S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

SPC8106 Technical Manual

Issue Date: 04/01/97

THIS PAGE INTENTIONALLY BLANK

1.0 CUSTOMER SUPPORT INFORMATION

Comprehensive Support Tools

S-MOS Systems provides to the system designer and computer OEM manufacturer a complete set of resources and tools for the development of VGA Graphics Systems.

Evaluation / Demonstration Board

- Assembled and fully tested graphics evaluation board with installation guide and schematics
- To borrow an evaluation board, please contact your local S-MOS sales representative or contact S-MOS at (408) 922-0200 Ext. 3440

VGA Chip Documentation

- Technical manual includes Data Sheet, Application Notes, and Programmer's Reference
- To remain on the mailing list for updates to this manual, please FAX the *Acknowledgment of Receipt* form to (408) 922-0238

Software

- Video BIOS
- OEM Utilities
- User Utilities
- Evaluation Software
- To download these programs, contact Application Engineering Support for an S-MOS Graphics BBS ID

Application Engineering Support

Engineering and Sales Support is provided by::

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

SPC8106 LCD/CRT VGA CONTROLLER

Data Sheet

THIS PAGE INTENTIONALLY BLANK

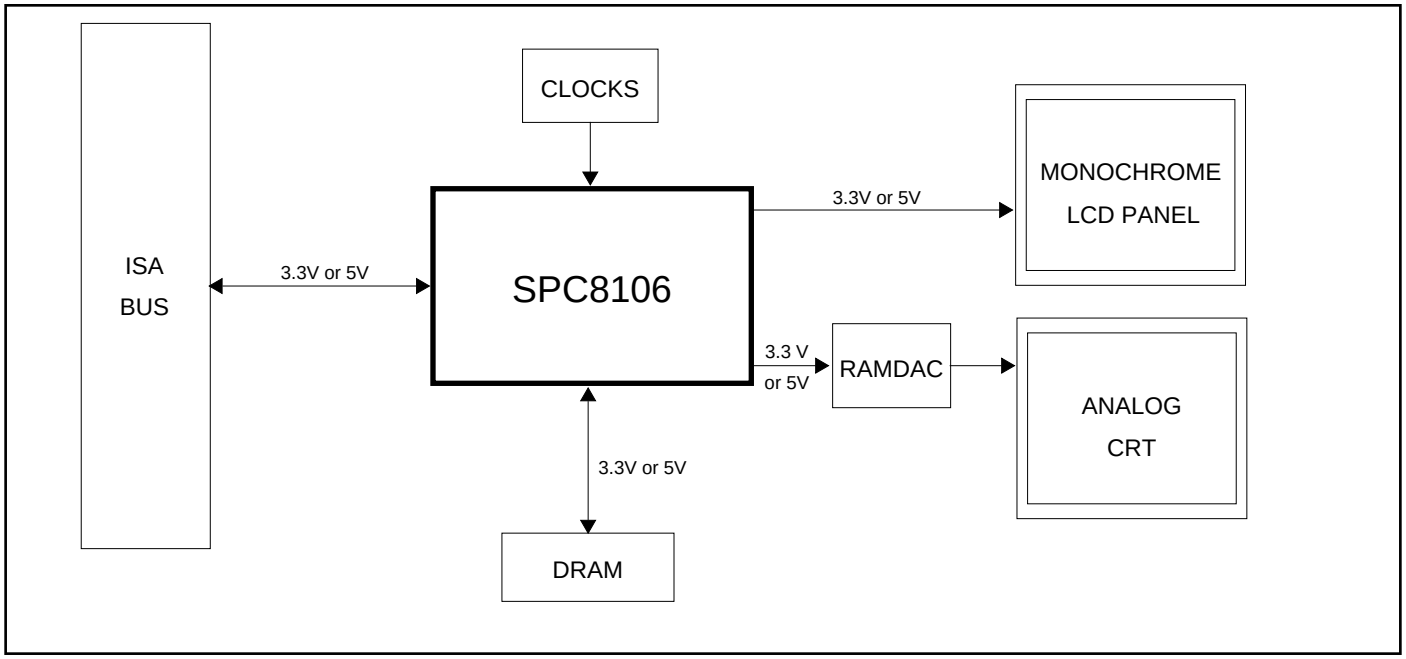
■ DESCRIPTION

The SPC8106 is a versatile mixed voltage VGA graphics controller capable of driving liquid crystal displays, TFT displays and analog CRT monitors. The controller integrates all LCD interface, sequencing and color modulation logic into one small form factor 144 pin package. With the addition of an industry standard ’477 compatible RAMDAC, the SPC8106 will also drive a VGA fixed frequency or multifrequency monitor.
The target products for this device are price and power sensitive 80x86 microprocessor based portable personal computer or other specialized LCD systems where 320 x 200 to 640 x 480 x 256 color LCD panel displays are the major design criteria.

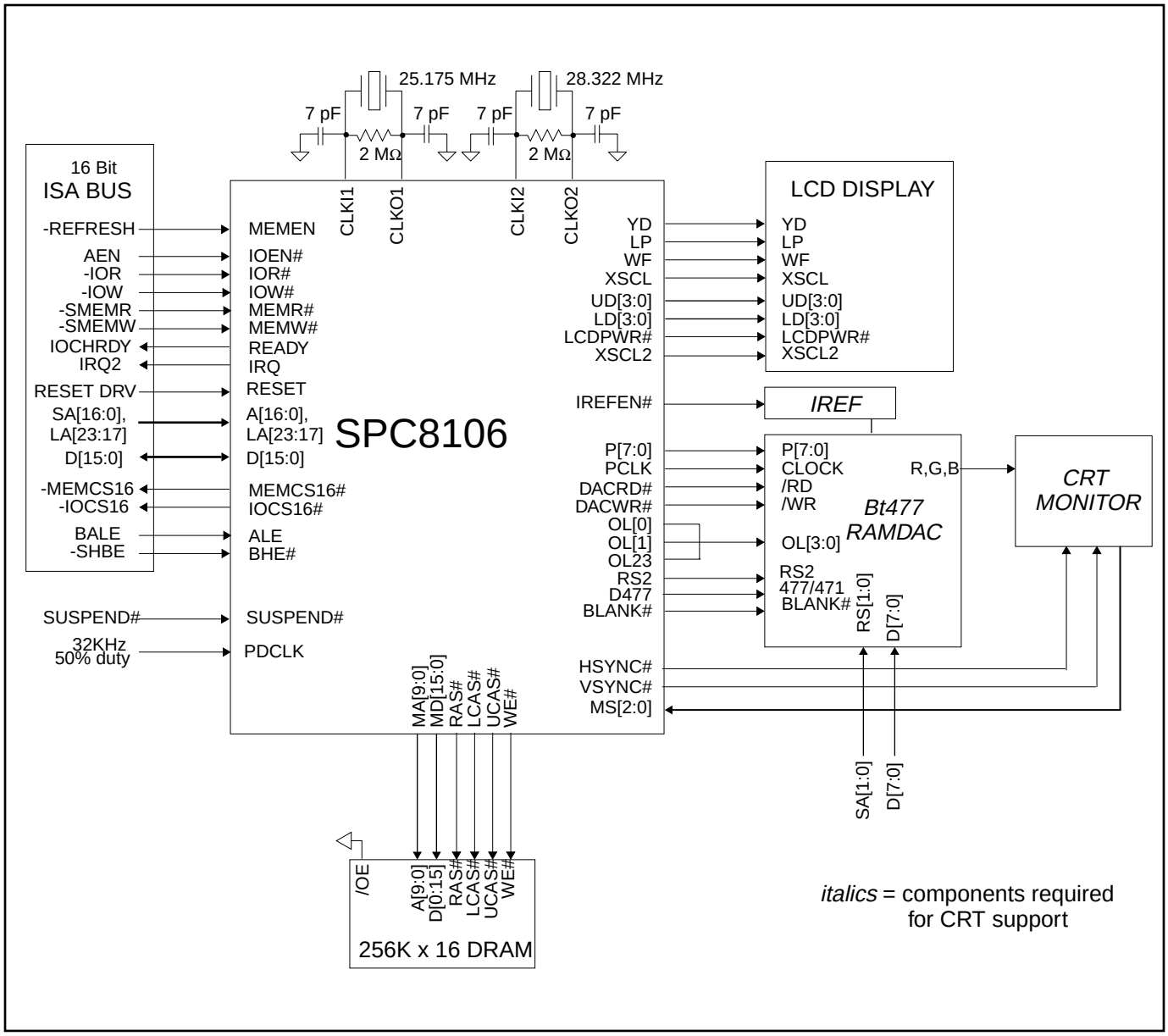
■ FEATURES

- Low-power CMOS technology
- Hardware VGA compatible
- 8- or 16-bit ISA support
- Supports one 256K x 16 80ns DRAM (self refresh optional)
- 64 x 64 x 2-bit pixel hardware cursor
- Two-terminal crystal or external oscillator support
- Hardware or software power-down
- Video BIOS, software driver and utility support
- 144-pin QFP package
- 9- or 12-bit color TFT panel interface for 640 x 480
- Single panel or dual panel interface for sizes 320 x 200 to 640 x 480
- On-chip 256 x 12-bit look-up table
- 16 gray shades or 4096 colors by FRM
- 64 gray shades by FRM and dithering
- Two programmable gray-scale weightings: NTSC and Green-Only
- Vertical centering and expansion for LCDs
- Full CRT support with ’477 compatible RAMDAC
- Pin Compatible with the SPC8108F0C
- Mixed voltage 3.3V/5V operation

■ SYSTEM BLOCK DIAGRAM



■ INTERFACE OPTIONS



Note: Example implementation, actual may vary.

■ SUPPORTED RESOLUTIONS

LCD 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
13	Graphics	N/A	N/A	320 x 200	640 x 400	64	256	A000
100	Graphics	N/A	N/A	640 x 400	640 x 400	64	256	A000
101	Graphics	N/A	N/A	640 x 480	640 x 480	64	256	A000
108	Text	8 x 8	80 x 60	640 x 480	640 x 480	16	16	B800

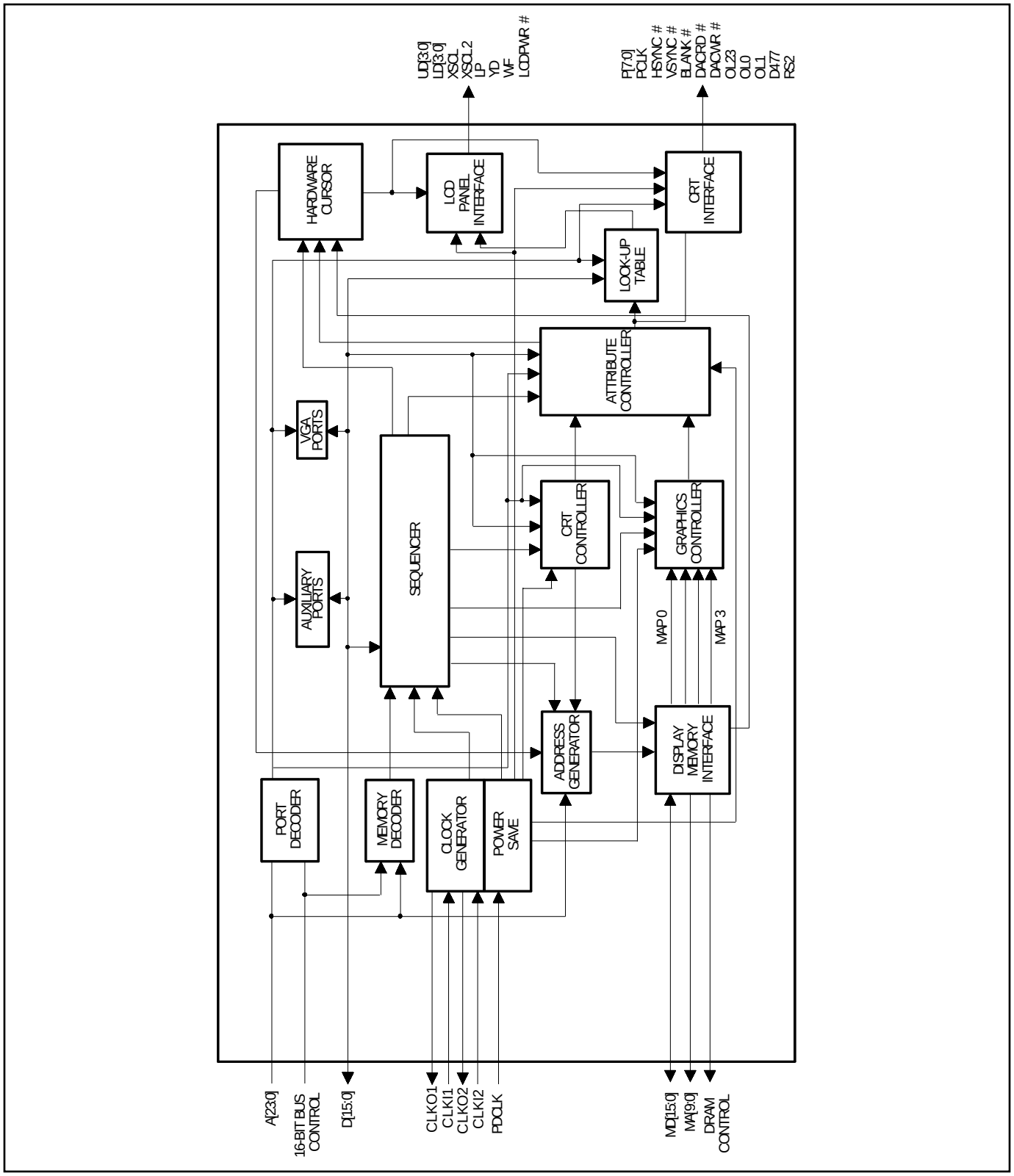
CRT Display Modes

Mode No.	Mode Type	Font	Characters	Resolution	Displayed Pixels	Colors	Memory Segment
0	Text	8 x 8	40 x 25	320 x 200	640x400	16	B800
0+	Text	8 x 14	40 x 25	320 x 350	640x350	16	B800
0++	Text	9 x 16	40 x 25	360 x 400	720x400	16	B800
1	Text	8 x 8	40 x 25	320 x 200	640x400	16	B800
1+	Text	8 x 14	40 x 25	320 x 350	640x350	16	B800
1++	Text	9 x 16	40 x 25	360 x 400	720x400	16	B800
2	Text	8 x 8	80 x 25	640 x 200	640x400	16	B800
2+	Text	8 x 14	80 x 25	640 x 350	640x350	16	B800
2++	Text	9 x 16	80 x 25	720 x 400	640x400	16	B800
3	Text	8 x 8	80 x 25	640 x 200	640x400	16	B800
3+	Text	8 x 14	80 x 25	640 x 350	640x350	16	B800
3++	Text	9 x 16	80 x 25	720 x 400	640x400	16	B800
4	Graphics	N/A	N/A	320 x 200	640x400	4	B800
5	Graphics	N/A	N/A	320 x 200	640x400	4	B800
6	Graphics	N/A	N/A	640 x 200	640x400	2	B800
7	Text	8 x 14	80 x 25	640 x 350	640x350	2	B000
7+	Text	9 x 16	80 x 25	720 x 400	720x400	2	B000
0D	Graphics	N/A	N/A	320 x 200	640x400	16	A000
0E	Graphics	N/A	N/A	640 x 200	640x400	16	A000
0F	Graphics	N/A	N/A	640 x 350	640x350	2	A000
10	Graphics	N/A	N/A	640 x 350	640x350	16	A000
11	Graphics	N/A	N/A	640 x 480	640x480	2	A000
12	Graphics	N/A	N/A	640 x 480	640x480	16	A000
13	Graphics	N/A	N/A	320 x 200	640x400	256	A000
100	Graphics	N/A	N/A	640 x 400	640x400	256	A000
101	Graphics	N/A	N/A	640 x 480	640 x 480	256	A000
108	Text	8 x 8	80 x 60	640 x 480	640 x 480	16	B800

■ SUPPORTED LCD INTERFACES

8-Bit Interface				4-Bit Interface	
Dual Panel		Single Panel		Single Panel	
Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical
640	400 480	640	1 to 480	320 480 640	200 240 320 400 480

■ BLOCK DIAGRAM



■ FUNCTIONAL BLOCK DESCRIPTION

The 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.

CRT Controller

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

CRT Interface

The CRT interface aligns CRT signals to the Pixel Clock and generates the I/O Control signals for CPU access to the RAMDAC.

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.

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.

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.

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.

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.

Port Decoder

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

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.

VGA Ports

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

Clock Generation

The Clock Generation contains oscillator support for external crystals.

Power Save

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

Lookup Table

The Lookup Table consists of a memory array of 256 locations of 12 bits each and hardware to convert VGA palette writes to gray-scale values.

LCD 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. For monochrome LCD panels, the LCD interface block generates a maximum 64 gray shades through frame rate modulation and dithering techniques. For color LCD panels, the LCD interface block generates 256 simultaneous colors from a possible 4096 colors through frame rate modulation.

Hardware Cursor

The Hardware Cursor block generates a 4 gray shade or color cursor/sprite that can be overlaid on the LCD or CRT display. The cursor is 64 x 64 pixels or optionally expanded to 128 x 128 through pixel replication.

■ 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} = 0V	4.5	5.0	5.5	V
LV_{DD}	Supply Voltage	V_{SS} = 0V	3.0	3.3	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 Current Consumption	Vcc Core = 3.3V V_{CC} I/O = 5.0V	typical I_{Core} = 52.31 typical I_{IO} = 13.85			mA

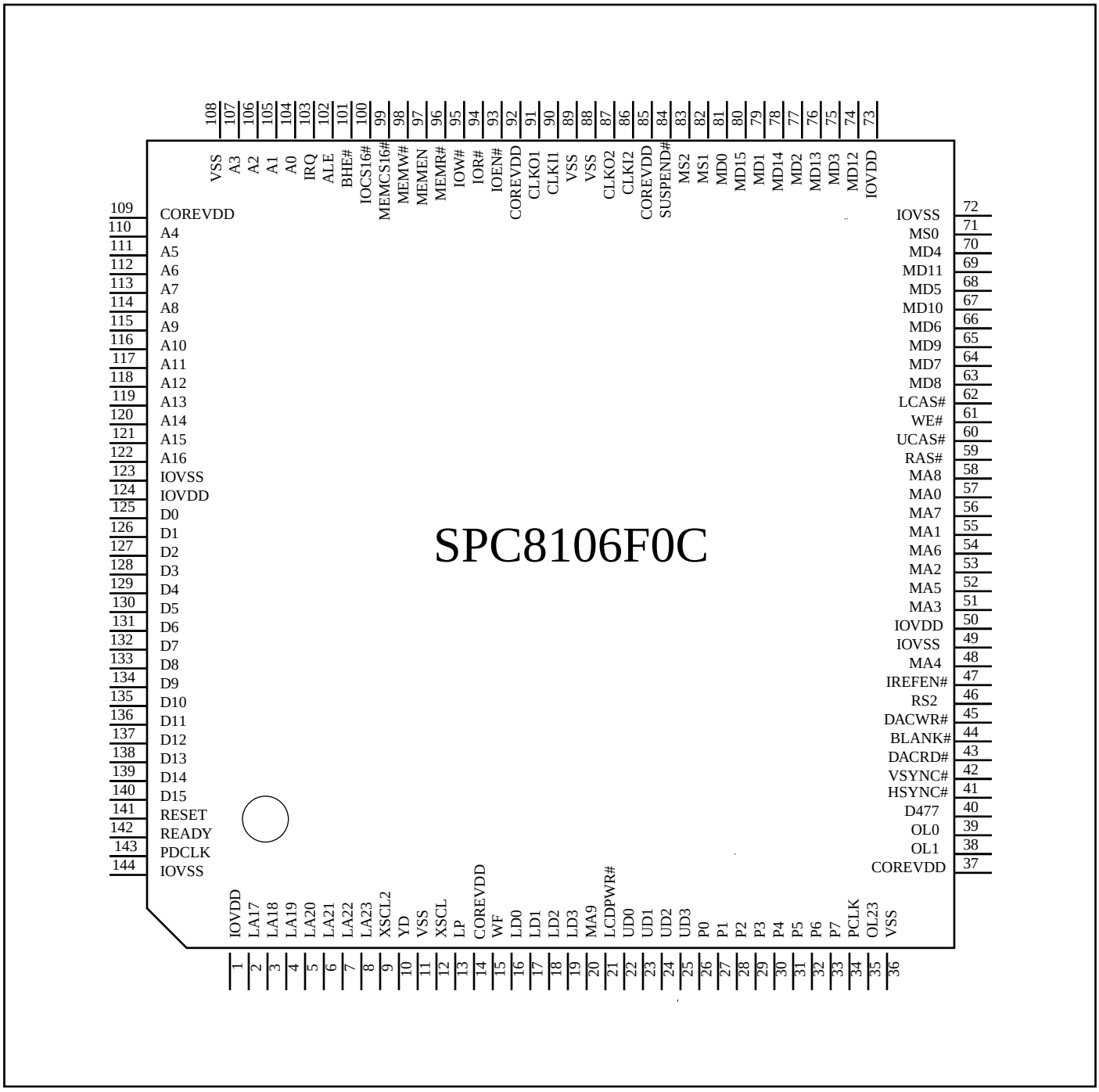
Input Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (CMOS inputs)	V_{DD} = MIN			1.0	V
V_{IH}	High Level Input Voltage (CMOS inputs)	V_{DD} = MAX	3.5			V
V_{IL}	Low Level Input Voltage (TTL inputs)	V_{DD} = MIN			0.8	V
V_{IH}	High Level Input Voltage (TTL inputs)	V_{DD} = MAX	2.0			V
V_{T+}	Positive-going Threshold (CMOS Schmitt inputs)	V_{DD} = 5.0			4.0	V
V_{T-}	Negative-going Threshold (CMOS Schmitt inputs)	V_{DD} = 5.0	0.8			V
V_H	Hysteresis Voltage (CMOS Schmitt inputs)	V_{DD} = 5.0	0.3			V
V_{T+}	Positive-going Threshold (TTL Schmitt inputs)	V_{DD} = 5.0			3.0	V
V_{T-}	Negative-going Threshold (TTL Schmitt inputs)	V_{DD} = 5.0	0.6			V
V_H	Hysteresis Voltage (TTL Schmitt inputs)	V_{DD} = 5.0	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_{PU2}	Pull Up Resistance	V_{DD} = 5.0 V	50	100	200	kΩ
R_{PU3}	Pull Up Resistance	V_{DD} = 5.0 V	100	200	400	kΩ
R_{PD}	Pull Down Resistance	V_{DD} = 5.0 V	100	200	400	kΩ

Output Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{OL2}	Low Level Output Current	$V_{OL}=V_{SS}+0.4V$ TS2	6.0			mA
I_{OH2}	High Level Output Current	$V_{OH}=V_{DD}-0.4V$ TS2	-2.0			mA
I_{OL3}	Low Level Output Current	$V_{OL}=V_{SS}+0.4V$ TS3	12.0			mA
I_{OH3}	High Level Output Current	$V_{OH}=V_{DD}-0.4V$ TS3	-4.0			mA
I_{OL4}	Low Level Output Current	$V_{OL}=V_{SS}+0.4V$ TS4	24.0			mA
I_{OH4}	High Level Output Current	$V_{OH}=V_{DD}-0.4V$ TS4	-8.0			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
C_{BID}	Bidirectional Pin Capacitance			10		pF

■ SPC8106F0C PIN OUTSOURCE: 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	104..107, 110..122, 2..4, 5..8	CPU bus address inputs. In Suspend Mode, the Address inputs are internally masked off. If the value on MD[5] at RESET = 1, then the ALE input pin is used to internally latch LA[19:17] and A[16:2], allowing these address bits to be driven by the processor address bus. If the value on MD[5] at RESET = 0, then standard ISA address timing is assumed, where pins A[0:16], LA[17:23] should be connected to the ISA bus signals SA[0:16], LA[17:23] respectively.
ALE	I	102	ISA Bus Address Latch Enable. In Suspend Mode the ALE input is disabled. If the value on MD[5] at RESET = 1, then the ALE input is used to internally latch LA[19:17] and A[16:2], allowing these address bits to be driven by the processor address bus. In this mode, the processor ADS# output should be connected to this pin. If the value on MD[5] at RESET = 0, then standard ISA address timing is assumed, and only the LA[19:17] inputs are internally latched.
D[0:15]	I/O	125..140	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.
MEMEN	I	97	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 IOR# input is disabled.
IOW#	I	95	ISA Bus I/O Write Strobe. In Suspend Mode the IOW# input is disabled.
MEMR#	I	96	ISA Bus System Memory Read Strobe. In Suspend Mode the MEMR# input is disabled.
MEMW#	I	98	ISA Bus System Memory Write Strobe. In Suspend Mode the MEMW# input is disabled.
IOEN#	I	93	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 IOEN# input is disabled.
READY	O	142	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	141	The active high Reset signal from the CPU clears all internal registers and forces all signals to their inactive state.
IRQ	O	103	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). This pin also is used for the output of the NAND tree in pin test mode.
MEMCS16#	O	99	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.

Pin Name	Type	Pin #	Description
IOCS16#	O	100	ISA Bus I/O Chip Select 16. Address inputs A[15:0] and IOEN# are decoded to drive this output low when a valid SPC8106 I/O register address appears on the bus,. Note that I/O addresses 3C6h-3C9h do not result in IOCS16# being driven low (i.e. RAMDAC and internal LUT register reads and writes are 8 bit cycles).
BHE#	I	101	ISA Bus Byte High Enable. In Suspend Mode the BHE# input is disabled.

Video Memory Interface

Pin Name	Type	Pin #	Description
MA[0:9]	O	57, 55, 53, 51, 48, 52, 54, 56, 58, 20	Multiplexed row/column address bits for video display memory.
MD[0:15]	I/O	81, 79, 77, 75, 70, 68, 66, 64, 63, 65, 67, 69, 74, 76, 78, 80	Data bits for video display memory. The output drivers of these pins are placed into a high-impedance state when RESET is high, or when the Sequencer is in a reset state. On the falling edge of RESET, the values on MD[3:0] and MD[12:9] are latched into a read-only Auxiliary Register and are available to be read as configuration inputs. Also, the value on MD[8:4] and MD[15:13] are used to configure various hardware options. See “Summary of Configuration Options” on page 18 for details.
RAS#	O	59	DRAM Row Address Strobe for single 256Kx16 DRAM.
LCAS# (LWE#)	O	62	Multiple Function: DRAM Column Address Strobe for low byte (LCAS#). For alternate function see “Multiple Function Pin Descriptions” on page 19.
UCAS# (CAS#)	O	60	Multiple Function: DRAM Column Address Strobe for high byte (UCAS#). For alternate function see “Multiple Function Pin Descriptions” on page 19.
WE# (UWE#)	O	61	Multiple Function: DRAM Write Enable Strobe (WE#). For alternate function see “Multiple Function Pin Descriptions” on page 19.

Clock Inputs

Pin Name	Type	Pin #	Description
CLKI1	I	90	This pin, along with CLKO1 is the 25.175 MHz 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin is the clock input.
CLKO1	O	91	This pin, along with CLKI1 is the 25.175 MHz2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin should be left unconnected.
CLKI2	I	86	This pin, along with CLKO2 is the 28.322 MHz 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin is the clock input.
CLKO2	O	87	This pin, along with CLKI2 is the 28.322 MHz 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin should be left unconnected.

LCD Panel Interface

Pin Name	Type	Pin #	Description
YD	O	10	Vertical Scanning Start Pulse output. A logic 1 on this signal, sampled by the LCD 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	13	Latch Pulse output. The falling edge of this signal is used to latch a row of display data in the LCD module's column driver shift registers and to turn on the row driver (Y driver) for that line.
XSCL	O	12	Shift Clock for LCD data. Display data is clocked out of the chip on the rising edge of this signal, to be shifted into the LCD panel module column drivers (X drivers) on each falling edge.
XSCL2	O	9	This second shift clock is used together with XSCL in 8-bit single color panel mode to shift in alternate sets of display data. XSCL2 is also used alone as the shift clock in 8-bit dual color panel mode and 4-bit single color panel mode.
UD[0:3]	O	22..25	Upper panel display data for dual panel - dual drive mode. For 8-bit single panel-single drive mode, these bits are the most significant 4-bits of the 8-bit output data to the panel (data[7:4]). For 4-bit single panel mode, these bits are the 4 bits of data output to the panel. For 16-bit LCD modes, these outputs are the multiplexed upper panel data if MD[7]=1 at RESET, or the lower nibble of the upper panel data if MD[7]=0 at RESET.
UD[4:7]	O	26..29	When MD[7]=0 at RESET, these pins are the upper nibble of the 16-bit LCD mode upper panel data.
LD[0:3]	O	16..19	Lower panel display data for dual panel-dual drive mode. For 8-bit single panel-single drive mode, these bits are the least significant 4 bits of the 8-bit output data to the panel (data[3:0]). For 4-bit single panel mode, these outputs are driven low. For 16-bit LCD modes, these outputs are the multiplexed lower panel data if MD[7]=1 at RESET, or the lower nibble of the lower panel data if MD[7]=0 at RESET.
LD[4:7]	O	30..33	When MD[7]=0 at RESET, these pins are the upper nibble of the 16-bit LCD mode lower panel data.
LCDPWR#	O	21	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, when Auxiliary Register 06 bit 0 is set to 1, or when 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	15	LCD Backplane Bias signal. This output toggles once every n LP periods, as programmed in Auxiliary Register [0D].

External CRT/RAMDAC Interface

Pin Name	Type	Pin #	Description
P[0:7]	O	26..33	When MD[7]=1 at RESET, these pins are the Pixel Data outputs. These 8 bits are connected to the pixel select inputs of the external RAMDAC.
PCLK	O	34	Pixel Clock. Pixel data is clocked out of the chip on the falling edge of PCLK.
BLANK#	O	44	Blank output. This output is clocked out on the falling edge of PCLK and is driven low during display blanking periods.
HSYNC#	O	41	Horizontal Sync. This output is clocked out on the falling edge of PCLK and is driven to indicate the horizontal retrace period. The polarity of this signal is determined by a control bit in register 3C2h.
VSYNC#	O	42	Vertical Sync. This output is clocked out on the falling edge of PCLK and is driven to indicate the vertical retrace period. The polarity of this signal is determined by a control bit in register 3C2h.
DACRD#	O	43	RAMDAC Read Strobe. This signal goes low when a valid read access to the VGA RAMDAC is decoded by the chip.
DACWR#	O	45	RAMDAC Write Strobe. This signal goes low when a valid write access to the VGA RAMDAC is decoded by the chip.
RS2	O	46	Register Select 2 output. This output should be connected to the RS2 input of the RAMDAC (Bt477 or equivalent). The logic level on this output may be set by setting Auxiliary Register [0B] bit 3. This signal is required to allow CPU access the control and overlay registers of the external RAMDAC.
OL[0:1]	I/O	39, 38	Multiple Function: Overlay Select outputs 1:0 When MD[13]=0 at RESET, these pins are outputs used to provide sprite/HW cursor function on the CRT display. In this case, these outputs should be connected to the OL[0:1] inputs of the RAMDAC (Bt477 or equivalent). They are used by the sprite circuitry to access the overlay registers in the RAMDAC. For alternate function see “Multiple Function Pin Descriptions” on page 19.
OL23	O	35	Overlay Select output 2/3. This output should be connected to both the OL2 and OL3 inputs of the RAMDAC (Bt477 or equivalent). This signal is used by the sprite circuitry to access the overlay registers in the RAMDAC.For alternate function see “Multiple Function Pin Descriptions” on page 19.
D477	O	40	477 Control Signal. This output should be connected to the 477/471 input of the RAMDAC (Bt477 or equivalent). This signal is used to access the control register of the RAMDAC and to allow it to be powered down. The logic level on this output can be controlled by setting Auxiliary Register [0B] bit 4, and is also controlled by the power save logic.
IREFEN#	O	47	IREF Enable output. This signal is used to control the external current reference source required by the RAMDAC, allowing powering down the analog circuitry when not required. When this signal is driven low, the external current reference should be enabled. When this signal is high, the external current reference should be shut off.
MS[2:0]	I/O	83, 82, 71	Monitor Sense inputs. These signals should be connected to the monitor sense lines from the CRT monitor cable. The status of these bits is readable in Auxiliary Register [08] bits 2:0, and is used by BIOS software to determine the presence and type of monitor connected. Optionally, the SENSE output of the RAMDAC may be connected to one of these inputs to allow the BIOS to read the SENSE signal and detect the monitor. MS[2:1] can be forced low by the DCC2 monitor support bits in Auxiliary Register [10] bits 1:0.

Power Save Mode Control

Pin Name	Type	Pin #	Description
SUSPEND#	I	84	A low level on this pin puts the chip into a hardware power down mode. The SUSPEND# signal overrides any software initiated power down modes, and disables the ISA-Bus interface inputs except RESET. Address and Data inputs are also masked when this signal is low. When in Suspend Mode the UD(3:0), LD(3:0), XSCL, XSCL2, LP, YD and WF signals are driven into a high impedance or low state (configurable) and the LCDPWR# signal is driven high.
PDCLK	I	143	Power Down Clock. This input may be used to provide a low frequency clock for generating refresh in Power Save Modes 4 and Suspend, as an optional alternative to using the pixel clock or MEMEN input as the refresh clock source. This clock input should be driven by either by a 32 kHz 50% duty cycle clock source, or a 64 kHz clock source with a high period as short as possible (but > minimum RAS low pulse width) to minimize DRAM current consumption during refresh. The PDCLK input is used to directly generate the RAS and CAS pulses during Power Save Mode 4 and Suspend.

Power Supply

Pin Name	Type	Pin #	Description
COREVDD	P	14, 37, 85, 92, 109	V_{DD} supply for core logic.
IOVDD	P	1, 50, 73, 124	V_{DD} supply for interface pins.
VSS	P	11, 36, 88, 89, 108	V_{SS} supply for core logic.
IOVSS	P	49, 72, 123, 144	V_{SS} supply for interface pins.

Pin Mapping for Various Display Modes

SPC8106 Pin Name	Display Mode		RGBI	12-bit RGB	TFT		
	CRT	LCD			9-bit	12-bit AUX[00]b5=1 AUX[0B]b1=0	12-bit AUX[00]b5=1 AUX[0B]b1=1
VSYNC#	VSYNC#	None	None	None	None	None	VSYNC#
HSYNC#	HSYNC#	None	None	None	None	None	HSYNC#
YD	None	YD	VSYNC#	VSYNC#	VSYNC#	VSYNC#	None
LP	None	LP	HSYNC#	HSYNC#	HSYNC#	HSYNC#	None
WF	None	WF	None (forced 0)	None (forced 0)	DATAEN	DATAEN	DATAEN
XCSL	None	XCSL	PCLK	PCLK	PANCLK	PANCLK	PANCLK
XCSL2	None	XCSL2	None (forced 0)	R[3]	R[2]	R[3]	R[3]
UD[3]	None	UD[3]	None (forced 0)	B[3]	B[2]	B[3]	B[3]
UD[2]	None	UD[2]	None (forced 0)	B[2]	B[1]	B[2]	B[2]
UD[1]	None	UD[1]	None (forced 0)	B[1]	B[0]	B[1]	B[1]
UD[0]	None	UD[0]	None (forced 0)	R[2]	R[1]	R[2]	R[2]
LD[3]	None	LD[3]	D[3]	G[3]	G[2]	G[3]	G[3]
LD[2]	None	LD[2]	D[2]	G[2]	G[1]	G[2]	G[2]
LD[1]	None	LD[1]	D[1]	G[1]	G[0]	G[1]	G[1]
LD[0]	None	LD[0]	D[0]	R[1]	R[0]	R[1]	R[1]
OL0	OL0	None	None	B[0]	None	B[0]	B[0]
OL1	OL1	None	None	G[0]	None	G[0]	G[0]
OL23	OL23	None	None	R[0]	None	R[0]	R[0]

Mixed Voltage Configurations

Core VDD	I/O VDD	
	3.3 V	5.0 V
3.3 V	OK	OK
5.0 V	NO	OK

Summary of Configuration Options

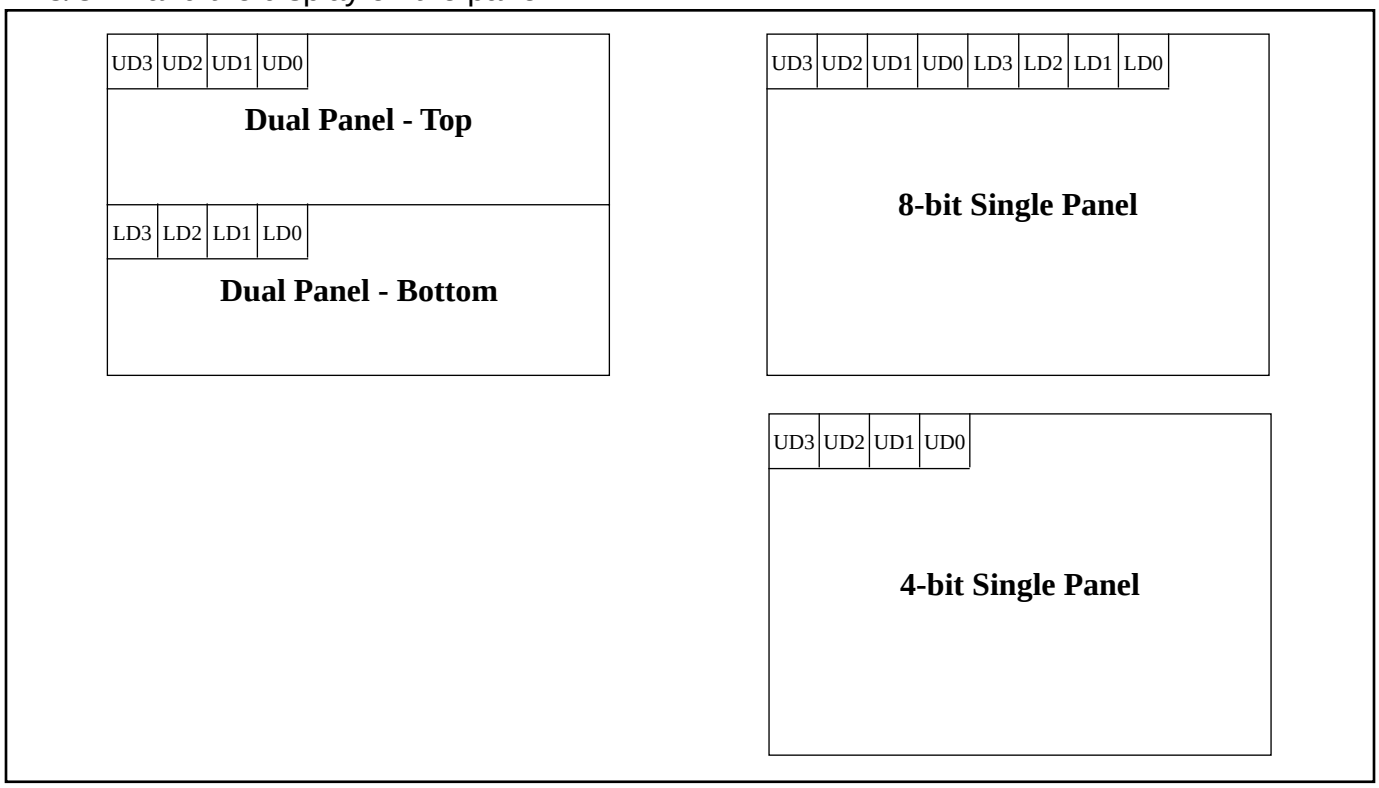
Pin Name	value on this pin at falling edge of RESET is used to configure: (1/0)
MD[3:0]	values latched into read-only Aux Reg[0C] bits 3:0 for software use
MD[4]	16-bit I/O interface (1) / 8-bit I/O interface (0)
MD[5]	A[19:2] latched internally by ALE (1) / standard ISA bus ALE - A[16:0] not latched (0)
MD[6]	2 CAS, 1 WE type DRAM (1) / 1 CAS, 2 WE type DRAM (0)
MD[7]	support 16-bit panel with external logic (1) / support 16-bit panel directly (0)
MD[8]	5 V core operating voltage (1) / 3.3 V core operating voltage (0)
MD[12:9]	values latched into read-only bits 7:4 of Aux Reg[0C] for software use
MD[13]	pins 38, 39 used for ext. RC for 32 kHz PDCLK (1) / pins 38, 39 used for OL[1:0] (0)
MD[14]	Internal PDCLK doubling disable (1) / enable (0)
MD[15]	3C3h used as video enable port (1) / 46E8h and 102h used as video enable port (0)

These inputs 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 (V_{DD} = 5.0V).

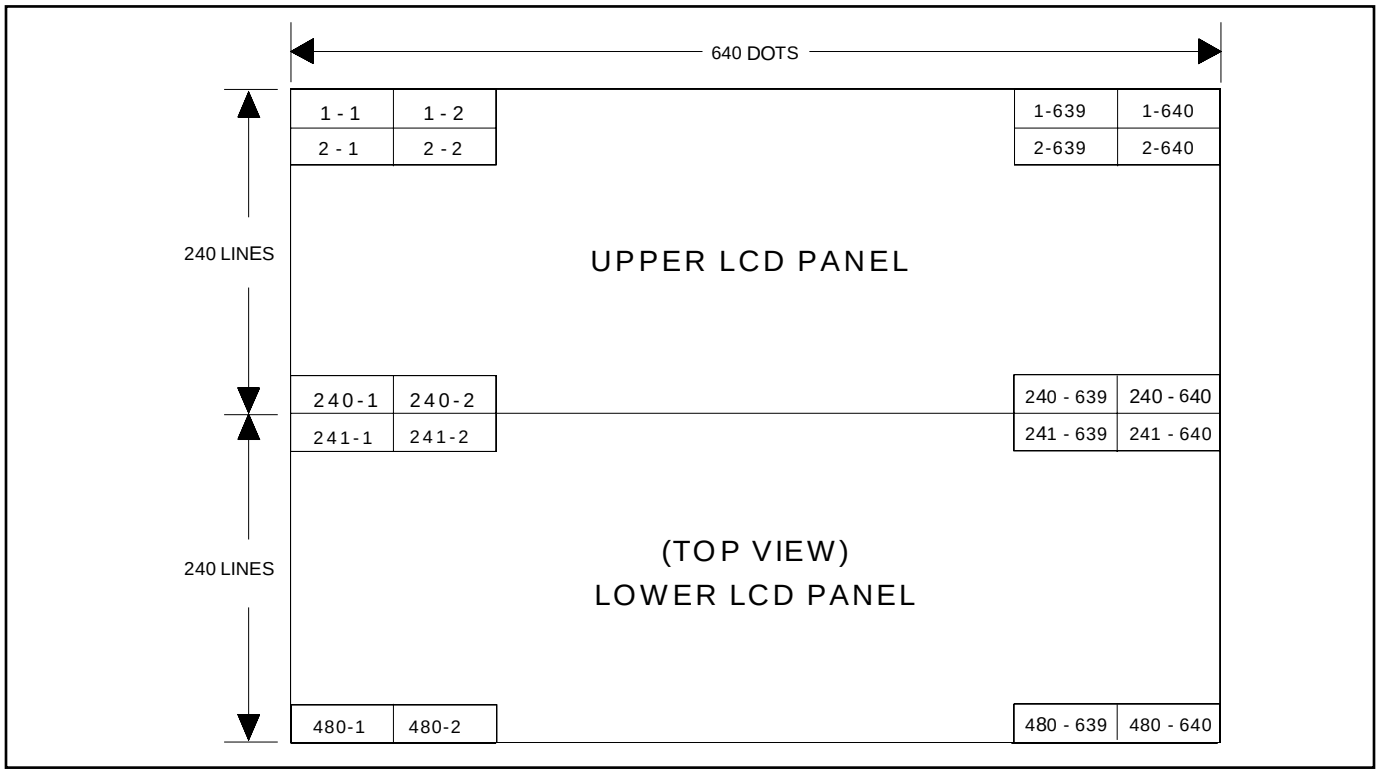
Multiple Function Pin Descriptions

Pin Name	Function	MD Line Status	Functional Description
LCAS#, LWE#	LCAS#	MD6 = 1	DRAM Column Address Strobe (Low Byte)
	LWE#	MD6 = 0	DRAM Write Strobe (Low Byte)
UCAS#, CAS#	UCAS#	MD6 = 1	DRAM Column Address Strobe (High Byte)
	CAS#	MD6 = 0	DRAM Column Address Strobe
WE#, UWE#	WE#	MD[6] = 1	DRAM Write Strobe
	UWE#	MD[6] = 0	DRAM Write Strobe (High Byte)
OL0, P320, B0	OL0	MD[13] = 0 AUX[00] b6=0	Overlay Bit 0. Used for CRT HW Cursor/Sprite support.
	P320	MD[13] = 1 MD[14] = 1	32 kHz Clock Output. Used with external RC when using external PDCLK support
	B0	MD[13] = 0 AUX[00] b6=1	Data bit B0 for 12-bit TFT support
OL1, P32I, G0	OL1	MD[13] = 0 AUX[00] b6=0	Overlay Bit 1. Used for CRT HW Cursor/Sprite support
	P32I	MD[13] = 1 MD[14] = 1	32 kHz Clock Input. Used with external RC when using external PDCLK support
	G0	MD[13] = 0 AUX[00] b6=1	Data bit G0 for 12-bit TFT support
OL23, R0	OL23	MD[13] = 0 AUX[00] b6=0	Overlay Bit 2. Used for CRT HW Cursor/Sprite support.
	R0	MD[13] = 0 AUX[00] b6=1	Data bit R0 for 12-bit TFT support
P[0:3]	P[0:3}	MD[7] = 1	Lower nibble of the CRT pixel data outputs
	UD[4:7}	MD[7] = 0	Upper nibble of the 16-bit LCD mode upper panel data
P[4:7]	P[4:7]	MD[7] = 1	Upper nibble of the CRT pixel data outputs
	LD[4:7]	MD[7] = 0	Upper nibble of the 16-bit LCD mode lower panel data

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:

