode)	Ts			ns

Tv = internal vertical retrace period
= 327680Ts if CLKI or self-refresh mode (approximately 13.6ms for 24MHz)
= 1024Tm if MEMEN refresh clock source is selected
= 512Tp if PDCLK refresh clock source is selected

Ty = YD period, typically 14.8ms

7.14 Clock Input Requirements

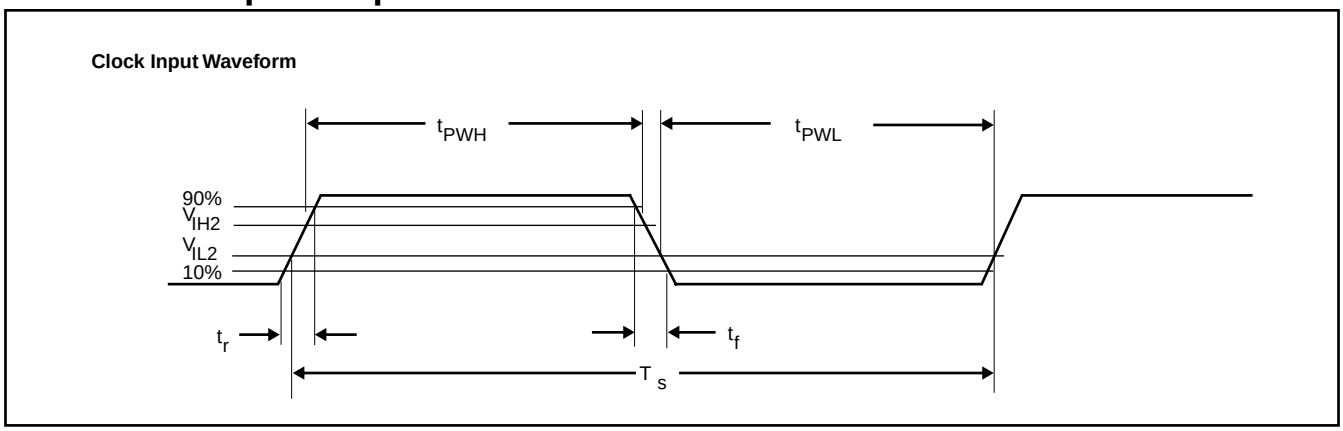


*Figure 17 : Clock Input Requirements*Source: a000081.cdr

Table 0-32 Clock Input Requirements

Symbol	Parameter	Min	Typ	Max	Units
T_S	Input Clock Period (CLKI)	36			ns
T_m	Input Clock Period (MEMEN)		15.625		µs
$T_{p\ (32)}$	Input Clock Period (32kHz PDCLK)		31.25		µs
t_{PWH}	Input Clock Pulse Width High (CLKI)	45		65	% T_S
t_{PWL}	Input Clock Pulse Width Low (CLKI)	45		65	% T_S
t_f	Input Clock Fall Time (10% - 90%)		5		ns
t_r	Input Clock Rise Time (10% - 90%)		5		ns

8.0 LCD INTERFACE OPTIONS

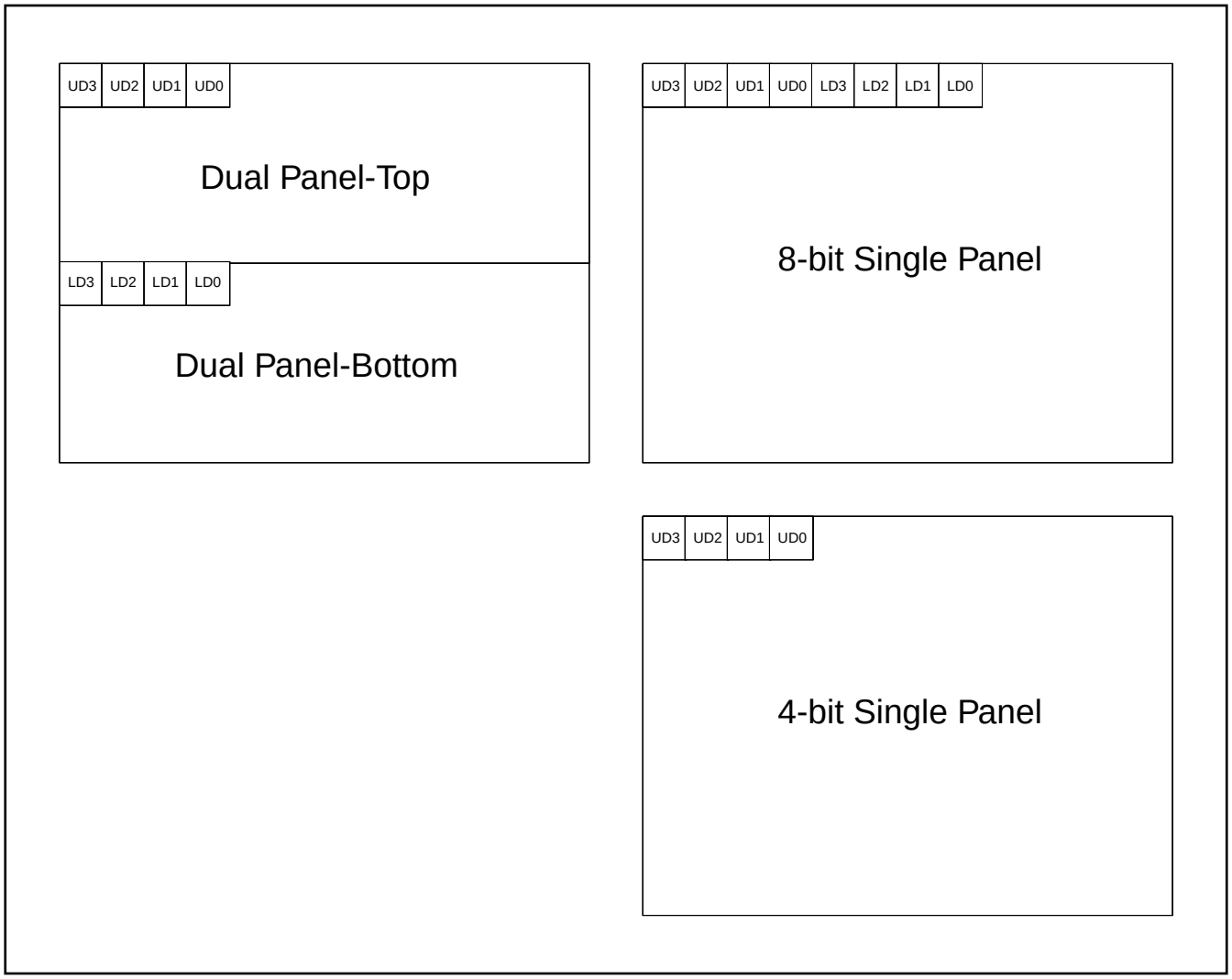


*Figure 18 : LCD Interface Pixel/Data Position*Source: a000083.cdr

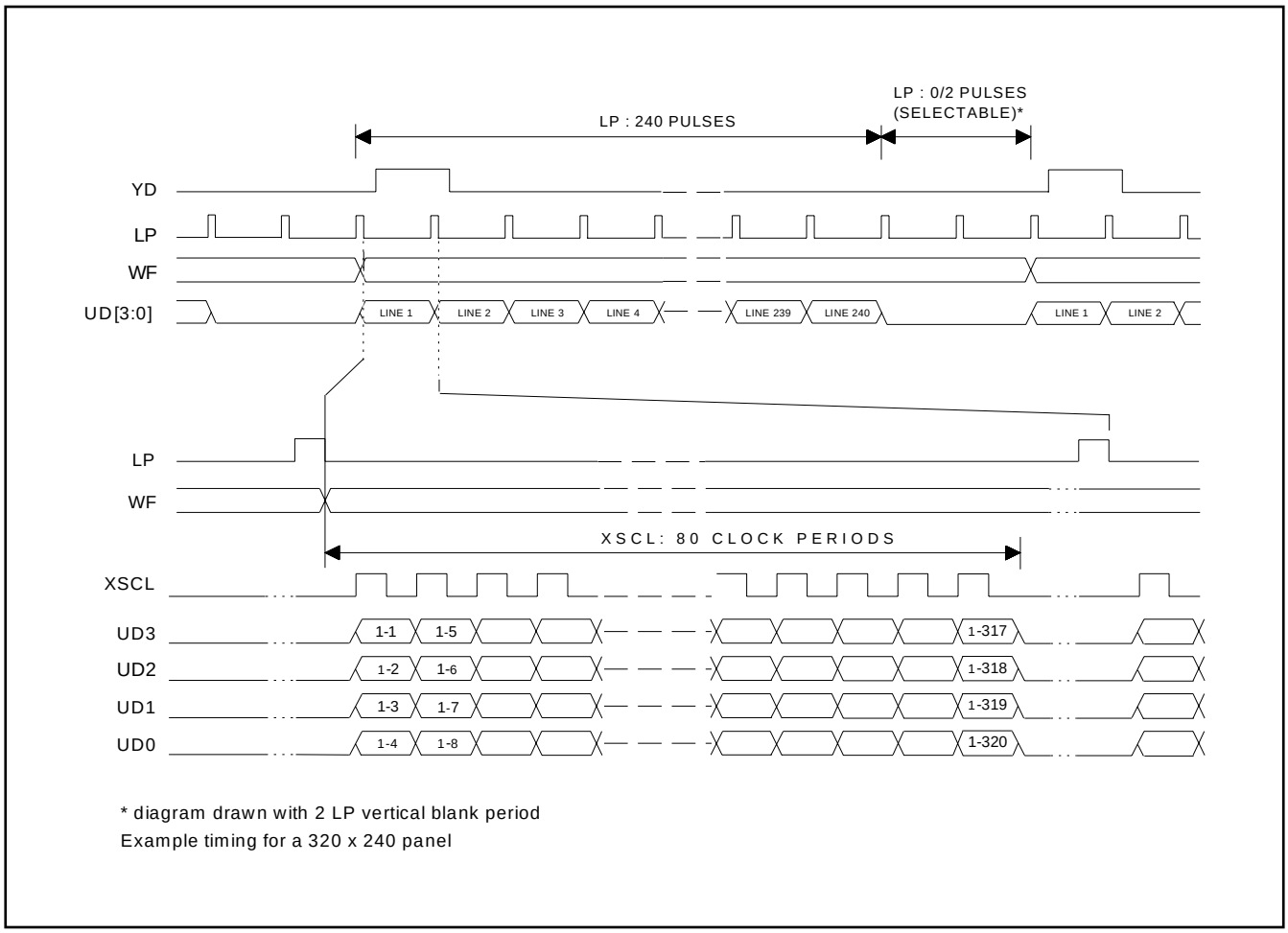


*Figure 19 : 4-bit Single Monochrome Panel Timing*Source: a000018.cdr*

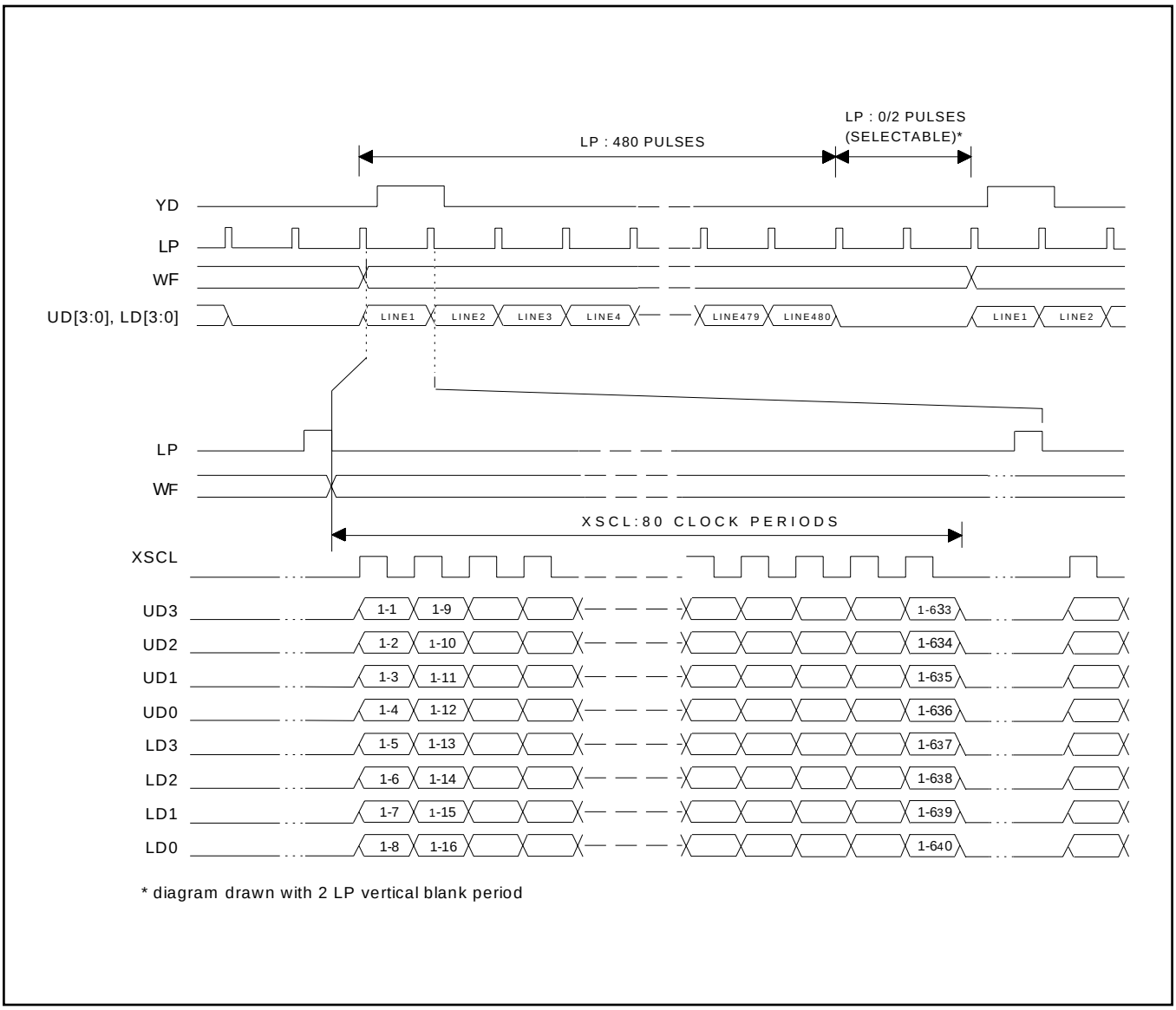


*Figure 20 : 8-bit Single Monochrome Panel Timing*Source: a000019.cdr

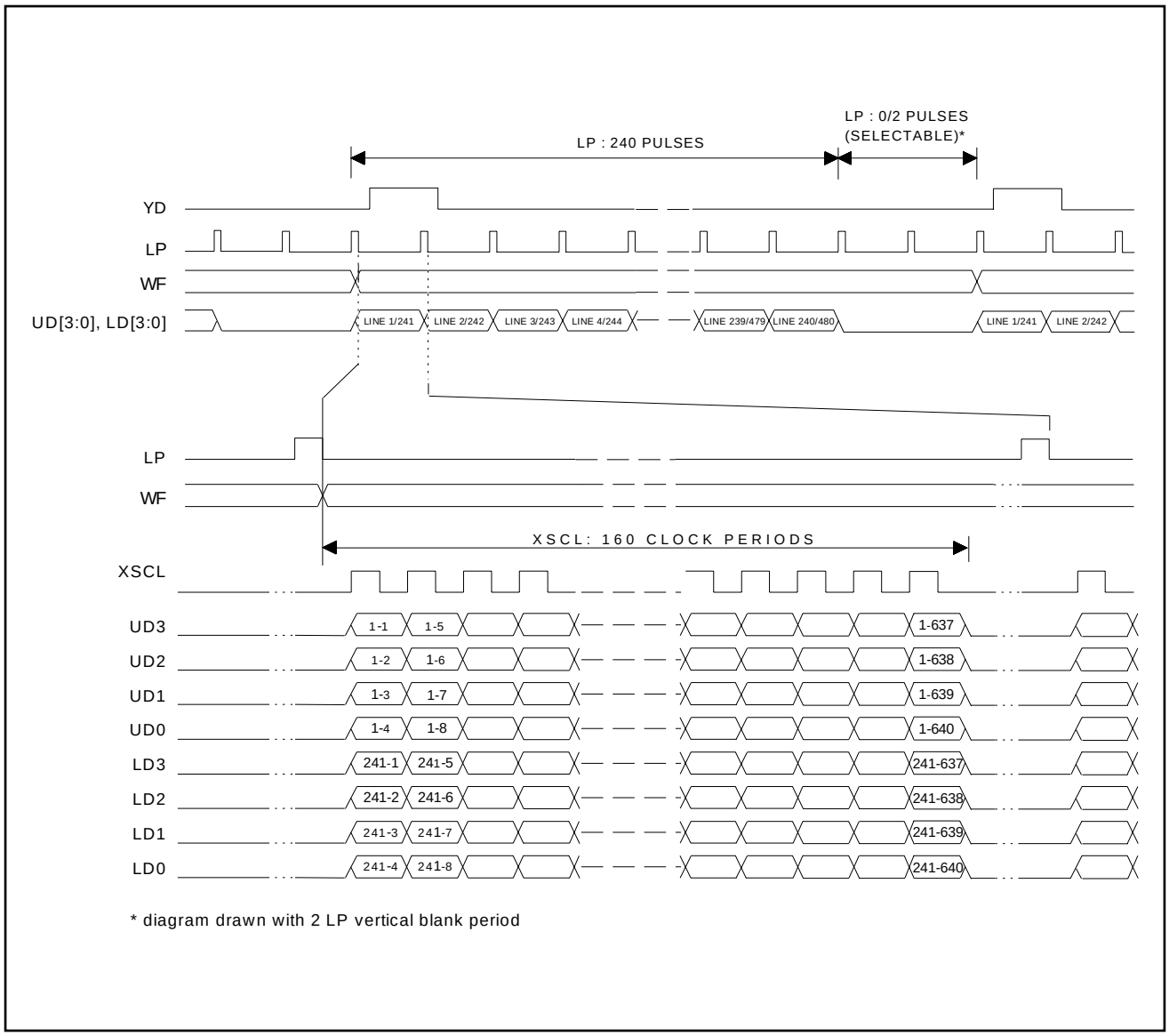


*Figure 21 : 8-bit Dual Monochrome Panel Timing*Source: a000022.cdr

9.0 MIM INTERFACE OPTIONS

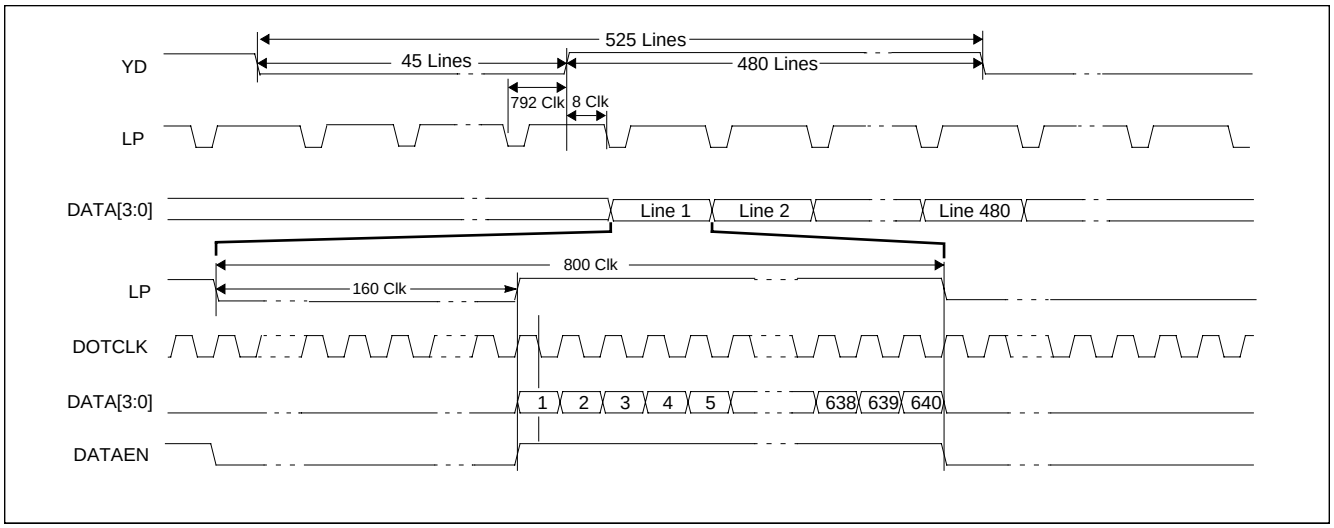


*Figure 22 : MIM Panel Timing*Source: a000084.cdr

10.0 POWER SAVE MODES

10.1 Power Save Modes

To accommodate the important need for power reduction in sub-notebook and palmtop computers, three Power Save Modes, Doze mode 1, Doze mode 2, and Suspend, have been incorporated into the SPC8104. All Power save modes are software-controlled or hardware-controlled. Additional options for these Power Save Modes can be enabled by setting bits in various Auxiliary registers, allowing flexibility in tailoring the power reduction scheme to any particular system implementation.
Suspend mode can be activated through input pin SUSPEND# or by enabling the Suspend Mode Enable bit in the Power Save Register (AUX[03]). There are some small differences between hardware Suspend mode and software Suspend mode. Hardware Suspend mode will override software Suspend mode. Doze mode can be activated through input pin DOZE# or by enabling the Doze Mode Enable bit. The Doze mode select bits determine the function of the Doze mode. Hardware Doze mode and software Doze mode are functionally identical. Whether hardware activated or software activated, Suspend mode always overrides Doze mode. The following descriptions outline the major functions of each power save mode. See the following tables and notes for more information.

Table 0-33 Power Save Mode Selection

Power Save Register Bits				Mode Activated
Suspend Mode Enable	**Doze Mode Enable**	**Doze Mode 2 Select**	**Doze Mode 1 Select**	
0	0	X	X	Normal Operation
0	1	0	1	Doze Mode 1
0	1	1	0	Doze Mode 2
0	1	1	1	Doze Mode 1 and 2
1	X	X	X	Software Suspend Mode

Doze Mode 1

Doze Mode 1 is intended to be used with RAM type LCD driver in order take advantage of its self-refresh mode. This power save mode has an Active State and a Power Down State. Initially the chip enters the Active state which is functionally the same as the normal operation. If there is no memory or I/O write for at least two vertical frames, the output pixels are first re-mapped to black and white. After two more frames with no memory or I/O write, XSCL is stopped to put the LCD driver into its self-refresh mode. In such a self-refresh mode, the binary image stored in the internal RAM of the LCD driver is used to refresh the LCD panel. Various internal blocks in the SPC8104 are also shut down. The chip is now in the Power Down State. Upon detecting a memory or I/O write, the disabled internal blocks are re-enabled and XSCL is reactivated at the beginning of the next frame or the second to next frame. With XSCL reactivated, the LCD driver leaves its self-refresh mode. The chip returns to the Active State and the above sequence repeats.

Active State

- Functionally the same as normal operation.
- Absence of display memory and I/O writes for a long period puts the chip into the Power Down State.

Power Down State

- Static display with black and white only gray shades.
- RAM type LCD driver in self-refresh mode.
- Various internal blocks not required for display refresh are shut down.
- Display memory and I/O read and write allowed, but display memory or I/O write returns the chip to the Active State.

Doze Mode 2

This power save mode has an Active State and a Power Down State. When there are no CPU to display memory access, the chip enters the Power Down State. In the Power Down State, the internal clock rate is reduced. Any display memory read or write request by the CPU returns the chip to the Active State to service the memory access. After the memory access is completed, the chip returns to the Power Down State.

Active State

- Functionally the same as normal operation.
- Completion or absence of memory read/write cycle puts the chip into the Power Down State.

Power Down State

- Internal clock is slowed down by a selectable factor from 2 to 8, while maintaining DRAM refresh rate.
- Static display with reduced frame rate.
- I/O read and write allowed.
- Display memory read or write request by the CPU returns the chip to the Active State.

Software Suspend Mode

- No video display accesses to display memory.
- No CPU accesses to/from display memory.
- Sequencer is halted.
- Display memory refresh is maintained and is generated from one of 3 selectable sources: 1) from the active CLKI input, 2) from the PDCLK pin (32 kHz 50% duty cycle), 3) or from a 64kHz clock source with short active low pulses connected to pin MEMEN (typically -REFRESH of the ISA bus).
- Refresh rate generated can be selected as either 64 kHz or 8 kHz, (for 256 cycle/4 msec, or 256 cycle/32 msec DRAM, respectively).
- I/O read/write of all registers is allowed (except LUT registers).
- LCDPWR# signal forced high.
- LCD interface output signals tri-stated or forced low, depending on the state of the MD[5] configuration pin during chip reset.

Options

- Select MEMEN input pin, PDCLK input pin, or internally divided down CLKI as the clock source for display memory refresh generation.
- Select self-refresh mode, for DRAMs that support self-refresh.
- CLKI can be masked off if it is not used for DRAM refresh.

Hardware Suspend Mode

- No video display accesses to display memory.
- No CPU accesses to/from display memory.
- Sequencer is halted.
- Display memory refresh is maintained and is generated from one of 3 selectable sources: 1) from the active CLKI input, 2) from the PDCLK pin (32 kHz 50% duty cycle), 3) or from a 64kHz clock source with short active low pulses connected to pin MEMEN (typically -REFRESH of the ISA bus).
- Refresh rate generated can be selected as either 64 kHz or 8 kHz, (for 256 cycle/4 msec, or 256 cycle/32 msec DRAM, respectively).
- No I/O register or memory accesses allowed (including LUT).
- LCDPWR# signal forced high.
- LCD interface output signals tri-stated or forced low, depending on the state of the MD[5] configuration pin during chip reset.
- Most CPU interface input signals are internally masked off (i.e. ignored). All CPU interface output signals are inactive (except MEMEN).
- CLKI will automatically masked off if it is not used for DRAM refresh.

Options

- Select MEMEN input pin, PDCLK input pin, or internally divided down CLKI as the clock source for display memory refresh generation.
- Select self-refresh mode, for DRAMs that support self-refresh.

10.2 Power Save Mode Function Summary.

Function	Power Save Mode (PSM)				
	Normal or Active State	Doze Mode 1 Power Down State	Doze Mode 2 Power Down State	Software Suspend	Hardware Suspend
Display Active	Yes	Yes note 16	Yes	No	No
I/O access allowed	Yes	Yes	Yes	Yes, except LUT note 17	No
Memory access allowed	Yes	Yes	Yes	No	No
DRAM refresh maintained	Yes	Yes	Yes	Yes	Yes
Internal LUT active	Yes	Yes	Yes	No	No
Sequencer running	Yes	Yes	Yes	No	No
internal clock rate reduced	No	No	Yes note 14	n/a	n/a
Display refresh circuitry active	Yes	No	Yes	No	No
Refresh generated from CLKI (Sequencer running)	Yes	Yes	Yes	n/a	n/a
Refresh generated from CLKI (Sequencer stopped)	n/a	n/a	n/a	option	option
Refresh generated from MEMEN	No	No	No	option note 1	option note 1
Refresh generated from PDCLK	No	No	No	option note 2	option note 2
Self-refresh	No	No	No	option note 3	option note 3
256cycle/4ms, /32ms refresh selectable	Yes	Yes	Yes	Yes note 1, 2	Yes note 1, 2
LCD Signals					
UD[3:0], LD[3:0]	Active	Inactive	Active	L/HiZ note 13	L/HiZ note 13
YD, LP, WF	Active	Active	Active	L/HiZ note 13	L/HiZ note 13
XSCL	Active	L	Active		
LCDPWR#	L	L	L	H	H
CPU Signals					
LA[23:20], A[19:0], D[15:0], IOR#, IOW#, IOEN#, BHE#, ALE, RESET	active	active	active	active	masked
MEMR#, MEMW#, MEMEN	active	active	active	masked note 15	masked note 15
Clocks					
CLKI	active	active	active	can be masked note 4	can be masked note 5, 6, 7, 8

See “Implementation Notes” following for further details.

10.3 Implementation Notes

1. For Software Suspend Mode and Hardware Suspend Mode using the MEMEN input as the refresh clock source (AUX[02] bit 3-2 = 01b, the clock source from MEMEN should be running at a frequency of 64 kHz. MEMEN's active low pulse width should be as short as possible (but 50 ns greater than the min. DRAM RAS pulse width requirement). The use of a 64 kHz clock source is required for meeting the 256 cycles/4 msec DRAM refresh specification. Normally MEMEN is connected to the ISA bus signal -REFRESH, a 64 KHz clock with an active low period ~ 500 ns. The 64 kHz input can be internally divided down to 8 kHz by setting the 32/4msecRefresh Select bit (AUX[02] bit 0 = 1).
2. For Software Suspend Mode and Hardware Suspend Mode using the PDCLK input as the refresh clock source (AUX[02] bit 3-2 = 11b, the clock source connected to the PDCLK input should be a 32 kHz 50% duty cycle clock. This 32 kHz clock input is doubled internally to a generate a 64 kHz clock source which provides the appropriate duty cycle and active period, as required by the 64 kHz refresh rate for 256cycles/4ms DRAM. The internal 64 kHz refresh rate can further be internally divided down to 8 kHz to support 256cycle/ 32msec DRAM by setting the 32/4msec Refresh Select bit (AUX[02] bit 0 = 1).
3. The self-refresh mode option (AUX[02] bit 3-2 = 10b) available in Software Suspend Mode and Hardware Suspend Mode must only be enabled if the DRAM installed supports self-refresh operation.
4. In Software Suspend Mode, software may set an Auxiliary Register bit to mask off the CLKI clock input. This can be used to further reduce system power consumption.
5. In Hardware Suspend Mode, the CLKI clock input is automatically masked off by hardware if the self-refresh option is enabled.
6. In Hardware Suspend Mode, if MEMEN is selected as the refresh clock source, then the CLKI clock input is automatically masked off by hardware.
7. In Hardware Suspend Mode, if PDCLK is selected as the refresh clock source, then the CLKI clock input is automatically masked off by hardware.
8. In Hardware Suspend Mode, if the CLKI clock input is used as the refresh clock source, then CLKI cannot be masked off by hardware.
9. The output pin LCDPWR# should be used to control the LCD panel's power supply via external circuitry. When LCDPWR# is high, the external panel power supply should be turned off. When LCDPWR# is low, the power supply should be enabled.
10. After RESET is asserted, LCDPWR# is held high until the CRTC is programmed and running (i.e. LCD interface signals are active).
11. Circuitry in the chip will ensure that upon entering Suspend Mode, LCDPWR# will be driven high (panel power shut off) *before* the interface signals are tri-stated or forced low. Upon exiting Suspend Mode, LCDPWR# will be driven low (panel power turned on) *after* the interface signals are returned to their active driving states. This sequencing of the LCDPWR# and interface signals is done to protect the panel from being damaged from DC signals applied to the interface while it is powered up.
12. If the Sequencer is stopped, (Sequencer Reset Register bit 1 or bit 0 = 0), then LCDPWR# will be driven high (panel power shut off) *before* the Sequencer is shut down and the LCD interface signals are halted. Upon restarting the Sequencer (by setting Sequencer Reset Register bit 1 and bit 0 to 1), LCDPWR# will be driven low (panel power turned on) *after* the Sequencer is has started running and the LCD interface signals are returned to their active driving states. This sequencing of the LCDPWR# and interface signals is done to protect the panel from being damaged from DC signals applied to the interface while the Sequencer is stopped and all chip output signals are inactive.
13. Power up configuration pin MD[5] allows selecting the Suspend mode state of the LCD interface signals. In Suspend mode, the LCD interface signals can all be driven low, or can be put into a high-impedance state, as selected by this option.

14. When the internal clock rate is reduced, the DRAM interface and the display interface are running at a slower rate so that power consumption can be reduced. However this may cause the LCD display to flicker. The clock dividing factor should be chosen so that the amount of display flicker is acceptable. The reduced internal clock rate will not affect the DRAM refresh rate. This option is intended to be used with active panels which are less susceptible to varying frame rate.
15. When the MEMEN pin is selected as the refresh clock source, this input will not be masked during Software Suspend Mode and Hardware Suspend Mode.
16. Static display with black and white only gray shades.
17. LUT cannot be accessed if the CLKI disable bit Aux[03] bit 4 is set in Software Suspend mode.

11.0 HARDWARE REGISTER INTERFACE

11.1 Display Modes Supported

Mode No.	Mode Type	Font	Characters	Resolution	Displayed Pixels	Gray Shades	Memory Segment
0	T	8 x 8	40 x 25	320 x 200	640x400	16	B800
0+	T	8 x 14	40 x 25	320 x 350	640x350	16	B800
0++	T	8 x 16	40 x 25	320 x 400	640x400	16	B800
1	T	8 x 8	40 x 25	320 x 200	640x400	16	B800
1+	T	8 x 14	40 x 25	320 x 350	640x350	16	B800
1++	T	8 x 16	40 x 25	320 x 400	640x400	16	B800
2	T	8 x 8	80 x 25	640 x 200	640x400	16	B800
2+	T	8 x 14	80 x 25	640 x 350	640x350	16	B800
2++	T	8 x 16	80 x 25	640 x 400	640x400	16	B800
3	T	8 x 8	80 x 25	640 x 200	640x400	16	B800
3+	T	8 x 14	80 x 25	640 x 350	640x350	16	B800
3++	T	8 x 16	80 x 25	640 x 400	640x400	16	B800
4	G	N/A	N/A	320 x 200	640x400	4	B800
5	G	N/A	N/A	320 x 200	640x400	4	B800
6	G	N/A	N/A	640 x 200	640x400	2	B800
7	T	8 x 14	80 x 25	640 x 350	640x350	2	B000
7+	T	8 x 16	80 x 25	640 x 400	640x400	2	B000
0D	G	N/A	N/A	320 x 200	640x400	16	A000
0E	G	N/A	N/A	640 x 200	640x400	16	A000
0F	G	N/A	N/A	640 x 350	640x350	2	A000
10	G	N/A	N/A	640 x 350	640x350	16	A000
11	G	N/A	N/A	640 x 480	640x480	2	A000
12	G	N/A	N/A	640 x 480	640x480	16	A000

11.2 Standard VGA Register Considerations

This section describes deviations of the SPC8104 from the VGA standard. section on page 55, gives a general description of the ways in which SPC8104 differs from the VGA standard. section on page 55, describes each difference in greater detail. section on page 56, gives a detailed list of the changes to the VGA register set. For a list of all registers actually supported by SPC8104, see section 11.3 on page 59.

Background

There are two ways in which the SPC8104 differs from the VGA standard.

Firstly, the CRTC timing registers differ since the SPC8104 is optimized to support single/dual LCD panels of various sizes. The panel size and timings are determined by four special registers (horizontal display size, vertical display size, horizontal non-display period, and vertical non-display period registers). The CRTC registers of the VGA standard which would normally be used to vary the display monitor timings to other display sizes have been removed and replaced by dummy registers that are read/writable but which have no other effect.

Secondly, there are some functions provided in the VGA register set that are never used in common VGA applications - the register bits for these functions have been removed from the register set supported in SPC8104 in order to optimize the design for VGA operation on a 16 gray-scale 640x480 dot LCD panel. In their place have are read/writable registers bits that have no effect.

General Discussion

CRTC Timing Registers

In the SPC8104, the non-timing related CRTC registers are the same as in the VGA standard. However, to reduce the size (and therefore, cost) of the device, the SPC8104 does not include any of the VGA timing registers that provide full monitor timing programmability since much of the timing circuitry has been fixed in hardware to support LCD panels of various sizes.

The SPC8104 horizontal timing circuitry is programmed by a Horizontal Display Size register and a Horizontal Non-Display Period register to provide flexible horizontal timing. The vertical timing circuitry is programmed by a Vertical Display Size register and a Vertical Non-Display Period register to provide flexible vertical timing. The functions of all timing registers have been omitted; only the VDE End register (index 12h) remains and acts as per the VGA specification. The functions of registers 00h to 06h, 10h, 15h, 16h and certain bits in registers 07h, 09h, 0Bh, and 11h have all been omitted - dummy read/write registers are provided.

Shift-Load, Shift-4, Count-by-2, Count-by-4 modes not supported

Shift-Load and Count-by-2 are normally used only in modes 0Fh and 10h when run on 64kbyte VGA cards. Since the SPC8104 is exclusively a 512kbyte device, this option is not useful. The Shift-4 and Count-by-4 functions are never used in any standard VGA mode or by standard software. No problems are expected arising from the deletion of these unused features. These bits will be removed from the register set.

Support for Maximum 640 Dot LCD Panel Only

The 720 dot text modes (0+,1+,2+,3+) must be reduced to 640 dot modes due to the fixed panel horizontal size. To simplify the design, the logic to select a 9-dot character clock has been removed.The associated register bit in the Sequencer still exists as a read/write bit with no other effect in hardware, since some software expects to use the value set in this bit to calculate screen size.

HRTC/2 mode has been omitted. In standard VGA, this mode was provided to support vertical resolutions greater than 1024.

VGA palette replaced by reduced Gray Scale Lookup Table

The VGA palette has been replaced by a 64x4 bit Gray Scale Lookup Table. This lookup table operates in a similar manner as the VGA palette it replaces, with its registers at the same I/O addresses. An Auxiliary Register control bit is provided to allow protecting the lookup table from writes, if desired.
The LCD controller is designed to display a maximum of 16 absolute shades of grey on a LCD rather than the 16 of 256k colors on a CRT monitor. Therefore a method of mapping colors originally intended for display on a 16/256 color CRT monitor onto a 16 gray LCD panel is required. Hardware is provided to map the RGB (18 bit) values that are written to the lookup table by software expecting to program a regular VGA palette. These RGB values are mapped using 1 of 2 selectable schemes into 4 bit gray scale values that are stored in the lookup table.

Details

This section gives a list of register bits in the SPC8104 that are different to the VGA standard. Only differences are listed here; any bits not mentioned are the same as VGA. For a complete list of the actual SPC8104 registers see section 11.3 on page 59

Misc. Output Reg. (3C2h W, 3CCh R)

bit 1	Enable DRAM Read / write only. No other effect in hardware.
bit 2,3	Clock Select Read / write only. No other effect in hardware.
bits 6,7	+/-Hsyn, Vsyn Read / write only. No other effect in hardware.

Input Status Reg.0 (3C2h R)

bit 4	Switch Sense bit Deleted.

Input Status Reg.1 (3DAh R)

bits 1,2	Light Pen Read Only as 0,1.
bits 4,5	Diagnostic Deleted.

Feature Control Reg. (3DAh W, 3CAh R)

Register not supported.

Sequencer Registers (3C4h,3C5h)

Clocking Mode Reg. (Index 01h)

bit 0	8/9 Dot Clock Read / write only. No other effect in hardware.
bit 1	CPU Bandwidth Read / write only. No other effect in hardware.
bit 2	Shift-Load Read / write only. No other effect in hardware.

bit 4 Shift-4
Read / write only. No other effect in hardware.

Memory Mode Reg. (Index 04h)

bit 1 Ext. Mem.
Read only as 1. Always 256kbytes of memory.

CRTC Registers (3D4h, 3D5h)

Horizontal Total (Index 00h)

Read / write only. No other effect in hardware.

Horizontal Display Enable (Index 01h)

Read / write only. No other effect in hardware.

Horizontal Blanking Start (Index 02h)

Read / write only. No other effect in hardware.

Horizontal Blank End (Index 03h)

Read / write only. No other effect in hardware.

Horizontal Retrace Start (Index 04h)

Read / write only. No other effect in hardware.

Horizontal Retrace End (Index 05h)

Read / write only. No other effect in hardware.

Vertical Total End (Index 06h)

Read / write only. No other effect in hardware.

CRTC Overflow (Index 07h)

bits 0,2,3,5,6,7
Read / write only. No other effect in hardware.

Maximum Scan Line (Index 09h)

bit 5 V.Blank Start 9
Read / write only. No other effect in hardware.

bit 6 Line Compare 9
Read / write only. No other effect in hardware.

Cursor Start (Index 0Ah)

bit 7 IBM Test Bit
Deleted.

Cursor End (Index 0Bh)

bits 5,6 Cursor Skew
Read / write only. No other effect in hardware.

Vertical Retrace Start (Index 10h)

Read / write only. No other effect in hardware.

VRTC End (Index 11h)

bits 0-3 VRTC End
Read / write only. No other effect in hardware.

bit 6 3/5 Refresh
Read / write only. No other effect in hardware.

VUnderline Location(Index 14h)

bit 5 Count-by-4
Read / write only. No other effect in hardware.

bit 6 Double Word Select
Read / write only. No other effect in hardware.

Vertical Blanking Start (Index 15h)

Read / write only. No other effect in hardware.

Vertical Blanking End (Index 16h)

Read / write only. No other effect in hardware.

Mode Control (Index 17h)

bit 2 HRTC/2 Select
Read / write only. No other effect in hardware.

bit 3 Count-by-2
Read / write only. No other effect in hardware.

bit 7 Enable Retraces
Read only as 1. Retraces always enabled.

Attribute Controller Registers (3C0h,3C1h)

Mode Control Register (Index 10h)

bit 2 8>9 Dot Copy
Deleted.

bit 6 Pel Width
Read / write only. No other effect in hardware.

Overscan Register (Index 11h)

bit 7,6 Overscan Color bits 7,6
Read / write only. No other effect in hardware.

Color Plane Enable (Index 12h)

bits 4,5 Video Status MUX. Read only 0,0. IBM diagnostic use only.

Color Select (Index 14h)

bits 3,2 Color Select bits 3,2
Deleted. Not required since only 64x4 lookup table.

Graphics Controller Registers (3CEh,3CFh)

Mode Control Register (Index 05h)

bit 6 256-Color Mode
Read / write only. No other effect in hardware.

DAC Pixel Mask Register (3C6h)

Read / write only. No other effect in hardware.

11.3 I/O Register Summary

This section summarizes the I/O registers of SPC8104 - only those register bits supported by the chip are shown. Note that the functionality of a subset of the IBM VGA standard registers is supported, with an additional set of Auxiliary Registers containing SPC8104 specific functions. Only details of the register functions which are not part of the VGA standard definition are given below.

Unless otherwise noted, all read/write register bits are cleared to 0 after a RESET.

All register bits marked as “n/a” are undefined. There is no effect if they are written to, and reading these bits will return an undefined value.

Auxiliary Registers;

Auxiliary Index/Data Register							
3DEh RW	**3DFh RW**						
n/a	n/a	n/a	n/a	Auxiliary Index bit 3	Auxiliary Index bit 2	Auxiliary Index bit 1	Auxiliary Index bit 0

00 Extended Function Register 0 RW							
16/8 Bit CPU I/O Select	16/8 Bit CPU Memory Select	LUT Read Disable	3C9h Write Disable	n/a	IRQ Output Enable	n/a	LCD B Reg Program Enable

bit 7 — 16/8 Bit CPU I/O Select
When this bit is set to 0, CPU I/O transfer is 8-bit only. When this bit set to 1, CPU I/O transfer can be 8-bit or 16-bit.

bit 6 — 16/8 Bit CPU Memory Select
When this bit is set to 0, CPU memory transfer is 8-bit only. When this bit set to 1, CPU memory transfer can be 8-bit or 16-bit.

bit 5 — LUT Read Disable
When this bit is set to 0, I/O reads to the LUT registers are allowed. When this bit is set to 1, I/O reads to the LUT registers are disabled.

bit 4 — 3C9h Write Disable
When this bit is set to 0, I/O writes to the LUT Data register (3C9h) is allowed. When this bit is set to 1, I/O writes to the LUT Data register (3C9h) is disabled.

bit 2 — IRQ Output Enable
When this bit is set to 0, the IRQ output is held in a high impedance state. When this bit is set to 1, the IRQ output pin is enabled and will be driven to indicate the Vertical Retrace interrupt status.

bit 0 — LCD B Registers Program Enable
This bit is used to access the hidden CRTC B Set Data Registers. When this bit is set to 0, accesses to CRTC register with index 01h, 05h, 10h, and 12h affect the normal CRTC registers. When this bit is set to 1, then the “B set” CRTC registers are enabled, and accesses to CRTC register with index 01h, 05h, 10h, and 12h affect the corresponding “B set” CRTC registers.