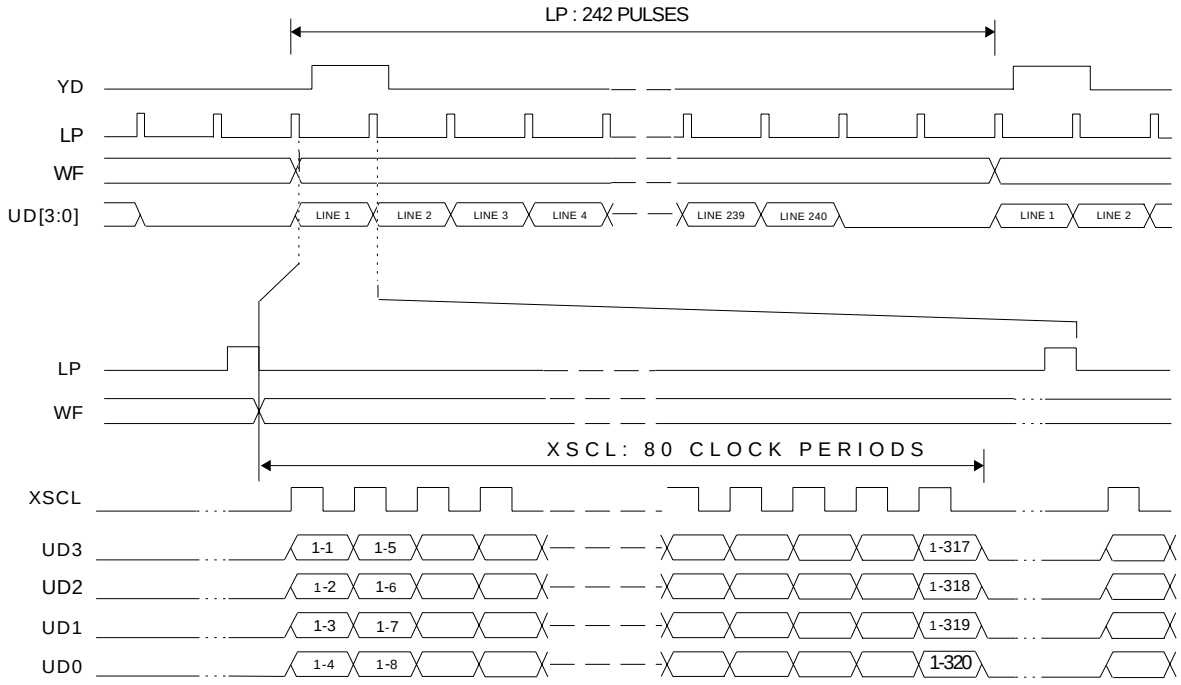


■ LCD PANEL PIXELS



■ MONOCHROME PASSIVE STN LCD PANEL INTERFACE

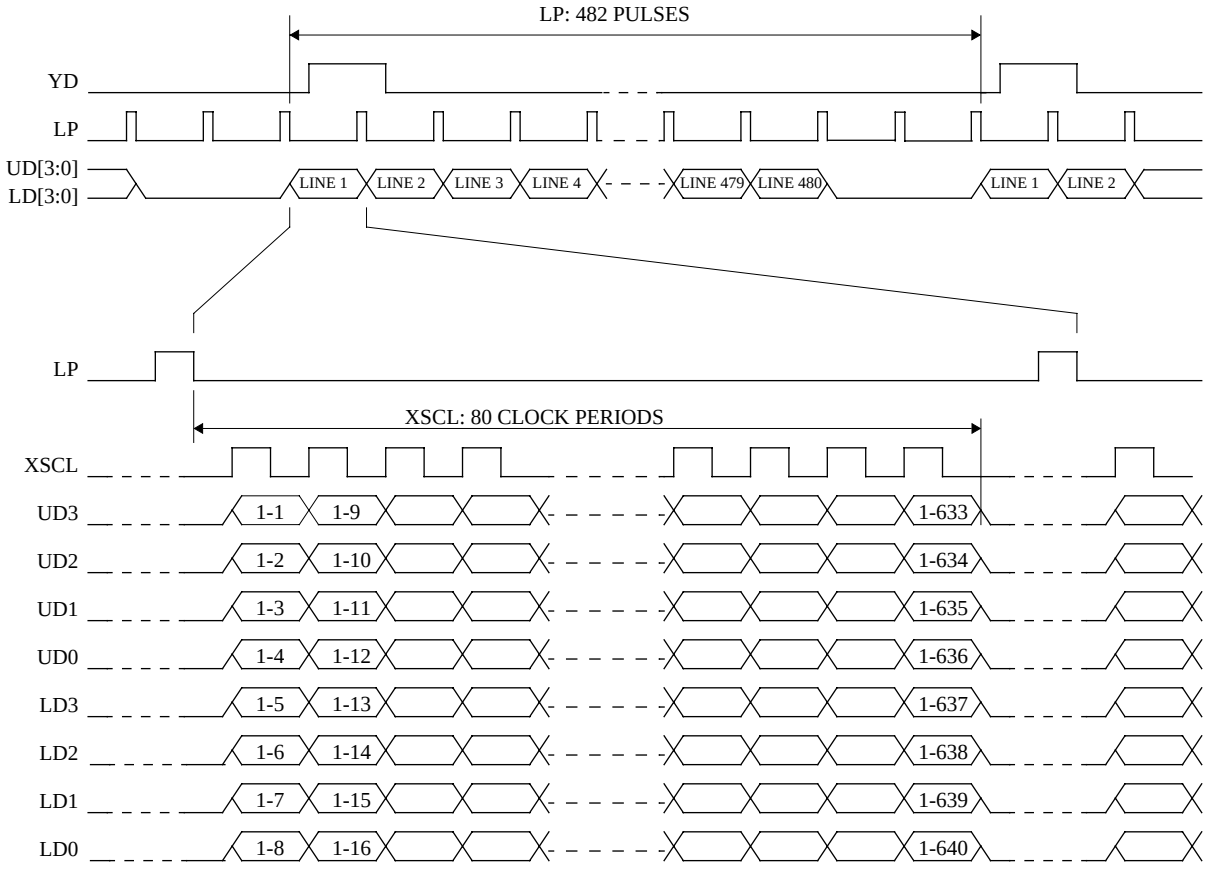
4-BIT SINGLE PANEL



Example timing for a 320 x 240 panel

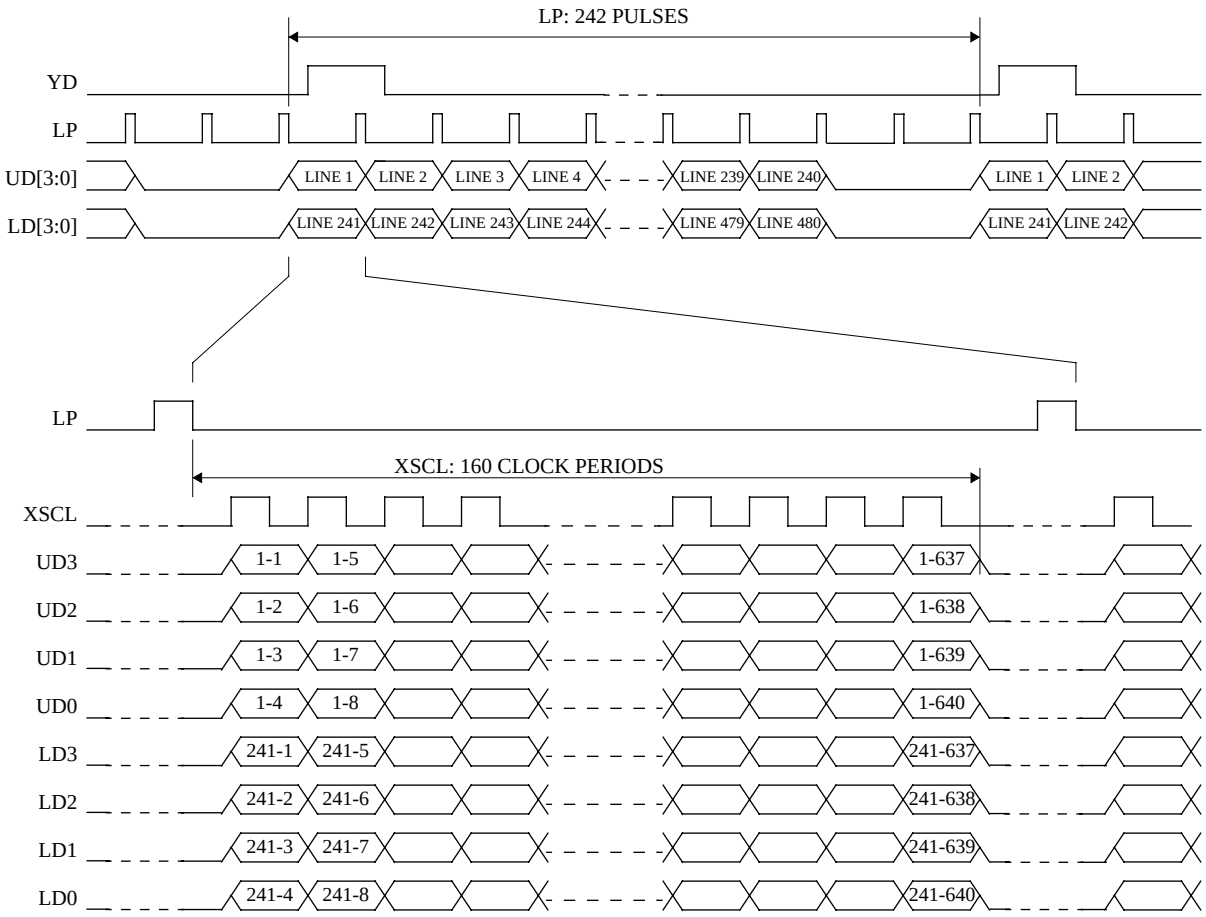
■ MONOCHROME PASSIVE STN LCD PANEL INTERFACE

8-BIT SINGLE PANEL



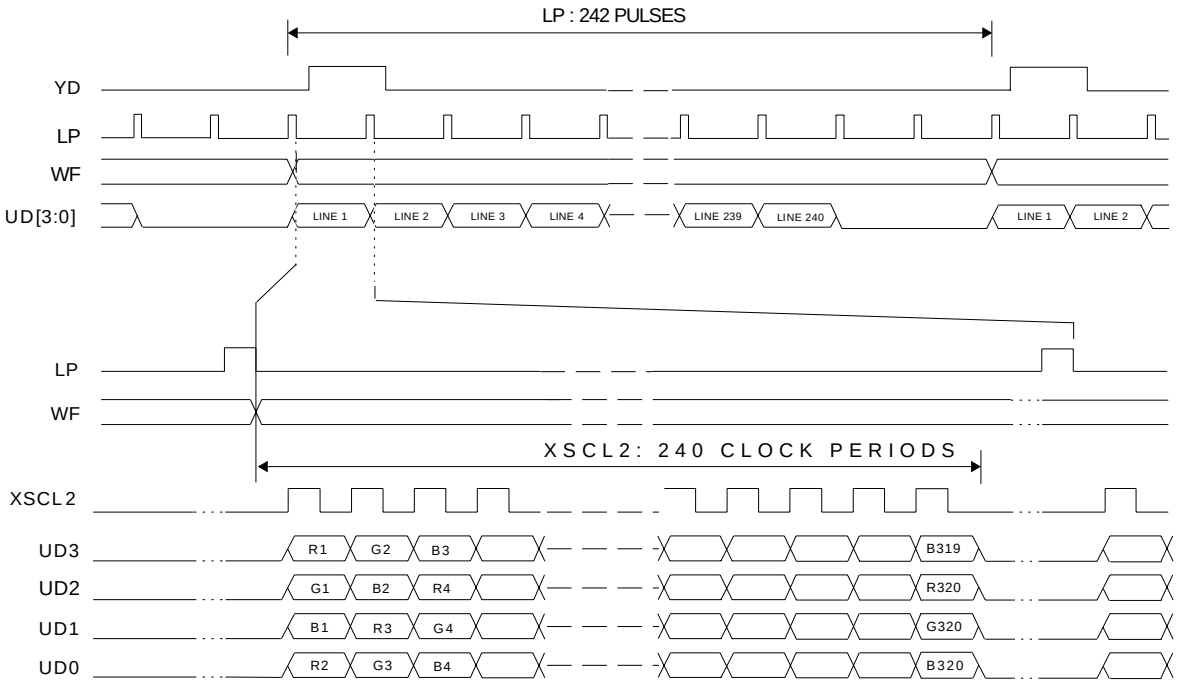
■ MONOCHROME PASSIVE STN LCD PANEL INTERFACE

8-BIT DUAL PANEL



■ COLOR STN LCD PANEL INTERFACE

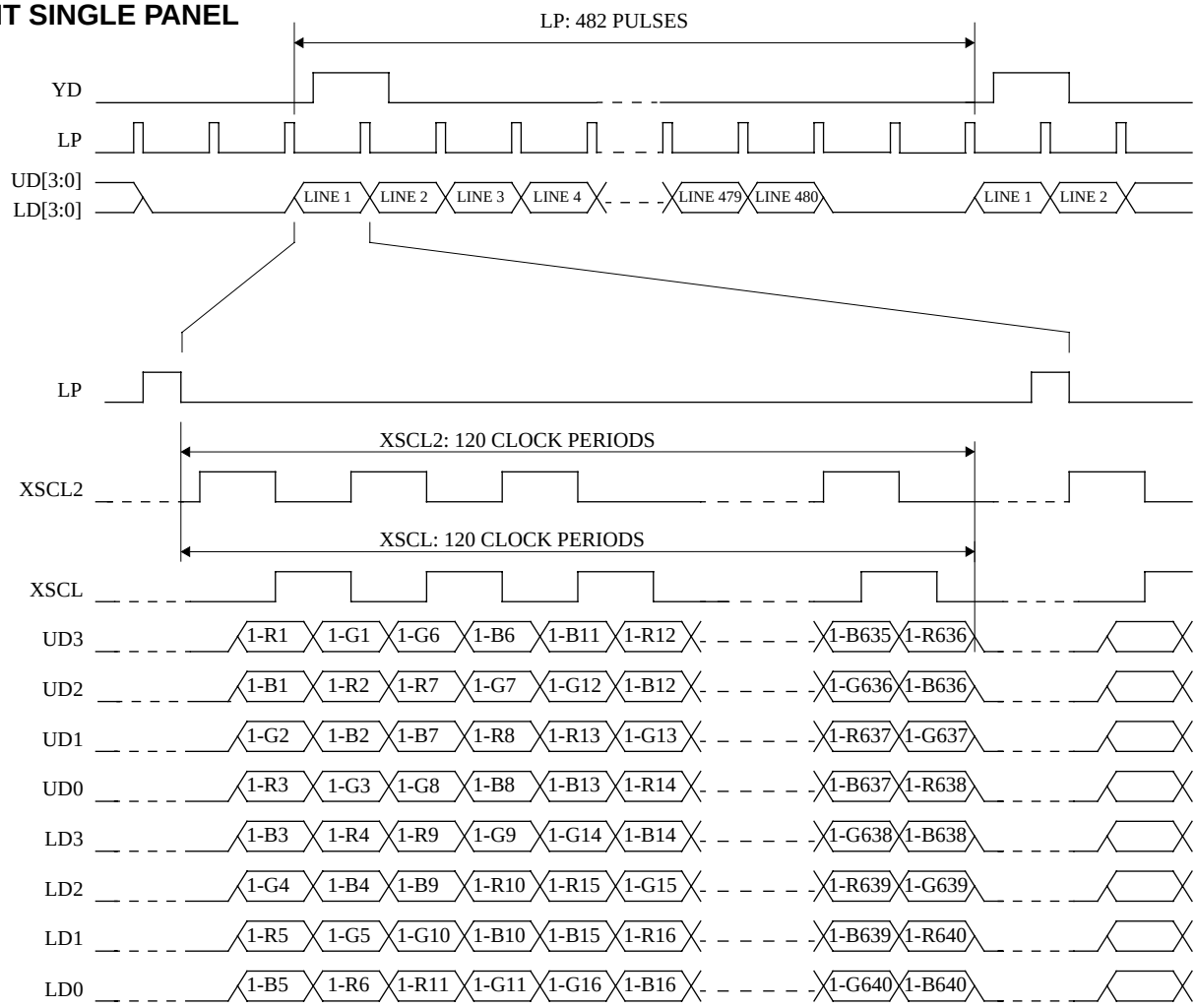
4-BIT SINGLE PANEL



Example timing for a 320 x 240 panel

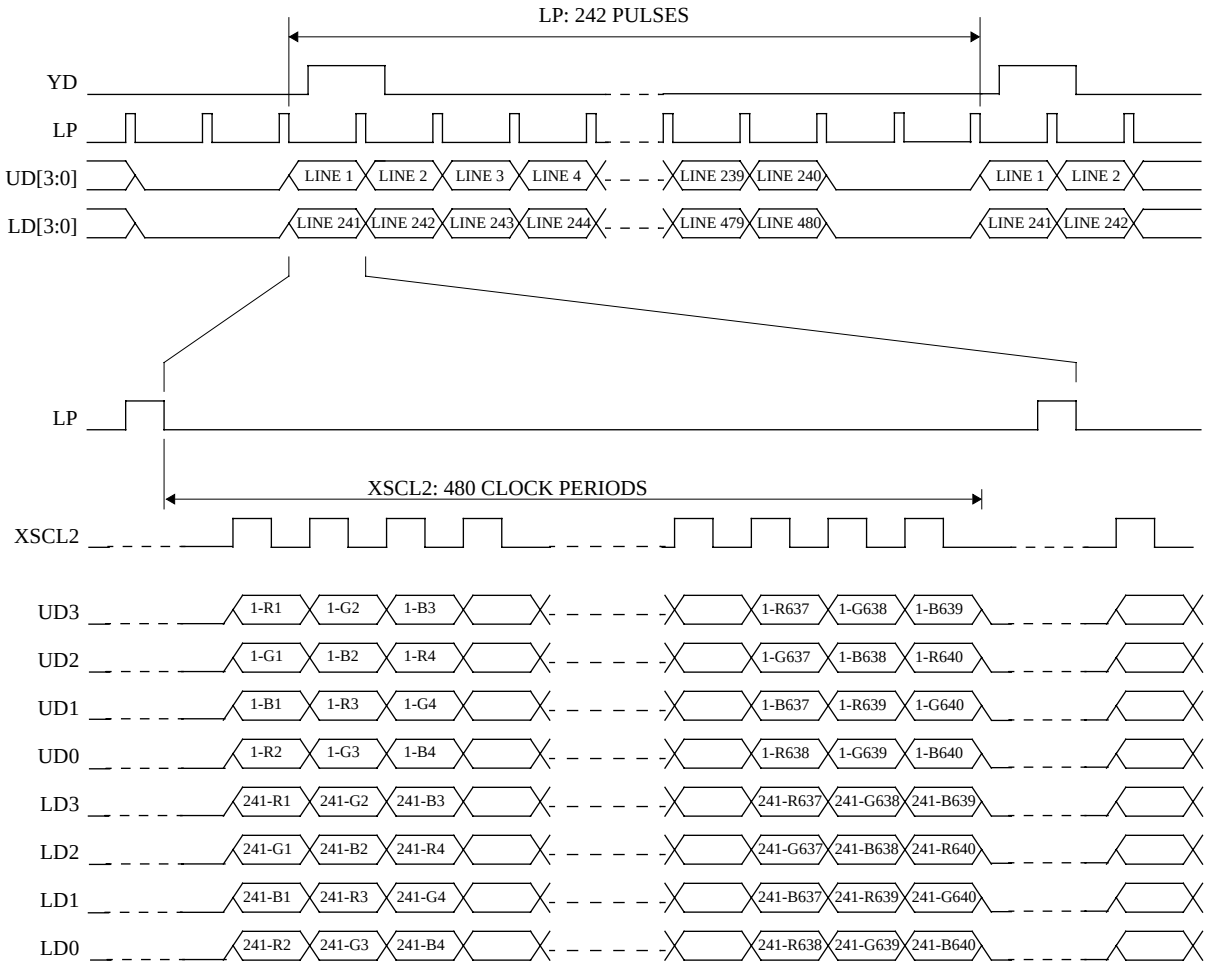
■ COLOR STN LCD PANEL INTERFACE

8-BIT SINGLE PANEL



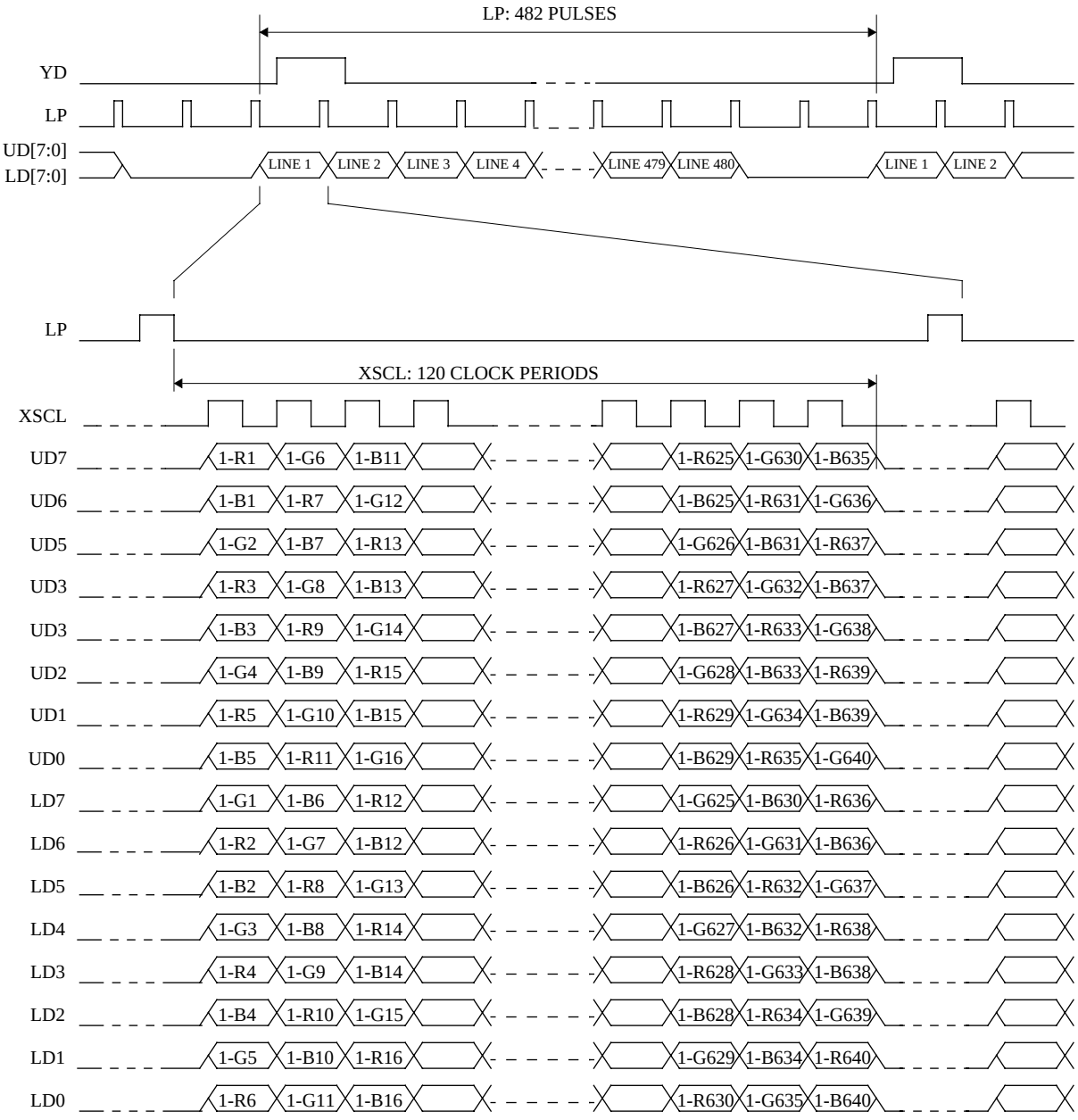
■ COLOR STN LCD PANEL INTERFACE

8-BIT DUAL PANEL



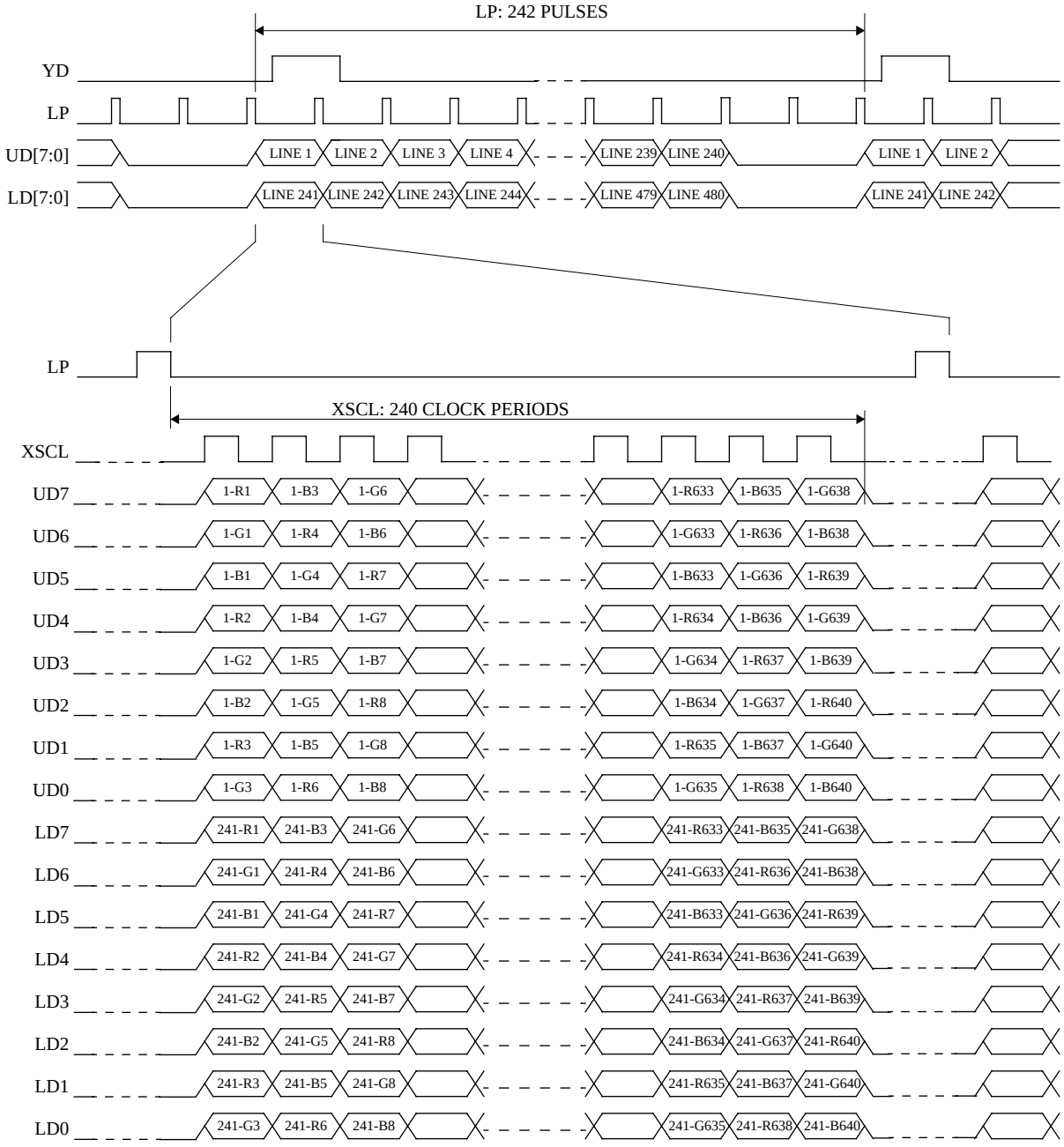
■ COLOR STN LCD PANEL INTERFACE

16-BIT SINGLE PANEL



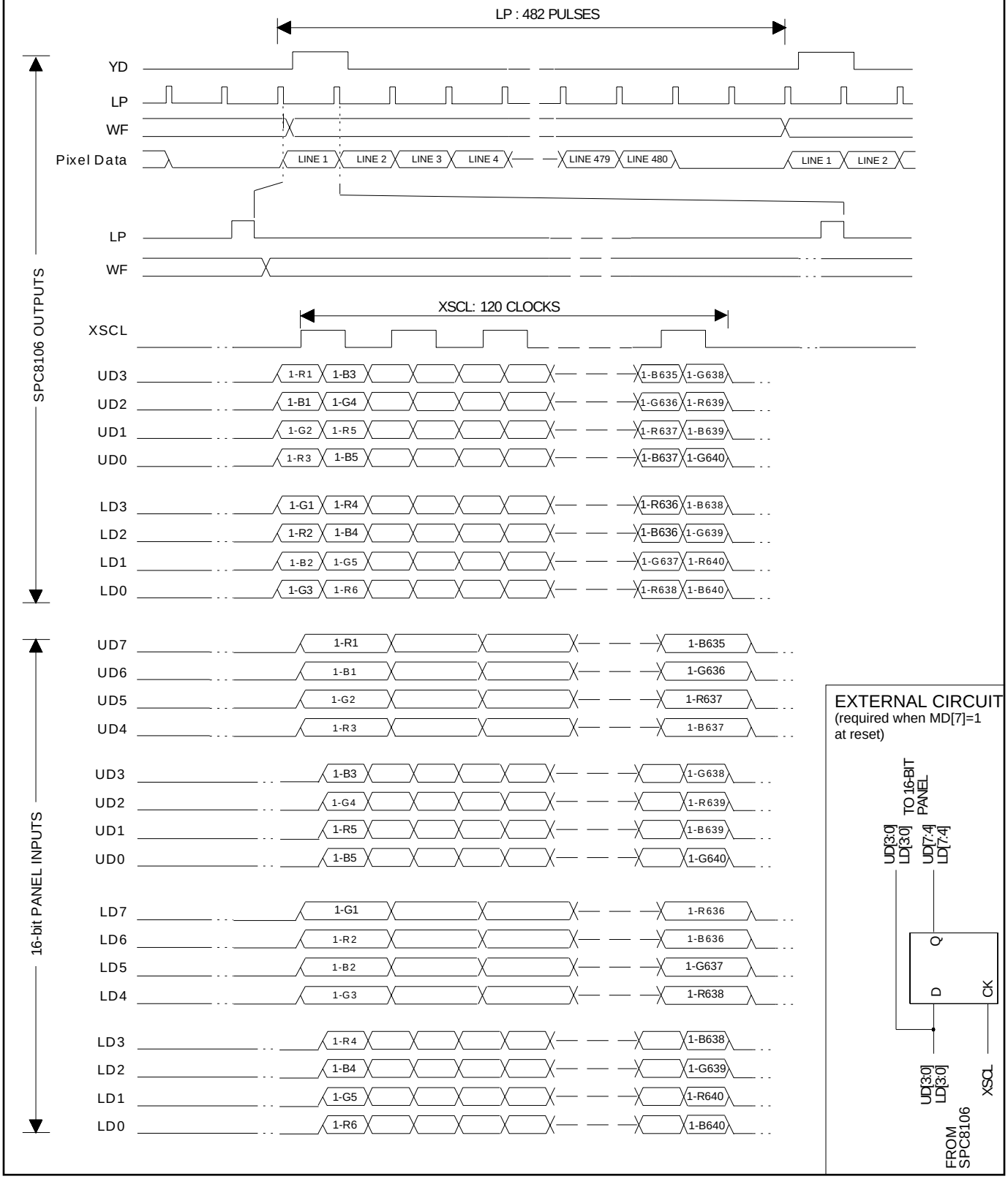
■ COLOR STN LCD PANEL INTERFACE

16-BIT DUAL PANEL



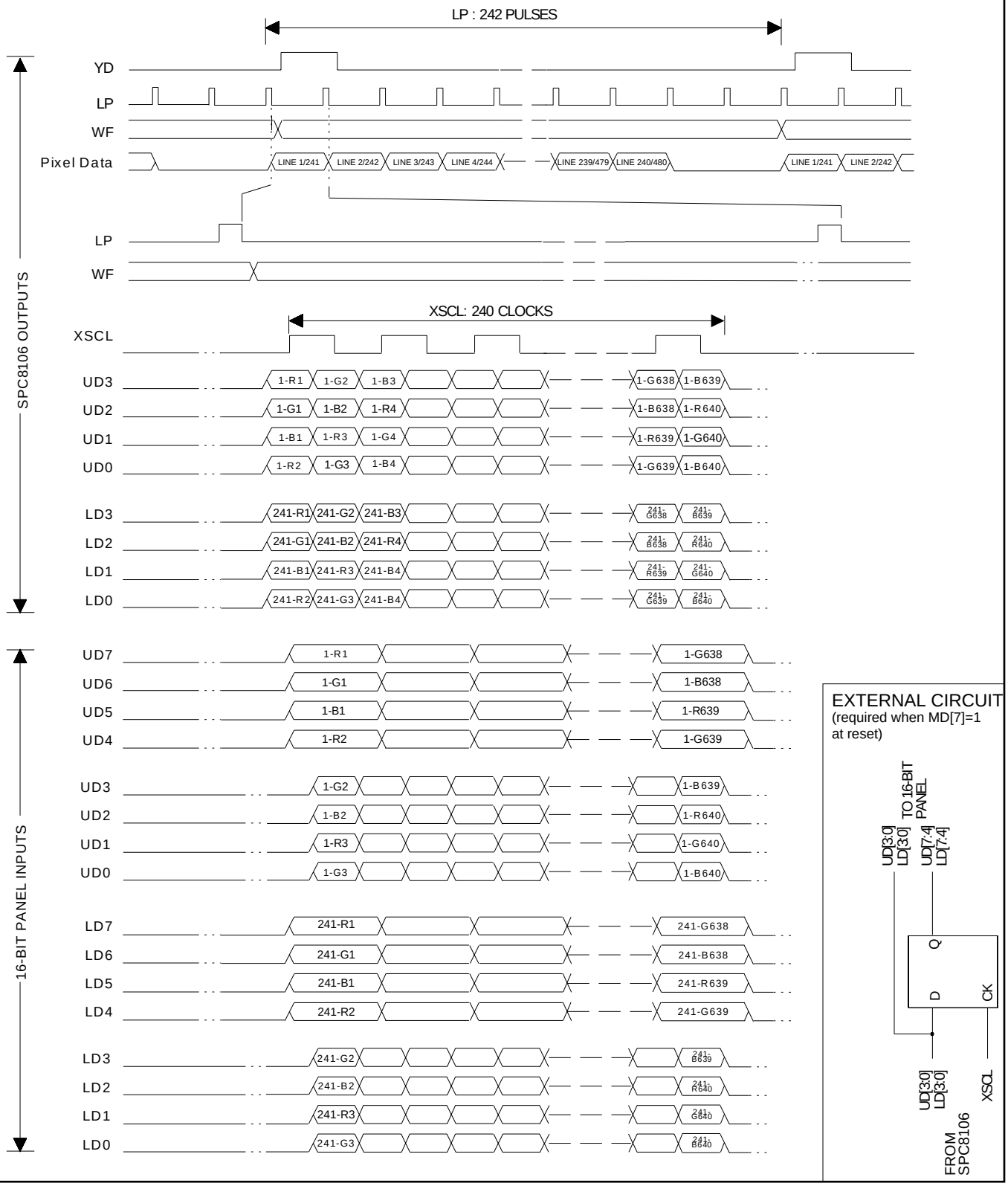
■ COLOR STN LCD PANEL INTERFACE

16-BIT SINGLE PANEL WITH EXTERNAL CIRCUIT



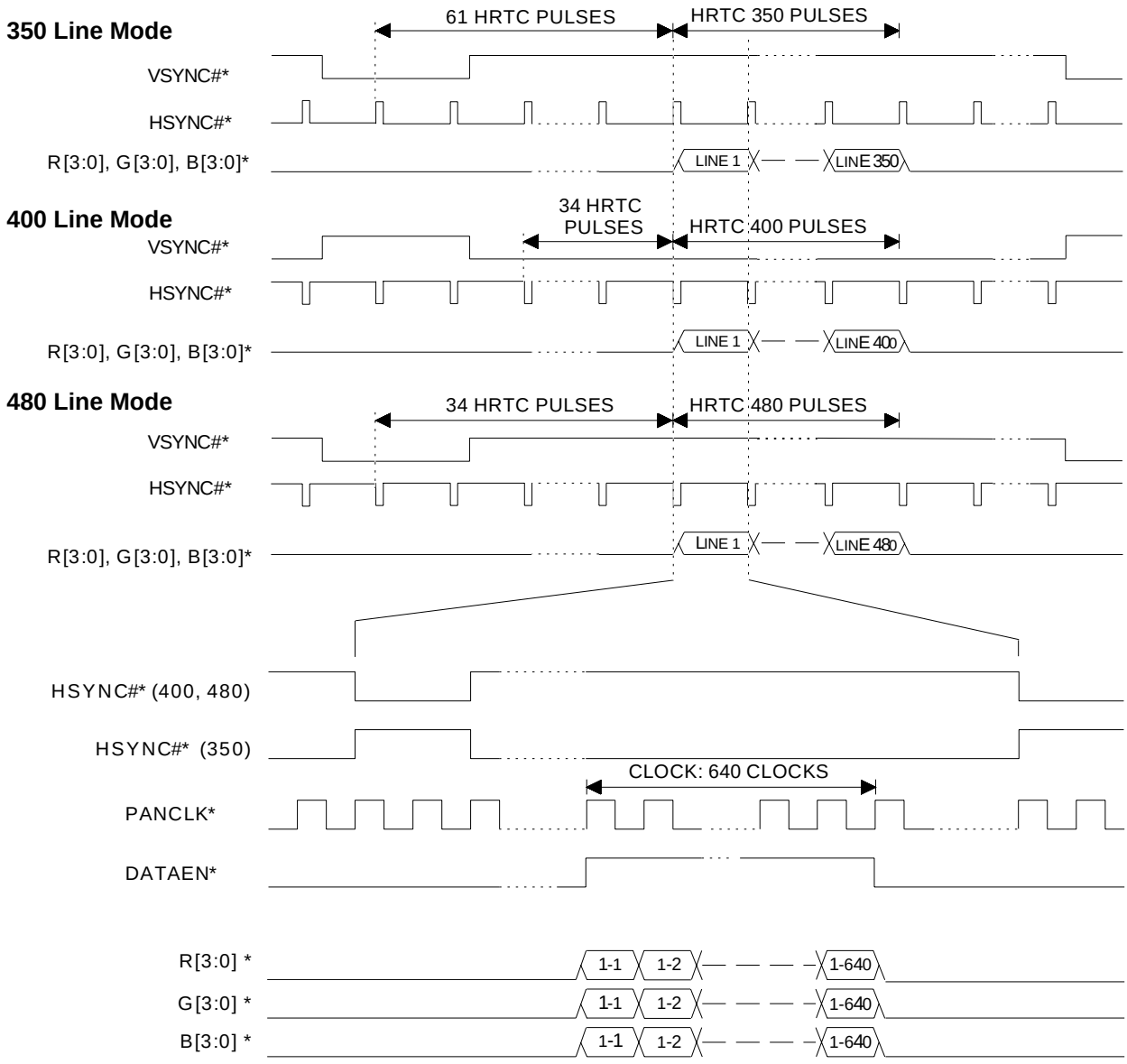
■ COLOR STN LCD PANEL INTERFACE

16-BIT DUAL PANEL WITH EXTERNAL CIRCUIT



■ COLOR TFT PANEL INTERFACE

Auxiliary Register [00] bit 5=1 and Auxiliary Register [0B] bit 1=1

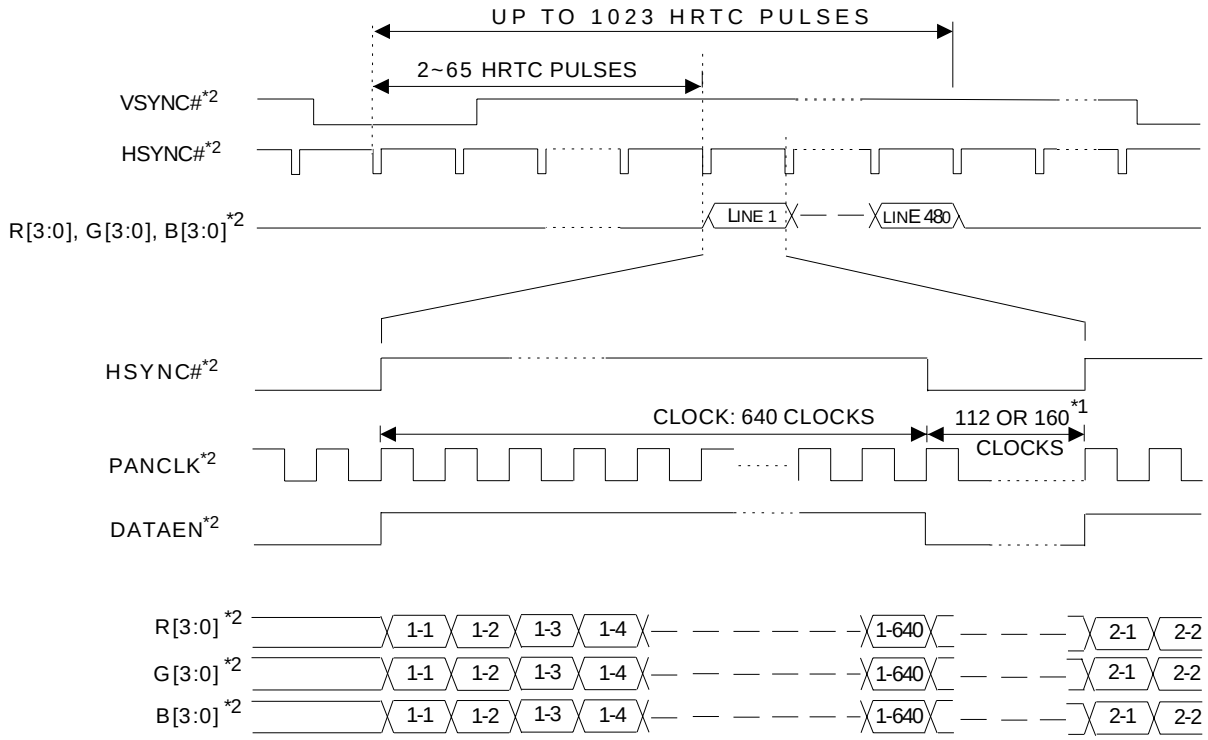


9-bit panels use data bits [2:0]

* Refer to “Pin Mapping for Various Display Modes” on page 15 for actual pin names

■ COLOR TFT PANEL INTERFACE

Auxiliary Register [00] bit 5=1 and Auxiliary Register [0B] bit 1=0



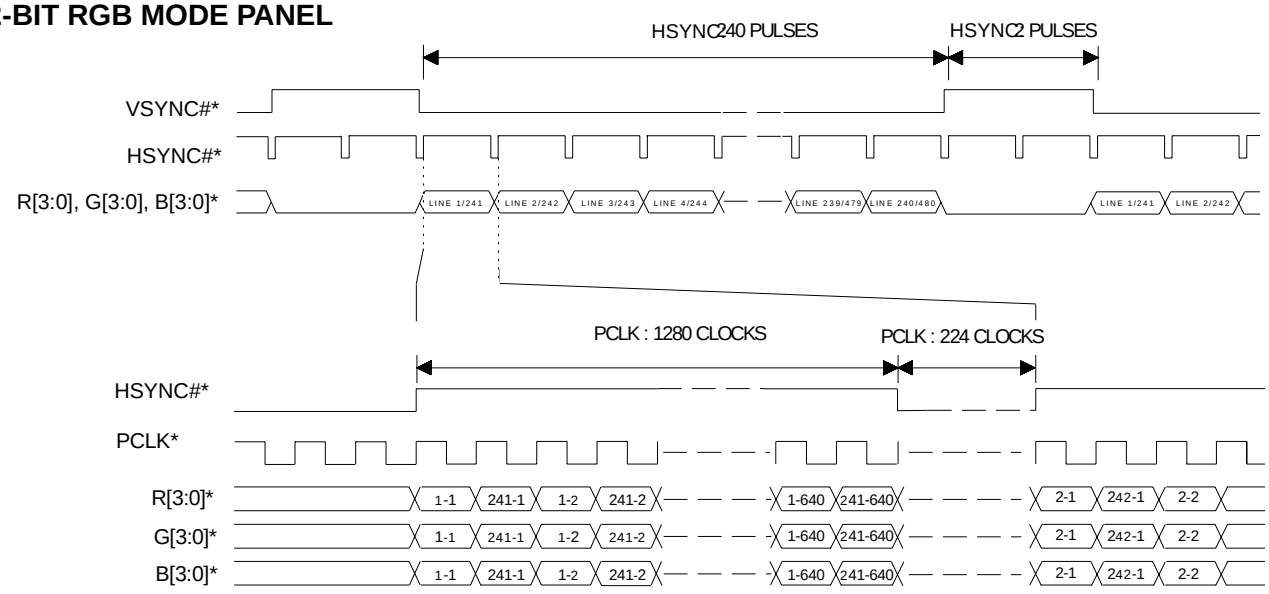
9-bit panels use data bits [2:0]

*1 - This number is controlled by Auxiliary Register [06] bit 2

*2 - Refer to “Pin Mapping for Various Display Modes” on page 15 for actual pin names

■ RGB MODE PANEL INTERFACE

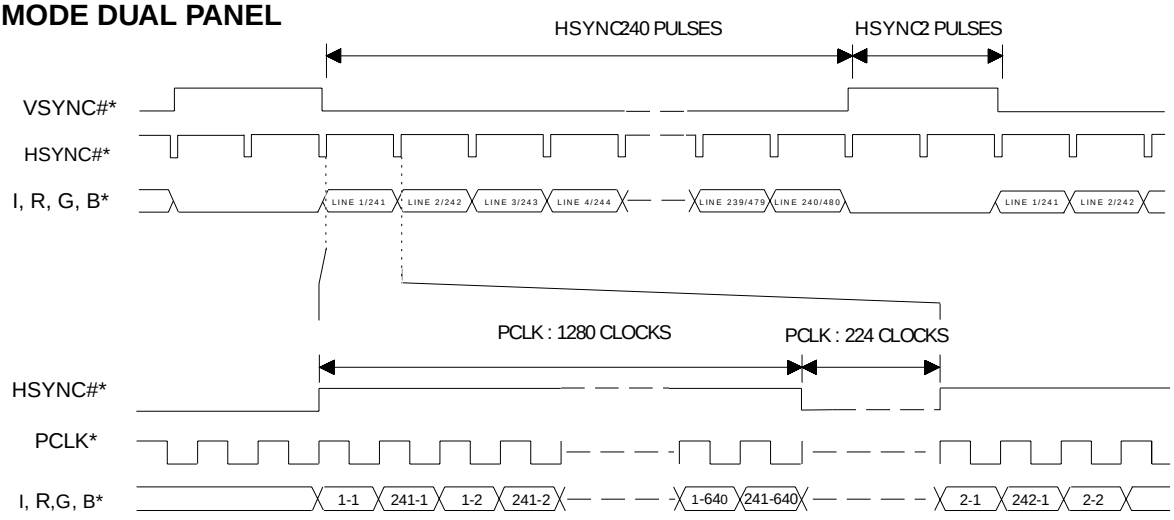
12-BIT RGB MODE PANEL



* Refer to “Pin Mapping for Various Display Modes” on page 15 for actual pin names

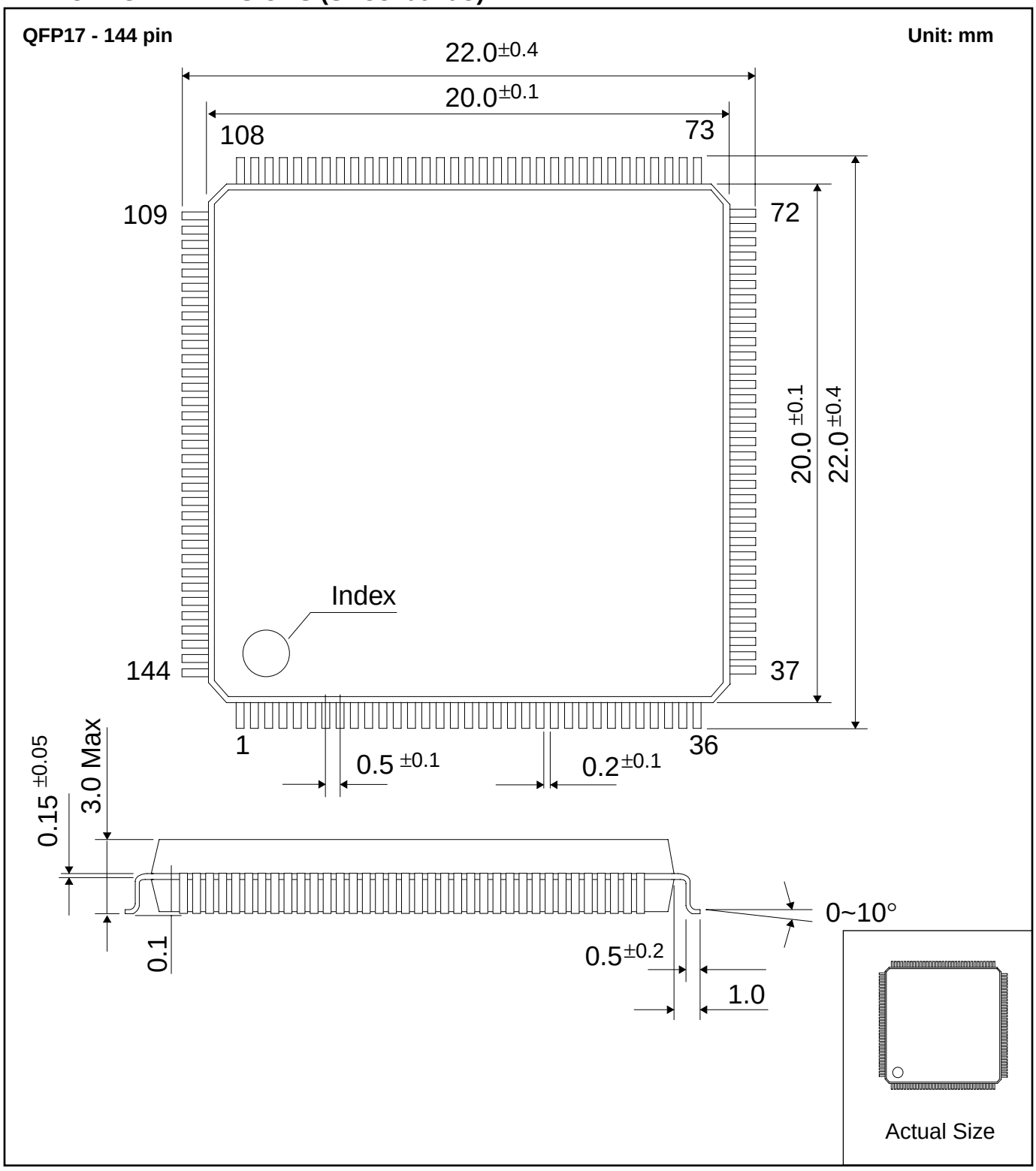
■ RGBI MODE DUAL PANEL INTERFACE

RGBI MODE DUAL PANEL



* Refer to “Pin Mapping for Various Display Modes” on page 15 for actual pin names

■ PACKAGE DIMENSIONS (SPC8106F0C)



S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

Hardware Functional Specification

Drawing Office No. X12-SP-001-07

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

THIS PAGE INTENTIONALLY BLANK

1.0 INTRODUCTION

1.1 Scope

This is the Functional Specification for the SPC8106 LCD VGA Controller Chip. Included in this document are timing diagrams, AC and DC characteristics, register descriptions, and power management descriptions. This document is intended for two audiences; Video Subsystem Designers and Software Developers.

2.0 FEATURES

2.1 Technology

- low power CMOS
- 3.3 and 5 volt operation
  - 5 V core and I/O operation
  - 3.3 V core and 5 V I/O operation
  - 3.3 V core and I/O operation
- 144 pin QFP17 surface mount package

2.2 System

- 8- or 16-bit ISA CPU data bus interface
- programmable 64x64x2-bit hardware sprite, or 64x64x2-bit hardware cursor
- two 2-terminal crystal inputs for internal oscillators, maximum 28.322 MHz frequency. External crystal oscillators also supported
- interfaces to a single 256Kx16 DRAM (70 ns for 3.3 V I/O and Core, 80 ns for 5 V I/O and Core, 1024x256x16 or 512x512x16).
  Access to full 512Kbytes of video memory allowed
- selectable DRAM interface configurations: 2 CAS/1 WE, or 1 CAS/2 WE
- selectable 256 cycle/4 ms or 256 cycle/32 ms DRAM refresh rate, or low power self-refresh mode (for DRAMs supporting self-refresh)
- 32 kHz 50% duty cycle power down clock support during Power Save Mode 4 and Suspend mode
- one hardware plus six software initiated power save modes
- ability to place external RAMDAC in power save mode
- low power consumption.
- 3C3h and 46E8h video enable registers supported
- all output pins (except clock interface) can be tri-stated and driven as inputs to allow board level pin testing

2.3 Compatibility

- support for all standard VGA video modes on LCD or CRT
- support for some extended VESA video modes
- external VGA RAMDAC (Bt477 or equivalent) required for CRT modes
- proprietary internal 256x6 gray scale lookup table provided for monochrome LCD modes
- proprietary internal 256x12 color lookup table provided for color LCD modes
- programmable hardware mapping of VGA palette-style writes to 64 level LCD gray scale values for monochrome LCD modes
- 64 gray shades on monochrome LCD in mode 13h (16 gray shades by FRM + dithering)
- hardware vertical expansion of 400 line graphics and text modes on LCD
- vertical interrupt function on IRQ pin supported

2.4 Display Support

- supports 8-bit 640x480 single panel-single drive and dual panel-dual drive monochrome and color LCD displays
- supports 16-bit 640x480 single panel-single drive and dual panel-dual drive color LCD displays
- supports 4-bit single panel-single drive monochrome and color LCD displays
- additional LCD panel sizes supported via programmable horizontal and vertical panel size configuration registers
- supports color 9/12-bit TFT displays, RGBI and 12-bit RGB mode displays
- analog CRT monitors supported for standard VGA modes with an external RAMDAC
- DoubleScan mode—simultaneous display of CRT and single panel LCD

3.0 OVERVIEW DESCRIPTION

The SPC8106 is a 3.3/5 V LCD 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 256 x 6 bit gray scale lookup table is provided to allow remapping of the 64 possible gray shades displayed on a monochrome LCD panel. For color LCD modes, an internal 256x12 bit VGA-style lookup table is provided (4 bits each of R, G, and B). An interface to an external RAMDAC is also provided to allow connecting a standard VGA monitor to the system.

The target markets for this device are small, cost sensitive mixed voltage 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, or 64 levels of gray in mode 13h on a monochrome LCD display, or up to 256 colors out of a palette of 4096 colors on a color LCD display. With an external RAMDAC, standard VGA modes are supported on a CRT display.

3.1 Typical System Implementation Diagram

The following figure shows a typical system implementation with SPC8106.

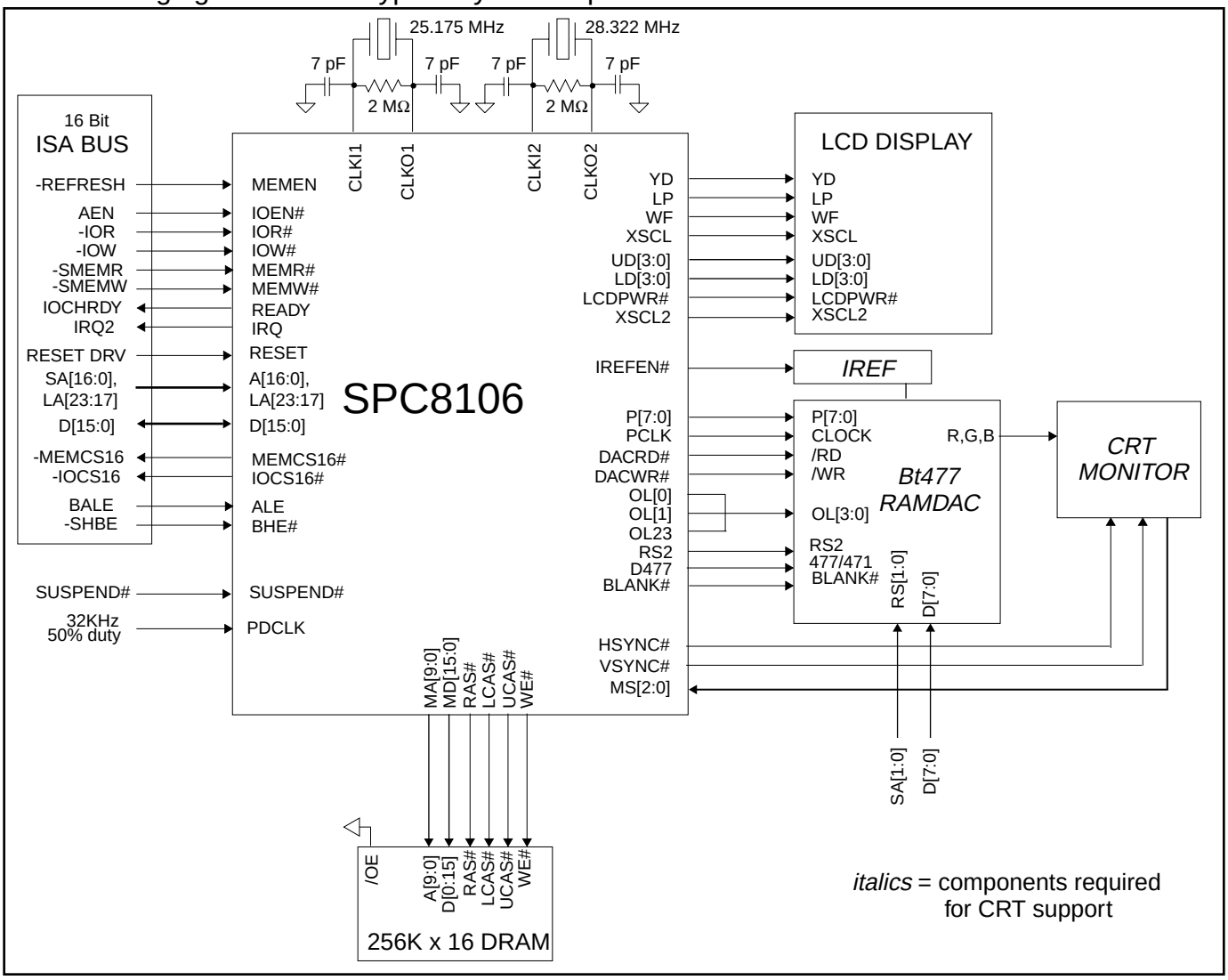


Figure 1 : Typical System Implementation Diagram Source: 8106f0c_sys_blk_01.can

3.2 Internal Block Diagram

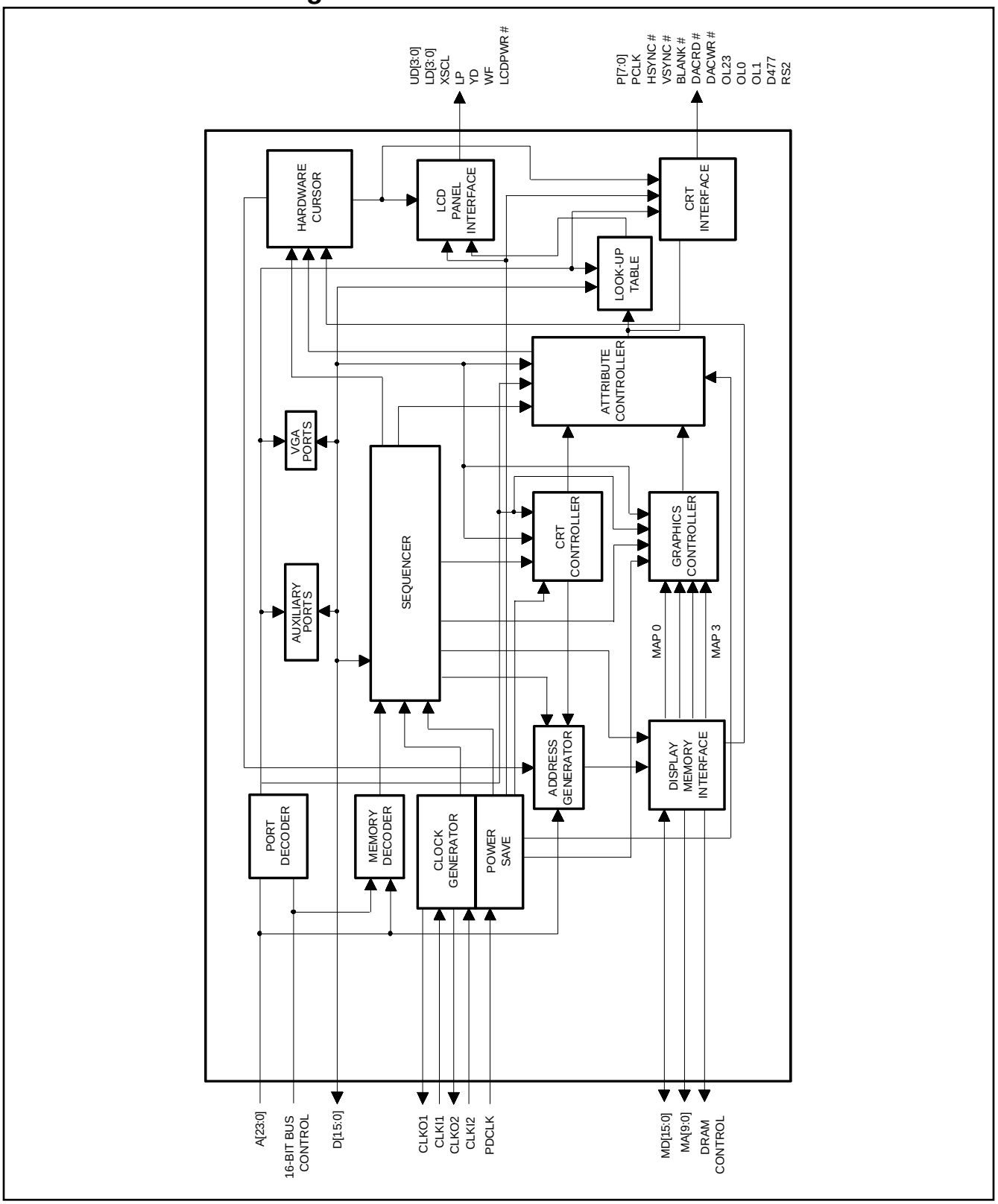


Figure 2 : Typical System Block Diagram Source: 8108fblk.drw

4.0 PINOUT DIAGRAM

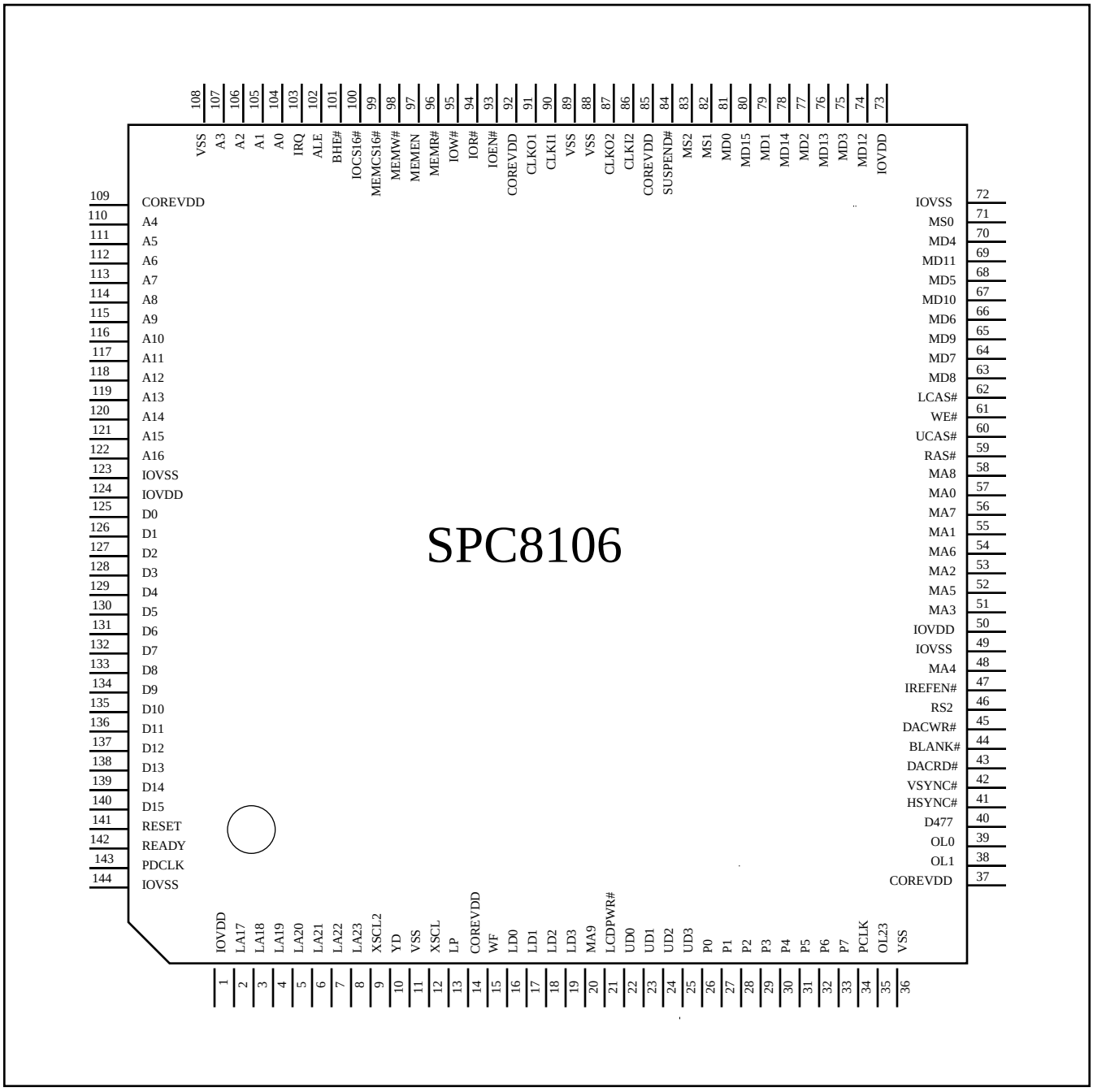


Figure 3 : Pinout Diagram Source: 8106f0c_po_02.can

Some pins have dual use - default configuration is shown. See “Multiple Function Pin Descriptions” on page 23 for details.

Package type: 144 pin surface mount QFP17

5.0 PIN DESCRIPTION

Key

C = CMOS level input
TTL = TTL level input
TTLS = TTL level input with hysteresis
TSx = Tri-state CMOS level driver, x denotes driver type - see “D.C. CHARACTERISTICS” on page 26 for rating.
TSxUy = Tri-state CMOS level driver with pull up resistor (y=2: 100 kΩ typical., y=3: 200 kΩ typ.), x denotes driver type - see “D.C. CHARACTERISTICS” on page 26 for rating.
TSxD = Tri-state CMOS level driver with pull down resistor (200 kΩ typ.), x denotes driver type - see “D.C. CHARACTERISTICS” on page 26 for rating.

pins marked with a ***** in the Type column are outputs in normal operation mode, but for pin test mode, these outputs are placed in a high impedance state and these pins become inputs. Therefore these pins are actually bidirectional, although only the normal output mode is shown in this table. For these pins, the input type for this test mode is shown in parentheses (*). See “PIN TEST MODE” on page 124 for more information.