01 Output Preference Register RW							
n/a	Graphics VExpand	Text VExpand	Green Only/ NTSC GS Weighting	Graphics Reverse	Text Reverse	Auto-Center Enable	Slow Blink Select

bit 6 Graphics VExpand
This bit is used to enable vertical expansion of 400 line graphics mode on the LCD display. If this bit is 0, then vertical expansion of graphics mode is disabled. If this bit is set to 1, then 400 line graphics modes are expanded vertically to 480 lines to fill the screen. Auto-centering is disabled when graphics mode vertical expansion is in effect. All the following conditions must be true for graphics mode vertical expansion to occur:

1) Graphics VExpand Enable =1

2) 400 line mode set, i.e. CRTC register 12h and associated overflow bits = 18Fh

3) Graphics mode set, i.e. Graphics Controller register 06h bit 0 = 1

bit 5 Text VExpand
This bit is used to enable vertical expansion of 400 line text mode on the LCD display. If this bit is 0, then vertical expansion of text mode is disabled. If this bit is set to 1, then 400 line text modes are expanded vertically to 480 lines to fill the screen. Auto-centering is disabled when text mode vertical expansion is in effect. All the following conditions must be true for text mode vertical expansion to occur:

1) Text VExpand Enable =1

2) 16 point font is set, i.e. the 5 least significant bits of CRTC register 09h = 0Fh

2) 400 line mode set, i.e. CRTC register 12h and associated overflow bits = 18Fh

3) Text mode set, i.e. Graphics Controller register 06h bit 0 = 0

bit 4 Green Only/NTSC GS Weighting Select
The bit is used to select one of two possible gray scale weighting functions to be applied to RGB data as it is written to the internal 64x4 LCD gray scale lookup table. When this bit is set to 0, RGB data values are mapped to gray values using NTSC weighting. When this bit set to 1, the green component of the RGB data is used as the gray value.

bit 3 Graphics Reverse
When this bit is 0, then normal display attributes are enabled. When this bit is 1, then inverse video is displayed on the LCD display when in graphics modes (GRC[06] bit 0 = 1).

bit 2 Text Reverse
When this bit is 0, then normal display attributes are enabled. When this bit is 1, then inverse video is displayed on the LCD display when in text modes (GRC[06] bit 0 = 0).

bit 1 Auto-Center Enable
This bit is used to control the auto-centering function which allows display modes with less than 480 lines to be vertically centered on the panel. When the Auto-Centering Enable bit is 0, then the auto-center function is disabled and all modes will be displayed with the first line at the top of the panel. If this bit is set to 1, then auto-centering is enabled and for all modes hardware will adjust the vertical position of the first line of the display so that the image is centered vertically on the panel.

bit 0 Slow Blink Select
This bit is used to select the blink rate of the cursor and text in text modes and graphics pixels in graphics modes, if blink is enabled. If this bit is set to 0, the cursor, text or graphics pixels blink at their normal rates. If this bit is set to 1, then everything blinks at half their respective normal rates. This option may be used to make the blinking cursor more visible on some LCD panels. Note that in both normal and slow settings, the text mode cursor always blinks at twice the frequency of any blinking characters.

02 LCD Interface Register RW							
n/a	MIM Panel Select	Single/Dual Panel	4/8 Bit Panel Interface	Suspend Refresh Clk Source Select		n/a	32/4 msec Refresh Select
				bit 1	bit 0		

bit 6 MIM Panel Select
When the MIM Panel Select bit is set to 1, MIM panel display hardware is enabled. When this bit is set to 0, LCD panel display hardware is enabled.

bit 5 Single/Dual Panel
The Single/Dual Panel bit is used to configure the chip timing for the correct LCD panel type. When this bit is 0, dual panel mode is enabled. When this bit is 1, single panel mode is enabled.

bit 4 4/8 Bit Panel Interface
The 4/8 Bit Panel Interface bit configures the LCD output data for either 4 bit or 8 bit single panels. When this bit is set to 0, an 8 bit single panel interface is provided, with 8 bit pixel data output on UD[3:0] (msbits) and on LD[3:0] (lsbits). When this bit is set to 1, then a 4 bit single panel interface is provided, where 4 bit pixel data is output only on UD[3:0]. Note that this bit must be set to 0 for all dual panel modes.

bits 3-2 Suspend Mode Refresh Clock Source Select Bits [1:0]
These bits are used to select the clock source used to generate DRAM refresh in Suspend mode as follows

Suspend Refresh Clock Source Select Bit 1	Suspend Refresh Clock Source Select Bit 0	DRAM Refresh Clock Source in Suspend Mode
0	0	CLKI
0	1	MEMEN
1	0	Self Refresh
1	1	PDCLK

CLKI
When this option is selected, the pixel clock input (CLKI) is used as the clock source for DRAM refresh in Suspend mode.

MEMEN
When this option is selected, the MEMEN input is used as the clock source for DRAM refresh in Suspend mode. See note 1 of section 10.3 on page 52, for details

Self Refresh
This option may only be used when the DRAMs installed are capable of self-refresh. When this option is selected, during Suspend mode, the DRAM control lines are driven in such a manner to cause the DRAM to enter self-refresh mode. When not in self-refresh mode, then CAS-before-RAS refresh cycles are used during Suspend mode. Note that regardless of the setting of these bits, CAS-before-RAS refresh cycles are used during Active mode and Doze modes.

PDCLK
When this option is selected, the PDCLK input pin is used as the clock source for DRAM refresh in Suspend mode. This input should be a 32kHz, 50% duty cycle clock. See note 2 of section 10.3 on page 52, for details.

bit 0 32/4 msec Refresh Select
The 32/4msc Refresh Select bit is used to select 256 cycle/4 msec or 256 cycle/32msec DRAM refresh timing in all modes of operation. When this bit is 0, then 4 msec refresh timing is generated. When this bit is 1, then 32msec refresh timing is generated. In active mode and Doze modes, this 4 or 32 msec refresh timing is generated from CLKI. In Suspend mode, this 4 or 32 msec refresh timing is generated from CLKI, from MEMEN input, or the PDCLK input, as selected by AUX[02] bits 3,2.

03 Power Save Register RW							
Doze Mode Enable	Doze Mode 2 Select	Doze Mode 1 Select	CLKI Disable	Doze Mode 2 Clock Divide		Suspend Mode Enable	LCD Power Disable
				bit 1	bit 0		

bit 7 Doze Mode Enable
The Doze Mode Enable bit is used to enable Doze mode, the function of which is determined by the Doze Mode 1 and 2 Select bits. If the SUSPEND# input pin is low or the Suspend Mode Enable bit set, then this bit is ignored.

bit 6 Doze Mode 2 Select
The Doze Mode 2 Select bit enables Doze mode 2 function when the Doze Mode Enable bit is set or when the DOZE# input pin is low. Seesection 10.1 on page 48, for details.

bit 5 Doze Mode 1 Select
The Doze Mode 2 Select bit enables Doze mode 1 function when the Doze Mode Enable bit is set or when the DOZE# input pin is low. See section 10.1 on page 48, for details.

bit 4 CLKI Disable
The CLKI Disable bit is intended to be used in Suspend mode with MEMEN, self-refresh, or PDCLK DRAM refresh clock source option selected (via AUX[02] bit 3, 2). When this bit is 1, CLKI clock source is masked off. When this bit is 0, CLKI clock source is enabled. Note that this bit should never to be set in Active mode or Doze mode.

bits 3-2 Doze Mode 2 Clock Divide Bits [1:0]
These bits select the amount of clock divide of CLKI, as described in the following table, during the power down state of Doze mode 2. These bits have no effect outside Doze mode 2. Note that the clock divide here applies to the CLKI source that may have been affected by Clock Control register bits.

Table 0-34 Doze Mode 2 Clock Division

Doze Mode 2 Clock Divide Bit 1	Doze Mode 2 Clock Divide Bit 0	CLKI Clock Divide Factor
0	0	1 (no divide)
0	1	2
1	0	4
1	1	8

bit 1 Suspend Mode Enable
The Suspend Mode Enable bit is used to enable Software Suspend mode. If the SUSPEND# input pin is low, then this bit is ignored. See section 10.1 on page 48, for details.

bit 0 LCD Power Disable
Setting this bit to 1 forces the LCDPWR# output pin high so that panel power can be turned off.

04 Clock Control Register RW							
n/a	n/a	n/a	Sequencer Scaling	n/a	n/a	n/a	n/a

bit 4 Sequencer Scaling
The Sequencer Scaling bit is used to better support small passive single panels that typically require lower frame rate. When this bit is set, the Sequencer and the pixel output data path are running at half the normal rate so that panel frame rate is effectively halved.

05 Mode Addressing Select Register RW							
n/a	n/a	n/a	n/a	n/a	Sequencer CPU Cycles Disable	3CDh Port Enable	Extended Display Page Enable

bit 3 Sequencer CPU Cycles Disable
Setting this bit to 1 prevents the Sequencer from generating DRAM cycles from a CPU R/W request. Any subsequent memory read cycles will read the contents of the display FIFO. A memory write with this bit set is an invalid cycle and will have no effect.

bit 2 3CDh Port Enable
Setting this bit to 1 enables access to the Page Select Register at port 3CDh. Setting this bit to 0 disables access to this register.

bit 0 Extended Display Page Enable
This bit is used to enable the Extended Display Page function. This function allows the display memory to wrap into the upper 256K bytes, and for 256K bytes of memory to be displayed starting anywhere in the entire 512K byte extended display memory space. If this bit is set to 1, then the extended display page function is enabled. If this bit is set to 0, then the extended display page function is disabled.

06 LCD Support Register RW							
XSCL Enable	LP Timing Select	WF Count					
		bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bit 7 XSCL Enable
This bit is used to adjust the XSCL shift clock output timing. When this bit is set to 0, XSCL is masked off during the horizontal non-display period. When this bit is set to 1, XSCL is not masked off during the horizontal non-display period. Refer to section 7.0 on page 24, and the LCD panel manufacturer's specification to determine the correct setting of this bit.

bit 6 LP Timing Select
This bit is used to adjust the LP latch pulse output timing. When this bit is set to 0, the LP latch pulse falling edge occurs 12 clock periods before the falling edge of the shift clock (XSCL). When this bit is set to 1, the LP latch pulse falling edge occurs 4 clock periods before the falling edge of the shift clock. Refer to section 7.0 on page 24, and the LCD panel manufacturer's specification to determine the correct setting of this bit.

bits 5-0 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. A of value of 0 programmed in these bits causes the WF output to toggle every frame. VAlues of 01h - 3Fh programmed in these bits result in WF toggling every (1 + n) LP pulses.

07 Frame Buffer Start Register RW							
Frame Buffer Start Position							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Frame Buffer Start Position Bits [7:0]
These bits store the 8 most significant bits of the 18-bit start address location for the frame buffer. The lower 10 bits of the frame buffer start address are always 0.

08 Primary Revision Code Register RO							
Primary Revision Code			n/a	n/a	n/a	n/a	Starting Addr. bit 16
bit 2	bit 1	bit 0					

bits 7-5 Primary Revision Code Bits [2:0]
The Primary Revision Code Bits 2 to 0 are read-only bits permanently set to 1. The current revision code of the chip is a combination of the primary and secondary revision code values. The secondary revision code bits are contained in register 0Fh.

bit 0 Starting Address Bit 16
This bit is used to set the most significant display start address bit when utilizing the upper 256K bytes of display memory to provide the Extended Display Page function. Along with the lower 16 bits of start address in CRTC registers 0Ch and 0Dh, this bit allows setting the start address of the image displayed to be anywhere in the 256K address space. For this bit to have an effect, the Extended Display Page enable bit (AUX[05] bit 0) must be set to 1.

09 Frame Buffer End High Register RW							
Frame Buffer End Position High Byte							
bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8

bits 7-0 Frame Buffer End Position Bits [15:8]
These are the 8 most significant bits of the a 16-bit total which determines the end address of the frame buffer. For a given panel size of Width x Height and a given 18-bit frame buffer start address, the 16-bit frame buffer end address is given by the following formula:

End Address = Lower 16 bit of {Start Address + (Height/2) [(Width -1) DIV 16 + 1] - 2}

0A Frame Buffer End Low Register RW							
Frame Buffer End Position Low Byte							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Frame Buffer End Position Bits [7:0]
These are the 8 least significant bits of the a 16-bit total which determines the end address of the frame buffer.

0C Configuration Readback Register RO							
n/a	n/a	n/a	n/a	MD3 Status on Reset RO	MD2 Status on Reset RO	MD1 Status on Reset RO	MD0 Status on Reset RO

bits 3-0 MD[3:0] Status on Reset
These are read-only bits which may be used for inputting power-up reset information to be read by software. On the falling edge of the RESET input, the logic values on the MD[3:0] pins are latched into the chip and then may be read in this register. These bits have no effect in hardware. Internal pullups on these input pins ensure that if nothing is connected externally to these inputs, then these register bits will read 1111.

0D Test / Scratch Register RW							
Test / Scratch Bits							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Test / Scratch Bits [7:0]
For normal operation, the Core Test Enable input pin (TSTCO) is low, and the Test/Scratch register can be used to provide 8 bits of read/write temporary storage. In normal operation mode these bits have no effect on hardware. When the TSTCO input pin is high, the chip is placed in a special test mode, and this register is used to select various internal test functions.

0E Auxiliary Enable Register RW							
n/a	n/a	n/a	Auxiliary Enable Code (1Ah)				
			bit 4	bit 3	bit 2	bit 1	bit 0

bits 4-0 Auxiliary Enable Code Bits [4:0]
The Auxiliary Enable Register can be used to prevent application software from accidentally overwriting the Auxiliary Registers (3DF index 0-F). When disabled, the only Auxiliary Registers that can be accessed are the Auxiliary Index Register (3DEh) and the Auxiliary Enable Register (3DFh index 0Eh). The Auxiliary Registers are disabled by writing any value to the Auxiliary Enable Register (index 0Eh), including the enable code 1Ah (the upper 3 bits of the enable byte are ignored). The Auxiliary Registers are enabled for access after the enable code 1Ah is written to and then read back from the Auxiliary Enable Register.

After a RESET, the Auxiliary Registers are disabled.

0F Secondary Revision Code Register RO							
Product					Revision		
bit 4	bit 3	bit 2	bit 1	bit 0	bit 2	bit 1	bit 0

bits 7-3 Product Code Bits [4:0]
The Product Code bits are read-only bits that are permanently fixed to the product code of the chip. For the SPC8104, the product code is 00010b.

bits 2-0 Revision Code Bits [2:0]
These are the read-only secondary revision code bits that are permanently fixed to the current revision code of the chip. Note that the primary revision code bits 2-0 in the Primary Revision Code Register (index 08h) are always set to 1. For the SPC8104, the secondary revision code is 001b.

Miscellaneous Registers

Miscellaneous Output Register 3C2h W 3CCh R							
Dummy VRTC Polarity	Dummy HRTC Polarity	Odd/Even Page bit	n/a	Dummy Clk Select		Dummy Enable Video RAM	3Dx/3Bx Select
				bit 1	bit 0		

bits 7, 6, 3, 2, and 1 of this register are dummy read/write bits that have no effect in hardware. These read/write bits are provided for compatibility of some software which expects to read valid settings in these VGA standard bits.

Input Status Register 0 **3C2h R**							
CRTC Vertical Interrupt Status	n/a Read as 0	n/a Read as 0	n/a Read as 0	n/a	n/a	n/a	n/a

Input Status Register 1 **3DAh R**							
n/a	n/a	n/a	n/a	Vertical Retrace Status	RO status Read as 1	RO status Read as 0	Display Enable Status

Page Select Register 3CDh RW							
n/a	n/a	n/a	Read Page	n/a	n/a	n/a	Write Page

Accessible only when AUX[05] Bit 1 = 1.

bits 4 Read Page bit
A 0 in this bit selects the lower 256K DRAM memory during CPU memory reads. A 1 in this bit selects the upper 256K DRAM memory during CPU memory reads.

bits 0 Write Page bit
A 0 in this bit selects the lower 256K DRAM memory during CPU memory writes. A 1 in this bit selects the upper 256K DRAM memory during CPU memory writes.

Video Subsystem Enable Register 3C3h RW							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	Video Subsystem Enable

Sequencer Registers

Sequencer Index/Data Register **3C4h RW 3C5h RW**							
n/a	n/a	n/a	n/a	n/a	Sequencer Index bit 2	bit 1	bit 0
00 Reset Register RW							
n/a	n/a	n/a	n/a	n/a	n/a	Sequencer Reset	Sequencer Reset
01 Clocking Mode Register RW							
n/a	n/a	Screen Off	Dummy Shift by 4 (see note)	Dotclock Divide by 2	Dummy Shift Load (see note)	Dummy Bandwidth (see note)	Dummy 8/9 Dot Select (see note)
02 Map Mask Register RW							
n/a	n/a	n/a	n/a	Plane 3 Write Enable	Plane 2 Write Enable	Plane 1 Write Enable	Plane 0 Write Enable
03 Character Map Select Register RW							
n/a	n/a	Map A Select bit 2	Map B Select bit 2	Map A Select bit 1	Map A Select bit 0	Map B Select bit 1	Map B Select bit 0
04 Memory Mode Register RW							
n/a	n/a	n/a	n/a	Chain 4	Odd/Even Map Select	RO Status Read as 1	n/a

index 01 Clocking Mode Register bits 4, 2, 1, 0 - are read/write bits that have no effect in hardware. These bits are provided for compatibility with some software that expects to read or write these bits.

index 00 bits 1, 0 Sequencer Reset bits - setting either or both of these bits to 0 will stop (i.e. reset) the Sequencer. Because stopping the Sequencer halts all interface signals to the LCD panel, logic exists to shut down the panel power (LCDPWR# goes high) before the Sequencer is actually shut down. During this delay between reset request (i.e. setting the reset bits) and actual Sequencer halting, any in-progress memory accesses will be completed. Also, during the actual period that the Sequencer is held in a reset state, if a memory access request is issued by the system, it will be held pending the restart of the Sequencer (while the memory cycle is held pending, the READY output is held low).

Graphics Controller Registers

Graphics Controller Index/Data Register 3CEh RW 3CFh RW							
n/a	n/a	n/a	n/a	Graphics Controller Index			
				bit 3	bit 2	bit 1	bit 0
00 Set/Reset Register RW							
n/a	n/a	n/a	n/a	Set/Reset Plane 3	Set/Reset Plane 2	Set/Reset Plane 1	Set/Reset Plane 0
01 Enable Set/Reset Register RW							
n/a	n/a	n/a	n/a	Enable Set/ Reset Plane 3	Enable Set/ Reset Plane 2	Enable Set/ Reset Plane 1	Enable Set/ Reset Plane 0
02 Color Compare Register RW							
n/a	n/a	n/a	n/a	Reference Color			
				bit 3	bit 2	bit 1	bit 0
03 Data Rotate Register RW							
n/a	n/a	n/a	Logic Function Select		Data Rotate Count		
			bit 1	bit 0	bit 2	bit 1	bit 0
04 Read Map Select Register RW							
n/a	n/a	n/a	n/a	n/a	n/a	Read Plane Select	
						bit 1	bit 0
05 Mode Register RW							
RO Status Read as 0	Dummy 256 Colour Select (see note)	Shift Register Interleave	Odd/Even Plane Select	Read Mode Select	RO Status Read as 0	Write Mode Select	
						bit 1	bit 0
06 Miscellaneous Register RW							
n/a	n/a	n/a	n/a	Display Memory Map		Odd Even Chain Select	Graphics Mode
				bit 1	bit 0		
07 Color Don't Care Register RW							
n/a	n/a	n/a	n/a	Compare Plane Select			
				bit 3	bit 2	bit 1	bit 0
08 Bit Mask Register RW							
Graphics Data Write Mask							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

index 05 Mode Register bit 6 - the Dummy 256 Colour Select bit is a read/write bit that has no effect in hardware. This bit is provided for compatibility with some software that expects to use the value stored in this VGA-defined register bit to determine if mode 13h is set.

Attributes Controller Registers

Attributes Controller Index/Data Register							
3C0h RW 3C0h W 3C1h R							
n/a	n/a	EGA Palette Enable	Attributes Controller Index				
			bit 4	bit 3	bit 2	bit 1	bit 0
00 - 0F EGA Palette Registers RW							
n/a	n/a	Secondary Red	Secondary Green	Secondary Blue	Primary Red	Primary Green	Primary Blue
10 Mode Control Register RW (see note below)							
EGA Palette Bits 4, 5 Control	Dummy Pel Width (see note)	Pixel Pan Compat	n/a	Blink/ Intensity	n/a	Mono Mode Select	Graphics Mode Select
11 Overscan Color Register RW (see note below)							
			Overscan Color				
Dummy bit 7 (see note)	Dummy bit 6 (see note)	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
12 Color Plane Enable Register RW							
n/a	n/a	RO status Read as 0	RO status Read as 0	Enable Plane 3	Enable Plane 2	Enable Plane 1	Enable Plane 0
13 Horizontal Panning Register RW (see note below)							
n/a	n/a	n/a	n/a	Horizontal Pan Count			
				bit 3	bit 2	bit 1	bit 0
14 Color Select Register RW							
n/a	n/a	n/a	n/a	n/a	n/a	Color Select	
						bit 1	bit 0

index 10 Mode Control Register bit 6 - the Dummy Pel Width bit is a read/write bit that has no effect in hardware. This bit is provided for compatibility with some software that expects to use the value stored in this VGA-defined register bit to determine if mode 13h is set.

index 11 Dummy Overscan Color Register bits 7,6 - are read/write bits that have no effect in hardware. These bits are provided for compatibility with some software that expects to read or write these bits.

index 13 Horizontal Panning Register - any pan value > 07h is treated as pan value = 0.

CRT Controller Registers

CRT Controller Index/Data Register 3B4h/3D4h RW 3B5h/3D5h RW							
n/a	n/a	CRT Controller Index					
		bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
00 Dummy Horizontal Total Register RW (see note below)							
Dummy Horizontal Total							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
01 Dummy Horizontal Display Enable End Register RW (see note below)							
Dummy HDE End							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
02 Dummy Horizontal Blanking Start Register RW (see note below)							
Dummy HBlank Start							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
03 Dummy Horizontal Blanking End Register RW (see note below)							
Dummy Read Select	Dummy Display Enable Skew Value		Dummy HBlank End				
	bit 1	bit 0	bit 4	bit 3	bit 2	bit 1	bit 0
04 Dummy Horizontal Retrace Start Register RW (see note below)							
Dummy HRTC Start							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
05 Dummy Horizontal Retrace End Register RW (see note below)							
Dummy HRTC End							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
06 Dummy Vertical Total End Register RW (see note below)							
Dummy VTotal							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
07 CRTC Overflow Register RW							
Dummy VRTC Start bit 9 (see note)	Dummy VDisplay End Position bit 9 (see note)	Dummy VTotal End bit 9 (see note)	Line Compare bit 8	Dummy VBlank Start bit 8 (see note)	Dummy VRTC Start bit 8 (see note)	VDisplay End Position bit 8	Dummy VTotal End bit 8 (see note)
08 Preset Row Scan Register RW							
n/a	Byte Pan		Preset Row Scan				
	bit 1	bit 0	bit 4	bit 3	bit 2	bit 1	bit 0

09 Maximum Scan Line Register RW							
Line Doubling Enable	Dummy Line Comp. bit 9 (see note)	Dummy VBlank Start bit 9 (see note)	Max Scan Line				
			bit 4	bit 3	bit 2	bit 1	bit 0
0A Cursor Start Register RW							
n/a	n/a	Cursor Disable	Cursor Start Row				
			bit 4	bit 3	bit 2	bit 1	bit 0
0B Cursor End Register RW							
n/a	Dummy Cursor Skew (see note)		Cursor End Row				
	bit 1	bit 0	bit 4	bit 3	bit 2	bit 1	bit 0
0C Start Address High Register RW							
Start Address High							
bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0D Start Address Low Register RW							
Start Address Low							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0E Cursor Position High Register RW							
Cursor Position High							
bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
0F Cursor Position Low Register RW							
Cursor Position Low							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
10 Dummy Vertical Retrace Start Register RW (see note below)							
Dummy VRTC Start							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
11 Vertical Retrace End Register RW							
Protect CRTC Registers 0-7	Dummy Refresh Cycles Select (see note)	Vertical Int Disable	Vertical Int Clear	Dummy VRTC End (see note)			
				bit 3	bit 2	bit 1	bit 0
12 Vertical Display Enable End Register RW							
VDisplay End Position							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

13 Offset Register RW							
Offset							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
14 Underline Location Register RW							
n/a	Dummy Double Word Select (see note)	Dummy Count by 4 (see note)	Underline Row Scan bit 4	bit 3	bit 2	bit 1	bit 0
15 Dummy Vertical Blanking Start Register RW (see note below)							
Dummy Vertical Blank Start							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
16 Dummy Vertical Blanking End Register RW (see note below)							
Dummy Vertical Blank End							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
17 Mode Control Register RW							
RO Status Read as 1	Word/Byte Mode Select	MA0 Select MA13/15	n/a	Dummy Count by 2 (see note)	Dummy HRTC/2 (see note)	MA14 Select MA14/rsc	Compat Mode Select
18 Line Compare Register RW							
Line Compare							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
24 Attribute Controller Addr/Data FF Register RO							
Attribute Ctl Index/Data	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Registers and bits marked as “Dummy” are read/write bits that have no effect in hardware. These registers and bits are provided for compatibility with some software that expects to use the values stored in these bits. The Dummy R/W bits are:
index 00, 01, 02, 03, 04, 05, 06, 10, 15, 16 - all bits
index 07 - bits 7, 6, 5, 3, 2, 0
index 09, 0B, 14 - bits 6, 5
index 11 - bits 6, 3, 2, 1, 0
index 17 - bits 3, 2

Index 11 bit 7 Protect CRTC Registers 0-7 - when set to 1, this bit protects the following registers and bits from being written: index 00, 01, 02, 03, 04, 05, 06 - all bits, and index 07 bits 7,6,5,3,2,1,0.

LCD "B Set" Panel Size Registers

01 Horizontal Panel Size Register RW (if AUX[00] B0 = 1)							
Horizontal Panel Size							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Horizontal Panel Size Register Bits [7:0]
The Horizontal Panel Size specifies the horizontal size of the LCD panel in the number of 8-dot characters. For example, a 640 dot panel is programmed with the value 640/÷8 = 50h.

05 Horizontal Non-Display Period RW (if AUX[00] B0 = 1)							
Horizontal Non-Display Period / 8							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Horizontal Non-Display Period Register Bits [7:0]
The Horizontal Non-Display Period specifies the horizontal size of the LCD non-display period in the number of 8-dot characters. This register should NEVER be programmed with a value less than 12 or 0ch. For example, a non-display period of 160 pixels would be programmed the value 160/8 = 14h. This register should be modified to control the frame rate of passive panels.

10 Vertical Non-Display Period RW (if AUX[00] B0 = 1)							
Vertical Non-Display Period in Lines							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Vertical Non-Display Period Register Bits [7:0]
The Vertical Non-Display Period specifies the vertical size of the non-display period in the number of lines. Most passive panels require a value of 2; however the full range of 0 to 255 is possible.

12 Vertical Display Size Register RW (if AUX[00] B0 = 1)							
Vertical Display Size / n (n=2 for single panel, 4 for dual panel)							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Vertical Display Size Register Bits [7:0]
The Vertical Display Size specifies the vertical size of the LCD panel in the number of lines/2 for single panels and the number of lines/4 for dual panels. For example a 480 line single panel is programmed with the value 480/2 = f0h. A 480 line dual panel would be programmed with the value 480/4 = 78h

LCD Gray Scale Lookup Table Registers

Dummy LCD Lookup Table Pixel Mask Register 3C6h RW							
Dummy LUT Pixel Mask							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Dummy Lookup Table Pixel Mask Register Bits [7:0]
This register contains 8 read/write bits that have no effect in hardware. These bits are provided for compatibility with some software that expects to use the value stored in this register.

Lookup Table Status Register 3C7h RO							
n/a	n/a	n/a	n/a	n/a	n/a	Read Write Mode Status	
						bit 1	bit 0

bits 1-0 Lookup Table Status Register Bits [1:0]
Directly after a write to the Lookup Table Read Address Register, the Read/Write Mode Status bits 1-0 are both set to 1. Directly after a write to the Lookup Table Write Address Register, both these bits are set to 0. Note that if the 3c9h Write Disable bit (bit 4 in Auxiliary Register 00) is set to 1, writes to the Lookup Table Data registers (3C9h) are ignored, but in this case the Read/Write Mode Status bits are still set to 0 as if the write did occur.

LCD Lookup Table Read Address Register 3C7h WO							
Lookup Table Read Address							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Lookup Table Read Addr Bits [7:0]
These 8 bits are used to select 1 of 64 gray scale lookup table registers to be read. Note that only the lower 6 bits of this register are used to address reads into the Lookup Table - in other words, reads will wrap every 64 addresses to the physical 64 Lookup Table locations at addresses 00-3Fh.

Once a valid read address is set in this register, the lookup table entry can be read by three successive reads (red, green, blue) of the Lookup Table Data Register 3C9h. After a successful read operation (i.e. three successive reads), the read address is automatically incremented.

Directly after writing an index value to this register, the Lookup Table Status Register Read/Write Mode Status bits 1-0 are both set to 1.

LCD Lookup Table Write Address Register 3C8h RW							
Lookup Table Write Address							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bits 7-0 Lookup Table Write Addr Bits [7:0]
These 8 bits are used to select 1 of 64 gray scale lookup table registers to be written. Note that all 8 write address bits are normally decoded so that the 64 lookup table registers reside at the unique addresses 00 - 3Fh. Any attempted write to lookup table addresses 40h - FFh will be ignored.

Once a valid write address is set in this register, the lookup table entry can be written by three successive writes (red, green, blue) to the Lookup Table Data Register 3C9h. After a successful write operation (i.e. three successive write), the write address is automatically incremented.

Directly after a writing an index value to this register, the Lookup Table Status Register Read/Write Mode Status bits 1-0 are both set to 0.

Note that a write to the Lookup Table Data Registers may only occur if the 3C9h Write Disable bit (bit 4 in Auxiliary Register 00) is set to 0, otherwise the write to the Data Registers will not have an effect. Writes to the Read Address or Write Address registers themselves are not affected by the 3C9 Write Disable bit.

LCD Lookup Table Data Register							
3C9h RW	**(see below)**						
n/a	n/a	Gray Scale Lookup Table Data					
		bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

The data written to the LCD Lookup Table Data Register varies depending on the setting of the Green-only/NTSC Weighting Select bit (AUX[01] bit 4) which selects either a Green-only or an approximated NTSC gray scale weighting scheme:

approx. NTSC Weighting:GS = [5R + 9G + 2B] / 16

Green-only Weighting: GS = G_5 G_4 G_3 G_2 G_1 G_0

The data read from the LCD Lookup Table Data Register always returns a 6-bit data value of which bits 5-2 are the 4-bit gray scale value and bits 1 and 0 are images of bits 5 and 4 (the same as). All three Red, Green, and Blue read accesses return the same value.

GS NTSC Weighting Writes (AUX[01] bit 4 = 0)

1st access (“Red”)							
n/a	n/a	Red Component Data					
		bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
2nd access (“Green”)							
n/a	n/a	Green Component Data					
		bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
3rd access (“Blue”)							
n/a	n/a	Blue Component Data					
		bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

GS Green-Only Weighting Writes (AUX[01] bit 4 = 1)

1st access (“Red”)							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
2nd access (“Green”)							
n/a	n/a	Green Component Data					
		bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
3rd access (“Blue”)							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

GS Weighting Reads

1st access (“Red”)							
n/a	n/a	Gray Scale Value				Read as bit 3	Read as bit 2
		bit 3	bit 2	bit 1	bit 0		
2nd access (“Green”)							
n/a	n/a	Gray Scale Value				Read as bit 3	Read as bit 2
		bit 3	bit 2	bit 1	bit 0		
3rd access (“Blue”)							
n/a	n/a	Gray Scale Value				Read as bit 3	Read as bit 2
		bit 3	bit 2	bit 1	bit 0		

11.4 LCD Gray Scale Lookup Table Architecture

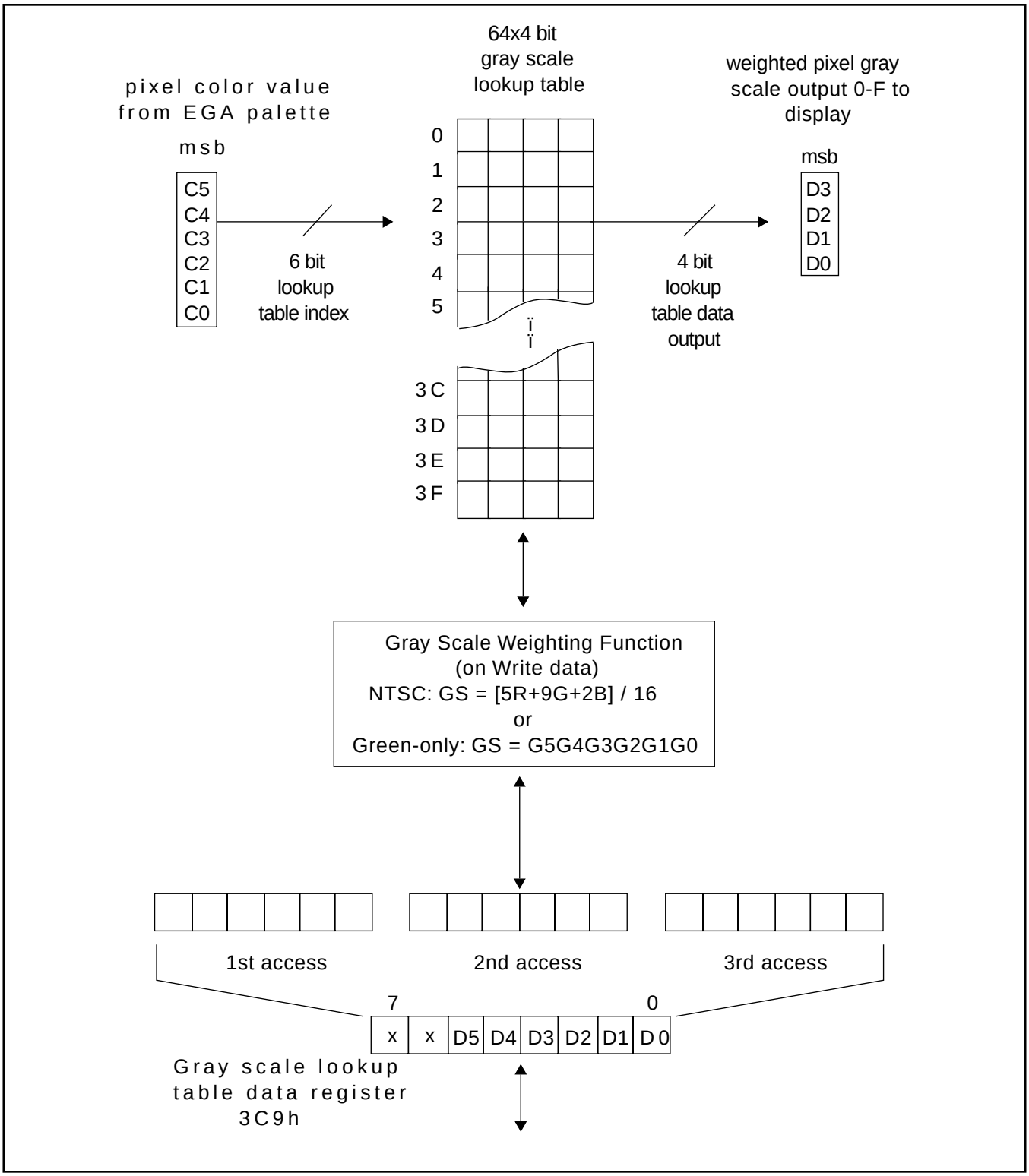
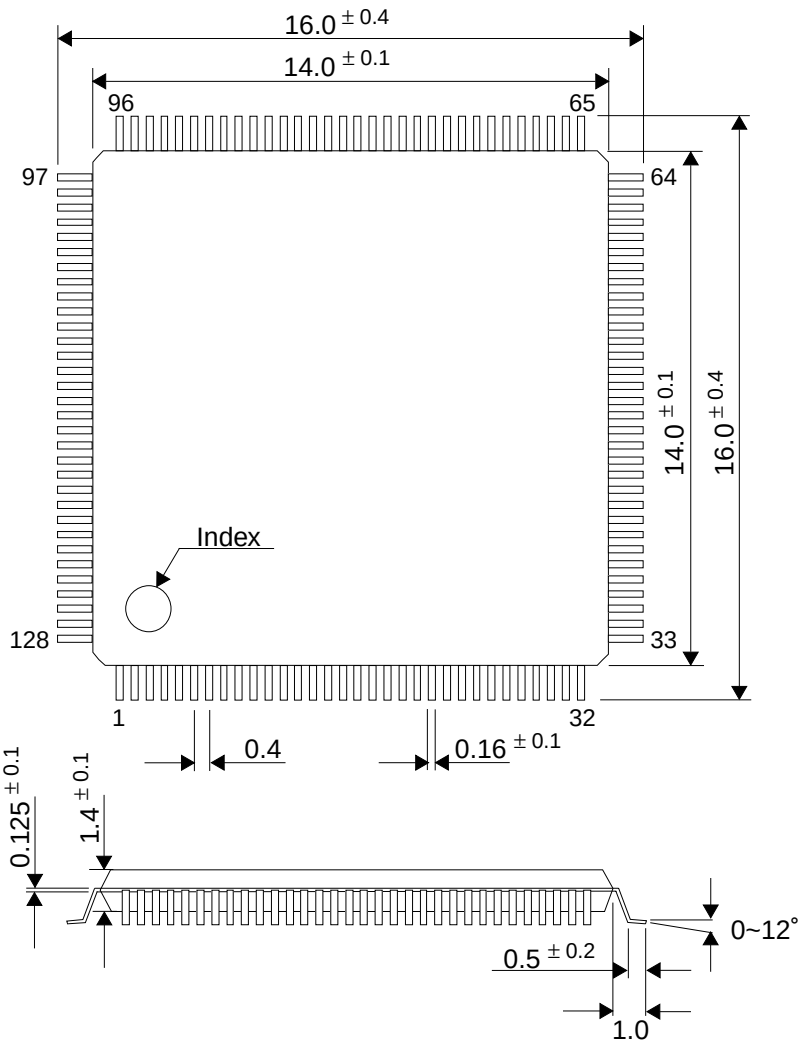


*Figure 23 : LCD Gray Scale Lookup Table Architecture*Source: a000082.cdr

12.0 PACKAGE DIMENSIONS



all dimensions are mm

Figure 24 : Package Dimensions 128 Pin QFP15

THIS PAGE INTENTIONALLY BLANK

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

BIOS Functional Specification

Drawing Office No. X15-SP-002-03

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY BLANK

1.0 INTRODUCTION

1.1 Scope and Objectives

This is the Functional Specification for the SPC8104F0A BIOS. This document specifies the functions, structures and characteristics of the SPC8104F0A BIOS. It is intended for use by persons familiar with VGA BIOS functions and describes differences between the SPC8104F0A and a standard VGA BIOS. In addition this manual documents some of the behaviour characteristics and structure of the BIOS.
This manual consists of several sections which include:

- Features
- Video Modes
- Main BIOS Summary
- VESA Function Summary
- SOLLEX Function Summary
- Physical layout of the BIOS
- and a series of Appendices with chip specific information

2.0 FEATURES

2.1 Technology

- 32 KB EPROM maximum size
- Microsoft MASM 5.1 compatible source

2.2 System and System Compatibility

- ISA bus architecture
- support for VESA VBE Core Functions 1.2 and SOLLEX BIOS extensions
- support for VESA VBE/PM Power Management Functions version 1.0
- programmable power-save modes
- 3C3h video enable register supported
- support for all standard IBM defined VGA modes, including modes 07h and 0Fh, except mode 13h
- selectable gray-scaling
- normal/reverse and autoswitch text/graphics of display polarity support
- autocenter support
- selectable cursor blink rate
- 400 to 475 scan line expansion in text and graphics modes

2.3 Display support

- supports a variety of single/dual monochrome LCD panels of various resolutions from 320 x 200 to 640 x 480, based on MDA line inputs

2.4 Video Modes

SPC8104F0A BIOS Display Modes

Mode No.	Mode Type	Font	Characters	Resolution	Displayed Pixels	Gray Shades	Colors	Memory Segment
0	Text	8 x 8	40 x 25	320 x 200	640 x 400	16	16	B800
0+	Text	8 x 14	40 x 25	320 x 350	640 x 350	16	16	B800
0++	Text	8 x 16	40 x 25	320 x 400	640 x 400	16	16	B800
1	Text	8 x 8	40 x 25	320 x 200	640 x 400	16	16	B800
1+	Text	8 x 14	40 x 25	320 x 350	640 x 350	16	16	B800
1++	Text	8 x 16	40 x 25	320 x 400	640 x 400	16	16	B800
2	Text	8 x 8	80 x 25	640 x 200	640 x 400	16	16	B800
2+	Text	8 x 14	80 x 25	640 x 350	640 x 350	16	16	B800
2++	Text	8 x 16	80 x 25	640 x 400	640 x 400	16	16	B800
3	Text	8 x 8	80 x 25	640 x 200	640 x 400	16	16	B800
3+	Text	8 x 14	80 x 25	640 x 350	640 x 350	16	16	B800
3++	Text	8 x 16	80 x 25	640 x 400	640 x 400	16	16	B800
4	Graphics	N/A	N/A	320 x 200	640 x 400	4	4	B800
5	Graphics	N/A	N/A	320 x 200	640 x 400	4	4	B800
6	Graphics	N/A	N/A	640 x 200	640 x 400	2	2	B800
7	Text	8 x 14	80 x 25	640 x 350	640 x 350	2	2	B000
7+	Text	8 x 16	80 x 25	640 x 400	640 x 400	2	2	B000
0D	Graphics	N/A	N/A	320 x 200	640 x 400	16	16	A000
0E	Graphics	N/A	N/A	640 x 200	640 x 400	16	16	A000
0F	Graphics	N/A	N/A	640 x 350	640 x 350	2	2	A000
10	Graphics	N/A	N/A	640 x 350	640 x 350	16	16	A000
11	Graphics	N/A	N/A	640 x 480	640 x 480	2	2	A000
12	Graphics	N/A	N/A	640 x 480	640 x 480	16	16	A000
108	Text	8 x 8	80 x 60	640 x 480	640 x 480	16	16	B800

3.0 OVERVIEW DESCRIPTION

The SPC8104F0A is a single chip LCD video controller based on VGA architecture. VGA standard mode functionality is supported using standard IBM VGA parameters; the exception to this is mode 13h which is not supported.
The target market for this device is specialized products requiring VGA at less than 256 colors. The ability to run all 16 color VGA software on a 640x480 LCD panel display is the major design consideration, therefore the BIOS must perform the same functions that the IBM VGA BIOS performs within the limitations of the SPC8104F0A. In addition, the BIOS is extended to take advantage of the hardware enhancements in the SPC8104F0A.
The BIOS is divided into 3 major sections: the main BIOS, the VESA Extensions and the Sollex Extensions. It requires 32 KB of EPROM space decoded by the system board.

3.1 Main BIOS

The main BIOS contains the core VGA compatible information. It requires 24 KB of the total 32 KB of EPROM space. This main BIOS is responsible for the initialization of the chip and for performing the IBM compatible function calls. It contains the Video Parameter Tables and the Character Tables.

3.2 VESA Extensions

The VESA Extensions are found in the last 8 KB of the 32 KB. This contains the VESA compatible functions as defined by the Video Electronics Standards Association. At time of printing the BIOS conforms to the Video BIOS Extensions Standard 1.2. These VESA functions are responsible for setting non-IBM modes and supplying functions for mode information. More information on VESA can be obtained by contacting the Video Electronics Standards Association located in San Jose, CA.

3.3 Sollex Extensions

The Sollex Extensions are defined by Seiko Epson Corporation to augment the functionality of the BIOS to include panel and power down functions. This also resides in the last 8 KB. More information on Sollex can be found in the *SOLLEX Specification* Drawing Office No. S03-SP-001-xx.

4.0 MAIN BIOS FUNCTION SUMMARY

4.1 Supported BIOS Functions

These functions are the IBM defined functions that are supported on all VGA compatible products. These functions are supported in the SPC8104F0A BIOS with the noted exceptions. These functions are called using the standard INT 10h interface. To call these functions:

```
MOV AH, function number
MOV other register, other parameters
INT 10H
```

Function 00h - Set Video Mode

```
Input:   AH=00h   Set Video Mode
         AL       Video mode (bit 7 set prevents VRAM clear)
Return:  n/a
```

Function 01h - Set Cursor Type

```
Input:   AH=01h   Set Cursor Type
         CH       Cursor start scan
         CL       Cursor end scan
Return:  n/a
```

Function 02h - Set Cursor Position

```
Input:   AH=02h   Set Cursor Position
         BH       Page number
         DL       Column (0-x)
         DH       Row (0-x)
Return:  n/a
Destroyed:AX,SI
```

Function 03h - Read Cursor Position

```
Input:   AH=03h   Read Cursor Position
         BH       Page number
Return:  CX       Current cursor mode
         DX       Current cursor position
```

Function 04h - Read Lightpen Position (Unsupported On VGA)

```
Input:   AH=04h   Read Lightpen Position
Return:  AH       Lightpen status (0=none, 1=active)
                  If AH=1 then:
                           BX - Pixel column
                           CX - Scan line
                           DX - Character row/column
```

Function 05h - Select Active Display Page

```
Input:   AH=05h   Select Active Display Page
         AL       New page number
Return:  n/a
```

Function 06h - Scroll Active Page Up

```
Input:   AH=06h   Scroll Active Page Up
         AL       Lines to scroll (0=blank window)
         BH       New line(s) attribute
         CX       Top-left corner of scroll window
         DX       Bottom-right corner of scroll window
Return:  n/a
Destroyed:AX, SI, DI, (and DS if text modes)
```

Function 07h - Scroll Active Page Down

```
Input:   AH=07h   Scroll Active Page Down
         AL       Lines to scroll (0=blank window)
         BH       New line(s) attribute
         CX       Top-left corner of scroll window
         DX       Bottom-right corner of scroll window
Return:  n/a
```

Function 08h - Read Character/attribute at Cursor Position

```
Input:   AH=08h   Read Character/attribute at Cursor Position
         BH       Page number
Return:  AL       Character read
         AH       Attribute read
Destroyed:AX, SI, DI, (and DS if text modes)
```

Function 09h - Write Character/attribute at Cursor Position

```
Input:   AH=09h   Write Character/attribute at Cursor Position
         AL       Character to write
         BL       Character attribute/color (b7 set for XOR)
         BH       Page number (Background color in Mode 13)
         CX       Character count
Return:  n/a
Destroyed: AX, SI, DI, (and DS if text modes)
```

Function 0Ah - Write Character Only at Cursor Position

```
Input:   AH=0Ah   Write Character only at Cursor Position
         AL       Character to write
         BH       Page number
         CX       Character count
Return:  n/a
Destroyed: AX, SI, DI, (and DS if text modes)
```

Function 0Bh - Set Color Palette

```
Input:   AH=0Bh   Set Color Palette
         BH=0     (selects background color)
                  BL=0-Fh   Background color
         BH=1     (selects palette)
                  BL=0     (Green, Red, Brown)
                  BL=1     (Cyan, Magenta, White)
Return:  n/a
```

Function 0Ch - Write Dot

```
Input:   AH=0Ch   Write Dot
         AL       Color (b7 set for XOR)
         BH       Page number
         CX       Column
         DX       Row
Return:  n/a
```

Function 0Dh - Read Dot

```
Input:   AH=0Dh   Read Dot
         BH       Page number
         CX       Column
         DX       Row
Return:  AL       Dot color
```

Function 0Eh - Write TTY Character to Active Page

```
Input:   AH=0Eh   Write TTY Character to Active Page
         AL       Character (CR, LF, BS, and BELL accepted)
         BL       Color in graphics mode
Return:  n/a
```

Function 0Fh - Get Current Video State

```
Input:   AH=0Fh   Get Current Video State
Return:  AL       Current video mode
         AH       Number of columns
         BH       Current page number
```

Function 10h - Palette Functions[1]

```
Input:   AH=10h   Palette Functions
         AL=00    Set palette register
         BL       Palette register
         BH       Value to be set
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=01    Set overscan register
         BH       Value to be set
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=02    Set all palette registers and overscan
         ES:DX    Pointer to 17-byte table
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=03    Toggle intensity/blink bit
         BL       1=Blink, 0=Intensity
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=07    Get palette register (VGA)
         BL       Palette register
Return:  BH       Palette register value
```

1. VGA palette reads and writes are mapped in hardware to 16 level grey-scale values. Values for Red, Blue, and Green will depend on the current hardware gray-scale mode.

```
Input:   AH=10h   Palette Functions
         AL=08    Get overscan register (VGA)
Return:  BH       Overscan register value

Input:   AH=10h   Palette Functions
         AL=09    Get all palette registers and overscan (VGA)
         ES:DX    Pointer to 17-byte table
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=10h   Set DAC color register (VGA)
         BX       Color register
         DH:CH:CL Red, Green, Blue data
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=12h   Set block of DAC registers (VGA)
         BX       Start color register
         CX       Number of registers
         ES:DX    Pointer to RGB table
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=13h   Select color page (VGA)
         BL       Paging function (0-1)
                  00       Select paging mode
                  01       Select page
         BH       If BL=0 (0=4 of 64, 1=16 of 16)
                  If BL=1 (Page number 0-3, 0-15)
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=15h   Get DAC color register (VGA)
         BX       Color register
Return:  DH       Red value
         CH       Green value
         CL       Blue value

Input:   AH=10h   Palette Functions
         AL=17h   Get block of DAC registers (VGA)
         BX       Start color register
         CX       Number of registers
         ES:DX    Pointer to RGB table
Return:  n/a
```

```
Input:   AH=10h   Palette Functions
         AL=18h   Set PEL Mask (VGA Undocumented)
         BL       PEL Mask to write
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=19h   Get PEL Mask (VGA Undocumented!)
         BX       Returned PEL Mask value
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=1Ah   Get current color page (VGA)
         BL       Returned paging mode
         BH       Returned page number
Return:  n/a

Input:   AH=10h   Palette Functions
         AL=1Bh   Convert all DAC registers to gray-scale (VGA)
         BX       Start color register
         CX       Number of registers
Return:  n/a
```

Overscan Subfunction 01 incorrectly puts the data into the parameter save area offset + 11h to be consistent with IBM code.

Function 11h - Character Generator Control

```
         AL               Character generator function (0-30h)
                          0x - Alpha load (x=0-4)
                          1x - Alpha load, recalculated (x=0-2, 4)
                          2x - Graphics load (x=0-4)
                          30 - Return information

Input:   AH=11h   Character generator function
         AL=00 or AL=10h (user alpha load):
         BL       Block to load
         BH       Points
         CX       Character count
         DX       Character offset
         ES:BP    Font table pointer
Return:  n/a

Input:   AH=11h   Character generator function
         AL=01 or AL=11h (ROM 8x14 set):
         BL       Block to load
Return:  n/a

Input:   AH=11h   Character generator function
```

```
          AL=02 or AL=12h (ROM 8x8 set):
          BL        Block to load
Return:   n/a

Input:    AH=11h    Character generator function
          AL=04 or AL=14h (ROM 8x16 set):  (VGA only)
          BL        Block to load
Return:   n/a

Input:    AH=11h    Character generator function
          AL=03     (set active block):
          BL        Value for sequencer register 3
Return:   n/a

Input:    AH=11h    Character generator function
          AL=20h     (user graphics characters):
          ES:BP     Font table pointer (chars 128-255)
Return:   n/a

Input:    AH=11h    Character generator function
          AL=21h     (user graphics load):
          BL        Rows select (0=user, 1=14, 2=25, 3=43)
          CX        Points
          DL        Rows Input: BL=0
          ES:BP     Font table pointer
Return:   n/a

Input:    AH=11h    Character generator function
          AL=22h     (ROM 8x14 set):
          BL        Rows select
Return:   n/a

Input:    AH=11h    Character generator function
          AL=23h     (ROM 8x8 set):
          BL        Rows select
Return:   n/a

Input:    AH=11h    Character generator function
          AL=24h     (ROM 8x16 set):  (VGA only)
          BL        Rows select
Return:   n/a

Input:    AH=11h    Character generator function
          AL=30h     (return information):
          BH        Function request (0-7)
                    0 - Get INT 1F pointer
```

```
                    1 - Get INT 43 pointer
                    2 - Get ROM 8x14 pointer
                    3 - Get ROM 8x8 pointer
                    4 - Get ROM 8x8 pointer (128-255)
                    5 - Get 9x14 fudge table pointer
                    6 - Get ROM 8x16 pointer (VGA)
                    7 - Get 9x16 fudge table pointer (VGA)
                    (fudge font never used, but
                    point size returned should be valid)
Return:  CX         Points
         DL         Rows
         ES:BP      Table pointer
```

Function 12h - Miscellaneous Functions

```
Input:   AH=12h     Miscellaneous Function
         BL=10h     Return EGA information
Return:  BL         Memory (0=64K, 1=128K, 2=192K, 3=256K)
         BH         0 = color mode active, 1 = mono mode active
         CL         Switch settings
         CH         Feature bits

Input:   AH=12h     Miscellaneous Function
         BL=20h     Select EGA print screen routine
Return:  n/a

Input:   AH=12h     Miscellaneous Function
         AL=30h     Set alpha mode scan count (VGA)
         AL         0=200, 1=350, 2=400 scans
Return:  AL=12h

Input:   AH=12h     Miscellaneous Function
         BL=31h     Palette load on mode set (VGA)
         AL         0=enable, 1=disable
Return:  AL=12h

Input:   AH=12h     Miscellaneous Function
         BL=32h     Video control (VGA)
         AL         0=enable, 1=disable
Return:  AL=12h

Input:   AH=12h     Miscellaneous Function
         BL=33h     gray-scale summing (VGA)
         AL         0=enable, 1=disable
Return:  AL=12h

Input:   AH=12h     Miscellaneous Function
```

```
          BL=34h   Cursor emulation control (VGA)
          AL       0=enable, 1=disable
Return:   AL=12h

Input:    AH=12h   Miscellaneous Function
          BL=35h   Display control (VGA)
          AL       Function request
                   00 - Adapter off (initial)
                   01 - Planar on (initial)
                   02 - Active off
                   03 - Inactive on
                   ES:DX    Pointer to 128-byte buffer
Return:   n/a

Input:    AH=12h   Miscellaneous Function
          BL=36h   Video data control (VGA)
          AL       0=enable, 1=disable
Return:   AL=12h
```

Function 13h - Write String Functions

```
Input:    AH=13h   Write String function
          AL       String function
                   0 - BL=attribute, string=char, char,...
                   1 - BL=attribute, string=char, char,... cursor moved
                   2 - String=char, attr, char, attr...
                   3 - String=char, attr, char, attr... cursor moved
          BL       Attribute (if AL=0 or 1)
          BH       Page number
          CX       Character count
          DX       Start cursor position
          ES:BP    String pointer
Return:   n/a
```

If any scrolling occurs, the active page will be scrolled, not the requested page! This “feature” is also present in IBM's code and it has been determined that it is better to reproduce this in our code for compatibility.

Function 14h To Function 19h - Null Functions

Functions 14h to 19h are reserved by VGA definition. Any requests of these functions will just simply return and nothing happens.

```
Input:    n/a
Return:   n/a
```

Function 1Ah - Read/Write Display Combination Code

```
Input:    AH=1Ah   Read/write Display Combination Code
          AL       0=read, 1=write
```

```
          BL       Active DCC (r/w)
          BH       Alternate DCC (r/w)
Return:   AL=1Ah
          BX       Set as above
```

Function 1Bh - Return Functionality/State Information

```
Input:    AH=1Bh   Return Functionality/State Information
          BX       Implementation type (must be 00)
          ES:DI    Pointer to buffer
Return:   AL=1Bh
          ES:DI    Contains information
```

The format of the information block is as follows:

Offset	Type	Description
00	DD	Pointer to static functionality table[1]
04	DB	CRT video mode
05	DW	CRT columns
07	DW	VRAM page length
09	DW	VRAM start address
0B 8h x	DW	Cursor row/column for 8 pages
1B	DW	Cursor type
1D	DB	Active page
1E	DW	CRTC address
20	DB	Port 3D8 data
21	DB	Port 3D9 data
22	DB	Number of rows
23	DW	Point size
25	DB	Active DCC
26	DB	Auxiliary DCC
27	DW	Number of colors this mode
29	DB	Number of pages this mode
2A	DB	Number of scans this mode (0=200 1=350 2=400 3=480)
2B	DB	Primary character block
2C	DB	Secondary character block
2D	DB	Misc. state info (see table below)
2E 3h x	DB	Reserved (set to zero)
31	DB	VRAM size (0=64K 1=128K 2=192K 3=256K)
32	DB	Save pointer state info (see table below)
33 Dh x	DB	Reserved (set to zero)

1. This table indicates no support for Mode 13h.

Table 0-1 Information Block Offset 2D

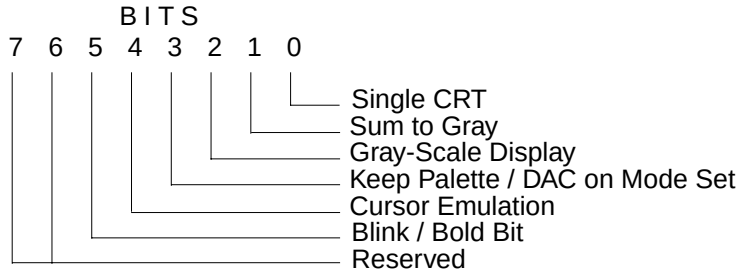
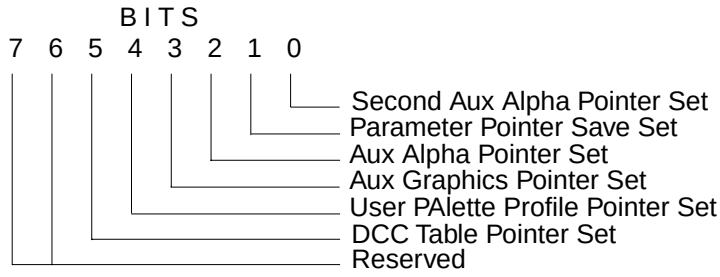


Table 0-2 Information Block Offset 32



Function 1Ch - Save/Restore Video State

```
Input:   AH=1Ch   Save/Restore Video State
         AL       Function request (0-2)
                  00 - Return size of buffer
                  01 - Save video state
                  02 - Restore video state
         ES:BX    Pointer to buffer
         CX       Bit map of request
                  001 - Video h/w status
                  010 - Video data arena
                  100 - Video DAC and color registers
Return:  AL=1Ch
         BX       Buffer size if AL input =00
```

While emulating IBM code this function does not save the latch data or color select register 14h of the Attributes Controller. Because it doesn't get saved or restored, we don't touch it at all.

The format of the save buffer is as follows:

Offsets

00h	DW	Offset to the Video Hardware Status buffer
02h	DW	Offset to the Video Data Arena buffer
04h	DW	Offset to the DAC and Color Registers buffer
06h 0Dh x	DW	Reserved (uninitialized)

Video Hardware Status

00h	DB	Sequencer (3C4) index
01h	DB	CRTC (3D4) index
02h	DB	Graphics controller (3CE) index
03h	DB	Attributes controller (3C0) index
04h	DB	Feature control register (3DA) data
05h 4h x	DB	Sequencer data
09h	DB	Misc. output register (3C2) data
0Ah 19h x	DB	CRTC data
23h 14h x	DB	Attributes controller data
37h 9h x	DB	Graphics controller data
40h	DW	CRTC base port
42h 4h x	DB	Latch data

Video Data Arena

00h	DB	EquipLow AND 30h
01h	DB	CRTMode
02h	DW	CRTColumns
04h	DW	VRAMPageLength
06h	DW	VRAMStartAddr
08h 8h x	DW	CursorRowCol
18h	DW	CursorType
1Ah	DB	ActivePage
1Bh	DW	CRTCAddr
1Dh	DB	Port3D8Data
1Eh	DB	Port3D9Data
1Fh	DB	MaxRow
0h	DW	Points
22h	DB	EGAInfo
23h	DB	DIPInfo
24h	DB	VGAInfo
25h	DB	DCCIndex
26h	DD	SavePtr
2Ah	DD	Vec05
2Eh	DD	Vec1D
32h	DD	Vec1F
36h	DD	Vec43

DAC and Color Registers

```
00h      DB              DAC state (3C7) AND 01h (1=read mode, 0=write mode)
01h      DB              DAC (3C8) index (adjusted)
02h      DB              Pel mask (3C6) data
03h 300h x DB            DAC RGB data
```

5.0 VESA VBE FUNCTION SUMMARY

These functions are defined by the Video Electronics Standards Association. They cover issues of inquiry on chip capability, and available modes. The following shows the SPC8104F0A implementation of these functions. The SPC8104F0A BIOS supports VESA VBE Core Functions 1.2, and VBE/FP Functions 1.0. BIOS versions 2.x, to be specified later, will support VBE Core Functions 2.0, eliminating the need for the Sollex Extensions found in Section 6 of this document.

5.1 Status Information

Every function returns status information in the AX register. The format of the status word is as follows:

```
AL ==    4Fh:    Function is supported
AL !=    4Fh:    Function is not supported
AH ==    00h:    Function call successful
AH ==    01h:    Function call failed
AH ==    02h:    Software supports this function, but the hardware does
                 not
AH ==    03h:    Function call invalid in current video mode
```

5.2 VESA Functions

Function 00h - Return VBE Controller Information

```
Input:   AX      = 4F00h  Return VBE Controller Information
         ES:DI   =        Pointer to buffer in which to place VbeInfoBlock
                          structure (VbeSignature should be set to 'VBE2' when
                          function is called to indicate VBE 2.0 information is
                          desired and the information block is 512 bytes in
                          size.)
Return:  AX      =        VBE Return Status
```

All other registers are preserved.

Function 01h - Return VBE Mode Information

```
Input:   AX      = 4F01h  Return VBE mode information
         CX      =        Mode number
         ES:DI   =        Pointer to ModeInfoBlock structure

Return:  AX      =        VBE Return Status
```

All other registers are preserved.

The mode information block has the following structure:

Function 02h - Set VBE Mode

```
Input:   AX      = 4F02h  Set VBE Mode
         BX      =        Desired Mode to set
```

```
                                D0-D8 = Mode number
                                D9-D13 = Reserved (must be 0)
                                D14 = 0 Use windowed frame buffer model
                                    = 1 Use linear/flat frame buffer model
                                D15 = 0 Clear display memory
                                    = 1 Don't clear display memory
  Return:  AX       =         VBE Return Status
```

All other registers are preserved.

Function 03h - Return current VBE Mode

```
  Input:   AX       = 4F03h   Return current VBE Mode
  Return:  AX       =         VBE Return Status
           BX       =         Current VBE mode
                              D0-D13 = Mode number
                              D14 = 0 Windowed frame buffer model
                                  = 1 Linear/flat frame buffer model
                              D15 = 0 Memory cleared at last mode set
                                  = 1 Memory not cleared at last mode set
```

All other registers are preserved.

Function 04h - Save/Restore state

```
  Input:   AX       =         4F04h Save/Restore state
           DL       =         00h Return save/restore state buffer size
                    =         01h Save state
                    =         02h Restore state
           CX       =         Requested states
                    D0 =      Save/restore controller hardware state
                    D1=       Save/restore BIOS data state
                    D2=       Save/restore DAC state
                    D3=       Save/restore Register state
           ES:BX    =         Pointer to buffer (if DL <> 00h)
  Return:  AX       =         VBE Return Status
           BX       =         Number of 64-byte blocks to hold the state buffer (if
                              DL=00h)
```

All other registers are preserved.

Function 05h - Display Window Control

```
  Input:   AX       = 4F05h   VBE Display Window Control
           BH       = 00h     Set memory window
                    = 01h     Get memory window
           BL       =         Window number
                    = 00h     Window A
                    = 01h     Window B
           DX       =         Window number in video memory in window granularity
                              units (Set Memory Window only)
  Return:  AX       =         VBE Return Status
```

	DX	=	Window number in window granularity units (Get Memory Window only)

Function 06h - Set/Get Logical Scan Line Length

Input:	AX	= 4F06h	VBE Set/Get Logical Scan Line Length
	BL	= 00h	Set Scan Line Length in Pixels
		= 01h	Get Scan Line Length
		= 02h	Set Scan Line Length in Bytes
		= 03h	Get Maximum Scan Line Length
	CX	=	If BL=00h Desired Width in Pixels If BL=02h Desired Width in Bytes (Ignored for Get Functions)
Return:	AX	=	VBE Return Status
	BX	=	Bytes Per Scan Line
	CX	=	Actual Pixels Per Scan Line (truncated to nearest complete pixel)
	DX	=	Maximum Number of Scan Lines

Function 07h - Set/Get Display Start

Input:	AX	= 4F07h	VBE Set/Get Display Start Control
	BH	= 00h	Reserved and must be 00h
	BL	= 00h	Set Display Start
		= 01h	Get Display Start
		= 80h	Set Display Start during Vertical Retrace
	CX	=	First Displayed Pixel In Scan Line (Set Display Start only)
	DX	=	First Displayed Scan Line (Set Display Start only)
Return:	AX	=	VBE Return Status
	BH	=	00h Reserved and will be 0 (Get Display Start only)
	CX	=	First Displayed Pixel In Scan Line (Get Display Start only
	DX	=	First Displayed Scan Line (Get Display Start only)

Function 08h - Set/Get DAC Palette Format

```
Input:   AX      = 4F08h  VBE Set/Get Palette Format
         BL      = 00h    Set DAC Palette Format
                 = 01h    Get DAC Palette Format
         BH      =        Desired bits of color per primary
                          (Set DAC Palette Format only)
Return:  AX      =        VBE Return Status
         BH      =        Current number of bits of color per primary
```

Function 09h - Set/Get Palette Data

```
Input:   AX      = 4F09h  VBE Load/Unload Palette Data
         BL      = 00h    Set Palette Data
                 = 01h    Get Palette Data
                 = 02h    Set Secondary Palette Data
                 = 03h    Get Secondary Palette Data
                 = 80h    Set Palette Data during Vertical Retrace with Blank
                          Bit on
         CX      =        Number of palette registers to update
         DX      =        First palette register to update
         ES:DI   =        Table of palette values (see below for format)
Return:  AX      =        VBE Return Status
```

Format of Palette Values: Alignment byte, Red byte, Green byte, Blue byte

Function 0Ah - Return VBE Protected Mode Interface

```
Input:   AX      = 4F0Bh  VBE 2.0 Protected Mode Interface
         BL      = 00h    Return protected mode table
Return:  AX      =        Status
         ES      =        Real Mode Segment of Table
         DI      =        Offset of Table
         CX      =        Length of Table including protected mode code
                          (for copying purposes)
```

The format of the table is as follows:

```
ES:DI + 00h      Word Offset in table of Protected mode code for
                 Function 5 for Set Window Call
ES:DI + 02h      Word Offset in table of Protected mode code for
                 Function 7 for set Display Start
ES:DI + 04h      Word Offset in table of Protected mode code for
                 Function 9 for set Primary Palette data
ES:DI + 06h      Word Offset in table of Ports and Memory Locations
                 that the application may need I/O privilege for.
                 (Optional: if unsupported this must be 0000h)
                 (See Sub-table for format)
ES:DI + ?        Variable remainder of Table including Code
```

Function 10h - Display Power Management Extensions

The VESA VBE sub-Function 10h is used to implement the VBE/PM services. The VBE/PM services are defined as follows:

Sub-Function 00h - Report VBE / PM Capabilities

Input:	AH	= 4Fh	VESA Extension.
	AL	= 10h	VBE/PM Services.
	BL	= 00h	Report VBE/PM Capabilities.
	CX	= 00h	Controller unit number (00 = primary controller).
	ES:DI		Null pointer, must be 0000:0000h in version 1.0. Reserved for future use.
Return:	AX	=	Status.
	BH	=	Power saving state signals supported by the controller. 1 = supported, 2 = not supported bit 0 STANDBY bit 1 SUSPEND bit 2 OFF bit 3 REDUCED ON bits 4-7 reserved for future power control of the display controller or other related circuits.
	BL	=	VBE/PM version number (0001 0000b for this version). bits 0-3 minor version number bits 4-7 major version number
	CX	=	Unchanged
	ES:DI		Unchanged

All other registers may be destroyed.

Sub-Function 01h - Set Display Power State

Input:	AH	= 4Fh	VESA Extension.
	AL	= 10h	VBE/PM Services.
	BL	= 01h	Set Display Power State.
	BH	= 00h	ON
		= 01h	STANDBY
		= 02h	SUSPEND
		= 04h	OFF
		= 08h	REDUCED ON
		All other BH values are currently undefined and are reserved for future power control of the display controller.	
	CX	= 00h	Controller unit number (00 = primary controller).
Return:	AX	=	Status.
	BH	=	Unchanged
	CX	=	Unchanged

All other registers may be destroyed.

Sub-Function 02h - Get Display Power State

Input:	AH	= 4Fh	VESA Extension.

	AL	= 10h	VBE/PM Services.
	BL	= 02h	Get Display Power State.
	CX	= 00h	Controller unit number (00 = primary controller).
Return:	AX	=	Status.
	BH		Power state currently requested by the controller.
		= 00h	ON
		= 01h	STANDBY
		= 02h	SUSPEND
		= 04h	OFF
		= 08h	REDUCED ON
		All other BH values are reserved and may be used to signal other power saving states in future revisions of VBE/PM. For future compatibility, applications written for VBE/PM 1.0 should ignore the value of bits 4 to 7.	
	CX	=	Unchanged

All other registers may be destroyed.

6.0 SOLLEX FUNCTION SUMMARY

These functions are defined by Seiko Epson as a generic interface for functions not covered by the Video Electronics Standards Association or IBM's standard video BIOS. This specification is soon to be outdated, as VESA is now in the process of developing Flat Panel, Power Management and Display Data Channel specifications. With version 2.x of the SPC8104 BIOS, the Sollex functions will be removed, to be replaced by the VESA specifications. The following is incomplete since much of the functions that Sollex was designed for have been removed. Only functions 00h, 01h, 02h, 04h, 05h, 07h, 08h, 0Ah and FFh are valid for the SPC8104F0A.

6.1 Sollex Status Information

Every function returns status information in the AX register. The format of the status words is as follows:

```
AL ==    7Fh:     Function is supported
AL !=    7Fh:     Function is not supported
AH ==    00h:     Function call successful
AH ==    01h:     Function call fails
```

6.2 Sollex Reserved Bits

All reserved bit returns will return 0 by default, unless otherwise noted.

6.3 Sollex Functions

Function 00h - Return Extensions Info

Not Applicable for SPC8104F0A

Function 01h - Adapter Control

```
Input:   AH=7Fh   SOLLEX Support
         AL=01h   Adapter Control
         BL=00h   Set Adapter
         CX       Adapter Request
Return:  AX       Status

Input:   AH=7Fh   SOLLEX Support
         AL=01h   Adapter Control
         BL=01h   Get Adapter
Return:  AX       Status
         BX       Adapter type
         DX       Display type
```

```
Input:   AH=7Fh   SOLLEX Support
         AL=01h   Adapter Control
         BL=02h   Return Adapter Support
         CX       Adapter Request
Return:  AX       Status
```

Table 0-3 Sub-Function 00h: Set Adapter

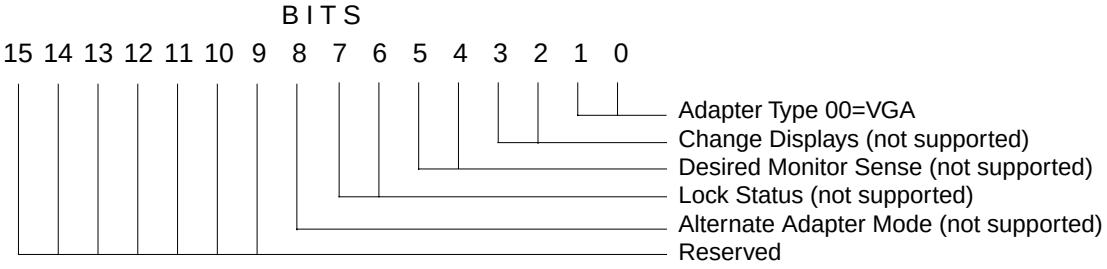
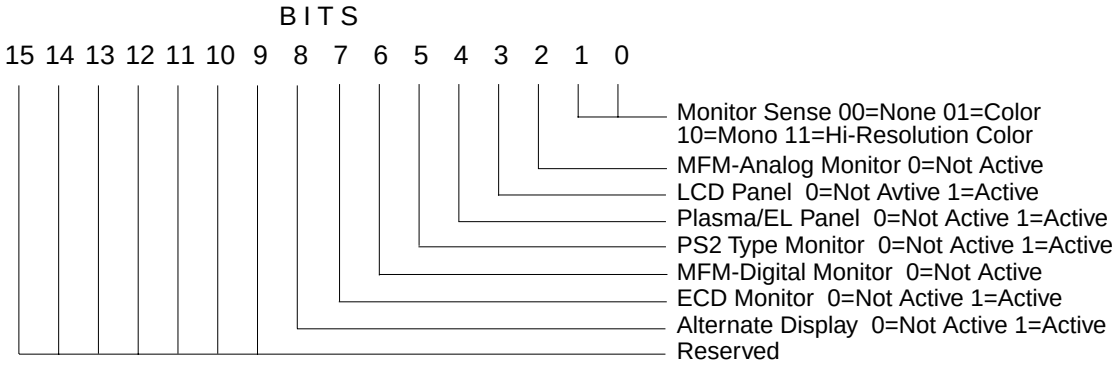


Table 0-4 Sub-Function 01h: Get Adapter



The Monitor Sense returns the folowing:

```
DX = DISPLAY-LCD
BX = ADAPTER_VGA
```

Sub-Function 02h: Return Adapter Support

Sub-Function 02h uses the same Adapter Request format as Sub-Function 00h to determine whether the requested adapter setting could be successfully set in the current environment.

Function 02h - Display Output Control

```
Input:   AH=7Fh   SOLLEX Support
         AL=02h   Display Output Control
         BL=00h   Set Display Output
         CX       Display Setting
Return:  AX       Status

Input:   AH=7Fh   SOLLEX Support
         AL=02h   Display Output Control
         BL=01h   Get Display Output
Return:  AX       Status
         BX       Display Output
         CX       Displays attached
```

Table 0-5 Sub-Function 00h: Set Display Output

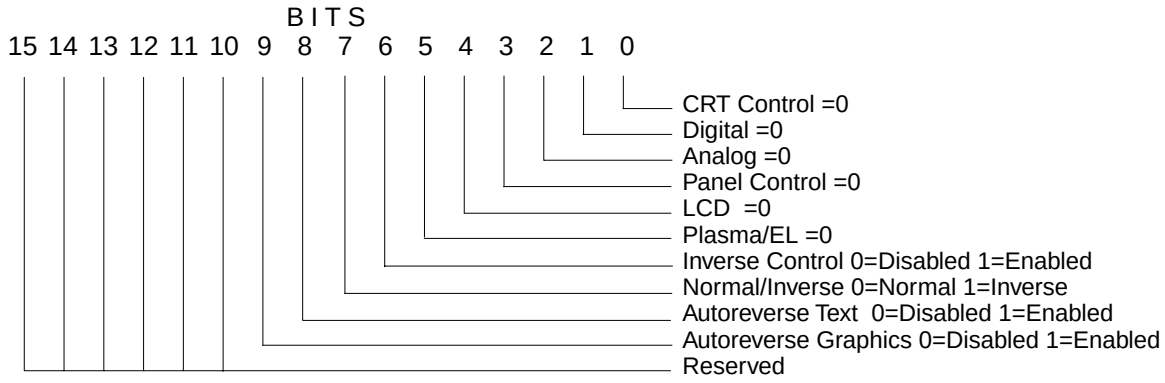
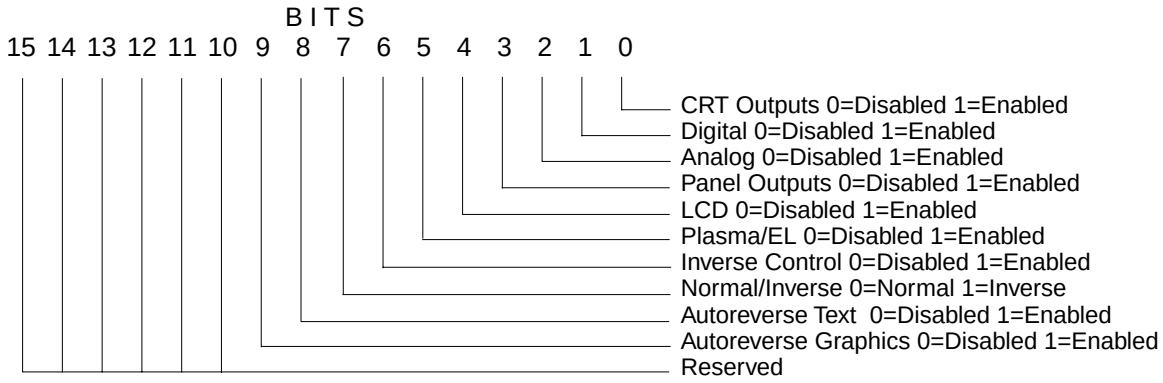
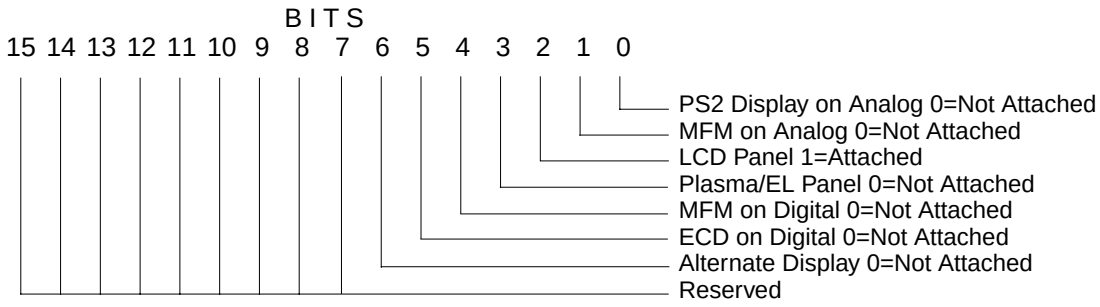


Table 0-6 Sub-Function 01h: Get Display Output



The CX register contains the following bit mask that describes the attached displays:

Table 0-7 Display Attached



Function 03h - Video Support Control

Not Applicable for SPC8104F0A

Function 04h - Power Control

```
Input:   AH=7Fh   SOLLEX Support
         AL=04h   Power Control
         BL=00h   Set Power State
         CX       Power State (0 to Maximum State)
Return:  AX       Status

Input:   AH=7Fh   SOLLEX Support
         AL=04h   Power Control
         BL=01h   Get Power State
Return:  AX       Status
         CX       Power State
         DX       Maximum State

Input:   AH=7Fh   SOLLEX Support (UNSUPPORTED in SPC8104F0A)
         AL=04h   Power Control
         BL=02h   Set Time Out Reset
         CX       Time Out Reset (0 to Maximum Time Out Reset)
Return:  AX       Status AL=7Fh, AH=01h

Input:   AH=7Fh   SOLLEX Support (UNSUPPORTED in SPC8104F0A)
         AL=04h   Power Control
         BL=03h   Get Time Out Reset
Return:  AX       Status AL=7Fh, AH=01h
```

Sub-Function 00h: Set Power State

Set Power State according to table below:

Sub-Function 01h: Get Power State

Returns Power State according to table below:

	State 0 (Normal Operation)	State 1 (Doze 1)	State 2 (Doze 2)	State 3 (Suspend)
Slow Pixel Clock	No	No	Yes	Stopped
Reduce Palette	No	Yes	Yes	N/A
Display off	No	No	No	Yes
Display Memory Read/Write	Yes	Yes	Yes	No
Backlight Off	No	No	No	Yes
Relative Power Saving	None	Low	High	Highest

Function 05h - Load Register

Not Applicable for SPC8104F0A

Function 06h - Multiple Font Control

Not Applicable for SPC8104F0A

Function 07h - Fill Video RAM

Not Applicable for SPC8104F0A

Function 08h - Autocenter Control

```
Input:   AH=7Fh   SOLLEX Support
         AL=08h   Autocenter control
         BL=00h   Set Autocenter control
         CX       0000h Disable Autocenter
                  0001h Enable Autocenter
Return:  AX       Status

Input:   AH=7Fh   SOLLEX Support
         AL=08h   Autocenter control
         BL=01h   Get Autocenter state
Return:  AX       Status
         BL       Autocenter control status
                  00h=Disabled
                  01h=Enabled
```

Function 09h - Lookup Table Control

Not supported in SPC8104F0A.

Function 0Ah - Non-Standard Font Control (SPC8104F0A uses 19pt.font for h/w text expansion*)

```
Input:   AH=7Fh   SOLLEX Support
         AL=0Ah   Non-Standard Font Control
```

```
            BL=00h   Set Non-Standard Font Control state
            CL       Font Width
            CH       Font Height (if CX=0, it will set normal system
                     font to be default)
    Return: AX       Status

    Input:  AH=7Fh   SOLLEX Support
            AL=0Ah   Non-Standard Font Control
            BL=01h   Get Non-Standard Font Control state
    Return: AX=      Status
            BL=      Font Width
            BH=      Font Height (normal system font will return proper
                     values not 0 as in the set)
            ES:DI    pointer to table of available fonts
                     (format width, height, width, height ... 00, 00)
```

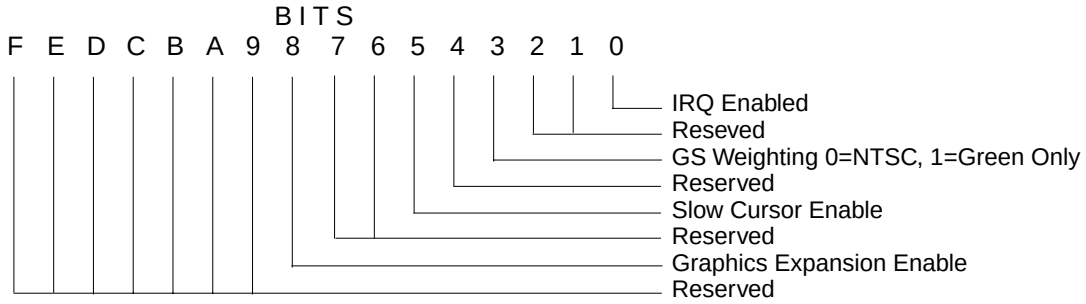
* use 19 point for set to enable Hardware Text Expansion

Function FFh - Chip Specific Function

```
Input:   AH=7Fh   SOLLEX Support
         AL=FFh   Chip Specific Function
         BL=00h   Set
         CX       Requested State
         DH       Graphic expansion line number (if D8 of CX is 1)
Return:  AX       Status

Input:   AH=7Fh   SOLLEX Support
         AL=FFh   Chip Specific Function
         BL=01h   Get
Return:  AX=      Status
         BH=      Chip Revision Product Code
         BL=      Chip Revision
         CX=      Current State
         DH=      Graphic expansion line number (if D8 of CX is 1)
```

Table 0-8 Chip Specific Function



7.0 PHYSICAL LAYOUT OF THE BIOS

7.1 Structure

The BIOS will exist in a 32 KB section of ROM located at C000 or any other configurable location. Segment references within the code will self define at initialization time, however an initial BIOS segment must be configured into the ROM image.
The layout of this segment is as follows:

Initialization Routine	Function Dispatch	Function Calls	Video Parameter Table	Character Fonts 8, 14, 16, & 18 or 19 if needed	Extensions Initialization	Extensions Dispatch	Extensions Function Calls	Extensions Parameter Tables

7.2 IBM Notice, Copyright Notice

Some software expects the word 'IBM' to be located at C000:001E to determine if it is running on an EGA or better. We have initially put the word 'IBM' at this location, but, if the BIOS is relocated to another segment, E000 for example, software that does check at C000:001E will not work. There is very little chance of running across current software with this test.
The BIOS will also contain the string “Copyright (c) Seiko Epson Corp. 1987, 1995. All rights reserved” in two locations. One of these must stay in the code, the other is part of the header that is displayed on power-up. The power-up string is configurable and can be overwritten by the OEM.

8.0 SMALL FORM FACTOR PANEL SUPPORT ISSUES

The SPC8104F0A BIOS has been made to be as flexible as possible. To aid software development on small panel sizes, we have incorporated special mode numbers for the most common vertical resolutions so drivers for programs such as Windows can use the panel appropriately. In addition to these changes, there are many limitations to running a small panel that should be mentioned.
Running VGA software on a small panel poses a couple of problems. These problems are viewing area and software compatibility. In addition, special non-standard IBM modes must be supported to aid in application development.

8.1 Special Mode Support

To support each small panel, the panel type must be selected using the MD0-MD3, and MD9 pin polarities as an index into a list of supported panels. Seiko Epson has chosen the most common vertical resolutions on small panels and designed planar graphics modes for these. They are as follows:

Resolution	Mode No.
200	41H
240	42H
256	43H
320	44H
400	45H

Once a vertical panel resolution is selected, the BIOS will set the physical panel size and will try to accommodate running standard IBM modes on this panel. The BIOS modifies standard CRTC timings and line doubling to get as much information on the small panel as possible for these IBM standard modes. In addition the BIOS also supplies the planar graphics modes listed above to help an OEM develop custom software. Note that the horizontal width of these resolutions defaults to 640 pixels wide, which can be adjusted after the mode is set.

Viewing Area

The major problem with small form factor panels, is that IBM defined its VGA modes around two basic resolutions: 640 x 480 and 720 x 400. To get the information presented in these modes onto the smaller panels, we have some workarounds which are different for Text and Graphics modes, however each has its limitations.

Text Modes

Standard VGA text mode displays 80 x 25 characters on a 720 x 400 monitor. To display the same information on a small panel, we force 720 text modes to 640 using an 8-dot clock; unfortunately in the horizontal this is all we can do, therefore if a panel is smaller than 640 dots wide we will truncate information. However, our solutions for getting more information in the vertical direction is much more flexible. For panel sizes less than 400 lines, we have chosen to use 8 pixel high characters instead of the regular 16 pixel high characters, so that at least 25 lines can be visible on the screen. For panel sizes 400 lines or greater, we can still use the normal 16 pixel high font and see everything on the screen.

Graphics Modes

As with text modes, the horizontal display will be clipped. But, for panels less than 400 lines, we have modified the modes which normally double 200 lines to 400 scan lines back to an undoubled 200, so that vertically they will display correctly. For modes that are a full 480 scan lines, the BIOS cannot undouble the mode, so the will be clipped in the vertical direction.

Software Compatibility

A major concern when modifying standard mode values (including font and offset information) is the software that does not compensate for these changes. Most text mode software expects a standard VGA font to be present (16 pixel high) and may not expect this to change. Programs that modify the text cursor, or program their own font probably will not work correctly. In addition, most DOS software expects text mode to be 80 characters wide and if the Horizontal Offset is modified (in the CRTC) , the display will get a “barber pole” effect, instead of being clipped.

Programming and Driver Considerations

Most OEMs using the SPC8104F0A on smaller than VGA standard panels will be doing so for custom applications. To aid in the development of these custom applications, we recommend the following:

- Set the appropriate mode number for your panel height as specified above.
- Set the CRTC Offset register 3D4h index 13h to your appropriate width.

The mode set will act like Mode 12h (planar graphics) in memory planes, colors and memory addresses and will make it very easy for OEMs to develop software in Mode 12h on a standard VGA to port to this new environment. For graphics mode applications, remember to modify the offset on your standard Mode 12h simulated environment to be the same as on the custom environment you have just set, then when you are ready to move it to your new platform, change the mode set from Mode 12h to a VESA Mode 4xh function call.
For those running Windows 3.1, we have produced a Windows driver that will take input from SYSTEM.INI to set the appropriate mode and then read the horizontal size from this system.ini command line. For more information on running Windows on a small panel see Windows 3.1 Panning Display Driver, Drawing Office No. X15-DI-001-xx.

Appendix A BIOS CONFIGURATION INPUTS

A.1 Panel Type

There are 16 BIOS configuration inputs in the SPC8104F0A. The panel type configuration inputs come in on MD0-3 and are read back in Auxiliary Register Index 0Ch bits 1-4.

Pin Name	Configuration Pin Functionality
MD0	B0 of panel table
MD1	B1 of panel table
MD2	B2 of panel table
MD3	B3 of panel table

Panel Configuration Bits

All panels default to 78 Hz frame rate.

bit 3	bit 2	bit 1	bit 0	Resolution	Mode	Bits	Comments
1	1	1	1	640x480	MIM	4	MIM (landscape)
1	1	1	0	640x480	Single	8	Monochrome
1	1	0	1	640x200	Single	4	Monochrome
1	1	0	0	320x240	Single	4	Monochrome
1	0	1	1	Reserved			
1	0	1	0	Reserved			
1	0	0	1	Reserved			
1	0	0	0	Reserved			
0	1	1	1	Reserved			
0	1	1	0	Reserved			
0	1	0	1	640x480	Dual	8	Monochrome
0	1	0	0	480x640	MIM	4	MIM (portrait)
0	0	1	1	Reserved			
0	0	1	0	Reserved			
0	0	0	1	Reserved			
0	0	0	0	OEM Config			

Appendix B POWER SAVE MODE DESCRIPTIONS

To accommodate the important need for power reduction in sub-notebook and palmtop computers, three Power Save Modes, Doze mode 1, Doze mode 2, and Suspend, have been incorporated into the SPC8104. All Power save modes are software-controlled or hardware-controlled. Additional options for these Power Save Modes can be enabled by setting bits in various Auxiliary registers, allowing flexibility in tailoring the power reduction scheme to any particular system implementation.