Table 0-1 CPU Interface Pins

Pin Name	Type	Pin #	Drv	Description
A[0:16], LA[17:23]	I	104~107, 110~122, 2~4, 5~8	TTL	CPU bus address inputs. In Suspend Mode, the Address inputs are internally masked off. If the value on MD[5] at RESET = 1, then the ALE input pin is used to internally latch LA[19:17] and A[16:2], allowing these address bits to be driven by the processor address bus. If the value on MD[5] at RESET = 0, then standard ISA address timing is assumed, where pins A[0:16], LA[17:23] should be connected to the ISA bus signals SA[0:16], LA[17:23] respectively.
ALE	I	102	TTL	ISA Bus Address Latch Enable. In Suspend Mode the ALE input is disabled. If the value on MD[5] at RESET = 1, then the ALE input is used to internally latch LA[19:17] and A[16:2], allowing these address bits to be driven by the processor address bus. In this mode, the processor ADS# output should be connected to this pin. If the value on MD[5] at RESET = 0, then standard ISA address timing is assumed, and only the LA[19:17] inputs are internally latched.
D[0:15]	I/O	125~140	TTL /TS2	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.
MEMEN	I	97	TTLS	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	TTLS	ISA Bus I/O Read Strobe. In Suspend Mode the IOR# input is disabled.
IOW#	I	95	TTLS	ISA Bus I/O Write Strobe. In Suspend Mode the IOW# input is disabled.
MEMR#	I	96	TTLS	ISA Bus System Memory Read Strobe. In Suspend Mode the MEMR# input is disabled.

Table 0-1 CPU Interface Pins (Continued)

Pin Name	Type	Pin #	Drv	Description
MEMW#	I	98	TTLS	ISA Bus System Memory Write Strobe. In Suspend Mode the MEMW# input is disabled.
IOEN#	I	93	TTLS	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 IOEN# input is disabled.
READY	O *	142	TS3 (* C)	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	141	TTLS	The active high Reset signal from the CPU clears all internal registers and forces all signals to their inactive state.
IRQ	O	103	TS3	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). This pin also is used for the output of the NAND tree in pin test mode.
MEMCS16#	O *	99	TS4 (* C)	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 *	100	TS4 (* C)	ISA Bus I/O Chip Select 16. Address inputs A[15:0] and IOEN# are decoded to drive this output low when a valid SPC8106 I/O register address appears on the bus,. Note that I/O addresses 3C6h-3C9h do not result in IOCS16# being driven low (i.e. RAMDAC and internal LUT register reads and writes are 8 bit cycles).
BHE#	I	101	TTL	ISA Bus Byte High Enable. In Suspend Mode the BHE# input is disabled.

Table 0-2 Video Memory Interface Pins

Pin Name	Type	Pin #	Drv	Description
MA[0:9]	O *	57, 55, 53, 51, 48, 52, 54, 56, 58, 20	TS2 (* C)	Multiplexed row/column address bits for video display memory.
MD[0:15]	I/O	81, 79, 77, 75, 70, 68, 66, 64, 63, 65, 67, 69, 74, 76, 78, 80	TTL/ TS2U2	Data bits for video display memory. The output drivers of these pins are placed into a high-impedance state when RESET is high, or when the Sequencer is in a reset state. On the falling edge of RESET, the values on MD[3:0] and MD[12:9] are latched into a read-only Auxiliary Register and are available to be read as configuration inputs. Also, the value on MD[8:4] and MD[15:13] are used to configure various hardware options. See “Power On / Reset Options” on page 22 for details.

Table 0-2 Video Memory Interface Pins (Continued)

Pin Name	Type	Pin #	Drv	Description
RAS#	O *	59	TS3 (* C)	DRAM Row Address Strobe for single 256Kx16 DRAM.
LCAS# (LWE#)	O *	62	TS3 (* C)	Multiple Function: DRAM Column Address Strobe for low byte (LCAS#). For alternate function see “Multiple Function Pin Descriptions” on page 23.
UCAS# (CAS#)	O *	60	TS3 (* C)	Multiple Function: DRAM Column Address Strobe for high byte (UCAS#). For alternate function see “Multiple Function Pin Descriptions” on page 23.
WE# (UWE#)	O *	61	TS3 (* C)	Multiple Function: DRAM Write Enable Strobe (WE#). For alternate function see “Multiple Function Pin Descriptions” on page 23.

Table 0-3 Clock Input Pins

Pin Name	Type	Pin #	Drv	Description
CLKI1	I	90	C	This pin, along with CLKO1 is the 25.175 MHz 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin is the clock input.
CLKO1	O	91	•	This pin, along with CLKI1 is the 25.175 MHz2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin should be left unconnected.
CLKI2	I	86	C	This pin, along with CLKO2 is the 28.322 MHz 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin is the clock input.
CLKO2	O	87	•	This pin, along with CLKI2 is the 28.322 MHz 2-terminal crystal interface when using a 2-terminal crystal as the clock input. If an external oscillator is used as a clock source, then this pin should be left unconnected.

Table 0-4 LCD Panel Interface Pins [a]

Pin Name	Type	Pin #	Drv	Description
YD	O *	10	TS4 (* C)	Vertical Scanning Start Pulse output. A logic 1 on this signal, sampled by the LCD 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 *	13	TS4 (* C)	Latch Pulse output. The falling edge of this signal is used to latch a row of display data in the LCD module’s column driver shift registers and to turn on the row driver (Y driver) for that line.
XSCL	O *	12	TS4 (* C)	Shift Clock for LCD data. Display data is clocked out of the chip on the rising edge of this signal, to be shifted into the LCD panel module column drivers (X drivers) on each falling edge.

Table 0-4 LCD Panel Interface Pins [a] (Continued)

Pin Name	Type	Pin #	Drv	Description
XSCL2	O *	9	TS4 (* C)	This second shift clock is used together with XSCL in 8-bit single color panel mode to shift in alternate sets of display data. XSCL2 is also used alone as the shift clock in 8-bit dual color panel mode and 4-bit single color panel mode.
UD[0:3]	O *	22~25	TS4 (* C)	Upper panel display data for dual panel - dual drive mode. For 8-bit single panel-single drive mode, these bits are the most significant 4-bits of the 8-bit output data to the panel (data[7:4]). For 4-bit single panel mode, these bits are the 4 bits of data output to the panel. For 16-bit LCD modes, these outputs are the multiplexed upper panel data if MD[7]=1 at RESET, or the lower nibble of the upper panel data if MD[7]=0 at RESET.
UD[4:7]	O *	26~29	TS2D (* C)	When MD[7]=0 at RESET, these pins are the upper nibble of the 16-bit LCD mode upper panel data.
LD[0:3]	O *	16~19	TS4 (* C)	Lower panel display data for dual panel-dual drive mode. For 8-bit single panel-single drive mode, these bits are the least significant 4 bits of the 8-bit output data to the panel (data[3:0]). For 4-bit single panel mode, these outputs are driven low. For 16-bit LCD modes, these outputs are the multiplexed lower panel data if MD[7]=1 at RESET, or the lower nibble of the lower panel data if MD[7]=0 at RESET.
LD[4:7]	O *	30~33	TS2D (* TTL)	When MD[7]=0 at RESET, these pins are the upper nibble of the 16-bit LCD mode lower panel data.
LCDPWR#	O *	21	TS2 (* C)	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, when Auxiliary Register 06 bit 0 is set to 1, or when 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 *	15	TS4D (* C)	LCD Backplane Bias signal. This output toggles once every n LP periods, as programmed in AUX[0D].

a. some of these pins have alternate uses in some display modes. See “Pin Mapping for Various DRAM Configurations” on page 25

Table 0-5 External CRT/RAMDAC Interface Pins

Pin Name	Type	Pin #	Drv	Description
P[0:7]	O *	26~33	TS2D (*TTL)	When MD[7]=1 at RESET, these pins are the Pixel Data outputs. These 8 bits are connected to the pixel select inputs of the external RAMDAC.
PCLK	O *	34	TS2D (* C)	Pixel Clock. Pixel data is clocked out of the chip on the falling edge of PCLK.
BLANK#	O *	44	TS2D (* C)	Blank output. This output is clocked out on the falling edge of PCLK and is driven low during display blanking periods.
HSYNC#	O *	41	TS4D (* C)	Horizontal Sync. This output is clocked out on the falling edge of PCLK and is driven to indicate the horizontal retrace period. The polarity of this signal is determined by a control bit in register 3C2h.

Table 0-5 External CRT/RAMDAC Interface Pins (Continued)

Pin Name	Type	Pin #	Drv	Description
VSYNC#	O *	42	TS4D (* C)	Vertical Sync. This output is clocked out on the falling edge of PCLK and is driven to indicate the vertical retrace period. The polarity of this signal is determined by a control bit in register 3C2h.
DACRD#	O *	43	TS3 (* C)	RAMDAC Read Strobe. This signal goes low when a valid read access to the VGA RAMDAC is decoded by the chip.
DACWR#	O *	45	TS3 (* C)	RAMDAC Write Strobe. This signal goes low when a valid write access to the VGA RAMDAC is decoded by the chip.
RS2	O *	46	TS2D (* C)	Register Select 2 output. This output should be connected to the RS2 input of the RAMDAC (Bt477 or equivalent). The logic level on this output may be set by setting AUX[0B] bit 3. This signal is required to allow CPU access the control and overlay registers of the external RAMDAC.
OL[0:1]	I/O	39, 38	OL0 C/ TS2 OL1 TTLS/ TS2	Multiple Function: Overlay Select outputs 1:0 When MD[13]=0 at RESET, these pins are outputs used to provide sprite/HW cursor function on the CRT display. In this case, these outputs should be connected to the OL[0:1] inputs of the RAMDAC (Bt477 or equivalent). They are used by the sprite circuitry to access the overlay registers in the RAMDAC. For alternate function see “Multiple Function Pin Descriptions” on page 23.
OL23	O *	35	TS2D (* C)	Overlay Select output 2/3. This output should be connected to both the OL2 and OL3 inputs of the RAMDAC (Bt477 or equivalent). This signal is used by the sprite circuitry to access the overlay registers in the RAMDAC. For alternate function see “Multiple Function Pin Descriptions” on page 23.
D477	O *	40	TS2 (* C)	477 Control Signal. This output should be connected to the 477/471 input of the RAMDAC (Bt477 or equivalent). This signal is used to access the control register of the RAMDAC and to allow it to be powered down. The logic level on this output can be controlled by setting AUX[0B] bit 4, and is also controlled by the power save logic.
IREFEN#	O *	47	TS2U3 (* C)	IREF Enable output. This signal is used to control the external current reference source required by the RAMDAC, allowing powering down the analog circuitry when not required. When this signal is driven low, the external current reference should be enabled. When this signal is high, the external current reference should be shut off.
MS[2:0]	I/O	83, 82, 71	TTL/ TTL2	Monitor Sense inputs. These signals should be connected to the monitor sense lines from the CRT monitor cable. The status of these bits is readable in Auxiliary register [08] bits 2:0, and is used by BIOS software to determine the presence and type of monitor connected. Optionally, the SENSE output of the RAMDAC may be connected to one of these inputs to allow the BIOS to read the SENSE signal and detect the monitor. These pins can be forced low by the DCC2 monitor support bits in AUX[10h] bits 1:0.

Table 0-6 Power Save Mode Control Pins

Pin Name	Type	Pin #	Drv	Description
SUSPEND#	I	84	TTLS	A low level on this pin puts the chip into a hardware power down mode. The SUSPEND# signal overrides any software initiated power down modes, and disables the ISA-Bus interface inputs except RESET. Address and Data inputs are also masked when this signal is low. When in Suspend Mode the UD[3:0], LD[3:0], XSCL, XSCL2, LP, YD and WF signals are driven into a high impedance or low state (configurable) and the LCDPWR# signal is driven high.
PDCLK	I	143	TTL	Power Down Clock. This input may be used to provide a low frequency clock for generating refresh in Power Save Modes 4 and Suspend, as an optional alternative to using the pixel clock or MEMEN input as the refresh clock source. This clock input should be driven by either by a 32 kHz 50% duty cycle clock source, or a 64 kHz clock source with a high period as short as possible (but > minimum RAS low pulse width) to minimize DRAM current consumption during refresh. The PDCLK input is used to directly generate the RAS and CAS pulses during Power Save Mode 4 and Suspend. Refer to “Implementation Notes” on page 80, for details.

Table 0-7 Power Supply Pins

Pin Name	Type	Pin #	Description
COREVDD	P	14, 37, 85, 92, 109	V_{DD} supply for core logic.
IOVDD	P	1, 50, 73, 124	V_{DD} supply for interface pins.
VSS	P	11, 36, 88, 89, 108	V_{SS} supply for core logic.
IOVSS	P	49, 72, 123, 144	V_{SS} supply for interface pins.

5.1 Configuration Options

Mixed Voltage Configurations

Table 0-8 Mixed Voltage Configuration

Core VDD	I/O VDD	
	3.3 V	5.0 V
3.3 V	OK	OK
5.0 V	NO	OK

5.2 Power On / Reset Options

Table 0-9 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[4]	16-bit ISA interface (1) / 8-bit ISA interface (0)
MD[5]	A[19:2] latched internally by ALE (1) / standard ISA bus ALE - A[16:0] not latched (0)
MD[6]	2 CAS, 1 WE type DRAM (1) / 1 CAS, 2 WE type DRAM (0)
MD[7]	support 16-bit panel with external logic (1) / support 16-bit panel directly (0)
MD[8]	5 V core operating voltage (1) / 3.3 V core operating voltage (0)
MD[12:9]	values latched into read-only bits 7:4 of AUX[0C] for software use
MD[13]	pins 38, 39 used for ext. RC for 32 kHz PDCLK (1) / pins 38, 39 used for OL[1:0] (0)
MD[14]	Internal PDCLK doubling disable (1) / enable (0)
MD[15]	3C3h used as video enable port (1) / 46E8h and 102h used as video enable port (0)

These inputs 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 (V_{DD} = 5.0V).

5.3 Multiple Function Pin Descriptions

Table 0-10 Multiple Function Pin Descriptions

Pin Name	Function	MD Line Status	Functional Description
LCAS#, LWE#	LCAS#	MD6 = 1	DRAM Column Address Strobe (Low Byte)
	LWE#	MD6 = 0	DRAM Write Strobe (Low Byte)
UCAS#, CAS#	UCAS#	MD6 = 1	DRAM Column Address Strobe (High Byte)
	CAS#	MD6 = 0	DRAM Column Address Strobe
WE#, UWE#	WE#	MD[6] = 1	DRAM Write Strobe
	UWE#	MD[6] = 0	DRAM Write Strobe (High Byte)
OL0, P320, B0	OL0	MD[13] = 0 AUX[00] b6 = 0	Overlay Bit 0. Used for CRT HW Cursor/ Sprite support.
	P320	MD[13] = 1 MD[14] = 1	32 kHz Clock Output. Used with external RC when using external PDCLK support
	B0	MD[13] = 0 AUX[00] b6 = 1	Data bit B0 for 12-bit TFT support
OL1, P32I, G0	OL1	MD[13] = 0 AUX[00] b6 = 0	Overlay Bit 1. Used for CRT HW Cursor/ Sprite support
	P32I	MD[13] = 1 MD[14] = 1	32 kHz Clock Input. Used with external RC when using external PDCLK support
	G0	MD[13] = 0 AUX[00] b6 = 1	Data bit G0 for 12-bit TFT support
OL23, R0	OL23	MD[13] = 0 AUX[00] b6 = 0	Overlay Bit 2. Used for CRT HW Cursor/ Sprite support
	R0	MD[13] = 0 AUX[00] b6 = 1	Data bit R0 for 12-bit TFT support
P[0:3]	P[0:3}	MD[7] = 1	Lower nibble of the CRT pixel data outputs
	UD[4:7}	MD[7] = 0	Upper nibble of the 16-bit LCD mode upper panel data
P[4:7]	P[4:7]	MD[7] = 1	Upper nibble of the CRT pixel data outputs
	LD[4:7]	MD[7] = 0	Upper nibble of the 16-bit LCD mode lower panel data

5.4 Pin Mapping for Various Display Modes

The pin functions for the various SPC8106 display options are given in the following table:.

Table 0-11 Display Modes Pin Mapping

SPC8106 Pin Name	Color TFT			Color STN LCD			Mono STN LCD		12-bit RGB	RGBI
	9-bit	12-bit AUX[00]b5=1 AUX[0B]b1=0	12-bit AUX[00]b5=1 AUX[0B]b1=1	16-bit	8-bit	4-bit	8-bit	4-bit		
LD0	R0	R1	R1	LD0	LD0		LD0		R1	D0
LD1	G0	G1	G1	LD1	LD1		LD1		G1	D1
LD2	G1	G2	G2	LD2	LD2		LD2		G2	D2
LD3	G2	G3	G3	LD3	LD3		LD3		G3	D3
LD4				LD4						
LD5				LD5						
LD6				LD6						
LD7				LD7						
UD0	R1	R2	R2	UD0	UD0	UD0	UD0	UD0	R2	
UD1	B0	B1	B1	UD1	UD1	UD1	UD1	UD1	B1	
UD2	B1	B2	B2	UD2	UD2	UD2	UD2	UD2	B2	
UD3	B2	B3	B3	UD3	UD3	UD3	UD3	UD3	B3	
UD4				UD4						
UD5				UD5						
UD6				UD6						
UD7				UD7						
OL0		B0	B0						B0	
OL1		G0	G0						G0	
OL23		R0	R0						R0	
HSYNC#			HSYNC#							
VSYNC#			VSYNC#							
XSCL	PANCLK	PANCLK	PANCLK	XSCL	XSCL		XSCL	XSCL	PCLK	PCLK
XSCL2	R2	R3	R3		XSCL2	XSCL2			R3	
LP	HSYNC#	HSYNC#		LP	LP	LP	LP	LP	HSYNC#	HSYNC#
YD	VSYNC#	VSYNC#		YD	YD	YD	YD	YD	VSYNC#	VSYNC#
WF	DE	DE	DE	WF	WF	WF	WF	WF		
LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#

5.5 Video Memory Interface Options

the logic values 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 has an internal pullup resistor, so no external resistor is required for this default case. Refer to the following table for details.

Table 0-12 Video Memory Interface Options

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

Pin Mapping for Various DRAM Configurations

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

Table 0-13 DRAM Configuration Pin Mapping

Pin #	MD[6] = 1	MD[6] = 0
60	UCAS#	CAS#
61	WE#	UWE#
62	LCAS#	LWE#

Address Mapping for 256Kx16 DRAMs

Two addressing configurations of 256Kx16 DRAMs are supported by the SPC8106:

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 SPC8106 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 SPC8106 (MA[9]) must be connected to the DRAM. For the 9x9 DRAM, only 9 row/column address pins exist, so MA[9] from the SPC8106 is left unconnected.

The following tables summarize the mapping of the SPC8106'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 0-14 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 0-15 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 (*italicized* in the above table).

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

6.0 D.C. CHARACTERISTICS

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

Conditions: HV_{DD} = 5.0V ± 10% T_a = 0°C to 70°C
LV_{DD} = 3.3V ± 10%

Table 0-16 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 10sec max at lead	°C

Table 0-17 Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
HV_{DD}	Supply Voltage	V_{SS} = 0 V	4.5	5.0	5.5	V
LV_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	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 Active Current Consumption	V_{CC} Core = 3.3V	typical I_{Core} = 52.31			mA
		V_{CC} I/O = 5.0V	typical I_{IO} = 13.85			

Table 0-18 Input Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (CMOS inputs)	V_{DD} = MIN			1.0	V
V_{IH}	High Level Input Voltage (CMOS inputs)	V_{DD} = MAX	3.5			V
V_{IL}	Low Level Input Voltage (TTL inputs)	V_{DD} = MIN			0.8	V
V_{IH}	High Level Input Voltage (TTL inputs)	V_{DD} = MAX	2.0			V
V_{T+}	Positive-going Threshold (CMOS Schmitt inputs)	V_{DD} = 5.0			4.0	V
V_{T-}	Negative-going Threshold (CMOS Schmitt inputs)	V_{DD} = 5.0	0.8			V
V_H	Hysteresis Voltage (CMOS Schmitt inputs)	V_{DD} = 5.0	0.3			V
V_{T+}	Positive-going Threshold (TTL Schmitt inputs)	V_{DD} = 5.0			3.0	V

Table 0-18 Input Specifications (Continued)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{T-}	Negative-going Threshold (TTL Schmitt inputs)	V_{DD} = 5.0	0.6			V
V_H	Hysteresis Voltage (TTL Schmitt inputs)	V_{DD} = 5.0	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_{PU2}	Pull Up Resistance	V_{DD} = 5.0 V	50	100	200	kΩ
R_{PU3}	Pull Up Resistance	V_{DD} = 5.0 V	100	200	400	kΩ
R_{PD}	Pull Down Resistance	V_{DD} = 5.0 V	100	200	400	kΩ

Table 0-19 Output Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{OL2}	Low Level Output Current	$V_{OL}=V_{SS}+0.4V$ TS2	6.0			mA
I_{OH2}	High Level Output Current	$V_{OH}=V_{DD}-0.4V$ TS2	-2.0			mA
I_{OL3}	Low Level Output Current	$V_{OL}=V_{SS}+0.4V$ TS3	12.0			mA
I_{OH3}	High Level Output Current	$V_{OH}=V_{DD}-0.4V$ TS3	-4.0			mA
I_{OL4}	Low Level Output Current	$V_{OL}=V_{SS}+0.4V$ TS4	24.0			mA
I_{OH4}	High Level Output Current	$V_{OH}=V_{DD}-0.4V$ TS4	-8.0			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
C_{BID}	Bidirectional Pin Capacitance			10		pF

7.0 A.C. CHARACTERISTICS

Conditions: V_{DD} = 5.0 V ± 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})

T_r, T_f for all TTL inputs must be < 5 nsec (V_{IN} = V_{ILMAX} ~ V_{IHMIN})

C_L = 100 pF (CPU, RAMDAC Interface, LCD Panel Interface)

C_L = 20 pF (Video Memory Interface)

CLK Signal Dependant Input Timing

Pixel Clock:

$$T_S = \left[\frac{1}{f_{CLKI1}}\right], or \left[\frac{1}{f_{CLKI2}}\right]$$

MEMEN input:

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

T_{ml} = low pulse width of MEMEN

PDCLK input:

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

T_{ph} = high pulse width of PDCLK (for 64 kHz input)

External RC Delay for PDCLK:

T_{ml} and T_{ph} should be chosen to be as short as possible, but 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

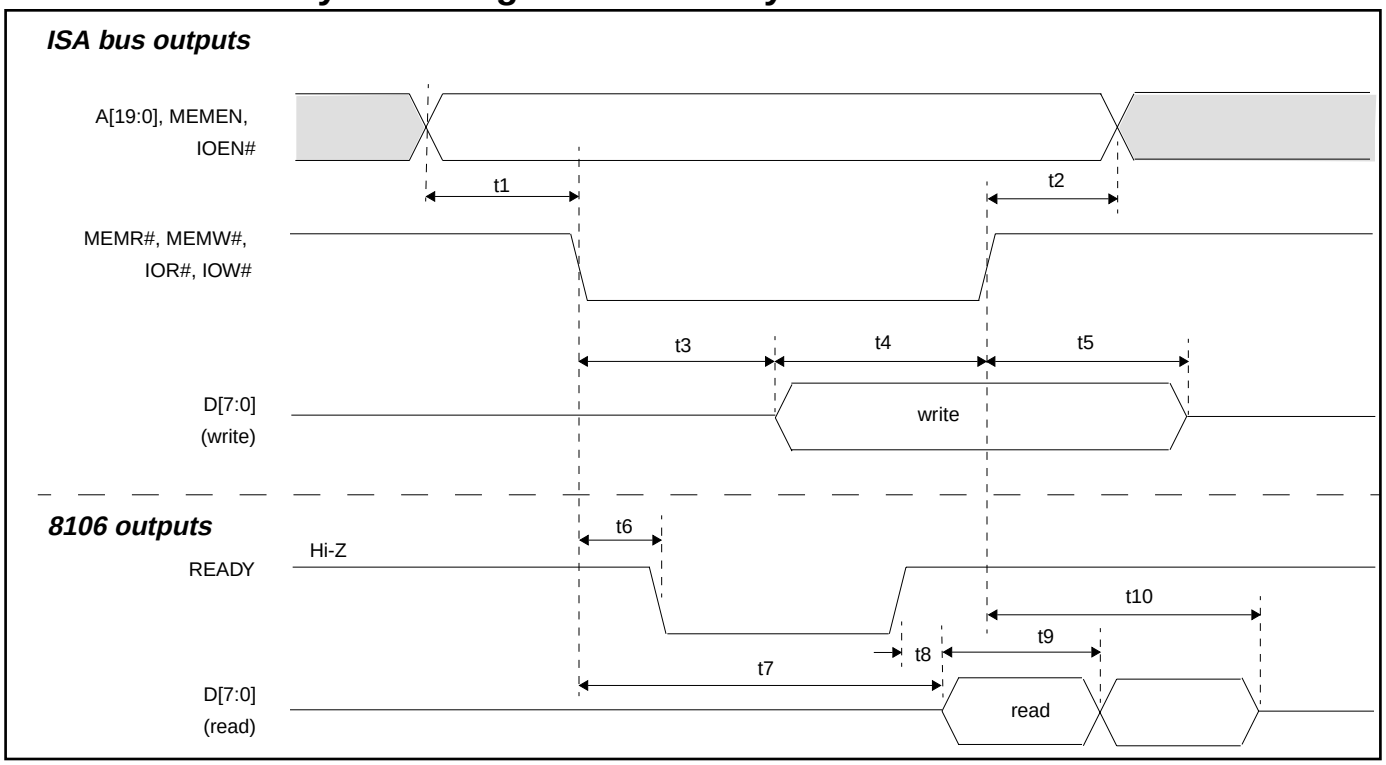


*Figure 4 : CPU Bus Cycle Timing - 8-bit Memory*source: 8106_cpu_bus8_01.can*

Table 0-20 CPU Bus Cycle Timing - 8-bit Memory

Symbol	Parameter	Min	Typ	Max	Units
t1a	A[19:0], MEMEN valid before MEMR#, MEMW# asserted	0			ns
t1b	A[19:0], IOEN# valid before IOR#, IOW# asserted	0			ns
t2a	A[19:0], MEMEN hold from MEMR#, MEMW# negated	0			ns
t2b	A[19:0], IOEN# hold from IOR#, IOW negated	20			ns
t3	MEMW# asserted to D[7:0] valid			3Ts	ns
t4	D[7:0] setup to IOW# negated	0			ns
t5a	D[7:0] hold from MEMW# negated	0			ns
t5b	D[7:0] hold from IOW# negated	10			ns
t6	MEMR#, MEMW# asserted to READY negated			30	ns
t7a	MEMR# asserted to D[7:0] valid, 8 bit read			102Ts+47	ns
t7b	IOR# asserted to D[7:0] valid			98	ns
t8	READY asserted to D[7:0] valid (read)			30	ns
t9a	D[7:0] hold from MEMR# negated	8			ns
t9b	D[7:0] hold from IOR# negated	8			ns
t10a	MEMR# negated to D[7:0] hi-Z delay			29	ns
t10b	IOR# negated to D[7:0] hi-Z delay			26	ns

7.2 CPU Bus Cycle Timing - 16-bit Memory

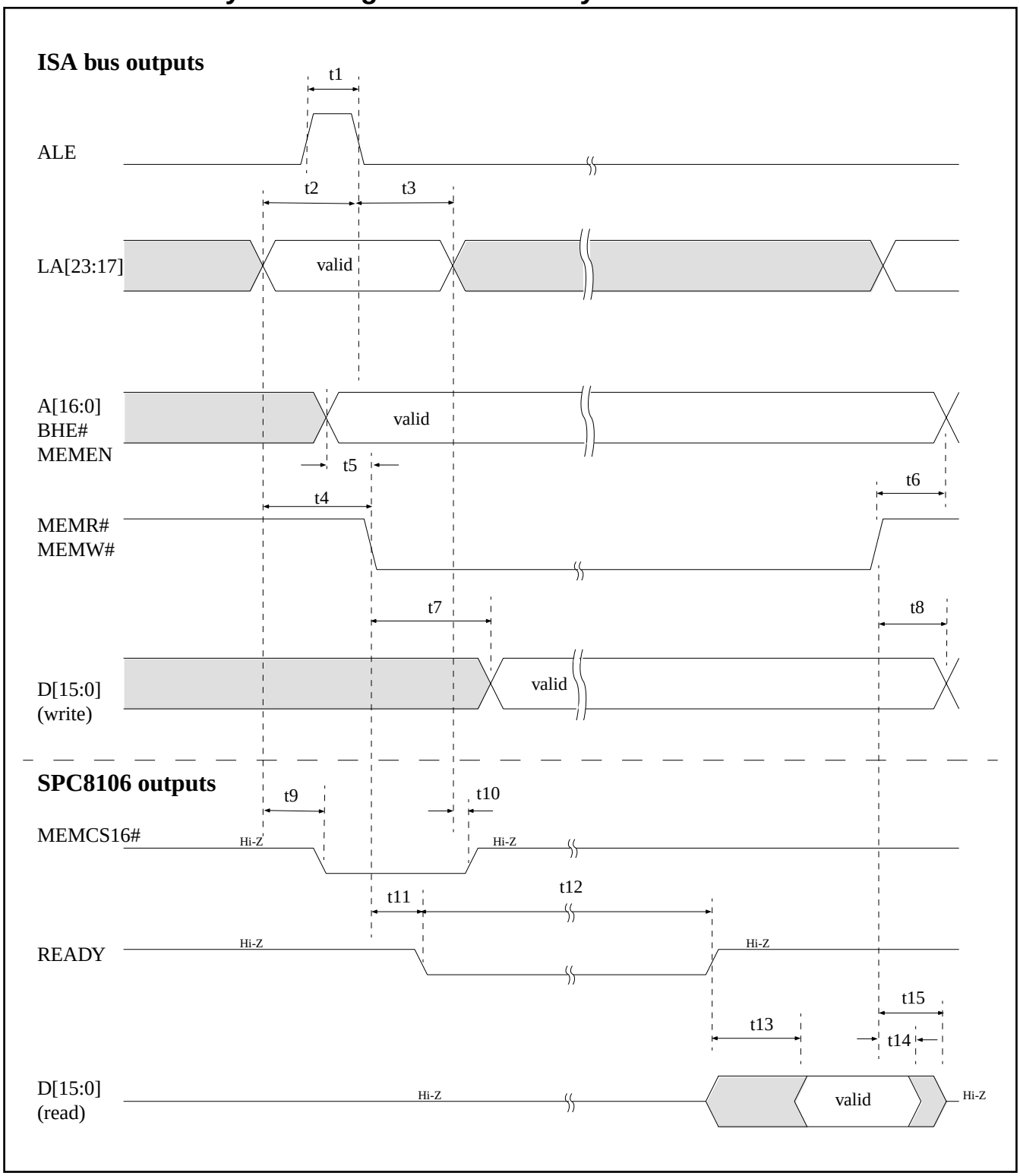


Figure 5 : CPU Bus Cycle Timing - 16-bit Memorysource: 8106_cpu_bus16m_01.can

Table 0-21 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 negated	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	D[15:0] (write) delay from MEMW# asserted			3Ts-10	ns
t8	D[15:0] (write) hold from MEMW# negated	10			ns
t9	MEMCS16# asserted from valid LA[23:17]			30	ns
t10	MEMCS16# hold from LA[23:17] invalid	6			ns
t11	READY negated from MEMR#, MEMW# asserted			50	ns
t12a	READY negated pulse width (dual panel)			138Ts +24	ns
t12b	READY negated pulse width (single panel)			107Ts +24	ns
t13	D[15:0] (read) valid from READY released			40	ns
t14	D[15:0] (read) hold from MEMR# negated	6			ns
t15	MEMR# negated to D[15:0] high-impedance			30	ns

This table refers to standard ISA CPU bus timing. When configuration input MD[5] = 1 on the falling edge of RESET, refer to “Bus Cycle Timing - 16-bit Memory (Modified Address Timing)” on page 34 for modified address timing.

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

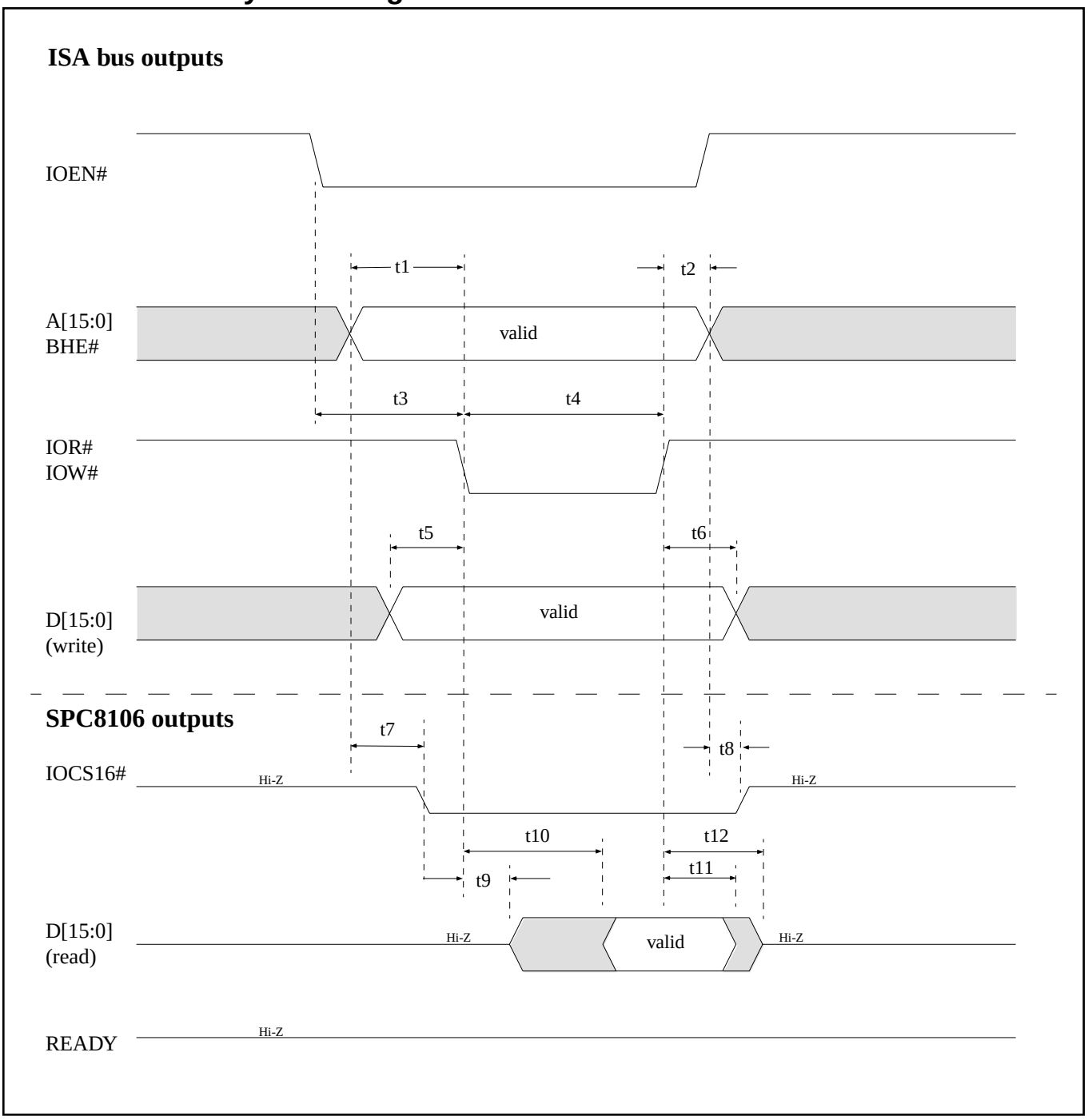


Figure 6 : CPU Bus Cycle Timing - 16-bit I/Osource: 8106_cpu_bus16_io_01.can

Table 0-22 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	10			ns
t3	IOEN# setup to IOR#, IOW# asserted	50			ns
t4a	IOW# pulse width (16 bit access)	120			ns
t4b	IOR# pulse width (16 bit access)	180			ns
t4c	IOW# pulse width (8 bit access)	200			ns
t4d	IOR# pulse width (8 bit access)	160			ns
t5	valid D[15:0] (write) setup to IOW# asserted	10			ns
t6	valid D[15:0] (write) hold from IOW# negated	10			ns
t7	IOCS16# asserted from A[15:0] valid			30	ns
t8	IOCS16# hold from A[15:0] invalid	12			ns
t9	D[15:0] (read) driven delay from IOR# asserted	5			ns
t10	valid D[15:0] (read) from IOR# asserted			150	ns
t11	D[15:0] (read) hold from IOR# negated	6			ns
t12	IOR# negated to D[15:0] high-impedance			30	ns

This table refers to standard ISA CPU bus timing. When configuration input MD[5] = 1 on the falling edge of RESET, refer to “CPU Bus Cycle Timing - 16-bit I/O (Modified Address Timing)” on page 35 for modified address timing.

7.4 Bus Cycle Timing - 16-bit Memory (Modified Address Timing)

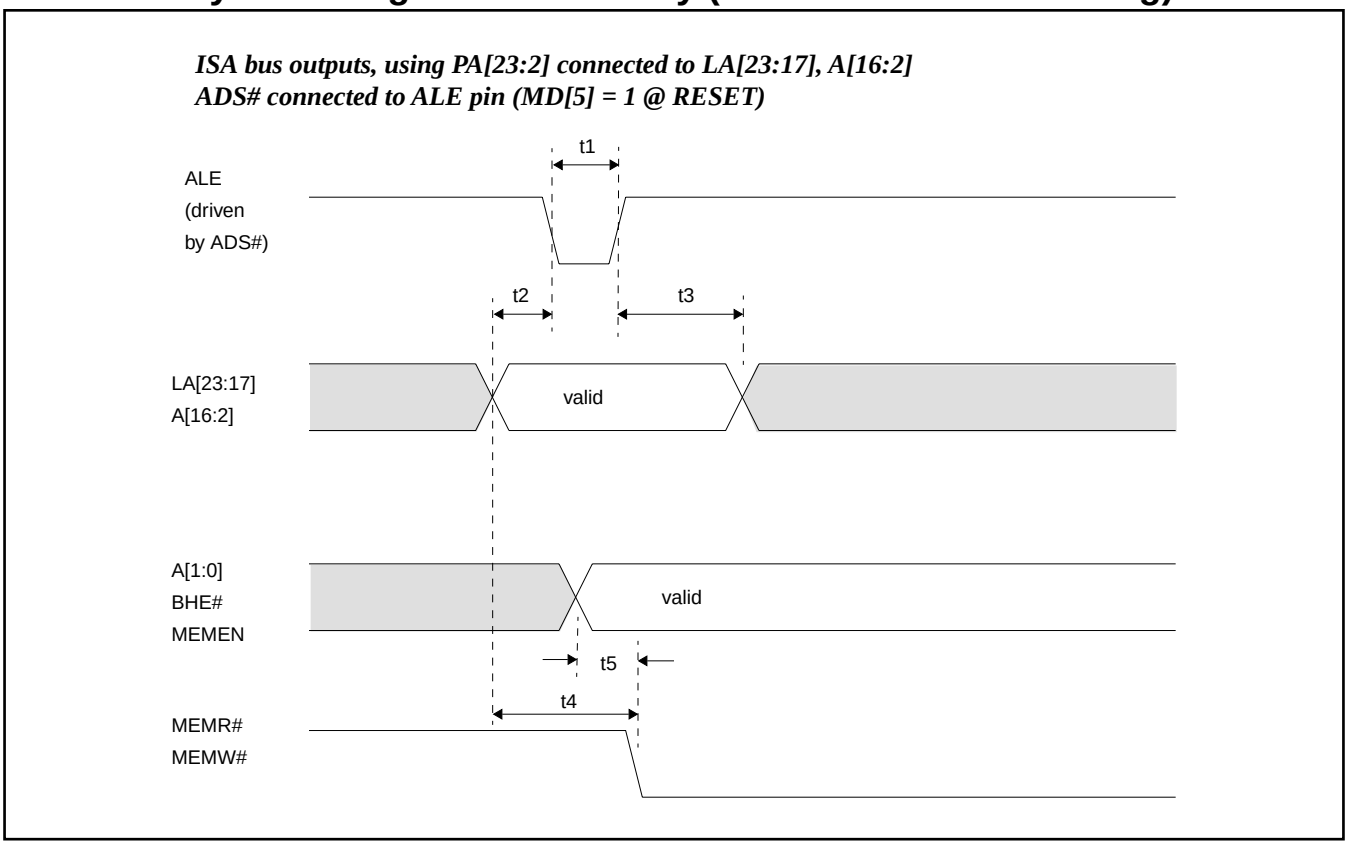


*Figure 7 : CPU Bus Cycle Timing - 16-bit Memory, Modified Address Timing*source: 8106_cpu_bus16_mma_01.can

Table 0-23 CPU Bus Cycle Timing - 16-bit Memory, Modified Address Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	ADS# pulse width (ALE pin)	30			ns
t2	LA[23:17], A[16:2] valid setup to ADS# asserted	0			ns
t3	LA[23:17], A[16:2] valid hold from ADS# negated	10			ns
t4	LA[23:17], A[16:2] valid setup to MEMR#, MEMW# asserted	10			ns
t5	A[1:0], BHE#, MEMEN valid setup to MEMR#, MEMW# asserted	10			ns

This table refers to CPU bus timing when configuration input MD[5] = 1 on the falling edge of RESET. In this case, processor address lines PA[23:2] may be connected to address inputs LA[23:17] and A[16:2]. These inputs will be latched by the SPC8106 on the rising edge of the ALE input pin. Note that the ALE pin should be driven by the ADS# signal from the CPU. Address inputs A[1:0] should be connected to the ISA bus address outputs A[1:0].

7.5 CPU Bus Cycle Timing - 16-bit I/O (Modified Address Timing)

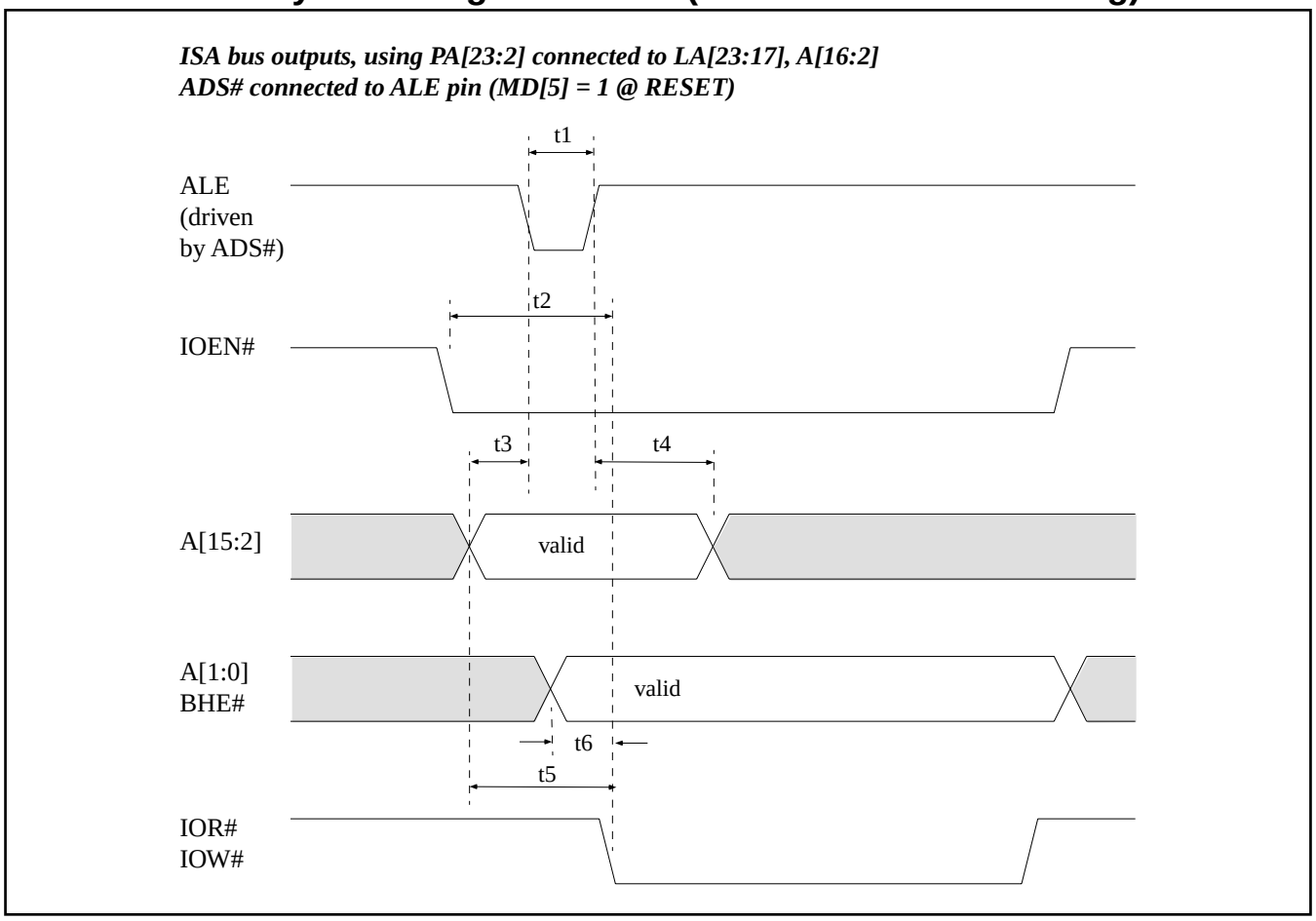


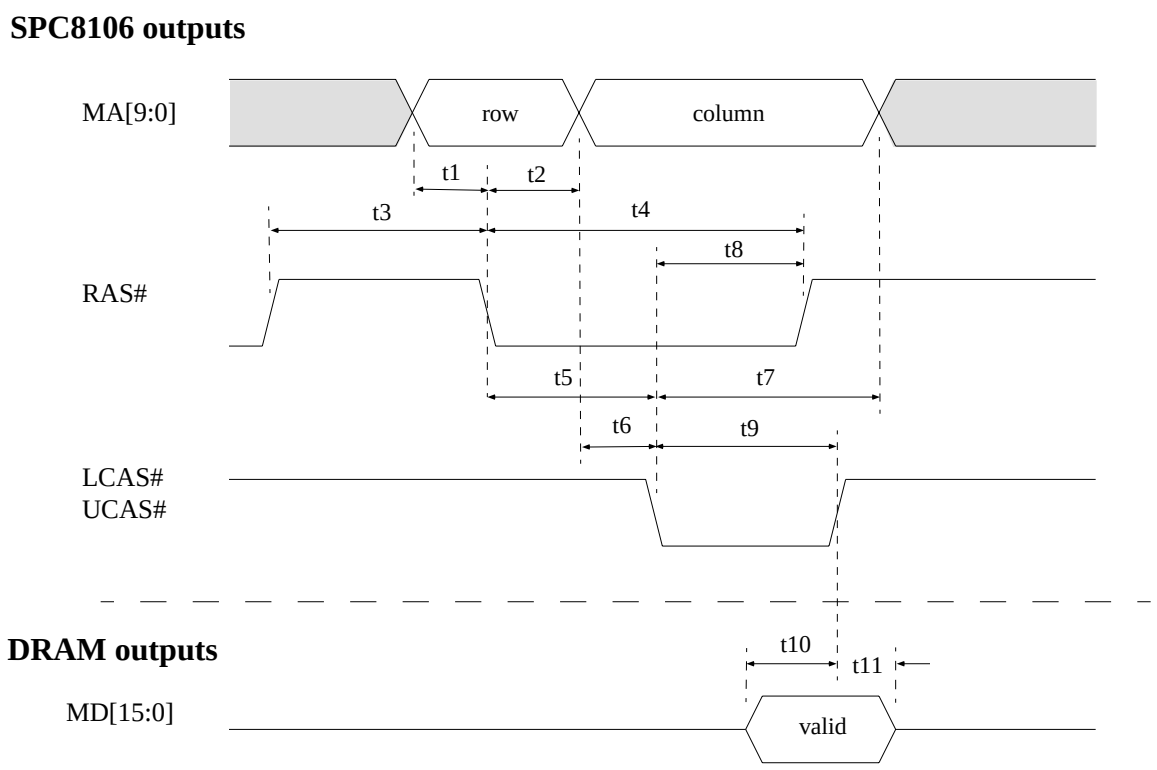
*Figure 8 : CPU Bus Cycle Timing - 16-bit I/O - Modified Address Timing*source: 8106_cpu_bus16_ioma_01.can

Table 0-24 CPU Bus Cycle Timing - 16-bit I/O - Modified Address Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	ADS# pulse width	30			ns
t2	IOEN# active setup IOR#, IOW# asserted	50			ns
t3	A[15:2] valid setup to ADS# asserted	0			ns
t4	A[15:2] valid hold from ADS# negated	10			ns
t5	A[15:2] valid setup to IOR#, IOW# asserted	10			ns
t6	A[1:0], BHE# valid setup to IOR#, IOW# asserted	10			ns

This table refers to CPU bus timing when configuration input MD[5] = 1 on the falling edge of RESET. In this case, processor address lines PA[23:2] may be connected to address inputs LA[23:17] and A[16:2]. These inputs will be latched by the SPC8106 on the rising edge of the ALE input pin. Note that the ALE pin should be driven by the ADS# signal from the CPU. Address inputs A[1:0] should be connected to the ISA bus address outputs A[1:0].

7.6 DRAM Read Cycle Timing - Non-Page Mode



Note: DRAM access time is guaranteed to be 80 nsec minimum.

Figure 9 : DRAM Read Cycle Timing - Non-Page Mode source: 8106_dram_read_npm_01.can

Table 0-25 DRAM Read Cycle Timing - Non-Page Mode

Symbol	Parameter	Min	Typ	Max	Units
t1	MA[9:0] row address setup to RAS# asserted	Ts-12			ns
t2	MA[9:0] row address hold from RAS# asserted	0.5Ts			ns
t3	RAS# precharge	2Ts-10			ns
t4	RAS# pulse width	2.5Ts+1		2.5Ts+10	ns
t5	RAS# asserted to LCAS#, UCAS# asserted	1.5Ts-5			ns
t6	MA[9:0] column address setup to LCAS#, UCAS# asserted	Ts-15			ns
t7	MA[9:0] column address hold from LCAS#, UCAS# asserted	Ts+1			ns
t8	LCAS#, UCAS# asserted to RAS# negated	Ts+2			ns
t9	LCAS#, UCAS# pulse width	Ts-3			ns
t10	MD[15:0] read data setup to LCAS#, UCAS# negated	12			ns
t11	MD[15:0] read data hold from LCAS#, UCAS# negated	0			ns

7.7 DRAM Read Cycle Timing - Page Mode

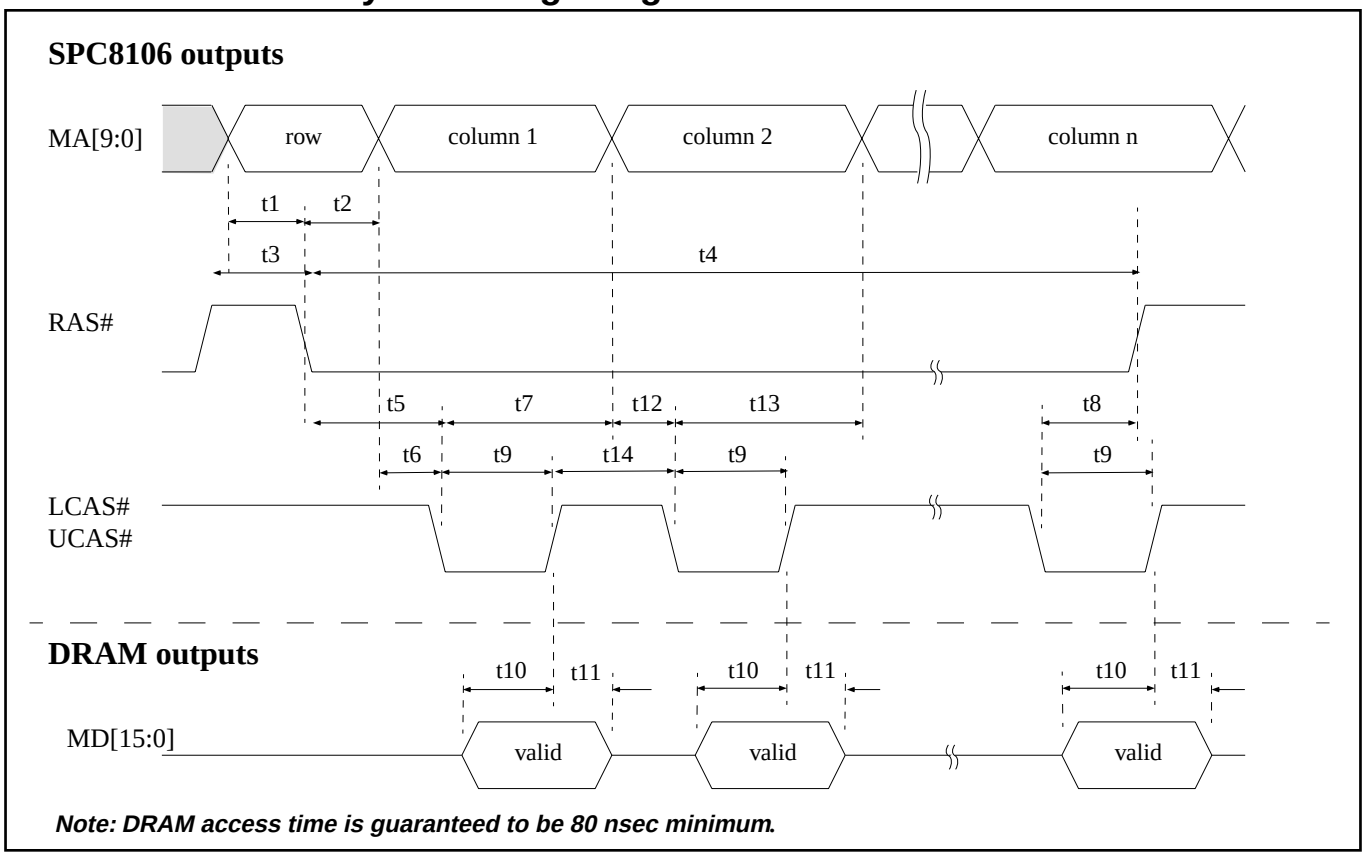


Figure 10 : DRAM Read Cycle Timing - Page Mode *source: 8106_dram_read_pm_01.can*

Table 0-26 DRAM Read Cycle Timing - Page Mode

Symbol	Parameter	Min	Typ	Max	Units
t1	MA[9:0] row address setup to RAS# asserted	Ts-14			ns
t2	MA[9:0] row address hold from RAS# asserted	0.5Ts			ns
t3	RAS# precharge	2Ts-10			ns
t4	RAS# pulse width	3.5Ts+4		24.5Ts+13	ns
t5	RAS# asserted to LCAS#, UCAS# asserted	1.5Ts-5			ns
t6	MA[9:0] col addr setup to LCAS#, UCAS# asserted (1st)	Ts-10			ns
t7	MA[9:0] col addr hold from LCAS#, UCAS# asserted(1st)	0.5Ts+1			ns
t8	LCAS#, UCAS# asserted to RAS# negated	0.5Ts+3			ns
t9	LCAS#, UCAS# pulse width	Ts-2			ns
t10	MD[15:0] read data setup to LCAS#, UCAS# negated	12			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 (2^{nd} - n^{th} page access)	Ts-10			ns
t13	MA[9:0] column address hold from LCAS#, UCAS# asserted (2^{nd} - n^{th} page access)	0.5Ts+1			ns
t14	LCAS#, UCAS# precharge	0.5Ts			ns

7.8 DRAM Write Cycle Timing

SPC8106 outputs

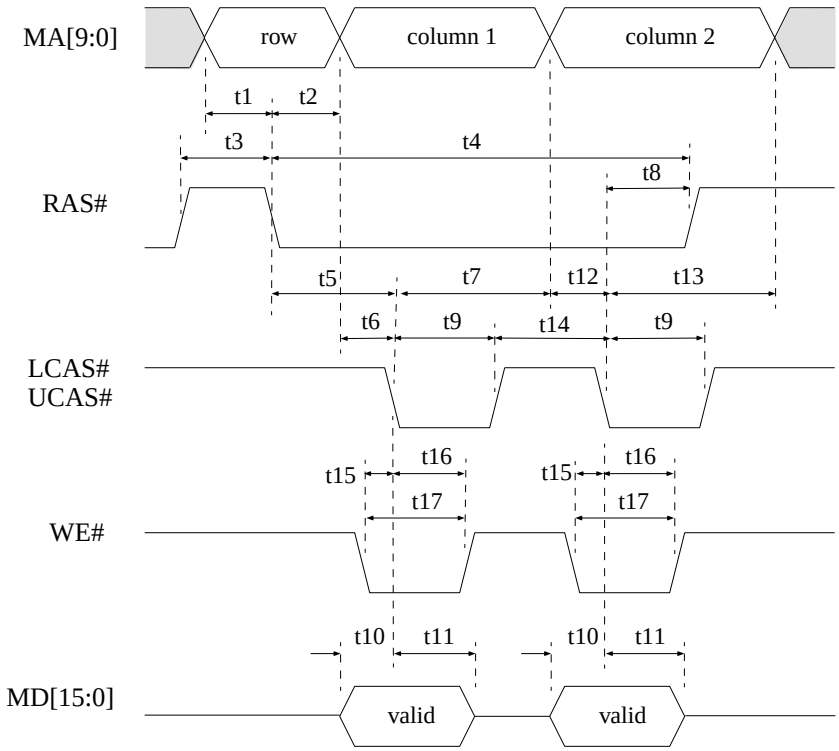


Figure 11 : DRAM Write Cycle Timing source: 8106_write_01.can

Table 0-27 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			ns
t3	RAS# precharge	2Ts-10			ns
t4	RAS# pulse width	3.5Ts+3			ns
t5	RAS# asserted to LCAS#, UCAS# asserted	1.5Ts-5			ns
t6	MA[9:0] col addr setup to LCAS#, UCAS# asserted (1st)	Ts-15			ns
t7	MA[9:0] col addr hold from LCAS#, UCAS# asserted(1st)	.5Ts+1			ns
t8	LCAS#, UCAS# asserted to RAS# negated	0.5Ts+3			ns
t9	LCAS#, UCAS# pulse width	Ts-3			ns
t10	MD[15:0] write data setup to LCAS#, UCAS# asserted	0			ns
t11	MD[15:0] write data hold from LCAS#, UCAS# asserted	0.5Ts+6			ns
t12	MA[9:0] col addr setup to LCAS#, UCAS# asserted (2^{nd})	Ts-10			ns
t13	MA[9:0] col addr held from LCAS#, UCAS# asserted (2^{nd})	0.5Ts			ns
t14	LCAS#, UCAS# precharge	0.5Ts-5			ns
t15	WE# setup to LCAS#, UCAS# asserted	0.5Ts-4			ns
t16	WE# hold from LCAS#, UCAS# asserted	0.5Ts-1			ns
t17	WE# pulse width	Ts-2			ns

7.9 DRAM Refresh Timing I

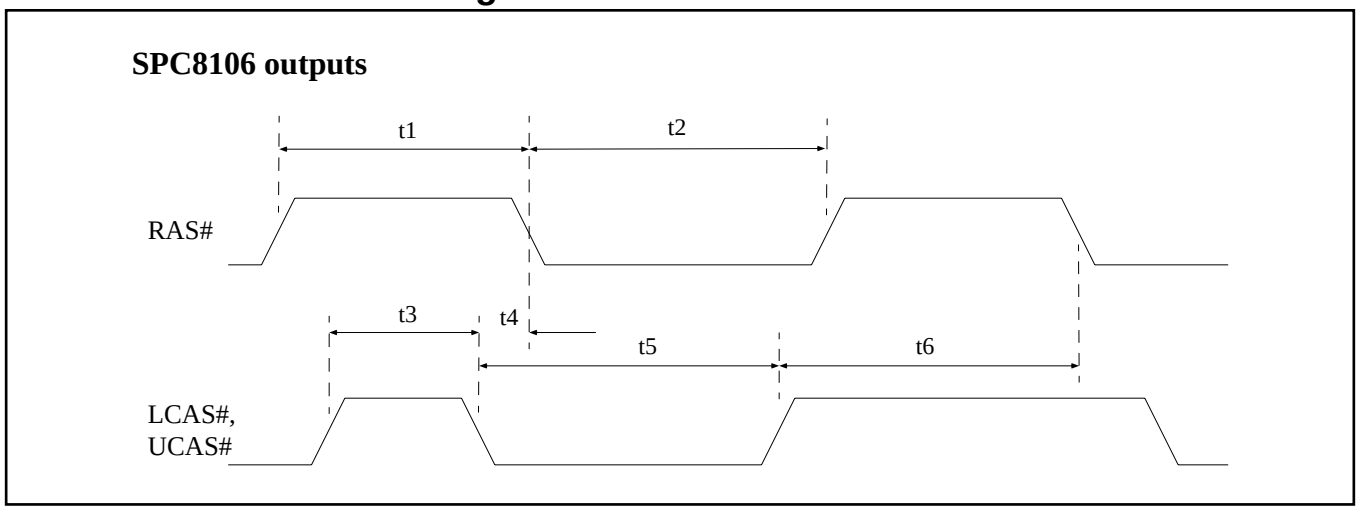


Figure 12 : DRAM Refresh Timing I (generated from CLKI, Sequence running)source:8106_refresh_01.can

Table 0-28 DRAM Refresh Timing I (generated from CLKI, Sequence running

Symbol	Parameter	Min	Typ	Max	Units
t1	RAS# high pulse width	2Ts-10			ns
t2	RAS# low pulse width	3.5Ts+3			ns
t3	LCAS#, UCAS# high pulse width	0.5Ts-5			ns
t4	LCAS#, UCAS# setup to RAS# asserted (CAS-before-RAS refresh)	Ts-4			ns
t5	LCAS#, UCAS# low pulse width	4Ts-3			ns
t6	LCAS#, UCAS# negated to RAS# asserted	2.5Ts-8			ns

These timing values apply to CAS-before-RAS refresh cycles occurring during all active modes, and all Power Save modes where the Sequencer is not halted (Power Save Mode 1, Power Save Mode 2 state 1, and Power Save Mode 5).