Suspend mode can be activated through input pin SUSPEND# or by enabling the Suspend Mode Enable bit in the Power Save Register (AUX[03]). There are some small differences between hardware Suspend mode and software Suspend mode. Hardware Suspend mode will override software Suspend mode. Doze mode can be activated through input pin DOZE# or by enabling the Doze Mode Enable bit. The Doze mode select bits determine the function of the Doze mode. Hardware Doze mode and software Doze mode are functionally identical. Whether hardware activated or software activated, Suspend mode always overrides Doze mode.

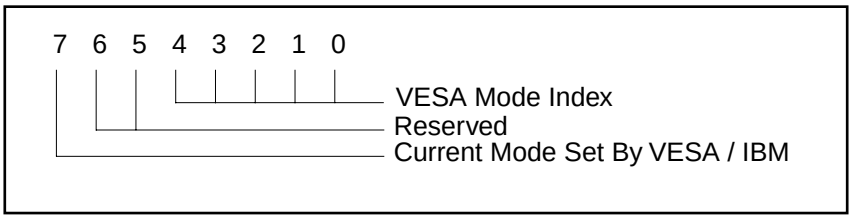
	State 0 (Normal Operation)	**State 1 (Doze 1)**	**State 2 (Doze 2)**	**State 3 (Suspend)**
Slow Pixel Clock	No	No	Yes	Stopped
Reduce Palette	No	Yes	Yes	N/A
Display off	No	No	No	Yes
Display Memory Read/Write	Yes	Yes	Yes	No
Backlight Off	No	No	No	Yes
Relative Power Saving	None	Low	High	Highest

Appendix C BIOS USE OF INTERNAL SCRATCH BITS

The BIOS uses scratch bits to store information, which has no equivalent in a standard IBM compatible BIOS. Changing a bit state has no immediate effect on the video state, scratch bits are typically processed at a mode set.

These scratch registers should not be programmed directly, using SOLLEX calls is recommended instead, as bit locations may change without notice.

Auxiliary Register D



VESA Mode Index

This is an index into a table of possible VESA modes. If all bits are zero, no VESA mode is currently active. The remaining seven combinations are indexes into a table of supported VESA modes. This table has the following values: 00 = no VESA mode, 01 = first VESA mode as returned by function 0, 02 = second VESA mode as returned by function 0, 03 = third VESA mode, etc. as returned by function 0.

Appendix D INITIALIZATION

(640x480 Single Color 8-Bit Display)

Init code for SPC8104F0A 640x480 dual mono panel. This code assumes 24MHz input clock and no power down clock. This code also assumes a possibility of an additional video card (CGA/MDA) to be present.

1. Turn off the active video system:

```
mov     ax,1201h
mov     bl,32h
int     10h
```

2. Enable the SPC8104 chip:

```
3c3h = 1
```

3. Set the default SPC8104 CRTC address to 3Dxh to prevent any conflict with a possible monochrome card:

```
3C2h = 67h
```

4. Enable the AUX registers

```
AUX[DEh] = 1Ah        ; read AUX[DEh] data (read port 3DFh).
```

5. Program the AUX registers to the required default values.

```
AUX[00h] = 00000000b  ; 8bit I/O, 8 bit memory
AUX[01h] = 00000011b  ; autocenter+slow blink
AUX[02h] = 00000000b  ; LCD interface register
AUX[03h] = 00001000b  ; Doze mode 2 Clock divide
AUX[04h] = 00000000b  ; clear clock register
AUX[05h] = 00000000b  ; disable memory paging etc.
AUX[06h] = 00100000b  ; LCD WF register = 20h
AUX[07H] = 10000000b  ; frame buffer start at 256k
AUX[08h] = 00000000b  ; bit 0 = Start Addr. B16 (R/W)
AUX[09h] = 00100101b  ; fr.buffer end high byte
AUX[0Ah] = 01111101b  ; fr.buffer end low byte
AUX[0Dh] = 00000000b  ; clear scratch register
```

6. Set video memory to A000h to prevent conflicts with MDA/CGA cards by programming the graphics controller.

```
Graphics Controller[06h] = 04h
```

7. Do CGA/MDA detection. If no card is found, the SPC8104 can switch into 16 bit memory and I/O mode.

```
AUX[00h] = 11000000b
```

   If a CGA card is found, we have to change the CRTC addressing to 3Bxh

8. Program the panel geometry. This is done by programming the CRTC_B set of registers at 3Dx or 3Bx.

```
AUX[00h] |= 00000001b ; enable CRTC_B space (or in bit 0 = 1)
CRTC[01h] = 640/8     ; CRTC_B[1] = 640/8 = 80 (dec)
CRTC[05h] = 12        ; horz. non-display period (for max frame rate)
CRTC[10h] = 2         ; vert. non-display periof = 2 lines
CRTC[12h] = 480/2     ; 8 MSB of 480
AUX[00h] &= 11111110b ; disable CRTC_B addressing (clear AUX[00h] bit 0)
```

9. At this point all non-standard registers have been initialized. They need not be programmed again (except for power-downs). Continue with standard inititialization:

   set up interrupt vectors 42h,10h,6dh ... set mode 3 (or 7 if CGA present)

10. Print signon message

Appendix E VESA INFORMATION

E.1 VESA Structures

The VBE 1.2 compatible information block has the following structure:

```
VGAInfoBlock struc
    VESASignature     db'VESA'; 4 signature bytes
    VESAVersion       dw102h; VESA version number
    OEMStringPtr      dd?   ; Pointer to OEM string
    Capabilities      dd?   ; capabilities of the video environment
    VideoModePtr      dd?   ; pointer to supported VESA modes
    TotalMemory       dw?   ; Number of 64kb memory blocks on board
    Reserved          db236 dup (0); Remainder of VGAInfoBlock
VGAInfoBlock ends
```

The VBE 2.0 information block has the following structure:

```
VbeInfoBlock struc
    VbeSignature      db'VESA'; VBE Signature
    VbeVersion        dw0200h; VBE Version
    OemStringPtr      dd?   ; Pointer to OEM String
    Capabilities      db4 dup (?); Capabilities of graphics cont.
    VideoModePtr      dd?   ; Pointer to Video Mode List
    TotalMemory       dw?   ; Number of 64kb memory blocks
                            ; Added for VBE 2.0
    OemSoftwareRev    dw?   ; VBE implementation Software revision
    OemVendorNamePtr  dd?   ; Pointer to Vendor Name String
    OemProductNamePtr dd?   ; Pointer to Product Name String
    OemProductRevPtr  dd?   ; Pointer to Product Revision String
    Reserved          db222 dup (?); Reserved for VBE implementation scratch
                            ; area
    OemData           db256 dup (?); Data Area for OEM Strings
VbeInfoBlock ends
```

The mode information block has the following structure:

```
ModeInfoBlock   struc
; Mandatory information for all VBE revisions
    ModeAttributes    dw  ? ; mode attributes
    WinAAttributes    db  ? ; window A attributes
    WinBAttributes    db  ? ; window B attributes
    WinGranularity    dw  ? ; window granularity
    WinSize           dw  ? ; window size
    WinASegment       dw  ? ; window A start segment
    WinBSegment       dw  ? ; window B start segment
    WinFuncPtr        dd  ? ; pointer to window function
    BytesPerScanLine  dw  ? ; bytes per scan line
; Mandatory information for VBE 1.2 and above
    XResolution       dw  ? ; horizontal resolution in pixels or
                            ; characters
    YResolution       dw  ? ; vertical resolution in pixels or
                            ; characters
    XCharSize         db  ? ; character cell width in pixels
    YCharSize         db  ? ; character cell height in pixels
    NumberOfPlanes    db  ? ; number of memory planes
    BitsPerPixel      db  ? ; bits per pixel
    NumberOfBanks     db  ? ; number of banks
    MemoryModel       db  ? ; memory model type
    BankSize          db  ? ; bank size in KB
    NumberOfImagePagesdb  ? ; number of images
    Reserved          db  1 ; reserved for page function

; Direct Color fields (required for direct/6 and YUV/7 memory models)
    RedMaskSize       db  ? ; size of direct color red mask in bits
    RedFieldPosition  db  ? ; bit position of lsb of red mask
    GreenMaskSize     db  ? ; size of direct color green mask in bits
    GreenFieldPositiondb  ? ; bit position of lsb of green mask
    BlueMaskSize      db  ? ; size of direct color blue mask in bits
    BlueFieldPosition db  ? ; bit position of lsb of blue mask
    RsvdMaskSize      db  ? ; size of direct color reserved mask in bits
    RsvdFieldPosition db  ? ; bit position of lsb of reserved mask
    DirectColorModeInfodb ? ; direct color mode attributes
; Mandatory information for VBE 2.0 and above
    PhysBasePtr       dd  ? ; physical address for flat memory
                              frame buffer
    OffScreenMemOffsetdd  ? ; pointer to start of off screen memory
    OffScreenMemSize  dw  ? ; amount of off screen memory in 1k units
    Reserved          db  206 dup (?0; remainder of ModeInfoBlock
ModeInfoBlock ends
```

E.2 VESA Mode Support

E.2.1 VESA Graphics Modes

15-bit Mode Number	7-bit Mode Number	Resolution	Colors	Supported
100h	-	640x400	256	No
101h	-	640x480	256	No
102h	6Ah	800x600	16	No
103h	-	800x600	256	No
104h	-	1024x768	16	No
105h	-	1024x768	256	No
106h	-	1280x1024	16	No
107h	-	1280x1024	256	No

E.2.2 VESA Text Modes

15-bit Mode Number	7-bit Mode Number	Columns	Rows	Supported
108h	-	80	60	Yes
109h	-	132	25	No
10Ah	-	132	43	No
10Bh	-	132	50	No
10Ch	-	132	60	No

E.2.3 VESA Graphics Modes

15-bit Mode Number	7-bit Mode Number	Resolution	Colors	Supported
10Dh	-	320x200	32K (1:5:5:5:)	No
10Eh	-	320x200	64K (5:6:5)	No
10Fh	-	320x200	16.8M (8:8:8)	No
110h	-	640x480	32K (1:5:5:5:)	No
111h	-	640x480	64K (5:6:5)	No
112h	-	640x480	16.8M (8:8:8)	No
113h	-	800x600	32K (1:5:5:5:)	No
114h	-	800x600	64K (5:6:5)	No
115h	-	800x600	16.8M (8:8:8)	No
116h	-	1024x768	32K (1:5:5:5:)	No
117h	-	1024x768	64K (5:6:5)	No
118h	-	1024x768	16.8M (8:8:8)	No
119h	-	1280x1024	32K (1:5:5:5:)	No
11Ah	-	1280x1024	64K (5:6:5)	No
11Bh	-	1280x1024	16.8M (8:8:8)	No

Appendix F SPECIAL CASES

1. BIOS mode set 13h.
   On an attempt to set mode 13h, the BIOS will fail to set the mode and remain in the current mode.
2. BIOS DAC support.
   Writes to the RAMDAC will be processed as though the RAMDAC actually existed. All BIOS functions on reads will read BUS value when appropriate.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

Programming Notes and Examples

Drawing Office No. X15-AN-003-02.1

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY BLANK

1.0 INITIALIZING THE SPC8104 IN MODE 03 WITHOUT A BIOS

The purpose of this document is to demonstrate how to initialize the SPC8104F0A into a mode similar to IBM's Standard VGA mode 3 by supplying sample code. The steps involved in this are:

- Enabling the SPC8104
- Unlocking the Auxiliary Port
- Setting up the SPC8104 panel specific registers
  In this case for a Dual STN, Monochrome 640x480 LCD panel
- Setting up the Standard VGA registers for Mode 3 (with the exception of only loading an 8 point font versus the 16 point font)
- Loading the font into Memory (required for text mode only)
- Initializing the Palette values
- Setting up the attribute registers
- Initializing Video Memory

The resulting mode is not BIOS supported, so no DOS text will work, however, writing to the memory location B800:0000 values 41 07 will put a white letter A in the top left corner.

```
; MODE3.ASM
;
; Programs the SPC8104 for dual 640x480 LCD panels.
;
; Sets up registers for mode 3 without the BIOS.
; The code ignores power down issues.
; The code assumes 4ms memory refresh clock.
;
;*****************************************************************
;
; Instructions:
;
;  - Make your changes
;  - Assemble       --  masm MODE3.asm;
;  - Link       --  link MODE3;
;  - Make a .COM file  --  exe2bin MODE3.exe MODE3.com
;
;
; Note:
;       This program was tested by MS MASM 5.1
;
;*****************************************************************

PW    EQU     8000h           ; Port write (RunTable)


TEXT SEGMENT WORD PUBLIC 'TEXT'
ASSUME CS:TEXT, DS:TEXT, SS:TEXT
```

```
      ORG     100h            ; needed for .COM files.
MODE3 PROC
      cld                     ; Set the direction flag.
      mov     ax, cs          ; Ensure that DS and ES point to this sement.
      mov     ds, ax          ;
      mov     es, ax          ;
      cli                     ; Interupts off while changing the stack.
      mov     ss, ax          ;
      mov     sp, 0FFFFh      ;
      sti                     ; Stack change done - restore interupts.

      call    Step1           ; Enable SPC8104 chip.
      call    Step2           ; Unlock the Aux port.

      ; at this point, all the registers are accessible, we need to
      ; program Auxilliary registers according to panel size, memory,
      ; and some preferences...

      call    Step3           ; Setup for the panel and enviroment.

      ; at this point all registers are fully accessible, following
      ; is a standard VGA stuff

      call    Step4           ; Setup the standard VGA registers.
      call    Step5           ; Load the font.
      call    Step6           ; Blow the DAC with mode 3 colors.
      call    Step7           ; Setup the attribute registers.
      call    Step8           ; Clear VRAM

;-----------------------------------------------------------------------
;
;  You should now be running as MODE 3 without BIOS support.
;
;-----------------------------------------------------------------------

      mov     si,offset Message
      call    TextOut         ; show we are up and running

      mov     ax, 4C00h       ; DOS - Exit with Return Code.
      int     21h             ; This gives us a clean exit under
MODE3 ENDP                    ;  MS-DOS.


Message:
      DB 'MODE3.COM : Program to initialize SPC8104 registers for '
      DB '8x8 font text mode without BIOS.',0
```

```
;*********************************************************************
;
;  STEP1 - Before ANY ports can be accessed, the SPC8104 chip must be
;        enabled.
;
;*********************************************************************

Step1:
      mov     dx,3c3h
      mov     al,1
      out     dx,al
      ret

;*********************************************************************
;
;  STEP2 - Unlock Auxiliary Port.
;
;  Before programming the Auxiliary port for configuration,
;  it must be unlocked. Writing 1Ah to 3DEh index DEh and reading
;  it back will unlock the Auxiliary Port.
;
;*********************************************************************

Step2:
      mov     dx, 3DEh          ; DX = port 3DEh.
      mov     ax, 1ADEh         ; AX = index DEh, value 1Ah.
      out     dx, ax            ;
      inc     dx                ; Point to 3DFh.
      in      al, dx            ; Do the read.

      ret                       ; End of Step2


;*********************************************************************
;
;  STEP3 - This sets the chip up for the correct panel and for
;          a proper working environment.
;
;
;*********************************************************************

Step3:
      mov     si, OFFSET SetupChipTbl
      call    RunTable
      ret                       ; End of Step3.
```

```
SetupChipTbl:
      DW      PW+03C2h,67h     ; Default card to 3Dx, 28Mhz Clock
      DW      03DEh
      DB       00h,11000001b   ; 16 bit Mem & I/O, enable CRTC-B set
      DB      -1

      DW      03D4h
      DB       01h, 80         ; CRTC_B[1] = 80 (dec)
      DB       05h, 12         ; CRTC_B[5] = 12 (dec)
      DB       010h,02         ; CRTC_B[10] = 2 (2 vert. lines)
      DB       012h,480/4      ; CRTC_B[12h] = 8 msb of 480/2
      DB      -1

      DW      03DEh
      DB       00h,11000000b   ; IRQ disable,disable CRTC-B set
      DB       01H,00000011b   ; autocenter, slow cursor
      DB       02H,00000000b   ; 8 bit dual I/F,4ms refresh
      DB       03H,00000000b   ; Full power
      DB       04H,00000000b   ; regular clocks
      DB       05H,00000000b   ; normal display pages
      DB       06H,00100000b   ; WF count = 32
      DB       07H,10000000b   ; frame buffer start at 256k
      DB       09H,00100101b   ; fr.buffer end high byte
      DB       0AH,01111101b   ; fr.buffer end low byte
      DB       0DH,00000000b   ; clear scratch register
      DB       -1
      DW      -1               ; end of the table

;*************************************************************************
;
;  STEP4 - Use tables to program the standard VGA registers.
;
;*************************************************************************

Step4:
      mov     si, offset VGAregs
      call    RunTable
      ret                      ; End Step4.
```

```
VGAregs:
      DW       PW+3C2h,0E3h     ; Misc. Output.

      DW       3C4h             ; Sequencer.
      DB        00h, 00h
      DB        01h, 01h
      DB        02h, 03h
      DB        03h, 00h
      DB        04h, 02h
      DB        00h, 03h
      DB       -1

      DW       3D4H             ; CRTC
      DB        11H, 0CH        ; enable CRTC regs 0..7
      DB        00H, 05Fh
      DB        01H, 04Fh
      DB        02H, 050h
      DB        03H, 082h
      DB        04H, 055h
      DB        05H, 081h
      DB        06H, 0BFh
      DB        07H, 01Fh
      DB        08H, 000h
      DB        09H, 0C7h       ; font height = 8
      DB        0AH, 006h
      DB        0BH, 007h
      DB        0CH, 000h
      DB        0DH, 000h
      DB        0EH, 000h
      DB        0FH, 000h
      DB        10H, 09Ch
      DB        11H, 08Eh
      DB        12H, 08Fh
      DB        13H, 028h
      DB        14H, 01Fh
      DB        15H, 096h
      DB        16H, 0B9h
      DB        17H, 0A3h
      DB        18H, 0FFh
      DB        -1
```

```
      DW      3CEH              ; Graphics.
      DB       00H,000h
      DB       01H,000h
      DB       02H,000h
      DB       03H,000h
      DB       04H,000h
      DB       05H,010h
      DB       06H,00Eh
      DB       07H,000h
      DB       08H,0FFh
      DB       -1
      DW      -1                ; End of STEP4

;*********************************************************************
;
;    STEP5 - Load Font. Load font into plane 2.
;
;*********************************************************************

Step5:
      mov     si, offset OpenFontMapTbl
      call    RunTable          ; estabilish planar addressing
                                ; at A000h, enable plane 2 for write

      mov     si,offset CharSet8 ; 8x8 font offset
      mov     ax,0A000h         ; Set ES to Vram segment
      mov     es,ax
      xor     di,di
      mov     ax,256            ; Set character count
@@:
      mov     cx,4              ; Set point count (4 WORDS)
      rep     movsw             ; Transfer this character

      add     di,32-8           ; Next map entry
      dec     ax                ; Count character
      jnz     @B                ; Do all characters

      mov     si, offset CloseFontMapTbl
      call    RunTable          ; estabilish text addressing again
      ret                       ; End of Step5.
```

```
OpenFontMapTbl:
      DW      3C4h              ; Sequencer.
      DB       02h,04h
      DB       04h,07h
      DB       -1
      DW      3CEH              ; Graphics.
      DB       05H,00h
      DB       06H,04h
      DB       -1
      DW      -1                ; End of table

CloseFontMapTbl:
      DW      3C4h              ; Sequencer.
      DB       02h,03h
      DB       04h,02h
      DB       -1
      DW      3CEH              ; Graphics.
      DB       05H,10h
      DB       06H,0Eh
      DB       -1
      DW      -1                ; End of table


CharSet8:
      DB      000h,000h,000h,000h,000h,000h,000h,000h ; 0
      DB      07Eh,081h,0A5h,081h,0BDh,099h,081h,07Eh ; 1
      DB      07Eh,0FFh,0DBh,0FFh,0C3h,0E7h,0FFh,07Eh ; 2
      DB      06Ch,0FEh,0FEh,0FEh,07Ch,038h,010h,000h ; 3
      DB      010h,038h,07Ch,0FEh,07Ch,038h,010h,000h ; 4
      DB      038h,07Ch,038h,0FEh,0FEh,07Ch,038h,07Ch ; 5
      DB      010h,010h,038h,07Ch,0FEh,07Ch,038h,07Ch ; 6
      DB      000h,000h,018h,03Ch,03Ch,018h,000h,000h ; 7
      DB      0FFh,0FFh,0E7h,0C3h,0C3h,0E7h,0FFh,0FFh ; 8
      DB      000h,03Ch,066h,042h,042h,066h,03Ch,000h ; 9
      DB      0FFh,0C3h,099h,0BDh,0BDh,099h,0C3h,0FFh ; 10
      DB      00Fh,007h,00Fh,07Dh,0CCh,0CCh,0CCh,078h ; 11
      DB      03Ch,066h,066h,066h,03Ch,018h,07Eh,018h ; 12
      DB      03Fh,033h,03Fh,030h,030h,070h,0F0h,0E0h ; 13
      DB      07Fh,063h,07Fh,063h,063h,067h,0E6h,0C0h ; 14
      DB      099h,05Ah,03Ch,0E7h,0E7h,03Ch,05Ah,099h ; 15
      DB      080h,0E0h,0F8h,0FEh,0F8h,0E0h,080h,000h ; 16
      DB      002h,00Eh,03Eh,0FEh,03Eh,00Eh,002h,000h ; 17
      DB      018h,03Ch,07Eh,018h,018h,07Eh,03Ch,018h ; 18
      DB      066h,066h,066h,066h,066h,000h,066h,000h ; 19
      DB      07Fh,0DBh,0DBh,07Bh,01Bh,01Bh,01Bh,000h ; 20
      DB      03Eh,063h,038h,06Ch,06Ch,038h,0CCh,078h ; 21
      DB      000h,000h,000h,000h,07Eh,07Eh,07Eh,000h ; 22
      DB      018h,03Ch,07Eh,018h,07Eh,03Ch,018h,0FFh ; 23
      DB      018h,03Ch,07Eh,018h,018h,018h,018h,000h ; 24
      DB      018h,018h,018h,018h,07Eh,03Ch,018h,000h ; 25
```

```
DB      000h,018h,00Ch,0FEh,00Ch,018h,000h,000h ; 26
DB      000h,030h,060h,0FEh,060h,030h,000h,000h ; 27
DB      000h,000h,0C0h,0C0h,0C0h,0FEh,000h,000h ; 28
DB      000h,024h,066h,0FFh,066h,024h,000h,000h ; 29
DB      000h,018h,03Ch,07Eh,0FFh,0FFh,000h,000h ; 30
DB      000h,0FFh,0FFh,07Eh,03Ch,018h,000h,000h ; 31
DB      000h,000h,000h,000h,000h,000h,000h,000h ;
DB      030h,078h,078h,030h,030h,000h,030h,000h ; !
DB      06Ch,06Ch,06Ch,000h,000h,000h,000h,000h ; "
DB      06Ch,06Ch,0FEh,06Ch,0FEh,06Ch,06Ch,000h ; #
DB      030h,07Ch,0C0h,078h,00Ch,0F8h,030h,000h ; $
DB      000h,0C6h,0CCh,018h,030h,066h,0C6h,000h ; %
DB      038h,06Ch,038h,076h,0DCh,0CCh,076h,000h ; &
DB      060h,060h,0C0h,000h,000h,000h,000h,000h ; '
DB      018h,030h,060h,060h,060h,030h,018h,000h ; (
DB      060h,030h,018h,018h,018h,030h,060h,000h ; )
DB      000h,066h,03Ch,0FFh,03Ch,066h,000h,000h ; *
DB      000h,030h,030h,0FCh,030h,030h,000h,000h ; +
DB      000h,000h,000h,000h,000h,030h,030h,060h ; ,
DB      000h,000h,000h,0FCh,000h,000h,000h,000h ; -
DB      000h,000h,000h,000h,000h,030h,030h,000h ; .
DB      006h,00Ch,018h,030h,060h,0C0h,080h,000h ; /
DB      07Ch,0C6h,0CEh,0DEh,0F6h,0E6h,07Ch,000h ; 0
DB      030h,070h,030h,030h,030h,030h,0FCh,000h ; 1
DB      078h,0CCh,00Ch,038h,060h,0CCh,0FCh,000h ; 2
DB      078h,0CCh,00Ch,038h,00Ch,0CCh,078h,000h ; 3
DB      01Ch,03Ch,06Ch,0CCh,0FEh,00Ch,01Eh,000h ; 4
DB      0FCh,0C0h,0F8h,00Ch,00Ch,0CCh,078h,000h ; 5
DB      038h,060h,0C0h,0F8h,0CCh,0CCh,078h,000h ; 6
DB      0FCh,0CCh,00Ch,018h,030h,030h,030h,000h ; 7
DB      078h,0CCh,0CCh,078h,0CCh,0CCh,078h,000h ; 8
DB      078h,0CCh,0CCh,07Ch,00Ch,018h,070h,000h ; 9
DB      000h,030h,030h,000h,000h,030h,030h,000h ; :
DB      000h,030h,030h,000h,000h,030h,030h,060h ; ;
DB      018h,030h,060h,0C0h,060h,030h,018h,000h ; <
DB      000h,000h,0FCh,000h,000h,0FCh,000h,000h ; =
DB      060h,030h,018h,00Ch,018h,030h,060h,000h ; >
DB      078h,0CCh,00Ch,018h,030h,000h,030h,000h ; ?
DB      07Ch,0C6h,0DEh,0DEh,0DEh,0C0h,078h,000h ; @
DB      030h,078h,0CCh,0CCh,0FCh,0CCh,0CCh,000h ; A
DB      0FCh,066h,066h,07Ch,066h,066h,0FCh,000h ; B
DB      03Ch,066h,0C0h,0C0h,0C0h,066h,03Ch,000h ; C
DB      0F8h,06Ch,066h,066h,066h,06Ch,0F8h,000h ; D
DB      0FEh,062h,068h,078h,068h,062h,0FEh,000h ; E
DB      0FEh,062h,068h,078h,068h,060h,0F0h,000h ; F
DB      03Ch,066h,0C0h,0C0h,0CEh,066h,03Eh,000h ; G
DB      0CCh,0CCh,0CCh,0FCh,0CCh,0CCh,0CCh,000h ; H
DB      078h,030h,030h,030h,030h,030h,078h,000h ; I
DB      01Eh,00Ch,00Ch,00Ch,0CCh,0CCh,078h,000h ; J
DB      0E6h,066h,06Ch,078h,06Ch,066h,0E6h,000h ; K
```

```
DB      0F0h,060h,060h,060h,062h,066h,0FEh,000h ; L
DB      0C6h,0EEh,0FEh,0FEh,0D6h,0C6h,0C6h,000h ; M
DB      0C6h,0E6h,0F6h,0DEh,0CEh,0C6h,0C6h,000h ; N
DB      038h,06Ch,0C6h,0C6h,0C6h,06Ch,038h,000h ; O
DB      0FCh,066h,066h,07Ch,060h,060h,0F0h,000h ; P
DB      078h,0CCh,0CCh,0CCh,0DCh,078h,01Ch,000h ; Q
DB      0FCh,066h,066h,07Ch,06Ch,066h,0E6h,000h ; R
DB      078h,0CCh,0E0h,070h,01Ch,0CCh,078h,000h ; S
DB      0FCh,0B4h,030h,030h,030h,030h,078h,000h ; T
DB      0CCh,0CCh,0CCh,0CCh,0CCh,0CCh,0FCh,000h ; U
DB      0CCh,0CCh,0CCh,0CCh,0CCh,078h,030h,000h ; V
DB      0C6h,0C6h,0C6h,0D6h,0FEh,0EEh,0C6h,000h ; W
DB      0C6h,0C6h,06Ch,038h,038h,06Ch,0C6h,000h ; X
DB      0CCh,0CCh,0CCh,078h,030h,030h,078h,000h ; Y
DB      0FEh,0C6h,08Ch,018h,032h,066h,0FEh,000h ; Z
DB      078h,060h,060h,060h,060h,060h,078h,000h ; [
DB      0C0h,060h,030h,018h,00Ch,006h,002h,000h ; \
DB      078h,018h,018h,018h,018h,018h,078h,000h ; ]
DB      010h,038h,06Ch,0C6h,000h,000h,000h,000h ; ^
DB      000h,000h,000h,000h,000h,000h,000h,0FFh ; _
DB      030h,030h,018h,000h,000h,000h,000h,000h ; ‘
DB      000h,000h,078h,00Ch,07Ch,0CCh,076h,000h ; a
DB      0E0h,060h,060h,07Ch,066h,066h,0DCh,000h ; b
DB      000h,000h,078h,0CCh,0C0h,0CCh,078h,000h ; c
DB      01Ch,00Ch,00Ch,07Ch,0CCh,0CCh,076h,000h ; d
DB      000h,000h,078h,0CCh,0FCh,0C0h,078h,000h ; e
DB      038h,06Ch,060h,0F0h,060h,060h,0F0h,000h ; f
DB      000h,000h,076h,0CCh,0CCh,07Ch,00Ch,0F8h ; g
DB      0E0h,060h,06Ch,076h,066h,066h,0E6h,000h ; h
DB      030h,000h,070h,030h,030h,030h,078h,000h ; i
DB      00Ch,000h,00Ch,00Ch,00Ch,0CCh,0CCh,078h ; j
DB      0E0h,060h,066h,06Ch,078h,06Ch,0E6h,000h ; k
DB      070h,030h,030h,030h,030h,030h,078h,000h ; l
DB      000h,000h,0CCh,0FEh,0FEh,0D6h,0C6h,000h ; m
DB      000h,000h,0F8h,0CCh,0CCh,0CCh,0CCh,000h ; n
DB      000h,000h,078h,0CCh,0CCh,0CCh,078h,000h ; o
DB      000h,000h,0DCh,066h,066h,07Ch,060h,0F0h ; p
DB      000h,000h,076h,0CCh,0CCh,07Ch,00Ch,01Eh ; q
DB      000h,000h,0DCh,076h,066h,060h,0F0h,000h ; r
DB      000h,000h,07Ch,0C0h,078h,00Ch,0F8h,000h ; s
DB      010h,030h,07Ch,030h,030h,034h,018h,000h ; t
DB      000h,000h,0CCh,0CCh,0CCh,0CCh,076h,000h ; u
DB      000h,000h,0CCh,0CCh,0CCh,078h,030h,000h ; v
DB      000h,000h,0C6h,0D6h,0FEh,0FEh,06Ch,000h ; w
DB      000h,000h,0C6h,06Ch,038h,06Ch,0C6h,000h ; x
DB      000h,000h,0CCh,0CCh,0CCh,07Ch,00Ch,0F8h ; y
DB      000h,000h,0FCh,098h,030h,064h,0FCh,000h ; z
DB      01Ch,030h,030h,0E0h,030h,030h,01Ch,000h ; {
DB      018h,018h,018h,000h,018h,018h,018h,000h ; |
DB      0E0h,030h,030h,01Ch,030h,030h,0E0h,000h ; }
```

```
DB      076h,0DCh,000h,000h,000h,000h,000h,000h ; ~
DB      000h,010h,038h,06Ch,0C6h,0C6h,0FEh,000h ; 127

; EXTENDED character set

DB      078h,0CCh,0C0h,0CCh,078h,018h,00Ch,078h ; 128
DB      000h,0CCh,000h,0CCh,0CCh,0CCh,07Eh,000h ; 129
DB      01Ch,000h,078h,0CCh,0FCh,0C0h,078h,000h ; 130
DB      07Eh,0C3h,03Ch,006h,03Eh,066h,03Fh,000h ; 131
DB      0CCh,000h,078h,00Ch,07Ch,0CCh,07Eh,000h ; 132
DB      0E0h,000h,078h,00Ch,07Ch,0CCh,07Eh,000h ; 133
DB      030h,030h,078h,00Ch,07Ch,0CCh,07Eh,000h ; 134
DB      000h,000h,078h,0C0h,0C0h,078h,00Ch,038h ; 135
DB      07Eh,0C3h,03Ch,066h,07Eh,060h,03Ch,000h ; 136
DB      0CCh,000h,078h,0CCh,0FCh,0C0h,078h,000h ; 137
DB      0E0h,000h,078h,0CCh,0FCh,0C0h,078h,000h ; 138
DB      0CCh,000h,070h,030h,030h,030h,078h,000h ; 139
DB      07Ch,0C6h,038h,018h,018h,018h,03Ch,000h ; 140
DB      0E0h,000h,070h,030h,030h,030h,078h,000h ; 141
DB      0C6h,038h,06Ch,0C6h,0FEh,0C6h,0C6h,000h ; 142
DB      030h,030h,000h,078h,0CCh,0FCh,0CCh,000h ; 143
DB      01Ch,000h,0FCh,060h,078h,060h,0FCh,000h ; 144
DB      000h,000h,07Fh,00Ch,07Fh,0CCh,07Fh,000h ; 145
DB      03Eh,06Ch,0CCh,0FEh,0CCh,0CCh,0CEh,000h ; 146
DB      078h,0CCh,000h,078h,0CCh,0CCh,078h,000h ; 147
DB      000h,0CCh,000h,078h,0CCh,0CCh,078h,000h ; 148
DB      000h,0E0h,000h,078h,0CCh,0CCh,078h,000h ; 149
DB      078h,0CCh,000h,0CCh,0CCh,0CCh,07Eh,000h ; 150
DB      000h,0E0h,000h,0CCh,0CCh,0CCh,07Eh,000h ; 151
DB      000h,0CCh,000h,0CCh,0CCh,07Ch,00Ch,0F8h ; 152
DB      0C3h,018h,03Ch,066h,066h,03Ch,018h,000h ; 153
DB      0CCh,000h,0CCh,0CCh,0CCh,0CCh,078h,000h ; 154
DB      018h,018h,07Eh,0C0h,0C0h,07Eh,018h,018h ; 155
DB      038h,06Ch,064h,0F0h,060h,0E6h,0FCh,000h ; 156
DB      0CCh,0CCh,078h,0FCh,030h,0FCh,030h,030h ; 157
DB      0F8h,0CCh,0CCh,0FAh,0C6h,0CFh,0C6h,0C7h ; 158
DB      00Eh,01Bh,018h,03Ch,018h,018h,0D8h,070h ; 159
DB      01Ch,000h,078h,00Ch,07Ch,0CCh,07Eh,000h ; 160
DB      038h,000h,070h,030h,030h,030h,078h,000h ; 161
DB      000h,01Ch,000h,078h,0CCh,0CCh,078h,000h ; 162
DB      000h,01Ch,000h,0CCh,0CCh,0CCh,07Eh,000h ; 163
DB      000h,0F8h,000h,0F8h,0CCh,0CCh,0CCh,000h ; 164
DB      0FCh,000h,0CCh,0ECh,0FCh,0DCh,0CCh,000h ; 165
DB      03Ch,06Ch,06Ch,03Eh,000h,07Eh,000h,000h ; 166
DB      038h,06Ch,06Ch,038h,000h,07Ch,000h,000h ; 167
DB      030h,000h,030h,060h,0C0h,0CCh,078h,000h ; 168
DB      000h,000h,000h,0FCh,0C0h,0C0h,000h,000h ; 169
DB      000h,000h,000h,0FCh,00Ch,00Ch,000h,000h ; 170
DB      0C3h,0C6h,0CCh,0DEh,033h,066h,0CCh,00Fh ; 171
DB      0C3h,0C6h,0CCh,0DBh,037h,06Fh,0CFh,003h ; 172
```

```
DB      018h,018h,000h,018h,018h,018h,018h,000h ; 173
DB      000h,033h,066h,0CCh,066h,033h,000h,000h ; 174
DB      000h,0CCh,066h,033h,066h,0CCh,000h,000h ; 175
DB      022h,088h,022h,088h,022h,088h,022h,088h ; 176
DB      055h,0AAh,055h,0AAh,055h,0AAh,055h,0AAh ; 177
DB      0DBh,077h,0DBh,0EEh,0DBh,077h,0DBh,0EEh ; 178
DB      018h,018h,018h,018h,018h,018h,018h,018h ; 179
DB      018h,018h,018h,018h,0F8h,018h,018h,018h ; 180
DB      018h,018h,0F8h,018h,0F8h,018h,018h,018h ; 181
DB      036h,036h,036h,036h,0F6h,036h,036h,036h ; 182
DB      000h,000h,000h,000h,0FEh,036h,036h,036h ; 183
DB      000h,000h,0F8h,018h,0F8h,018h,018h,018h ; 184
DB      036h,036h,0F6h,006h,0F6h,036h,036h,036h ; 185
DB      036h,036h,036h,036h,036h,036h,036h,036h ; 186
DB      000h,000h,0FEh,006h,0F6h,036h,036h,036h ; 187
DB      036h,036h,0F6h,006h,0FEh,000h,000h,000h ; 188
DB      036h,036h,036h,036h,0FEh,000h,000h,000h ; 189
DB      018h,018h,0F8h,018h,0F8h,000h,000h,000h ; 190
DB      000h,000h,000h,000h,0F8h,018h,018h,018h ; 191
DB      018h,018h,018h,018h,01Fh,000h,000h,000h ; 192
DB      018h,018h,018h,018h,0FFh,000h,000h,000h ; 193
DB      000h,000h,000h,000h,0FFh,018h,018h,018h ; 194
DB      018h,018h,018h,018h,01Fh,018h,018h,018h ; 195
DB      000h,000h,000h,000h,0FFh,000h,000h,000h ; 196
DB      018h,018h,018h,018h,0FFh,018h,018h,018h ; 197
DB      018h,018h,01Fh,018h,01Fh,018h,018h,018h ; 198
DB      036h,036h,036h,036h,037h,036h,036h,036h ; 199
DB      036h,036h,037h,030h,03Fh,000h,000h,000h ; 200
DB      000h,000h,03Fh,030h,037h,036h,036h,036h ; 201
DB      036h,036h,0F7h,000h,0FFh,000h,000h,000h ; 202
DB      000h,000h,0FFh,000h,0F7h,036h,036h,036h ; 203
DB      036h,036h,037h,030h,037h,036h,036h,036h ; 204
DB      000h,000h,0FFh,000h,0FFh,000h,000h,000h ; 205
DB      036h,036h,0F7h,000h,0F7h,036h,036h,036h ; 206
DB      018h,018h,0FFh,000h,0FFh,000h,000h,000h ; 207
DB      036h,036h,036h,036h,0FFh,000h,000h,000h ; 208
DB      000h,000h,0FFh,000h,0FFh,018h,018h,018h ; 209
DB      000h,000h,000h,000h,0FFh,036h,036h,036h ; 210
DB      036h,036h,036h,036h,03Fh,000h,000h,000h ; 211
DB      018h,018h,01Fh,018h,01Fh,000h,000h,000h ; 212
DB      000h,000h,01Fh,018h,01Fh,018h,018h,018h ; 213
DB      000h,000h,000h,000h,03Fh,036h,036h,036h ; 214
DB      036h,036h,036h,036h,0FFh,036h,036h,036h ; 215
DB      018h,018h,0FFh,018h,0FFh,018h,018h,018h ; 216
DB      018h,018h,018h,018h,0F8h,000h,000h,000h ; 217
DB      000h,000h,000h,000h,01Fh,018h,018h,018h ; 218
DB      0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh ; 219
DB      000h,000h,000h,000h,0FFh,0FFh,0FFh,0FFh ; 220
DB      0F0h,0F0h,0F0h,0F0h,0F0h,0F0h,0F0h,0F0h ; 221
DB      00Fh,00Fh,00Fh,00Fh,00Fh,00Fh,00Fh,00Fh ; 222
```

```
DB      0FFh,0FFh,0FFh,0FFh,000h,000h,000h,000h ; 223
DB      000h,000h,076h,0DCh,0C8h,0DCh,076h,000h ; 224
DB      000h,078h,0CCh,0F8h,0CCh,0F8h,0C0h,0C0h ; 225
DB      000h,0FCh,0CCh,0C0h,0C0h,0C0h,0C0h,000h ; 226
DB      000h,0FEh,06Ch,06Ch,06Ch,06Ch,06Ch,000h ; 227
DB      0FCh,0CCh,060h,030h,060h,0CCh,0FCh,000h ; 228
DB      000h,000h,07Eh,0D8h,0D8h,0D8h,070h,000h ; 229
DB      000h,066h,066h,066h,066h,07Ch,060h,0C0h ; 230
DB      000h,076h,0DCh,018h,018h,018h,018h,000h ; 231
DB      0FCh,030h,078h,0CCh,0CCh,078h,030h,0FCh ; 232
DB      038h,06Ch,0C6h,0FEh,0C6h,06Ch,038h,000h ; 233
DB      038h,06Ch,0C6h,0C6h,06Ch,06Ch,0EEh,000h ; 234
DB      01Ch,030h,018h,07Ch,0CCh,0CCh,078h,000h ; 235
DB      000h,000h,07Eh,0DBh,0DBh,07Eh,000h,000h ; 236
DB      006h,00Ch,07Eh,0DBh,0DBh,07Eh,060h,0C0h ; 237
DB      038h,060h,0C0h,0F8h,0C0h,060h,038h,000h ; 238
DB      078h,0CCh,0CCh,0CCh,0CCh,0CCh,0CCh,000h ; 239
DB      000h,0FCh,000h,0FCh,000h,0FCh,000h,000h ; 240
DB      030h,030h,0FCh,030h,030h,000h,0FCh,000h ; 241
DB      060h,030h,018h,030h,060h,000h,0FCh,000h ; 242
DB      018h,030h,060h,030h,018h,000h,0FCh,000h ; 243
DB      00Eh,01Bh,01Bh,018h,018h,018h,018h,018h ; 244
DB      018h,018h,018h,018h,018h,0D8h,0D8h,070h ; 245
DB      030h,030h,000h,0FCh,000h,030h,030h,000h ; 246
DB      000h,076h,0DCh,000h,076h,0DCh,000h,000h ; 247
DB      038h,06Ch,06Ch,038h,000h,000h,000h,000h ; 248
DB      000h,000h,000h,018h,018h,000h,000h,000h ; 249
DB      000h,000h,000h,000h,018h,000h,000h,000h ; 250
DB      00Fh,00Ch,00Ch,00Ch,0ECh,06Ch,03Ch,01Ch ; 251
DB      078h,06Ch,06Ch,06Ch,06Ch,000h,000h,000h ; 252
DB      070h,018h,030h,060h,078h,000h,000h,000h ; 253
DB      000h,000h,03Ch,03Ch,03Ch,03Ch,000h,000h ; 254
DB      000h,000h,000h,000h,000h,000h,000h,000h ; 255
```

```
;**********************************************************************
;
;    STEP6 - program the DAC.
;
;**********************************************************************

Step6:
      mov      al, 0FFh          ; Turn on PEL mask.
      mov      dx, 03c6h         ;
      out      dx, al            ;

      mov      al,0              ; Set data index to zero.
      mov      dx,3c8h           ;
      out      dx,al             ;

      mov      dx,3C9h           ; Set DAC write index
      mov      cx,3 * 256        ; CX = # bytes (R+G+B * 256).
      mov      si,offset NewDAC ; SI -> new DAC data.
DACloop:
      lodsb                      ; Get the byte.
      out      dx, al            ; Set the PEL data.
      loop     DACloop           ; Continue till CX == 0.

      ret                        ; End Step6

NewDACDB       00h, 00h, 00h , 00h, 00h, 2ah , 00h, 2ah, 00h , 00h, 2ah, 2ah
      DB       2ah, 00h, 00h , 2ah, 00h, 2ah , 2ah, 2ah, 00h , 2ah, 2ah, 2ah
      DB       00h, 00h, 15h , 00h, 00h, 3fh , 00h, 2ah, 15h , 00h, 2ah, 3fh
      DB       2ah, 00h, 15h , 2ah, 00h, 3fh , 2ah, 2ah, 15h , 2ah, 2ah, 3fh
      DB       00h, 15h, 00h , 00h, 15h, 2ah , 00h, 3fh, 00h , 00h, 3fh, 2ah
      DB       2ah, 15h, 00h , 2ah, 15h, 2ah , 2ah, 3fh, 00h , 2ah, 3fh, 2ah
      DB       00h, 15h, 15h , 00h, 15h, 3fh , 00h, 3fh, 15h , 00h, 3fh, 3fh
      DB       2ah, 15h, 15h , 2ah, 15h, 3fh , 2ah, 3fh, 15h , 2ah, 3fh, 3fh
      DB       15h, 00h, 00h , 15h, 00h, 2ah , 15h, 2ah, 00h , 15h, 2ah, 2ah
      DB       3fh, 00h, 00h , 3fh, 00h, 2ah , 3fh, 2ah, 00h , 3fh, 2ah, 2ah
      DB       15h, 00h, 15h , 15h, 00h, 3fh , 15h, 2ah, 15h , 15h, 2ah, 3fh
      DB       3fh, 00h, 15h , 3fh, 00h, 3fh , 3fh, 2ah, 15h , 3fh, 2ah, 3fh
      DB       15h, 15h, 00h , 15h, 15h, 2ah , 15h, 3fh, 00h , 15h, 3fh, 2ah
      DB       3fh, 15h, 00h , 3fh, 15h, 2ah , 3fh, 3fh, 00h , 3fh, 3fh, 2ah
      DB       15h, 15h, 15h , 15h, 15h, 3fh , 15h, 3fh, 15h , 15h, 3fh, 3fh
      DB       3fh, 15h, 15h , 3fh, 15h, 3fh , 3fh, 3fh, 15h , 3fh, 3fh, 3fh
      DB       00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
      DB       00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
      DB       00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
      DB       00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
      DB       00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
      DB       00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
      DB       00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
      DB       00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
      DB       00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
```

```
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
DB      00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h , 00h, 00h, 00h
```

```
;*********************************************************************
;
;  STEP7 - Program Attribute controller.
;
;  Reset Index/Data Flip Flop of Attribute controller.
;  Write attributes information.
;
;*********************************************************************

Step7:
      mov     dx, 3DAh; Set attributes register to index.
      in      al, dx          ;

      mov     si, offset AttribTable
      mov     dx, 3C0h        ;
      mov     cx, 15h         ; Write to 15h registers.
      mov     bx, 0h          ; Use BX as the index counter.
Step7_Lp:
      mov     ax, bx          ; Set the next index.
      out     dx, al          ;
      inc     bx              ;
      lodsb                   ; Read the next byte.
      out     dx, al          ; Write the next data.
      loop    Step7_Lp        ;

      mov     dx, 3DAh; Set attributes register to index.
      in      al, dx          ;

      mov     al, 20h
      mov     dx, 3C0h
      out     dx, al
      ret                     ; End of Step7


AttribTable:
      db      00h, 01h, 02h, 03h, 04h, 05h, 14h, 07h
      db      38h, 39h, 3Ah, 3Bh, 3Ch, 3Dh, 3Eh, 3Fh
      db      0ch, 00h, 0Fh, 00h, 00h
```

```
;*********************************************************************
;
;    STEP8 - Clear Vram. Initialize video memory with "normal" attributes
;            and ASCII spaces (blanks).
;
;*********************************************************************

Step8:
      mov     ax, 0B800h;
      mov     es, ax          ;
      mov     cx, 4000h       ; 16k Words = 32k Bytes
      xor     di, di          ;
      mov     ax, 720h        ; normal attribute, ASCII space
      cld
      rep     stosw           ;
      ret                     ; End of Step8.



; ADDITIONAL SUPPORT ROUTINES
;----------------------------

; RunTable
; This routine interprets the configuration tables that we have
; presented above.  PW is a mask bit that causes a branch from
; index/data to a simple port write of data (ie. if High bit on
; Port value is set, then branch to simple port write BR3)

;
;     Entry:  CS:SI - Pointer to big register table
;     Exit:   n/a
;     Uses:   AX,DX,SI

;-----------------------------------------------------------------------------
;     RunTable - Initialize registers from table
;
;     Entry:  DS:SI - Pointer to register table
;
;     Exit:   n/a
;     Uses:   AX,DX,SI
;-----------------------------------------------------------------------------

br0:
      dec     si              ; Undo last byte overrun
```

```
RunTable:
      lodsw                   ; Read next control word
      cmp     ax,-1           ; End of table?
      je      brx             ; Exit if so

      mov     dx,ax           ; Get port in DX
      and     dh,7fh          ; Mask off port write bit
      test    ah,80h          ; Test if port write?
      jnz     br3             ; Jump if so
br2:
      lodsw                   ; Read index,data
      cmp     al,-1           ; End of run
      je      br0             ; Jump back if so
      out     dx,ax           ; Write the data
      jmp     br2             ; Back for more
br3:
      lodsw                   ; Read data
      out     dx,al           ; Write the data
      jmp     RunTable        ; Back for more
brx:
      ret


TextOut:
      xor     di,di
      mov     ax,0b800h
      mov     es,ax           ; es:di points to video buffer
@@:
      lodsb                   ; get a character
      test    al,al           ; end of string?
      jz      TextOutX        ; ..yes, quit
      stosb                   ; ..no, copy character into video buffer
      inc     di              ; skip attribute
      jmp     @B              ; do them all
TextOutX:
      ret

TEXT  ENDS
      END     MODE3
```