7.10 DRAM Refresh Timing II

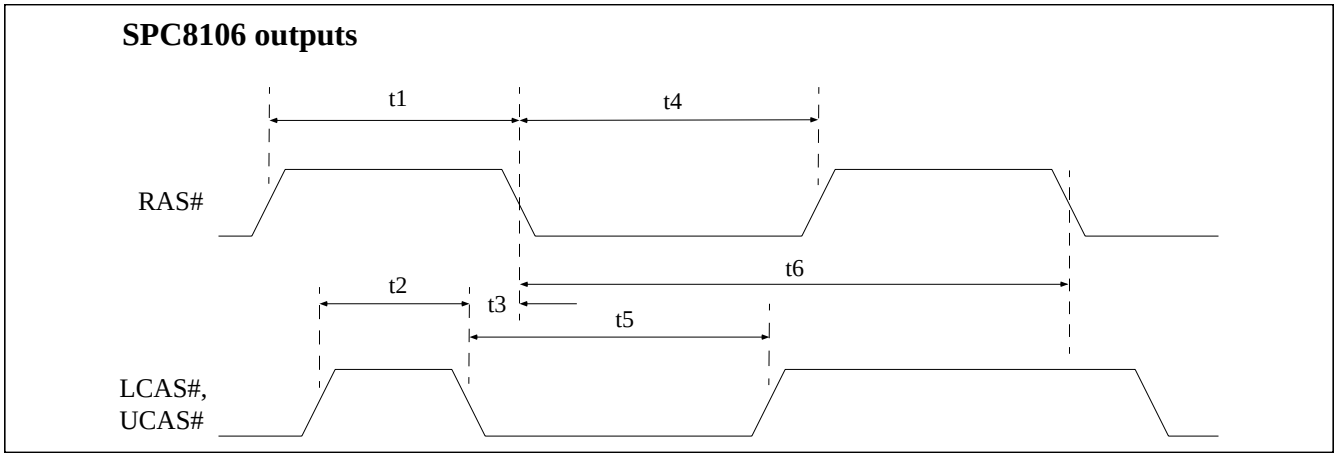


Figure 13 : DRAM Refresh Timing II (Sequencer stopped)source: 8106_refresh2_01.can

Table 0-29 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	10			ns
t4a	RAS# low pulse width (Sequencer stopped, selected CLKI used as refresh clock source)	4Ts-58			ns
t4b	RAS# low pulse width (Sequencer stopped, MEMEN input used as refresh clock source)	Tml-58			ns
t4c	RAS# low pulse width (Sequencer stopped, 64 kHz PDCLK input used as refresh clock source)	Tph-58			ns
t4d	RAS# low pulse width (Sequencer stopped, 32 kHz PDCLK input used as refresh clock source, internal clock doubling enabled)	150			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			ns
t5c	LCAS#, UCAS# low pulse width (Sequencer stopped, 64 kHz PDCLK input used as refresh clock source)	Tph			ns
t5d	RAS# low pulse width (Sequencer stopped, 32 kHz used as refresh source, internal clock doubling enabled)	166			ns
t6a	Power Save Mode 4 SUSPEND# Mode refresh period (Sequencer stopped, selected CLKI used as refresh source	384Ts		3072Ts	us
t6b	Power Save Mode 4 SUSPEND# Mode refresh period (Sequencer stopped, MEMEN input used as refresh clock source)	Tm		8Tm	us
t6c	Power Save Mode 4 SUSPEND# Mode refresh period (Sequencer stopped, 64 kHz PDCLK input used as refresh clock source)	Tp		8Tp	us
t6d	Power Save Mode 4 SUSPEND# Mode refresh period (Sequencer stopped, 32 kHz used as refresh source, internal clock doubling enabled)	0.5Tp		4Tp	us

These timing values apply to CAS-before-RAS refresh cycles occurring only during those modes where the Sequencer is stopped and the refresh timing is being generated from the selected CLKI input, from MEMEN input, or from PDCLK input (Power Save Mode 2 state 2, Power Save Mode 4, and Hardware Suspend Mode). These values also apply to self-refresh modes (Power Save Mode 4, and Hardware Suspend Mode).

7.11 RAMDAC Interface Timing

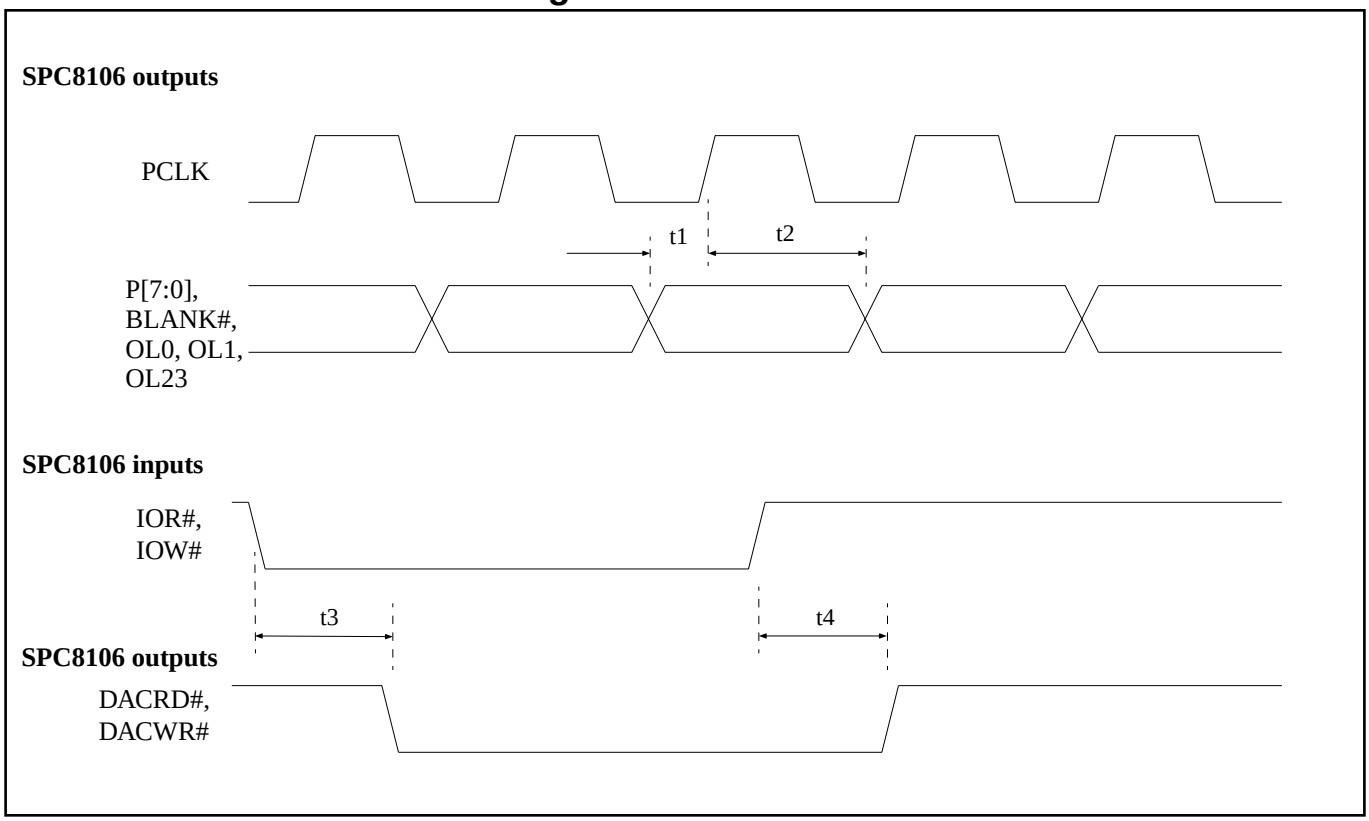


Figure 14 : RAMDAC Interface Timing

Table 0-30 RAMDAC Interface Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	P[7:0], BLANK#, OL0, OL1, OL23 setup to PCLK rising edge	6			ns
t2	P[7:0], BLANK#, OL0, OL1, OL23 hold from PCLK rising edge	6			ns
t3	IOR#, IOW# asserted to DACRD#, DACWR# asserted			15	ns
t4	IOR#, IOW# negated to DACRD#, DACWR# negated			18	ns

7.12 Power Save Mode LCD Signal Timing - Software Power Save Modes

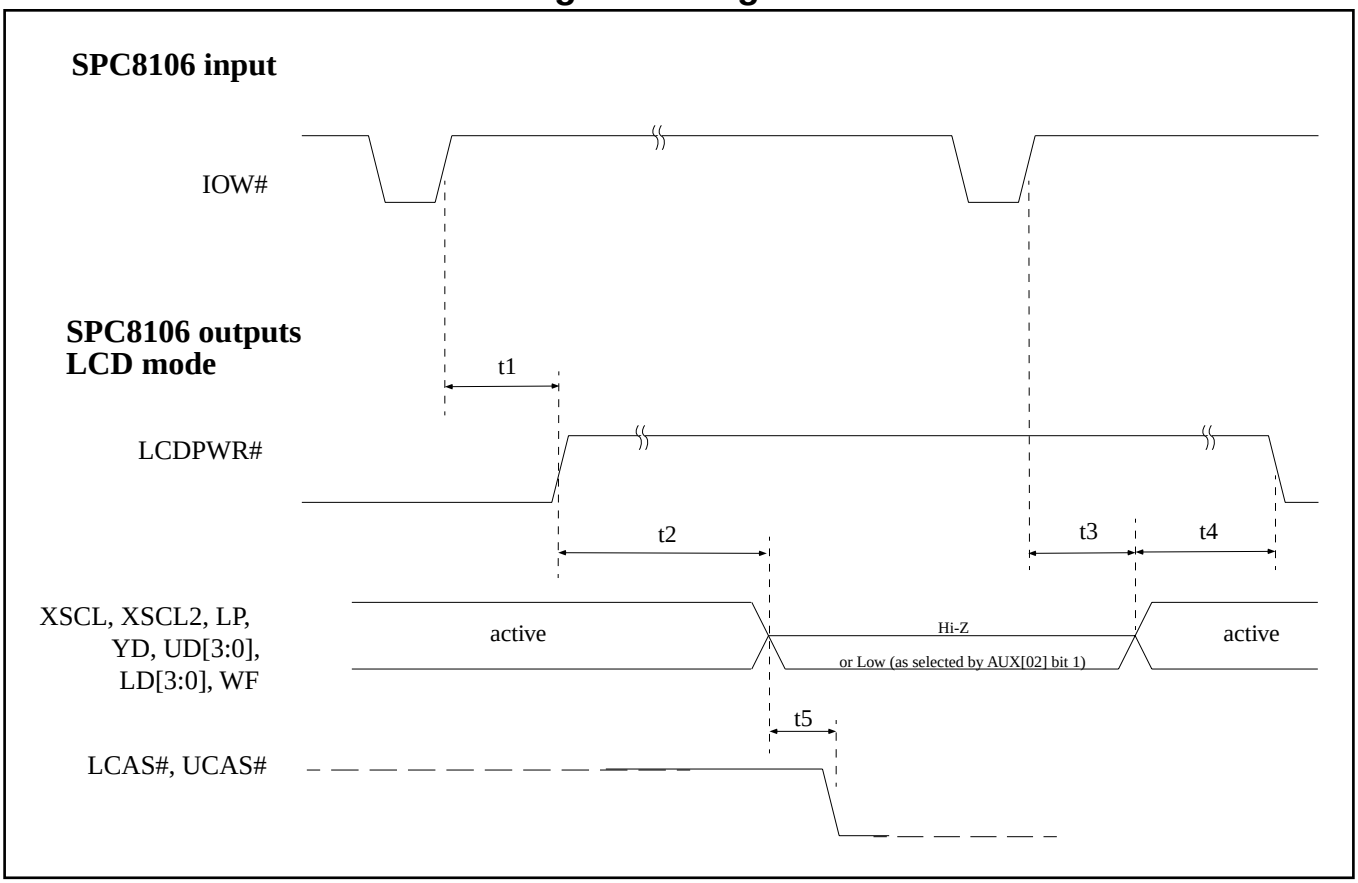


Figure 15 : Power Save Mode LCD Signal Timing

Table 0-31 Power Save Mode LCD Signal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Power Save Register IOW# negated (valid Power Save Mode set) to LCDPWR# negated			4Ts+20	ns
t2	LCDPWR# negated to LCD interface signals high-impedance, or driven low (as selected by AUX[02] bit 1)	2Ty		3Ty	ns
t3	Power Save Register IOW# negated (return to active mode) to LCD interface signals active			Ty+4Ts +20	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, Power Save Mode 4 only)	Ts			ns

Ty = YD period, typically 12.8 ms

7.13 Power Save Mode CRT Signal Timing - Software Power Save Modes

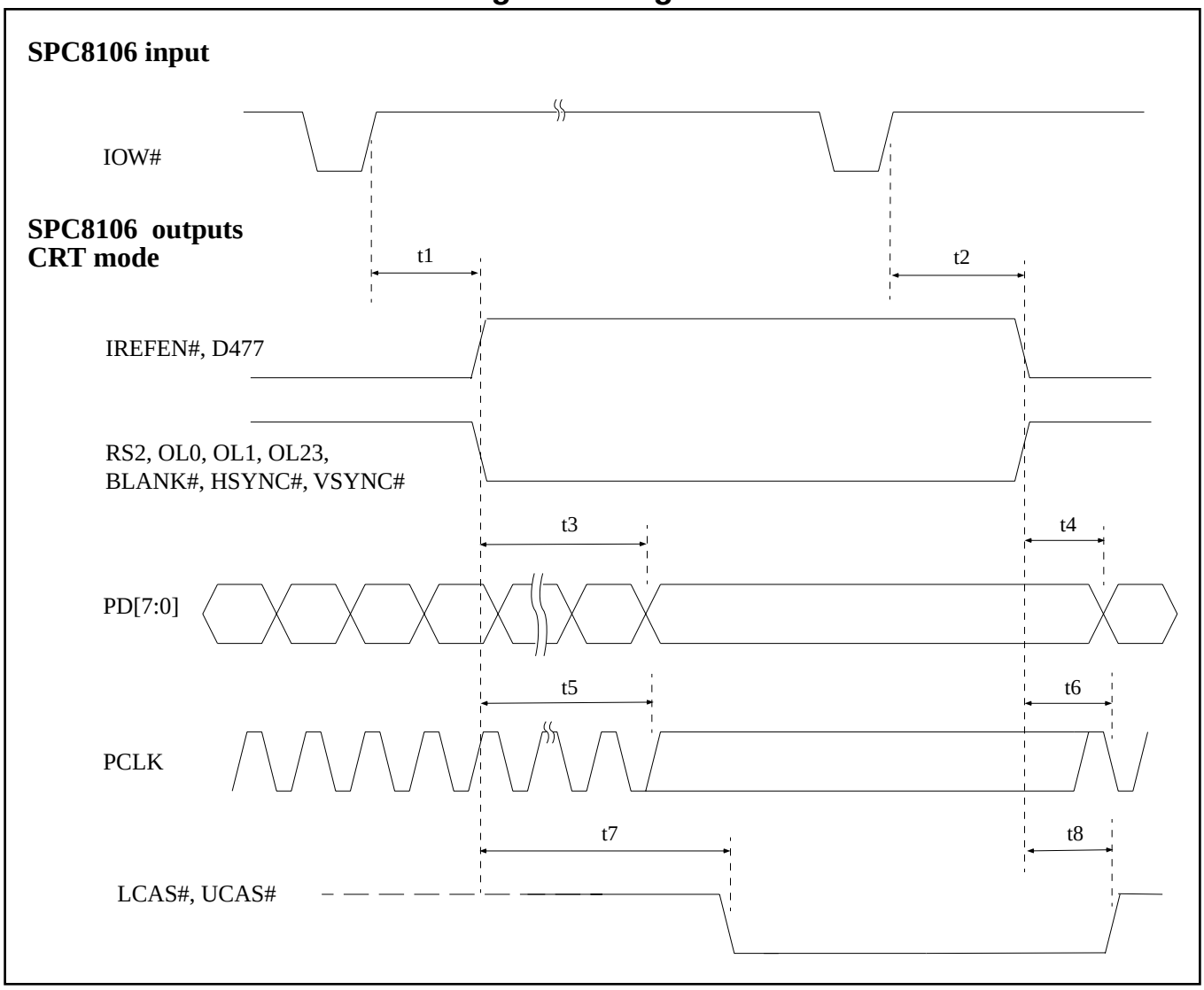


*Figure 16 : Power Save Mode CRT Signal Timing*source: 8106_psm_crt_sig_01.can

Table 0-32 Power Save Mode CRT Signal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Power Save Register IOW# negated (valid Power Save Mode set) to IREFEN#, D477 driven high, and RS2, OL0, OL1, OL23, BLANK#, HSYNC#, VSYNC# driven low.	2Ty		3Ty+4Ts+ 20	ns
t2	Power Save Register IOW# command negated (return to active mode) to IREFEN#, D477 driven low, and RS2, OL0, OL1, OL23, BLANK#, HSYNC#, VSYNC# active.			4Ts+20	ns
t3a	IREFEN# negated to PD[7:0] inactive			40Ts	ns
t3b	IREFEN# negated to PD[7:0] inactive (Power Save Modes 3,4 only)			20Ts	ns
t4	IREFEN# asserted to PD[7:0] active			80Tx	ms
t5	IREFEN# negated to PCLK halted (Power Save Modes 3,4 only)			20Ts	ns
t6	IREFEN# asserted to PCLK active (Power Save Modes 3,4 only)			20Ts	ns
t7	IREFEN# negated to LCAS#, UCAS# asserted (to initialize DRAM self-refresh mode, Power Save Mode 4 only)	Ts			ns
t8	IREFEN# asserted to RAS#, LCAS#, UCAS# negated (terminate DRAM self-refresh mode, Power Save Mode 4 only)			10Ts	ns

Ty = VSYNC# period (typically 14.3 ms or 16.7 ms, depending on video mode)
Tx= HSYNC# period (typically 31.8 μs)

7.14 Power Save Mode LCD Signal Timing - Hardware Suspend Mode

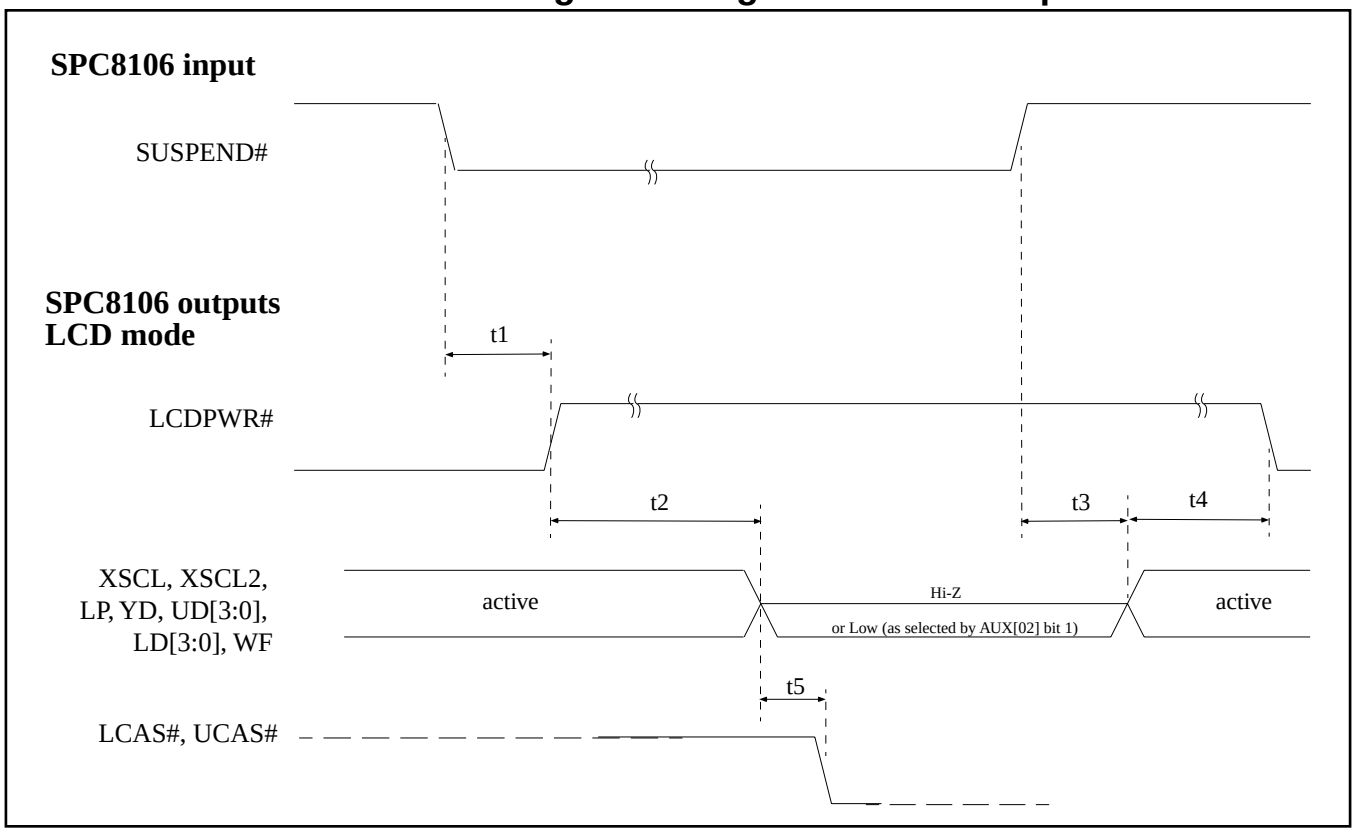


*Figure 17 : Suspend Mode LCD Signal Timing*source: 8106foc_sm_lcd_sig_01.can

Table 0-33 Suspend Mode LCD Signal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	SUSPEND# asserted to LCDPWR# negated			4Ts	ns
t2	LCDPWR# negated 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)			Ty + 4Ts	ns
t3b	SUSPEND# negated to LCD interface signals driven (self-refresh mode, MEMEN, or PDCLK selected as refresh clock source)	Tv+3Ts		2Tv + Ty +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
= 393,216Ts if CLKI or self-refresh mode is selected
(approximately13.9 ms for 28.322 MHz, 15.7 ms for 25.175 MHz)
= 1024 Tm if MEMEN Refresh clock source is selected
= 1024 Tp if 64 kHz PDCLK Refresh clock source is selected
= 512 Tp if 32 kHz PDCLK Refresh clock source is selected

Ty = YD period, typically 12.8 ms

7.15 Power Save Mode CRT Signal Timing - Hardware Suspend Mode

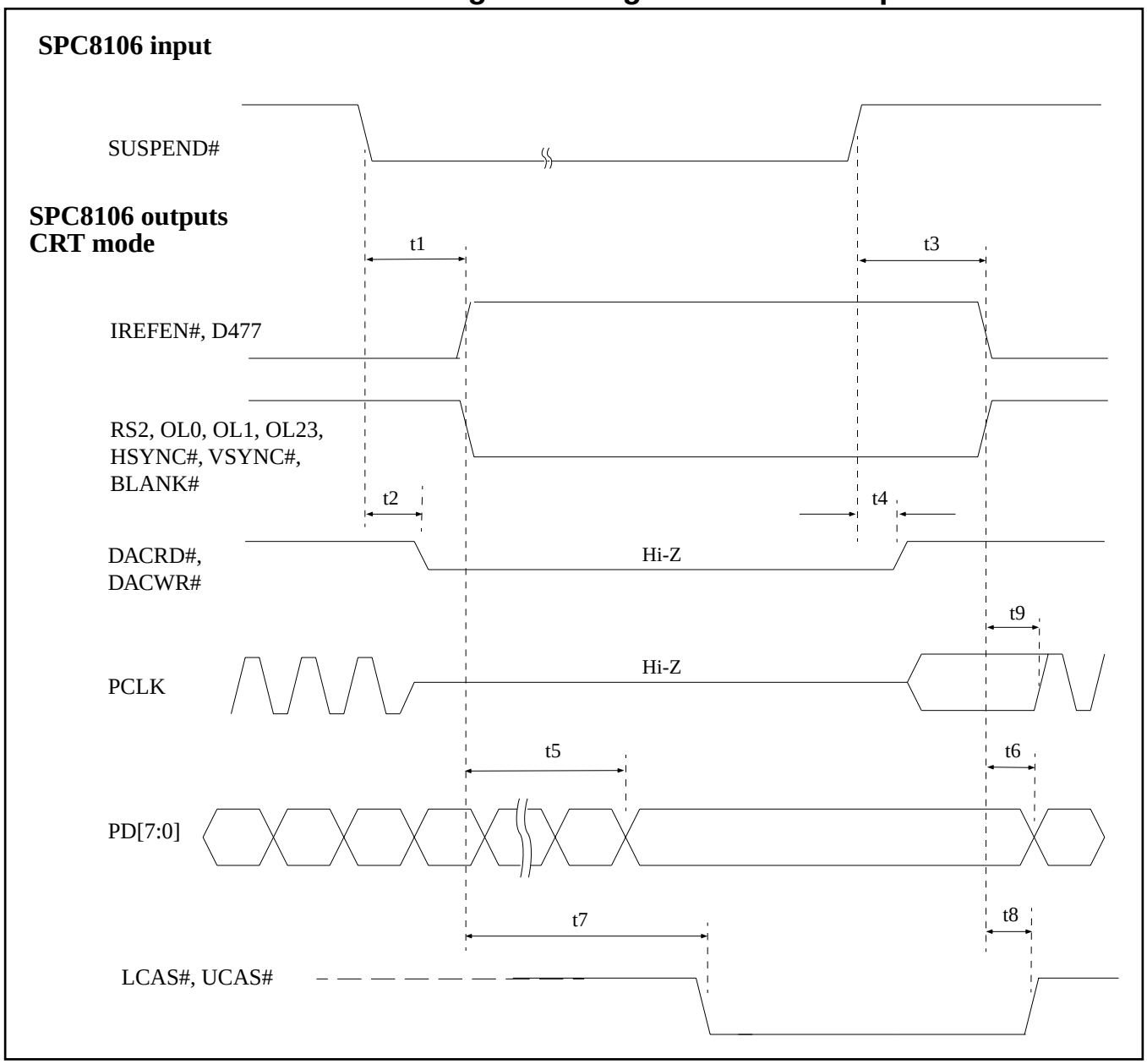


Figure 18 : Suspend Mode CRT Signal Timing source: 8106_sm_crt_sig_01.can

Table 0-34 Suspend Mode CRT Signal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	SUSPEND# asserted to IREFEN#, D477 driven high, and RS2, OL0, OL1, OL23, BLANK#, HSYNC#, VSYNC# driven low.	2Ty		3Ty+4Ts	ns
t2	SUSPEND# asserted to PCLK, DACRD#, DACWR# high-impedance			20	ns
t3a	SUSPEND# negated to IREFEN#, D477 driven low, and RS2, OL0, OL1, OL23, BLANK#, HSYNC#, VSYNC# active. (CLKI selected as refresh clock source)			4Ts	ns
t3b	SUSPEND# negated to IREFEN#, D477 driven low, and RS2, OL0, OL1, OL23, BLANK#, HSYNC#, VSYNC# active. (self-refresh, MEMEN or PDCLK selected as refresh clock source)	Tv+3Ts		2Tv+4Ts	ns
t4	SUSPEND# negated to DACRD#, DACWR# driven high, PCLK driven (halted)			20	ns
t5	IREFEN# negated to PD[7:0] inactive			20Ts	ns
t6	IREFEN# asserted to PD[7:0], PCLK active			80Tx	ms
t7	IREFEN# negated to LCAS#, UCAS# asserted (to initialize DRAM self-refresh mode)	Ts			ns
t8	IREFEN# asserted to RAS#, LCAS#, UCAS# negated (terminate DRAM self-refresh mode)			10Ts	ns
t9	IREFEN# asserted to PCLK active			20Ts	ns

Tv = internal vertical retrace period
= 393,216Ts if CLKI or self-refresh mode is selected
(approximately 13.9 ms for 28.322 MHz, 15.7 ms for 25.175 MHz)
= 1024 Tm if MEMEN Refresh clock source is selected
= 1024 Tp if 64 kHz PDCLK Refresh clock source is selected
= 512 Tp if 32 kHz PDCLK Refresh clock source is selected

Ty = VSYNC# periods (typically 14.3 ms or 16.7 ms, depending on video mode

Tx = HSYNC# period (typically 31.8 µs)

7.16 Clock Input Requirements

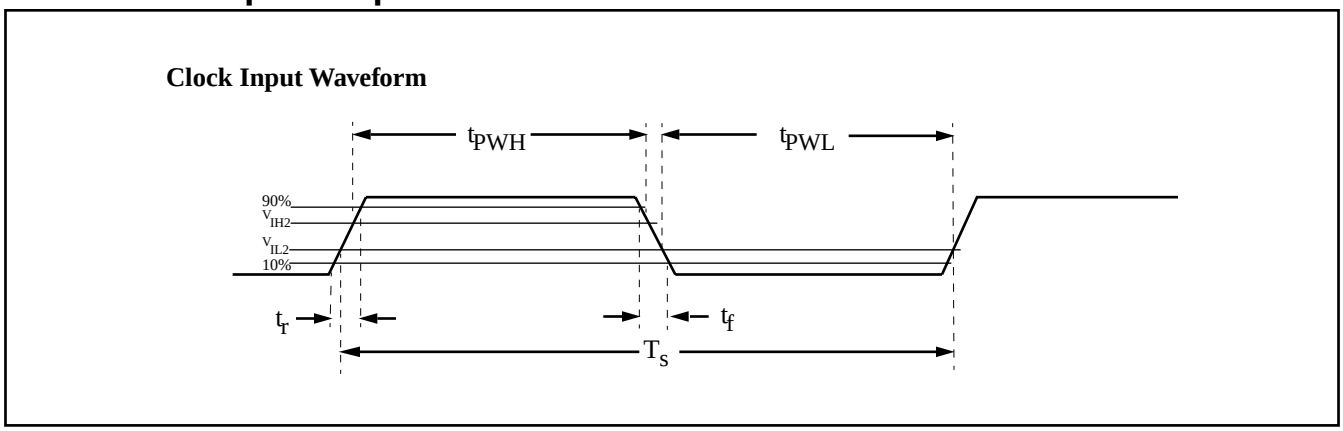


*Figure 19 : Clock Input Requirements*source: 8106_clk_in_01.can

Table 0-35 Clock Input Requirements

Symbol	Parameter	Min	Typ	Max	Units
T_S	Input Clock Period (CLKI1, CLKI2)	35.3			ns
T_m	Input Clock Period (MEMEN)		15.625		µs
$T_{p\ (32)}$	Input Clock Period (32 kHz PDCLK)		31.25		µs
$T_{p\ (64)}$	Input Clock Period (64 kHz PDCLK)		15.625		µs
t_{PWH}	Input Clock Pulse Width High (CLKI1, CLKI2)	40		60	Ts
t_{PWL}	Input Clock Pulse Width Low (CLKI1, CLKI2)	40		60	Ts
t_f	Input Clock Fall Time (10% - 90%)		5		ns
t_r	Input Clock Rise Time (10% - 90%)		5		ns

7.17 LCD Interface Timing - 4-Bit Single Monochrome Panel

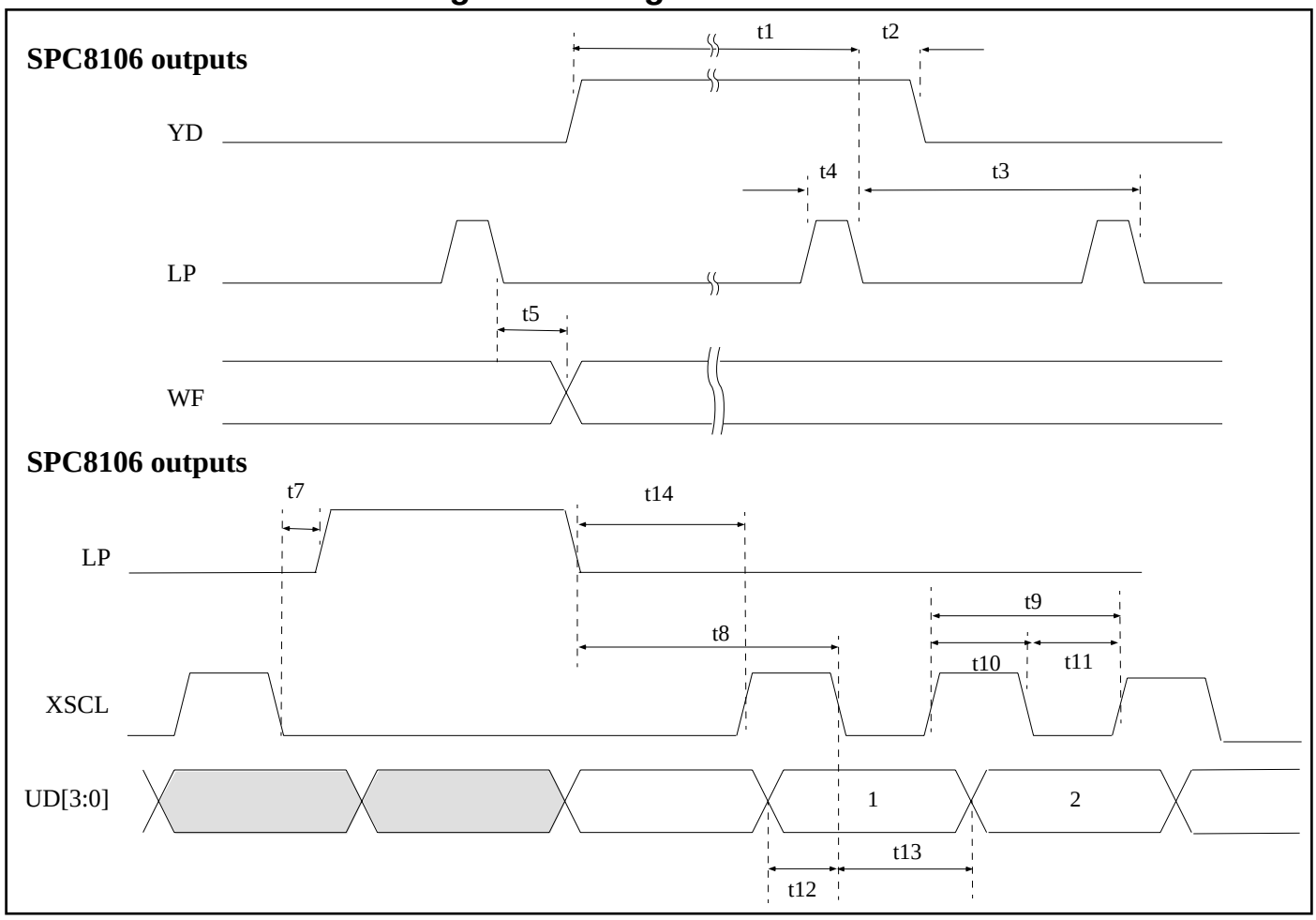


Figure 20 : LCD Interface Timing - 4-Bit Single Monochrome Panel

Table 0-36 LCD Interface Timing - 4-Bit Single Monochrome Panel

Symbol	Parameter	Min	Typ	Max	Units
t1	YD setup to LP negated (single panel mode)	739Ts-24			ns
t2	YD hold from LP negated (single panel mode)	13Ts-24			ns
t3	LP period (single panel mode)		752Ts		ns
t4a	LP pulse width (AUX[0D] bit 6 = 0)	5Ts-24			ns
t4b	LP pulse width (AUX[0D] bit 6 = 1)	6Ts-24			ns
t5	WF delay from LP negated	0		20	ns
t7	LP hold from XSCL falling edge (AUX[0D] bit 7, 6 = 00)	n/a	n/a	n/a	
t8	LP negated to XSCL falling edge (AUX[0D] bits 7,6 =00)	7Ts-24			ns
t9	XSCL period	4Ts-24			ns
t10	XSCL high width	2Ts-24			ns
t11	XSCL low width	2Ts-24			ns
t12	UD[3:0], LD[3:0] setup to XSCL falling edge	2Ts-24			ns
t13	UD[3:0], LD[3:0] hold from XSCL falling edge	2Ts-24			ns
t14	LP negated to XSCL rising edge (AUX[0D] bits 7,6 = 00)	5Ts-24			ns

7.18 LCD Interface Timing - 8-Bit Single/Dual Monochrome Panel

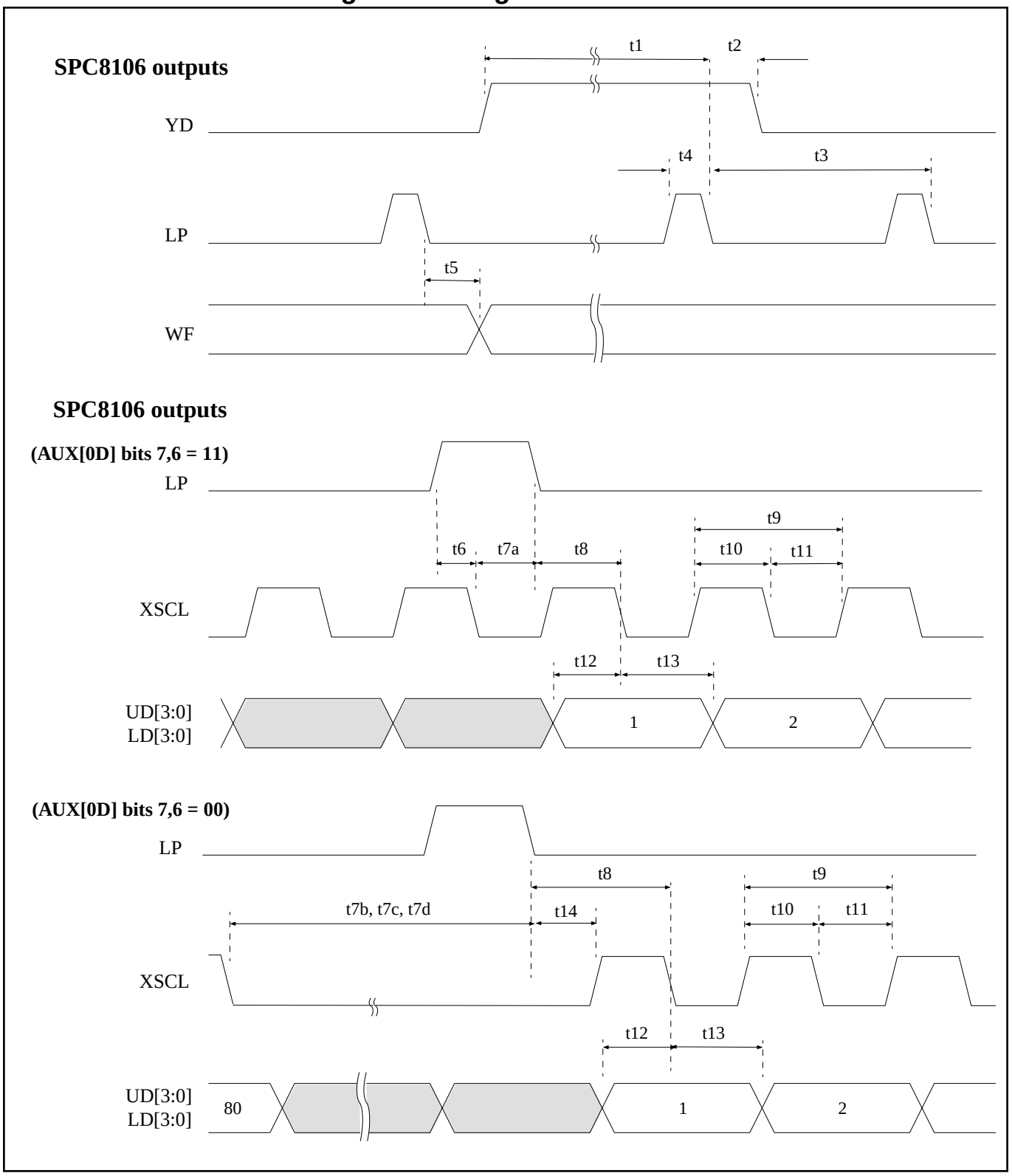


Figure 21 : LCD Interface Timing - 8-Bit Single/Dual Monochrome Panel

Table 0-37 LCD Interface Timing - 8-Bit Single/Dual Monochrome Panel

Symbol	Parameter	Min	Typ	Max	Units
t1a	YD setup to LP negated (single panel mode)	739Ts-24			ns
t1b	YD setup to LP negated (dual panel mode)	1478Ts - 24			ns
t2a	YD hold from LP negated (single panel mode)	13Ts-24			ns
t2b	YD hold from LP negated (dual panel mode)	13Ts-24			ns
t3a	LP period (single panel mode)		752Ts		ns
t3b	LP period (dual panel mode)		1504Ts		ns
t4a	LP pulse width (AUX[0D] bit 6 = 0)	5Ts-24			ns
t4b	LP pulse width (AUX[0D] bit 6 = 1)	6Ts-24			ns
t5	WF delay from LP negated	0		20	ns
t6a	LP setup to XSCL falling edge (AUX[0D] bit 7, 6 = 00)	n/a	n/a	n/a	ns
t6b	LP setup to XSCL falling edge (AUX[0D] bit 7, 6 = 11)	2Ts-24			ns
t7a	LP hold from XSCL falling edge (AUX[0D] bit 7, 6 = 11)	4Ts-24			ns
t7b	LP hold from XSCL falling edge (AUX[0D] bit 7, 6 = 00)	n/a	n/a	n/a	ns
t7c	XSCL falling edge to LP falling edge - single panel mode (AUX[0D] bit 7,6 = 00 only)	103Ts-24			ns
t7d	XSCL falling edge to LP falling edge - dual panel mode (AUX[0D] bit 7, 6 = 00 only)	206Ts-24			ns
t8a	LP negated to XSCL falling edge (AUX[0D] bits 7,6 = 00)	9Ts-24			ns
t8b	LP negated to XSCL falling edge (AUX[0D] bits 7,6 =11)	4Ts-24			ns
t9	XSCL period	8Ts-24			ns
t10	XSCL high width	4Ts-24			ns
t11	XSCL low width	4Ts-24			ns
t12	UD[3:0], LD[3:0] setup to XSCL falling edge	4Ts-24			ns
t13	UD[3:0], LD[3:0] hold from XSCL falling edge	4Ts-24			ns
t14a	LP negated to XSCL rising edge (AUX[0D] bits 7,6 = 00)	5Ts-24			ns
t14b	LP negated to XSCL rising edge (AUX[0D] bits 7,6 = 11)	0			ns

7.19 LCD Interface Timing - 16-Bit Single/Dual, 8-Bit Dual, 4-Bit Single Color Panels

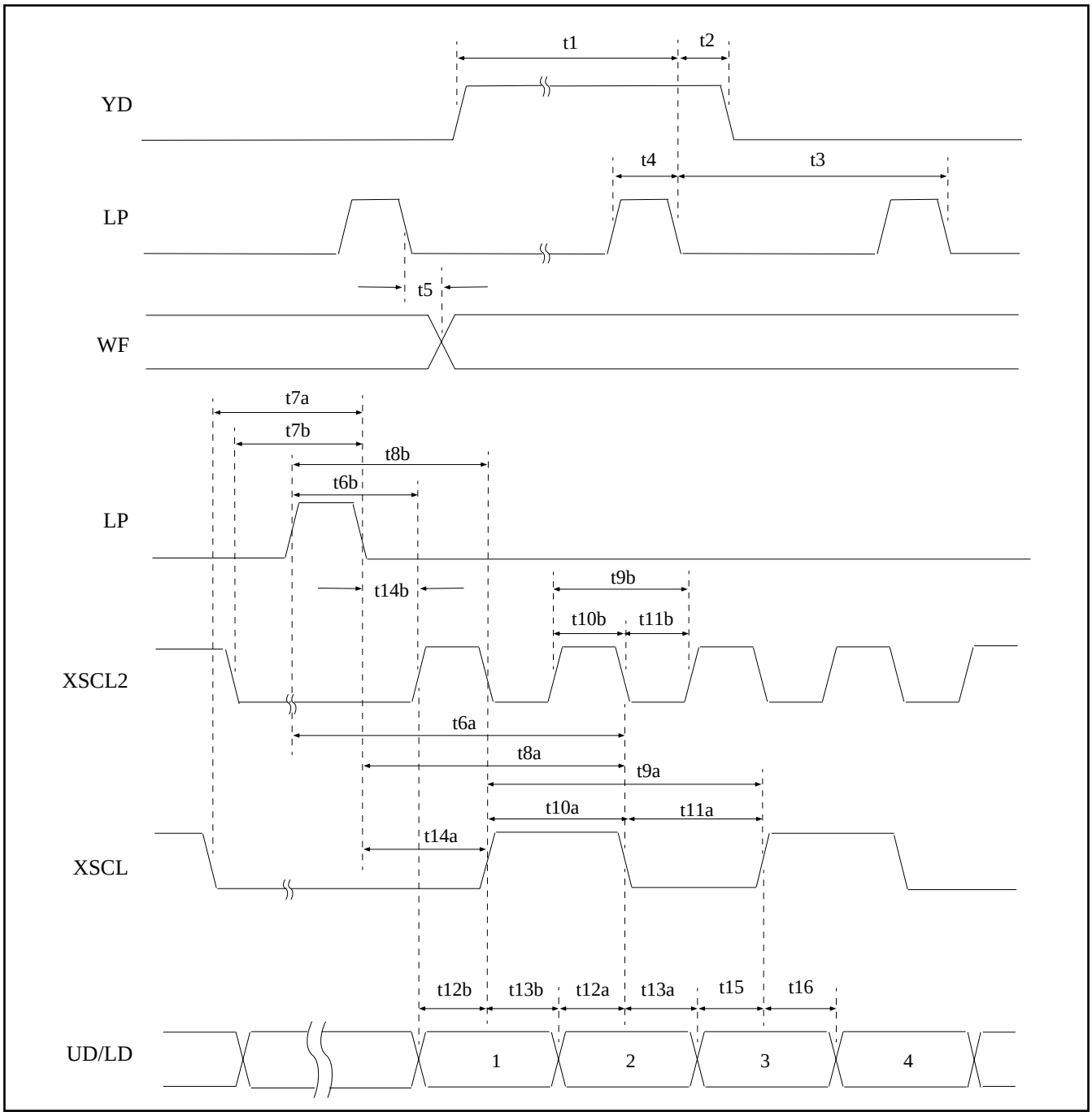


Figure 22 : LCD Interface Timing - 16-Bit Single/Dual, 8-Bit Dual, 4-Bit Single Color Panels

Table 0-38 LCD Interface Timing - 16-Bit Single/Dual, 8-Bit Dual, 4-Bit Single Color Panels

Symbol	Parameter	Min	Typ	Max	Units
t1	YD setup to LP negated	HDP + HNDP - 13Ts			ns
t2	YD hold from LP negated	13Ts - 24			ns
t3	LP period	HDP + HNDP			ns
t4	LP pulse width	5Ts - 10			ns
t5	WF delay from LP negated	0			ns
t6a	LP setup to XSCL falling edge	22Ts - 24			ns
t6b	LP setup to XSCL2 falling edge	19.5Ts - 10			ns
t7a	XSCL falling edge to LP falling edge	HNDP - 17Ts			ns
t7b	XSCL2 falling edge to LP falling edge	HNDP - 14.5Ts			ns
t8a	LP negated to XSCL falling edge	17Ts - 10			ns
t8b	LP negated to XSCL2 falling edge	14.5Ts - 10			ns
t9a	XSCL period	5Ts - 10		6Ts - 10	ns
t9b	XSCL2 period	2.5Ts - 10		3Ts - 10	ns
t10a	XSCL high width	2Ts - 10		3Ts - 10	ns
t10b	XSCL2 high width	Ts - 10		Ts - 10	ns
t11a	XSCL low width	3Ts - 10			ns
t11b	XSCL2 low width	1.5Ts - 10			ns
t12a	UD/LD setup to XSCL falling edge	1.5Ts - 10[a]			ns
t12b	UD/LD setup to XSCL2 falling edge	1.5Ts - 10			ns
t13a	UD/LD hold from XSCL falling edge	Ts - 10[a]			ns
t13b	UD/LD hold from XSCL2 falling edge	Ts - 10			ns
t14a	LP negated to XSCL rising edge	15Ts - 10			ns
t14b	LP negated to XSCL2 rising edge	13.5Ts - 10			ns
t15	UD/LD setup to XSCL rising edge	2Ts - 10[a]			ns
t16	UD/LD hold from XSCL rising edge	0.5Ts - 10[a]			ns

a. if AUX[06] b5=0 then t12=1.5Ts, t13=Ts, t15=2Ts, and t16=0.5Ts
if AUX[06] b5=1 then t12=1Ts, t13=1.5Ts, t15=1.5Ts, and t16=Ts

Ts = pixel clock period
HDP = horizontal display period in units of Ts (640 typical for single panel, 1280 typical for dual panel)
HNDP = horizontal non-display period in units of Ts (112 typical for single panel, 224 typical for dual panel)

7.20 LCD Interface Timing - 8-bit Single Color Panel

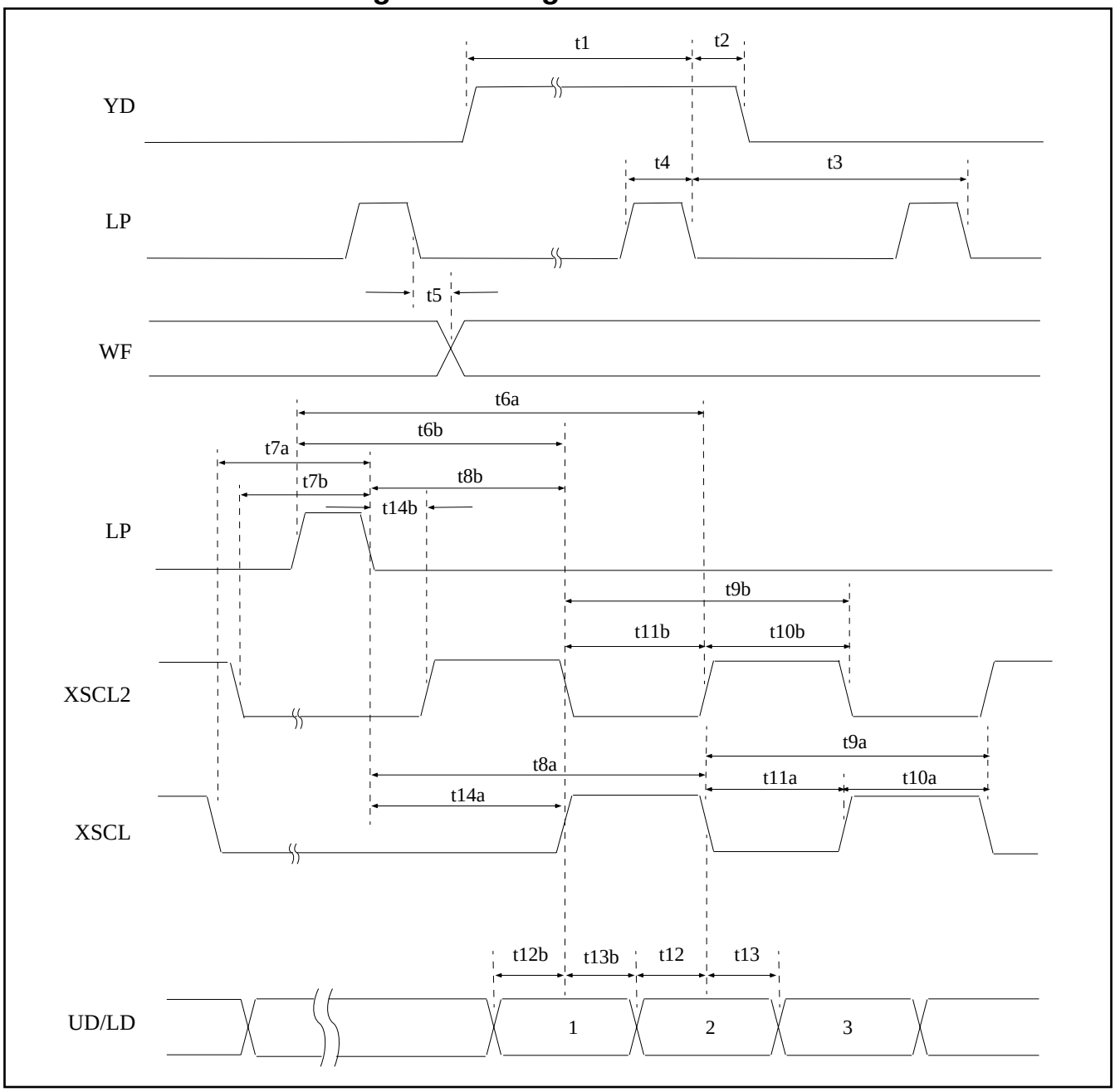


*Figure 23 : LCD Interface Timing - 8-bit Single Color Panel*source: 8106_c8s.can