2.0 INITIALIZING THE SPC8104 IN MODE 12H WITHOUT A BIOS

The purpose of this document is to demonstrate how to initialize the SPC8104F0A into a mode similar to IBM's Standard VGA mode 12h by supplying sample code. The steps involved in this are:

- Enabling the SPC8104
- Unlocking the Auxiliary Port
- Setting up the SPC8104 panel specific registers
  In this case for a Single STN, Color 640x480 LCD panel
- Setting up the Standard VGA registers for Mode 12h
- Setting up the attribute registers
- Initializing the RAMDAC values
- Clearing Video Memory

The resulting mode is not BIOS supported, so no DOS text will work, however, routines developed to write pixels without the BIOS into mode 12h video memory will work correctly.

```
; MODE12.ASM
;
; Programs the SPC8104 for 640x480 Single LCD panel.
; Sets up registers for mode 12h without the BIOS.
;
; This program assumes:
;       - memory refresh 4ms

;*****************************************************************
;
; Instructions:
;
;  - Make your changes
;  - Assemble       --  masm mode12.asm;
;  - Link       --  link mode12;
;  - Make a .COM file  --  exe2bin mode12.exe mode12.com
;
;
;*****************************************************************

TEXT SEGMENT WORD PUBLIC 'TEXT'
ASSUME CS:TEXT, DS:TEXT, SS:TEXT

      ORG     100h
Mode12 PROC
      cld                     ; Set the direction flag.
      mov     ax, cs          ; Ensure that DS and ES point to this sement.
      mov     ds, ax          ;
      mov     es, ax          ;
      cli                     ; Interupts off while changing the stack.
```

```
      mov     ss, ax          ;
      mov     sp, 0FFFFh      ;
      sti                     ; Stack change done - restore interupts.

      call    Step1           ; Enable SPC8104 chip.
      call    Step2           ; Unlock the Aux port.

      ; at this point, all the registers are accessible, we need to
      ; program Auxilliary registers according to panel size, memory,
      ; and some preferences...

      call    Step3           ; Setup for the panel and enviroment.

      ; at this point all registers are fully accessible

      call    Step4           ; Setup the standard VGA registers.
      call    Step5           ; Setup the attribute registers.
      call    Step6           ; Blow the DAC with mode 12h colors.
      call    Step7           ; Clear VRAM - just to be nice.

;-----------------------------------------------------------------------
;
;  You should now be running as MODE 12h without BIOS support.
;
;-----------------------------------------------------------------------
      mov     ax, 4C00h       ; DOS - Exit with Return Code.
      int     21h             ; This gives us a clean exit under
Mode12ENDP                    ;  MS-DOS.




;***********************************************************************
;
;  STEP1 - Before ANY ports can be accessed, the SPC8104 chip must be
;        enabled.
;
;***********************************************************************

Step1:
      mov     dx,3c3h
      mov     al,1
      out     dx,al
      ret
```

```
;*********************************************************************
;
;  STEP2 - Unlock Auxiliary Port.
;
;  Before programming the Auxiliary port for configuration,
;  it must be unlocked. Writing 1Ah to 3DEh index DEh and reading
;  it back will unlock the Auxiliary Port.
;
;*********************************************************************

Step2:
      mov     dx, 3DEh         ; DX = port 3DEh.
      mov     ax, 1ADEh        ; AX = index DEh, value 1Ah.
      out     dx, ax           ;
      inc     dx               ; Point to 3DFh.
      in      al, dx           ; Do the read.
      ret


;*********************************************************************
;
;  STEP3 - This sets the chip up for the correct panel and for
;          a proper working environment.
;
;
;*********************************************************************

Step3:
      mov     si, OFFSET SetupChipTbl
      call    RunTable
      ret


PW    EQU     8000h              ; Port write (RunTable)

SetupChipTbl:
      DW      PW+03C2h,67h     ; Default card to 3Dx, 28Mhz Clock
      DW      03DEh
      DB       00h,11000001b   ; 16 bit Mem & I/O, enable CRTC-B set
      DB      -1

      DW      03D4h
      DB       01h, 80         ; CRTC_B[1] = 80 (dec)
      DB       05h, 12         ; CRTC_B[5] = 12 (dec)
      DB       010h,02         ; CRTC_B[10] = 2 (2 vert. lines)
      DB       012h,480/2      ; CRTC_B[12h] = 8 msb of 480
      DB      -1
```

```
      DW      03DEh
      DB       00h,11000000b   ; IRQ disable,disable CRTC-B set
      DB       01H,00000011b   ; autocenter, slow cursor
      DB       02H,00100000b   ; 8 bit single I/F,4ms refresh
      DB       03H,00000000b   ; Full power
      DB       04H,00000000b   ; regular clocks
      DB       05H,00000000b   ; normal display pages
      DB       06H,00100000b   ; WF count = 32
      DB       07H,00000000b  ; N/A (frame buffer start)
      DB       09H,00000000b  ; N/A (fr.buffer end high byte)
      DB       0AH,00000000b  ; N/A (fr.buffer end low byte)
      DB       0DH,00000000b  ; clear scratch register
      DB       -1
      DW      -1               ; end of the table


;*************************************************************************
;
;  STEP4 - Use tables to program the standard VGA registers.
;
;*************************************************************************

Step4:
      mov     si, offset VGAregs
      call    RunTable
      ret                      ; End Step5.

VGAregs:
      DW      PW+3C2h,0E3h     ; Misc. Output.

      DW      3C4h             ; Sequencer.
      DB       00h,00h
      DB       01h,01h
      DB       02h,0fh
      DB       03h,00h
      DB       04h,06h
      DB       00h,03h
      DB      -1

      DW      3D4h             ; CRTC
      DB       11h,0Ch
      DB       00h,05Fh
      DB       01h,04Fh
      DB       02h,050h
      DB       03h,082h
      DB       04h,054h
      DB       05h,080h
      DB       06h,00Bh
      DB       07h,03Eh
      DB       08h,000h
```

```
DB        09h,040h
DB        0Ah,000h
DB        0Bh,000h
DB        0Ch,000h
DB        0Dh,000h
DB        0Eh,000h
DB        0Fh,000h
DB        10h,0EAh
DB        11h,08Ch
DB        12h,0DFh
DB        13h,028h
DB        14h,000h
DB        15h,0E7h
DB        16h,004h
DB        17h,0E3h
DB        18h,0FFh

  ; The following are sprite specific registers.
  ; Note the low byte values are written before high byte values.
  ; ( writing the high byte latches both low+high byte )

  DB        38h,00h
  DB        37h,00h
  DB        36h,00h
  DB        35h,00h
  DB        34h,00h
  DB        33h,00h
  DB        32h,00h
  DB        31h,00h
  DB        30h,00h
DB         -1

DW       3CEh             ; Graphics.
DB        00h,000h
DB        01h,000h
DB        02h,000h
DB        03h,000h
DB        04h,000h
DB        05h,000h
DB        06h,005h
DB        07h,00Fh
DB        08h,0FFh
DB        -1
DW       -1
```

```
;*********************************************************************
;
;  STEP5 - Program Attribute controller.
;
;  Reset Index/Data Flip Flop of Attribute controller.
;  Write attributes information.
;
;*********************************************************************

Step5:
      mov     dx, 3DAh         ; Set attributes register to index.
      in      al, dx           ;

      mov     si, offset AttribTable
      mov     dx, 3C0h         ;
      mov     cx, 14h          ; Write to 14h registers.
      mov     bx, 0h           ; Use BX as the index counter.
@@:
      mov     ax, bx           ; Set the next index.
      out     dx, al           ;
      inc     bx               ;
      lodsb                    ; Read the next byte.
      out     dx, al           ; Write the next data.
      loop    @B

      mov     al, 20h         ; al = enable palette bit
      mov     dx, 3C0h
      out     dx, al          ; enable palette
      ret


AttribTable:
      db      00h, 01h, 02h, 03h, 04h, 05h, 14h, 07h
      db      38h, 39h, 3Ah, 3Bh, 3Ch, 3Dh, 3Eh, 3Fh
      db      01h, 00h, 0Fh, 00h, 00h
```

```
;**********************************************************************
;
;    STEP6 - program the DAC.
;            Although the palette is only 64 entries, we program the
;            full 256 VGA palette entries. All entries above 64 will
;            be ignored. This is done to demonstrate proper behaviour
;            of third party software that expects 256 palette entries
;**********************************************************************

Step6:
      mov     al, 0FFh         ; Turn on PEL mask.
      mov     dx, 03c6h        ;
      out     dx, al           ;

      xor     al,al            ; Set data index to zero.
      mov     dx,3c8h          ;
      out     dx,al            ;

      mov     dx,3C9h          ; Set DAC write index
      mov     cx,3 * 256       ; CX = # bytes (RGB * 256).
      mov     si,offset NewDAC ; SI -> new DAC data.
DACloop:
      lodsb                    ; Get the byte.
      out     dx, al           ; Set the PEL data.
      loop    DACloop          ; Continue till CX == 0.

      ret                      ; End Step4

NewDAC DB     00h, 00h, 00h, 00h, 00h, 2ah, 00h, 2ah, 00h, 00h, 2ah, 2ah
      DB      2ah, 00h, 00h, 2ah, 00h, 2ah, 2ah, 2ah, 00h, 2ah, 2ah, 2ah
      DB      00h, 00h, 15h, 00h, 00h, 3fh, 00h, 2ah, 15h, 00h, 2ah, 3fh
      DB      2ah, 00h, 15h, 2ah, 00h, 3fh, 2ah, 2ah, 15h, 2ah, 2ah, 3fh
      DB      00h, 15h, 00h, 00h, 15h, 2ah, 00h, 3fh, 00h, 00h, 3fh, 2ah
      DB      2ah, 15h, 00h, 2ah, 15h, 2ah, 2ah, 3fh, 00h, 2ah, 3fh, 2ah
      DB      00h, 15h, 15h, 00h, 15h, 3fh, 00h, 3fh, 15h, 00h, 3fh, 3fh
      DB      2ah, 15h, 15h, 2ah, 15h, 3fh, 2ah, 3fh, 15h, 2ah, 3fh, 3fh
      DB      15h, 00h, 00h, 15h, 00h, 2ah, 15h, 2ah, 00h, 15h, 2ah, 2ah
      DB      3fh, 00h, 00h, 3fh, 00h, 2ah, 3fh, 2ah, 00h, 3fh, 2ah, 2ah
      DB      15h, 00h, 15h, 15h, 00h, 3fh, 15h, 2ah, 15h, 15h, 2ah, 3fh
      DB      3fh, 00h, 15h, 3fh, 00h, 3fh, 3fh, 2ah, 15h, 3fh, 2ah, 3fh
      DB      15h, 15h, 00h, 15h, 15h, 2ah, 15h, 3fh, 00h, 15h, 3fh, 2ah
      DB      3fh, 15h, 00h, 3fh, 15h, 2ah, 3fh, 3fh, 00h, 3fh, 3fh, 2ah
      DB      15h, 15h, 15h, 15h, 15h, 3fh, 15h, 3fh, 15h, 15h, 3fh, 3fh
      DB      3fh, 15h, 15h, 3fh, 15h, 3fh, 3fh, 3fh, 15h, 3fh, 3fh, 3fh
      DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
      DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
      DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
      DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
      DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
      DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
```

```
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
DB      00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h
```

```
;*********************************************************************
;
;    STEP7 - Clear Vram.
;
;*********************************************************************

Step7:
      mov     ax, 0A000h
      mov     es, ax
      mov     cx, 0FFFFh /2
      xor     di, di
      xor     ax, ax
      rep     stosw
      ret



; ADDITIONAL SUPPORT ROUTINES
;---------------------------

;-----------------------------------------------------------------------------
;     RunTable - Initialize registers from table
;
;     Entry:   DS:SI - Pointer to register table
;
;     Exit:    n/a
;     Uses:    AX,DX,SI
;-----------------------------------------------------------------------------

br0:
      dec     si              ; Undo last byte overrun

RunTable:
      lodsw                   ; Read next control word
      cmp     ax,-1           ; End of table?
      je      brx             ; Exit if so

      mov     dx,ax           ; Get port in DX
      and     dh,7fh          ; Mask off port write bit
      test    ah,80h          ; Test if port write?
      jnz     br3             ; Jump if so
br2:
      lodsw                   ; Read index,data
      cmp     al,-1           ; End of run
      je      br0             ; Jump back if so
      out     dx,ax           ; Write the data
      jmp     br2             ; Back for more
```

```
br3:
      lodsw                     ; Read data
      out      dx,al            ; Write the data
      jmp      RunTable         ; Back for more
brx:
      ret



TEXT  ENDS
      END      Mode12
```

3.0 MANUAL SETTING OF POWER STATES ON THE SPC8104

To set a powerdown mode, one has to assume the worst case scenario, i.e. we already are in a powerdown mode, the oscillator is disabled and only ports 3DEh/3DFh are decoded. Every powerdown sequence therefore first powers the chip up to a known state and then powers the chip down to the requested state. Note that a small delay is needed after the oscillator is re-enabled, this is to accommodate for the fact it takes a finite amount of time for the oscillators to start oscillating after being enabled.To avoid any visual anomalies or to ensure proper powerdown timings, most powerdown sequences wait for several frames, this is accomplished by counting vertical retraces. (To do this, we must have full address decoding enabled.)
Powerdown sequences are summarized in the following tables.
The powerdown Mode 4 comes in two flavors, if we have any PD clock, we can disable the oscillator. If the PD clock is CLOCKI, we cannot disable the oscillator, as we would in essence disable the PD clock itself. The presence of the PD clock is determined at run-time based on the MD lines.
All power down modes are programmed using AUX[03h]. There are three power down modes available:

- doze mode 1
- doze mode 2
- suspend

For doze mode 2 we can optionally program automatic clock division, suspend mode can disable CLKI altogether, providing we have specified different DRAM refresh clock in AUX[02h].
The following is pseudo-code to illustrate how to set individual power save modes regardless of the current power save mode. The lines marked with * are needed if we want to shut down CLKI in suspend mode:

Doze Mode 1:

```
*     AUX[03h] &= 11101111b ; Make sure CLKI Enabled
*     AUX[03h] &= 00001101b ; clear all power down modes (Full Power)
      AUX[03h] |= 10100000b ; set doze mode 1
```

Doze Mode 2:

```
*     AUX[03h] &= 11101111b ; Make sure CLKI Enabled
*     AUX[03h] &= 00001101b ; clear all power down modes (Full Power)
      AUX[03h] |= 11000000b ; set doze mode 2
```

Suspend:

```
*     AUX[03h] &= 11101111b ; Make sure CLKI Enabled
*     AUX[03h] &= 00001101b ; clear all power down modes (Full Power)
      AUX[03h] |= 00000010b ; keep CLKI, suspend
*     Wait For Vsync        ; wait for at least 3 vertical frames
*     Wait For Vsync        ; before shutting down CLKI
*     Wait For Vsync
*     AUX[03h] |= 00010000b ; kill CLKI
```

Full Power:

```
*     AUX[03h] &= 11101111b ; Make sure CLOCKI Enabled
      AUX[03h] &= 00001101b ; clear all power down modes (Full Power)
```

4.0 IDENTIFYING THE SPC8104

The purpose of the code examples below are to demonstrate the methods to identifying the SPC8104F0A graphics controller. We have two recommended methods:

1. Identifying the SPC8104 through a Seiko Epson BIOS specific BIOS call ;

Sample code (in C):

```
//-------------------------------------------------------------------------
 //always enable Auxiliary register first

 outpw (0x3de, 0x1ade);
 inp (0x3df);

 //try SOLLEX call to get chip product code & revision

 regs.x.ax = 0x7fff;
 regs.h.bl = 1;
 int86(0x10, &regs, &regs);

 if ( regs.x.ax == 0x7f)//SOLLEX call is successful
     {
      if ( regs.h.bh == 0xe2 )
         printf ( "SPC8104 detected.\n" );
      else
         printf ( "SPC8104 not found.\n" );
     }
//-------------------------------------------------------------------------
```

2. By reading registers within the SPC8104F0A and interpreting the register contents.

Sample code (in C):

```
//-----------------------------------------------------------------------------
 //always enable Auxiliary register first

 outpw (0x3de, 0x1ade);
 inp (0x3df);

 outp (0x3de, 8);
 tmp = inp(0x3df) & 0xe0;// read AUX[8]

 if ( tmp == 0xe0 )
    {
      outp (0x3de, 0xf);// read AUX[0f]
      tmp1 = inp (0x3df) & 0xf8;
      if ( tmp1 == 0x10 )
         printf ( "SPC8104 detected.\n" );
      else
         printf ( "SPC8104 not detected.\n" );
    }
 else
    printf ( "SPC8104 not detected.\n" );
//-----------------------------------------------------------------------------
```

5.0 ENABLING AND DISABLING THE SPC8104

The information on Enabling the SPC8104F0A is contained in the examples located earlier in the documents “Initializing the SPC8104 in Mode 03h without a BIOS”; and “Initializing the SPC8104 in Mode 12h without a BIOS”.

The code required to disable the chip when using the enable port 3C3h is:

```
mov     dx, 3C3h
mov     al, 0
out     dx, al
```

6.0 ADJUSTING THE SPC8104 FRAME RATE

The purpose of this document is to describe the methodology behind determining the frame rate on any given panel hooked up to an SPC8104. With the information on how to determine the frame rate, the reader can then extrapolate the data to adjust the frame rate on any give panel. (i.e., modification of the Horizontal and Vertical Non-Display periods).
SPC8104F0A panel frame rate depends on several factors:

1. The pixel clock: this is dependent on the input clock (typically 24MHz) and Integer and Fractional clock divide:

   ```
   PixelClock = FclkI/(FracClockDivide*IntegerClockDivide)
   ```

   Clock Divide factors (integer and fractional) are programmed in AUX[04h].

2. Horizontal total, this is the sum of horizontal display and non-display periods:

   ```
   Htotal = (CRTCB[01h] + CRTCB[05h])
   ```

3. Vertical total, this is the sum of vertical display and non-display periods:

   ```
   Vtotal = (CRTCB[10h] + CRTCB[12h]*2)
   ```

4. 4-bit or 8-bit mode

   The relation between frame rate and panel geometry can be calculated using the following formaulae:

   single 8-bit panel:

   ```
   FR = FclkI/[8*Htotal*Vtotal*FracClockDivide*SlowClock]
   ```

   single 4-bit panel, dual 8 bit panel:

   ```
   FR = FclkI/[16*Htotal*Vtotal*FracClockDivide*SlowClock]
   ```

   Based on these formulae, panel frame rate can be modified with the following restriction: horizontal non-display period must not be smaller than 12 (0Ch). Check the SPC8104F0A Hardware Functional Specification for restrictions on the operating range of the input clock.

   The following code fragment illustrates how to calculate the current frame rate:

```
double CalcFrameRate(void)
   {
   int panel,i,c10,c12,c05,c01,aux2;
   long vtotal;
   double FR,clock;
   static clockdiv[4] = {1,2,4,4};
   static frac[4] = {8,7,6,5};
   panel = 0;              /* assume dual 8bit */
   aux2 = ReadAuxReg(0x2);
   c12  = ReadBreg(0x12);
   c10  = ReadBreg(0x10);
   c01  = ReadBreg(0x01);
   c05  = ReadBreg(0x05);
   if( aux2 & 0x20)        /* single panel */
      {
      panel = 1;           /* 8bit single  */
      if( aux2 & 0x10 )
         panel = 2;        /* 4bit single  */
      }
```

```
switch(panel)
   {
   case 0:  /* 8 bit dual */
      vtotal = 16*((long)(c12*2 + c10));
      break;
   case 1:  /* 8 bit single */
      vtotal = 8*((long)(c12*2 + c10));
      break;
   case 2:  /* 4 bit single */
      vtotal = 16*((long)(c12*2 + c10));
      break;
   }
clock = 24000000.0;
i = ReadAuxReg(0x4);    /* get clock divide bits */
i >>= 2;
i &= 0x3;
clock /= (double)clockdiv[i];
i = ReadAuxReg(0x4);    /* get fract. divide bits */
i &= 0x3;
clock *= (double)frac[i];
clock /= 8;
FR = clock/((((double)c01+c05)*vtotal));
return FR;
}
```

THIS PAGE INTENTIONALLY BLANK

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

Software Utility Disk Installation Guidelines

Drawing Office No. X00-UI-004-01.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

Individual utilities and drivers have thier own installation guides within this Technical Manual. Unfortunately, when it comes time to ship these utilities it is often required that we compress the program images into smaller archives to fit the release on a standard 1.44MB floppy.
Actual installation instructions for the entire release are contained within the file README.TXT on the floppy diskette. Please follow the directions contained in README.TXT for the latest installation instructions.

THIS PAGE INTENTIONALLY BLANK

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

SETFONT.EXE Display Utility

Drawing Office No. X15-UI-001-02.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

SETFONT is a display enhancement utility which can change the size and appearance of on-screen fonts. SETFONT's primary function is to load alternative font sets from the default font sets within the BIOS.

1.1 Program Requirements

Video Controller	**:**	SPC8104, SPC8106, SPC8108, SPC8110
Display Type	**:**	640x480 LCD
BIOS	**:**	Seiko Epson VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Windows 95 only
Windows DOS Full Screen	**:**	Windows 95 only
OS/2 DOS Full Screen	**:**	2.0 or greater

1.2 Installation

Copy the file **setfont.exe** from the distribution disk to a directory on your hard drive that is in your DOS path.

1.3 Usage

SETFONT can be run three ways, as a DOS command line program, a DOS device driver, or a menu interface. SETFONT is a Terminate and Stay Resident (TSR) program.

1.3.0.1 DOS Command Line

SETFONT can be invoked from the DOS command line as follows.

```
setfont [-iFontHeight] [-sScanLines] [ChildProgram] [/?]
```

Where**:**

`FontHeight` can be 8, 14, 16, or 19.
`ScanLines` can be 400 or 480.
`ChildProgram` can be used to load a program in conjunction with a font change. When the child program is exited any font changes stipulated with the loading of the program will be undone. This feature gives the user an easy way to make temporary font changes for use with specific programs.
`/?` produces the usage message shown above.

Example: the DOS command line **`setfont.exe -i8 -s480 x.exe`** will run the program x.exe in 80 characters x 60 rows display mode. On exiting x.exe, the screen will return to the previous setting.

1.3.0.2 DOS Device Driver

SETFONT can be loaded as a device driver. Place a line with the following format in the CONFIG.SYS file.

```
device=[path]setfont.exe [-iFontHeight] [-sScanLines]
```

Example: the line **device=c:\bin\setfont.exe -i8 -s480** in your config.sys file will set the display to 80 x 60 display mode. DOS will now operate with 60 visible lines on the screen verses the original 25 lines.

Font changes stipulated in the CONFIG.SYS take effect on boot-up and remain in effect until altered by running SETFONT from the DOS prompt.

1.3.0.3 Menu Interface

Running SETFONT without parameters produces a menu interface (see below). The menu displays the current state and a font preview window displays the currently selected font. To select an option from a menu, move the highlight using the UP- or DOWN-arrow keys. With the desired options highlighted press ENTER to make the changes and exit SETFONT.

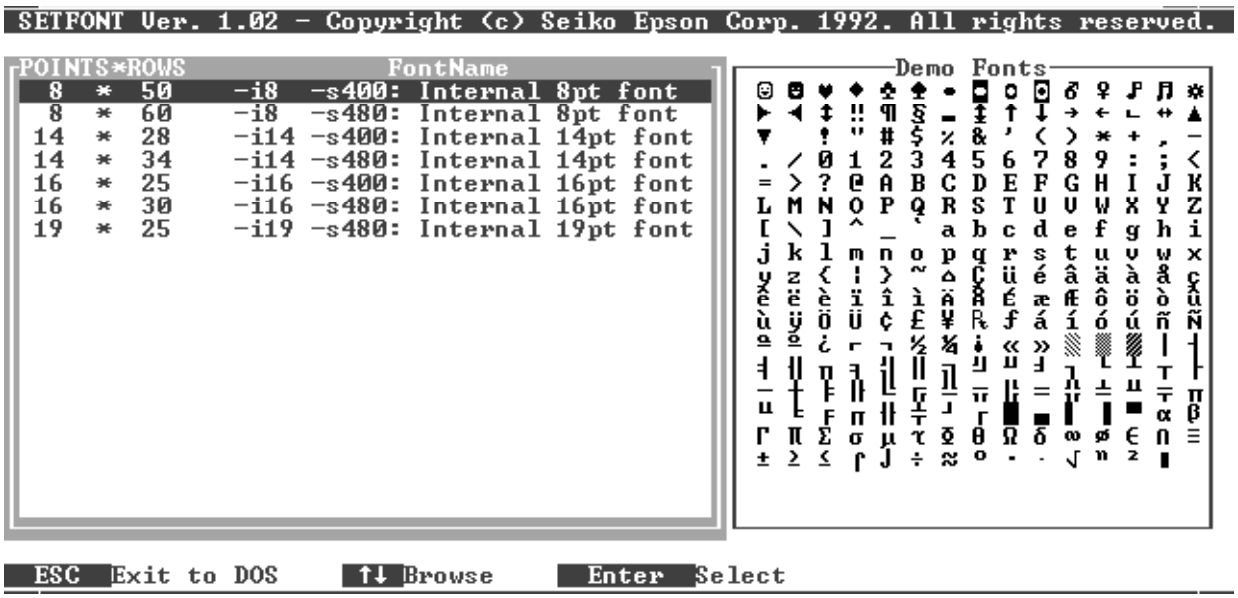


Figure 1 : SETFONT Menu Interface

1.4 Comments

- When SETFONT is run a portion of the program remains resident in memory and occupies approximately 8K bytes of memory.
- Some application programs assume a static font height and loading a different sized font may produce unexpected results in the display.
- The internal 19-point display font is actually a stretched 16-point font on an LCD display. Selecting the 19-point font selects Text Vertical Expansion Enable.

1.5 Program Messages

ERROR: VGA required.

A VGA display must be present to run SETFONT.

ERROR: VGA must be the active display.

The active display must be the VGA to run SETFONT. If the active display is currently a monochrome adapter, run the DOS command **mode co80** to switch to the VGA device, and re-run SETFONT.

ERROR: As a device driver SETFONT cannot run a child program.

To load a child program run SETFONT from the DOS prompt.

ERROR: SETFONT not installed.

SETFONT failed to load as a device driver. For some reason DOS has failed to load the device driver. Check the CONFIG.SYS file for errors.

ERROR: Not enough memory to load SETFONT.

The amount of free system memory below 640 Kilobytes is not enough to run SETFONT. SETFONT requires 8 Kilobytes of system memory. Free up more system memory and run SETFONT again.

ERROR: Cannot execute command line.

SETFONT could not load the specified child program. This may be due to one of many factors. Ensure that the command line is correct and that the child program can be loaded in a command shell.

ERROR: Cannot operate in a command shell.

SETFONT cannot install as a TSR when it is run within another program or within an environment like Windows.

ERROR: SETFONT is already installed as a device driver and cannot be installed again.

SETFONT cannot be installed more than once in the CONFIG.SYS file. Remove the redundant line from CONFIG.SYS.

ERROR: SPC8109 /SPC8108 /SPC8106 /SPC8110 /SPC8104 required.

This version of SETFONT requires an SPC8104, SPC8106, SPC8108, SPC8109 or SPC8110 VGA Controller.

ERROR: Seiko Epson Video BIOS Extension required.

SETFONT requires the Seiko Epson Video BIOS and its Extensions. If your system uses another manufacturer's BIOS this program will not run.

1.6 Engineering Note:

Setfont can also load alternative fonts as supplied in the .FNT files. The format for this .FNT file is as follows:

```
DataHeader STRUC
  DHValidityDW1234h;
  FileSizeDW?     ;
  DHFontHeightDW? ;
  FontWidthDW?    ;
  DataLengthDW?   ;
  ;          1         2         3         4         5
  ;        012345678901234567890123456789012345678901234567890123456789012345
  FileNameDB'     '
  DescriptionDB'012345678901234567890123456789012'
  DWVersionDWVersion
  DataBufferDW0
  ReserveDB0
  DHTrailerDW1234h
DataHeader ENDS
```

ERROR: DOS open, read or close error on the font file.

A DOS file error has occurred.

ERROR: unrecognized font file format.

The font file format is invalid.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

BOLD.EXE Display Utility

Drawing Office No. X15-UI-002-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

BOLD.EXE provides more readable text on LCD displays by changing the default text attribute to high-intensity white from its normal low-intensity white (for text modes only). It can be installed as either a DOS device driver or as a Terminate and Stay Resident (TSR) program. BOLD modifies the user palette profile feature of a VGA BIOS to change the standard low-intensity text attribute 07h to the value 3Fh.

Other text mode application programs which modify the RAMDAC values of entry 3Fh may interfere with the operation of BOLD.

1.1 Program Requirements

Video Controller	**:**	Any VGA
Display Type	**:**	LCD
BIOS	**:**	Any manufacturer's IBM compatible VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Windows 95 only
Windows DOS Full Screen	**:**	Windows 95 only
OS/2 DOS Full Screen	**:**	Yes

1.2 Installation

Copy the program file **bold.exe** from the installation diskette to a directory on your hard drive which is in the DOS path.

1.3 Usage

BOLD may be installed either from the DOS command line as a TSR, or from the CONFIG.SYS file as a device driver.

1.3.0.1 DOS Command Line

```
bold on|off
```

When BOLD is loaded from the DOS command line with the "on" option the text attribute is changed to high intensity. Running BOLD with the "off" option returns the text attribute to normal.

For example: typing at the DOS prompt

```
bold on
```

will load BOLD.EXE as a TSR and will change the text attribute.

An incorrect command line parameter will cause the display of a usage screen.

1.3.0.2 DOS Device Driver

To load the program as a device driver, add the following line to the CONFIG.SYS file:

```
device=[path]bold.exe on|off
```

For example the line:

```
device=c:\bin\bold.exe on
```

in the config.sys file loads BOLD.EXE as a device driver and changes the text attribute.

1.4 Comments

- The program will install under graphics modes but will not take effect until the display is in a text mode.
- This program cannot be installed from within a DOS shell (when a program provides access to the DOS command line and returns to the program via the EXIT command, such as a Windows 3.1x DOS box). It **will** operate in a DOS session under OS/2 or Windows 95.

1.5 Program Messages

`ERROR: Cannot operate in a DOS command shell.`

The program cannot be installed in a DOS shell. Exit your application and re-run BOLD.EXE.

`ERROR: VGA required.`

The program works in VGA text mode only.

`ERROR: BOLD is already installed as a device driver and cannot be installed again.`

BOLD cannot be installed more than once in the CONFIG.SYS file. Remove the redundant line from the CONFIG.SYS.

`Changes to the text mode attribute will not affect the current graphics mode. Changes will appear when the display is placed in text mode.`

BOLD is a text mode program. When the display is in graphics mode BOLD will not alter the appearance of the display. When the display is switched to text mode any changes invoked during graphics mode will become apparent.

1.6 Engineering Notes

This program does not work with the TSENG LABS ET4000 BIOS due to a bug in the BIOS.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

REVERSE.COM Display Utility

Drawing Office No. X15-UI-003-02.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

REVERSE.COM provides a flexible method for the user to change the screen polarity between normal and reverse video (i.e., white on black becomes black on white). REVERSE is not a Terminate and Stay Resident (TSR) program. It can be run multiple times with no side effects.

1.1 Program Requirements

Video Controller	**:**	Any SPC81xx or SPC80xx
Display Type	**:**	LCD
BIOS	**:**	Seiko Epson Video BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes (with limitations), Windows 3.1x and Windows 95
Windows DOS Full Screen	**:**	Yes, Windows 3.1x and Windows 95
OS/2 DOS Full Screen	**:**	Yes

1.2 Installation

Copy the program file **reverse.com** from the installation diskette to a directory on your hard drive which is in the DOS path.

1.3 Usage

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

```
reverse on|off|text|graphics
```

Where:

on	reverses the screen in all modes
off	restores the screen to its default condition
text	reverses the screen for text modes only
graphics	reverses the screen for graphics modes only

An incorrect command line parameter will cause the display of a usage screen.

1.4 Comments

- This utility requires the Seiko Epson video BIOS.
- In a Windows DOS box, the DOS box text mode is emulated with graphics. Text modes are not actually text modes at all, but are graphics modes. Running REVERSE in this case behaves as would be expected when running in graphics mode. Switching to a full screen DOS box using <ALT>-<Enter> switches the emulated text mode to an actual text mode and the program will then behave normally (text, reverse, on and off will apply).

1.5 Program Messages

`ERROR: Requires Seiko Epson Video BIOS Extensions.`

This program works with the Seiko Epson LCD Extensions only.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

POWERSAVE.EXE Power Save Utility

Drawing Office No. X15-UI-004-03.1

THIS PAGE INTENTIONALLY BLANK

1.0 PWRSAVE POWER MANAGEMENT UTILITY

PWRSAVE is a power saving utility for SPC81xx series LCD Controllers. It is a multi-purpose utility which allows the OEM to do power measurements or demonstrate the power management capabilities of the VGA LCD controller. PWRSAVE can be configured as a terminate and stay resident (TSR) program or it can be run as a normally terminated DOS application. PWRSAVE is provided for demonstration purposes only and is not intended for distribution to end users.

1.1 Program Requirements

Video Controller	**:**	All SPC81xx VGA LCD Controllers
Display Type	**:**	640x480 LCD
BIOS	**:**	Seiko Epson VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.2 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	see Usage Note 2 below
Windows DOS Full Screen	**:**	see Usage Note 2 below
OS/2 DOS Full Screen	**:**	2.0 or greater

1.2 Installation

Copy the file **pwrsave.exe** from the distribution disk to your system's disk. For convenience you may choose to place the program in a directory in the DOS path.

1.3 Usage

PWRSAVE can be run from the DOS prompt or from a batch file. To automatically initiate power saving, place "PWRSAVE" in the system's AUTOEXEC.BAT file.

PWRSAVE is invoked from the DOS command line as follows:

```
pwrsave [x] [/u] [/off] [/Tn x] [/?] [/r]
```

Where:

x	sets Power Save mode x, and returns to original Power Save mode upon keystroke or mouse movement. PWRSAVE.EXE does not stay resident in memory in this case.
/u	removes PWRSAVE from memory.
/off	disables all power savings.
/Tn x	installs PWRSAVE in memory and sets Power Save mode x if there is no keystroke or mouse movement occuring for n seconds. Returns to original Power Save mode upon keystroke or mouse movement.
/?	displays the help message.
/r	remain in reduced gray scale mode

1. n is a decimal number from 5 to 3600 and x is a decimal number from 1 to 3 for the SPC8104.
   PWRSAVE 0 is equivalent to PWRSAVE /off.
2. the command line switches x, /off, and /r will work under Windows 95. The switches /u and /Tn x
   do not work under Windows 95. There is an option undr the Windows 95 desktop "Prevent DOS
   Application from Detecting Windows 95". Users should check to ensure this option is turned OFF,
   otherwise you may experience dificulties when using the swithes /u and /Tn x.

1.4 Comments

- The /Tn x option does not work when in a Windows DOS full screen session.
- This utility requires the Seiko Epson Video BIOS and Extensions.
- Power Save mode 1 is intended for use with a RAM type LCD driver in order to take advantage of its self-refresh mode

1.5 Program messages

ERROR: DOS Version 3.2 or later required.

PWRSAVE requires MS-DOS 3.2 or later.

ERROR: SPC81XX with Seiko Epson Video BIOS Extensions required.

PWRSAVE requires an SPC81xx and Seiko Epson Video BIOS. PWRSAVE will fail to operate with any other configuration.

ERROR: Cannot remove PWRSAVE from memory.

PWRSAVE cannot be removed from memory. There may be some other TSRs installed after PWRSAVE.

ERROR: Cannot set Power Save mode x.

An error ocurred while setting Power Save mode x.

ERROR: Cannot operate in a Command Shell.

PWRSAVE has detected that it was requested to run in a DOS command shell as a child program. It will not operate in this mode. Exit the command shell and the application spawning the command shell, and re-run PWRSAVE.

ERROR: This option does not work with Windows 95.

PWRSAVE options /u and /Tn x do not work under Windows 95.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

READCHIP.EXE Diagnostic Utility

Drawing Office No. X15-UI-005-03.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

READCHIP is an OEM utility program which enables the user to read the VGA register contents, lower BIOS data area and video RAM. It is a useful utility for OEMs wishing to submit a problem report for the video controller. If run with a Seiko Epson BIOS, it will try to interpret the BIOS settings.

1.1 Program Requirements

Video Controller	**:**	Any VGA
Display Type	**:**	LCD
BIOS	**:**	Any manufacturer's IBM compatible VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes, Windows 3.1x and Windows 95 (Note 2.)
Windows DOS Full Screen	**:**	Yes, Windows 3.1x and Windows 95
OS/2 DOS Full Screen	**:**	Yes

1. READCHIP uses "stdout" calls and may be redirected to a file or piped to a DOS filter such as MORE.COM.
2. READCHIP works, but the value it reads may not be correct. This is because the VDD has virtualized the CRTC port (i.e., in a Windows DOS box the DOS text mode is emulated with graphics).

1.2 Installation

Copy the file **readchip.exe** from the installation diskette to a directory on your hard drive which is in the DOS path.

1.3 Usage

From DOS prompt, type the following:

```
readchip [/d]
```

`readchip` without any argument will read the Lower BIOS Data Area, CRT Controller, Graphics Controller, Attribute Controller, Sequencer, Auxiliary Registers and other General Registers.
`/d` option will read the DAC Registers (color look up table).
To generate a report, simply type

```
readchip /d > report.txt
```

and the information which READCHIP obtains from the video controller will be stored in the file report.txt.

1.4 Comments

- If READCHIP encounters a non-Seiko Epson VGA chip, it will do a standard IBM VGA chip read, and will only report back values in registers associated with an IBM VGA.
- READCHIP will try and interpret BIOS settings such as panel type. If the BIOS is configured using 81xxCFG the settings may actually be different than default and these interpretations may be incorrect.

1.5 Program Messages

`ERROR: VGA required.`

READCHIP can only read IBM compatible VGA controllers, if any other video controller is present READCHIP will not run.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

WGS.EXE Windows Gray-Scale Utility

Drawing Office No. X15-UI-006-01.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

Under some circumstances while using monochrome LCD panels, the translation of colors to shades of gray produces unattractive and unreadable low contrast results. WGS.EXE is a Windows application which allows the user to select between gray-scaling algorithms which may improve the display quality. WGS takes advantage of the SPC8104F0A graphics controller's special features which offers a selection of two different gray-scaling algorithms.

1.1 Program Requirements

Video Controller	:	SPC8104, SPC8106, SPC8107, SPC8108, SPC8110
Display Type	:	Monochrome LCD
BIOS	:	Seiko Epson VGA BIOS
DOS Program	:	No
DOS Version	:	N/A
Windows Program	:	Yes, Windows 3.1x and Windows 95
Windows DOS Box	:	N/A
Windows DOS Full Screen	:	N/A
OS/2	:	No

1.2 Installation

Copy **wgs.exe** to a sub-directory on your computer hard drive.
In Windows, use the "New" option from the Program Manager File Menu to create a new item and install an icon for WGS in a program group. Alternately, WGS can be run by selecting the "Run" option from the File Menu in either Program Manager or File Manager.

1.3 Usage

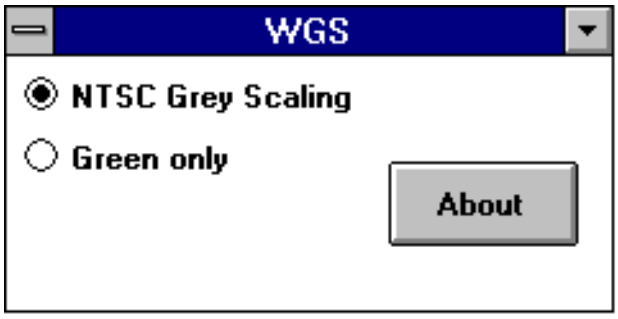


Figure 1 : The WGS Dialog Box

To select a gray-scaling algorithm, click the mouse pointer in the corresponding check box. Changes will take place immediately.
The program options are:

NTSC Gray Scaling

NTSC uses an approximation of the standard 59% Green, 30% Red, 11% Blue standard gray-scaling algorithm.

Green Only

Green Only ignores the Red and Blue components of a color and bases the gray-scale on the Green component only.

About

Clicking on the “About” button will display information about the program including the version number and the copyright notice. Click on the OK button to return to the main dialog box.

1.4 Comments

- The **last** selected gray-scale algorithm selected remains active in a DOS box and after exiting Windows.