Table 0-39 LCD Interface Timing - 8-bit Single Color Panel

Symbol	Parameter	Min	Typ	Max	Units
t1	YD setup to LP negated	HDP + HNDP - 13Ts			ns
t2	YD hold from LP negated	13Ts - 24			ns
t3	LP period	HDP + HNDP			ns
t4	LP pulse width	5Ts - 10			ns
t5	WF delay from LP negated	0			ns
t6a	LP setup to XSCL falling edge	22Ts - 24			ns
t6b	LP setup to XSCL2 falling edge	19.5Ts - 10			ns
t7a	XSCL falling edge to LP falling edge	HNDP - 17Ts			ns
t7b	XSCL2 falling edge to LP falling edge	HNDP - 14.5Ts			ns
t8a	LP negated to XSCL falling edge	17Ts - 10			ns
t8b	LP negated to XSCL2 falling edge	14.5Ts - 10			ns
t9a	XSCL period	4Ts - 10		6Ts - 10	ns
t9b	XSCL2 period	4Ts - 10		7Ts - 10	ns
t10a	XSCL high width	Ts - 10		2Ts - 10	ns
t10b	XSCL2 high width	Ts - 10		2Ts - 10	ns
t11a	XSCL low width	3Ts - 10			ns
t11b	XSCL2 low width	3Ts - 10			ns
t12a	UD/LD setup to XSCL falling edge	1.5Ts - 10			ns
t12b	UD/LD setup to XSCL2 falling edge	1.5Ts - 10			ns
t13a	UD/LD hold from XSCL falling edge	Ts - 10			ns
t13b	UD/LD hold from XSCL2 falling edge	1.5Ts - 10			ns
t14a	LP negated to XSCL rising edge	16Ts - 10			ns
t14b	LP negated to XSCL2 rising edge	13.5Ts - 10			ns

Ts = pixel clock period
HDP = horizontal display period in units of Ts (640 typical)
HNDP = horizontal non-display period in units of Ts (112 typical)

7.21 Interface Timing - TFT Panel

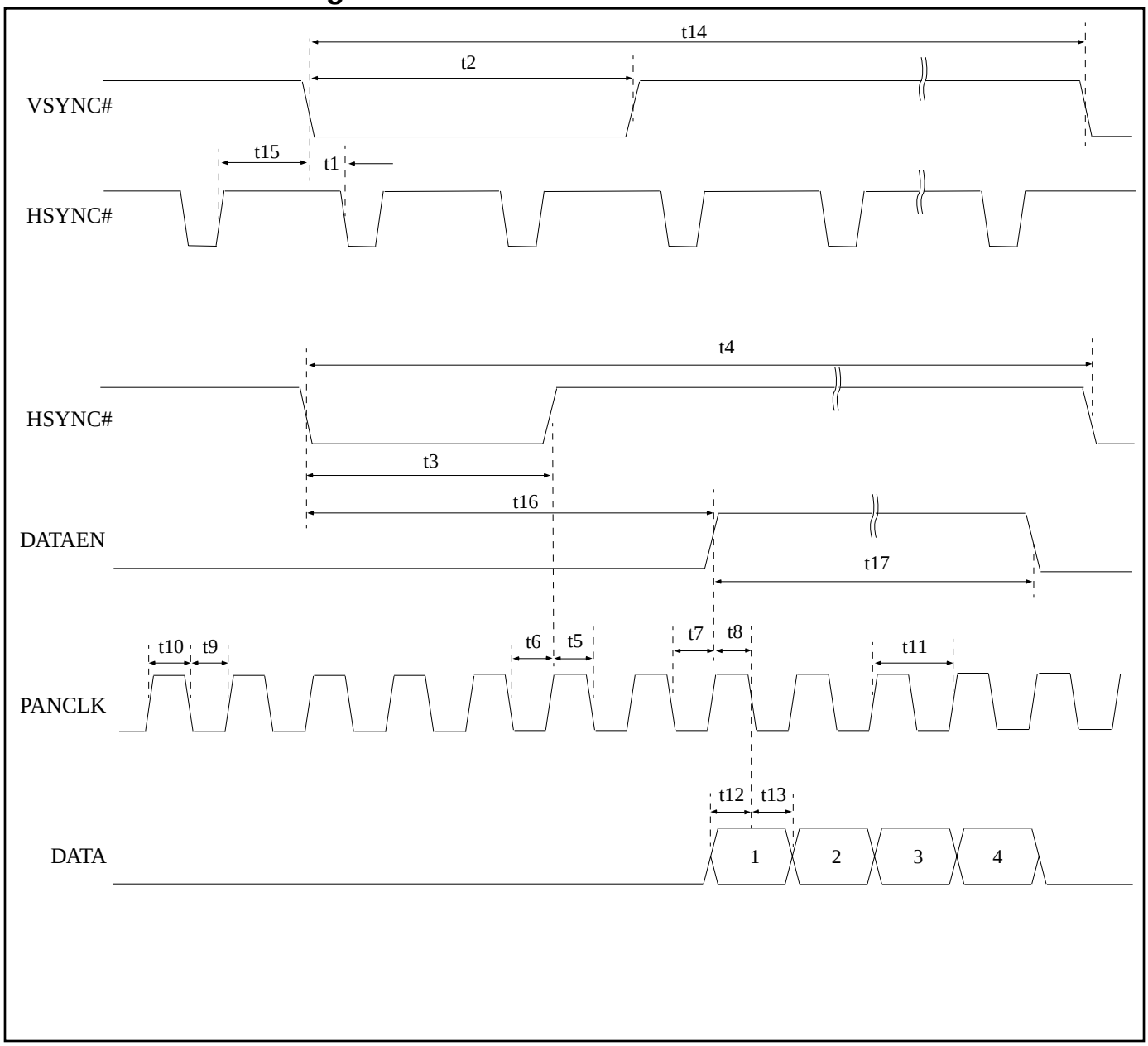


*Figure 24 : Interface Timing - TFT*source: 8106_tft_1.can*

Table 0-40 TFT Interface Timing when AUX[00]b5=1 and AUX[0B]b1=1

Symbol	Parameter	Min	Typ	Max	Units
t1	VSYNC# setup to HSYNC#	20			PANCLK
t2	VSYNC# Pulse Width		2		HSYNC
t3	HSYNC# Pulse Width		96		PANCLK
t4	Horizontal (HSYNC) Period		800		PANCLK
t5	HSYNC# Setup to PANCLK Low	0.4Ts-3			ns
t6	HSYNC# Hold from PANCLK Low	0.4Ts-3			ns
t7	DATAEN Hold from PANCLK Low	0.4Ts-3			ns
t8	DATAEN Setup to PANCLK Low	0.4Ts-1			ns
t9	PANCLK Low Pulse Width	0.4Ts-1			ns
t10	PANCLK High Pulse Width	0.4Ts-3			ns
t11	PANCLK Period	Ts-10			ns
t12	DATA Setup to PANCLK Low	0.4Ts-2			ns
t13	DATA Hold from PANCLK Low	0.4Ts-1			ns
t14	VSYNC Period		525		HSYNC
t15	VSYNC# Hold from HSYNC# Active	780			PANCLK
t16	DATAEN High from HSYNC# Active		144		PANCLK
t17	DATAEN High Width		640		PANCLK

Table 0-41 TFT Interface Timing when AUX[00]b5=1 and AUX[0B]b1=0

Symbol	Parameter	Min	Typ	Max	Units
t1	VSYNC# setup to HSYNC#	20PANCLK-10			ns
t2	VSYNC# Pulse Width		2		HSYNC
t3	HSYNC# Pulse Width		HNDP		PANCLK
t4	Horizontal (HSYNC) Period		HNDP +160		PANCLK
t5	HSYNC# Setup to PANCLK Low	0.4Ts-3			ns
t6	HSYNC# Hold from PANCLK Low	0.4Ts-3			ns
t7	DATAEN Hold from PANCLK Low	0.4Ts-3			ns
t8	DATAEN Setup to PANCLK Low	0.4Ts-1			ns
t9	PANCLK Low Pulse Width	0.4Ts-1			ns
t10	PANCLK High Pulse Width	.4Ts			ns
t11	PANCLK Period	Ts-10			ns
t12	DATA Setup to PANCLK Low	0.4Ts-2			ns
t13	DATA Hold from PANCLK Low	0.4Ts-1			ns
t14	VSYNC Period		525		HSYNC
t15	VSYNC# Hold from HSYNC# Active	780			PANCLK
t16	DATAEN High from HSYNC# Active		HNDP		PANCLK
t17	DATAEN High Width		640		PANCLK

Ts = pixel clock period

HNDP = Horizontal Non-Display Period

8.0 LCD INTERFACE OPTIONS

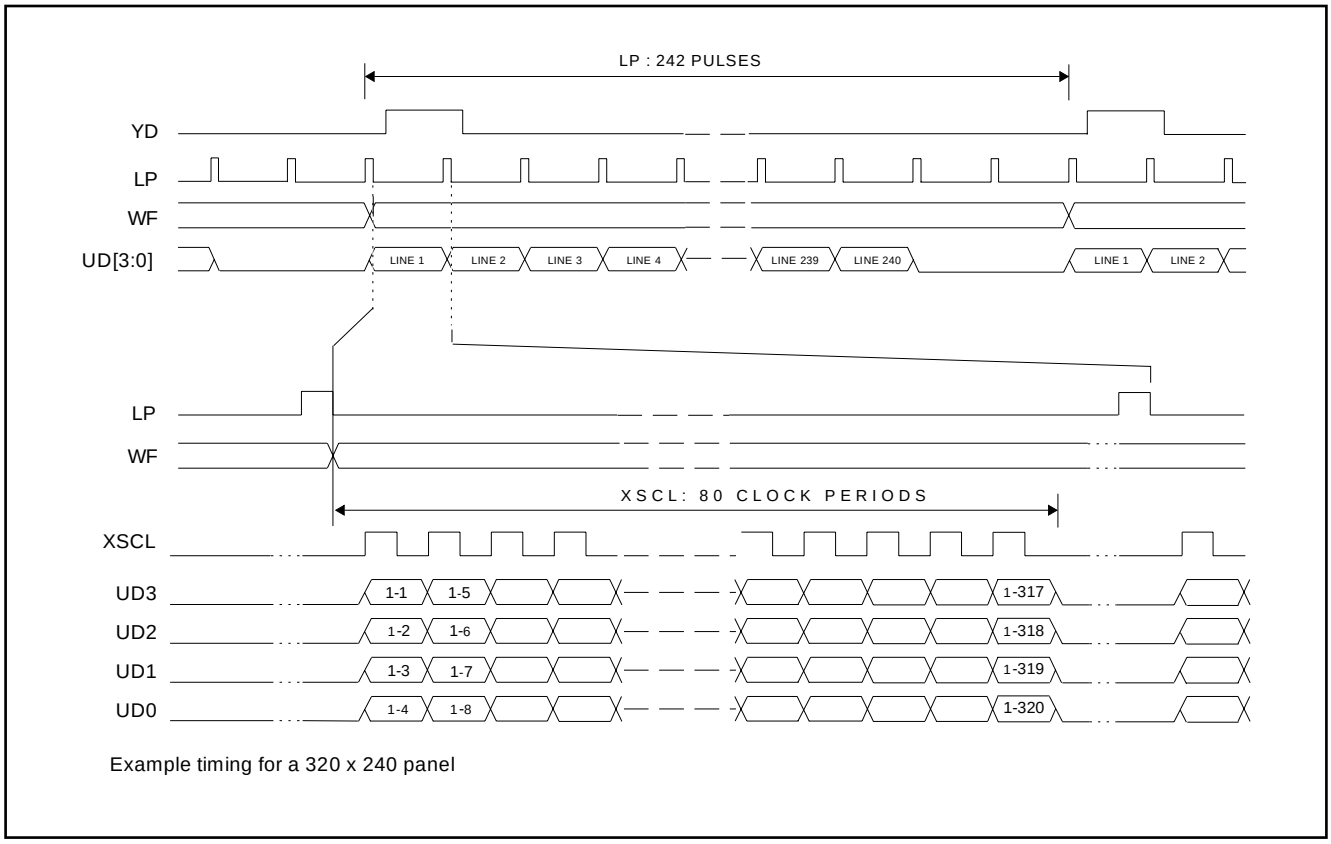


Figure 25 : 4-bit Single Monochrome Panel Timingsource: 06l_4sm.drw

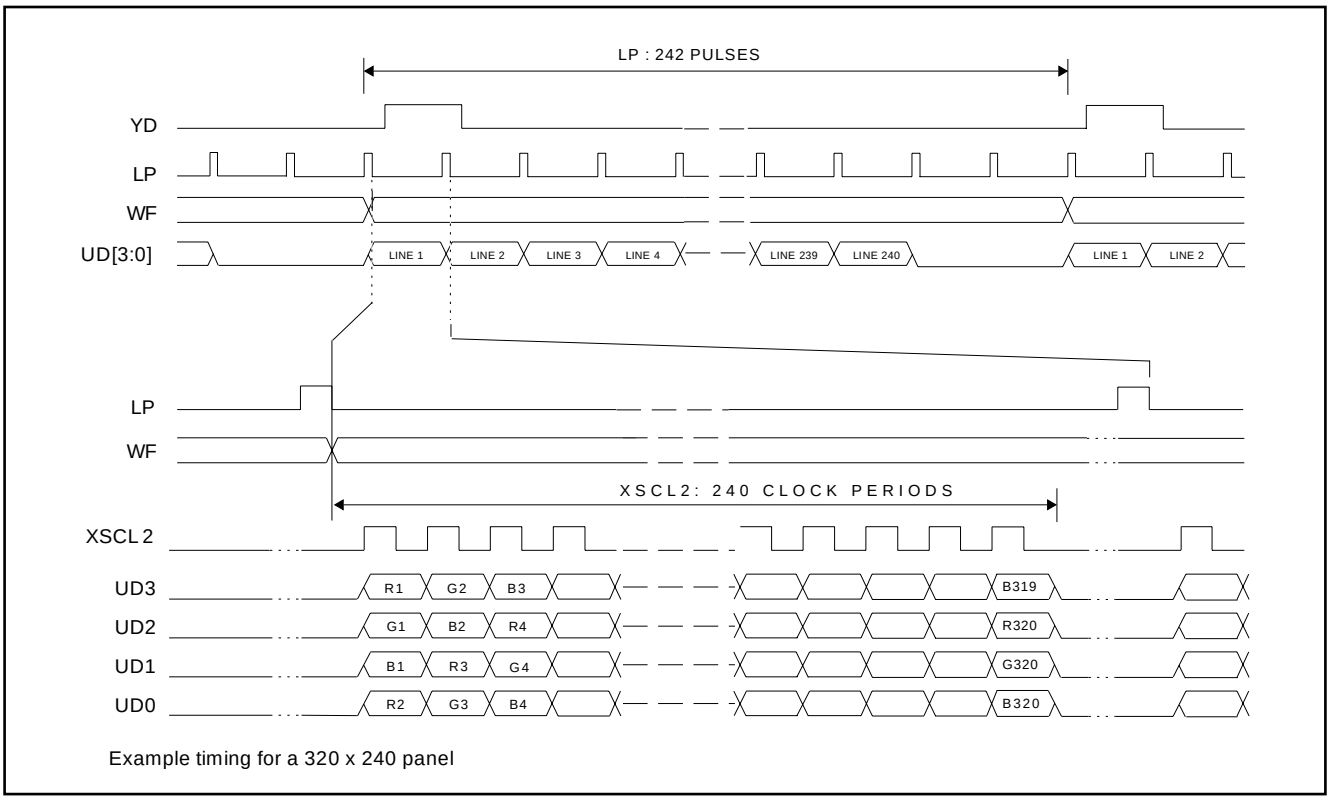


Figure 26 : 4-bit Single Color Panel Timing

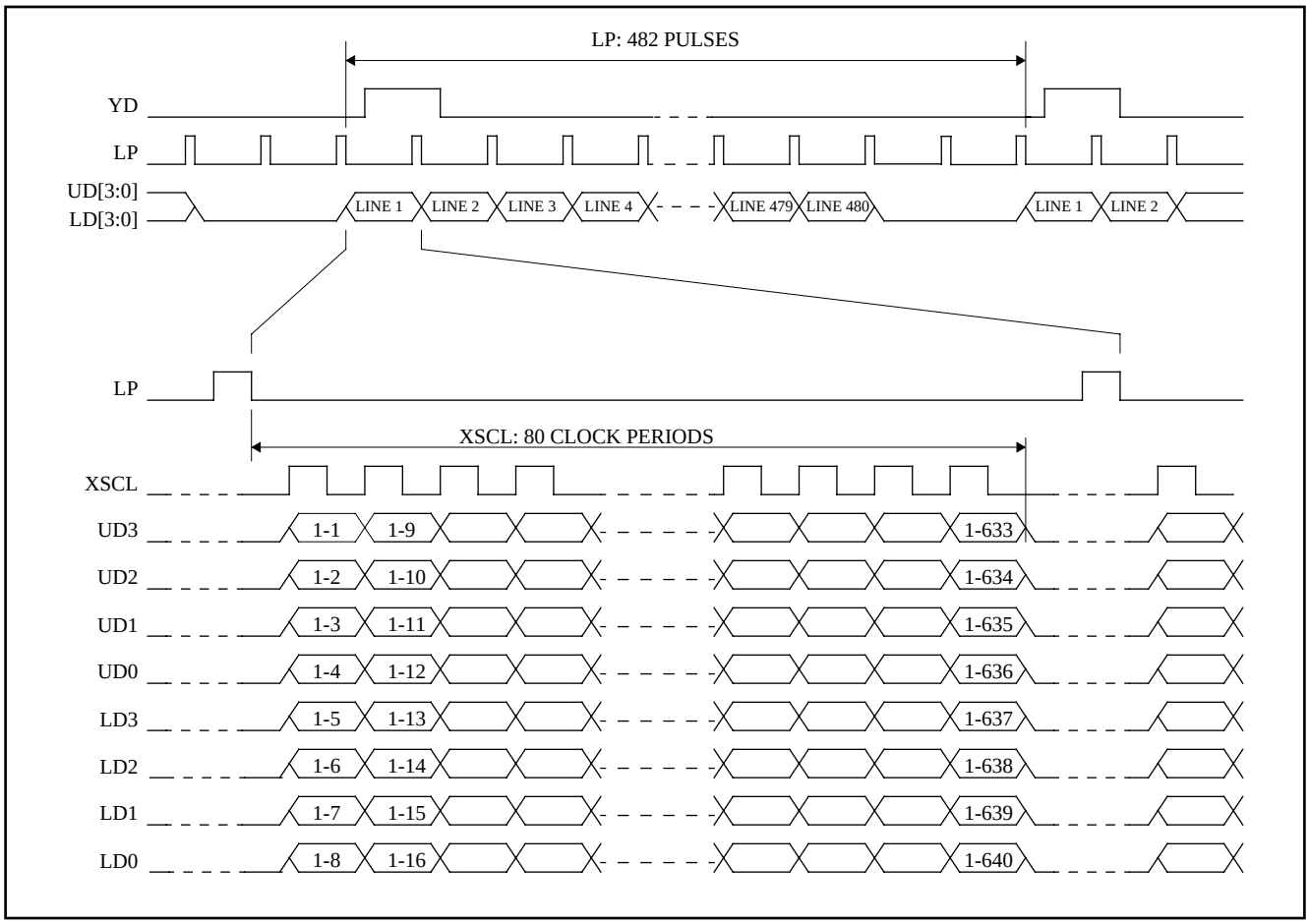


Figure 27 : 8-bit Single Monochrome Panel Timing

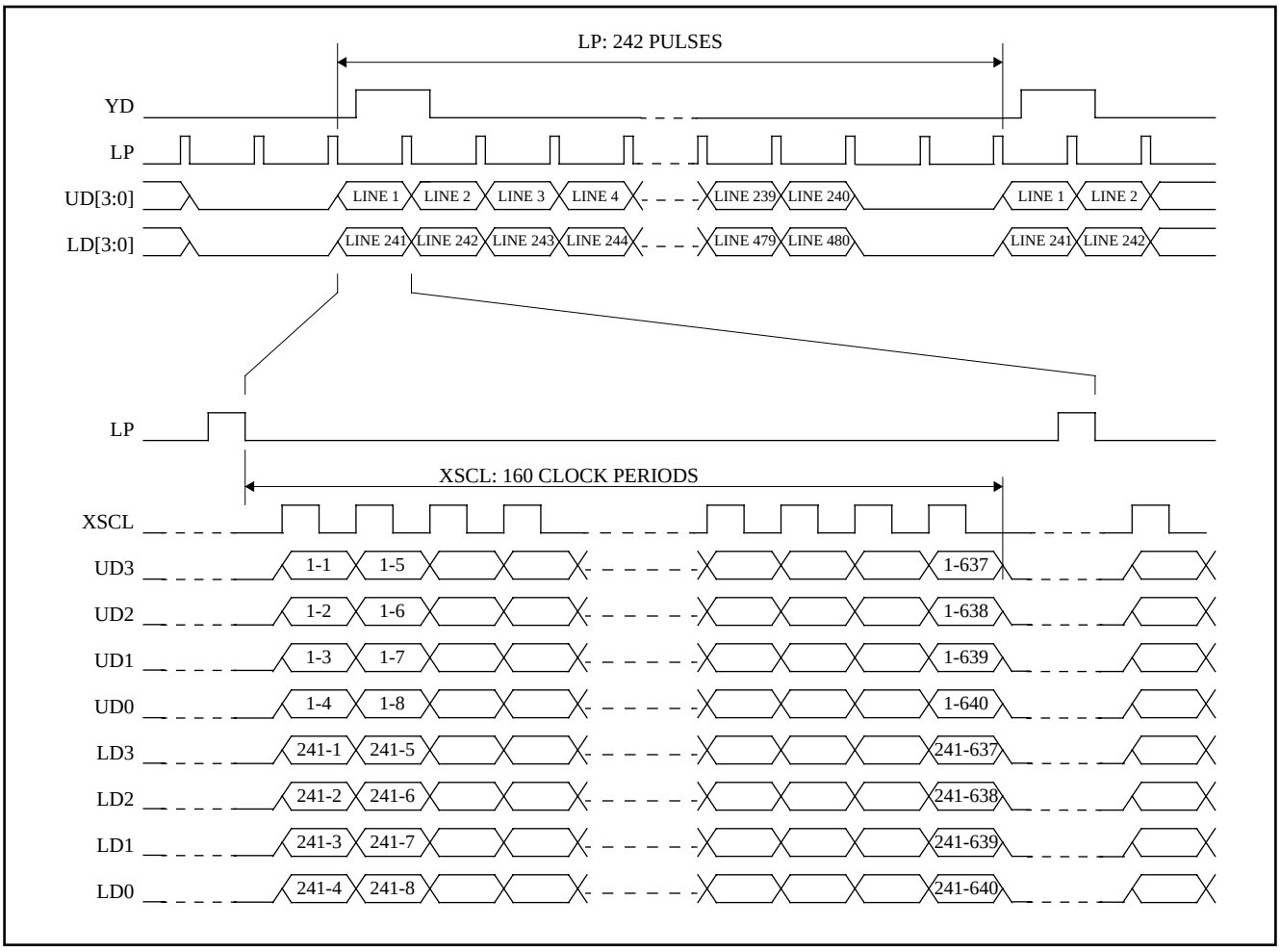


Figure 28 : 8-bit Dual Monochrome Panel Timing

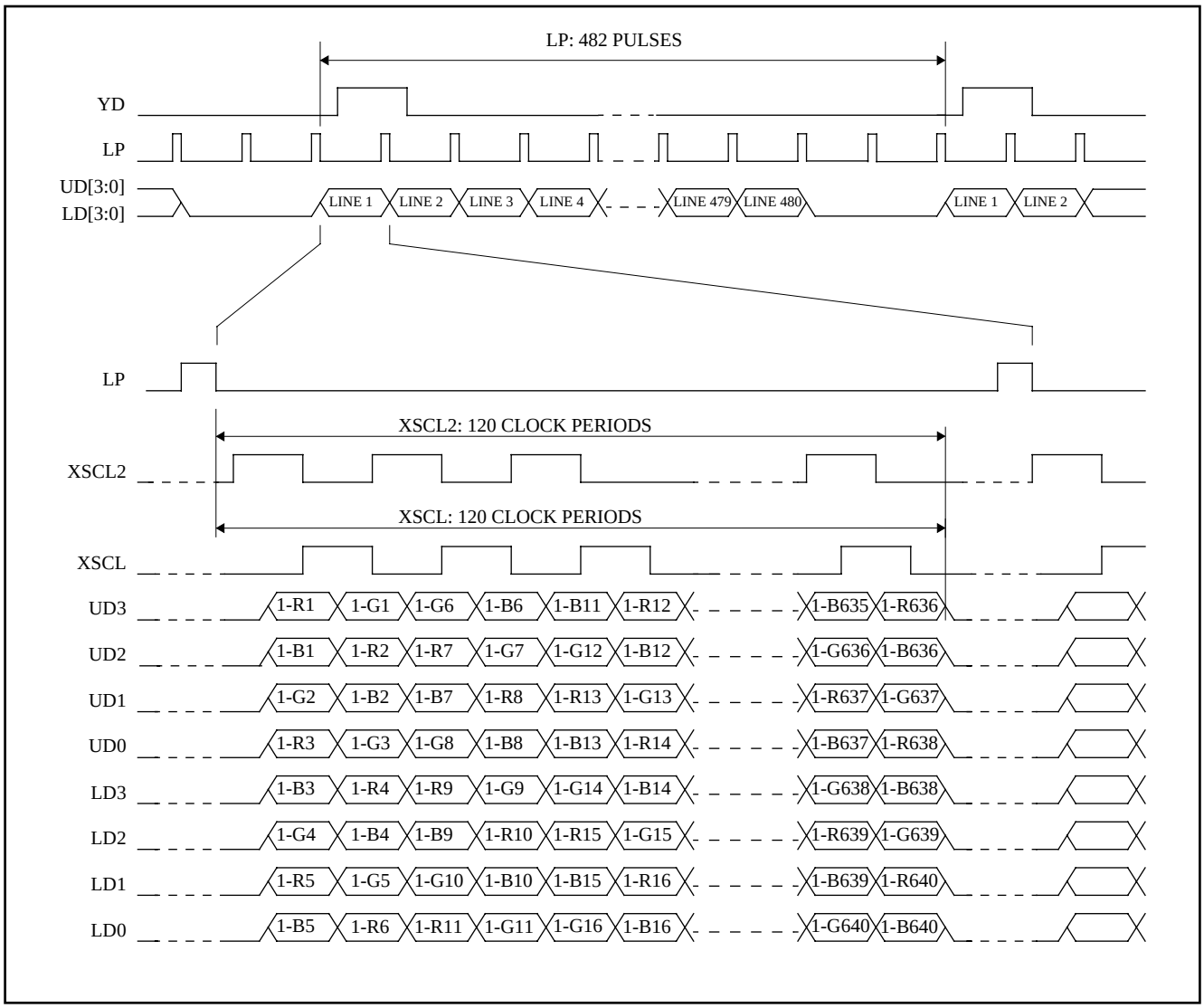


Figure 29 : 8-bit Single Color Panel Timing

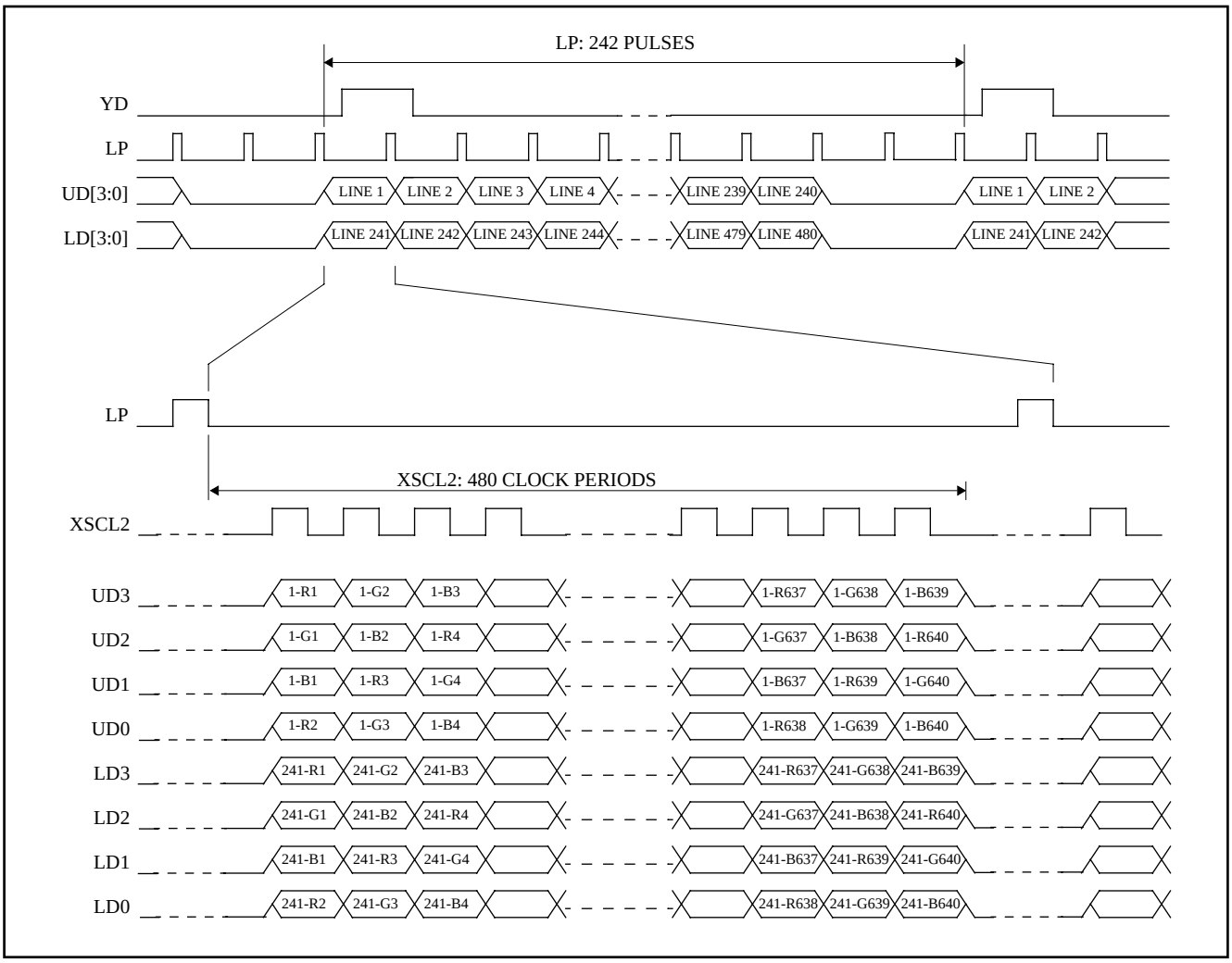


Figure 30 : 8-bit Dual Color Panel Timing

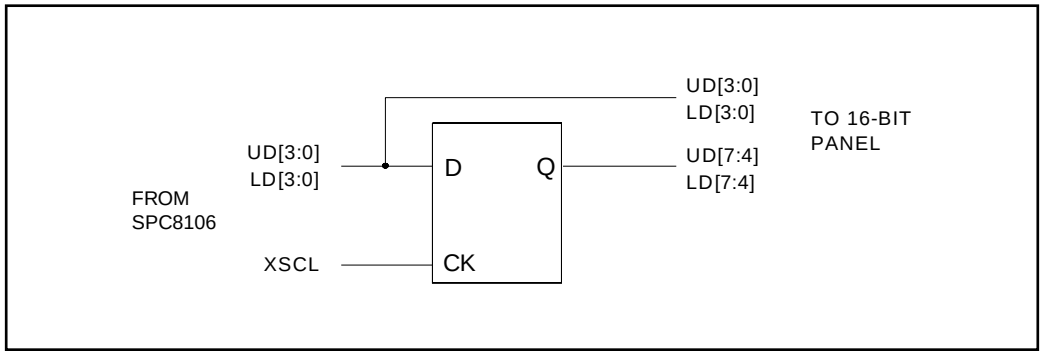


Figure 31 : External Circuit Required for 16-Bit Panel When MD[7]=1 at RESET

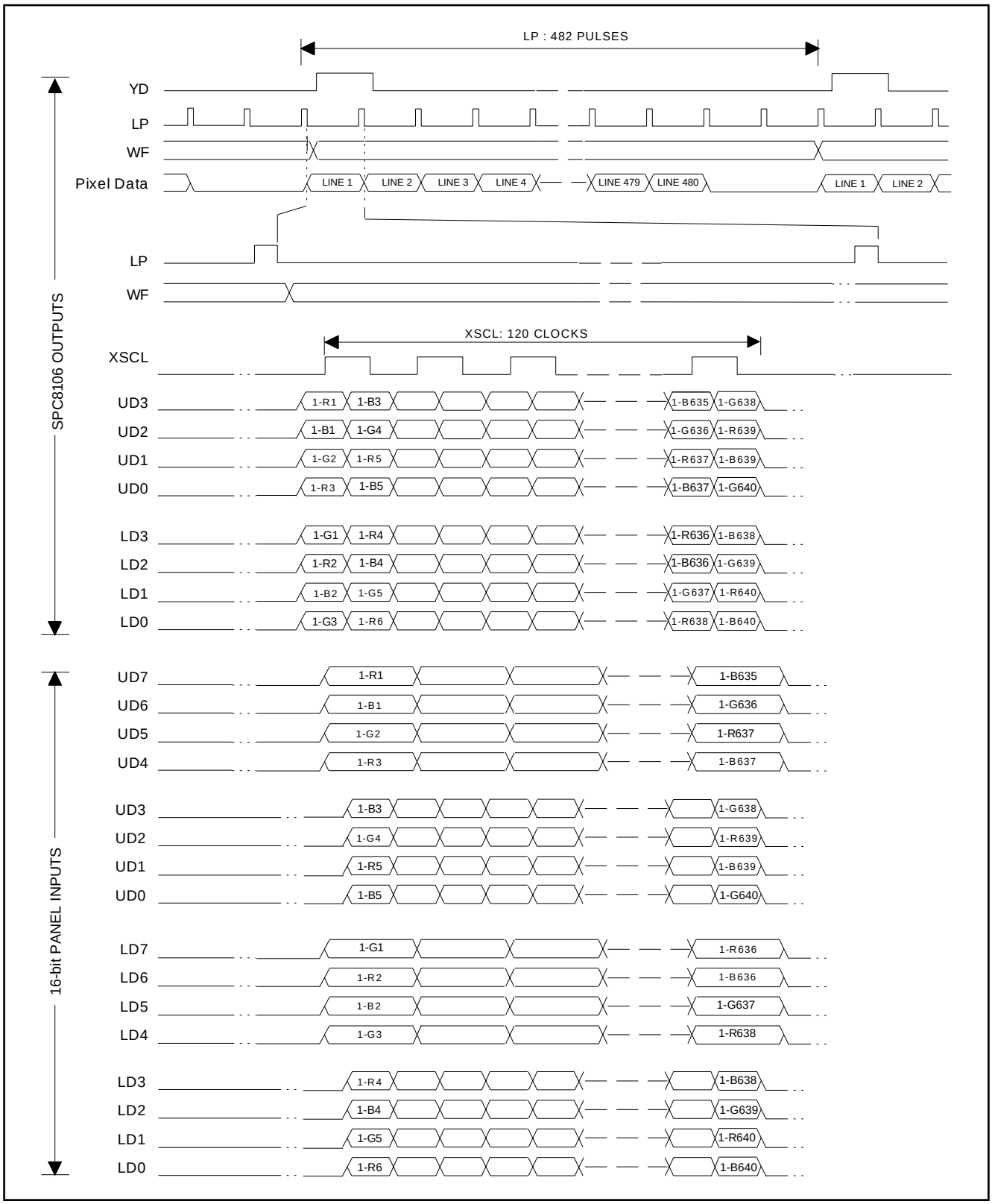


Figure 32 : 16-bit Single Color Panel Timing with External Circuit

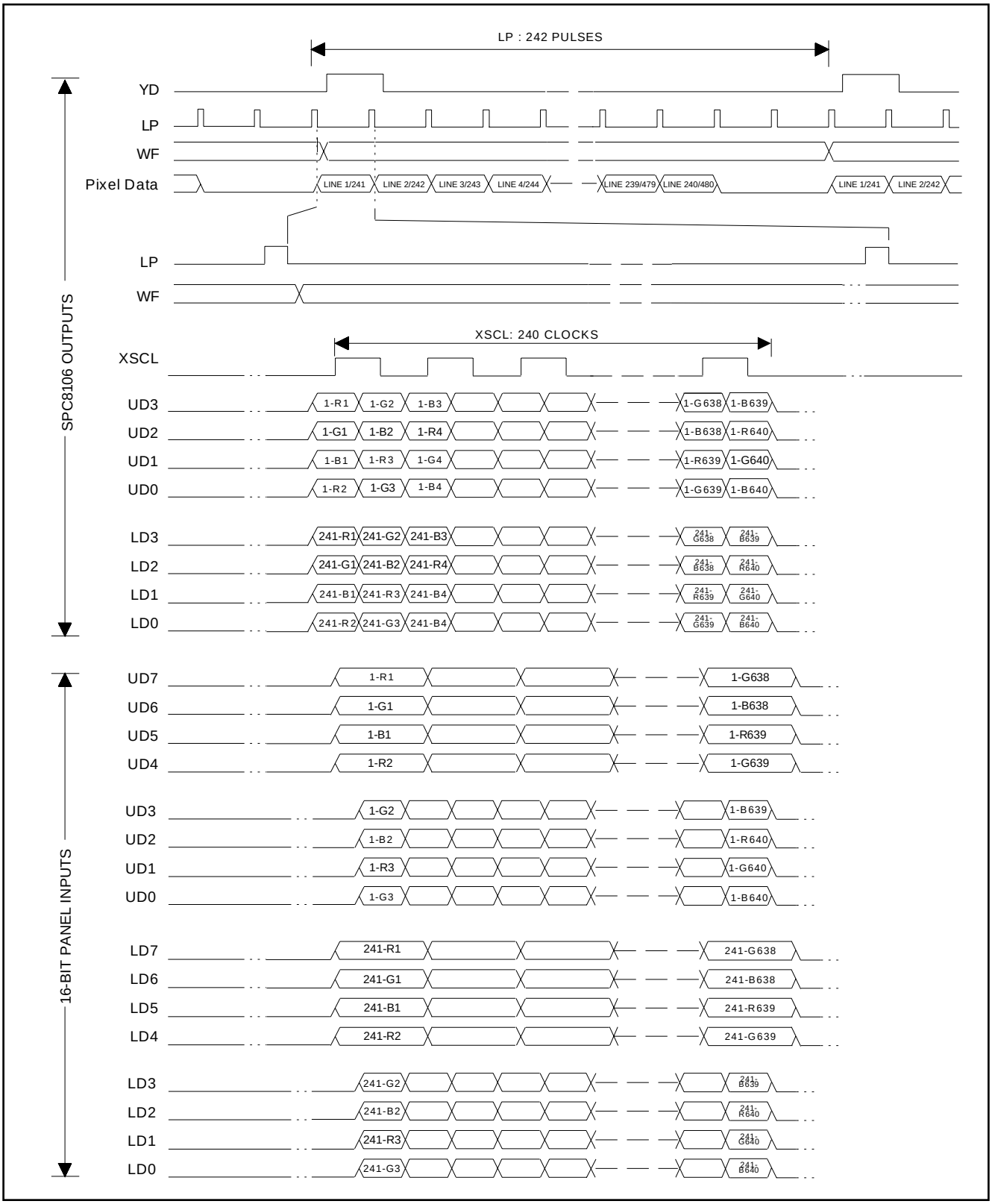


Figure 33 : 16-bit Dual Color Panel Timing with External Circuit

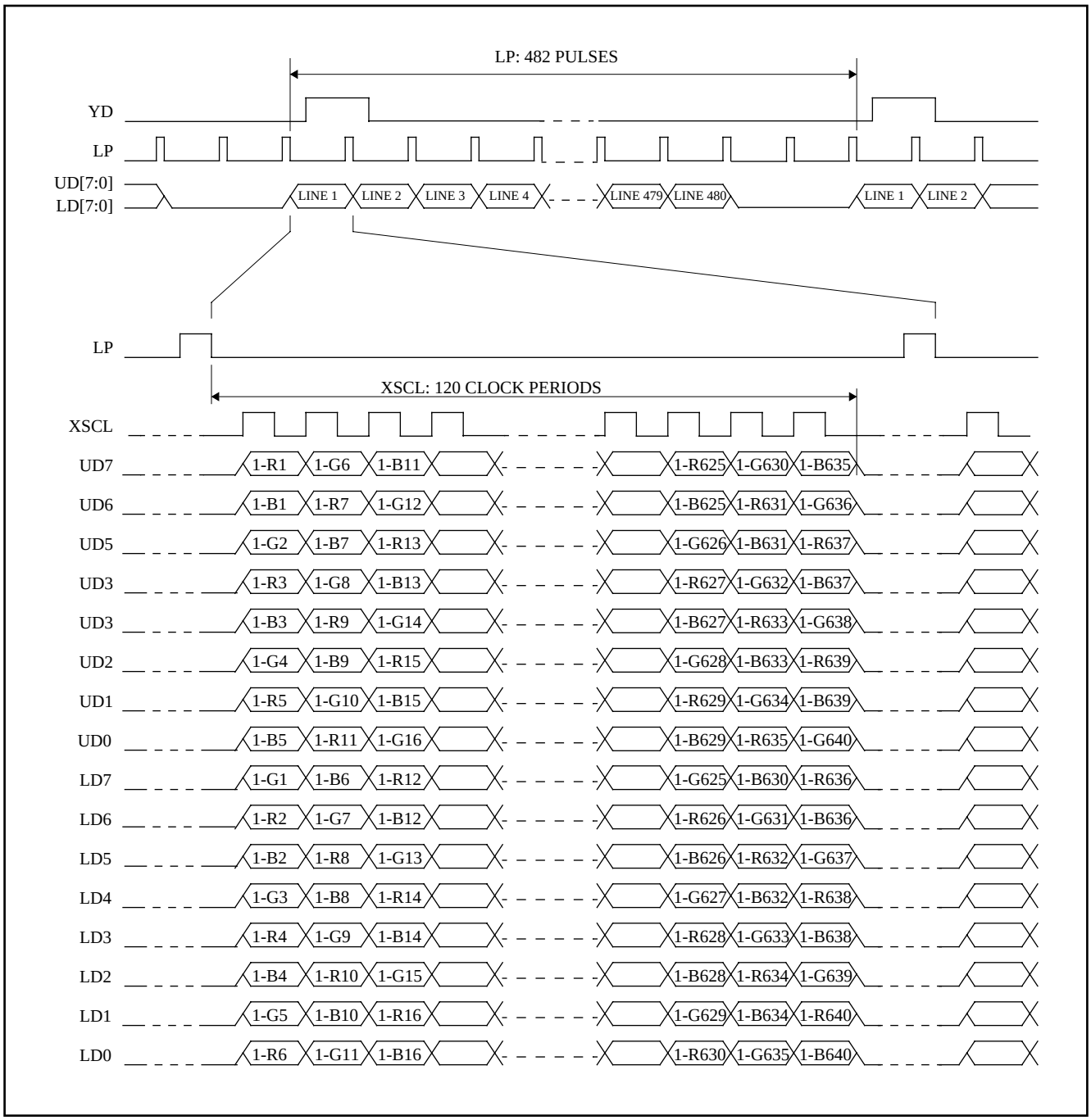


Figure 34 : 16-bit Single Color Panel Timing without External Circuit

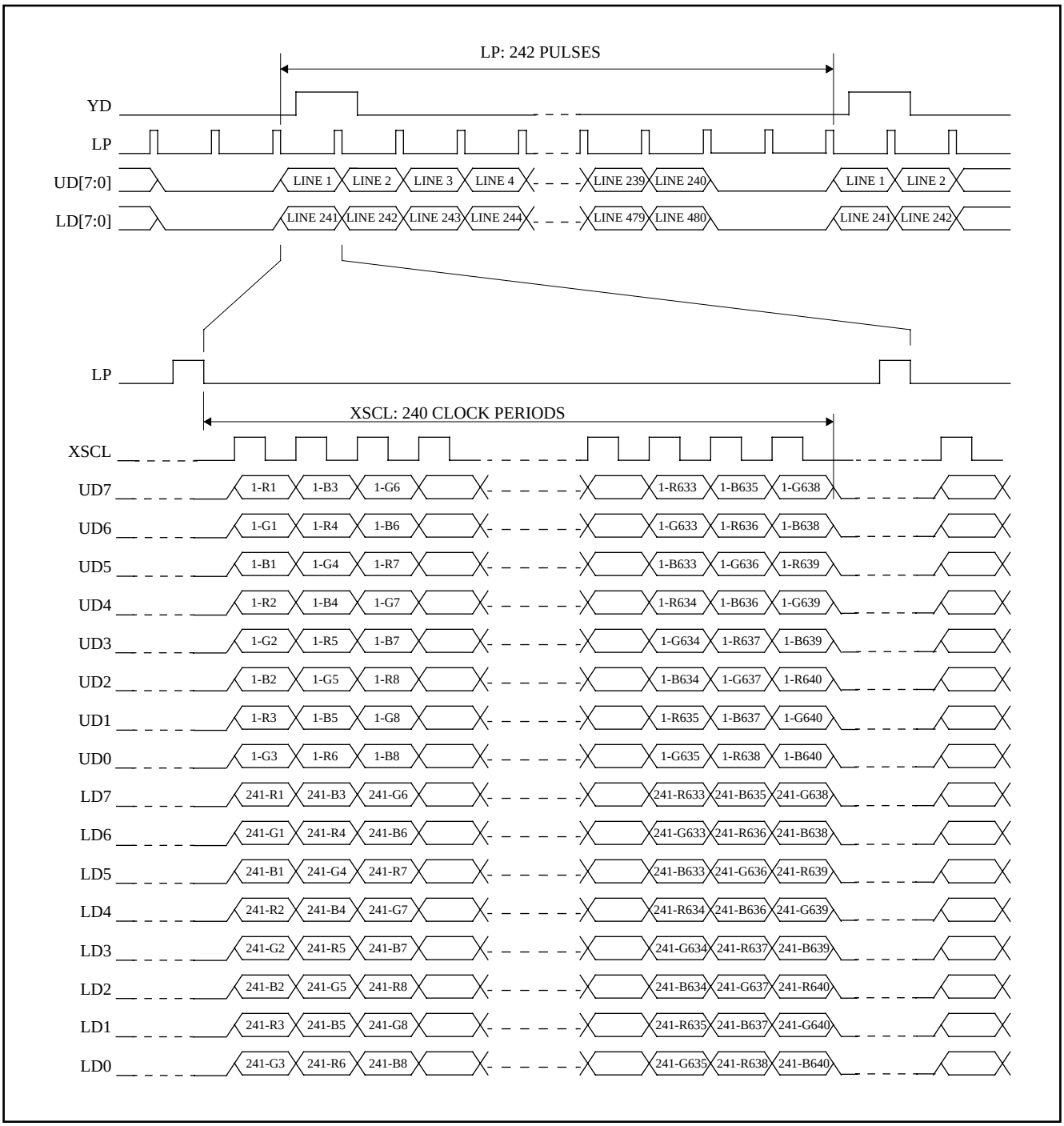
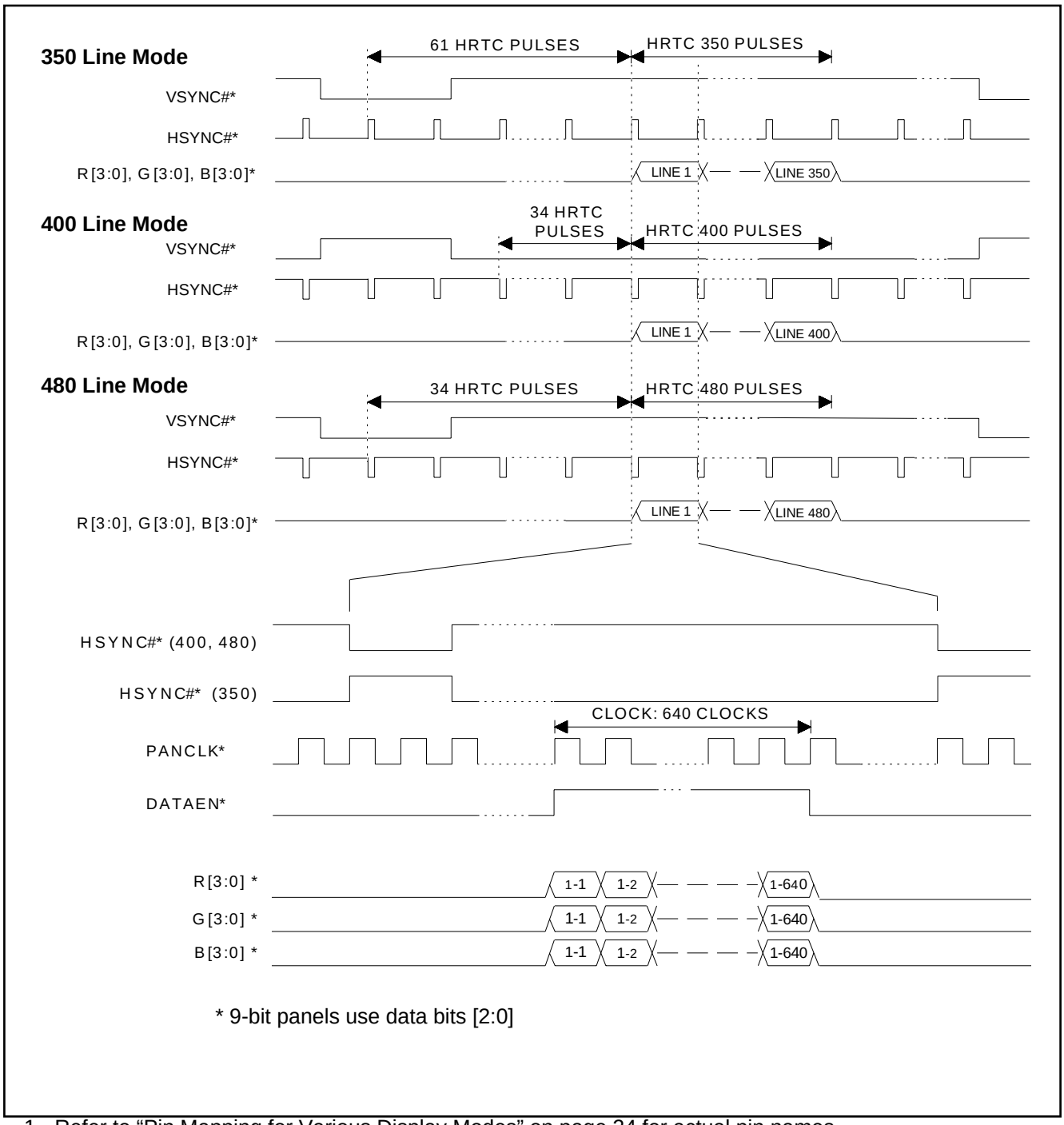
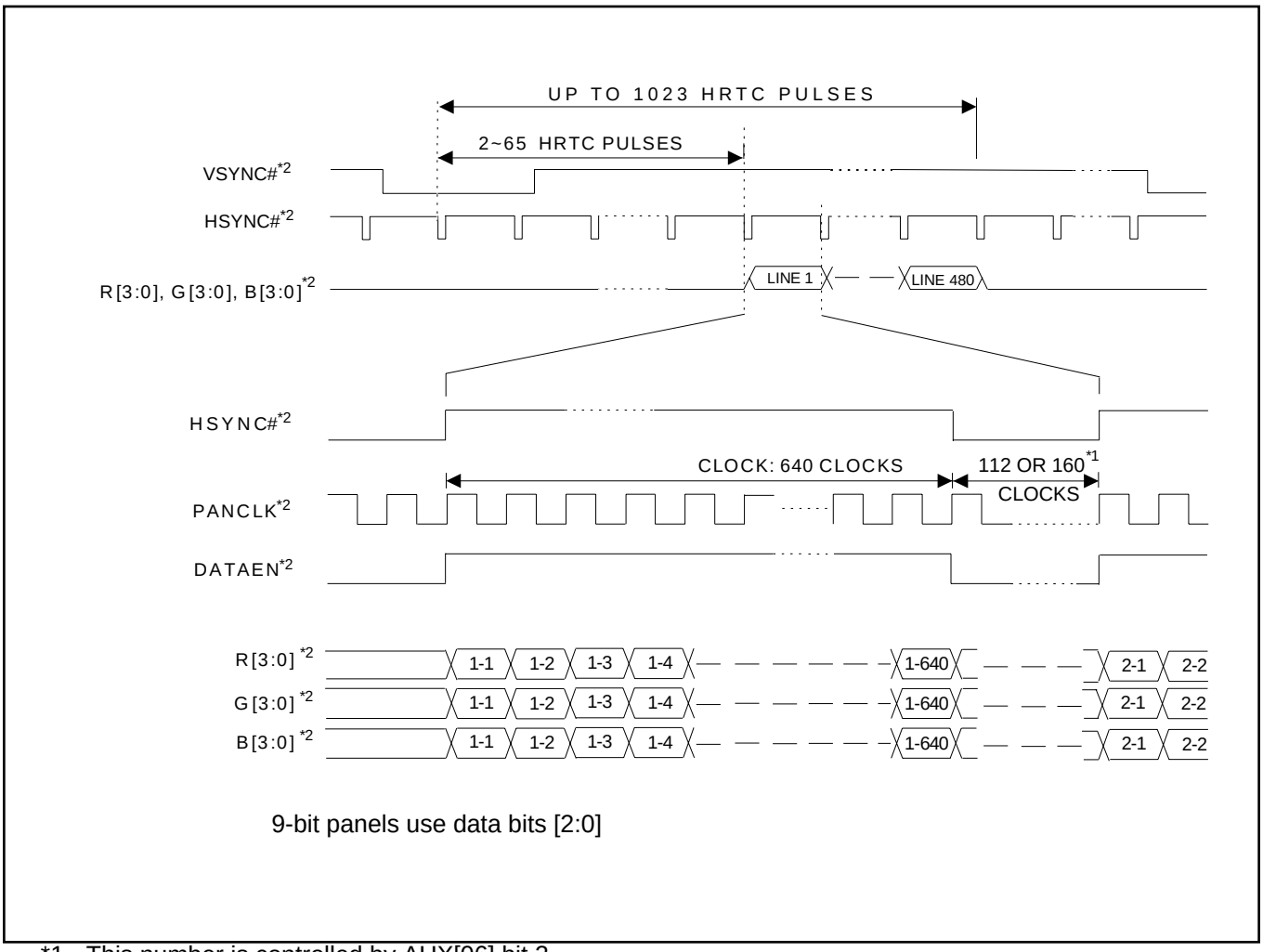


Figure 35 : 16-bit Dual Color Panel Timing without External Circuit



1 - Refer to “Pin Mapping for Various Display Modes” on page 24 for actual pin names

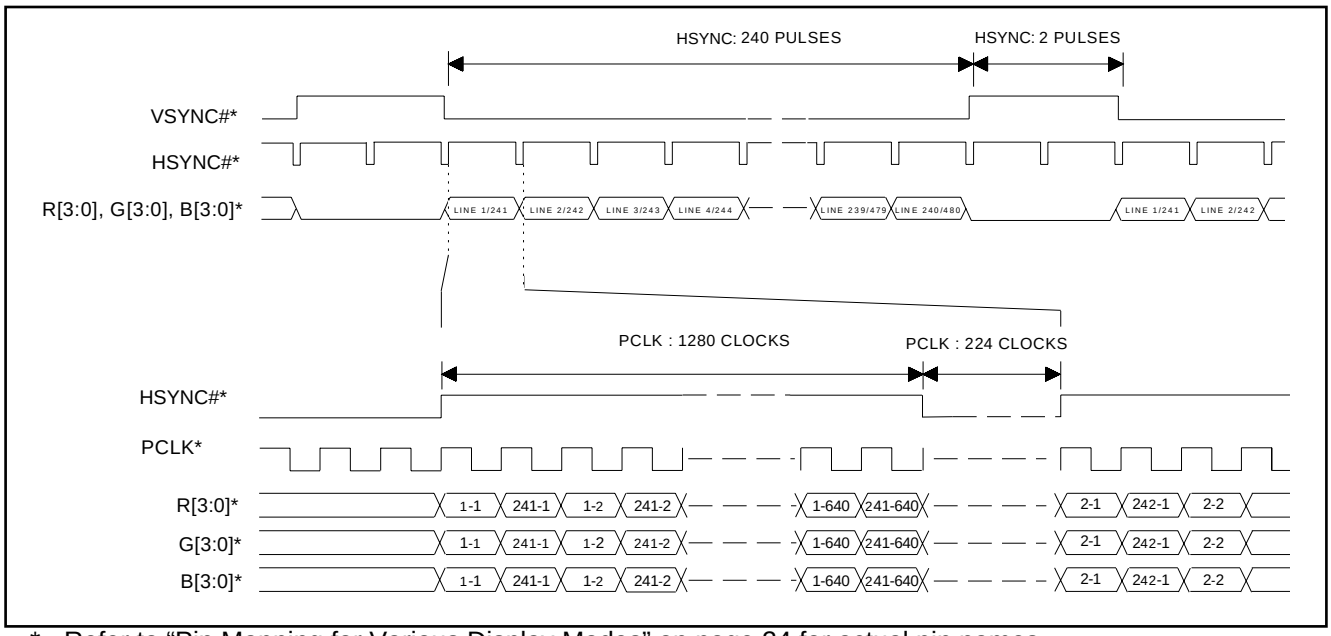
Figure 36 : Color TFT Panel Timing (when AUX[00]b5=1 and AUX[0B]b1=1)



*1 - This number is controlled by AUX[06] bit 2

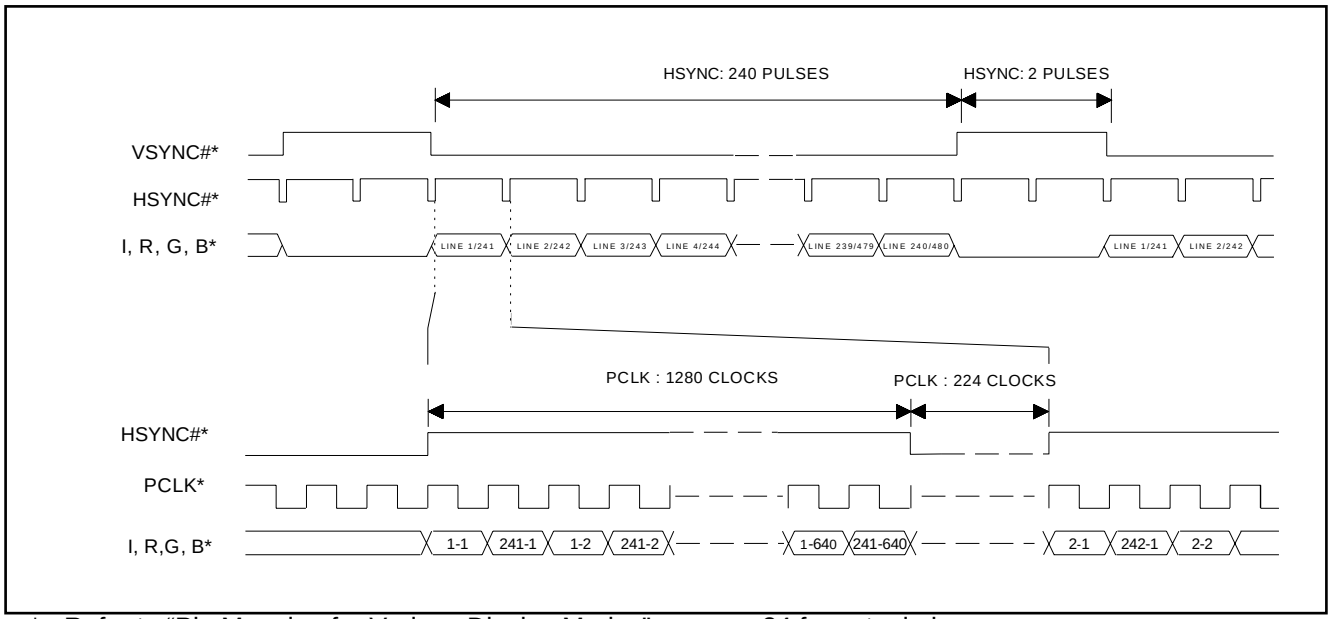
*2 - Refer to “Pin Mapping for Various Display Modes” on page 24 for actual pin names

Figure 37 : Color TFT Panel Timing (when AUX[00]b5=1 and AUX[0B]b1=0)



* - Refer to “Pin Mapping for Various Display Modes” on page 24 for actual pin names

Figure 38 : 12-bit RGB Mode LCD Panel Timing



* - Refer to “Pin Mapping for Various Display Modes” on page 24 for actual pin names

Figure 39 : RGBI Mode Dual LCD Panel Timing

9.0 POWER SAVE MODES

9.1 Power Save Modes

To accommodate the important need for power reduction in sub-notebook and palmtop computers, one hardware controlled and five software controlled Power Save Modes have been incorporated into the SPC8106. Options for these Power Save Modes can be enabled by setting bits in various Auxiliary registers, allowing flexibility in tailoring of the power reduction scheme to any given system implementation.

Software Power Save Modes

The Power Save Mode bits in the Power Save Register, AUX[03], select one of the six software Power Save Modes as shown in the table below. Note that if hardware Suspend mode is activated, the software power save mode setting in this register is ignored (i.e. hardware Suspend mode overrides the software power save modes). If these bits are set to a values 110b or 111b, then the chip will remain in normal active mode. The following descriptions outline the major functions of each power save mode. Some differences in power save mode operation exist between LCD and CRT display modes - see the following tables and notes for more information.

Table 0-42 Software Power Save Modes

Power Save Mode Select			Mode Activated
bit 2	**bit 1**	**bit 0**	
0	0	0	Normal Operation
0	0	1	Power Save Mode 1 enable
0	1	0	Power Save Mode 2 enable (toggle between states 1 & 2, see below)
0	1	1	Power Save Mode 3 enable
1	0	0	Power Save Mode 4 enable
1	0	1	Power Save Mode 5 enable (n/a for CRT mode)
1	1	0	Power Save Mode 6 enable (n/a for CRT mode)

Software Power Save Mode 1

- No video display accesses to display memory.
- Sequencer is dedicated to CPU accesses to/from display memory.
- Display memory refresh is maintained and is generated from active CLKI input (28 MHz for LCD, 25 MHz or 28 MHz for CRT). Refresh rate can be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- I/O read/write of all registers is allowed.
- LCDPWR# and IREFEN# signals forced high.
- LCD and TFT interface output signals tri-stated or forced low, depending on the state of the LCD Signal PS Mode bit in LCD Support Register 1, AUX[02] bit 1.
- If CRT display enabled, CRT and RAMDAC interface signals forced low (except DACRD# and DACWR# which remain active).

Software Power Save Mode 2

Power Save Mode 2 has two states. Initially when Power Save Mode 2 is set, the chip enters State 1. If no display memory read or write is detected for about two horizontal lines (approximately 63.5 μs), the chip enters State 2. If a display memory read or write is requested while in State 2, the chip returns to State 1 to service the display memory access within 3 clock periods of the active CLKI input clock.

State 1

- No video display accesses to display memory.
- Sequencer is dedicated to CPU accesses to/from display memory.
- Display memory refresh is maintained and is generated from the active CLKI input (28 MHz for LCD, 25 MHz or 28 MHz for CRT). Refresh rate can be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- I/O read/write of all registers is allowed.
- LCDPWR# and IREFEN# signals forced high.
- LCD and TFT interface output signals tri-stated or forced low, depending on the state of the LCD Signal PS Mode bit in LCD Support Register 1, AUX[02] bit 1.
- If CRT display enabled, CRT and RAMDAC interface signals forced low (except DACRD# and DACWR# which remain active).

State 2

- 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 the active CLKI input (28 MHz for LCD, 25 MHz or 28 MHz for CRT). Refresh rate can be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- I/O read/write of all registers is allowed.
- LCDPWR# and IREFEN# signals forced high.
- LCD and TFT interface output signals tri-stated or forced low, depending on the state of the LCD Signal PS Mode bit in LCD Support Register 1, AUX[02] bit 1.
- If CRT display enabled, CRT and RAMDAC interface signals forced low (except DACRD# and DACWR# which remain active).

Software Power Save Mode 3

- No video display accesses to display memory.
- No CPU accesses to/from display memory.
- Sequencer is halted.
- No display memory refresh.
- I/O read/write of all registers is allowed (except LUT and RAMDAC registers).
- LCDPWR# and IREFEN# signals forced high.
- LCD and TFT interface output signals tri-stated or forced low, depending on the state of the LCD Signal PS Mode bit in LCD Support Register 1, AUX[02] bit 1.
- If CRT display enabled, CRT and RAMDAC interface signals forced low (except DACRD# and DACWR# which are forced high).

Options

- I/O read/write to all registers except Auxiliary Registers can be disabled.
- The active internal clock oscillator cell can be disabled if a 2-terminal crystal is used. Note that the non-selected internal clock oscillator is automatically disabled in all active and power-save modes.

Software Power Save Mode 4

- 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 (28 MHz for LCD, 25 MHz or 28 MHz for CRT), 2) from the PDCLK pin (32 kHz 50% duty cycle, or 64 kHz with short high pulse duration), 3) or from a clock source connected to pin MEMEN.
- Refresh rate generated from CLKI can be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- Refresh rate generated from MEMEN or PDCLK can also be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- I/O read/write of all registers is allowed (except LUT and RAMDAC registers).
- LCDPWR# and IREFEN# signals forced high.
- LCD and TFT interface output signals tri-stated or forced low, depending on the state of the LCD Signal PS Mode bit in LCD Support Register 1, AUX[02] bit 1.
- If CRT display enabled, CRT and RAMDAC interface signals forced low (except DACRD# and DACWR# which are forced high).

Options

- I/O read/write to all registers except Auxiliary Registers can be disabled.
- 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.
- The active internal clock oscillator cell can be disabled if a 2-terminal crystal is used. Note that the non-selected internal clock oscillator is automatically disabled in all active and power-save modes.

Software Power Save Mode 5 (LCD Mode Only)

- Video display accesses to display memory allowed.
- CPU accesses to/from display memory allowed.
- Display memory refresh as in normal active mode.
- I/O read/write of all registers is allowed.
- LCDPWR# signal remains low (i.e. panel power enabled) and video display remains visible on LCD.
- Internal LUT disabled.
- Displayed gray-shades reduced to 3 levels on monochrome LCD (black, white, 50% gray).
- Displayed colors reduced to 3 colors on color LCD (black, green, light green).
- This power save mode not available when CRT display active.

Options

- Internal clock can be slowed to 4/5 of normal rate.

Software Power Save Mode 6 (LCD Mode Only)

- Video display accesses to display memory allowed.
- CPU accesses to/from display memory allowed.
- Display memory refresh as in normal active mode.
- I/O read/write of all registers is allowed.
- LCDPWR# signal remains low (i.e. panel power enabled) and video display remains visible on LCD.
- Attribute Controller is disabled (for details see“Windows Power Save Mode” on page 93)
- Power Save Mode 6 is intended for Mode 12h operation under Windows employing a custom Windows driver

Hardware Power Save Mode (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 (28 MHz for LCD, 25 MHz or 28 MHz for CRT), 2) from the PDCLK pin (32 kHz 50% duty cycle, or 64 kHz with short high pulse duration), 3) or from a clock source connected to pin MEMEN.
- Refresh rate generated from CLKI can be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- Refresh rate generated from MEMEN or PDCLK can also be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- No I/O register or memory accesses allowed (including LUT and RAMDAC).
- LCDPWR# and IREFEN# signal forced high.
- LCD and TFT interface output signals tri-stated or forced low, depending on the state of the LCD Signal PS Mode bit in LCD Support Register 1, AUX[02] bit 1.
- If CRT display enabled, CRT and RAMDAC interface signals forced low (except DACRD# and DACWR#)
- All CPU interface input signals except RESET are internally masked off (i.e. ignored). All CPU interface output signals are inactive (except MEMEN).
- Active internal clock oscillator will be automatically disabled unless CLKI is selected as the clock source for display memory refresh generation.

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.

Power Save Mode Function Summary

Table 0-43 LCD Only [a] (no CRT attached) - AUX[0B] bits 1,0 = 01

PSM	Active	SPSM1	SPSM2 s1	SPSM2 s2	SPSM3	SPSM4	SPSM5	Suspend	SPSM6
Function									
LCD Display Active?	On	Off	Off	Off	Off	Off	On	Off	On
CRT Display Active?	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	On
I/O access possible?	Yes	Yes	Yes	Yes	Yes, except LUT, RAMDAC	Yes, except LUT, RAMDAC	Yes	No	Yes
Memory access possible?	Yes	Yes	Yes	No	No	No	Yes	No	Yes
Memory refresh maintained?	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Internal LUT active?	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes
External RAMDAC active?	Sleep *	Sleep *	Sleep *	Sleep *	Sleep *	Sleep *	Sleep *	Powered Down *	Yes
Sequencer running?	Yes	Yes	Yes	No	No	No	Yes	No	Yes
Refresh generated from CLKI (Sequencer running)	Yes	Yes	Yes	n/a	n/a	n/a	Yes	n/a	Yes
Refresh generated from CLKI (Sequencer stopped)	n/a	n/a	n/a	Yes	n/a	option	n/a	option	n/a
Refresh generated from MEMEN	No	No	No	No	n/a	option note 1	No	option note 1	n/a
Refresh generated from PDCLK	No	No	No	No	n/a	option note 2	No	option note 2	n/a
Self-refresh	No	No	No	No	n/a	option note 3	No	option note 3	n/a
256 cycle/4 ms, /32 ms refresh selectable	Yes	Yes	Yes	Yes	n/a	Yes note 1, 2	Yes	Yes note 1, 2	Yes

a. System design should ensure that external RAMDAC is powered off in Suspend Mode; in all other modes, D477 pin should be pulled high and RAMDAC sleep control bit set to placing the RAMDAC into sleep mode

Table 0-44 LCD Only (no CRT attached)

PSM	Active	SPSM1	SPSM2 s1	SPSM2 s2	SPSM3	SPSM4	SPSM5	Suspend	SPSM6
LCD Signals									
UD[3:0], LD[3:0]	Active	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	Active	L/HiZ note 13	Active
YD, LP, XSCL, XSCL2, WF	Active	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	Active	L/HiZ note 13	Active
LCDPWR#	L	H	H	H	H	H	L	H	L
RAMDAC Signals									
IREFEN#	H	H	H	H	H	H	H	H	H
PD[7:0]	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ
PCLK	Active	Active	Active	Active	Halted	Halted	Active	HiZ	Active
DACRD#, DACWR#	Active	Active	Active	Active	Active/H note 21	Active/H note 21	Active	HiZ	Active
RS2, OL0, OL1, OL23	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20
D477	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20	HiZ note 20
BLANK#	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ
CRT Signals									
HSYNC#, VSYNC#	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ	HiZ
MS[2:0]	n/c	n/c	n/c	n/c	n/c	n/c	n/c	n/c	n/c
CPU Signals									
LA[23:17], A[16:0], D[15:0], IOR#, IOW#, IOEN#, BHE#, ALE	Active	Active	Active	Active	Active	Active	Active	masked	Active
MEMR#, MEMW#, MEMEN	Active	Active	Active	Active	masked	masked note 15	Active	masked note 15	Active
Clocks									
25 MHz	Active	Active	Active	Active	can be disabled note 4	can be disabled note 4	Active	can be disabled note 5-8	Active
28 MHz	Active	Active	Active	Active	can be disabled note 4	can be disabled note 4	Active	can be disabled note 5-8	Active

See “Implementation Notes” on page 80 for details.

.

Table 0-45 CRT Only [a] (LCD off) - AUX[0B] bits 1,0 = 10

PSM	Active	SPSM1	SPSM2 s1	SPSM2 s2	SPSM3	SPSM4	SPSM5	Suspend	SPSM6
Function									
LCD Display Active?	Off	Off	Off	Off	Off	Off	n/a	Off	Off
CRT Display Active?	Active	Off	Off	Off	Off	Off	n/a	Off	Active
I/O access possible?	Yes	Yes	Yes	Yes	Yes, except LUT / RAMDAC	Yes, except LUT / RAMDAC	n/a	No	Yes
Memory access possible?	Yes	Yes	Yes	No	No	No	n/a	No	Yes
Memory refresh maintained?	Yes	Yes	Yes	Yes	No	Yes	n/a	Yes	Yes
Internal LUT active?	WR only note 16	WR only note 16	WR only note 16	WR only note 16	Disabled	Disabled	n/a	Disabled	Enabled
External RAMDAC active?	Active	Active or Sleep *	Active or Sleep *	Active or Sleep *	Disabled	Disabled	n/a	Powered Down *	Yes
Sequencer running?	Yes	Yes	Yes	No	No	No	n/a	No	Yes
Refresh generated from CLKI (Sequencer running)	Yes	Yes	Yes	n/a	n/a	n/a	n/a	n/a	n/a
Refresh generated from CLKI (Sequencer stopped)	n/a	n/a	n/a	Yes	n/a	option	n/a	option	n/a
Refresh generated from MEMEN	No	No	No	No	n/a	option note 1	n/a	option note 1	n/a
Refresh generated from PDCLK	No	No	No	No	n/a	option note 2	n/a	option note 2	n/a
Self-refresh	No	No	No	No	n/a	option note 3	n/a	option note 3	n/a
256cycle/4 ms, /32 ms refresh selectable	Yes	Yes	Yes	Yes	n/a	Yes note 1, 2	n/a	Yes note 1, 2	n/a

a. System design should ensure that external RAMDAC is powered off in Suspend Mode.
For Software Power Save Modes 1 and 2, setting the RAMDAC sleep control bit allows the RAMDAC to be placed in sleep mode

Table 0-46 CRT Only (LCD off) - AUX[0B] bits 1,0 = 10

PSM	Active	SPSM1	SPSM2 s1	SPSM2 s2	SPSM3	SPSM4	SPSM5	Suspend	SPSM6
LCD Signals									
UD[3:0], LD[3:0]	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	n/a	L/HiZ note 13	L/HiZ note 13
YD, LP, XSCL, WF	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	L/HiZ note 13	n/a	L/HiZ note 13	L/HiZ note 13
LCDPWR#	H	H	H	H	H	H	n/a	H	H
RAMDAC Signals									
IREFEN#	L	H	H	H	H	H	n/a	H	L
PD[7:0]	Active	Halted	Halted	Halted	Halted	Halted	n/a	Halted	Active
PCLK	Active	Active	Active	Active	Halted	Halted	n/a	HiZ	Active
DACRD#, DACWR#	Active	Active	Active	Active	Active/H note 21	Active/H note 21	n/a	HiZ	Active
RS2, OL0, OL1, OL23	Active	L note 20	L note 20	L note 20	L note 20	L note 20	n/a	L note 20	Active
D477	L	H note 18,20	H note 18,20	H note 18,20	H note 18,20	H note 18,20	n/a	H note 18,20	L
BLANK#	Active	L	L	L	L	L	n/a	L	Active
CRT Signals									
HSYNC#, VSYNC#	Active	L	L	L	L	L	n/a	L	Active
MS[2:0]	connected	connected	connected	connected	connected	connected	n/a	connected	connected
CPU Signals									
LA[23:17], A[16:0], D[15:0], IOR#, IOW#, IOEN#, BHE#, ALE	Active	Active	Active	Active	Active	Active	n/a	masked	Active
MEMR#, MEMW#, MEMEN	Active	Active	Active	Active	masked	masked note 15	n/a	masked note 15	Active
Clocks									
25 MHz	Active	Active	Active	Active	can be disabled note 4	can be disabled note 4	n/a	can be disabled note 5-8	Active
28 MHz	Active	Active	Active	Active	can be disabled note 4	can be disabled note 4	n/a	can be disabled note 5- 8	Active

See “Implementation Notes” on page 80 for details.

Implementation Notes

1. For Software Power Save Mode 4 and the Hardware Power Save Mode (Suspend mode), 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 greater than the minimum DRAM RAS pulse width requirement). The use of a 64 kHz clock source is required for meeting the 256 cycles/4 ms DRAM refresh specification. Optionally, an 8 kHz clock source may be connected to MEMEN for DRAMs supporting 256 cycles/32 ms, or the 64 kHz input can be internally divided down to 8 kHz by setting the 32/4 ms Refresh Select bit (AUX[02] bit 0 = 1).

2. For Software Power Save Mode 4 and the Hardware Power Save Mode (Suspend mode), the clock source connected to the PDCLK input can be either a 32 kHz 50% duty cycle clock, or a 64 kHz clock with duty cycle similar to the requirement for MEMEN described above, except the 64 kHz clock should have a short high pulse instead of low pulse as for MEMEN. The 32 kHz input clock can be accommodated in two ways; an external RC circuit can be utilized and be attached to pins 38, 39 with MD[13] = 1 and MD14 = 1 at RESET in order to create an ~100ns delayed clock. This is used internally to generate a 64 kHz clock from the 32 kHz source with the appropriate duty cycle, as required by the 64 kHz refresh rate for 256 cycles/4 ms DRAM. If MD[14] = 0 at RESET, then the external RC circuit is not necessary and the 32 kHz clock input is doubled internally to generate a 64 kHz clock source which provides the appropriate duty cycle, as required by the 64 kHz refresh rate for 256 cycles/4 ms DRAM. The internal PDCLK is recommended as it leaves pins 38, 39 free to use as Overlay support for the CRT Sprite/HW cursor. For either type of PDCLK source input, the resulting internal 64 kHz refresh rate can be internally divided down to 8 kHz to support 256 cycle/32 ms DRAM by setting the 32/4 ms Refresh Select bit (AUX[02] bit 0 = 1).

3. The self-refresh mode option available in Power Save Mode 4 and the Hardware Power Save Mode (Suspend mode) must only be enabled if the DRAM installed supports self-refresh operation.

4. In Software Power Save Modes 3 and 4, software may set AUX[03] bit 4 to disable the internal clock oscillators. This can be used to further reduce system power consumption.

5. In Hardware Power Save Mode (Suspend mode), the internal clock oscillators are automatically turned off by hardware if the self-refresh option is enabled.

6. In Hardware Power Save Mode (Suspend), if MEMEN is selected as the refresh clock source, then the internal clock oscillators are automatically turned off by hardware.

7. In Hardware Power Save Mode (Suspend), if PDCLK is selected as the refresh clock source, then the internal clock oscillators are automatically turned off by hardware.

8. In Hardware Power Save Mode (Suspend), if the internal active clock is used as the refresh clock source, then the internal clock oscillators cannot be turned 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 a power save mode, LCDPWR# will be driven high (panel power shut off) *before* the interface signals are tri-stated or forced low. Upon exiting a power save 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. Similarly, 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 sig-

nals 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. A control bit in Auxiliary Register 1 allows selecting the power save mode state of the LCD interface signals. In power save modes, the LCD interface signals can all be driven low, or can be put into a high-impedance state, as selected by this option.
14. The output pin IREFEN# should be used to control the current reference source for the external RAMDAC. When IREFEN# is high, the current reference should be shut off - this will ensure that the DAC analog circuitry is not active. When IREFEN# is low, the current reference should be enabled. If a voltage reference is used for the RAMDAC, then IREFEN# is not required and may be left unconnected.
15. When the MEMEN pin is selected as the refresh clock source, this input will not be masked during Power Save Mode 4 or Suspend mode.
16. If AUX[0B] bit 2 = 1, then reads to the I/O addresses 3C6h, 3C8h, and 3C9h will be decoded as external RAMDAC reads. If AUX[0B] bit 2 = 0, then reads to these I/O addresses will access the internal LUT registers. For CRT modes, this bit should be set to 1. If CRT mode is enabled, and the chip is in active mode, Power Save Mode 1 or Power Save Mode 2, then writes to these I/O addresses and 3C7h will result in data being written to both the internal LUT registers and the external RAMDAC registers. Reads from 3C7h will always return the internal RAMDAC/Lookup Table Status Register.
17. In active mode if the CRT is enabled, the logic value on the D477 pin is determined by AUX[0B] bit 4.
18. In order to properly make use of SLEEP mode of the RAMDAC, software is required to program the sleep bit in the external RAMDAC control register on system initialization. When the SPC8106 forces the D477 pin high in power save modes, the RAMDAC will enter sleep mode if this bit has been programmed correctly.
19. In Power Save Modes 3 and 4 you cannot access the LUT/RAMDAC. However, if the LUT/RAMDAC is written to in this mode, the last I/O write will actually get written.
20. If AUX[06] bit 4=1, RS2 remains active and D447 remains L in LCD only mode or in any Power Save modes.
21. In Software Power Save Modes 3 and 4, software may set AUX[03] bit 3 to force DACRD# and DACWR# high. This can be used to further reduce system power consumption.

10.0 SOFTWARE CONSIDERATIONS

10.1 Display Modes Supported

Table 0-47 LCD 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
13	Graphics	N/A	N/A	320 x 200	640 x 400	64	256	A000
100	Graphics	N/A	N/A	640 x 400	640 x 400	64	256	A000
101	Graphics	N/A	N/A	640 x 480	640 x 480	64	256	A000
108	Text	8 x 8	80 x 60	640 x 480	640 x 480	16	16	B800

Table 0-48 CRT Display Modes

Mode No.	Mode Type	Font	Characters	Resolution	Displayed Pixels	Colors	Memory Segment
0	Text	8 x 8	40 x 25	320 x 200	640x400	16	B800
0+	Text	8 x 14	40 x 25	320 x 350	640x350	16	B800
0++	Text	9 x 16	40 x 25	360 x 400	720x400	16	B800
1	Text	8 x 8	40 x 25	320 x 200	640x400	16	B800
1+	Text	8 x 14	40 x 25	320 x 350	640x350	16	B800
1++	Text	9 x 16	40 x 25	360 x 400	720x400	16	B800
2	Text	8 x 8	80 x 25	640 x 200	640x400	16	B800
2+	Text	8 x 14	80 x 25	640 x 350	640x350	16	B800
2++	Text	9 x 16	80 x 25	720 x 400	640x400	16	B800
3	Text	8 x 8	80 x 25	640 x 200	640x400	16	B800
3+	Text	8 x 14	80 x 25	640 x 350	640x350	16	B800
3++	Text	9 x 16	80 x 25	720 x 400	640x400	16	B800
4	Graphics	N/A	N/A	320 x 200	640x400	4	B800
5	Graphics	N/A	N/A	320 x 200	640x400	4	B800
6	Graphics	N/A	N/A	640 x 200	640x400	2	B800
7	Text	8 x 14	80 x 25	640 x 350	640x350	2	B000
7+	Text	9 x 16	80 x 25	720 x 400	720x400	2	B000
0D	Graphics	N/A	N/A	320 x 200	640x400	16	A000
0E	Graphics	N/A	N/A	640 x 200	640x400	16	A000
0F	Graphics	N/A	N/A	640 x 350	640x350	2	A000
10	Graphics	N/A	N/A	640 x 350	640x350	16	A000
11	Graphics	N/A	N/A	640 x 480	640x480	2	A000
12	Graphics	N/A	N/A	640 x 480	640x480	16	A000
13	Graphics	N/A	N/A	320 x 200	640x400	256	A000
100	Graphics	N/A	N/A	640 x 400	640x400	256	A000
101	Graphics	N/A	N/A	640 x 480	640 x 480	256	A000
108	Text	8 x 8	80 x 60	640 x 480	640 x 480	16	B800

10.2 DoubleScan Support

DoubleScan is a feature in the SPC8106 that allows simultaneous display of both the CRT and LCD panel. Refer to the following table for supported video modes and limitations.

DoubleScan is not supported when using a Dual panel / Dual drive LCD.

All non-480 line modes (when using a 480 line panel) will show a wrap-around effect if DoubleScan is enabled. For example: display a 400 line mode on a 480 line panel and enable DoubleScan. You would see the top 78 lines duplicated on the bottom 78 lines (wrap-around affect).

If using a panel with less than 480 vertical lines, DoubleScan will be supported, however, the maximum LCD frame-rate may be violated. Therefore, the specific panel should be referenced.

If supporting a TFT panel requiring CRT-like timing (AUX[00] bit 5=1 and AUX[0B] bit 1=1) the panel handles the 350, 400 and 480 line modes, providing screen positioning internally. As the result of this direct support, DoubleScan is supported for all standard VGA and some extended modes.

If supporting a TFT panel requiring LCD-like timing (AUX[00] bit 5=1 and AUX[0B] bit 1=0) the DoubleScan mode is not supported.

Table 0-49 DoubleScan Supported Video Modes

Mode No.	Single Panel (640x480)	TFT - Color 9/12-bit (640x480) AUX[00] bit 5=1 AUX[0B] bit 1=1
0	No	Yes
0+	No	Yes
0++	No	Yes
1	No	Yes
1+	No	Yes
1++	No	Yes
2	No	Yes
2+	No	Yes
2++	No	Yes
3	No	Yes
3+	No	Yes
3++	No	Yes
4	No	Yes
5	No	Yes
6	No	Yes
7	No	Yes
7+	No	Yes
0D	No	Yes
0E	No	Yes
0F	No	Yes
10	No	Yes
11	Yes	Yes
12	Yes	Yes
13	No	Yes
100	No	Yes
101	Yes	Yes
108	Yes	Yes

10.3 Standard VGA Register Considerations

This section describes deviations of the SPC8106 from the VGA standard. Section “Background” on page 85 gives a general description of the ways in which SPC8106 differs from the VGA standard. Section “General Discussion” on page 85 describes each difference in greater detail. Section “Details” on page 86 gives a detailed list of the changes to the VGA register set. For a list of all registers actually supported by SPC8106, see “I/O Register Summary” on page 89.

10.4 Background

There are two ways in which the SPC8106 differs from the VGA standard.

Firstly, the CRTC timing registers differ since the SPC8106 is optimized to support 640x480 single/dual panel LCD's, with some flexibility to support other size panels. In LCD mode, the panel timing is determined by two special registers (horizontal and vertical panel size), so the CRTC registers of the VGA standard which would normally be used to vary the display monitor timings are ignored in LCD mode.

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 SPC8106 in order to optimize the design for VGA operation on a 16 gray-scale 640x480 dot LCD panel. In their place are read/writable registers bits that have no effect.

General Discussion

CRTC Timing Registers

In the SPC8106, the non-timing related CRTC registers are the same as in the VGA standard. However, in LCD modes the SPC8106 does not use any of the timing registers that provide full monitor timing programmability since much of the timing circuitry has been optimized in hardware to support 640x480 panels.

For LCD mode, the SPC8106 timing circuitry is programmed by a Horizontal Panel Size register and a Vertical Panel Size register - all other LCD timing parameters are fixed in hardware, so the functions of all the regular timing registers is ignored in LCD mode. The functions of registers 00h, 02h to 06h,10h, 15h, 16h, and certain bits in registers 07h, 09h and 11h are ignored in LCD mode - in their place dummy read/write registers are provided. Note that CRTC registers 01h and 12h (Horizontal/Vertical Display Enable End, respectively) have full functionality in both LCD and CRT modes.

For CRT mode, full CRTC register support is provided, with some minor exceptions (see next item).

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 SPC8106 is exclusively a 256kbyte 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 have been 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 size. To simplify the design, the logic to select a 9-dot character clock has been removed for LCD display modes. The associated register bit in the Sequencer functions as normal in CRT modes, but for LCD modes this bit has no effect and only 8 dot modes are displayed on the LCD.

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

Lookup Table for LCD Display

For monochrome LCD modes, the VGA palette has been replaced by an internal 256x6 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.
For mode 256 color modes, the LCD controller will display a maximum of 64 shades of grey on a monochrome LCD rather than the 256 of 256k colors on a CRT monitor. These 64 shades of gray are achieved by using 16 gray-level frame-rate modulation plus dithering. For modes other than 256 color modes, the controller will display up to 16 shades of gray using frame-rate modulation. Because of the smaller number of grey shades displayable on the LCD, a method of mapping colors originally intended for display on a 16/256 color CRT monitor onto a 16/64 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 6 bit gray scale values that are stored in the lookup table.
For color LCD modes, the VGA palette is replaced by an internal 256 x 12 bit color lookup table. This lookup table operates identically to the VGA palette it replaces except only the most significant 4 bits of each of the 6 bit VGA R, G, B color values are stored. This allows a maximum of 256 colors displayed (in 256 color modes) on a color LCD out of a palette of 4096 possible colors.
For CRT display, the external RAMDAC supports the full 256 of 256k colors. If CRT mode is enabled and AUX[0B] bit 6 = 0, writes to the RAMDAC registers will also go to the internal LUT registers. An Auxiliary Register bit allows reading from either the external RAMDAC or internal LUT registers in this case.

Details

This section gives a list of register bits in the SPC8106 that are different to the VGA standard. Only differences are listed here. Any bits not mentioned are the same as the IBM VGA. For a complete list of the actual SPC8106 registers see “I/O Register Summary” on page 89.

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

bit 3	Clock Select bit 1 Read/write only. No other effect in hardware.
bit 1	Enable DRAM 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 5-4	Diagnostic Deleted.
bits 2-1	Light Pen Read Only as 0,1.

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

Register not supported.

Sequencer Registers (3C4h, 3C5h)

Clocking Mode Reg. (Index 01h)

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

bit 2 Shift-Load
Read/write only. No other effect in hardware.

bit 1 CPU Bandwidth
Read/write only. No other effect in hardware.

bit 0 8/9 Dot Select
For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT modes, this bit selects 8 or 9 dot mode.
For dual display mode the 9th bit is stripped by hardware.

Memory Mode Reg. (Index 04h)

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

CRTC Registers (3D4h, 3D5h)

Horizontal Total (Index 00h)

For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

Horizontal Display Enable (Index 01h)

For LCD/CRT or Doublescan modes, if AUX[00] bit 0 = 1, this register accesses the LCD Horizontal Panel Size register.
For LCD-only mode (not Doublescan) and AUX[00] bit 0 = 0, this register is read/write only. No other effect in hardware.
For CRT or Doublescan modes, AUX[00] bit 0 = 0, normal VGA function.

Horizontal Blanking Start (Index 02h)

For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

Horizontal Blank End (Index 03h)

bits7-5 Read/write only. No other effect in hardware.

bits 4-0 For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

Horizontal Retrace Start (Index 04h)

For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

Horizontal Retrace End (Index 05h)

For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

Vertical Total End (Index 06h)

For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

CRTC Overflow (Index 07h)

bits 7-5, 3, 2, 0 For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

Maximum Scan Line (Index 09h)

bits 7-6 For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

Cursor Start (Index 0Ah)

bit 7 IBM Test Bit
Deleted.

Cursor End (Index 0Bh)

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

Vertical Retrace Start (Index 10h)

For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

VRTC End (Index 11h)

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

Vertical Display Enable End (Index 12h)

For LCD/CRT or Doublescan modes, if AUX[00] bit 0 = 1 then this register accesses the LCD Vertical Panel Size Register.
For LCD only modes (not Doublescan) and AUX[00] bit 0 = 0, this register is read/write only. No other effects in hardware.
For CRT or Doublescan modes and AUX[00] bit 0 = 0, normal VGA function.

Underline Location (Index 14h)

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

Vertical Blanking Start (Index 15h)

For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

Vertical Blanking End (Index 16h)

For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT or Doublescan modes, normal VGA functionality.

Mode Control (Index 17h)

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

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

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

Attribute Controller Registers (3C0h, 3C1h)

Mode Control Register (Index 10h)

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

bit 2 Line Graphics 8>9 Dot
For LCD-only modes (AUX[0B] bits 1,0 = 01) - Read/write only. No other effect in hardware.
For CRT modes, this bit enables 8-9 dot copy for line graphics characters in text mode.

Color Plane Enable (Index 12h)

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

Horizontal Panning Register (Index 13h)

bits 3-0 Horizontal Panning Bit 3-0
For non-CRT modes (AUX[0B] bit 1 = 0) or if forced 8-dot mode (AUX[01] bit 7 = 1), a panning value of greater than 7 will have the same effect as a panning value of 0.

10.5 I/O Register Summary

This section summarizes the I/O registers of SPC8106 - 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 SPC8106 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	Auxiliary Index Bit 4	Auxiliary Index Bit 3	Auxiliary Index Bit 2	Auxiliary Index Bit 1	Auxiliary Index Bit 0

00 Extended Function Register 0 RW							
RGBI Mode Enable	12 Bit RGB Mode Enable	TFT Enable	Test Mode Enable (must = 0)	n/a	IRQ Output Enable	Multi-Font Enable	LCD B Reg Program Enable

bit 7 RGBI Mode Enable
This bit enables an RGBI, 4-bit value to be directly driven out on the LD[3:0] pins at the pixel clock rate. The RGBI value represents the EGA palette index. This bit, when asserted, alters the function of the LCD interface. See “RGBI Mode Dual LCD Panel Timing” on page 70.

bit 6 12-bit RGB Mode Enable
This bit enables a 12 bits per pixel, 4 bits per primary color to be directly driven on the LCD and CRT overlay output pins at the pixel clock rate. The 12-bit value represents the output from the internal look up table. See “12-bit RGB Mode LCD Panel Timing” on page 70. This bit must be set to 1 for 12-bit TFT panel operation.

bit 5 TFT Enable
This bit enables the TFT mode. When this bit = 1, it forces the LCD enable bit, AUX[0B] bit 0 to 1 enabling the LCD interface. This bit works in conjunction with the CRT enable bit as follows:

Table 0-50 Display Enable Bits

12-Bit RGB Mode Enable AUX[00] bit 6	TFT Enable Bit AUX[00] bit 5	CRT Enable Bit AUX[0B] bit 1	LCD Enable Bit AUX[0B] bit 0	Display Mode
0	0	0	0	no display
			1	Standard LCD mode
		1	0	CRT mode
			1	LCD DoubleScan mode
	1	0	X	9-bit TFT LCD mode
		1	X	9-bit TFT CRT mode / DoubleScan
1	1	0	X	12-bit TFT LCD mode
		1	X	12-bit TFT CRT mode / DoubleScan

bit 4 Test Mode Enable
The Test Mode Enable bit must be set to 0 for normal operation. When set to 1, the chip is placed into device test mode and the test input and test output selector bits in Auxiliary Register 04h are enabled.

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

bit 1 Multi-Font Enable
When the Multi-Font Enable bit = 0, normal text mode font selection is enabled. When this bit = 1, it allows bits 0 through 2 of the attribute byte (foreground color) to select one of eight simultaneously displayable fonts. In this case the attribute byte foreground color bits (normally bits 0 to 2) are forced to 1, the font selection bit (bit 3) is not used, and the blink/intensity bit (bit 7) functions normally.

bit 0 LCD B Registers Program Enable
This bit is used to access the hidden Horizontal Panel Size and Vertical Panel Size registers (LCD B Registers), which reside in the same address space as their CRT mode counterparts. These hidden timing registers only have an effect when the LCD is the active display. When this bit = 0, accesses to CRTC Register [01] affect the normal Horizontal Total Register, and accesses to CRTC Register [12] affect the normal Vertical Display Enable End Register. When this bit = 1, then the “B-set” registers are accessible, and accesses to CRTC Register [01] affect the Horizontal Panel Size Register, and accesses to CRTC Register [12] affect the Vertical Panel Size Register. These “B-set” registers are used for LCD only mode (AUX[0B] bits 1,0 = 01). The Protect Registers bit (CRTC[11] bit 7) has no effect when programming the CRTC “B-set” registers.

01 LCD Support Register 0 RW							
Force 8-dot Mode	Slow Blink Select	Force Overscan Low	Grn-only/ NTSC GS Weighting	Reverse/ Normal Graphics	Reverse/ Normal Text	Auto-Centering Enable	Panel Config Bit 0

bit 7 Force 8-dot Mode
This bit is intended for use in TFT mode. When this bit = 1, 8-dot mode timing is forced and Horizontal Panning bit 3 (ATC[13] bit 3) is forced to 0. Additionally in CRT mode (AUX[0B] bit 1 = 1) setting this bit to 1 also forces the 25 MHz clock to be selected.

bit 6 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 = 0, the cursor, text or graphics pixels blink at their normal rates. If this bit = 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.

bit 5 Force Overscan Low
This bit is intended for use in TFT mode. When this bit = 1, pixel data output during overscan period is forced to 0.

bit 4 Green-only/NTSC GS Weighting Select
This 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 256x6 LCD gray scale lookup table or the internal sprite palette registers. When this bit = 0, RGB data values are mapped to gray values using NTSC weighting. When this bit = 1, the green component of the RGB data is used as the gray value. See “LCD Gray Scale/Color Lookup Table Architecture” on page 122 for details. Note that this bit only has an effect when monochrome LCD mode is selected (AUX[02] bit 6 = 0). If color LCD mode is selected (AUX[02] bit 6 = 1), the gray scale weighting logic is bypassed.

bit 3 Reverse/Normal Graphics
When the Reverse/Normal Graphics bit = 0, then normal display attributes are enabled. When this bit = 1, then inverse video is displayed on the LCD display when in graphics modes (GRC[06] bit 0 = 1). This bit also affects the overscan color as set in bit 5 of this register. This bit has no effect on the image displayed on the CRT display.

bit 2 Reverse/Normal Text
When the Reverse/Normal Text bit = 0, then normal display attributes are enabled. When this bit = 1, then inverse video is displayed on the LCD display when in text modes (GRC[06] bit 0 = 0). This bit also affects the overscan color as set in bit 5 of this register. This bit has no effect on the image displayed on the CRT display.

bit 1 Auto-Centering 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 LCD panel. If the Auto-Centering Enable bit = 0, then all modes will be displayed with the first line at the top of the panel. If this bit = 1, 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.

If vertical expansion is active (i.e. all conditions for a vertical expansion mode are met; see appropriate bit description in AUX[7]), then the setting of this bit is ignored and auto-centering is disabled. If CRT mode is enabled (AUX[0B] bit 1 = 1) then auto-centering is disabled and this bit has no effect.

bit 0 Panel Configuration Bit 0
See Panel Configuration Bits 3:1, AUX[02] bits 6:4.

02 LCD Support Register 1 RW							
CRT Sprite Enable	Panel Config Bit 3	Panel Config Bit 2	Panel Config Bit 1	PSM4/S Refresh Clk Select Bit 1	PSM4/S Refresh Clk Select Bit 0	LCD Signal PS Mode State	32/4 ms Refresh Select

bit 7 CRT Sprite Enable

The CRT Sprite Enable bit is used to enable or disable the sprite display on the CRT monitor. When this bit = 0, the sprite will not be displayed on the CRT monitor. When this bit = 1 and AUX[05] bit 5 = 1, the sprite circuitry for the CRT display is enabled and it will be possible to display the sprite on the CRT monitor, provided that the value on MD[13] = 0 at RESET to allow use of the OL1 and OL0 output pins.

bits 6-4 Panel Configuration Bits [3:1]

The following table describes the function of the Panel Configuration Bits [3:0]. Bit 0 is located in AUX[01] bit 0 (see above).

Table 0-51 Panel Configuration Bit Table

Panel Description	**Panel Config Bit 3 AUX [02] b6**	**Panel Config Bit 2 AUX [02] b5**	**Panel Config Bit 1 AUX [02] b4**	**Panel Config Bit 0 AUX [01] b0**	**AUX[00] Bits**		
					b7	**b6**	**b5**
8 bit Dual Monochrome	0	0	0	0	0	0	0
8 bit Single Monochrome	0	0	0	1	0	0	0
4 bit Single Monochrome	0	0	1	1	0	0	0
8 bit Single Color	1	0	0	1	0	0	0
8/16[a] bit Dual Color/4 bit Single Color	1	1	X	0	0	0	0
16 bit Single Color	1	1	X	1	0	0	0
9 bit color TFT	1	X	X	1	0	0	1
12 bit color TFT	1	X	X	1	0	1	1
12 bit RGB	1	X	X	0	0	1	0
RGBI	1	X	X	0	1	0	0

a. 16-bit panels require external circuitry for DoubleScan compatibility.
X bits are DON'T CARE

bits 3-2 PSM4/S Refresh Clock Select Bits [1:0]

These bits are used to select the refresh clock source during Power Save Mode 4 or Suspend mode, according to the following table:

Table 0-52 PSM4/S Refresh Clock Select

PSM4/S Refresh Clock Select 1	**PSM4/S Refresh Clock Select 0**	**Refresh Clock Source in Power Save Mode 4 and Suspend**
0	0	CLKI1, CLKI2
0	1	MEMEN
1	0	Self Refresh
1	1	PDCLK

CLKI1, CLKI2

When this option is selected, then the active pixel input clock (CLKI1 or CLKI2) is used to generate all Power Save mode refresh timing. The active pixel clock is determined by the Clock Select bits in Misc Output Register (3C2), and by the LCD Enable and CRT Enable bits in AUX[0B].

MEMEN
When this option is selected, then the MEMEN input pin is used as the clock source in Power Save Mode 4 and Suspend.

Self Refresh
This option may only be used when the DRAMs installed are capable of self-refresh. When this option is selected, during Power Save Mode 4 and 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 Power Save Mode 4 or Suspend mode. Note that regardless of the setting of these bits, CAS-before-RAS refresh cycles are used during active mode and Power Save Modes 1, 2 and 5.

PDCLK
When this option is selected, the PDCLK input pin is used as the clock source in Power Save Mode 4 and Suspend. For normal refresh rate DRAM (256 cycle/4 ms), this input should be a 64 kHz clock source. If a 64 kHz clock source is attached to this input, for lowest possible DRAM power consumption this input clock should have as short as possible high duration (but > min RAS pulse width). It is possible to use a 32 kHz 50% duty cycle clock for PDCLK - see “Power Save Mode Control Pins” on page 21 for details.

bit 1 LCD Signal PS Mode State
The LCD Signal PS Mode State bit controls the states of the LCD interface signals (UD[3:0], LD[3:0], XSCL, XSCL2, LP, YD, WF) when the chip goes into a power save mode. When this bit = 0, the LCD signals are put into a high-impedance state when a power save mode is entered. When this bit = 1, then when the chip is in a power save mode, the LCD interface signals will be forced low. On RESET, this bit is set to 1.

bit 0 32/4 ms Refresh Select
The 32/4 ms Refresh Select bit is used to select 256 cycle/4 ms or 256 cycle/32 ms DRAM refresh timing in all modes of operation. When this bit is 0, then 4 ms refresh timing is generated. When this bit is 1, then 32 ms refresh timing is generated. In active mode and Power Save Modes 1, 2 and 5, this 4 or 32 ms refresh timing is generated from the selected CLKI source (28 MHz for LCD modes, 25 MHz or 28 MHz for CRT modes as selected by Clock Select bits in Misc Output Register 3C2h). For Power Save Mode 4 and Suspend, this 4 or 32 ms refresh timing is generated from the active CLKI, from MEMEN input, or the PDCLK input, as selected by AUX[02] bits 3,2.

03 Power Save Register RW							
Windows Power Save Mode	n/a	Clock Slow Down	Oscillator Disable	Aux Reg Only Decode	Power Save Mode Select Bit 2	Power Save Mode Select Bit 1	Power Save Mode Select Bit 0

bit 7 Windows Power Save Mode
Setting the Windows Power Save Mode bit to 1 bypasses the Graphics Blink and Pel Panning logic to reduce power consumption. This bit should be set for mode 12h operation from within a Windows driver only.

bit 5 Clock Slow Down
The Clock Slow Down bit is used to provide additional power savings in some LCD modes. When this bit is set to 1, then the active internal clock rate is reduced by 20%. When this bit is set to 0, then the internal clock rate equals the input clock rate. This bit is intended for use in Power Save Mode 5. If CRT mode is enabled (AUX[0B] bit 1 = 1), then this bit is ignored and has no effect.

bit 4 Oscillator Disable
The Oscillator Disable bit is used to control the operation of the internal clock oscillators connected to the external 2-terminal crystals. When this bit = 0 the oscillators are enabled. When this bit 1 the oscillators are disabled and the corresponding CLKI inputs are masked off.

bit 3 Aux Reg Only Decode
The Aux Reg Only Decode bit is intended for use by power save mode software. In power save modes, this bit may be set to 1 to disable all I/O access decoding except to the Auxiliary Registers. Note that setting this bit to 1 would normally be useful only when in Power Save Modes 3 or 4. At all other times this bit should be set to 0 to enable all I/O address decoding.

bits 2-0 Power Save Mode Select Bits [2:0]
These bits are used to select 1 of 5 software power save modes. When these bits are set to 000, 110, or 111 then the chip operates in normal active mode. Binary values of 001, 010, 011, 100, or 101 written to these bits cause the chip to enter power save modes 1, 2, 3, 4 or 5 respectively. If the SUSPEND# input pin is low, then the power save mode setting in this register is ignored. If CRT mode is enabled, programming power save mode 5 has no effect.

04 General Storage and Test Register 0 RW							
Test Input Select Bit 3	Test Input Select Bit 2	Test Input Select Bit 1	Test Input Select Bit 0	Test Output Select Bit 3	Test Output Select Bit 2	Test Output Select Bit 1	Test Output Select Bit 0

bits 7-0 General Storage and Test Bits
For normal operation, AUX[00] bit 4 (Test Mode Enable) is set to 0, and then the General Storage and Test Register can be used to provide 8 bits of read/write temporary storage. In normal operation mode these bits have no effect on hardware.

Caution

When the Test Mode Enable bit is set to 1 the chip is placed in a special test mode and this register is used to select various internal test functions.

05 Extended Function Register 1 RW							
Packed 4-bit/ pixel Mode Enable	Hardware Zoom Enable	Sprite/HW Cursor Enable	n/a (reserved)	Start Address Bit SA16	Extended Display Page Enable	Hi-Res 256 Color Mode Enable	CPU Upper 256K Access Enable

bit 7 Packed 4-bit per Pixel Mode Enable
When this bit = 1 the 4-bit per pixel packed pixel format video mode is enabled.

bit 6 Hardware Zoom Enable
When this bit = 1 every second display pixel and line are removed to that the image is shrunk in half. This applies to all video modes. This bit is intended to be used with sub-VGA single drive single panels, thus allowing a 640 x 480 image to be displayed on a 320 x 240 panel.

bit 5 Sprite/HW Cursor Enable This bit is used to enable the sprite and hardware cursor functions. When this bit = 0, the sprite/hardware cursor function is disabled. When this bit = 1, the sprite/hardware cursor is enabled. See “Sprite/HW Cursor Registers” on page 113 for more information.

bit 3 Start Address Bit SA16
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 (see below). Along with the lower 16 bits of start address in CRTC registers 0C and 0D, this bit allows setting the start address of the image displayed to be anywhere in the 512K address space. For this bit to have an effect, the Extended Display Page Enable bit must be set to 1.

bit 2 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.

bit 1 Hi-Res 256-color Mode Enable
When this bit = 1 the 640 x 480 256-color mode for single panel LCD displays and CRT monitors is enabled.

bit 0 CPU Upper 256K Access Enable
This bit is used to enable CPU accesses into the upper 256K bytes of display memory. When this bit = 1, the CPU can access the upper half of display memory through memory addresses B0000 - BFFFF. The lower 256K of video memory can still be accessed at addresses A0000 - AFFFF. When this bit = 0, only the lower 256K bytes of display memory can be accessed by the CPU. Note that the Upper Page Swap Enable bit can be used to swap the upper/lower page addresses. This function will only be useful in planar graphics modes.

06 Extended Function Register 1B RW							
n/a	n/a	Panel Data Setup/Hold Select	RS2/D477 Control	Page Select Register Port Enable	Horizontal Non-display Period Width	Sequencer CPU Cycles Disable	LCD Power Disable

bit 5 Panel Data Setup/Hold Select
When this bit = 1, 16-bit color panel data setup/hold timings are set to 1Ts/1.5Ts minimum.
When this bit = 0, 16-bit color panel data setup/hold timings are set to 1.5Ts/1Ts minimum.

bit 4 RS2/D477 Control
When this bit = 1, RS2 and D477 remain active all the time.
When this bit = 0, RS2 and D477 are tri-stated in non-CRT mode and forced low and high respectively in power save modes.

bit 3 Page Select Register 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 2 Horizontal Non-display Period Width
Setting this bit to 1 forces the Horizontal Non-display Period to 20 fast/10 slow character clocks. Setting this bit to 0 forces the Horizontal Non-display Period to 14 fast/7 slow character clocks. This bit has an effect only when AUX[0B] bit 1 (CRT Enable) is set to 0.

Table 0-53 Horizontal Non-Display Period Width

Dual/Single (1/0)	Slow/Fast (1/0)	One Character Clock
0	0	8 Ts
0	1	16Ts
1	0	16Ts
1	1	32Ts

bit 1 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 32-bit display latch. A memory write with this bit set is an invalid cycle and will have no effect.

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

07 Extended Function Register 2 RW							
Mode 13h 32/64 Gray Select	Dithering Control	Auto Disable Line/Pixel Doubling	Reserved Test Bit	Reserved Test Bit	Reserved Test Bit	Graphics VExpand Enable	Text VExpand Enable

bit 7 Mode 13h 32/64 Gray Select
This bit allows the selection of 32 or 64 gray shades on a monochrome LCD panel when in mode 13h (256 Color Mode, Graphics Controller Mode Register index [05], bit 6 Mode13h Select = 1). When this bit = 0, 64 gray shades are displayed in mode 13h (16 gray levels by FRM + dithering). When this bit = 1, then 32 gray shades are displayed in mode 13h (16 gray levels by FRM + dithering). This bit has no effect on an image displayed on a CRT display.

bit 6 Dithering Control
This bit is used to control the dithering logic for the monochrome LCD display. When this bit = 0, dithering is enabled for mode 13h (Graphics Controller register [05] bit 6 = 1) and disabled for other modes - this is the normal setting to provide 64 gray shades in mode 13h. When this bit = 1, then dithering is disabled for mode 13h. This bit has no effect on an image displayed on a CRT display. Dithering is also disabled in Power Save Mode 5.

bit 5 Automatic Line and Pixel Doubling Disable
When this bit = 1 the following occurs:

graphics modes: If pixels are doubled from 320 to 640 (slow dot clock) the pixels are "undoubled". If lines are doubled from 200 to 400 (CRTC[09]b7=1) the lines are "undoubled". This also applies for mode 13h, where line doubling is achieved by setting the font height to 1 in CRTC[09].

a. *text modes 0 and 1 (slow dot clock)*: If pixels are doubled from 320 to 640 the pixels are "undoubled". In 200 line modes the display stays at 200 lines. In 350 and 400 line modes only even lines are displayed. To avoid any loss of information in these modes a special double line font should be used.

Setting this bit affects both CRT and LCD operations. Line "undoubling" applies to both CRT and LCD operations. Pixel "undoubling" applies to LCD operations only.

bits 4-2 Reserved Test Bits
These test bits must always be programmed to 000b.

bit 1 Graphics VExpand Enable
This bit is used to enable vertical expansion of 400 line graphics modes on the LCD display, using selective line duplication. If this bit is 0, then vertical expansion of graphics modes is disabled. If this bit is set to 1, then 400 line graphics modes are expanded vertically to fill the screen. All the following conditions must be true for graphics vertical expansion to occur:

Graphics VExpand Enable = 1

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

c. graphics mode set, i.e. Graphics Controller register 06h bit 0 = 1

If a 400 line graphics mode is not set, then graphics expansion will not occur and this bit will have no effect. Note that if the first three conditions for graphics vertical expansion are met, then the Auto Centering Enable bit (Auxiliary Register 01 bit 1) is ignored and autocentering is disabled.

If CRT mode is enabled (AUX[0B] bit 1 = 1) then graphics vertical expansion is disabled and this bit has no effect.

bit 0 Text VExpand Enable
This bit is used to enable vertical expansion of 400 line text modes on the LCD display, using selective line duplication. If this bit is 0, then vertical expansion of text modes is disabled. All the following conditions must be true for text vertical expansion to occur:

Text VExpand Enable = 1

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

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

f. text mode is set, i.e. Graphics Controller register 06h bit 0 = 0.

If these conditions are not all met, then text expansion will not occur and this bit will have no effect. If the conditions are met, then the Auto Centering Enable bit (Auxiliary Register 01 bit 1) is ignored and autocentering is disabled.

If CRT mode is enabled (AUX[0B] bit 1 = 1) then text vertical expansion is disabled and this bit has no effect.

08 Primary Revision Code Register RO							
Primary Revision Code Bit 2	Primary Revision Code Bit 1	Primary Revision Code Bit 0	n/a	n/a	Monitor ID Bit 2	Monitor ID Bit 1	Monitor ID 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.

bits 2-0 Monitor ID Bits [2:0]
The Monitor ID Bits allow software to read the status on the monitor sense input pins MS[2:0]. These inputs are not latched and are expected to have external pullup resistors attached, so that if nothing is connected to them these register bits will read 111.

09 Sprite Write Select Register RW							
n/a	n/a	n/a	n/a	Sprite Page Select	Sprite Logical Plane Sel	Upper Page Swap Enable	Sprite Write Mode Enable

bit 3 Sprite Page Select
This bit is used to select the current sprite page (page 0 or 1) accessed at the A000h address segment when the sprite write mode is enabled. The mapping of the sprite pages to the Sprite Page Select bit and the Upper Page Swap Enable bit are given in the following table:

Table 0-54 Sprite Page Mapping

Upper Page Swap Enable	**Sprite Page Select**	**Addressable Sprites**
0	0	0 to 127
0	1	128 to 255
1	0	256 to 383
1	1	384 to 511

Sprites 0 to 255 reside in standard VGA display memory space and should be used with caution.

bit 2 Sprite Logical Plane Select
This bit is used in sprite write mode to select which sprite bit plane is currently being written. When this bit is set to 1, logical bit plane 1 is selected, and when this bit is set to 0, logical bit plane 0 is selected.

bit 1 Upper Page Swap Enable
The Upper Page Swap Enable bit is used to swap access addresses for the upper and lower 256 sprites. When this bit is set to 1, the upper 256 sprites residing in the B000h address segment can be addressed at A000h. When this bit is set to 0, the upper 256 sprites are addressed at B000h. The recommended setting for this bit is 1 when writing sprite data.

bit 0 Sprite Write Mode Enable
This bit is used to enable sprite write mode. When this bit is set to 1, the Graphics Controller write mode logic is disabled and CPU data can be written directly to the selected sprite plane. When this bit is set to 0, the Graphics Controller write mode logic functions normally on all CPU write data.