1.5 Program Messages

The above error message appears when the program is run on a system which does not use the Seiko Epson Video Controller Extensions (SOLLEX).

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

SESHELL.EXE Shell Utility

Drawing Office No. X15-UI-007-02.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

SESHELL provides a menu interface to demonstrate the operation and command line interfaces for SPC8104F0A DOS utilities. Not all programs shipped with the SPC8104F0A will run under SESHELL, but the majority will.

1.1 Program Requirements

Video Controller	**:**	SPC8104
Display Type	**:**	LCD
BIOS	**:**	Any manufacturer's VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes, Windows 3.1x and Windows 95
Windows DOS Full Screen	**:**	Yes, Windows 3.1x and Windows 95
OS/2 DOS Full Screen	**:**	Yes

1.2 Installation

Copy **seshell.exe** and **seshell.ini** from the installation diskette to a directory on your hard drive which is in the DOS path.

1.3 Usage

To use start program simply type:

```
seshell
```

and press **<Enter>.**

The Demo Shell has three windows. The Programs Window displays a list of the available programs. The Commands Window displays each program with the appropriate parameters for various options. The Description Window provides instructions for each program. The SESHELL screen looks like the following figure:

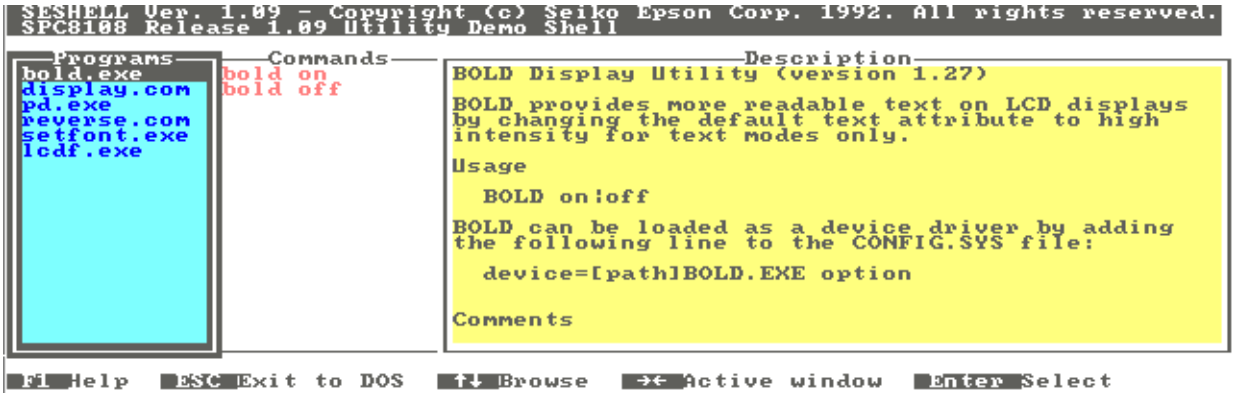


Figure 1 : SESHELL Screen

Running a Program

To run a program select the program in the Programs Window using Up-Arrow or Down-Arrow keys. Activate the Command Window using the Right-Arrow key and select the desired option using the Up- or Down-Arrow keys. Press ENTER to execute the program.

Optionally, after executing the utility, SESHELL may place the user in a DOS shell. To return to SESHELL, at the DOS command prompt enter:

```
exit
```

Viewing the Help File

To view program documentation, activate the Description Window using the Right-Arrow key. Use the Up- and Down-Arrow keys to scroll through the document.

Exiting SESHELL

To exit SESHELL, press **<ESC>**.

1.4 The SESHELL.INI File

The SESHELL program reads a configuration file called SESHELL.INI. This file contains information on the programs available through the shell, their parameters and their help files. SESHELL.INI must reside in the same directory as SESHELL.EXE.

1.5 Comments

- Any program that loads as a TSR will uninstall upon exiting SESHELL. SESHELL is designed to demonstrate programs and their command line parameters, not as a program launcher.
- Users should not attempt to use the uninstall features of a program that was loaded TSR prior to running SESHELL. This may cause unpredictable results.
- SESHELL will run under the DOS session of OS/2, but some programs may not install as a TSR.

1.6 OEM Note

SESHELL can be used by an OEM for end-users if desired. Scripting information can be made available.

1.7 Program Messages

`SESHELL already loaded.`

SESHELL cannot be run from within SESHELL.

`Failed to open seshell.ini.`

The SESHELL.INI file is missing.

`Failed to run PROGRAM NAME.`

SESHELL cannot find or cannot load the requested program. Strike any key to return to SESHELL. Make sure the program is found in the path and retry.

Editing the SESHELL.INI File

The user may wish to change the operation of SESHELL by editing the INI file. Always make a backup copy of the file before editing it in case of error. The SESHELL.INI file must be saved as an ASCII Text file. The following rules must be followed while altering the SESHELL.INI file:

- each program entry must be preceded by [begin]. Only lines following [begin] are valid;
- help text must be sandwiched between [help] and [endhelp];
- there MUST be at least one space in front of and after the "=" character, i.e.:

```
program = aaa.exe
```

not:

```
program=aaa.exe
```

Following is an example of the format of a program entry in the SESHELL.INI file:

```
[begin]
program = ProgramName
cmdline = [ProgramPath] ProgramName Arg1 Arg2 Arg3 Arg4
cmdline = [ProgramPath] ProgramName Arg5
dosshell = no
keeptsr = yes
[help]
Sample Help text is found here.
[endhelp]
```

The "program" directive contains the name of the executable file.
The "cmdline" directive contains the path, the program name, and the command line arguments. If the program locates in the same directory of the SESHELL, the path is not needed. A maximum of four arguments are passed to the utility program. Any further arguments specified with the "cmdline" directive are ignored.
Setting "dosshell" to yes causes SESHELL to load a DOS shell (or command line) after running the specified utility.

Setting the “keeptsr” directive to “YES” allows the program to be loaded as a Terminate-and-Stay-Resident program. However, the TSR program will be removed from resident memory on exiting SESHELL.
The line length of the help text should not exceed 50 characters so that it will fit in the Description Window.

1.8 Typical Program Flow

1. Command line is executed.
2. DOS shell is entered (except when “dosshell = no”).
3. DOS shell terminates (by typing EXIT).
4. Any programs that attempted to stay resident in memory are forcefully released (except when “keeptsr = yes”)
5. Press the ESC key to exit SESHELL. Any programs that attempt to stay resident are forcefully released, including “keeptsr=yes”.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

SHOWMODE.EXE Demonstration Program

Drawing Office No. X15-UI-008-01.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

SHOWMODE displays all available video modes for Seiko Epson VGA/SVGA controllers with Seiko Epson Video BIOS Extensions. It also displays standard VGA modes for other makes of IBM compatible VGA controllers.

1.1 Program Requirements

Video Controller	**:**	Any VGA
Display Type	**:**	LCD
BIOS	**:**	Any manufacturer's VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Windows 95 only
Windows DOS Full Screen	**:**	Yes, Windows 3.1x and Windows 95
OS/2 DOS Full Screen	**:**	Yes

1.2 Installation

Copy the file **showmode.exe** from the installation diskette to a directory on your hard drive which is in the DOS path.

1.3 Usage

To display all available video modes (including both VGA and SVGA modes) in order, enter the following command line from the DOS prompt:

```
showmode
```

To list all available video modes, type the following command line:

```
showmode /l
```

To display a specific video mode, type the following command line:

```
showmode ModeNumber
```

Where **ModeNumber** is the hex number of the desired video mode, as in **showmode 101**

To display the SHOWMODE usage message, at the DOS command prompt enter:

```
showmode /?
```

1.4 Comments

- SHOWMODE displays the standard IBM VGA modes assuming standard IBM VGA numbers (i.e., mode 3 is 80x25 characters). This program may have difficulties displaying some customized video modes if other applications have changed the display from the IBM default settings.
- If a non-Seiko Epson BIOS or non-Seiko Epson chip is present, SHOWMODE will restrict its display options to IBM standard modes 0 through 13 hex.

1.5 Program Messages

`ERROR: cannot allocate memory.`

There is not enough memory to run this program. Free up more system memory below 640 Kilobytes and re-run SHOWMODE.

`ERROR: VESA function #n not supported.`

The VESA function #n is not supported. The Video Electronics Standards Association (VESA) BIOS extensions are required for extended mode support.

`ERROR: VESA function #n not successful.`

An error occurred while setting VESA function #n. This is an unknown error condition which could mean that in the current hardware setup the function failed. Change the hardware or BIOS configuration and try again.

`Error setting mode xxh. This mode is not supported in hardware`

VESA BIOS supports mode xxh, however, the hardware does not support this mode. This error condition can occur when a high resolution mode is not supported by the LCD panel. i.e., mode 102h is an 800x600 high resolution mode, CRT monitors can handle it, but LCD panels (640x480) do not handle this mode.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

LCDF.EXE Display Utility

Drawing Office No. X15-UI-009-01.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

LCDF is a display utility for changing the LCD panel frame rate using the SPC8104 VGA Controller. This is desirable for adjusting the contrast and viewability of the display under different lighting conditions. LCDF can be loaded as a command line executable with parameters, or as an interactive executable when no parameters are specified. To change the frame rate, LCDF modifies the content of the CRTC_B[05] register in the SPC8104. LCDF is provided for demonstration purposes only and is not intetended for distribution to end-users.

1.1 Program Requirements

Video Controller	**:**	SPC8104, SPC8106, SPC8108
Display Type	**:**	640x480 LCD
BIOS	**:**	Any manufacturer's IBM compatible VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	No
Windows DOS Full Screen	**:**	Yes, Windows 3.1x and Windows 95
OS/2 DOS Full Screen	**:**	2.0 or greater

1.2 Installation

Copy the file **lcdf.exe** from the distribution disk to your system's disk. For convenience you may choose to place the program in a directory in the DOS path.

1.3 Usage

LCDF is invoked from the DOS command line as follows:

```
lcdf [n][/?]
```

where:

- n will set the desired frame rate frequency in decimal Hertz. (e.g. 78 = 78Hz) and will display the parameters programmed as well as the copyright notice and version number.
- /? displays the copyright notice, version number and usage screen.

When no command line option is specified the interactive mode of the program is run. When in interactive mode, LCDF displays a graphic pattern and current frame rate.
The frame rate can be increased/decreased by repeatedly pressing 'U' or 'D'. The quality of the displayed pattern is used to determine the optimal frame rate for a given panel.
Pressing the 'Q' key exits the program.

1.4 Comments

- The program will automatically determine the panel type (single or dual) and calculate the correct frame rate.
- High frame rates could potentially damage the panel. Consult the manufacturer's specification for recommended values.
- The displayed frame rate in interactive mode assumes a 24.000 MHz clock for the SPC8104.
- The frame rate will always return to the default when the computer is restarted.

1.5 Program Messages

`ERROR: SPC8108 / SPC8106 / SPC8104 not present!`

This program requires the SPC8108, SPC8106, or SPC8104 VGA LCD controller.

`ERROR: Frame rate not valid!`

The frame rate requested exceeds the video controller limits, try a number closer to 78 Hz.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

VRTEXP.EXE Display Utility

Drawing Office No. X15-UI-010-02.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

The VRTEXP program is used to control Text and Graphics Vertical Expansion on an LCD panel. This program calls the Seiko Epson BIOS Extentions (SOLLEX) function which modifies LCD control bits within the Seiko Epson video chip. VRTEXP.EXE does not Terminate and Stay Resident (TSR) and can be run as many times as the user desires.

1.1 Program Requirements

Video Controller	**:**	SPC8104, SPC8106, SPC8108, SPC8110
Display Type	**:**	LCD
BIOS	**:**	Seiko Epson VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes, with limitations Windows 3.1x and Windows 95
Windows DOS Full Screen	**:**	Yes, Windows 3.1x and Windows 95
OS/2 DOS Full Screen	**:**	Yes

1.2 Installation

Copy the file **vrtexp.exe** from the distribution disk to a directory on your hard drive that is in your DOS path.

1.3 Usage

VRTEXP.EXE is run from the command line as follows:

```
vrtexp /T | /G | /B | /N
```

Where:

/T enables Text Vertical Expansion and disables Graphics Vertical Expansion.
/G enables Graphics Vertical Expansion and disables Text Vertical Expansion.
/B enables both Text & Graphics Vertical Expansion (BOTH)
/N disables both Text & Graphics Vertical Expansion (NONE)

1.4 Comments

- This utility requires the Seiko Epson video BIOS.
- In a Windows DOS box, the DOS box text mode is emulated with graphics. Text modes are not actually text modes at all, but are graphics modes. Running VRTEXP in this case behaves as would be expected when running in graphics mode. Switching to a full screen DOS box using <ALT>-<Enter> switches the emulated text mode to an actual text mode and the program will then behave normally.

1.5 Program Messages

ERROR: Seiko Epson BIOS Extensions required.

This program works with the Seiko Epson LCD Extensions only.

ERROR: SPC8104, 8106, 8108, 8109 or 8110 required.

This version of VRTEXP requires an SPC8104, SPC8106, SPC8108, SPC8109, or SPC8110 VGA controller.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

8104CFG.EXE BIOS Configuration Utility

Drawing Office No. X15-UI-011-01.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

8104CFG is the OEM utility for custom configuring the SPC8104 VGA Controller BIOS binary image. This utility allows the OEM, or product support people to easily customize the Seiko Epson SPC8104 BIOS for the OEM's requirements. These configurations may be as simple as modifying the Signon Message or as complex as modifying the default panel support.

8104CFG consists of four files; 8104cfg.exe, 8104cfg.hlp, 8104cfg.hx, and 8104cfg.hs. This program has three user levels incorporated into it for ease of operation. These levels are:

- **typical**
- **sophisticated**
- **expert**

We highly recommend reading the on-line help documentation for details on the configuration options before making any changes.

1.1 Program Requirements

Video Controller	**:**	Any VGA
Display Type	**:**	Any VGA
BIOS	**:**	Any manufacturer's VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes, Windows 3.1x and Windows 95
Windows DOS Full Screen	**:**	Yes, Windows 3.1x and Windows 95
OS/2 DOS Full Screen	**:**	2.0 or greater

1.2 Installation

Copy the files **8104cfg.exe**, **8104cfg.hlp**, **8104cfg.hx**, and **8104cfg.hs** from the distribution disk to your system's disk. For convenience you may choose to place the files in a directory in the DOS path.

1.3 Usage

8104CFG is invoked from the DOS command line as follows:

```
8104cfg [/s|/x][/?]
```

where:

no parameters starts 8104CFG in the typical user mode.

`/s` starts 8104CFG in the sophisticated user mode.

`/x` starts 8104CFG in the expert user mode.

`/?` displays the copyright notice and version number.

1.4 Comments

The 8104CFG program is menu driven with extensive on-line context sensitive help. The user interface is by either keyboard or mouse.
The following BIOS properties can be configured:

By the Typical User - These are the most common BIOS configuration changes, and typically can be modified without risking functionality.

IBM RAMDAC and Palette compatibility: mode 7, mode 11; (ISA or PS/2)
BIOS Boot segment
BIOS Signon message
Panel preferences for each supported panel:
- text mode expansion
- text mode screen polarity
- graphic mode expansion
- graphic mode screen polarity
- gray scale method
- autocenter
- blink rate

By the Sophisticated User - These are more sophisticated changes which can affect the functionality of the graphics system. These changes are typically made by the system board designer who understands the layout.

Same as the Typical User plus the following:
Re-assign MD lines - any of the 16 supported panels can be assigned to any MD lines value, thus selecting a “default” panel
IRQ2 enable / tristate
Power down clock source
Power down clock (Doze clock) divisor
Memory refresh clock period

By the Expert User - These are the most sophisticated changes that can be made without producing a custom BIOS. These options allow the definition of new panel types, and should only be made by qualified individuals who understand both the chip and panel interface issues. Changes with this option enabled can damage panels.

Same as the Sophisticated User plus the following:

- Edit a new panel entry -up to 16 different panels. Any panel can be assigned to any MD lines value.
- A variety of configurable options are available to accommodate various panel types.
- The following properties are configurable independently for each panel:
  - panel type (dual, single, 8-bit, 4-bit, MIM)
  - frame rate
  - non-display vertical lines
  - panel dimensions
  - miscellaneous LCD support bits:
    - slow CLKI support
    - clock divide support (integer and fractional)
    - sequencer scaling support
    - XSCL on / off
    - LP timings - 8 / 6 clocks
    - WF count - 0 to 63
    - configurable power off protection for BIOS video mode set

1.5 Using Your Newly Configured BIOS

To use your newly configured BIOS, program a new EPROM with the new image. Depending on your EPROM programmer, you may need to convert the image format from binary to one which your EPROM programmer can use.

THIS PAGE INTENTIONALLY BLANK

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

Windows 3.1 16-Color Panning Display Driver Including VLIMIT.EXE Display Utility

Drawing Office No. X15-DI-001-01.1

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

The SPC8104 Panning Display Driver produces a usable display desktop in Windows 3.1x which can exceed the size of the physical display device. The Panning Driver allows the physical display to act as a window which moves over the larger virtual desktop. The driver uses hardware panning and scrolling techniques keep the mouse pointer visible on the screen.
This driver uses OEM-definable physical and virtual display sizes and is capable of supporting many different combinations of vertical and horizontal resolutions including small panel sizes.

1.1 Requirements

Video Controller	**:**	SPC8104, SPC8106, SPC8108, SPC8110
Display Type	**:**	OEM definable LCD or CRT
BIOS	**:**	Seiko Epson VGA BIOS
DOS Program	**:**	No
DOS Version	**:**	3.0 or greater
Windows Program	**:**	Windows 3.1x only
Windows DOS Box	**:**	N/A
Windows DOS Full Screen	**:**	N/A
OS/2 DOS Full Screen	**:**	No

1.2 Installation

If you have not already installed Windows 3.1x use the Custom Setup procedure outlined in *Getting Started with Microsoft Windows* to install the Virtual Driver along with Windows.
If Windows has been installed previously with another display driver, use SETUP.EXE (found in the Windows directory) to install the Virtual Driver. Follow the instructions under Installing a Device Driver Not Supplied with Microsoft Windows 3.1 in the Microsoft Windows User's Guide. From the list of displays presented by SETUP, choose "Other display (requires disk from OEM)". When prompted, insert the SPC8104 Utilities Diskette in the appropriate drive and follow the on-screen instructions.

1.3 Operation

The Panning Display driver can work with a physical display of practically any size including 200, 240, 256 and 400 line panels. Although virtual sizes can range up to 1024 x 768 pixels, only three virtual sizes are supported by the supplied OEMSETUP.INF file. They are:

- PHYSICAL (virtual size equals physical size);
- 640 x 480 (standard VGA);
- 800 x 600 (standard SVGA).

During operation the displayed image shifts beneath the window to keep the mouse pointer visible at all times. For example, as the mouse pointer approaches the right side of the screen, the screen image will shift left to expose more of the right side of the image. As the mouse pointer

moves to the bottom of the screen, the image will shift upwards to expose more of the bottom of the virtual display area.

1.4 Comments

- This Driver requires the Seiko Epson video BIOS.
- When a Windows program is maximized it will expand to fill the entire virtual screen, even if the physical display is only 640 x 480 pixels or less. Because Windows or an application program has no way of determining the physical display size, buttons and scroll bars may not appear in the expected position or even on the display. The companion program VLIMIT.EXE can be used to turn the virtual display feature on and off without reinstalling the driver or restarting Windows.

2.0 VLIMIT.EXE UTILITY

VLIMIT is a Windows 3.1x utility used to control the behavior of the Windows 3.1 Panning Display Driver. When a window is maximized by clicking on the Maximize Button or double clicking the Title Bar, Windows attempts to fill the entire display area. If the panning driver has created a display area larger than the physical display, some of the window will fall outside the display. VLIMIT prevents a maximized window from exceeding the limits of the physical display.

2.1 Installation

Copy the files VLIMIT.EXE, VLIMIT.DLL and VLIMIT.HLP to a directory in the path.
To install VLIMIT as an icon in a program manager group use either of the following two methods:

1. use the “New” option from the Program Manager’s “File” menu

or

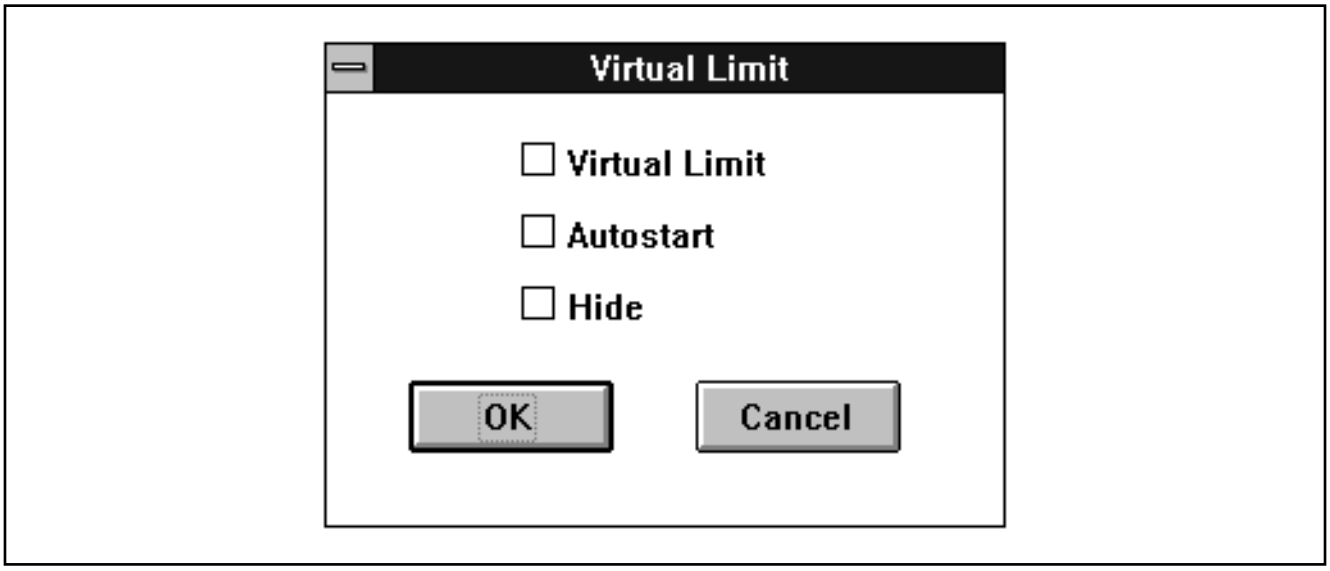
2. use the file manager to “drag” the program into a program manager group.

Both techniques are described in the Microsoft Windows User’s Guide.

VLIMIT requires the panning display driver W31GRAX.DRV to be installed.
Install the display driver before attempting to run VLIMIT.

2.2 Usage

Run VLIMIT by double clicking on the Program Manager icon or by selecting the “Run” option from the Program Manager’s File menu.
The VLIMIT Control Panel is pictured below. Options are selected by checking the appropriate boxes and clicking the OK button. Changes take place immediately.



2.2.0.1 The VLIMIT Interface

The Virtual Limit Check Box

This Check Box enables virtual limiting. When checked VLIMIT restricts the size of a maximized window to the size of the display area. This means that when a window is maximized on a 640x480 display it will limit its maximized size to 640x480, not the size of the virtual desktop, which could be 800x600.

The AutoStart Check Box

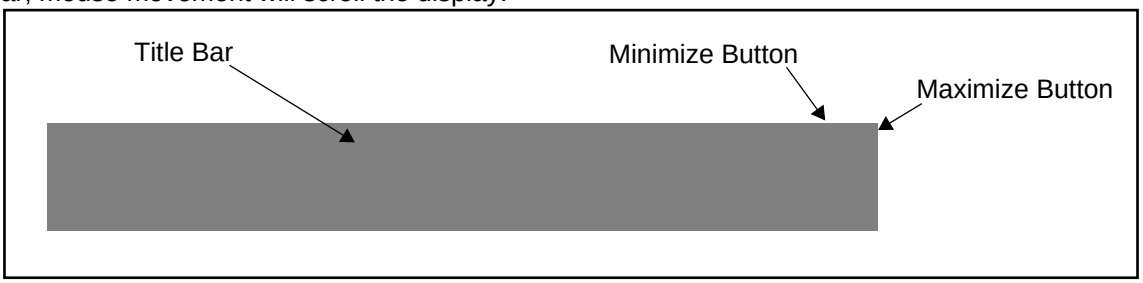
Check Autostart to have VLIMIT run automatically every time you start Windows. Options in effect during the last VLIMIT session will still be in effect on the next windows start-up.

The Hide Check Box

This option hides the VLIMIT icon. It is not necessary for the icon to be displayed during the normal operation of VLIMIT. When VLIMIT is hidden it does not appear in the task list of Task Manager, even though it is still active. To change settings or close VLIMIT, rerun the program from its icon.

2.3 Comments

- **Scroll Locking** - Depending on the method used to maximize the window, the virtual display may be scrolled or frozen at the current position. If the window is maximized by clicking on the Maximize Button, the display is locked to prevent scrolling. Attempting to move the mouse beyond the display boundaries will not scroll the screen. If the window is maximized by double clicking the left mouse button on the window's Title Bar, mouse movement will scroll the display.



- Under some circumstances VLIMIT turns off Scroll Locking to prevent the screen from being locked with the Minimize, Maximize and Control Menu Buttons beyond the reach of the user. Anytime the top level window is minimized Scroll Locking will be released. This prevents a window which exceeds the bounds of the physical screen from having its control buttons locked off the screen when it becomes the new top level window. Minimizing and then maximizing the new top level window will restore Scroll Locking.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

SDU8104B0B Rev 1.0 ISA Bus Evaluation Board

Drawing Office No. X15-AN-001-01

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY BLANK

1.0 SDU8104B0B REV. 1.0 EVALUATION BOARD

This manual reflects the use of the SDU8104B0B Rev. 1.0 evaluation board in conjunction with the SPC8104F0A Controller while operating in a ISA-Bus environment. The SDU8104B0B Rev. 1.0 will operate as a stand-alone video adapter card with on-board video BIOS support. Refer to the SPC8104F0A Functional Specification, Drawing Office No. X15-SP-001-xx for further details.

1.1 Features

- 128 pin QFP15-128 package (socketed)
- mixed SMD / PTH devices
- Monochrome STN LCD support
- MIM support
- 8 / 16-bit ISA-Bus support
- On-board video BIOS with additional support logic provided
- External oscillator support
- Power Down Clock support
- 3.3v, 512K DRAM support (256K x 16, symmetrical, Mitsubishi, M5M4V4260-8 TSOP)
- Configuration Options via DIP-Switch
- Support for Software Power Save Modes
- Support for Hardware Power Save Modes (SUSPEND# and DOZE#)
- On-board -23V adjustable power supply (VLCD for dual monochrome panels)
- Adjustable Core voltage (VDD CORE)(PCBVDD1)
- Adjustable IO voltage (VDD IO)(PCBVDD2)
- Adjustable 3.3v Power Supply
- Voltage translation circuitry for ISA-Bus signals
- Voltage translation circuitry for panel interface signals
- LCDBIAS Control (LCDPWR#)
- Power Measurement capabilities
- Test-point Header strips
- Prototype area

1.2 Installation and Configuration

The SDU8104B0B Rev. 1.0 directly supports the 16-bit ISA-Bus architecture. The SPC8104F0A has 6 configurable inputs (MD[6:5], MD[3:0]) which are read on power-up. DIP switches are provided for configuration of each one of these inputs.

Configuration DIP Switch Settings

Switch	Signal	Closed	Open
SW1-1	MD0	LCD Panel Config bit 0	see table below
SW1-2	MD1	LCD Panel Config bit 1	see table below
SW1-3	MD2	LCD Panel Config bit 2	see table below
SW1-4	MD3	LCD Panel Config bit 3	see table below
SW1-6	MD5	LCD Signal State in Suspend Mode: Hi-Z	LCD Signal State in Suspend Mode: Low
SW1-7	SUSPEND#	SUSPEND# Enabled	SUSPEND# Disabled
SW1-8	DOZE#	DOZE# Enabled	DOZE# Disabled

The polarity of the Dip-Switches is defined as (Closed = ’0’ or ’low’).

External 100 K ohm pull-ups have been added to MD[6:5]. MD6 is not user configurable for this evaluation board and has been set to ’high’ to support symmetrical DRAM.

Use as the default setting

Jumper Setting:

	1-2	2-3
JP1	16-Bit BUS Support	8-bit BUS Support
JP2	16-Bit BUS Support	8-bit BUS Support
JP3	16-Bit BUS Support	8-Bit BUS Support
JP4	CPU BUS tri-stated	CPU BUS Enabled
JP5	Buffered IRQ connected to Bus	Buffered IRQ not connected to Bus

All JP1-3 must be in the same position for correct operation either all 1-2 or all 2-3.

JP4 should only be used for testing of power consumption with no BUS activity (default setting should be position 2-3)

Use as the default setting

ISA Bus IRQ’s are NOT shared signals. Therefore only one device can be assigned to any one interrupt signal. If you encounter any contention type problems with IRQ connected to the Bus, change the jumper position to disconnect it and re-run the specific test causing the problem. Position 2-3 should be used as default and position 1-2 used for test purposes only.

LCD Panel Configuration

The BIOS configuration for different LCD panels is done through the state of the MD lines. The

following tables describes the settings and the associated panel type.

Pin Name	Configuration Pin Functionality
MD0	B0 of panel table
MD1	B1 of panel table
MD2	B2 of panel table
MD3	B3 of panel table

bit 3	bit 2	bit 1	bit 0	Resolution	Mode	Bits	Comments
1	1	1	1	640x480	MIM	4	MIM (landscape)
1	1	1	0	640x480	Single	8	Monochrome
1	1	0	1	640x200	Single	4	Monochrome
1	1	0	0	320x240	Single	4	Monochrome
1	0	1	1	Reserved			
1	0	1	0	Reserved			
1	0	0	1	Reserved			
1	0	0	0	Reserved			
0	1	1	1	Reserved			
0	1	1	0	640x480	TFT	18	TFT
0	1	0	1	640x480	Dual	8	Monochrome
0	1	0	0	480x640	MIM	4	MIM (portrait)
0	0	1	1	Reserved			
0	0	1	0	Reserved			
0	0	0	1	Reserved			
0	0	0	0	OEM Config			

OEM Configuration options allows for very specific resolution/timing criteria to be programmed for a given panel. Refer to SPC8104F0A LCD Panel Configuration Options, Drawing Office No. X15-AN-xxx-xx for further details.

LCD Signal Connector Pinouts

LCD TFT Connector J6		TFT				LCD Mono Connector J5		Mono STN LCD		MIM
Pin Name	**Pin No.**	**18-bit color**	**12-bit color**	**9-bit color Sharp**		**Pin No.**	**Pin Name**	**8-bit**	**4-bit**	**4-bit**
LD2	1	R4	R2	R1		1	LD0	LD0		D0
LD3	3	R5	R3	R2		3	LD1	LD1		D1
LD2	5	G0				5	LD2	LD2		D2
LD3	7	G1				7	LD3	LD3		D3
LD2	9	R0				9	UD0	UD0	UD0	
LD3	11	R1				11	UD1	UD1	UD1	
LD0	13	R2	R0			13	UD2	UD2	UD2	
LD1	15	R3	R1	R0		15	UD3	UD3	UD3	
LD2	17	B0				17				
LD3	19	B1				19				
LD0	21	B2	B0			21				
LD1	23	B3	B1	B0		23				
LD0	25	G2	G0			25				
LD1	27	G3	G1	G0		27				
LD2	29	G4	G2	G1		29				
LD3	31	G5	G3	G2		31				
LD2	33	B4	B2	B1		33				
LD3	35	B5	B3	B2		35				
XSCL	37	PCLK	PCLK	PCLK		37	XSCL	XSCL	XSCL	PCLK
	39					39				
LP	40	HS	HS	HS		40	LP	LP	LP	HSYNC
YD	38	VS	VS	VS		38	YD	YD	YD	VSYNC
WF	36	DE	DE	DE		36	WF	WF	WF	DE
VDDH	34					34				
VLCD	32					32		VLCD	VLCD	
LCD PWR	30	LCD PWR	LCD PWR	LCD PWR		30	LCD PWR	LCD PWR	LCD PWR	tbd
N/C	28					28				tbd
N/C	26					26				tbd
+5 V	24	+5 V	+5 V	+5 V		24		+5 V	+5 V	tbd
+12 V	22	+12 V	+12 V	+12 V		22		+12 V	+12 V	tbd
GRND	20-2	GRND	GRND	GRND		20-2		GRND	GRND	GND

These configurations require two connectors, one for TFT panels and one for STN panels. These connectors are backwards compatible with the SDU8110BOB rev 2.0 evaluation board.

1.3 Technical Description

16-Bit ISA-Bus Support

This board directly supports the 16-bit ISA-Bus and is intended to be used as a ISA-Bus Target add-in-board. All SPC8104F0A related electrical connections come from the standard in-line ISA-Bus connectors. Note that all Bus related signals are separated from the ISA-Bus and the SPC8104F0A by voltage translations circuits. The nature of this design is such that the bus interface I/O pins of the SPC8104F0A are 3.3v only. 8-Bit ISA is supported through JP1-3 and requires an 8-bit only BIOS.

512K DRAM Support

A one chip 3.3v, 512K solution is supported using a 256K x 16 DRAM. Symmetrical DRAM in an TSOP package is supported. As the specific DRAM used are 3.3v, there are no voltage translators in between the 8104 and the memory.

Monochrome LCD Support

The SPC8104F0A 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 (J5) and pass through a 3 V to 5V voltage translation circuitry. The interface signals are alternated with grounds on the cable to prevent cross talk and noise related problems.
Refer to *LCD Panel Configuration and LCD Signal Connector Pinouts* above for your specific settings.

MIM Support

The SPC8104F0A also supports 4-bit MIM panels. All the necessary signals are provided on the 40-pin ribbon cable header (J5) and pass through a 3 V to 5V voltage translation circuitry. The interface signals are alternated with grounds on the cable to prevent cross talk and noise related problems.
Refer to *LCD Panel Configuration and LCD Signal Connector Pinouts* above for your specific settings

TFT Support

The SPC8104F0A also supports 9-, 12-, and 18-bit TFT panels for the purpose of testing only. Support of the TFT panels allows for more complete testing of the SPC8104F0A. All the necessary signals can be found on the 40-pin ribbon cable connector (J6) and pass through a 3 V to 5V voltage translation circuitry. Note that support for TFT is done in gray shades only.
Refer to *LCD Panel Configuration and LCD Signal Connector Pinouts* above for your specific settings

LCD Power Supply Control

A control signal is provided on the connector (J5 and J6), LCDPWR#. This signal is a low current direct output from the SPC8104F0A and are intended to be used in-conjunction with external circuitry to provide higher current switching. It is used to control the -23v Power Supply (VLCD).

VGA BIOS Support

A 32K EPROM contains the VGA BIOS. Support logic consists of an Address Decoder (C000H-C7FFH), and a Data Buffer operating from the ISA-Bus. With the BIOS installed the board will operate as a stand-alone Video adapter.

Power Save Modes

The SPC8104F0A supports 3 Power Save Modes that can be activated via either hardware or software. A utility program is supplied to control both these modes. Software control is performed by directly writing the SPC8104F0A associated internal registers. The hardware control is performed by input pins on the SPC8104F0A. These inputs are controlled with the on-board DIP-SWITCH (SW-7 and SW-8).

Negative Power Supply

The majority of Dual Panel Dual Drive Monochrome STN LCD panels require a negative power supply to provide between -18 V and -23 V (I_{out}=45mA) (VLCD). For ease of implementation, such a power supply has been provided as an integral part of this design. The signal VLCD can be found on J3 (pin 32) and its output voltage can be controlled by R10 and control signal LCD-PWR#.
R10 (100K potentiometer) can be adjusted to provide an output voltage from -18 V to -23 V. LCDPWR# is a control signal provided by the SPC8104F0A which is used to enable/disable the VLCD power supply. This signal guarantees correct power sequencing and prohibits subsequent panel damage.
Determine the panel's specific power requirements and set R10 accordingly before connecting the panel.

> If a positive bias voltage is required, an external source must be used. Ensure that the proper power sequencing can be observed or panel damage may occur.

External Oscillator Support

The SPC8104F0A requires an external clock for the CLKI input (pin 77). The clock source for MIM panels should be 24 MHz while the clock input for passive panels should be 28 MHz (in order to provide a reasonable frame rate). The 5.0v oscillator outputs pass through voltage translators for interface into the 8104 3.3v inputs.

Power Down Clock Support

A 32 KHz oscillator is provided for connection to the PDCLK input (pin 79). This clock source is required to provide the DRAM refresh cycles in some power down modes. The 5.0v oscillator outputs pass through voltage translators for interface into the 8104 3.3v inputs.

Adjustable Core Logic Power Supply

An independent power supply for the core section (VDD CORE) is provided. This power supply (PCBVDD1) can be adjustable between 1.8 V and 3.9 V @ 1 A. Adjust R13 appropriately.

Adjustable I/O Logic Power Supply

Another independent power supply (PCBVDD2) is required to supply the 3.3 V I/O VDD pins on the SPC8104F0A. All of the I/O pins are connected to the one power rail. This power supply (PCBVDD2) can be adjustable between 1.8 V and 3.9 V @ 1 A. Adjust R19 appropriately.

3.3V Power Supply

There is a third on-board power supply which is used to provide 3.3v to devices that should not be adjusted when testing the 8104 under variable voltage conditions. Refer to the schematics for details.

Voltage Translation Circuitry

Voltage translation circuitry is required on all ISA-Bus interface signals to prevent the forward biasing of the input protection diodes of the SPC8104F0A Bi-directional control of the data bus is controlled by the output signal RDACK#. The LCD interface also has translation circuits as most panels require a 5.0v interface.

Power Measurement Capabilities

All associated 8104 power pins (VDD CORE and VDD IO) will have a 1 ohm 1% resistor in series with it for calculation of power consumption of the individual logic sections.

Test-point Header Strips

All of the pins on the SPC8104F0A are connected to surrounding header strips for easy signal debugging and testing. Refer to the pin descriptions in X15-SP-001-xx for further details. These header strips will be mounted on the Solder-side of the board with an associated silk-screen to identify the pin numbers.

Prototype Area

An area of plate-thru holes with power and ground connections top and bottom is provided on the board for prototyping additional logic.

PCB Layout Considerations

To provide low signal noise as well as reduced capacitive loading, the LCD Connector is located near the SPC8104F0A.
Due to the density of the socket, a 6 layer board was necessary so that all signal traces could be routed to and from the socket pins. Trace thickness must also be reduced to 7mil with 7mil spacing.