0A General Storage Register 1 RW							
General Storage Bit 7	General Storage Bit 6	General Storage Bit 5	General Storage Bit 4	General Storage Bit 3	General Storage Bit 2	General Storage Bit 1	General Storage Bit 0

bits 7-0 General Storage Bits
The General Storage Register 1 can be used to provide 8 bits of read/write temporary storage. These bits have no effect on hardware.

0B Extended Function Register 3 RW							
Ext Palette Write Disable	Int. LUT Write Disable	Anti-sparkle Enable	Ext. DAC Ctl Reg Write Enable/ D477	Ext. DAC RS2 Bit	Ext. DAC Read Select	CRT Enable	LCD Enable

bit 7 External Palette Write Disable
When this bit is set to 0, I/O writes to the External RAMDAC are allowed. When this bit is set to 1, I/O writes to the External RAMDAC are masked off. This masking includes writes to the External Write Address Register (3C8h), the Palette Data Register (3C9h), the LCD Lookup Table Pixel Mask Register (3C6h), and the External Read Address Register (3C7h). The resultant effect is that reads from the External RAMDAC are valid only for the last valid Read Address, and new writes are masked off, however, the writes will affect External DAC Status Register (3C7h).

bit 6 Internal LUT Write Disable
When this bit is set to 0, I/O writes to the internal LUT registers are allowed. When this bit is set to 1, I/O writes to the internal LUT are masked off (3C7h, 3C9h). Even if writes are disabled by this bit, attempted I/O writes will still affect the LUT/RAMDAC status register.

bit 5 Anti-sparkle Enable
Setting this bit to 1 enables the Anti-sparkle circuitry which copies the previous display pixel during Look Up Table (LUT) accesses. Setting this bit to 0 disables this circuitry. This bit is intended to be used with TFT displays.

bit 4 External DAC Control Register Write Enable
This bit is used to control the logic level on output signal D477. If this bit is set to 0, then the D477 output is driven low. If this bit = 1, then the D477 output is driven high. Note that the logic level on this pin is also controlled by power save logic and CRT mode hardware, and this bit setting may be ignored in some modes. This bit should be set to 1 before writing to the external RAMDAC control register, and should be reset to 0 after the control register is written.

bit 3 External DAC RS2 Bit
This bit is used to control the logic level on output signal RS2. If this bit = 0, then the RS2 output is driven low. If this bit = 1, then the RS2 output is driven high. This bit should be set to 1 before the CPU accesses the external RAMDAC overlay and control registers. This bit should be set back to 0 for normal operation so that the standard RAMDAC palette registers may be accessed. When this bit is set to 1, reads and writes to I/O addresses 3C6h, 3C8h, and 3C9h will not access the 256 VGA palette locations in the internal LCD LUT, but instead will allow access the internal sprite palette registers, or external RAMDAC overlay registers.

bit 2 External DAC Read Select
The External DAC Read Select bit is used to enable reads of the external RAMDAC palette and overlay registers. When this bit = 0, reads to 3C6h, 3C8h, and 3C9h return values from the internal LCD gray scale lookup table registers or sprite palette. When this bit = 1, then reads to these addresses are decoded as external RAMDAC palette or overlay register reads. This bit should be set to 1 if an external RAMDAC is present.

bit 1 CRT Enable
When the CRT Enable bit = 1, CRT display hardware is enabled. When this bit = 0, CRT display hardware is disabled. This bit will be initialized to 0 on RESET. This bit works in conjunction with LCD Enable and TFT Enable. See the following table.

bit 0 LCD Enable

When the LCD Enable bit = 1, LCD display hardware is enabled. When this bit = 0, LCD display hardware is disabled. This bit will be initialized to 0 on RESET. Note that when LCD only is enabled (LCD Enable = 1, CRT Enable = 0), the active pixel clock is forced to the CLKI2 input, i.e. 28.322 MHz. This bit works in conjunction with CRT Enable. See the following table.

Table 0-55 Display Enable Bits

12-Bit RGB Mode Enable AUX[00] bit 6	TFT Enable Bit AUX[00] bit 5	CRT Enable Bit AUX[0B] bit 0	LCD Enable Bit AUX[0B] bit 1	Display Mode
0	0	0	0	no display
			1	Standard LCD mode
		1	0	CRT mode
			1	LCD DoubleScan mode
	1	0	X	9-bit TFT LCD mode
		1	X	9-bit TFT CRT mode / DoubleScan
1	1	0	X	12-bit TFT LCD mode
		1	X	12-bit TFT CRT mode / DoubleScan

0C Configuration Readback Register RO							
MD12 Status on Reset RO	MD11 Status on Reset RO	MD10 Status on Reset RO	MD9 Status on Reset RO	MD3 Status on Reset RO	MD2 Status on Reset RO	MD1 Status on Reset RO	MD0 Status on Reset RO

bits 7-4 MD[12:9] 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[12:9] 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.

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 1111b.

0D LCD Support Registers 2 RW							
XSCL Enable	LP Timing Select	WF Count Bit 5	WF Count Bit 4	WF Count Bit 3	WF Count Bit 2	WF Count Bit 1	WF Count Bit 0

bit7 XSCL Enable
This bit is used to adjust the XSCL shift clock output timing in monochrome mode. When this bit = 0, XSCL is masked off during the horizontal non-display period. When this bit = 1, XSCL is not masked off during the horizontal non-display period. Refer to “A.C. CHARACTERISTICS” on page 28 and the LCD panel manufacturer's specification to determine the correct setting of this bit. This bit does not affect either XSCL or XSCL2 in color panel modes.

bit 6 LP Timing Select
This bit is used to adjust the LP latch pulse output timing. When this bit = 0, then the LP latch pulse falling edge occurs 9 clock periods before the falling edge of the shift clock (XSCL). When this bit = 1, then the LP latch pulse falling edge occurs 4 clock periods before the falling edge of the shift clock. Refer to “A.C. CHARACTERISTICS” on page 28 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. The power up reset value of these bits is 20h. A 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.

0F Secondary Revision Code Register RO							
Secondary Revision Code Bit 7	Secondary Revision Code Bit 6	Secondary Revision Code Bit 5	Secondary Revision Code Bit 4	Secondary Revision Code Bit 3	Secondary Revision Code Bit 2	Secondary Revision Code Bit 1	Secondary Revision Code Bit 0

bits 7-0 Secondary Revision Code Bits [7:0]
The secondary revision code bits are read-only 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 08) are always set to 1 for the SPC8106. For the SPC8106, the Secondary Revision Code Register contains the value 63h. (The Secondary Revision Code for the SPC8106F0A is 60h and for the SPC8106F0B is 61h.)

10 Extended Function Register 4 RW							
n/a	n/a	Monitor ID Bit 2	Monitor ID Bit 1	n/a	n/a	DDC2 Monitor Support Bit 1	DDC2 Monitor Support Bit 0

bits 1-0 DDC2 Monitor Support Bits [1:0]
These two bits read 11b on power-up.
When bit 0 = 1 the MS1 pin is input only; when bit 0 = 0 the MS1 pin is forced low.
When bit 1 = 1 the MS2 pin is input only; when bit 0 = 0 the MS2 pin is forced low.

bits 5-4 Monitor ID Bits [2:1]
The Monitor ID Bits allow software to read the status on the monitor sense input pins MS[2:1]. These inputs are not latched and are expected to have external pullup resistors attached, so that if nothing is connected to them these register bits will read 11.

1A Scratch Register 0 RW							
Scratch Register Bit 7	Scratch Register Bit 6	Scratch Register Bit 5	Scratch Register Bit 4	Scratch Register Bit 3	Scratch Register Bit 2	Scratch Register Bit 1	Scratch Register Bit 0

bits 7-0 Scratch Register Bits [7:0]
These read/write bits have no effect in hardware.

1B Scratch Register 1 RW							
Scratch Register Bit 7	Scratch Register Bit 6	Scratch Register Bit 5	Scratch Register Bit 4	Scratch Register Bit 3	Scratch Register Bit 2	Scratch Register Bit 1	Scratch Register Bit 0

bits 7-0 Scratch Register Bits [7:0]
These read/write bits have no effect in hardware.

1C Relative Start Address High Register RW							
Relative Start Address High Bit 7	Relative Start Address High Bit 6	Relative Start Address High Bit 5	Relative Start Address High Bit 4	Relative Start Address High Bit 3	Relative Start Address High Bit 2	Relative Start Address High Bit 1	Relative Start Address High Bit 0

bits 7-0 Relative Start Address High Bits [7:0]
This register contains the 8 most significant bits of the 16-bit Relative Start Address of the screen buffer. The start address of the screen buffer is the sum of the 17-bit CRTC start address and the 16-bit Relative Start Address.

1D Relative Start Address Low Register RW							
Relative Start Address Low Bit 7	Relative Start Address Low Bit 6	Relative Start Address Low Bit 5	Relative Start Address Low Bit 4	Relative Start Address Low Bit 3	Relative Start Address Low Bit 2	Relative Start Address Low Bit 1	Relative Start Address Low Bit 0

bits 7-0 Relative Start Address Low Bits [7:0]
This register contains the 8 least significant bits of the 16-bit Relative Start Address of the screen buffer. The start address of the screen buffer is the sum of the 17-bit CRTC start address and the 16-bit Relative Start Address.

DE Auxiliary Enable Register RW							
n/a	n/a	n/a	Auxiliary Enable Code Bit 4	Auxiliary Enable Code Bit 3	Auxiliary Enable Code Bit 2	Auxiliary Enable Code Bit 1	Auxiliary Enable Code Bit 0

This register exists in all SPC81xx series video controllers and should be used as DEh to guarantee software portability.

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 or 1Eh). The Auxiliary Registers are disabled by writing any value to the Auxiliary Enable Register (index 0Eh or 1Eh), 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.

Miscellaneous Registers

Setup Mode Video Enable Register 102h RW							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	Video Subsystem Enable

Only the 3 least significant bits of this address are decoded.

bit 0 Video Subsystem Enable
Enables VGA for I/O and memory accesses.

Miscellaneous Output Register 3C2h W 3CCh R							
VRTC Polarity	HRTC Polarity	Odd/Even Page Bit	n/a	Clock Select Bit 1	Clock Select Bit 0	Dummy En DRAM (see note)	3Dx/3Bx Select

bit 1 of this register is a dummy read/write bit that has no effect in hardware. This read/write bit is provided for compatibility of some software which expects to read valid setting of this VGA standard bit.

bits 3-2 Clock Select Bits [1:0]
These bits are used to select which clock input (CLKI1 or CLKI2) is used as the active pixel clock. For normal operation, the CLKI1 input should be connected to a 25.175 MHz crystal or oscillator, and the CLKI2 input should be connected to a 28.322 MHz crystal or oscillator.

Table 0-56 Clock Select Bits

Clock Select Bit 1	Clock Select Bit 0	Clock input selected
0	0	CLKI1 (25.175 MHz)
0	1	CLKI2 (28.322 MHz)
1	0	CLKI1 (25.175 MHz)
1	1	CLKI2 (28.322 MHz)

Note that for LCD only operation (AUX[0B] bits 1:0 = 01), the active pixel clock is forced to 28.322 MHz (CLKI2) and the setting of the above bits is ignored. For any other setting of AUX[0B] bits 1:0, the Clock Select bits determine the active pixel clock. Even though it is not selected, the internal oscillator for the inactive clock is continues to run. For power save modes, both internal oscillators may be disabled by setting AUX[03] bit 4 = 1.

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

Page Select Register **3CDh R/W**							
n/a	Read Page Bit 2	Read Page Bit 1	Read Page Bit 0	n/a	Write Page Bit 2	Write Page Bit 1	Write Page Bit 0

Accessible only when AUX[06] bit 3 = 1.

bits 6-4 Read Page bits [2:0]
In 640 x 480 256 Color mode or Packed 4-bpp mode these bits select 1 of 8 pages of DRAM memory during CPU memory reads. In planer modes Read Page bit 2 selects the upper or lower 256 kbyte page of DRAM memory during CPU memory reads. These bits have no effect when the Sprite Write Mode Enable bit or the CPU Upper 256k Access bit is set to 1.

bits 2-0 Write Page Bits [2:0]
In 640 x 480 256 Color mode or Packed 4-bpp mode these bits select 1 of 8 pages of DRAM memory during CPU memory writes. In planer modes Write Page bit 2 selects the upper or lower 256 kbyte page of DRAM memory during CPU memory reads. These bits have no effect when the Sprite Write Mode Enable bit or the CPU Upper 256k Access bit is set to 1.

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

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

this port is accessible only if MD15 = 1 at the falling edge of RESET.

bit 0 Video Subsystem Enable
Setting this bit to 1 enables VGA for I/O and memory accesses.

Video Subsystem Enable Register **46E8h 56E8h 66E8h 76E8h WO**							
n/a	n/a	n/a	Setup Mode Enable	Video Subsystem Enable	n/a	n/a	n/a

this port is accessible only if MD15 = 0 at the falling edge of RESET.

bit 4 Setup Mode Enable
Setting this bit to 1 places the chip in Setup Mode so that only I/O port 102h is accessible.

bit 3 Video Subsystem Enable
Setting this bit to 1 enables VGA for I/O and memory accesses.

Sequencer Registers

Sequencer Index/Data Register **3C4h RW 3C5h RW**							
n/a	n/a	n/a	n/a	n/a	Sequencer Index Bit 2	Sequencer Index Bit 1	Sequencer Index 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)	8/9 Dot Select
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	256KB DRAM RO Status Read 1	n/a

index 01 Clocking Mode Register bits 4, 2, 1 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 1 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 RDY output is held low);

index 01 bit 0 - 8/9 Dot Select Bit, this bit only has an effect for CRT modes. If LCD Only mode is selected (AUX[0B] bits 1:0 = 0,1), then this bit is a read/write only bit with no effect in hardware.

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	Graphics Controller Index Bit 2	Graphics Controller Index Bit 1	Graphics Controller Index 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	Reference Color Bit 2	Reference Color Bit 1	Reference Color Bit 0
03 Data Rotate Register RW							
n/a	n/a	n/a	Logic Function Select Bit 1	Logic Function Select Bit 0	Data Rotate Count Bit 2	Data Rotate Count Bit 1	Data Rotate Count 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	Read Plane Select Bit 0
05 Mode Register RW							
n/a	Mode 13h Select	Shift Register Interleave	Odd/Even Plane Select	Read Mode Select	n/a	Write Mode Select Bit 1	Write Mode Select Bit 0
06 Miscellaneous Register RW							
n/a	n/a	n/a	n/a	Display Memory Map Bit 1	Display Memory Map Bit 0	Odd Even Chain Select	Graphics Mode
07 Color Don't Care Register RW							
n/a	n/a	n/a	n/a	Compare Plane Select Bit 3	Compare Plane Select Bit 2	Compare Plane Select Bit 1	Compare Plane Select Bit 0
08 Bit Mask Register RW							
Graphics Data Write Mask Bit 7	Graphics Data Write Mask Bit 6	Graphics Data Write Mask Bit 5	Graphics Data Write Mask Bit 4	Graphics Data Write Mask Bit 3	Graphics Data Write Mask Bit 2	Graphics Data Write Mask Bit 1	Graphics Data Write Mask Bit 0

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	Attributes Controller Index Bit 3	Attributes Controller Index Bit 2	Attributes Controller Index Bit 1	Attributes Controller Index 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							
EGA Palette Bits 4, 5 Control	Dummy Pel Width (see note)	Pixel Pan Compat	n/a	Blink/ Intensity	Line Graphics Char Code Enable	Mono Mode Select	Graphics Mode Select
11 Overscan Color Register RW							
Overscan Color Bit 7	Overscan Color Bit 6	Overscan Color Bit 5	Overscan Color Bit 4	Overscan Color Bit 3	Overscan Color Bit 2	Overscan Color Bit 1	Overscan Color Bit 0
12 Color Plane Enable Register RW							
n/a	n/a	Video Status RO status Read 0	Video Status RO status Read 0	Enable Plane 3	Enable Plane 2	Enable Plane 1	Enable Plane 0
13 Horizontal Panning Register RW							
n/a	n/a	n/a	n/a	Horizontal Pan Count Bit 3	Horizontal Pan Count Bit 2	Horizontal Pan Count Bit 1	Horizontal Pan Count Bit 0
14 Color Select Register RW							
n/a	n/a	n/a	n/a	Color Select Bit 3	Color Select Bit 2	Color Select Bit 1	Color Select 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 10 Mode Control Register bit 2 - the Line Graphics 8->9 Dot bit only has an effect for CRT modes. If LCD mode is selected (AUX[0B] bits 1:0 = 0,1), then this bit is a read/write only bit with no effect in hardware;

index 13 Horizontal Panning Register bit 3 - the Horizontal Pan Count bit 3 only has an effect for CRT modes. If LCD mode is selected (AUX[0B] bits 1:0 = 0,1), any pan value > 07h is treated as pan value = 0.

CRT Controller Registers

CRT Controller Index/Data Register 3B4/D4h RW 3B5/D5h RW							
n/a	n/a	CRT Controller Index Bit 5	CRT Controller Index Bit 4	CRT Controller Index Bit 3	CRT Controller Index Bit 2	CRT Controller Index Bit 1	CRT Controller Index Bit 0
00 Horizontal Total Register RW							
Horizontal Total Bit 7	Horizontal Total Bit 6	Horizontal Total Bit 5	Horizontal Total Bit 4	Horizontal Total Bit 3	Horizontal Total Bit 2	Horizontal Total Bit 1	Horizontal Total Bit 0
01 Horizontal Display Enable End Register RW (if AUX[00] bit 0 = 0)							
Horizontal DE End Bit 7	Horizontal DE End Bit 6	Horizontal DE End Bit 5	Horizontal DE End Bit 4	Horizontal DE End Bit 3	Horizontal DE End Bit 2	Horizontal DE End Bit 1	Horizontal DE End Bit 0
02 Horizontal Blanking Start Register RW							
Horizontal Blanking Start Bit 7	Horizontal Blanking Start Bit 6	Horizontal Blanking Start Bit 5	Horizontal Blanking Start Bit 4	Horizontal Blanking Start Bit 3	Horizontal Blanking Start Bit 2	Horizontal Blanking Start Bit 1	Horizontal Blanking Start Bit 0
03 Horizontal Blanking End Register RW (see note below)							
Dummy Read Select R/W (see note)	Dummy Display Skew R/W Bit 1 (see note)	Dummy Display Skew R/W Bit 0 (see note)	Horizontal Blanking End Bit 4	Horizontal Blanking End Bit 3	Horizontal Blanking End Bit 2	Horizontal Blanking End Bit 1	Horizontal Blanking End Bit 0
04 Horizontal Retrace Start Register RW							
HRTC Start Bit 7	HRTC Star Bit 6	HRTC Start Bit 5	HRTC Start Bit 4	HRTC Start Bit 3	HRTC Start Bit 2	HRTC Start Bit 1	HRTC Start Bit 0
05 Horizontal Retrace End Register RW							
Horizontal Blanking End Bit 5	Horizontal Retrace Delay Bit 1	Horizontal Retrace Delay Bit 0	HRTC End Bit 4	HRTC End Bit 3	HRTC End Bit 2	HRTC End Bit 1	HRTC End Bit 0
06 Vertical Total End Register RW							
VTotal End Bit 7	VTotal End Bit 6	VTotal End Bit 5	VTotal End Bit 4	VTotal End Bit 3	VTotal End Bit 2	VTotal End Bit 1	VTotal End Bit 0
07 CRTC Overflow Register RW							
VRTC Start Bit 9	VDisplay End Pos'n Bit 9	VTotal End Bit 9	Line Compare Bit 8	VBlank Start Bit 8	VRTC Start Bit 8	VDisplay End Pos'n Bit 8	VTotal End Bit 8
08 Preset Row Scan Register RW							
n/a	Byte Pan Bit 1	Byte Pan Bit 0	Preset Row Scan Bit 4	Preset Row Scan Bit 3	Preset Row Scan Bit 2	Preset Row Scan Bit 1	Preset Row Scan Bit 0
09 Maximum Scan Line Register RW							
Line Doubling Enable	Line Compare Bit 9	VBlank Start Bit 9	Max Scan Line Bit 4	Max Scan Line Bit 3	Max Scan Line Bit 2	Max Scan Line Bit 1	Max Scan Line Bit 0
0A Cursor Start Register RW							
n/a	n/a	Cursor Disable	Cursor Start Row Bit 4	Cursor Start Row Bit 3	Cursor Start Row Bit 2	Cursor Start Row Bit 1	Cursor Start Row Bit 0

CRT Controller Index/Data Register							
0B Cursor End Register RW							
n/a	Dummy Cursor Skew R/W Bit 1 (see note)	Dummy Cursor Skew R/W Bit 0 (see note)	Cursor End Row Bit 4	Cursor End Row Bit 3	Cursor End Row Bit 2	Cursor End Row Bit 1	Cursor End Row Bit 0
0C Start Address High Register RW							
Start Addr High Bit 15	Start Addr High Bit 14	Start Addr High Bit 13	Start Addr High Bit 12	Start Addr High Bit 11	Start Addr High Bit 10	Start Addr High Bit 9	Start Addr High Bit 8
0D Start Address Low Register RW							
Start Addr Low Bit 7	Start Addr Low Bit 6	Start Addr Low Bit 5	Start Addr Low Bit 4	Start Addr Low Bit 3	Start Addr Low Bit 2	Start Addr Low Bit 1	Start Addr Low Bit 0
0E Cursor Position High Register RW							
Cursor Position High Bit 15	Cursor Position High Bit 14	Cursor Position High Bit 13	Cursor Position High Bit 12	Cursor Position High Bit 11	Cursor Position High Bit 10	Cursor Position High Bit 9	Cursor Position High Bit 8
0F Cursor Position Low Register RW							
Cursor Position Low Bit 7	Cursor Position Low Bit 6	Cursor Position Low Bit 5	Cursor Position Low Bit 4	Cursor Position Low Bit 3	Cursor Position Low Bit 2	Cursor Position Low Bit 1	Cursor Position Low Bit 0
10 Vertical Retrace Start Register RW							
VRTC Start Bit 7	VRTC Start Bit 6	VRTC Start Bit 5	VRTC Start Bit 4	VRTC Start Bit 3	VRTC Start Bit 2	VRTC Start Bit 1	VRTC Start Bit 0
11 Vertical Retrace End Register RW							
Protect CRTC Registers 0-7	Dummy Refresh Cycles R/W Bit (see note)	Vertical Int Disable	Vertical Int Clear	VRTC End Bit 3	VRTC End Bit 2	VRTC End Bit 1	VRTC End Bit 0
12 Vertical Display Enable End Register RW (if AUX[00] bit 0 = 0)							
VDisplay End Pos'n Bit 7	VDisplay End Pos'n Bit 6	VDisplay End Pos'n Bit 5	VDisplay End Pos'n Bit 4	VDisplay End Pos'n Bit 3	VDisplay End Pos'n Bit 2	VDisplay End Pos'n Bit 1	VDisplay End Pos'n Bit 0
13 Offset Register RW							
Offset Bit 7	Offset Bit 6	Offset Bit 5	Offset Bit 4	Offset Bit 3	Offset Bit 2	Offset Bit 1	Offset Bit 0
14 Underline Location Register RW							
n/a	Double Word Select	Dummy Count by 4 R/W Bit (see note)	Underline Row Scan Bit 4	Underline Row Scan Bit 3	Underline Row Scan Bit 2	Underline Row Scan Bit 1	Underline Row Scan Bit 0
15 Vertical Blanking Start Register RW							
VBlank Start Bit 7	VBlank Start Bit 6	VBlank Start Bit 5	VBlank Start Bit 4	VBlank Start Bit 3	VBlank Start Bit 2	VBlank Start Bit 1	VBlank Start Bit 0
16 Vertical Blanking End Register RW							
VBlank End Bit 7	VBlank End Bit 6	VBlank End Bit 5	VBlank End Bit 4	VBlank End Bit 3	VBlank End Bit 2	VBlank End Bit 1	VBlank End Bit 0

CRT Controller Index/Data Register 17 Mode Control Register RW							
Retrace Enable RO Status Read 1	Byte/Word Mode Select	MA0 Select MA13/15	n/a	Dummy Count by 2 R/W	Dummy HRTC/2 R/W	MA14 Select MA14/rsc	Compat Mode Select
18 Line Compare Register RW							
Line Compare Bit 7	Line Compare Bit 6	Line Compare Bit 5	Line Compare Bit 4	Line Compare Bit 3	Line Compare Bit 2	Line Compare Bit 1	Line Compare Bit 0
24 Attribute Controller Addr/Data Flip-Flop 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” or “Dummy R/W Bit” 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;

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.
in LCD only mode (AUX[0B] bits 1,0 = 0,1), the following CRTC registers are ignored and are treated as Dummy R/W bits:
index 00, 02, 03, 04, 05, 06, 10, 15, 16 - all bits
index 07 - bits 7, 6, 5, 3, 2, 0
index 09 - bits 6, 5
index 11 - bits 3, 2, 1, 0
index 17 - bits 2, 3

LCD “B Set” Panel Size Registers

CRT Controller Index/Data Register 3B4/D4h RW 3B5/D5h RW							
n/a	n/a	CRT Controller Index Bit 5	CRT Controller Index Bit 4	CRT Controller Index Bit 3	CRT Controller Index Bit 2	CRT Controller Index Bit 1	CRT Controller Index Bit 0

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

bits 7-0 Horizontal Panel Size Bits [7:0]
The value in this register 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. This register is ignored if AUX[0B] bit 1 = 1, i.e. if CRT mode is enabled.

10 TFT Auto-Centering Adjust Register RW (if AUX[00] bit 0 = 1)							
n/a	n/a	Auto Centering Adjust Bit 5	Auto Centering Adjust Bit 4	Auto Centering Adjust Bit 3	Auto Centering Adjust Bit 2	Auto Centering Adjust Bit 1	Auto Centering Adjust Bit 0

bits 5-0 Auto Centering Adjust Bits [5:0]
This 6-bit value is added to the autocentering offset to center the image on a TFT display when the CRT is not enabled and autocentering is enabled. Because this register affects all autocentered (LCD) displays, it should be programmed to 0 for non-TFT displays.

11 TFT Supplementary Register RW (if AUX[00] bit 0 = 1)							
Vertical Panel Size MSB	Vertical Panel Size LSB	Vertical Start Bit 5	Vertical Start Bit 4	Vertical Start Bit 3	Vertical Start Bit 2	Vertical Start Bit 1	Vertical Start Bit 0

bit 7 Vertical Panel Size MSB
The MSB bit of the Vertical Panel Size is required to support TFT displays (i.e. displays with more than 511 lines). For all non-TFT panels this bit should be programmed to 0.

bit 6 Vertical Panel Size LSB
The LSB bit of the Vertical Panel Size is required to support TFT displays with an odd number of lines. For all non-TFT panels this bit should be programmed to 0.

bits 5-0 Vertical Start Bits [5:0]
This 6-bit value determines the number of blank lines following vertical retrace when CRT is not enabled. Passive LCD panels require 0 blank lines and TFT panels typically require 32 lines. This value has no effect when autocentering is enabled. This value should be programmed to 0 for all non-TFT panels.

12 Vertical Panel Size Register RW (if AUX[00] bit 0 = 1)							
Vertical Panel Size Bit 8	Vertical Panel Size Bit 7	Vertical Panel Size Bit 6	Vertical Panel Size Bit 5	Vertical Panel Size Bit 4	Vertical Panel Size Bit 3	Vertical Panel Size Bit 2	Vertical Panel Size Bit 1

bits 7-0 Vertical Panel Size Bits [7:0]
For dual-panel mode, this register contains the 8 most significant bits of the 9-bit number of lines in half the panel. For single-panel mode, this register contains the 8 most significant bits of the 9-bit number of lines in the panel. The least significant bit is always 0. For example, for a 480 line dual panel, this register is programmed with the 8 MSB of the 9-bit binary representation of 480÷2=240, i.e. 78h. For a 480 line single panel, this register is programmed with the 8 MSB of the 9-bit binary representation of 480 = F0h.

These LCD B Registers are accessed in CRTC register space by setting AUX[00] bit 0, LCD B Registers Program Enable, to 1. Also, if CRT mode is enabled, i.e. CRT Enable, AUX[0B] bit 1 = 1, these registers are ignored. The CRTC[11h] bit 7 Protect Registers has no affect on AUX B[01].

15 Vertical Non-Display Period Register RW (if AUX[00] bit 0 = 1)							
Vertical Non-Display Period Bit 8	Vertical Non-Display Period Bit 7	Vertical Non-Display Period Bit 6	Vertical Non-Display Period Bit 5	Vertical Non-Display Period Bit 4	Vertical Non-Display Period Bit 3	Vertical Non-Display Period Bit 2	Vertical Non-Display Period Bit 1

bits 7-0 Vertical Non-Display Period Bits [7:0]
The value of this register specifies the size of the non-display period in the number of lines minus 2. For example; a value of 0 indicates 2 non-display period lines.

Sprite/HW Cursor Registers

CRT Controller Index/Data Register 3B4/D4h RW 3B5/D5h RW							
n/a	n/a	CRT Controller Index Bit 5	CRT Controller Index Bit 4	CRT Controller Index Bit 3	CRT Controller Index Bit 2	CRT Controller Index Bit 1	CRT Controller Index Bit 0

the Sprite/Hardware Cursor registers are accessed via the CRTC index/data registers (3B4/3D4 and 3B5/3D5), at index 30h - 3Fh.

30 Sprite/HW Cursor X Position High Register RW							
n/a	n/a	n/a	n/a	n/a	n/a	Sprite/HW Cursor X Pos'n Bit 9	Sprite/HW Cursor X Pos'n Bit 8

bits 1-0 Sprite/HW Cursor X Position Bits [9:8]
This register contains the most significant two bits of the current horizontal left-most displayed pixel position of the sprite or hardware cursor.

31 Sprite/HW Cursor X Position Low Register RW							
Sprite/HW Cursor X Pos'n Bit 7	Sprite/HW Cursor X Pos'n Bit 6	Sprite/HW Cursor X Pos'n Bit 5	Sprite/HW Cursor X Pos'n Bit 4	Sprite/HW Cursor X Pos'n Bit 3	Sprite/HW Cursor X Pos'n Bit 2	Sprite/HW Cursor X Pos'n Bit 1	Sprite/HW Cursor X Pos'n Bit 0

bits 7-0 Sprite/HW Cursor X Position Bits [7:0]
This register contains the least significant eight bits of the current horizontal left-most displayed pixel position of the sprite or hardware cursor.

32 Sprite/HW Cursor Y Position High Register RW							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	Sprite/HW Cursor Y Pos'n Bit 8

bit 0 Sprite/HW Cursor Y Position Bit [8]
This register contains the most significant bit of the current horizontal upper-most displayed pixel position of the sprite or hardware cursor.

33 Sprite/HW Cursor Y Position Low Register RW							
Sprite/HW Cursor Y Pos'n Bit 7	Sprite/HW Cursor Y Pos'n Bit 6	Sprite/HW Cursor Y Pos'n Bit 5	Sprite/HW Cursor Y Pos'n Bit 4	Sprite/HW Cursor Y Pos'n Bit 3	Sprite/HW Cursor Y Pos'n Bit 2	Sprite/HW Cursor Y Pos'n Bit 1	Sprite/HW Cursor Y Pos'n Bit 0

bits 7-0 Sprite/HW Cursor Y Position Bits [7:0]
This register contains the least significant eight bits of the current horizontal upper-most displayed pixel position of the sprite or hardware cursor.

34 X Display Start Column Register RW							
n/a	n/a	X Disp Start Col Bit 5	X Disp Start Col Bit 4	X Disp Start Col Bit 3	X Disp Start Col Bit 2	X Disp Start Col Bit 1	X Disp Start Col Bit 0

bits 5-0 X Display Start Column Bits [5:0]
This register specifies the first displayed column of the sprite or cursor. When this register is set to 0, there is no clipping of the left side of the sprite or cursor image.

35 Y Display Start Row Register RW							
n/a	n/a	Y Disp Start Row Bit 5	Y Disp Start Row Bit 4	Y Disp Start Row Bit 3	Y Disp Start Row Bit 2	Y Disp Start Row Bit 1	Y Disp Start Row Bit 0

bits 5-0 Y Display Start Row Bits [5:0]
This register specifies the first displayed row of the sprite or cursor. When this register is set to 0, there is no clipping of the top of the sprite or cursor image.

36 Sprite/HW Cursor Address High Register RW							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	Sprite/HW Cursor Addr Bit 8

bit 0 Sprite/HW Cursor Address Bit [8]
This is the most significant bit of the address location of the top left hand point of the sprite or hardware cursor pattern stored in memory (see below).

37 Sprite/HW Cursor Address Low Register RW							
Sprite/HW Cursor Addr Bit 7	Sprite/HW Cursor Addr Bit 6	Sprite/HW Cursor Addr Bit 5	Sprite/HW Cursor Addr Bit 4	Sprite/HW Cursor Addr Bit 3	Sprite/HW Cursor Addr Bit 2	Sprite/HW Cursor Addr Bit 1	Sprite/HW Cursor Addr Bit 0

bits 7-0 Sprite/HW Cursor Address Bits [7:0]
These are the least significant 8 bits of the address location of the top left hand point of the sprite or hardware cursor pattern stored in memory. The complete 18 bit memory address for the sprite/HW cursor is formed as follows:

Table 0-57 Sprite/HW Cursor Address Bits

MA[17:9]	Sprite/HW Cursor Address [8:0]
MA[8:3]	Row [5:0]
MA[2:0]	Column [2:0]

The row bits represent the current row of the sprite/HW cursor being fetched (0-63). The column bits represent the upper 3 bits of the horizontal position of the current pixels being fetched from memory, with each fetch retrieving 16 bits from memory or 8 pixels of the Sprite/HW cursor image.

38 Sprite/HW Cursor Function Register RW							
n/a	n/a	n/a	n/a	Vertical Doubling Enable	Horizontal Doubling Enable	HW Cursor/ Sprite Select	Sprite Palette Scheme Sel

bit 3 Vertical Doubling Enable
When this bit is set to 1, the displayed image of the sprite/HW cursor is doubled in the vertical dimension, providing the sprite/HW cursor function is enabled. When this bit is set to 0, the sprite/HW cursor is not doubled. If the sprite/HW cursor function is not enabled, then this bit has no effect. Note that the origin of the sprite/HW cursor is not affected by enabling vertical line doubling.

bit 2 Horizontal Doubling Enable
When this bit is set to 1, the displayed image of the sprite /HW cursor is doubled in the horizontal dimension, providing the sprite/HW cursor function is enabled. When this bit is set to 0, the sprite/HW cursor is not doubled. If the sprite/HW cursor function is not enabled, then this bit has no effect. Note that the origin of the sprite/HW cursor is not affected by enabling horizontal line doubling.

bit 1 HW Cursor/Sprite Select
Provided that the sprite/HW cursor function is enabled (Auxiliary Register 05 bit 5 = 1), when this bit is set to 0, the sprite function is selected, and when this bit is set to 1, the hardware cursor function is selected. If the sprite/HW cursor function is not enabled, then this bit has no effect.

bit 0 Sprite Palette Scheme Select
This bit is used to select the palette scheme used for the sprite. The 2 bit sprite pixel color may represent either 1 of 4 colors, or 1 of 3 colors plus transparency. Provided that the sprite/HW cursor function is enabled (AUX[05] bit 5 = 1), when this bit is set to 0, the sprite pixel bits select 1 of 4 sprite palette entries as the value of the currently displayed pixel. When this bit is set to 1, the sprite pixel values 1 to 3 select sprite palette entries 1 to 3 respectively, and sprite pixel value 0 selects transparency, setting the current pixel to the value of the displayed image behind the sprite pixel.

The sprite palette entries 0-3 are accessed via the internal LUT register address space when AUX[0B] bit 3 = 1, at LUT index values 0Ch to 0Fh, respectively. See “LCD Gray Scale/Color Lookup Table Registers” on page 116 for details.

Sprite/HW Cursor Bit Definitions

The sprite and hardware cursor functions share common logic and cannot be displayed simultaneously. The following table defines the usage of the bits in off-screen memory representing the sprite or hardware cursor image:

Table 0-58 Sprite/HW Cursor Bit Definitions

		CRTC[38] bits 1,0 = 00	CRTC[38] bits 1,0 = 01	CRTC[38] bits 1 = 1
Pixel bit 1	**Pixel bit 0**	**Sprite Pixel Displayed**	**Sprite Pixel Displayed**	**Cursor Pixel Displayed**
0	0	Sprite Palette Entry 0	Screen	Cursor Background
0	1	Sprite Palette Entry 1	Sprite Palette Entry 1	Cursor Foreground
1	0	Sprite Palette Entry 2	Sprite Palette Entry 2	Screen
1	1	Sprite Palette Entry 3	Sprite Palette Entry 3	NOT Screen[a]

a. When CRT is enabled, the RAMDAC Index is inverted. The color is not inverted.

The pixel bits are packed in memory in the following configuration:

Table 0-59 Pixel Bit Memory Packing Configuration

	Address X								Address X+1							
Plane/ Cursor Bit	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
0/(Bit 0)	0	1	2	3	4	5	6	7	16	17	18	19	20	21	22	23
1/(Bit 1)	0	1	2	3	4	5	6	7	16	17	18	19	20	21	22	23
2/(Bit 0)	8	9	10	11	12	13	14	15	24	25	26	27	28	29	30	31
3/(Bit 1)	8	9	10	11	12	13	14	15	24	25	26	27	28	29	30	31

where:

X
X = Displayed Pixel X

LCD Gray Scale/Color Lookup Table Registers

The LCD Gray Scale/Color Lookup Table (LUT) Registers are provided for mapping pixels to gray scale values on a monochrome LCD display, and to color values on a color LCD display. For CRT modes, an external RAMDAC interface is provided. The LUT Registers share the same I/O address space as the external VGA RAMDAC, namely I/O addresses 3C6h, 3C8h, and 3C9h. Reads from these addresses will result in accesses to either the internal LUT or the external RAMDAC depending on the setting of the External DAC Read Select bit in AUX[0B] bit 2. If this bit is set to 0, then reads will access the internal LUT registers. If this bit is 1, then reads will access the external RAMDAC registers. For CRT modes, this bit should be set to 1. Note however that the External DAC RS2 Bit (AUX[0B] bit 3) must be set to 0 in order to allow accesses to the internal 256 VGA palette locations in the LUT registers. If AUX[0B] bit 3 is set to 1, then these 256 VGA palette locations in the internal LUT cannot be read or written.
I/O writes to the external RAMDAC registers and the internal LUT registers can be disabled by setting bits 7 and 6 of AUX[0B].

The internal LUT Register I/O address space is also used to address the 4 sprite palette locations. This internal sprite palette operates similarly to the external palette supported in the 477-type RAMDAC and may be accessed via the internal LUT address space when the External DAC RS2 Bit (AUX[0B] bit 3) is set to 1.

The internal LUT registers can be configured as either a 256x6 gray scale lookup table for monochrome LCD displays, or as a 256x12 color lookup table for color LCD displays (4 bits each of R, G, B). The lookup table configuration is selected by the value in AUX[02] bit 6. Similarly, this setting determines whether the internal sprite palette registers are configured as a 4x6 bit gray scale palette or a 4x12 bit color palette (4 bits each of R, G, B).

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

bits 7-0 Dummy LCD Lookup Table Pixel Mask 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 while in LCD mode.

RAMDAC/Lookup Table Status Register **3C7h RO**							
n/a	n/a	n/a	n/a	n/a	n/a	Read Write Mode Status Bit 1	Read Write Mode Status Bit 0

bits 1-0 RAMDAC/Lookup Table Status Bits [1:0]
Directly after a write to I/O address 3C7h (Lookup Table or RAMDAC Read Address Register), the Read/Write Mode Status bits 1-0 are both set to 1. Directly after a write to I/O address 3C8h (Lookup Table or RAMDAC Write Address Register), both these bits are set to 0.

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

bits 7-0 LCD Lookup Table Read Address Bits [7:0]
These 8 bits are used to select 1 of 256 gray scale/color lookup table registers to be read, when AUX[0B] bit 3 = 0.

When AUX[0B] bit 3 = 1, then this read address is used to select 1 of 4 internal sprite palette registers. In this case, only read addresses 0Ch - 0Fh select a valid sprite palette location. (In the current implementation, only the low 2 bits of the read address are actually decoded to select 1 of 4 internal sprite palette registers. However, the specific addresses 0Ch-0Fh should be used to maintain compatibility with the overlay registers used in the external RAMDAC.)

Once a valid read address is set in this register, the lookup table or sprite palette 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 RAMDAC/Lookup Table Status Register Read/Write Mode Status bits 1-0 are both set to 1.

When AUX[0B] bit 7 = 1 writes to the read address are NOT disabled.

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

bits 7-0 LCD Lookup Table Write Address Bits [7:0]
These 8 bits are used to select 1 of 256 lookup table registers to be written, when AUX[0B] bit 3 = 0.

When AUX[0B] bit 3 = 1, then this write address is used to select 1 of 4 internal sprite palette registers. In this case, only write addresses 0Ch - 0Fh select a valid hardware cursor/sprite palette location. (In the current implementation, only the low 2 bits of the write address are actually decoded to select 1 of 4 internal sprite palette registers. However, the specific addresses 0Ch-0Fh should be used to maintain compatibility with the overlay registers used in the external RAMDAC.)

Once a valid write address is set in this register, the lookup table or sprite palette 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 writes), the write address is automatically incremented.

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

When AUX[0B] bit 7 = 1 writes to this register are disabled.

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

bits 5-0 LCD Lookup Table Data Bits [5:0]
The use of the LCD Lookup Table Data Register varies depending on the setting of AUX[02] bit 6 and on the setting of the Green-only/NTSC Weighting Select bit (Auxiliary Register index 01, bit 4). The chip allows a color lookup or 2 different gray scale weighting schemes, applied to data as it is written to the lookup table.

Table 0-60 LCD Lookup Table Function Selection

AUX[02] bit 6	Aux [01] bit 4	GS Weighting Function Selected
1	X	Color LCD mode - bypass GS weighting circuit and enable 256x12 LUT

Table 0-60 LCD Lookup Table Function Selection

AUX[02] bit 6	Aux [01] bit 4	GS Weighting Function Selected
0	0	GS Weighting Mode 0 (NTSC) - enable 256x6 LUT GS = [19G + 9R + 4B] /32
0	1	GS Weighting Mode 1 (Green only)- enable 256x6 LUT GS = $G_5G_4G_3G_2G_1G_0$

Table 0-61 Color Mode Writes (AUX[02] bit 6 = 1)

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

Table 0-62 Color Mode Reads (AUX[02] bit 6 = 1)

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

Table 0-63 GS Weighting Modes 0 Writes (AUX[02] bit 6 = 1 and AUX[01] bit 4 = 0)

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

Table 0-63 GS Weighting Modes 0 Writes (AUX[02] bit 6 = 1 and AUX[01] bit 4 = 0)

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

GS = [19G + 9R + 4B] /32

Table 0-64 GS Weighting Mode 0 Reads (AUX[02] bit 6 = 0 and AUX[01] bit 4 = 0)

1st access (“Red”)							
n/a	n/a	Gray Scale Value Bit 5	Gray Scale Value Bit 4	Gray Scale Value Bit 3	Gray Scale Value Bit 2	Gray Scale Value Bit 1	Gray Scale Value Bit 0
2nd access (“Green”)							
n/a	n/a	Gray Scale Value Bit 5	Gray Scale Value Bit 4	Gray Scale Value Bit 3	Gray Scale Value Bit 2	Gray Scale Value Bit 1	Gray Scale Value Bit 0
3rd access (“Blue”)							
n/a	n/a	Gray Scale Value Bit 5	Gray Scale Value Bit 4	Gray Scale Value Bit 3	Gray Scale Value Bit 2	Gray Scale Value Bit 1	Gray Scale Value Bit 0

Table 0-65 GS Weighting Mode 1 Writes (AUX[02] bit 6 = 0 and 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	Green Component Data Bit 4	Green Component Data Bit 3	Green Component Data Bit 2	Green Component Data Bit 1	Green Component Data Bit 0
3rd access (“Blue”							
n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

GS Value = $G_5G_4G_3G_2G_1G_0$

Table 0-66 GS Weighting Mode 1 Reads (AUX[02] bit 6 = 0 and AUX[01] bit 4 = 1)

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

Table 0-66 GS Weighting Mode 1 Reads (AUX[02] bit 6 = 0 and AUX[01] bit 4 = 1)

1st access (“Red”)							
n/a	n/a	Green Component Data Bit 5	Green Component Data Bit 4	Green Component Data Bit 3	Green Component Data Bit 2	Green Component Data Bit 1	Green Component Data Bit 0

10.6 LCD Gray Scale/Color Lookup Table Architecture

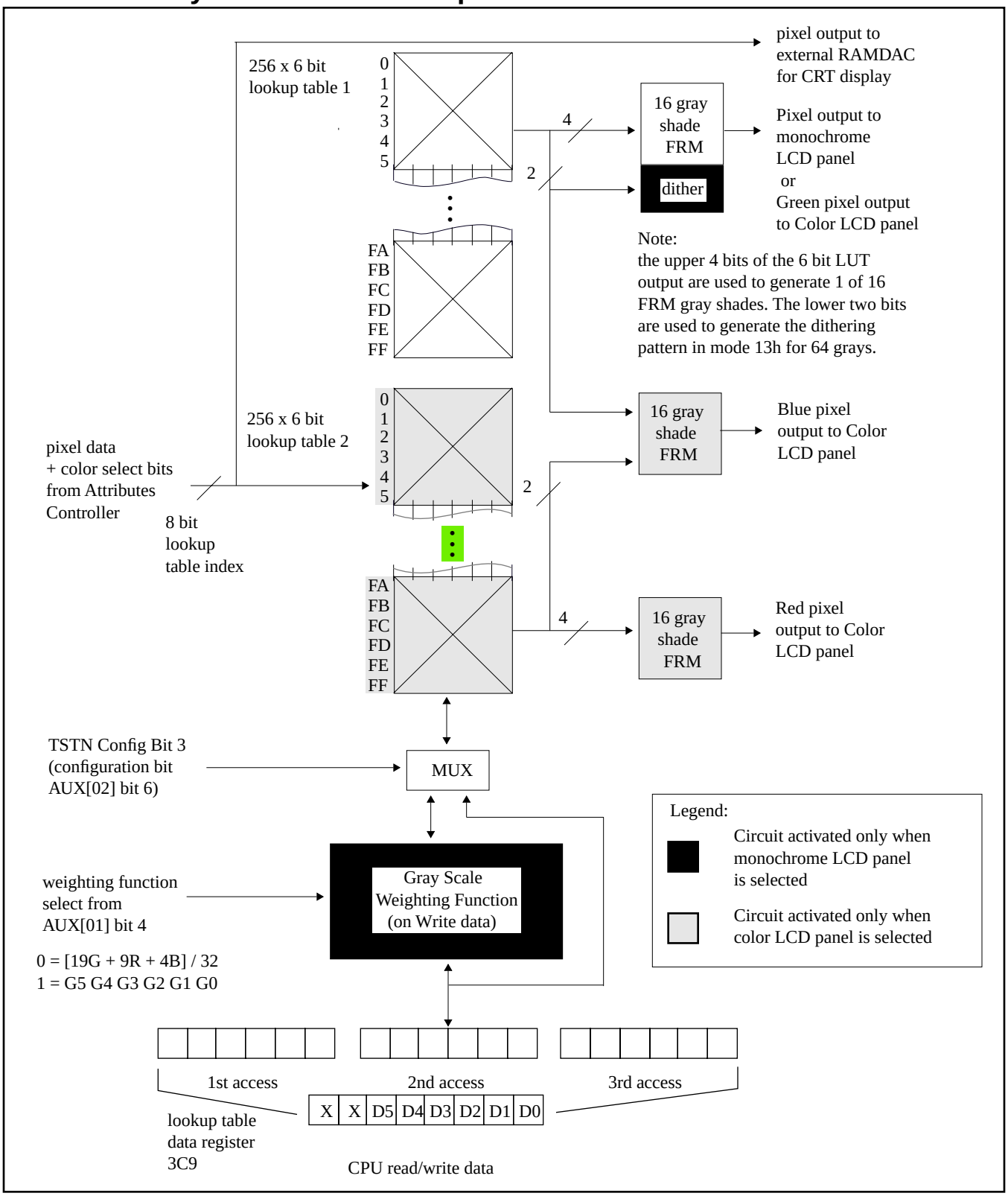


Figure 40 : LCD Gray Scale Lookup Table Architecture

Internal Sprite Palette

The internal sprite palette operates similarly to the external palette supported in the 477 RAMDAC. When the RS2 bit (AUX[0B] bit 3) is set to 1, accesses to the internal Look Up Table (LUT) are disabled and accesses to the internal sprite palette are enabled for index values 0Ch through 0Fh. The sprite palette is accessed in the same manner as the internal LUT, the only difference being the RS2 value.

This duplicates the functionality of the external RAMDAC and allows the sprite palettes for the CRT and the LCD to be written simultaneously, reducing the amount of code required (provided that the OL[0:1] and OL23 outputs are connected to the external RAMDAC and that MD[13] = 0 on the falling edge of RESET to allow use of the 477 RAMDAC's overlay registers). If the SPC8106 is set to operate with a color LCD, then the internal sprite palette values will match those of the external overlay registers. If the SPC8106 is set to operate with a monochrome LCD, then the color value written to the 477 overlay palette will be internally gray scaled using the same NTSC encoding used in the SPC8106 internal palette.

When AUX[0B] bit 7 = 1 writes to this register are disabled.

Table 0-67 Palette Location Accesses

RS2 Bit AUX[0B] bit 3	Ext DAC Read Select AUX[0B] bit 2	LUT Index	Palette Location Accessed (Writes)	Palette Location Accessed (Reads)
0	0	00-FFh	internal LUT and external RAMDAC VGA palette registers	internal VGA LUT registers
0	1	00-FFh	internal LUT and external RAMDAC VGA palette registers	external RAMDAC VGA palette
1	0	0C-0Fh	internal Sprite Palette Entries 0-3 and external RAMDAC overlay register 0Ch-0Fh	internal Sprite Palette Entry 0Ch-0Fh
1	1	0C-0Fh	internal Sprite Palette Entries 0-3 and external RAMDAC overlay register 0Ch-0Fh	external RAMDAC overlay register 0Ch-0Fh
1	X	00-0Bh 10-FFh	n/a	n/a

Sprite Palette Entry 0 (and overlay register 0) is the hardware cursor Background Color, when Hardware Cursor is enabled. Similarly, Sprite Palette Entry 1(and overlay register 1) is the hardware cursor Foreground Color, when Hardware Cursor is enabled.

11.0 PIN TEST MODE

To allow board level connectivity testing of the SPC8106, the pins have been defined such that all output pins except IRQ are actually bidirectional pins whose outputs are placed in tri-state when the SPC8106 is placed in pin test mode.

Pin test mode is enabled when any three of MEMR#, MEMW#, IOR# and IOW# are simultaneously low. When in scan mode the SPC8106 operates as shown in the following figure, with all the SPC8106 pins, except CLKI1, CLKI2, CLKO1, CLKO2 and IRQ, connected as inputs to a NAND-tree. The input order of the NAND-tree is given in the table on the following page. Note that the MEMR#, MEMW#, IOR# and IOW# inputs to the NAND tree are inverted allowing the signals to remain low, enabling scan mode, and still remain part of the NAND-tree.

To avoid race conditions all four of the MEMR#, MEMW#, IOR# and IOW# signals should be low before setting any of the MEMR#, MEMW#, IOR# and IOW# signals high during scan.

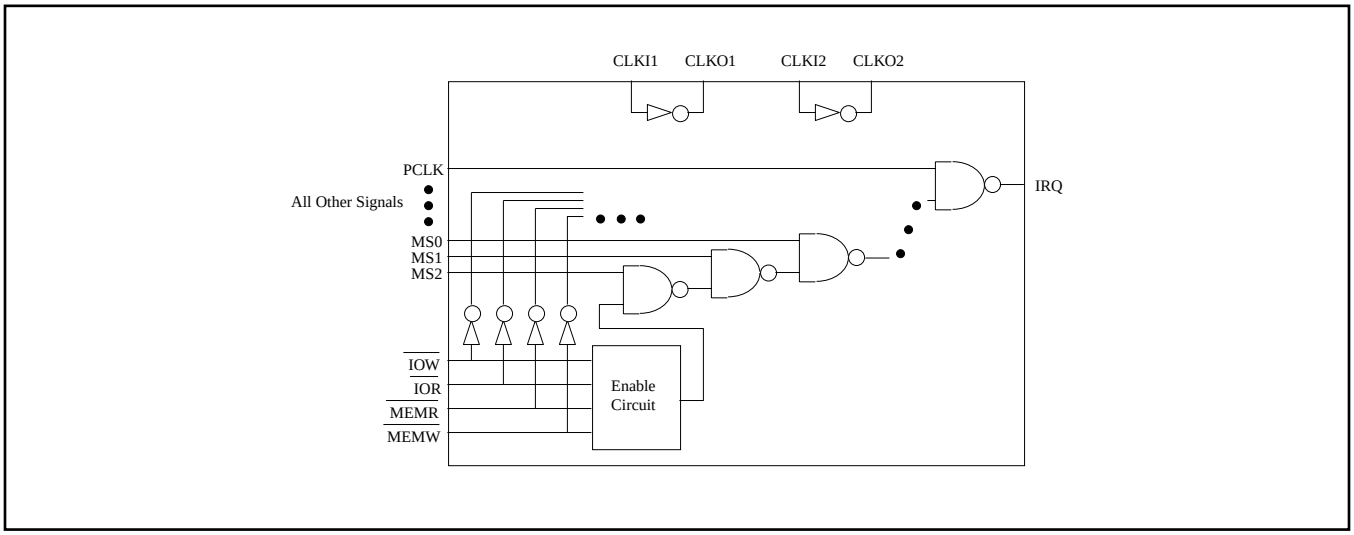


Figure 41 : SPC8106 Pin Test Configuration

Table 0-68 Pin Test Input Order in NAND Tree

Position	Signal	Position	Signal	Position	Signal
1	MS2	41	UD1	81	MD15
2	MS1	42	UD0	82	MD14
3	MS0	43	LD3	83	MD13
4	MEMW#	44	LD2	84	MD12
5	IOR#	45	LD1	85	MD11
6	IOW#	46	LD0	86	MD10
7	MEMR#	47	MA9	87	MD9
8	BHE#	48	MA8	88	MD8
9	MEMEN	49	MA7	89	MD7
10	IOEN	50	MA6	90	MD6
11	ALE	51	MA5	91	MD5
12	RESET	52	MA4	92	MD4

Table 0-68 Pin Test Input Order in NAND Tree (Continued)

Position	Signal	Position	Signal	Position	Signal
13	SUSPEND#	53	MA3	93	MD3
14	LCDPWR#	54	MA2	94	MD2
15	XSCL2	55	MA1	95	MD1
16	READY	56	MA0	96	MD0
17	MEMCS16#	57	LA23	97	D15
18	IOCS16#	58	LA22	98	D14
19	PDCLK	59	LA21	99	D13
20	WE#	60	LA20	100	D12
21	UCAS#	61	A19	101	D11
22	LCAS#	62	A18	102	D10
23	RAS#	63	A17	103	D9
24	XSCL	64	A16	104	D8
25	LP	65	A15	105	D7
26	YD	66	A14	106	D6
27	WF	67	A13	107	D5
28	BLANK#	68	A12	108	D4
29	VSYNC#	69	A11	109	D3
30	HSYNC#	70	A10	110	D2
31	OL23	71	A9	111	D1
32	OL1	72	A8	112	D0
33	OL0	73	A7	113	P0
34	IREFEN#	74	A6	114	P1
35	D477	75	A5	115	P2
36	RS2	76	A4	116	P3
37	DACWR#	77	A3	117	P4
38	DACRD#	78	A2	118	P5
39	UD3	79	A1	119	P6
40	UD2	80	A0	120	P7
				121	PCLK

12.0 PACKAGE DIMENSIONS (SPC8106F0C)

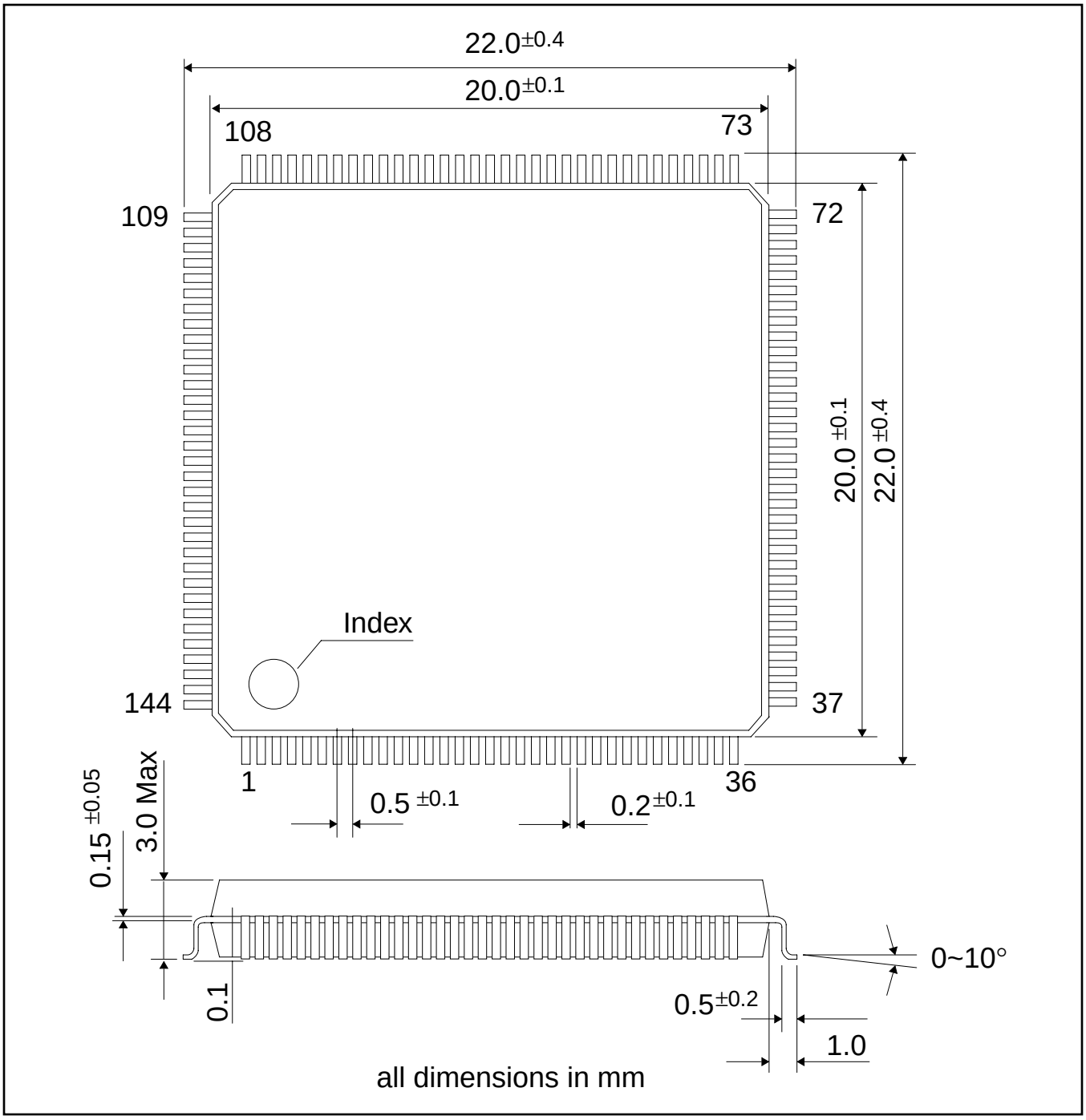


Figure 42 : 144 Pin QFP17 Package Dimensions

13.0 KNOWN LIMITATIONS

Software Power Save Modes 3 and 4

Although data writes to the internal LUT do not happen during Software Power Save Modes 3 and 4, the chip fails to prevent the latching of a write request during these Power Save Modes. The result is that the last attempted write to the LUT during Power Save Modes 3 and 4 will be latched and written to the LUT when the chip is returned to active state. A similar phenomenon will occur with the external palette registers during Power Save Modes 3 and 4.