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, C13-44	0.01uF	1206 pckg
2	1	C2	68pF	1206 pckg
3	2	C3, C4	56uF/35V	LXF35VB56RM6X11LL
4	3	C5-7	0.1uF	1206
5	5	C8-10, C45-46	10uF / 16V	D-SIZE 10uF Tantalum
6	2	C11-12	2.2uF	PTH radial .1 spc.
7	1	D1	1N4148	Signal Diode / PTH
8	3	JP1-4	3 pin Male Header	.1 spec Male Header (supply with Shunt)
9	1	J5-6	CON40A	Shrouded Header 40 pin Dual-row center-key
10	4	J7-10	32 Pin Dual Row Male Header	.1 spc. 16 pin per side dual row header strip
11	1	Q2	2N3905	PNP Signal Transistor
12	1	Q1	2N3903	NPN Signal Transistor
13	2	R1-2	1 OHM	PTH 1 Ohm 1 %
14	4	R3-6	10K	10K Ohm / 1206 pckg / 5%
15	2	R7, R11	1K	1K Ohm / 1206 pckg 5%
16	4	R8-9, R22-23	100K	100K Ohm / 1206 pckg / 5%
17	1	R10	100K	100K Ohm Trim Pot (Bourns 3386 type)
18	3	R12, R15, R18	240	240 Ohm / 1206 pckg / 5%
19	3	R13, R16, R19	500	500 Ohm Trim Pot (Bourns 3386 type)
20	3	R14, R17, R20	100	100 Ohm / 1206 pckg / 5%
21	1	R21	27K / 16 pin DIP	Bourns 4116R-001-273
22	2	R24-25	4.7K	4.7K Ohm / PTH / 1/4w / 5%
23	1	S1	SW DIP-8	Switch Dip 8 position
24	1	U1	SPC8104F0A	QFP-15 pckg (socketted)
25	2	U2-3	SN74LVT16244	48 pin SSOP pckg
26	1	U4	SN74LVT244	DW020 pckg
27	1	U5	SN74LVT16245	48pin SSOP pckg
28	1	U6	M5M4V4260-8 TSOP	256Kx16 DRAM / 44 pin TSOP pckg
29	1	U7	EPN001	XENTECK - Negative Power Supply Module
30	3	U8, U9, U10	LM317T	TO-220 pckg
31	1	U4	NM27C256Q-200	EPROM socket and component
32	1	U5	74LS244	IC 74LS244 BUFFER DIP pckg
33	1	U6	74LS688	IC 74LS688 COMPARATOR DIP pckg
34	1	U14	24.0MHz OSCILATOR	assembled in 14 pin socket
35	1	U15	32KHz OSCILATOR	supply only 14 pin socket (SMOS will supply actual oscilator)
36	2	U16-17	74HCT244	20 pin DIP assembled in socket

Appendix B SDU8104B0B Rev. 1.0 Schematic Diagrams

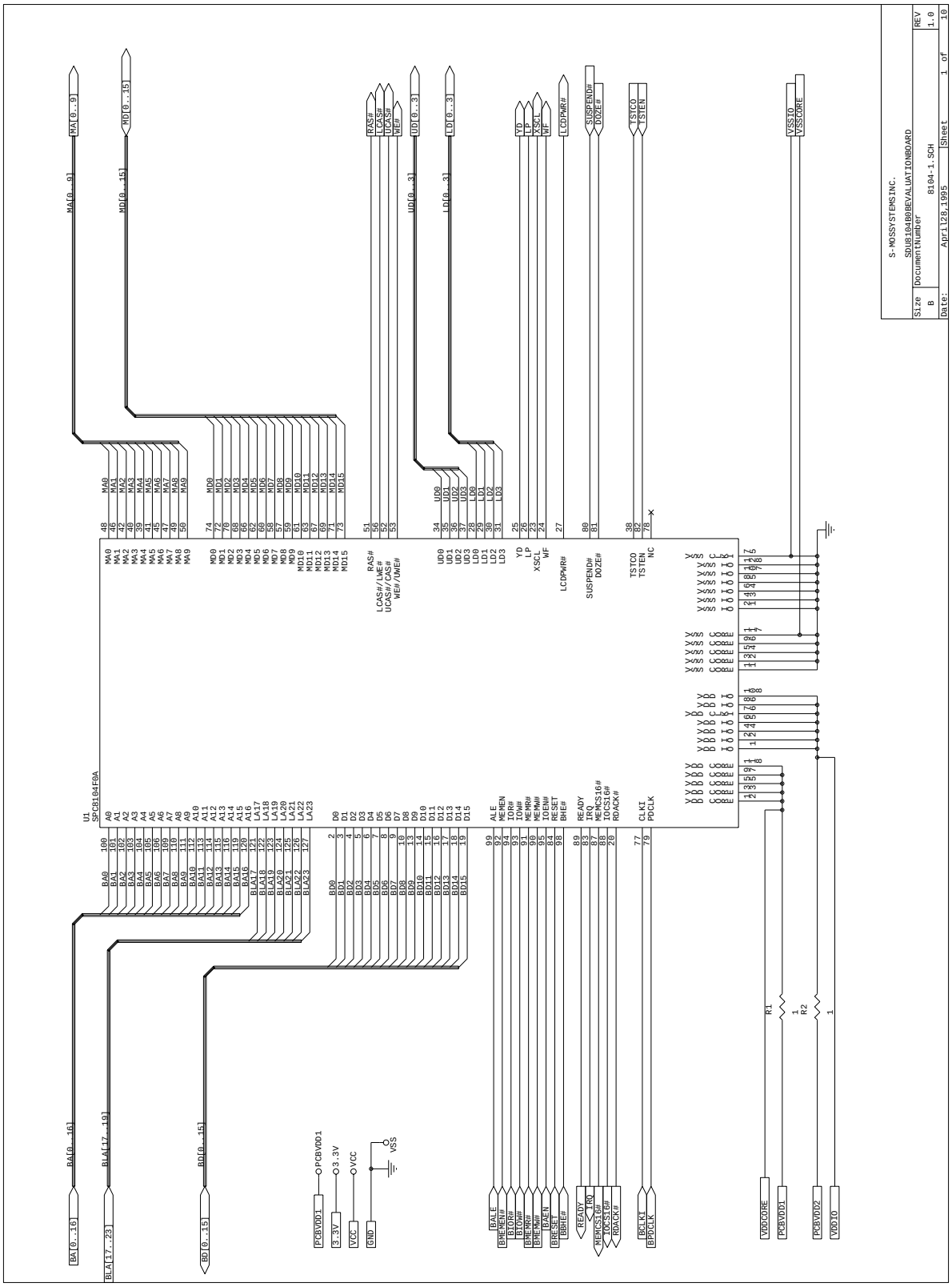


Figure 1 : SDU8104B0B Rev. 1.0 Schematic Diagram (1 of 10)

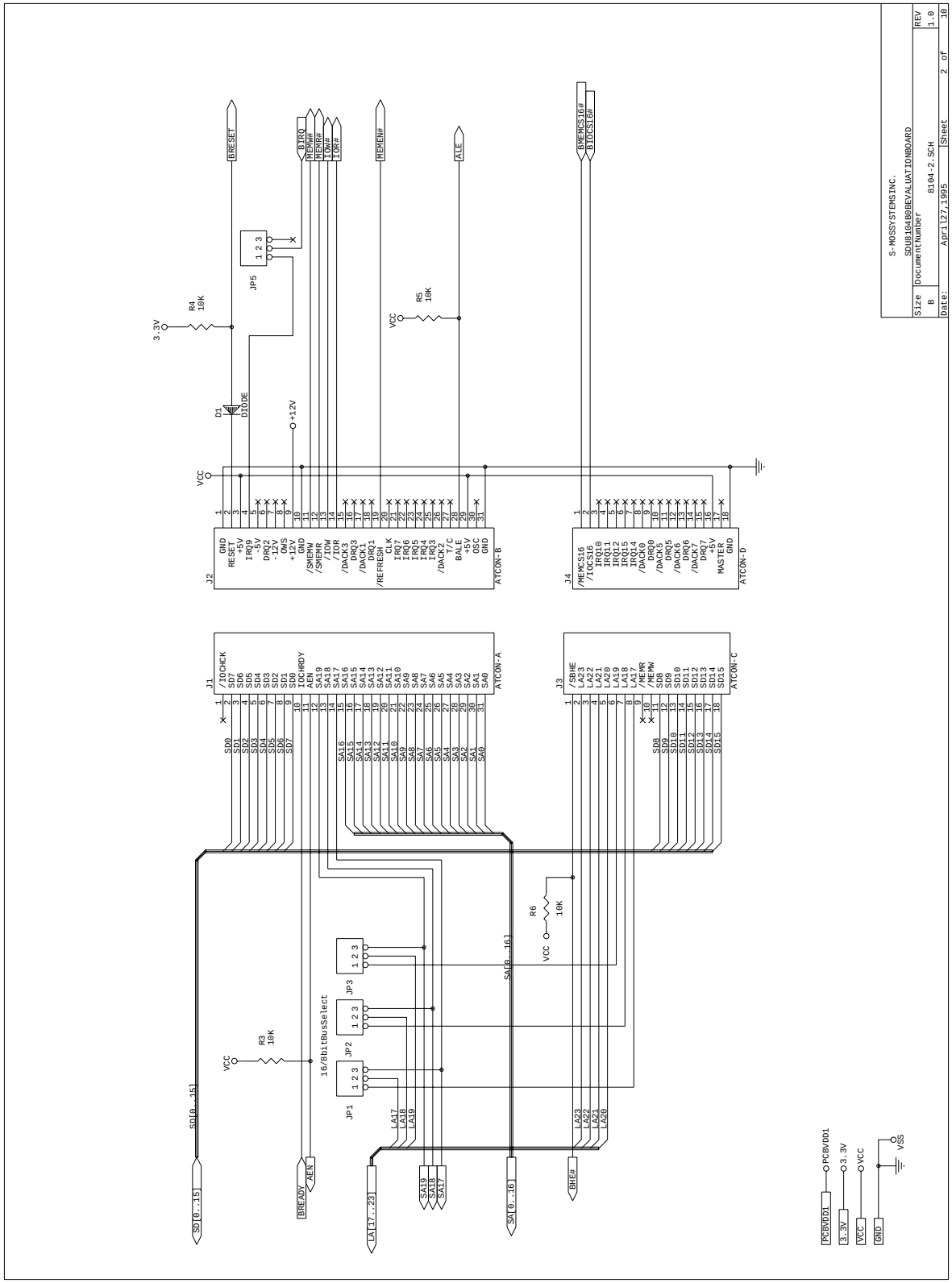


Figure 2 : SDU8104B0B Rev. 1.0 Schematic Diagram (2 of 10)

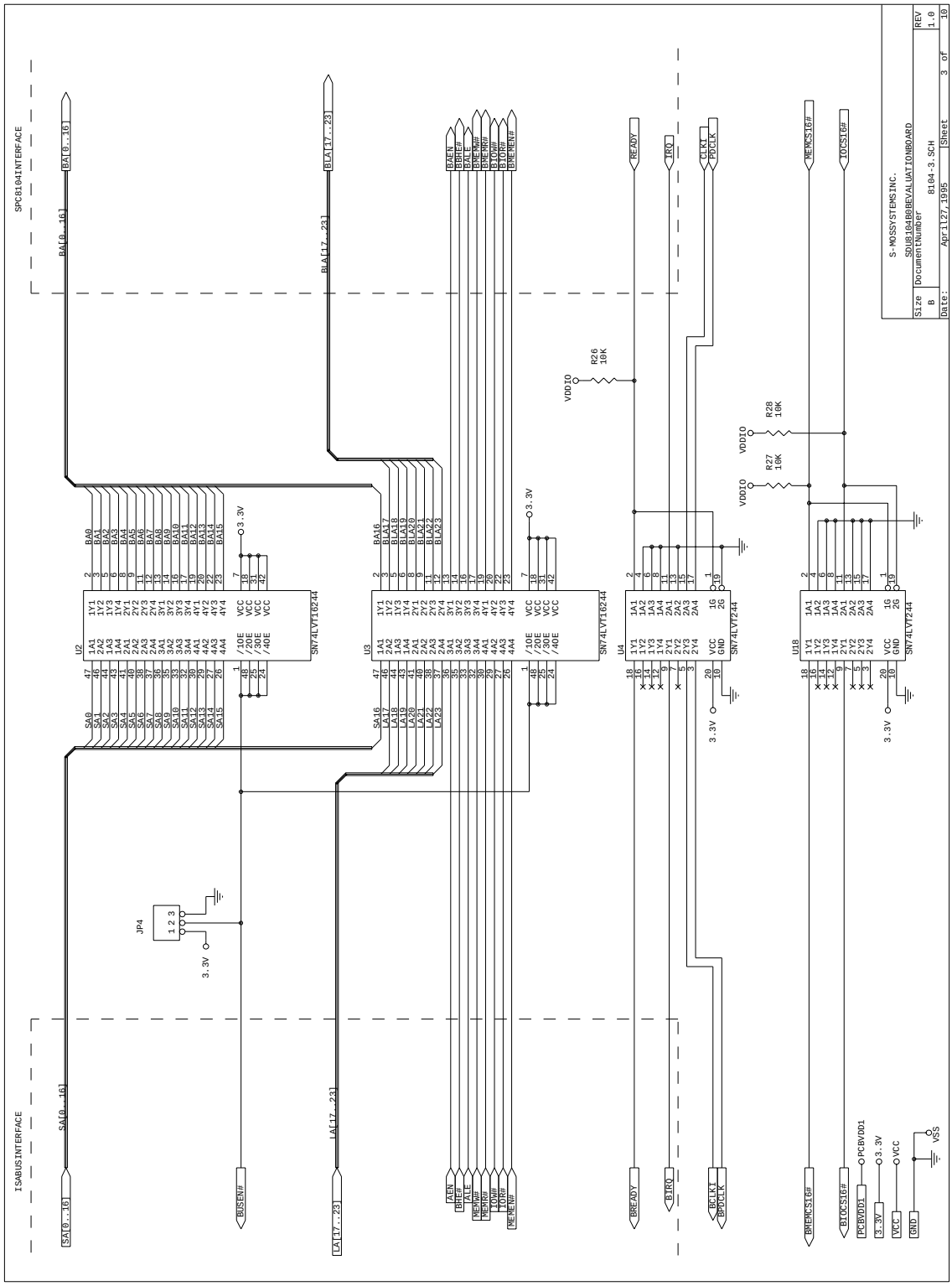


Figure 3 : SDU8104B0B Rev. 1.0 Schematic Diagram (3 of 10)

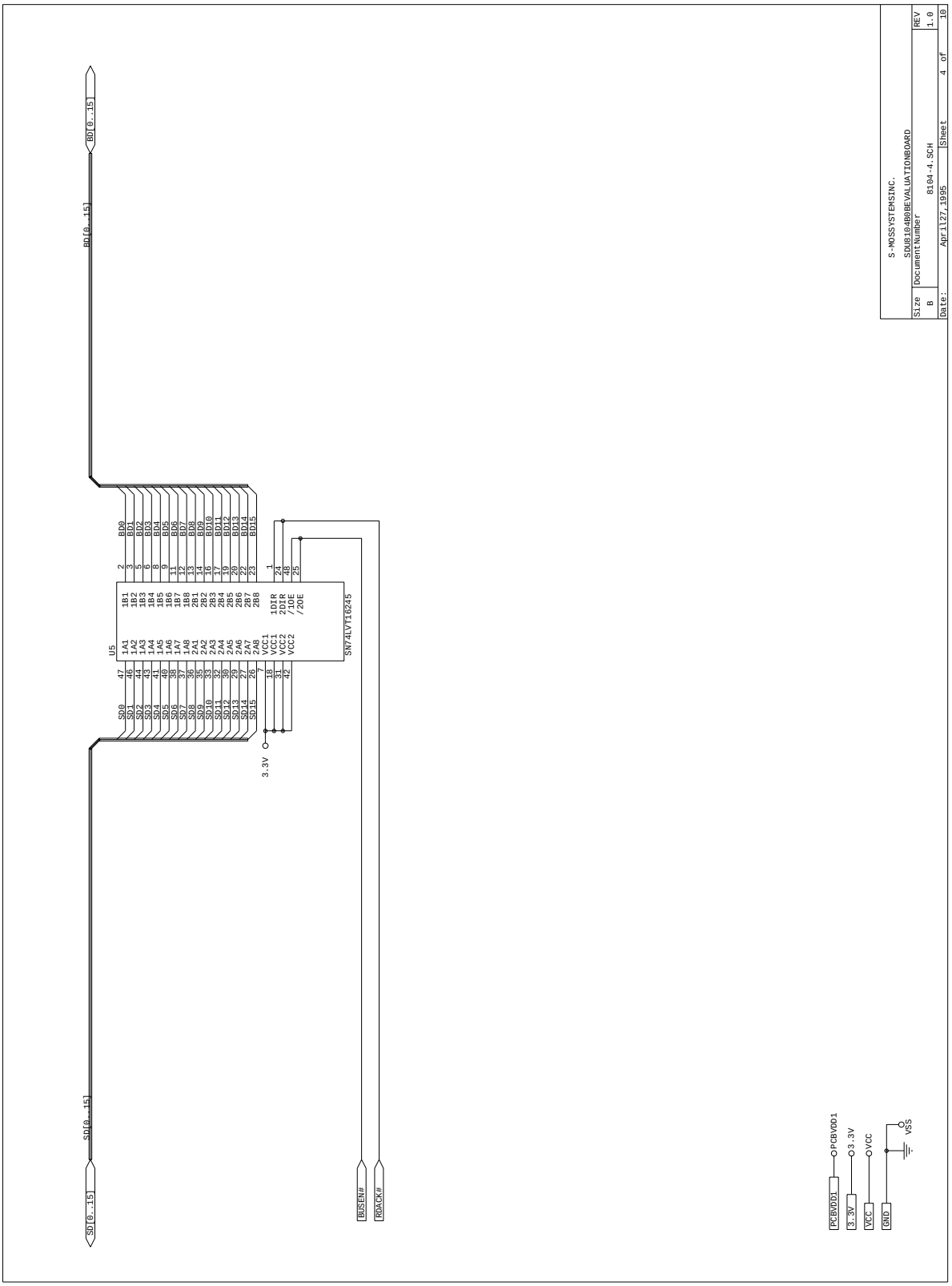


Figure 4 : SDU8104B0B Rev. 1.0 Schematic Diagram (4 of 10)

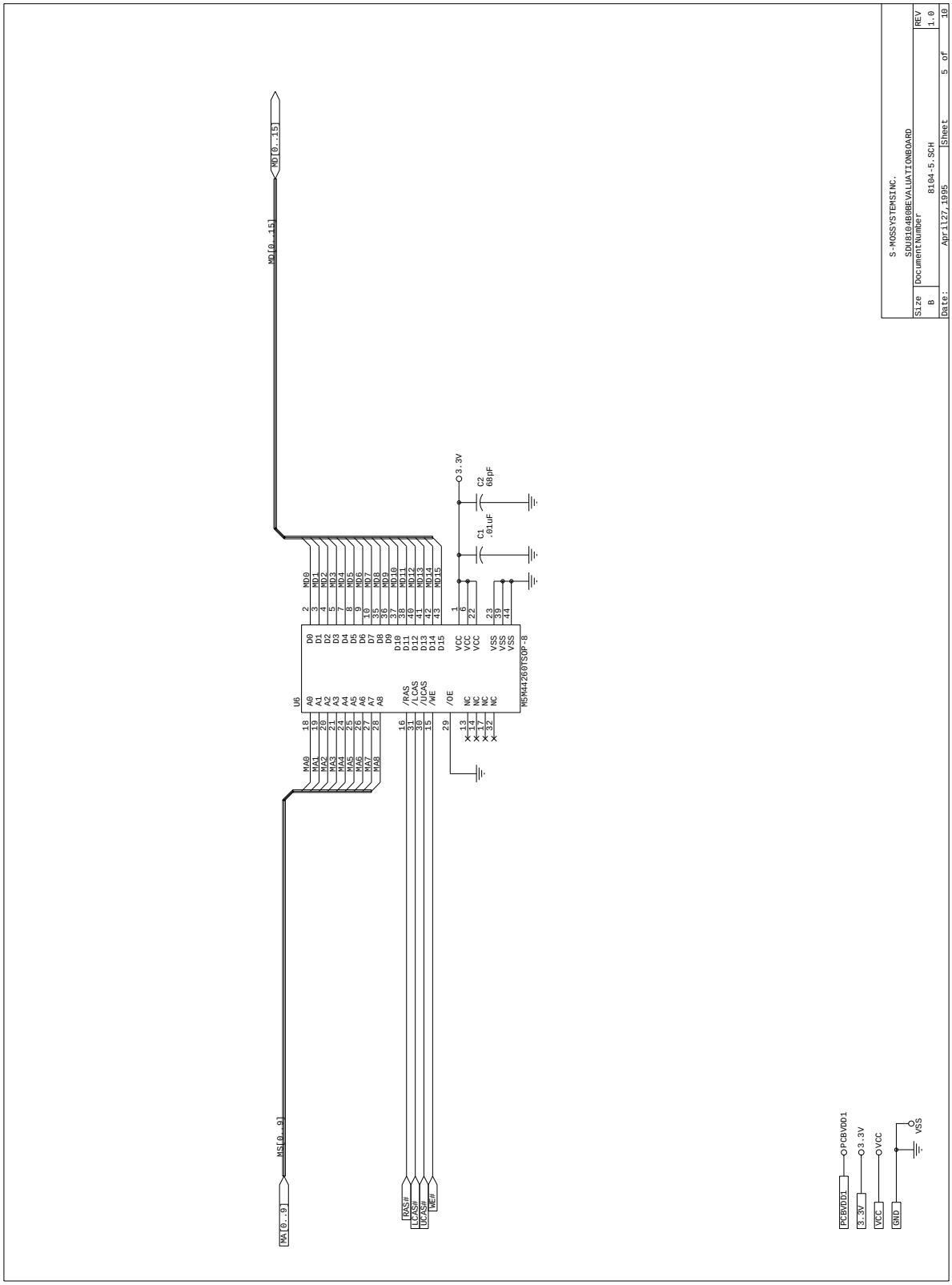


Figure 5 : SDU8104B0B Rev. 1.0 Schematic Diagram (5 of 10)

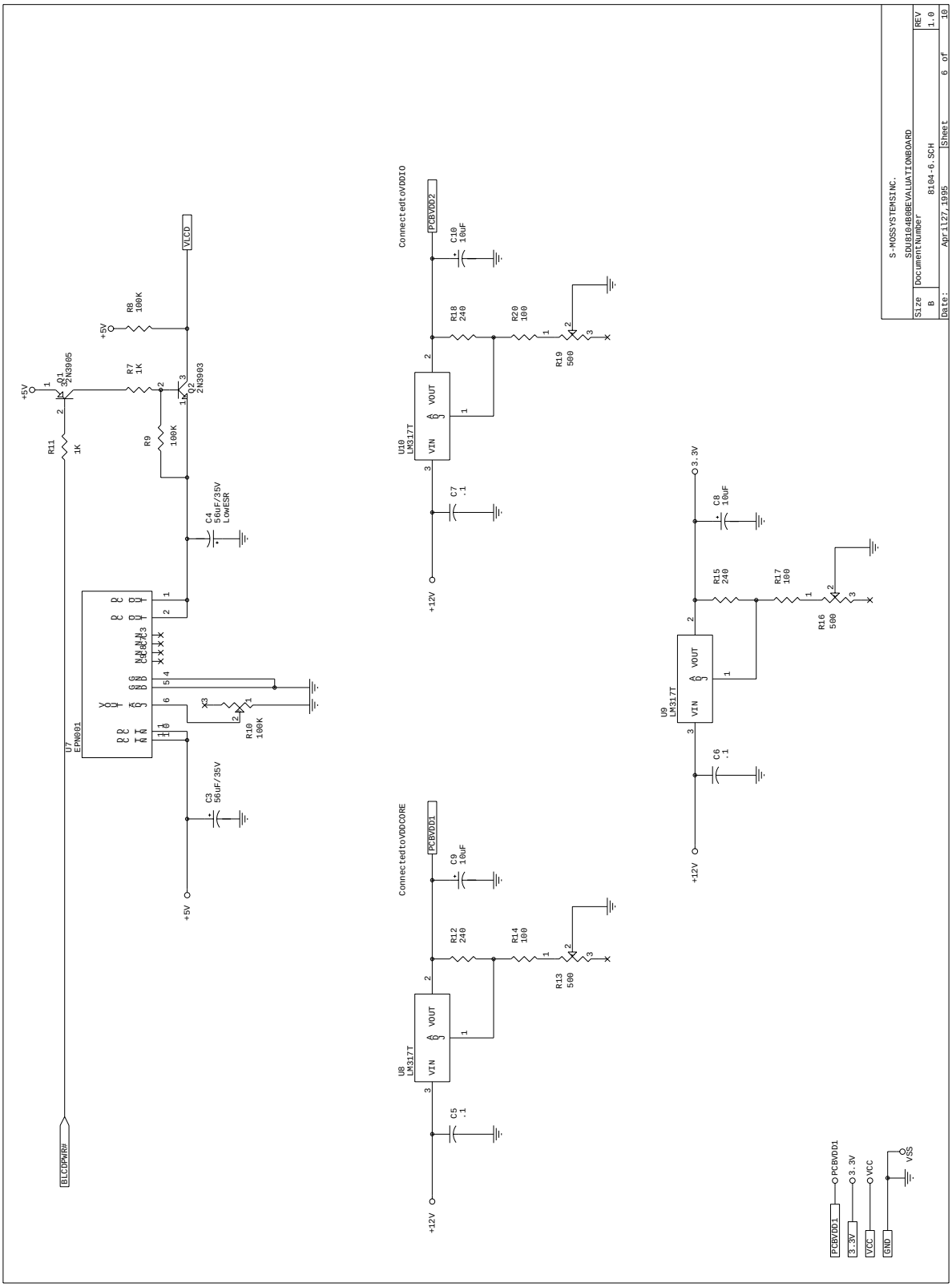


Figure 6 : SDU8104B0B Rev. 1.0 Schematic Diagram (6 of 10)

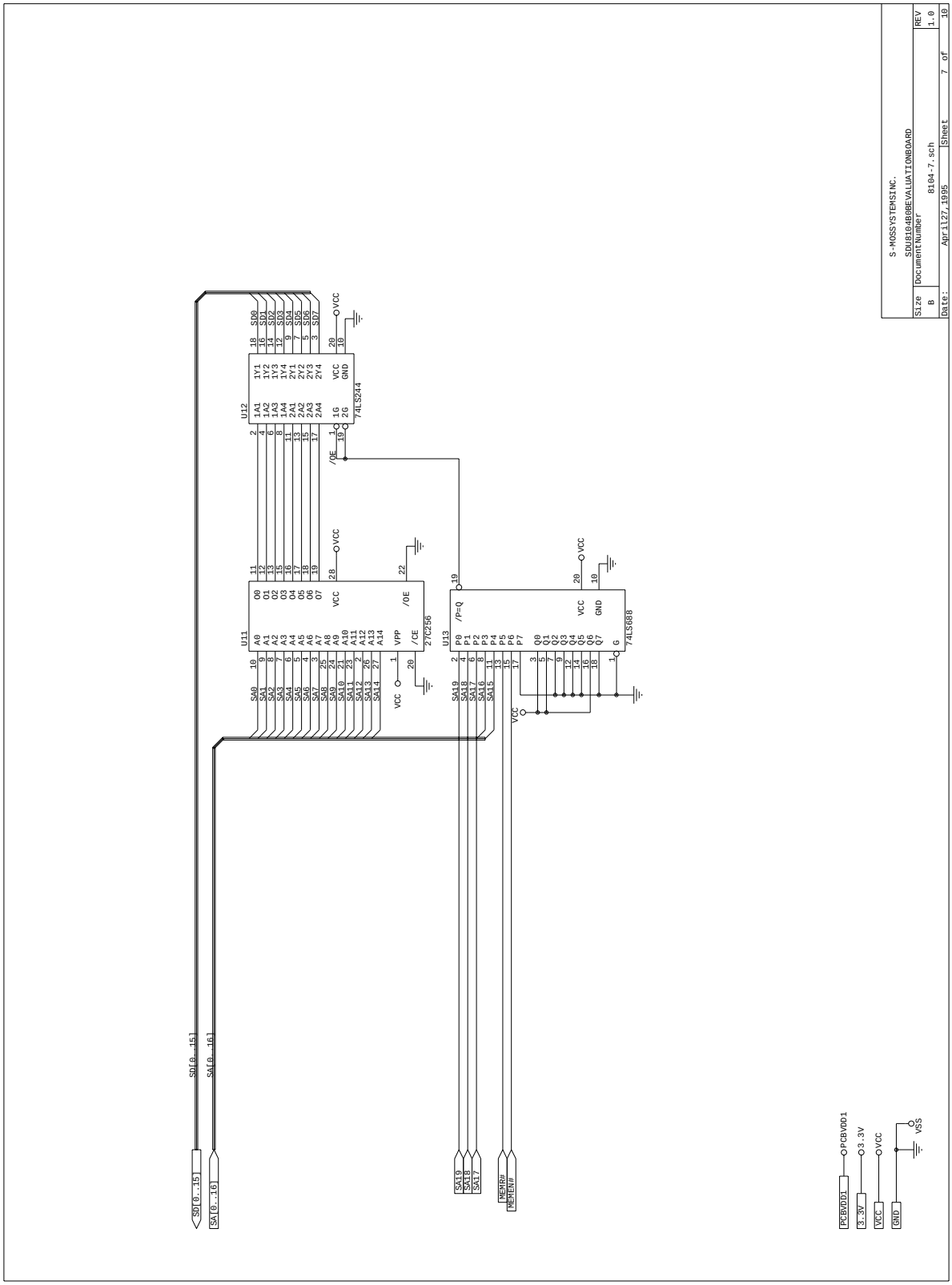


Figure 7 : SDU8104B0B Rev. 1.0 Schematic Diagram (7 of 10)

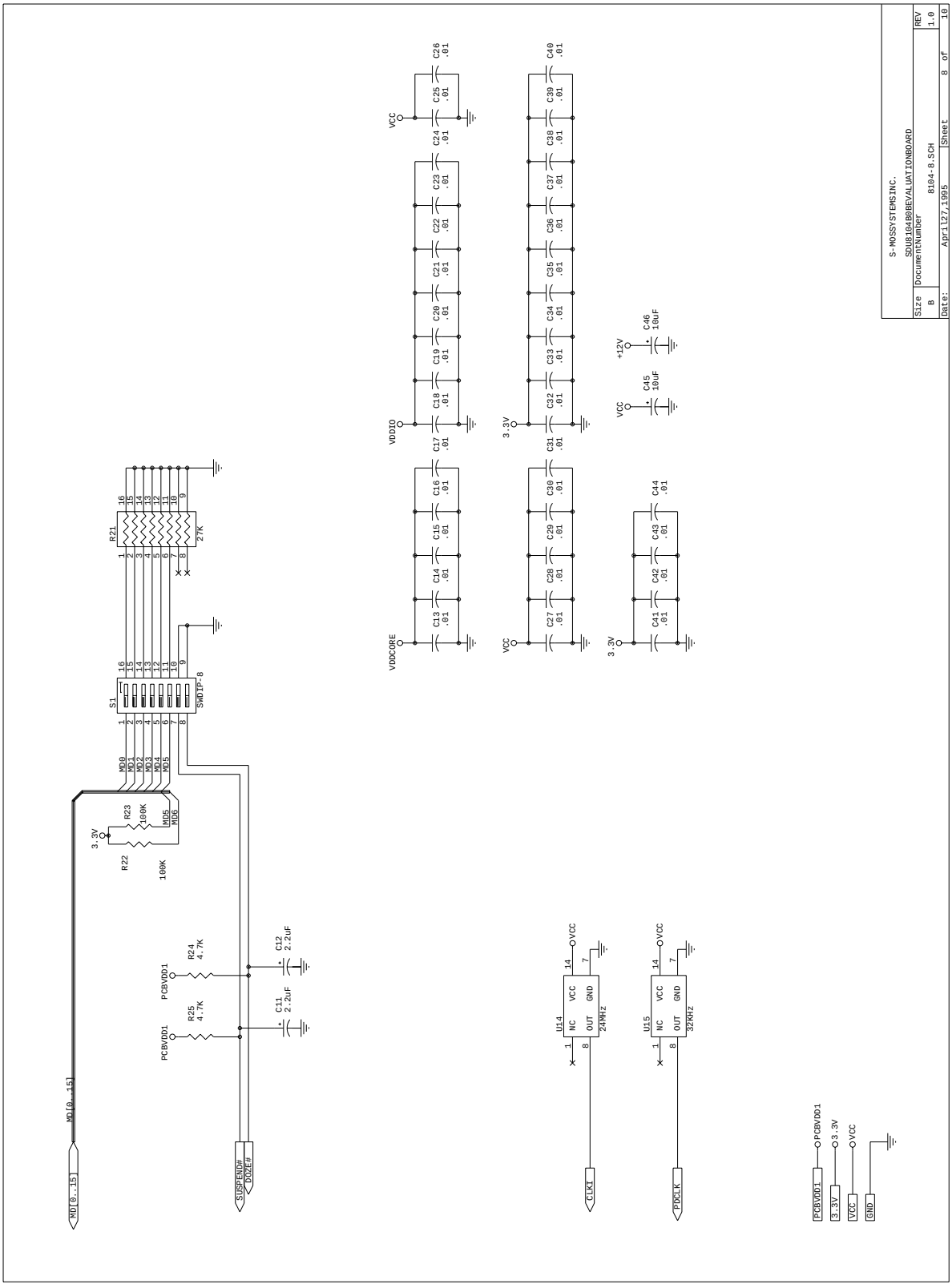


Figure 8 : SDU8104B0B Rev. 1.0 Schematic Diagram (8 of 10)

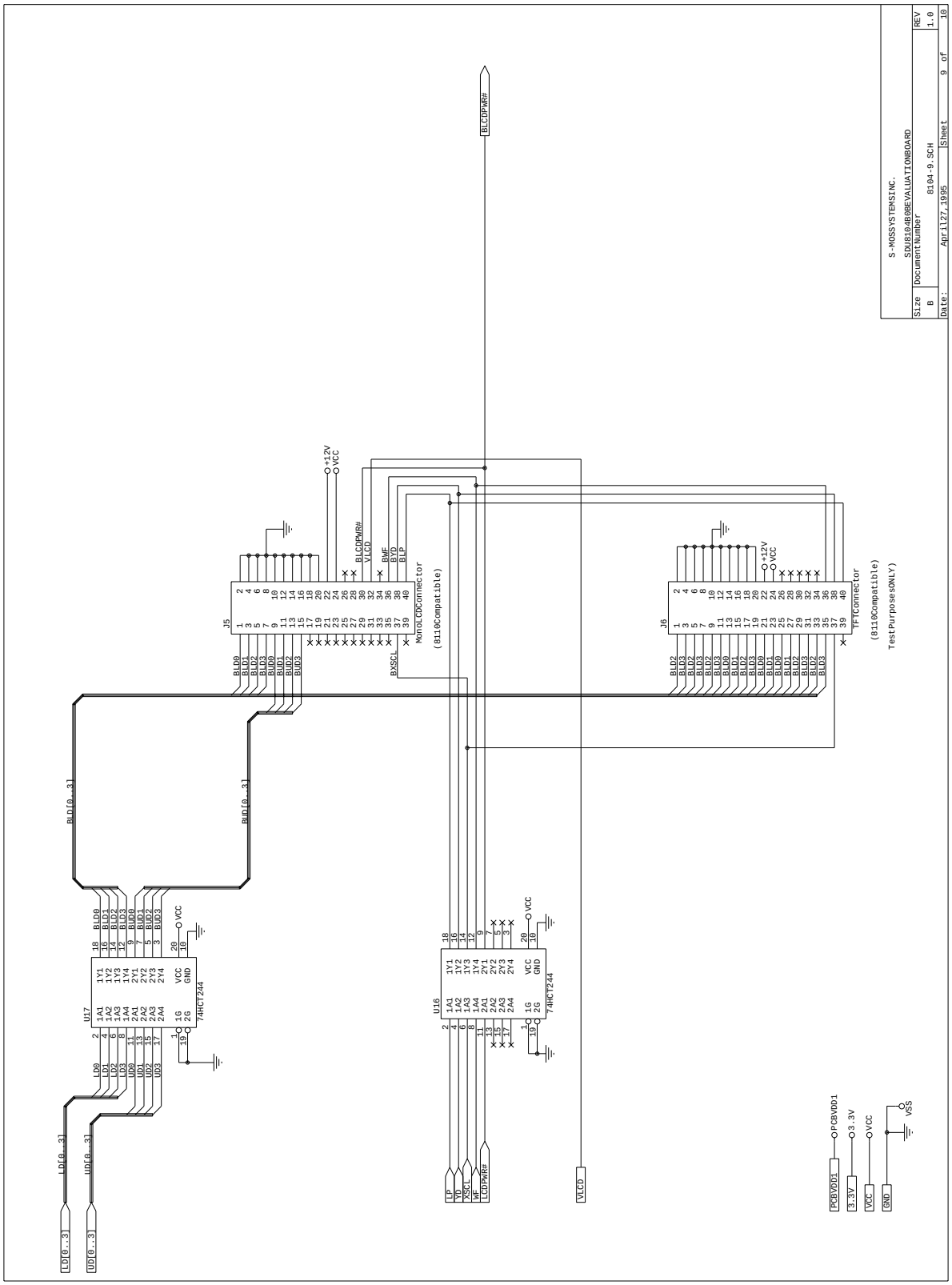


Figure 9 : SDU8104B0B Rev. 1.0 Schematic Diagram (9 of 10)

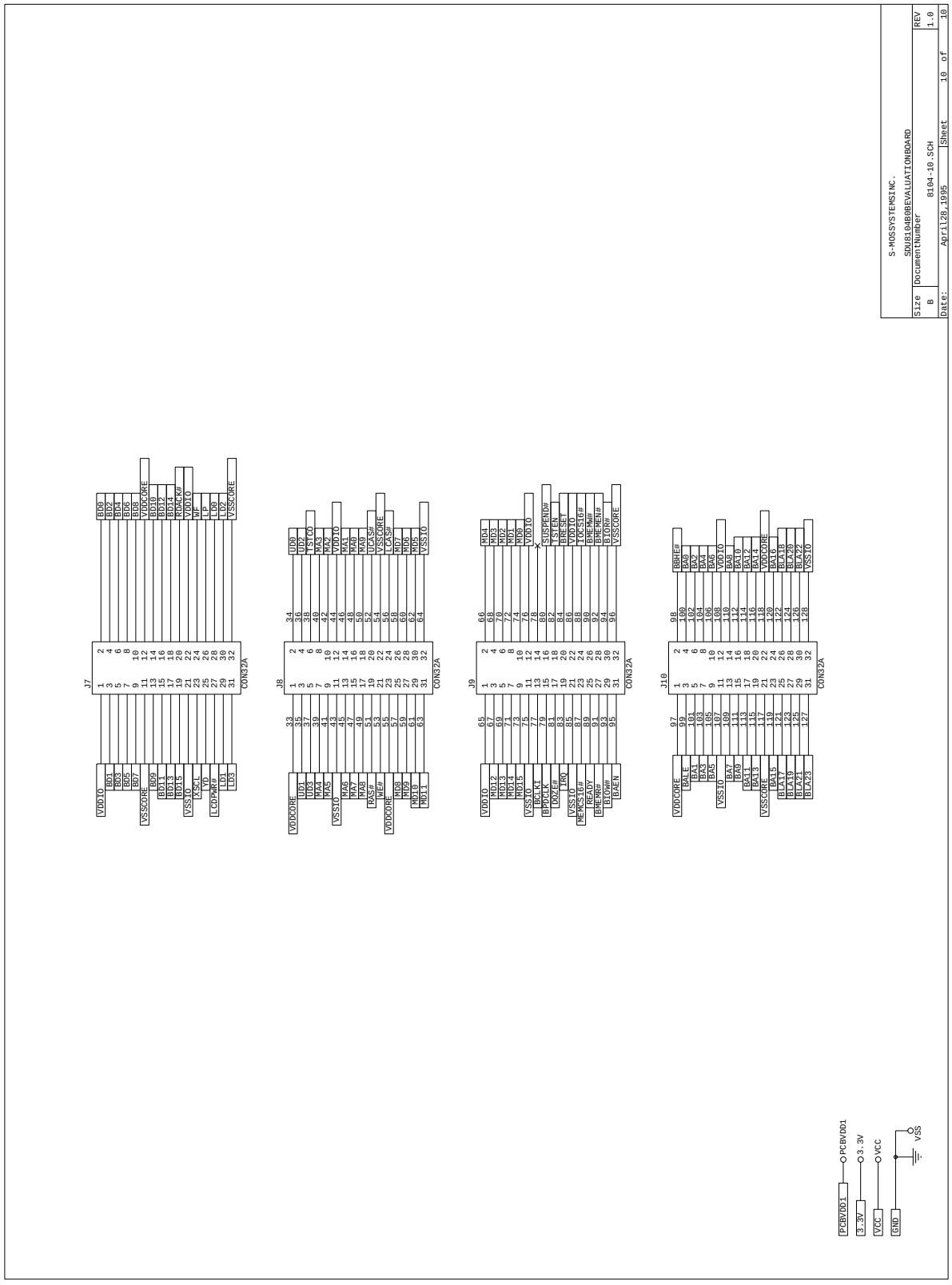


Figure 10 : SDU8104B0B Rev. 1.0 Schematic Diagram (10 of 10)

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

SDU8104B0C Rev 1.1 ISA Bus Evaluation Board

Drawing Office No. X15-AN-004-02.1

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY BLANK

1.0 SDU8104B0C REV. 1.1 EVALUATION BOARD

This manual reflects the use of the SDU8104B0C Rev. 1.1 evaluation board in conjunction with the SPC8104F0A Controller while operating in a ISA-Bus environment. The SDU8104B0C Rev. 1.1 will operate as a stand-alone video adapter card with on-board video BIOS support. Refer to the SPC8104F0A Functional Specification, Drawing Office No. X15-SP-001-xx for further details.

1.1 Features

- 128 pin QFP15-128 package
- mixed SMD / PTH devices
- Monochrome STN LCD support
- MIM support
- 16-bit ISA-Bus support
- On-board video BIOS with additional support logic provided
- External oscillator support
- 3.3V, 512K DRAM support (256K x 16, symmetrical, NEC upd424260LG5-A70 TSOP)
- Configuration Options via DIP-Switch
- Support for Software Power Save Modes
- On-board adjustable LCD BIAS negative power supply
- On-board adjustable LCD BIAS positive power supply
- Fixed 3.3V Power Supply
- Voltage translation circuitry for ISA-Bus signals
- Voltage translation circuitry for panel interface signals
- LCDBIAS Control (LCDPWR#)

1.2 Installation and Configuration

The SDU8104B0C Rev. 1.0 directly supports the 16-bit ISA-Bus architecture. The SPC8104F0A has 6 configurable inputs (MD[6:5], MD[3:0]) which are read on power-up. For the purpose of this design, MD[6:5] are factory set and therefore not configurable.

Configuration DIP Switch Settings

Switch	Signal	Closed	Open
SW1-1	MD0	LCD Panel Config bit 0	see table below
SW1-2	MD1	LCD Panel Config bit 1	see table below
SW1-3	MD2	LCD Panel Config bit 2	see table below
SW1-4	MD3	LCD Panel Config bit 3	see table below

The polarity of the Dip-Switch is defined as (Closed = ’0’ or ’low’).

External 100 K ohm pull-ups have been added to MD[6:5].
MD6 = ’high’ = symmetrical DRAM support
MD5 = ’high’ = LCD Signal state = ’low’ in Power Save Modes

LCD Panel Configuration

The BIOS configuration for different LCD panels is handled through the state of the MD lines during RESET. The following tables describes the settings and the associated panel type.

Pin Name	Configuration Pin Functionality
MD0	B0 of panel table
MD1	B1 of panel table
MD2	B2 of panel table
MD3	B3 of panel table

bit 3	bit 2	bit 1	bit 0	Resolution	Mode	Bits	Comments
1	1	1	1	640x480	MIM	4	MIM
1	1	1	0	640x480	Single	8	Normal Video
1	1	0	1	640X480	Single	8	Text/Graphics Vertical Expansion enabled
1	1	0	0	640x480	Single	8	Reverse Video (text/graphics)
1	0	1	1	640x480	Dual	8	Normal Video
1	0	1	0	640x480	Dual	8	Text/Graphics Vertical Expansion enabled
1	0	0	1	640x480	Dual	8	Reverse Video (text/graphics)
1	0	0	0	640x200	Single	4	Normal Video
0	1	1	1	640x200	Single	4	Reverse Video
0	1	1	0	320x240	Single	4	Normal Video
0	1	0	1	320x240	Single	4	Reverse Video
0	1	0	0	320x240	Single	4	Normal Video, Slow Blink
0	0	1	1	OEM			
0	0	1	0	OEM			
0	0	0	1	OEM			
0	0	0	0	OEM			

The above pre-set options are intended to show an example of programmable panel features. OEM Configuration options allows for very specific resolution/timing criteria to be programmed for a given panel. Refer to SPC8104F0A LCD Panel Configuration Options, Drawing Office No. X15-AN-xxx-xx for further details. All panel features can be configured into the BIOS using the SMOS utility program 8104cfg.exe.

LCD Signal Connector J5 Pinout

LCD Mono Connector J1 Pin No.	SPC8104 Pin Name	Mono STN LCD		MIM
		8-bit	4-bit	4-bit
1	LD0	LD0		D0
3	LD1	LD1		D1
5	LD2	LD2		D2
7	LD3	LD3		D3
9	UD0	UD0	UD0	
11	UD1	UD1	UD1	
13	UD2	UD2	UD2	
15	UD3	UD3	UD3	
17				
19				
21				
23				
25				
27				
29				
31				
33				
35				
37	XSCL	XSCL	XSCL	PCLK
39				
40	LP	LP	LP	HSYNC
38	YD	YD	YD	VSYNC
36	WF	WF	WF	DE
34		VDDH	VDDH	VDDH
32		VLCD	VLCD	VLCD
30	LCDPWR#			
28				
26				
24		+5 V	+5 V	+5 V
22		+12 V	+12 V	+12 V
20-2		GRND	GRND	GND

1.3 Technical Description

16-Bit ISA-Bus Support

This board directly supports the 16-bit ISA-Bus and is intended to be used as a ISA-Bus Target add-in-board. All SPC8104F0A related electrical connections come from the standard in-line ISA-Bus connectors. Note that all Bus related signals are separated from the ISA-Bus and the SPC8104F0A by voltage translations circuits. The nature of this design is such that the bus interface I/O pins of the SPC8104F0A are 3.3V only.

512K DRAM Support

A one chip 3.3V, 512K solution is supported using a 256K x 16 DRAM. Symmetrical DRAM in an TSOP package is supported. As the specific DRAM used are 3.3V, there are no voltage translators between the SPC8104F0A and the memory.

Monochrome LCD Support

The SPC8104F0A 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 (J5) and pass through a 3V to 5V voltage translation circuitry. The interface signals are alternated with grounds on the cable to reduce cross talk and noise related problems.

Refer to *LCD Panel Configuration and LCD Signal Connector (J5) Pinout* above for your specific settings.

MIM Support

The SPC8104F0A also supports 4-bit MIM panels. All the necessary signals are provided on the 40-pin ribbon cable header (J5) and pass through a 3V to 5V voltage translation circuitry. The interface signals are alternated with grounds on the cable to reduce cross talk and noise related problems.

Refer to *LCD Panel Configuration and LCD Signal Connector (J5) Pinout* above for your specific settings

VGA BIOS Support

A 32K EPROM contains the VGA BIOS. Support logic consists of an Address Decoder (C000H-C7FFH), and a Data Buffer operating from the ISA-Bus. With the BIOS installed the board will operate as a stand-alone Video adapter.

Power Save Modes

The SPC8104F0A supports three Power Save Modes that can be activated via either hardware or software. A utility program is supplied to control these modes. Software control is performed by directly writing the SPC8104F0A associated internal registers. The hardware power save modes are controlled by input pins on the SPC8104F0A, however for the purpose of this design the associated inputs are tied to their inactive states and are not used.

1.4 Adjustable LCD BIAS Positive Power Supply

Depending on the specific LCD panel, it may require a positive power supply to provide between +23V and +40V (Iout=45 mA) for use as the LCD BIAS voltage. For ease of implementation, such a power supply has been provided as an integral part of this design The VDDH signal can be found on LCD connector J5 at pin 34. Output voltage is varied by potentiometer R?. To prevent panel damage, VDDH power supply sequencing is implemented by enabling the VDDH power supply only when the SPC8104F0A signal LCDPWR# is at a low logic level.

1.5 Adjustable LCD BIAS Negative Power Supply

Depending on the specific LCD panel, it may require a negative power supply to provide between -18V and -23V (Iout=45 mA) for use as its LCD BIAS voltage. For ease of implementation, such a power supply has been provided as an integral part of this design. The VLCD signal can be found on LCD connector J5 at pin 32. Output voltage is varied by potentiometer R?. To prevent panel damage, VLCD power supply sequencing is implemented by enabling the VLCD power supply only when the SPC8104F0A signal LCDPWR# is at a low logic level.

External Oscillator Support

A 3.3V, 24.0MHz oscillator is provided on-board.

Fixed 3.3V Power Supply

There is a fixed 3.3V power supply on-board used by the SPC8104, DRAM and all associated voltage translator circuitry.

Voltage Translation Circuitry for ISA-Bus Interface

Voltage translation circuitry is required on all ISA-Bus interface signals to prevent the forward biasing of the input protection diodes of the SPC8104F0A bi-directional control of the data bus is controlled by the output signal RDACK#. The LCD interface also has translation circuits as most panels require a 5.0V interface.

PCB Layout Considerations

This will be a 4 layer PCB with its physical dimensions defined by ISA-Bus and component size restrictions. All appropriate components will be surface-mount in an effort to reduce layout size. All end-user controls (i.e., adjustable potentiometers, Dip-Switches, LCD Connector, etc.) will be located on the top edge on the PCB allowing for ease of use.

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	28	C1,C14-40	0.01uF	1206 pckg
2	1	C2	68pF	1206 pckg
3	3	C3,C4,C8	56uF/35V	LXF35VB56RM6X11LL
4	3	C5,C6,C7	10uF/63V	LXF63VB10RM5X11LL
5	1	C12	0.1uF	1206
6	5	C9,C10,C11	10uF / 16V	D-SIZE 10uF Tantalum
7	1	C13	33uF	D-Size 33uF/16v Tantalum
8	1	J5	CON40A	Shrouded Header 40 pin Dual-row center-key
9	1	L1	1uH	DELEVAN 1537-12
10	1	Q1	2N3905	PNP Signal Transistor, PTH T0-92
11	1	Q2	2N3903	NPN Signal Transistor, PTH TO-92
12	3	R1,R2,R3	10K	10K Ohm / 1206 pckg / 5%
13	3	R4,R5,R6	1.2K	1.2K Ohm / 1206 pckg 5%
14	2	R7,R11	1K	1K Ohm / 1206 pckg / 5%
15	4	R8,R9,R15, R16	100K	100K Ohm / 1206 pckg / 5%
16	1	R10	100K	100K Ohm Trim Pot (Bourns 3386 type)
17	1	R12	470K	470K Ohm / 1206 pckg / 5%
18	1	R13	200K	200K Ohm Trim Pot (Bourns 3386 type)
19	1	R14	14K	14K Ohm / 1206 pckg / 5%
20	4	R17,R18,R19, R20	15K	15K Ohm / 1206 pckg / 5%
21	1	S1	SW DIP-4	Switch Dip 4 position
22	1	U1	SPC8104F0A	QFP-15 pckg SMOS SUPPLIED
23	2	U2-3	SN74LVT16244	48 pin SSOP pckg
24	2	U4,U5	SN74LVT244	DW020 pckg
25	1	U6	SN74LVT16245	48pin SSOP pckg
26	1	U7	NEC upd424260LG5-A70	256Kx16 DRAM / 44 pin TSOP pckg
27	1	U8	EPN001	XENTECK - Negative Power Supply Module
28	1	U9	RD-0412	XENTECK - Positive Power Supply Module
29	1	U10	NM27C256Q-200	"EPROM, socket and component"
30	1	U11	74LS244	DWO20
31	1	U12	74LS688	DW020
32	1	U13	24.0MHz OSCILATOR	SMD - SMOS SUPPLIED
33	1	U14	National LP2960AIM-3.3	surface-mount package (custom size ?)
34	2	U15,U16	74HCT244	DW020

APPENDIX B SPC8110F0A REV. 1.1 SCHEMATICS

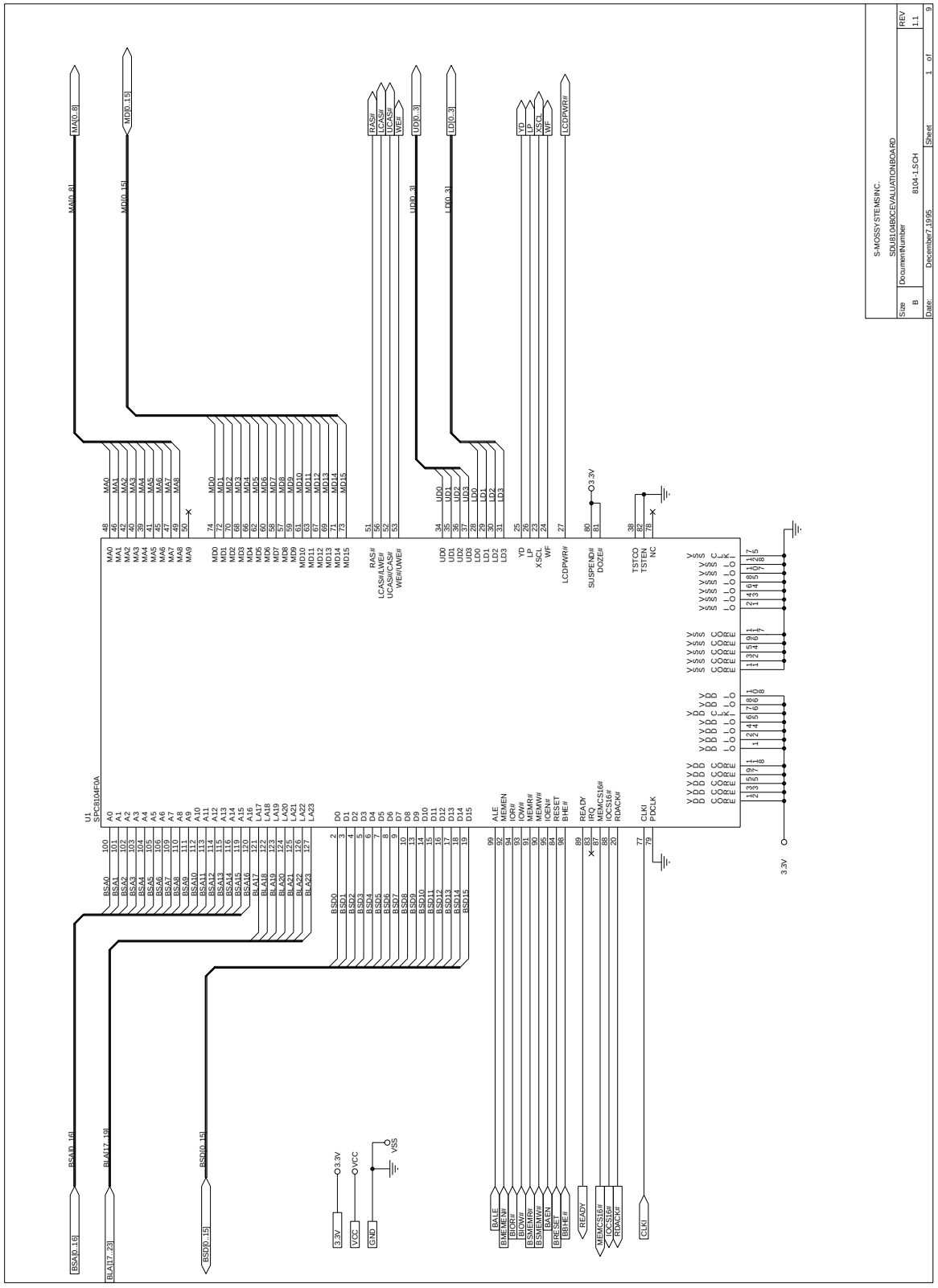


Figure 1 : SDU8104B0C Rev. 1.1 Schematics (1 of 9)

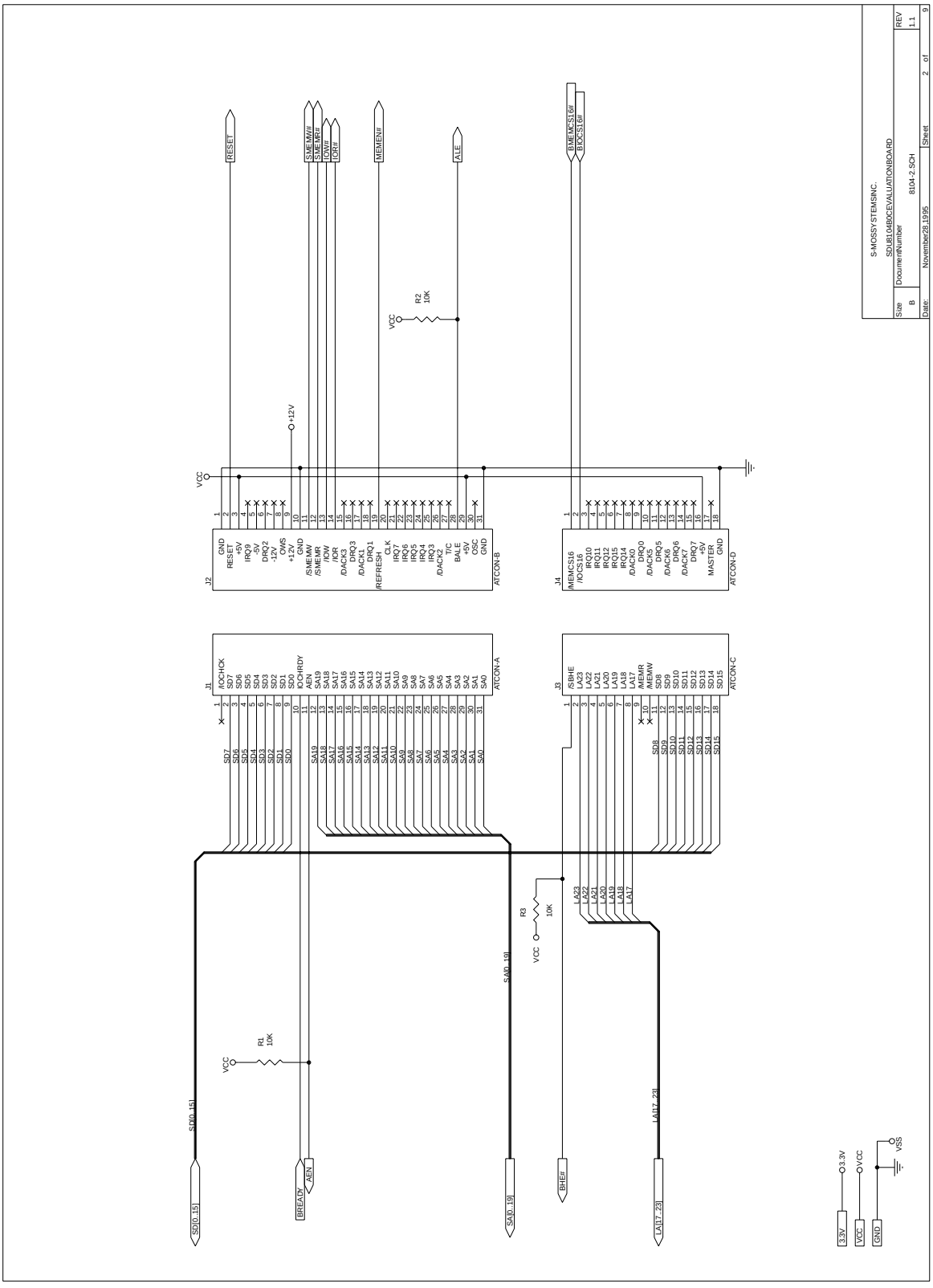


Figure 2 : SDU8104B0C Rev. 1.1 Schematics (2 of 9)

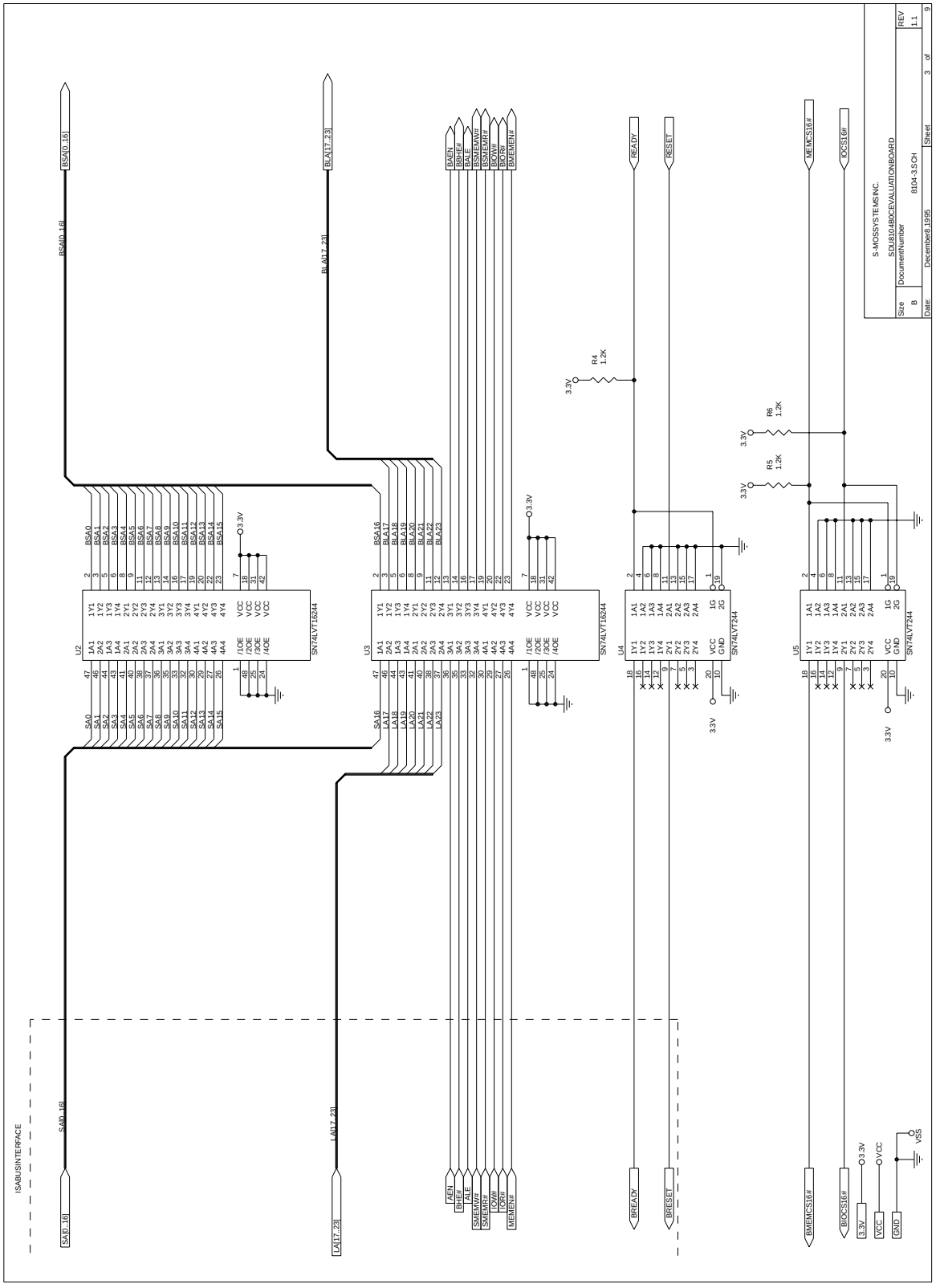


Figure 3 : SDU8104B0C Rev. 1.1 Schematics (3 of 9)

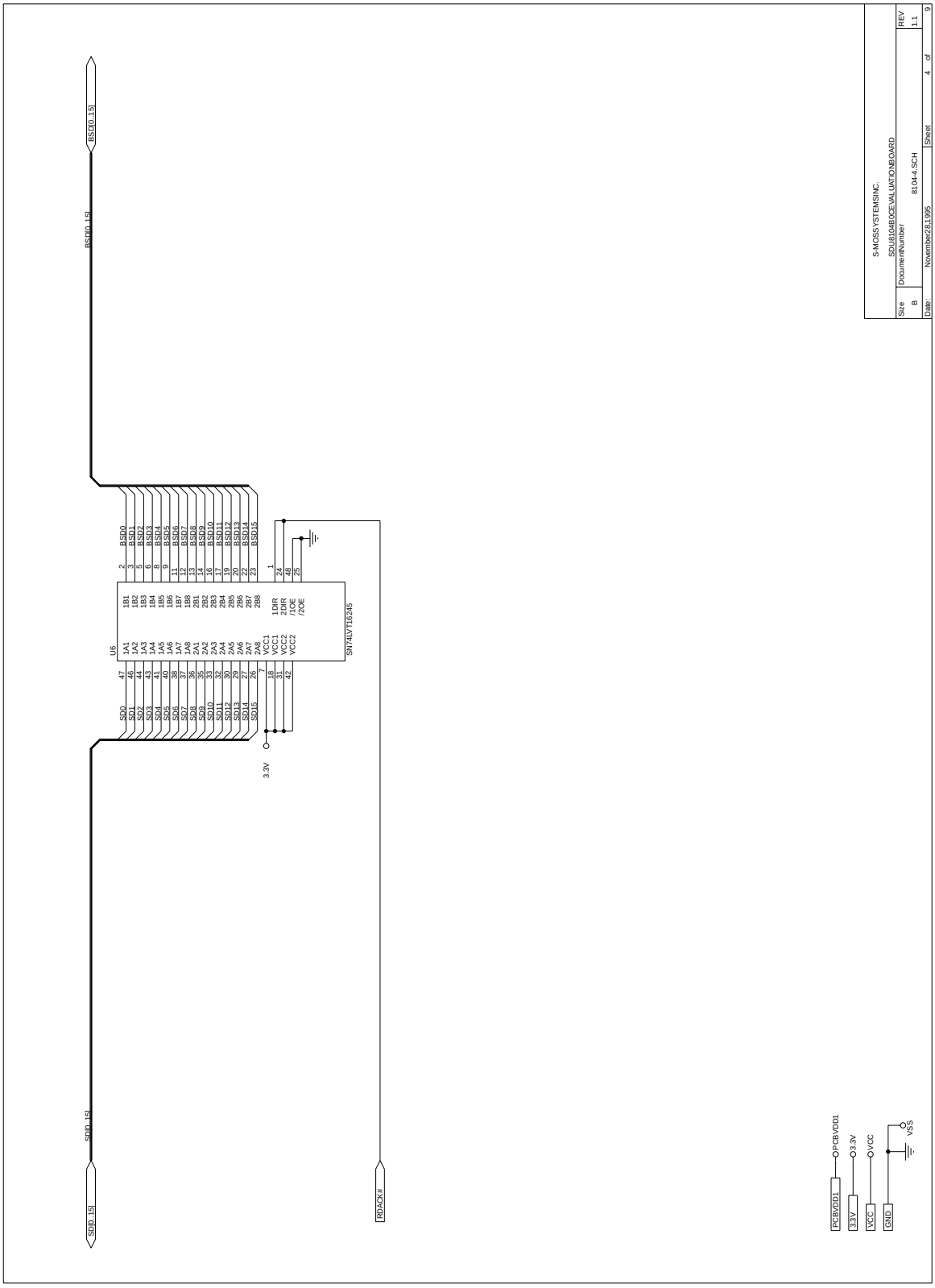


Figure 4 : SDU8104B0C Rev. 1.1 Schematics (4 of 9)

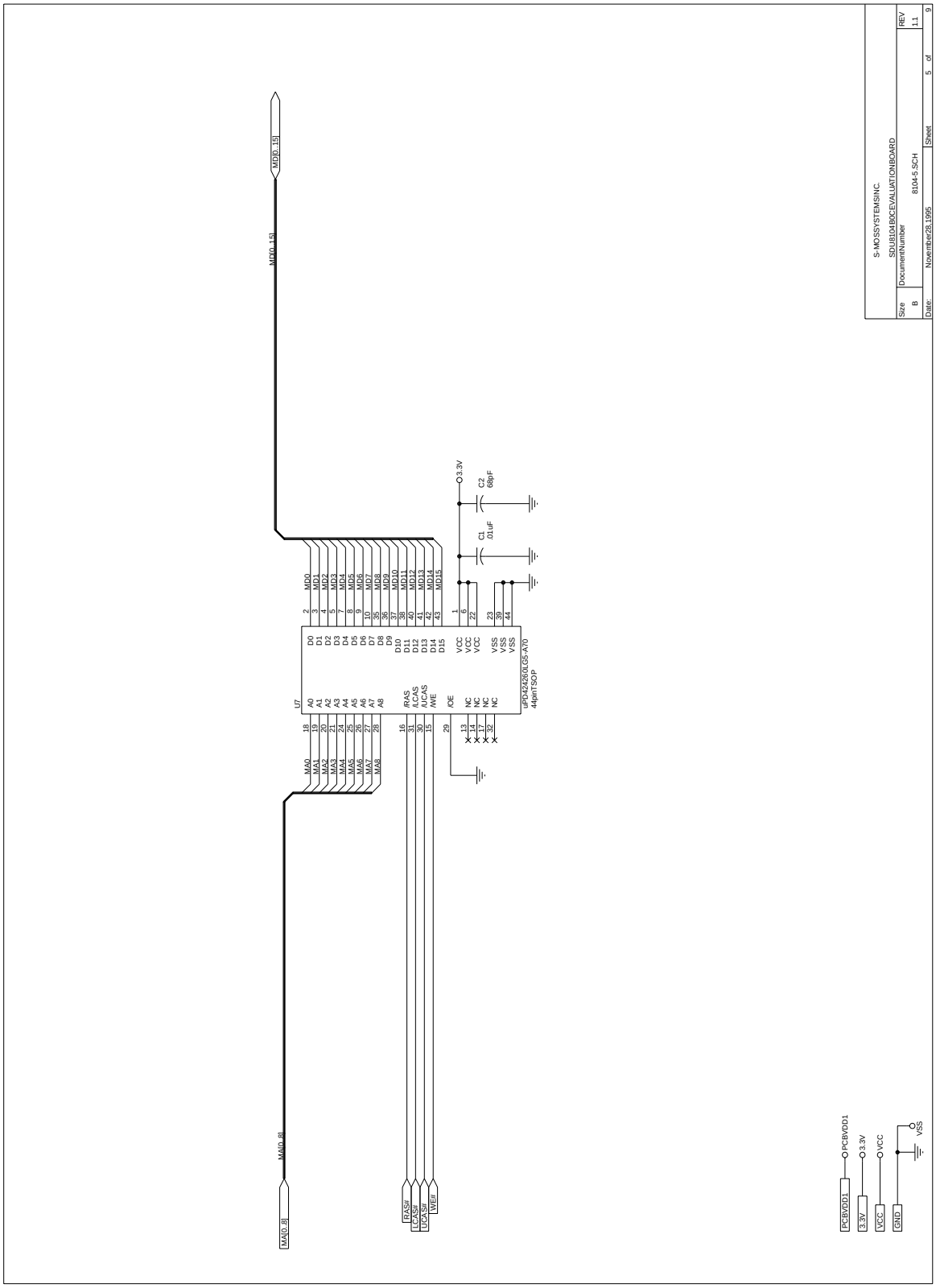


Figure 5 : SDU8104B0C Rev. 1.1 Schematics (5 of 9)

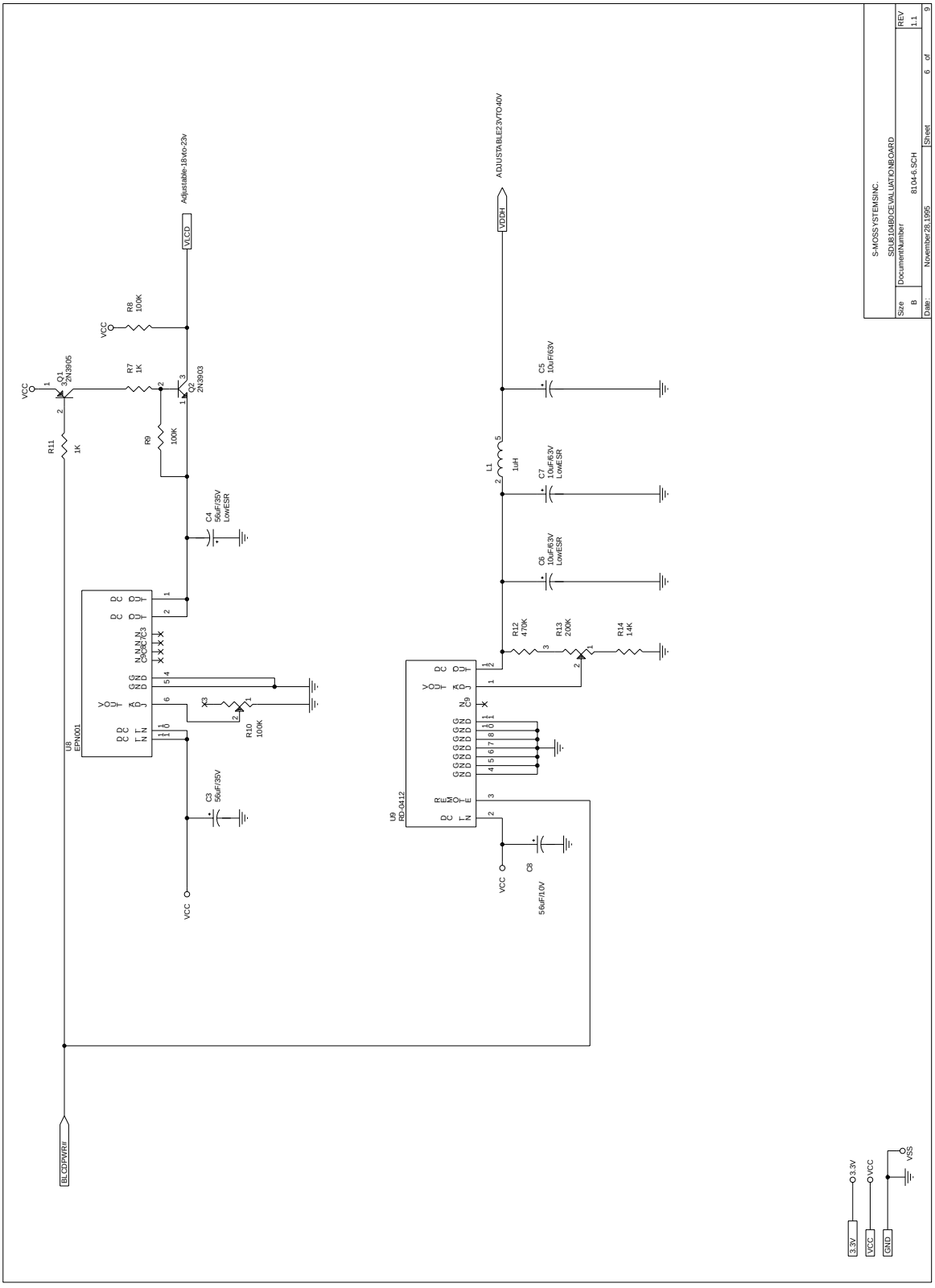


Figure 6 : SDU8104B0C Rev. 1.1 Schematics (6 of 9)

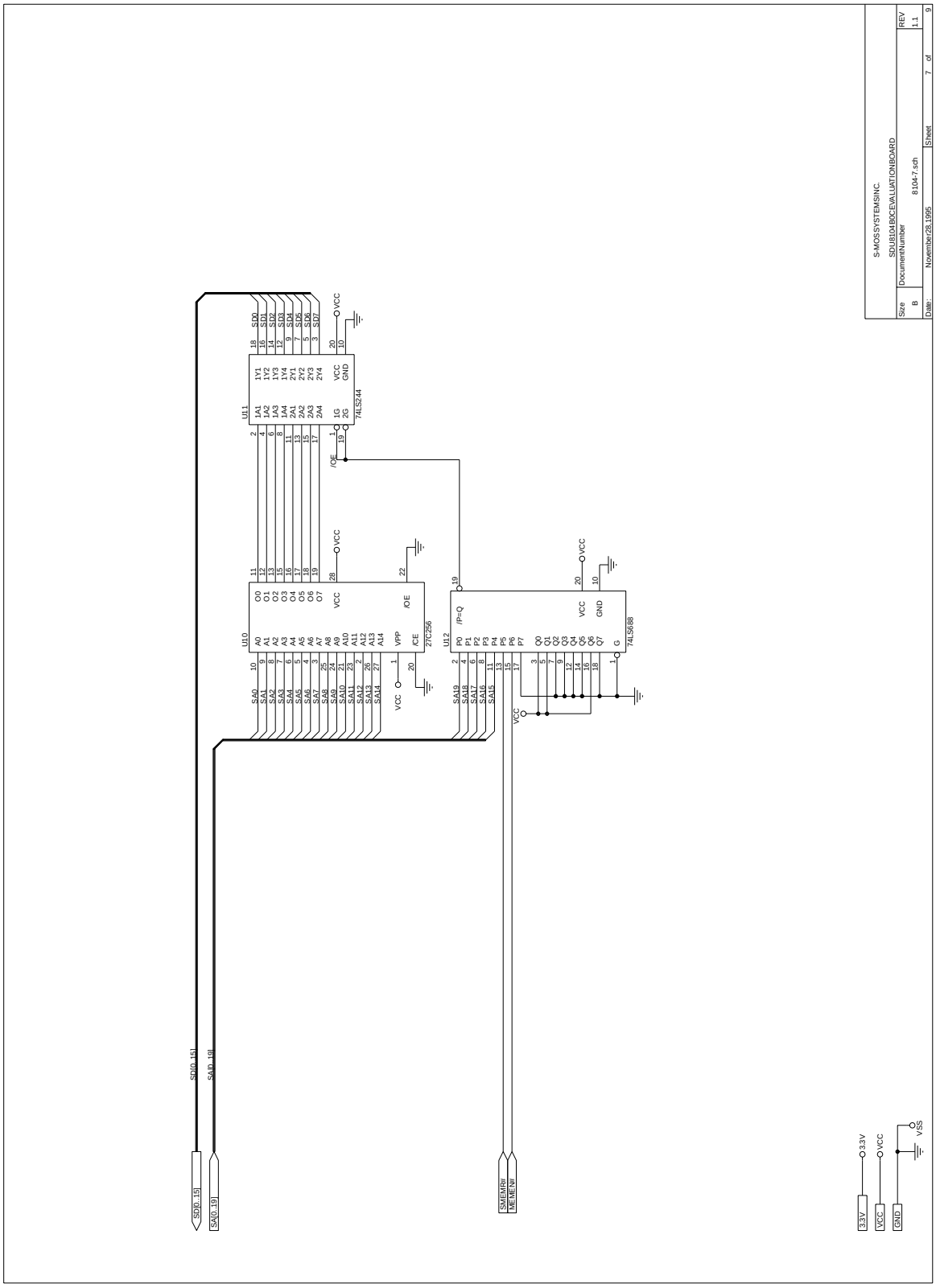


Figure 7 : SDU8104B0C Rev. 1.1 Schematics (7 of 9)

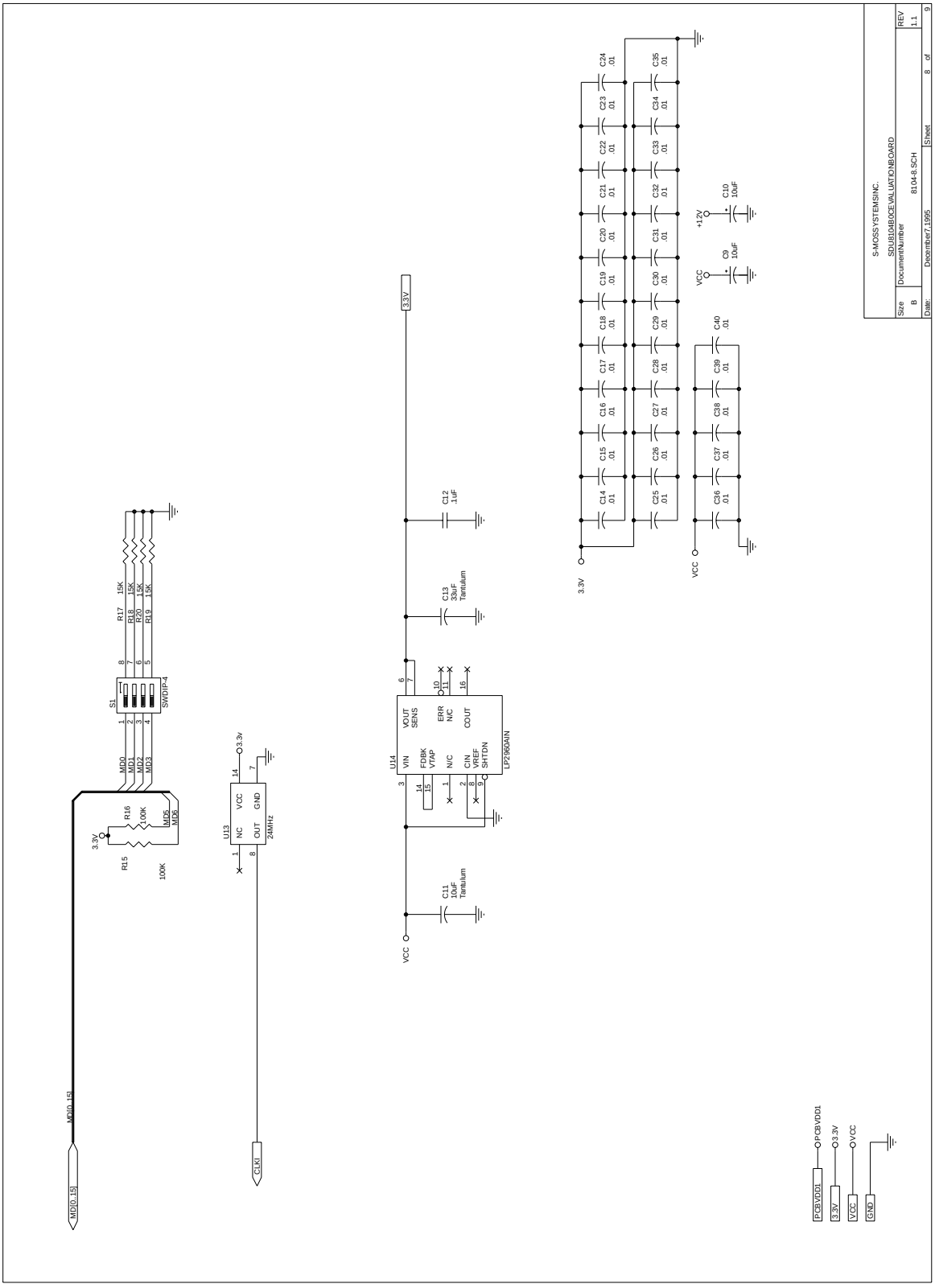


Figure 8 : SDU8104B0C Rev. 1.1 Schematics (8 of 9)

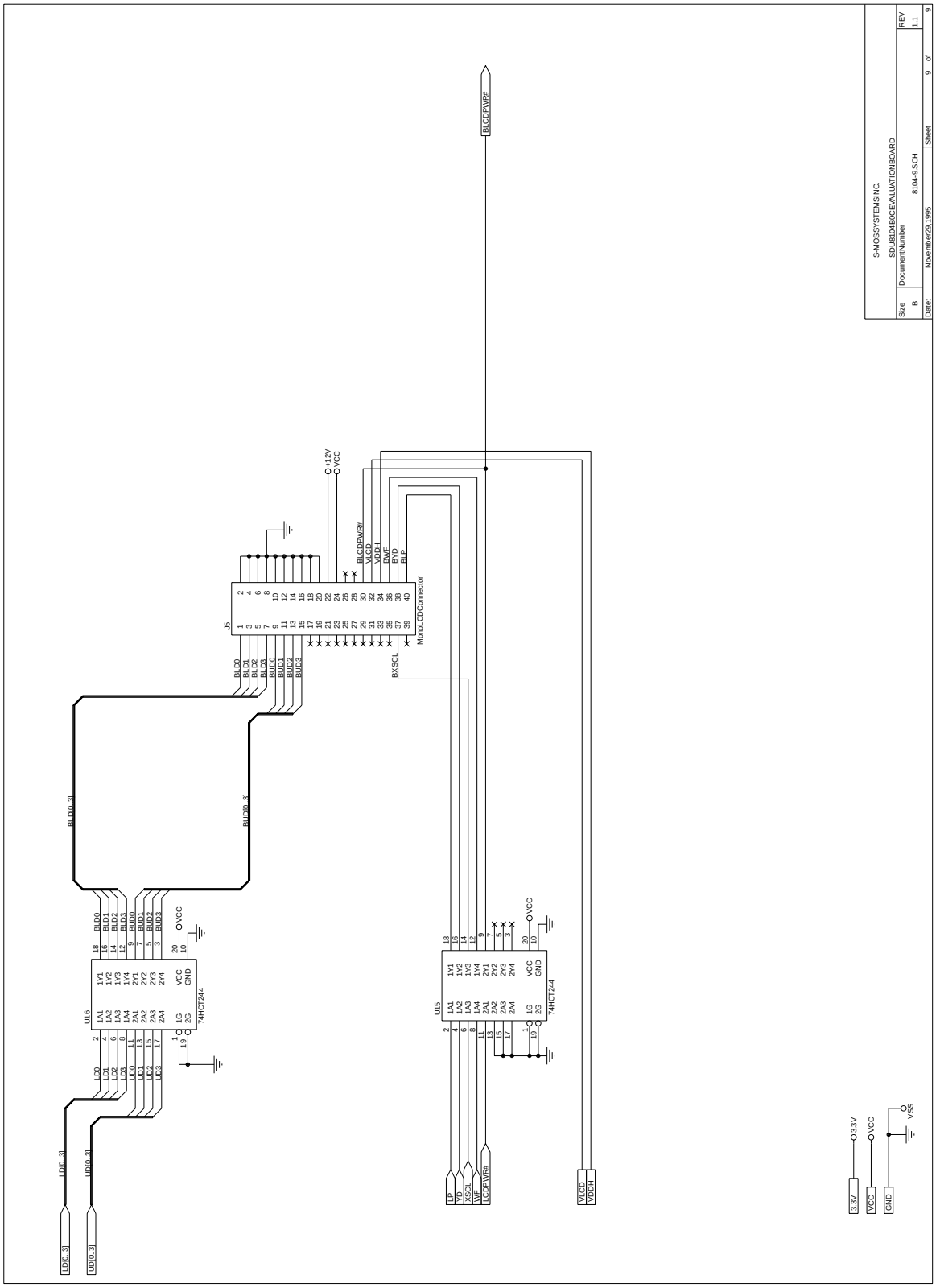


Figure 9 : SDU8104B0C Rev. 1.1 Schematics (9 of 9)

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

Power Consumption

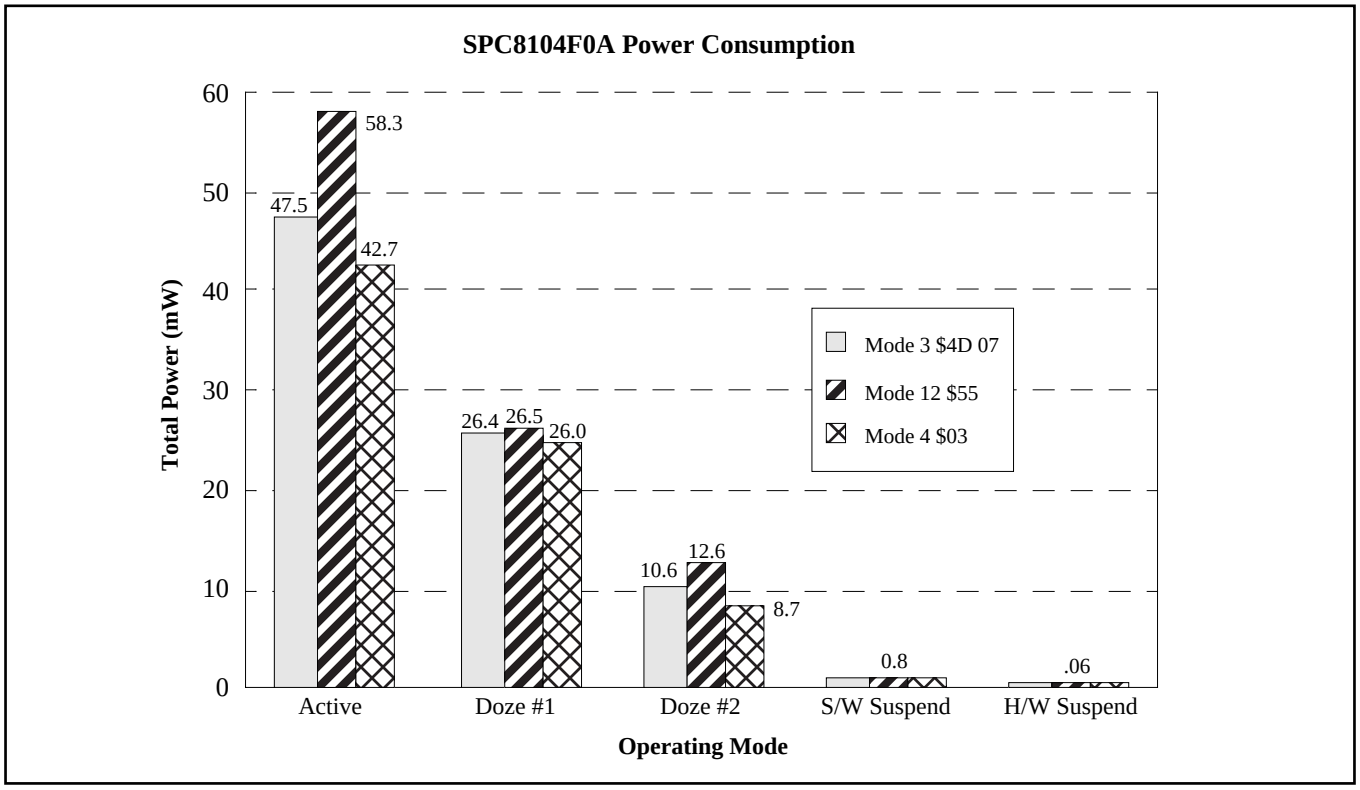
Drawing Office No. X15-AN-002-01

THIS PAGE INTENTIONALLY BLANK

1.0 SPC8104F0A POWER CONSUMPTION

1.1 Conditions

1. Mode 3: Screen filled with 4Dh 07h
2. Mode 4: Screen filled with 03h
3. Mode 12: Screen filled with 55h
4. Frequency: 24.000 MHz
5. V_{DD1} (core): 2.5 volts
   V_{DD2} (I/O): 3.3 volts
6. Controller operating in Single Panel Display mode
7. No display connected



	Active	Doze #1	Doze #2	Software Suspend	Hardware Suspend	Units
Mode 03	47.5	26.4	10.6	0.8	0.6	mW
Mode 04	42.7	26.0	8.7	0.8	0.6	mW
Mode 12	58.3	26.5	12.6	0.8	0.6	mW

THIS PAGE INTENTIONALLY BLANK

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8104 VGA LCD CONTROLLER

Power Save Mode-Doze Mode 1 Implementation Notes

Drawing Office No. X15-AN-005-01.1

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY BLANK

1.0 INTRODUCTION

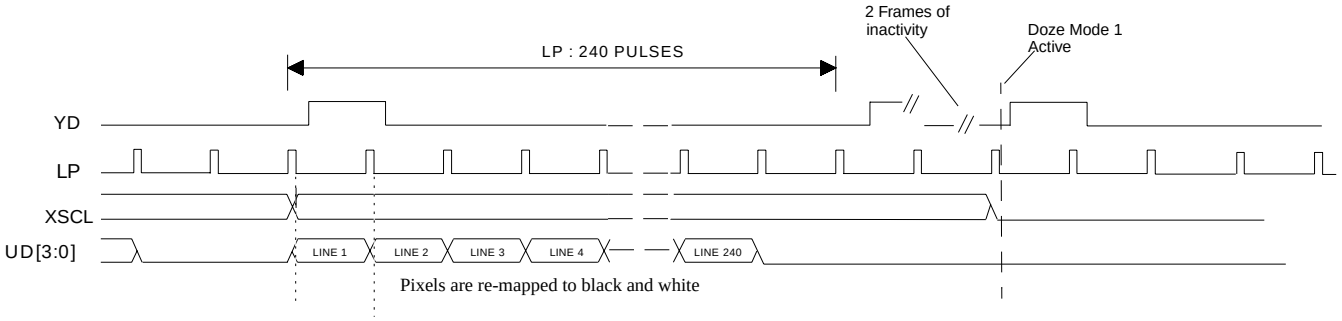
The SPC8104F0A supports three different Power Save Modes; Doze Mode 1, Doze Mode 2, and Hardware Suspend Mode. This document will describe the technical details and advantages of using Doze Mode 1.
Doze Mode 1 is intended to be used in-conjunction with RAM type LCD Drivers in order to take advantage of their Self-Refresh Mode of operation.

1.1 Reference Material

Refer to the SPC8104F0A Hardware Functional Specification (X15-SP-001-08)
Refer to Seiko Epson Column Driver SED1570 Preliminary Electrical Specification (Revision 2)
Refer to SMOS Systems SED1580 Specification (SED1580D0B revision 0.4)
SPC8104F0A VGA LCD Controller Power Consumption Application Note (X15-AN-002-01)

2.0 DOZE MODE 1 OPERATIONAL DESCRIPTION

This power save mode has an Active State and a Power Down State. When Doze Mode 1 is enabled the chip enters the Active state which is functionally the same as the normal operation. If there is no memory or I/O write for at least two vertical frames, the output pixels are first re-mapped to black and white. After two more frames with no memory or I/O write, XSCL is stopped.



Example shown: 320x240 Single 4-bit LCD

The SPC8104F0A ensures that 1 complete frame, where the pixels are re-mapped to black and white, has been transferred before stopping the XSCL signal.

Various internal blocks in the SPC8104 are also shut down to conserve power. The chip is now in the Power Down State. Upon detecting a memory or I/O write, the disabled internal blocks are re-enabled and XSCL is reactivated at the beginning of the next frame or the second to next frame.

2.1 RAM type LCD Driver - Self-Refresh Mode

RAM type LCD Drivers are drivers containing internal static RAM as well as having the ability to generate the LC drive signals. This enables them to maintain display refresh of a static image. The intended use of RAM type LCD Drivers are in applications where the display data is static for long periods of time. This will allow the display controller and various other logic devices to be shut down, thus saving power.

Self-Refresh Mode is the term referring to the situation where the transmission of display data is suspended by the display controller and the displayed image is maintained by the LCD Driver's internal RAM. This Self-Refresh mode is automatically enabled when the data shift clock (XSCL) is held a logic low for a number of horizontal display periods. The actual number of lines varies between different RAM type LCD Drivers.

2.2 Power Consumption

The power consumption of the SPC8104F0A when in Doze Mode 1 is 50% of normal mode operation. Refer to the SPC8104F0A VGA LCD Controller Power Consumption Application Note (X15-AN-002-01) for complete details.