To prevent the latched data from being written to the LUT, it is recommended that the AUX[0B] bits 6 and 7 be turned on before entering these Power Save Modes, then restoring them to 0 when leaving these modes. These bits prevent writes to the internal and external palette registers and will prevent the latched data from being written.

Appendix A REVISION HISTORY

Revision History: briefly describes the changes made to each issue of this document.

First Draft	05/10/95 modify and use X12-SP-001-03 as base for this specification apply correct styles to document
Second Draft	05/30/95 Change text and graphics throughout as needed to reflect SPC8106
Third Draft	05/31/95 miscellaneous changes throughout this manual
Fourth Draft	06/23/95 add TFT figures and timing
Fifth Draft	08/24/95 add 4-bit single mono panel timing changes to TFT panel timing add figure 38
Sixth Draft	10/27/95 minor corrections to timing numbers add table titles minor corrections to text
RELEASED	10/27/95 Revision 4, D.O.N. X12-SP-001-04
Draft 1	11/05/95 Revision 5 changes to Table 5.11 ~ add more information
RELEASED	11/08/95 Revision 5, D.O.N. X12-SP-001-05
Draft 1	04/18/96, Revision 6 pg 22, table 5-5, MS[2:0] ~ changed TTLS to TTL2 pg 30, removed reference to 8-Bit ISA Implementation App Note pg 91, fixed table 10-4 ~ CRT enable bit and LCD enable bit reversed pg 92, bit 0 ~ corrected LCD Panel Size register to Horizontal Size register pg 99, bits 7-6 corrected to bits 7-5 pg 102, xE Auxiliary Enable Register renamed to DE Auxiliary Enable Register and moved to pg 104
RELEASED	04/26/96 Revision 6, D.O.N. X12-SP-001-06

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

BIOS Functional Specification

Drawing Office No. X12-SP-002-03.1

THIS PAGE INTENTIONALLY BLANK

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 SPC8106 BIOS. It is intended for use by persons familiar with VGA BIOS functions and describes differences between the SPC8106 and a standard VGA BIOS. In addition this manual documents some of the behavior 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 machine architecture
- support for VESA VBE Core Functions version 1.2 and SOLLEX BIOS extensions
- support for VESA VBE/PM power management functions version 1.0 for BIOS version 1.14 or later
- five software power-save modes
- 3C3h or 46E8h video enable register supported (configurable)
- support for all standard IBM defined VGA modes, including modes 07h and 0Fh
- support for VESA VBE/DDC display data channel functions version 1.0 for BIOS version 1.2 or later
- selectable gray-scaling
- normal/reverse and autoswitch text/graphics of display polarity support
- autocenter support
- IRQ enable/disable 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 LCD panels of various resolutions from 320 x 200 to 640 x 480, based on MD line inputs
- supports CRT with External RAMDAC attached

2.4 Video Modes

LCD 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
13	Graphics	N/A	N/A	320 x 200	640 x 400	64	256	A000
100	Graphics	N/A	N/A	640 x 400	640 x 400	64	256	A000
101	Graphics	N/A	N/A	640 x 480	640 x 480	64	256	A000
108	Text	8 x 8	80 x 60	640 x 480	640 x 480	16	16	B800

CRT Display Modes

Mode No.	Mode Type	Font	Characters	Resolution	Displayed Pixels	Colors	Memory Segment
0	Text	8 x 8	40 x 25	320 x 200	640x400	16	B800
0+	Text	8 x 14	40 x 25	320 x 350	640x350	16	B800
0++	Text	9 x 16	40 x 25	360 x 400	720x400	16	B800
1	Text	8 x 8	40 x 25	320 x 200	640x400	16	B800
1+	Text	8 x 14	40 x 25	320 x 350	640x350	16	B800
1++	Text	9 x 16	40 x 25	360 x 400	720x400	16	B800
2	Text	8 x 8	80 x 25	640 x 200	640x400	16	B800
2+	Text	8 x 14	80 x 25	640 x 350	640x350	16	B800
2++	Text	9 x 16	80 x 25	720 x 400	640x400	16	B800
3	Text	8 x 8	80 x 25	640 x 200	640x400	16	B800
3+	Text	8 x 14	80 x 25	640 x 350	640x350	16	B800
3++	Text	9 x 16	80 x 25	720 x 400	640x400	16	B800
4	Graphics	N/A	N/A	320 x 200	640x400	4	B800
5	Graphics	N/A	N/A	320 x 200	640x400	4	B800
6	Graphics	N/A	N/A	640 x 200	640x400	2	B800
7	Text	8 x 14	80 x 25	640 x 350	640x350	2	B000
7+	Text	9 x 16	80 x 25	720 x 400	720x400	2	B000
0D	Graphics	N/A	N/A	320 x 200	640x400	16	A000
0E	Graphics	N/A	N/A	640 x 200	640x400	16	A000
0F	Graphics	N/A	N/A	640 x 350	640x350	2	A000
10	Graphics	N/A	N/A	640 x 350	640x350	16	A000
11	Graphics	N/A	N/A	640 x 480	640x480	2	A000
12	Graphics	N/A	N/A	640 x 480	640x480	16	A000
13	Graphics	N/A	N/A	320 x 200	640x400	256	A000
100	Graphics	N/A	N/A	640 x 400	640x400	256	A000
101	Graphics	N/A	N/A	640 x 480	640 x 480	256	A000
108	Text	8 x 8	80 x 60	640 x 480	640 x 480	16	B800

2.5 Simultaneous Display Support

Simultaneous Display is a feature in the SPC8106 that allows simultaneous display of both the CRT and LCD panel. Refer to the following table for supported video modes and limitations.

Simultaneous Display is not supported when using a Dual panel / Dual drive LCD.

All non-480 line modes (when using a 480 line panel) will show a wrap-around effect if Simultaneous Display is enabled. For example: display a 400 line mode on a 480 line panel and enable Simultaneous Display. You would see the top 78 lines duplicated on the bottom 78 lines (wrap-around affect).

If using a panel with less than 480 vertical lines, Simultaneous Display will be supported, however, the maximum LCD frame-rate may be violated. Therefore, the specific panel should be referenced.

If supporting a TFT panel requiring CRT-like timing (AUX[00] bit 5=1 and AUX[0B] bit 1=1) the panel handles the 350, 400 and 480 line modes, providing screen positioning internally. As the result of this direct support, Simultaneous Display is supported for all standard VGA and some

extended modes.

If supporting a TFT panel requiring LCD-like timing (AUX[00] bit 5=1 and AUX[0B] bit 1=0) the Simultaneous Display mode is not supported.

Mode No.	Single Panel (640x480)	TFT - Color 9/12-bit (640x480) AUX[00] bit 5=1 AUX[0B] bit 1=1
0	No	Yes
0+	No	Yes
0++	No	Yes
1	No	Yes
1+	No	Yes
1++	No	Yes
2	No	Yes
2+	No	Yes
2++	No	Yes
3	No	Yes
3+	No	Yes
3++	No	Yes
4	No	Yes
5	No	Yes
6	No	Yes
7	No	Yes
7+	No	Yes
0D	No	Yes
0E	No	Yes
0F	No	Yes
10	No	Yes
11	Yes	Yes
12	Yes	Yes
13	No	Yes
100	No	Yes
101	Yes	Yes
108	Yes	Yes

3.0 OVERVIEW DESCRIPTION

The SPC8106 is a single chip 5 Volt LCD/CRT video controller based on VGA architecture and optimized for driving an LCD panel display. VGA standard mode functionality is supported using standard IBM VGA parameters. A proprietary 256 x 6-bit gray-scale lookup table is provided to allow remapping of the 64 possible gray shades displayed on a monochrome LCD panel. A 256 x 12-bit lookup table (4 bits per primary) provides 256 out of a possible 4096 colors on a color LCD panel.

The target market for this device is notebook computers, or other specialized consumer products. The ability to run all 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 SPC8106.

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 SPC8106 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

```
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

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 (VGA)

```
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 (VGA)

```
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	
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	+These are values
1B	DW	Cursor type	|directly copied
1D	DB	Active page	|from low memory.
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)	

Table 0-1 Information Block Offset 2D

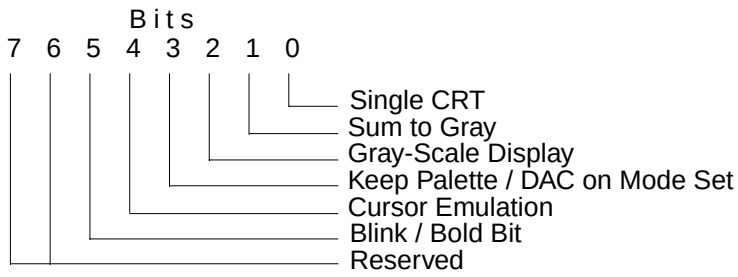
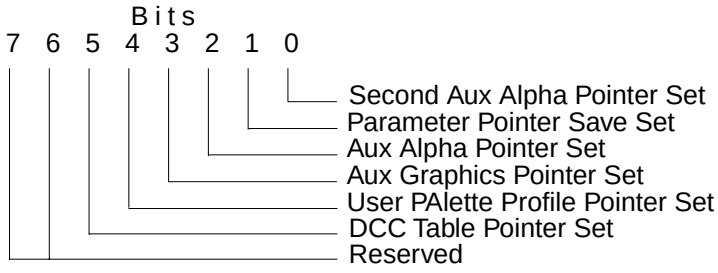


Table 0-2 Information Block Offset 32



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
```

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 section shows the SPC8106 implementation of these functions. The SPC8106 BIOS supports VESA VBE Core functions 1.2, VBE/PM functions 1.0 and VBE/DDC function 1.0. BIOS version 2.0 will support VBE Core functions 2.0.

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.

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 com-
                             plete 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

(valid for BIOS version 1.2 or greater)

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.

Function 15h - Display Identification Extensions (valid for BIOS version 1.2 or greater)

The VESA VBE sub-Function 15h is used to implement the VBE / DDC services. The VBE / DDC services are defined as follows:

Sub-Function 00h - Report VBE / DDC Capabilities

Input:	AH	= 4Fh	VESA Extension.
	AL	= 15h	VBE/DDC Services.
	BL	= 00h	Report VBE/DDC 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	=	Approximate time, in seconds, rounded up, to transfer one EDID block (128 bytes).
	BL	=	DDC level supported. bit 0 = 0 DDC1 not supported = 1 DDC1 supported bit 1 = 0 DDC2 not supported = 1 DDC2 supported bit 2 = 0 screen not blanked during data transfer = 1 screen blanked during data transfer
	CX	=	Unchanged
	ES:DI		Unchanged

All other registers may be destroyed.

Sub-Function 01h - Read EDID

Input:	AH	= 4Fh	VESA Extension.
	AL	= 15h	VBE/DDC Services.
	BL	= 01h	Read EDID.

	CX	= 00h	Controller unit number (00 = primary controller).
	DX	= 00h	EDID block number. Zero is the only valid value in version 1.0.
	ES:DI		Pointer to the area in which the EDID block (128 bytes) shall be returned.
Return:	AX	=	Status
	BH	=	Unchanged
	CX	=	Unchanged
	ES:DI		Pointer to the area in which the EDID block (128 bytes) is returned.

All other registers may be destroyed.

Sub-Function 02h - Read VDIF Block

Input:	AH	= 4Fh	VESA Extension.
	AL	= 15h	VBE/DDC Services.
	BL	= 02h	Read VDIF block.
	CX	= 00h	Controller unit number (00 = primary controller).
	DX	= 00h	VDIF block number (64 byte block).
	ES:DI		Pointer to the area in which the VDIF block (64 bytes) shall be returned.
Return:	AX	=	Status
	BH	=	Unchanged
	CX	=	Unchanged
	ES:DI		Pointer to the area in which the VDIF block (64 bytes) is returned.

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. The following has been marked up to show the SPC8106 implementation of these functions.

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 supported in SPC8106.

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

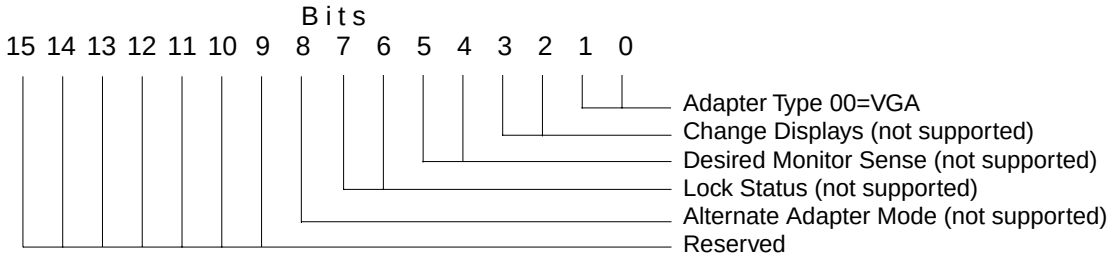
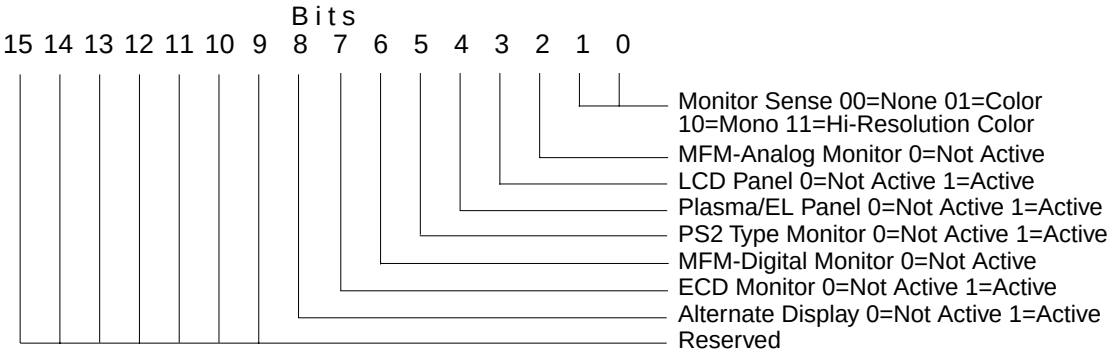


Table 0-4 Sub-Function 01h: Get Adapter



SubFunction 2 uses the same Adapter Request format as SubFunction 0 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

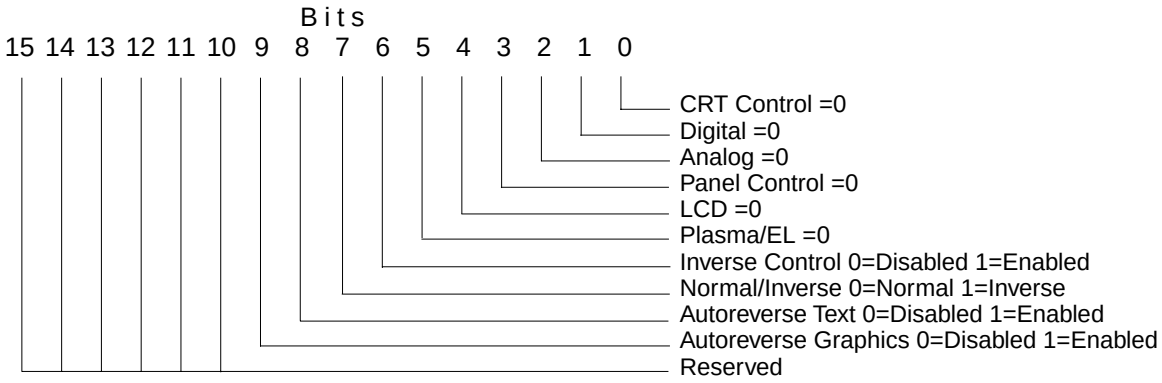
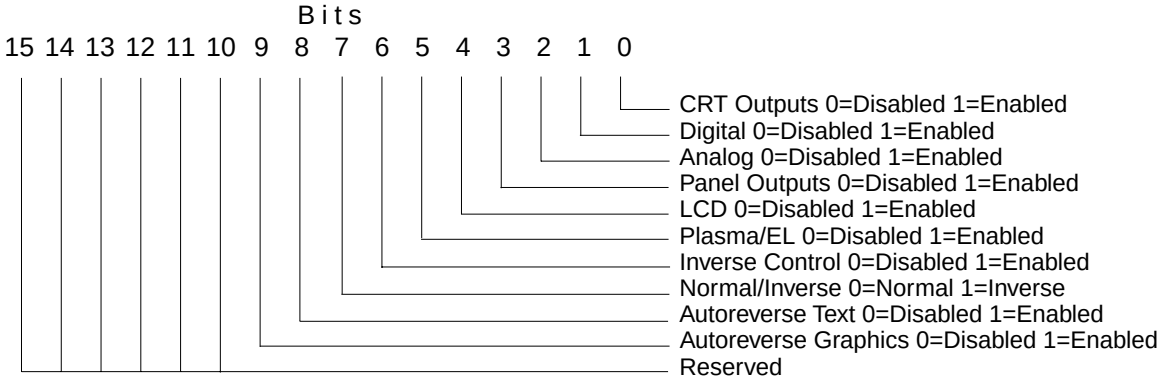
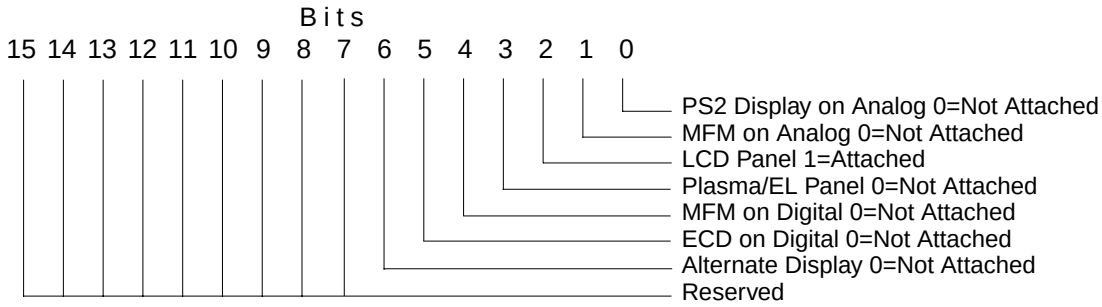


Table 0-6 Sub-Function 01h: Get Display Output



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

Table 0-7 Display Attached



Function 03h - Video Support Control

Not supported in SPC8106.

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 SPC8106)
         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 SPC8106)
         AL=04h   Power Control
         BL=03h   Get Time Out Reset
Return:  AX       Status AL=7Fh, AH=01h
```

SubFunction 00h: Set Power State

Set Power State according to table below:

SubFunction 01h: Get Power State

Returns Power State according to table below:

	State 0	State 1	State 2	State 3	State 4	State 5***
Clock	Yes	Yes	Yes	No	Yes*	Yes****
Display (panel) enabled	Yes	No	No	No	No	Yes
CRT display access	Yes	No	No	No	No	Yes
CPU to VRAM Refresh	Yes	Yes	No	No	No	Yes
VRAM Refresh	Yes	Yes	Yes	No	External	Yes
IO Write/IO Read	Yes	Yes	Yes	Aux Registers	Aux Registers	Yes
Ext.RamDAC	Yes	No**	No**	No	No	No
Relative Power Saving	None	Low	Medium	High	High	minimal

* can use Power Down clock if available
** read/write allowed
*** not available on CRT
**** Clock is slower by 25%

Function 05h - Load Register

Not supported in SPC8106.

Function 06h - Multiple Font Control

```
Input:   AH=7Fh   SOLLEX Support
         AL=06h   Multiple Font Control
         BL=00h   Set Multiple Font State
         CX       Multiple Font State
                  0h=off
                  1h=on
Return:  AX       Status

Input:   AH=7Fh   SOLLEX Support
         AL=06h   Multiple Font Control
         BL=01h   Get Multiple Font State
Return:  AX       Status
         BL       Multiple Font State
                  0h=off
                  1h=on
```

Function 07h - Fill Video RAM

Not supported in SPC8106.

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 SPC8106.

Function 0Ah - Non-Standard Font Control (SPC8106 uses 19pt.font for h/w text expansion*)

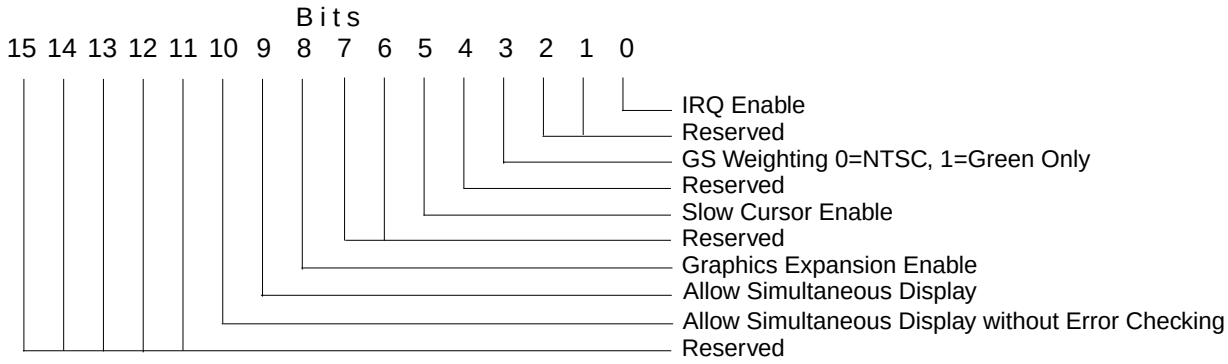
```
Input:   AH=7Fh   SOLLEX Support
         AL=0Ah   Non-Standard Font Control
         BL=00h   Set Non-Standard Font Control state
         CL       Font Width
         CH       Font Height (if CX=0, it will set normal system
                  font to be default)
Return:  AX       Status

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 Single Segment 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

The single segment structure of the BIOS versus a dual image structure (where the Extensions have their own segment) has the advantage that OS/2 2.0 will work correctly in the virtual DOS box.

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, 1993. 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 SPC8106 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.

0.0.0.1 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.

0.0.0.2 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 SPC8106 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.

Appendix A BIOS CONFIGURATION INPUTS

A.1 Panel Type

There are 8 BIOS configuration inputs in the SPC8106. The panel type configuration inputs come in on MD0-3 and are read back in Auxiliary Register Index 0Ch bits 0-3.

Pin Name	Pin No.	Configuration Pin Functionality
MD3	75	B3 of panel table
MD2	77	B2 of panel table
MD1	79	B1 of panel table
MD0	81	B0 of panel table
MD9	64	B4 of panel table

A.1.0.1 Panel Configuration Bits

MD9	MD3	MD2	MD1	MD0	Resolution	Mode	Bits	Comments
1	1	1	1	1	640x480	Dual	8	Monochrome
1	1	1	1	0	640x480	Single	8	Monochrome
1	1	1	0	1	640x400	Dual	8	Monochrome
1	1	1	0	0	640x400	Single	8	Monochrome
1	1	0	1	1	640x200	Dual	8	Monochrome
1	1	0	1	0	640x200	Single	4	Monochrome
1	1	0	0	1	480x320	Dual	8	Monochrome
1	1	0	0	0	480x320	Single	4	Monochrome
1	0	1	1	1	320x256	Dual	8	Monochrome
1	0	1	1	0	320x256	Single	4	Monochrome
1	0	1	0	1	320x240	Dual	8	Monochrome
1	0	1	0	0	320x240	Single	4	Monochrome
1	0	0	1	1	320x200	Dual	8	Monochrome
1	0	0	1	0	320x200	Single	4	Monochrome
1	0	0	0	1	320x240	Single	16	Color
1	0	0	0	0	640x480	Single	12	TFT MODE (CRT)
0	1	1	1	1	640x480	Dual	8	Color
0	1	1	1	0	640x480	Single	8	Color
0	1	1	0	1	640x480	Dual	16	Color
0	1	1	0	0	640x480	Single	16	Color
0	1	0	1	1	640x480	Dual		RGBI
0	1	0	1	0	640x480	Single	9	TFT MODE (CRT)
0	1	0	0	1	640x480	Dual	12	RGB
0	1	0	0	0	320x240	Single	4	Color
0	0	1	1	1	640x480	Single	12	TFT MODE (LCD)
0	0	1	1	0	640x480	Single	9	TFT MODE (LCD)
0	0	1	0	1				For OEM configuration
0	0	1	0	0				For OEM configuration
0	0	0	1	1				For OEM configuration
0	0	0	1	0				For OEM configuration
0	0	0	0	1				For OEM configuration
0	0	0	0	0				For OEM configuration

A.2 DRAM Refresh

Pin Name	Pin No.	Configuration Pin Functionality
MD10	67	32/4 ms DRAM refresh

This status is used by the BIOS at power up to set the 32/4 ms bit in Auxiliary Register 2 bit 0. If the bit is 0, then 4 ms DRAMs are present and Auxiliary Register 2 bit should be set to 0. If the bit is 1, then 32 ms DRAMs are present and the bit should be set to 1. The status of MD10 can be read at Auxiliary Configuration Readback Register 0 (3DFH index 0CH) bit 5.

A.3 Power Save Mode Clock Source

Pin Name	Pin No.	Configuration Pin Functionality
MD11	69	Power Save Mode 4 Clock Source bit 0
MD12	74	Power Save Mode 4 Clock Source bit 1

MD11 and MD12 are used by the BIOS during initialization to set the appropriate power save mode 4 clock source bits as per the table below. The status of MD11 can be read at Auxiliary Configuration Readback Register (3DFH index 0CH) bit 6. The status of MD12 can be read at Auxiliary Configuration Readback Register (3DFH index 0CH) bit 7.

	MD12	MD11
CLKI	0	0
MEMEN	0	1
Self Refresh	1	0
PDCLK	1	1

Appendix B POWER SAVE MODE DESCRIPTIONS

The SPC8106 BIOS has built in function calls to set the 4 Software Power Modes (see Sollex function summary). The BIOS programs the bits in Auxiliary register 03h to set the base mode, then programs some options. Some options are selected at power up, some at run-time. This section is a description of the resulting power states as set by the BIOS.

B.1 Software Power Save Mode 1

- No video display accesses to display memory.
- Sequencer is dedicated to CPU accesses to/from display memory.
- Display memory refresh is maintained and is generated from active CLKI input (28 MHz for LCD, 25 MHz or 28 MHz for CRT). Refresh rate can be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- I/O read/write of all registers is allowed.
- /LCDPWR signal forced high, IREFCNT is forced high.
- LCD interface output signals are low by default, but can be changed by 8106CFG.EXE.
- The External RAMDAC will be forced by software to a sleep state whether in LCD, CRT or Simultaneous Display mode.

B.2 Software Power Save Mode 2

Mode 2 has two states. Initially the chip enters State 1. If no display memory read or write is detected for about two horizontal lines, the chip enters State 2. If a display memory read or write is requested while in State 2, the chip returns to State 1 to service the display memory access.

State 1

- No video display accesses to display memory.
- Sequencer is dedicated to CPU accesses to/from display memory.
- Display memory refresh is maintained and is generated from active CLKI input (28 MHz for LCD, 25 MHz or 28 MHz for CRT). Refresh rate can be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- I/O read/write of all registers is allowed.
- /LCDPWR signal forced high, IREFCNT is forced high.
- LCD interface output signals are low by default, but can be changed by 8106CFG.EXE.
- The External RAMDAC will be forced by software to a sleep state whether in LCD, CRT or Simultaneous Display mode.

State 2

- No video display accesses to display memory.
- No CPU accesses to/from display memory.
- Sequencer is halted.
- I/O read/write of all registers is allowed.
- /LCDPWR signal forced high, IREFCNT is forced high.
- LCD interface output signals are low by default, but can be changed by 8106CFG.EXE.
- The External RAMDAC will be forced by software to a sleep state whether in LCD, CRT or Simultaneous Display mode.

B.3 Software Power Save Mode 3

- No video display accesses to display memory.
- No CPU accesses to/from display memory.
- Sequencer is halted.
- No display memory refresh.
- /LCDPWR signal forced high, IREFCNT is forced high.
- LCD interface output signals are low by default, but can be changed by 8106CFG.EXE.
- The External RAMDAC will be forced by software to a sleep state whether in LCD, CRT or Simultaneous Display mode.
- Disable Address decoding, allow access to Auxiliary Ports only.

B.4 Software Power Save Mode 4

- No video display accesses to display memory.
- No CPU accesses to/from display memory.
- Sequencer is halted.
- /LCDPWR signal forced high, IREFCNT is forced high.
- LCD interface output signals are low by default, but can be changed by 8106CFG.EXE.
- The External RAMDAC will be forced by software to a sleep state whether in LCD, CRT or Simultaneous Display mode.
- Display memory refresh is maintained and is generated from one of 3 selectable sources: 1) from the active CLKI input (28 MHz for LCD, 25 MHz or 28 MHz for CRT), 2) from the PDCLK pin (32 kHz 50% duty cycle, or 64 kHz with short low pulse duration), 3) or for ISA bus configuration only, from a clock source connected to pin MEMEN.
- Refresh rate generated from CLKI can be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- Refresh rate generated from MEMEN or PDCLK can also be selected: 64 kHz or 8 kHz, (for 256 cycle/4 ms, or 256 cycle/32 ms DRAM, respectively).
- Disable Address decoding, allow access to Auxiliary Ports only.

B.5 Software Power Save Mode 5 (not available on CRT)

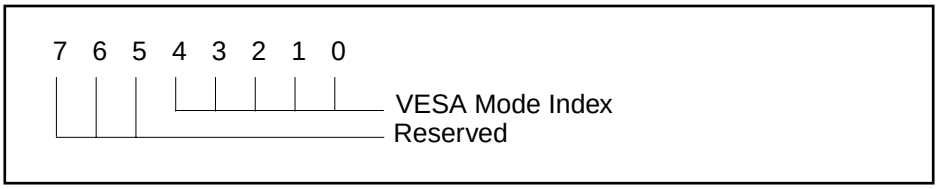
- Fully active mode except that gray-scaling cells are disabled.
- Three gray levels visible through on/off/50% duty cycle.
- Optionally the clock is slowed by 20%.

Appendix C BIOS USE OF INTERNAL SCRATCH BITS

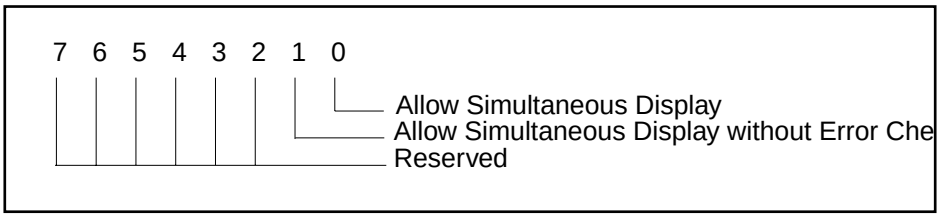
The BIOS uses scratch bits to store information, which has no equivalent in a standard IBM compatible BIOS. Changing a bit state has no immediate effect on the video state, scratch bits are typically processed at a mode set.

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

Auxiliary Register 4



Auxiliary Register A



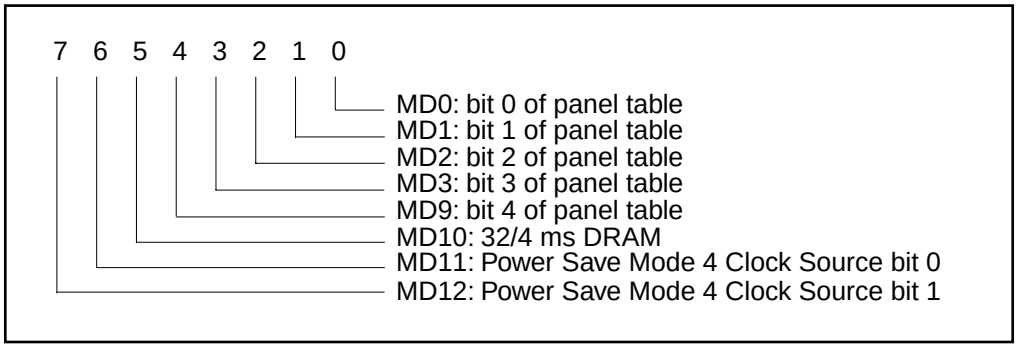
Auxiliary Register B

VESA Mode Index

This is an index into a table of possible VESA modes. If all four 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.

Auxiliary Register C (Configuration Readback Register)

Auxiliary register C contains the states of MD lines 0-3, 9-12. The BIOS interprets the states in the following manner:



Appendix D INITIALIZATION

D.1 (640x480 Single Color 8-Bit Display)

1. Turn off the active video system.

```
MOV      AX,1201H          ; Setup function 12
MOV      BL,32H            ; Video control
INT      10H               ; Turn off active video system
```

2. Enable the SPC8106 Chip.

```
3C3H = 1                   ; enable chip
```

3. Default SPC8106 to 3DxH and 28 Mhz clock. This clock is guaranteed to be present.

```
3C2H = 67H                 ; default to color addresses, 28 MHz clock
```

4. Enable AUX registers.

```
AUX[DEH] = 1AH             ; write magic number to AUX[DEH]
read AUX[DEH] data         ; to complete enable cycle
                           ; note: (DEH decodes to 0EH)
```

5. Program AUX registers to required default values.

```
AUX[0H] = 00000000b        ; IRQ disable, multifont disable,;
                           ; B-Regs program disable, select LCD panel
AUX[1H] = 01000011b        ; autocenter, slow cursor, NTSC, assume
                           ; single panel
AUX[2H] = 11000000b        ; CRT sprite, color panel, 8-bit data
AUX[3H] = 00000000b        ; Full power
AUX[4H] = 00000000b        ; Clear scratch register
AUX[5H] = 00000000b        ; Disable second memory bank
AUX[6H] = 00000000b        ; Disable page register at 3CD
AUX[7H] = 00000011b        ; Stretch text and graphics
AUX[9H] = 00000000b        ; No sprite memory mapping
AUX[0AH]= 00000000b        ; Clear scratch register
AUX[0BH]= 00000000b        ; LCD off, CRT off
AUX[0DH]= 00100000b        ; Set WF timing (may need modification)
AUX[1AH]= 00000000b        ; scratch pad
AUX[1BH]= 00000000b        ; scratch pad
AUX[1CH]= 00000000b        ; relative start address high
AUX[1DH]= 00000000b        ; relative start address high
```

6. Set video memory to A000H to prevent conflicts with MDA/CGA cards by programming the graphic controller.

```
Graphics Controller [6H] = 04H; set A000H addressing
```

7. We use Memory Data lines MD0-15 to set different chip states. These are usually set with DIP switches on the board. Eight of these lines are readable in Auxiliary register 0CH (MD0-3 are bits 0-3, MD9-12 are bits 4-7). They have no hardware effect so BIOS developers can redefine them freely. We use this information to

determine panel type (single or dual), DRAM refresh timing (32 or 4 ms) and power save mode 4/suspend clock source select. If a hardware configuration is fixed these values could be hard coded in step 5 above. The bits a developer will likely change are listed below.

panel type	AUX[0H] bits 5,6,7
single/dual panel	AUX[1H] bit 0
refresh 32/4 ms	AUX[2H] bit 0
power down clock	AUX[2H] bits 2,3
WF timing	AUX[0DH]

8. Program the CRTC-B registers (CRTC-B[1H] and CRTC-B[12H]) to required frame rate and vertical panel sizes. This is done by enabling access to the B set registers, programming the values and then disabling access to the B set.

```
AUX[00H]b0 = 1              ; enable B set read/write access
CRTC(3D4H) index 01H = 050H ; for 640 pixels wide panel
CRTC(3D4H) index 10H = 0    ; TFT related
CRTC(3D4H) index 11H = 0    ; TFT related
CRTC(3D4H) index 12H = 0F0H ; for a 480 single panel (8 MSB of 480)
CRTC(3D4H) index 15H = 0    ; 2 non-display lines
AUX[00H]b0 = 0              ; disable B set r/w access
```

9. Wake up external RAMDAC. We assume the external RAMDAC is present. A more rigorous initialization code would do presence check here.

```
AUX[0BH] = 00011110b        ; enable writes to external DAC
                            ; ctrl register
Initialize RAMDAC
```

10. Turn on either display panel or CRT. In either case, we select external RAMDAC read. Enable panel display if no monitor is detected. We use AUX[08H] bits 0, 1, 2 (monitor ID bits) to detect the presence of a monitor. If AUX[08H] & 07H = 07H, no monitor is present.

```
AUX[0BH] = 00101010b        ; enable LCD + External DAC read select
                            ; disable writes to external DAC ctrl register
                            ; enable anti-sparkle/power down RAMDAC
                            ; (any time LCD is enabled with no CRT - power down
                            ;  the RAMDAC)
        If a monitor is detected enable the CRT.
AUX[0BH] = 00100110b        ; enable CRT + External DAC read select
```

11. Do standard VGA initialization.

```
set up interrupt vectors 42H, 10H, 6DH
do CGA/MDA detection
...
set mode 3 (or 7 if CGA present)
```

12. Print signon message.

Appendix E BIOS DEFAULT CONFIGURATION OPTIONS

The BIOS has many customizable features:

- Sign on Message
  The power up displayed message can be altered or removed using the supplied 8110CFG program.
- Auto-Center
  The default state of Auto-Center can be set.
- Normal/Reverse/AutoReverse Text/Autoreverse Graphics
  The default screen color polarity for monochrome panels can be set.
- Gray-scale Weighting Default Algorithm (Green only/NTSC)
  The default gray-scale weighting can be set.
- Text/Graphics Expansion default
  The default state of expansion can be set.
- Default Segment (either C000h/E000h/F000h)
  The default BIOS segment can be set.
- Slow Cursor Enable
  The default cursor blink speed can be set.
- Default Cursor Shape
  The default cursor shape can be set.
- Panel Frame rate
  Custom panel frame rates can be changed.
- DAC ISA/PS/2
  The default DAC programming, IBM ISA format or IBM PS/2 format can be chosen.
- WF Timings
  WF timings can be altered.
- Custom Panel Type (Vert. size, 4/8 bit I/F, single/dual)
  Custom panel types and sizes can be added.
- Mode 7/F on color
  Support for Mode 7/F when running as a color VGA can be enabled, or like the IBM ISA VGA adapter, it can be disabled.
- Power Save State
  The default power save states can be changed, and DPMS added/removed.

E.1 Supported RAMDACs

The BIOS supports the following RAMDACs:

- BT481
- BT497
- BT477

Appendix F VESA INFORMATION

F.1 VESA Structures

The VBE 1.2 compatible information block has the following structure:

```
VGAInfoBlock struc
    VESASignature     db 'VESA'; 4 signature bytes
    VESAVersion       dw 102h; 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          db 236 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        dw 0200h; VBE Version
    OemStringPtr      dd ?   ; Pointer to OEM String
    Capabilities      db 4 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          db 222 dup (?); Reserved for VBE implementation
                             ; scratch area
    OemData           db 256 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 (?); remainder of ModeInfoBlock
ModeInfoBlock ends
```

F.2 VESA Mode Support

F.2.1 VESA Graphics Modes

15-bit Mode Number	7-bit Mode Number	Resolution	Colors	Supported
100h	-	640x400	256	Yes
101h	-	640x480	256	Yes
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

F.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

F.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 G SPECIAL CASES

One of the most useful features in the SPC8106 is the Hardware Cursor/Sprite. This affects the BIOS in the following ways:

- The BIOS disables the cursor on Mode Sets as it is the responsibility of the mouse driver to re-initialize the cursor.

Appendix H SIMULTANEOUS DISPLAY ISSUES

Here are the BIOS limitations for this display mode.

- Simultaneous Display will only be officially supported in 480 scan line modes.
- The BIOS will use two reserved bits in Auxiliary Register 0Ah to determine if Simultaneous Display should occur on a mode set. This will follow the rules set in the following table.
- Simultaneous Display will only work on a Single Panel.
- Autocenter and Stretch will be hardware suppressed during Simultaneous Display, they will continue to read-back as though they were on.
- All power save modes will continue to work in Simultaneous Display.
- LCD frame rate may change when Simultaneous Display occurs.

The basic rules for Simultaneous Display:

```
IF Allow SimultaneousDisplay = 1
      IF Mode 11/12/108 OR Allow SimultaneousDisplay Without Error Checking = 1
               turn LCD on
               turn CRT on
      ELSE
               IF CRT Available
               disable LCD
               turn CRT on.
               ENDIF
      ENDIF
ENDIF
```

	Allow SimultaneousDisplay = 0	**Allow SimultaneousDisplay = 1**
Allow SimultaneousDisplay Without Error Checking = 0	On a mode set: no Simultaneous Display will occur	On a mode set: If mode 11/12/101/108, enable LCD, but only if originally on CRT. If not mode 11/12/101/108, set CRT active, disable LCD.
Allow SimultaneousDisplay Without Error Checking = 1	On a mode set: no Simultaneous Display will occur	On a mode set: all modes will be Simultaneous Display

Appendix I REVISION HISTORY

Initial Draft	08/30/95
Second Draft	10/31/95
	changes to Section 2
	minor changes to Section 4
	minor changes to Section 5 ~ 5.3 moved to end of section
	changes to Section 6 ~ removed Functions 00, 03, 05, and 07
	changes to Section 7 ~ Panel Configuration Bits table
	minor changes to Section 8
	changes to Section 9
	minor changes to Section 10
	minor changes to Appendix A
	major changes to Appendix C ~ register changes added
	rewrite Appendix D
RELEASED	Revision 1, 11/07/95
Draft 1	Revision 3, 04/22/96
	re-arrange entire document
	formatting changes throughout
	change doublescan to simultaneous display
RELEASED	Revision 3, 04/30/96

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

Programming Notes and Examples

Drawing Office No. X12-AN-003-02

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY BLANK

1.0 INITIALIZING THE SPC8106 IN MODE 03H WITHOUT A BIOS

The purpose of this document is to demonstrate how to initialize the SPC8106 into a mode similar to IBM's Standard VGA mode 3 by supplying sample code. The steps involved in this are:

- Enabling the SPC8106
- Unlocking the Auxiliary Port
- Setting up the SPC8106 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 RAMDAC 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 SPC8106 for dual monochrome 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 code was compiled using 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 segment.
      mov     ds, ax          ;
      mov     es, ax          ;
      cli                     ; Interrupts off while changing the stack.
      mov     ss, ax          ;
      mov     sp, 0FFFFh      ;
      sti                     ; Stack change done - restore interupts.

      call    Step1           ; Enable SPC8106 chip.
      call    Step2           ; Unlock the Aux port.

      ; at this point, all the registers are accessible, we need to
      ; program Auxiliary 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           ; init 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 SPC8106 registers for '
      DB '8x8 font text mode without BIOS.',0

;***********************************************************************
;
;  STEP1 - Before ANY ports can be accessed, the SPC8106FOC chip must be
;        enabled.
;
;***********************************************************************

Step1:
      mov     dx,3C3h
      mov     al,1
      out     dx,al
      ret

;;; if chip enabled at 46E8, use the following code instead:
;;;
;;;
;;;Step1:
;;;   mov     si, OFFSET Enable46E8
;;;   call    RunTable
;;;   ret
;;;
;;;Enable46E8:
;;;   DW      PW+46E8h,16h
;;;   DW      PW+0102h,01h
;;;   DW      PW+46E8h,0Eh
;;;   DW      PW+4AE8h,00h
;;;   DW      -1
```

```
;*********************************************************************
;
;  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,00000001b  ; enable CRTC-B set
           DB -1              ; end of 3DE table

           DW 03D4h
           DB  01h,80          ; CRTC_B[01h] = 80 (dec)
           DB  10h,0           ; CRTC_B[10h] = 0 (TFT stuff)
           DB  11h,0           ; CRTC_B[11h] = 0 (TFT stuff)
           DB  12h,480/4       ; CRTC_B[12h] = 8 msb of 480/2
           DB  15h,0           ; CRTC_B[15h] = 0 ( 2 non-display lines)
           DB -1               ; end of 3D4 table
```

```
          DW  03DEh
          DB  00h,00000000b  ; IRQ dis.,multifont dis.,disable CRTC-B set
          DB  01H,01000010b  ; autocenter, slow cursor, dual panel
          DB  02H,00000010b  ; LOW state LCD, 8 bit data, mono panel
          DB  03H,00000000b  ; Full power
          DB  04H,00000000b  ; clear scratch register
          DB  05H,00000000b  ; disable second memory bank
          DB  06H,00000000b  ; disable page register at 3CDh
          DB  07H,00000000b  ; no stretch font,64 gs,dithering
          DB  09H,00000000b  ; No funny sprite stuff
          DB  0AH,00000000b  ; clear scratch register
          DB  0BH,00000001b  ; enable LCD
          DB  0DH,00100000b  ; WF defaults
          DB  1AH,00000000b  ; Clear scratch pad
          DB  1BH,00000000b  ; Clear scratch pad
          DB  1CH,00000000b  ; Clear relative start address high
          DB  1DH,00000000b  ; Clear relative start address low
          DB -1              ; end of 3DE table
          DW 03CEh
          DB  06h,00000100b  ; set A000 addressing
          DB  -1             ; end of 3CE table
          DW -1     ; end of the whole 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        ; ...stop
      DB       01h, 01h
      DB       02h, 03h
      DB       03h, 00h
      DB       04h, 02h
      DB       00h, 03h         ; ...reset
      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
```

```
;*********************************************************************
;
;    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

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
```

```
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
```

```
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       ; Video RAM segment
      mov     es, ax           ;
      mov     cx, 4000h        ; 16k Words = 32k Bytes
      xor     di, di           ;
      mov     ax, 720h         ; normal attribute(7h), ASCII space(20h)
      cld
      rep     stosw            ; initialize
      ret



; 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 SPC8106 IN MODE 12H WITHOUT A BIOS

The purpose of this document is to demonstrate how to initialize the SPC8106 into a mode similar to IBM's Standard VGA mode 12h by supplying sample code. The steps involved in this are:

- Enabling the SPC8106
- Unlocking the Auxiliary Port
- Setting up the SPC8106 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 SPC8106FOC for 640x480 Single Color LCD panel.
; Sets up registers for mode 12h without the BIOS.
;
; This program assumes:
;       - no external RAMDAC
;       - chip enable at port 3c3h.
;       - 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
Mode12PROC
      cld                     ; Set the direction flag.
      mov     ax, cs          ; Ensure that DS and ES point to this segment.
      mov     ds, ax          ;
      mov     es, ax          ;
      cli                     ; Interrupts off while changing the stack.
      mov     ss, ax          ;
      mov     sp, 0FFFFh      ;
      sti                     ; Stack change done - restore interupts.

      call    Step1           ; Enable SPC8106FOC chip.
      call    Step2           ; Unlock the AUX port.

      ; at this point, all the registers are accessible, we need to
      ; program Auxiliary 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, independent of CRT or panel type

      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 12 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 SPC8106FOC chip must be
;        enabled.
;
;*********************************************************************

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

;;; if chip enabled at 46e8, use the following code instead:
;;;
;;;
;;;Step1:
;;;   mov     si, OFFSET Enable46E8
;;;   call    RunTable
;;;   ret
;;;
;;;Enable46E8:
;;;   DW      PW+46E8h,16h
;;;   DW      PW+0102h,01h
;;;   DW      PW+46E8h,0Eh
;;;   DW      PW+4AE8h,00h
;;;    DW     -1

;*********************************************************************
;
;  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)

SetupChipTblDW PW+03C2h,67h; Default card to 3Dx, 28Mhz Clock
              DW 03DEh
              DB  00h,00000001b  ;   enable CRTC-B set
              DB -1

              DW 03D4h
              DB  01h,80          ; CRTC_B[01h] = 80 (dec)
              DB  10h,0           ; CRTC_B[10h] = 0
              DB  11h,0           ; CRTC_B[11h] = 0
              DB  12h,480/2       ; CRTC_B[12h] = 8 msb of 480
              DB  15h,0           ; CRTC_B[15h] = 0 ( 2 non-display lines)
              DB -1

              DW 03DEh
              DB  00h,00000000b  ; IRQ disable,multifont disable,disable CRTC-B set
              DB  01H,01000011b  ; autocenter, slow cursor, single panel
              DB  02H,01000010b  ; LOW state LCD, 8 bit data, color panel
              DB  03H,00000000b  ; Full power
              DB  04H,00000000b  ; clear scratch register
              DB  05H,00000000b  ; disable second memory bank
              DB  06H,00001000b  ; Enable page register at 3CDh
              DB  07H,00000000b  ; no stretch font,64 gs,dithering
              DB  09H,00000000b  ; No funny sprite stuff
              DB  0AH,00000000b  ; clear scratch register
              DB  0BH,00000001b  ; enable LCD
              DB  0DH,00100000b  ; defaults
              DB  1AH,00000000b  ; Clear scratch pad
              DB  1BH,00000000b  ; Clear scratch pad
              DB  1CH,00000000b  ; Clear relative start address high
              DB  1DH,00000000b  ; Clear relative start address low
              DB -1
```

```
              DW 03CEh
              DB 06h,00000100b   ; set A000 addressing
              DB -1
              DW -1


;*****************************************************************
;
;  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.
;
;**********************************************************************

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

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 - 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 INITIALIZING THE SPC8106 IN MODE 101H WITHOUT A BIOS

The purpose of this document is to demonstrate how to initialize the SPC8106 into a mode similar to VESA SVGA mode 101h (640x480x256 colors) by supplying sample code. The steps involved in this are:

- Enabling the SPC8106
- Unlocking the Auxiliary Port
- Setting up the SPC8106 panel specific registers
  In this case for a Dual STN, Color 640x480 LCD panel
- Setting up the Standard VGA registers for Mode 101h
- Setting up the attribute registers
- Initializing the RAMDAC values to values used for Mode 13h
- Initializing 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 101h video memory will work correctly. For more information on paging for this mode, please refer to the SPC8106 Hardware Functional Specification.

```
; MODE101.ASM
;
; Programs the SPC8106FOC for 640x480 Dual Color LCD panel.
; Sets up registers for hirez 256 colour mode (mode "101h") without the BIOS.
;
; This program assumes:
;       - no external RAMDAC
;       - chip enable at port 3c3h.
;       - memory refresh 4ms
;

;*****************************************************************
;
; Instructions:
;
;  - Make your changes
;  - Assemble       --  masm mode101.asm;
;  - Link       --  link mode101;
;  - Make a .COM file  --  exe2bin mode101.exe mode101.com
;
;
```

```
;*********************************************************************

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

      ORG     100h
Mode101 PROC
      cld                     ; Set the direction flag.
      mov     ax, cs          ; Ensure that DS and ES point to this segment.
      mov     ds, ax          ;
      mov     es, ax          ;
      cli                     ; Interrupts off while changing the stack.
      mov     ss, ax          ;
      mov     sp, 0FFFFh      ;
      sti                     ; Stack change done - restore interupts.

      call    Step1           ; Enable SPC8106FOC chip.
      call    Step2           ; Unlock the AUX port.

      ; at this point, all the registers are accessible, we need to
      ; program Auxiliary 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, independent of CRT or panel type

      call    Step4           ; Setup the standard VGA registers.
      call    Step5           ; Setup the attribute registers.
      call    Step6           ; Blow the DAC with mode 13h colors.
      call    VertLines       ; just to have something on the screen

;---------------------------------------------------------------------
;
;  You should now be running as MODE 101 without BIOS support.
;
;---------------------------------------------------------------------
      mov     ax, 4C00h       ; DOS - Exit with Return Code.
      int     21h             ; This gives us a clean exit under
Mode12 ENDP                   ;  MS-DOS.
```

```
;*********************************************************************
;
;  STEP1 - Before ANY ports can be accessed, the SPC8106FOC chip must be
;        enabled.
;
;*********************************************************************

Step1:
      mov     dx,3C3h
      mov     al,1
      out     dx,al
      ret

;;; if chip enabled at 46E8, use the following code instead:
;;;
;;;
;;;Step1:
;;;   mov     si, OFFSET Enable46E8
;;;   call    RunTable
;;;   ret
;;;
;;;Enable46E8:
;;;   DW      PW+46E8h,16h
;;;   DW      PW+0102h,01h
;;;   DW      PW+46E8h,0Eh
;;;   DW      PW+4AE8h,00h
;;;   DW      -1

;*********************************************************************
;
;  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,00000001b  ;   enable CRTC-B set
              DB -1

              DW 03D4h
              DB  01h,80          ; CRTC_B[01h] = 80 (dec)
              DB  10h,0           ; CRTC_B[10h] = 0
              DB  11h,0           ; CRTC_B[11h] = 0
              DB  12h,240/2       ; CRTC_B[12h] = 8 msb of 240
              DB  15h,0           ; CRTC_B[15h] = 0 ( 2 non-display lines)
              DB -1

              DW 03DEh
              DB  00h,00000000b   ; IRQ disable,multifont disable,disable CRTC-B set
              DB  01H,01000010b  ; autocenter, slow cursor, dual panel
              DB  02H,01100010b  ; LOW state LCD,  color panel
              DB  03H,00000000b  ; Full power
              DB  04H,00000000b  ; clear scratch register
              DB  05H,00000110b  ; Hirez bit on, Extende display page
              DB  06H,00001000b  ; Enable page register at 3CDh
              DB  07H,00000000b  ; no stretch font,64 gs,dithering
              DB  09H,00000000b  ; No funny sprite stuff
              DB  0AH,00000000b  ; clear scratch register
              DB  0BH,00000001b  ; enable LCD
              DB  0DH,00100000b  ; defaults
              DB  1AH,00000000b  ; Clear scratch pad
              DB  1BH,00000000b  ; Clear scratch pad
              DB  1CH,00000000b  ; Clear relative start address high
              DB  1DH,00000000b  ; Clear relative start address low
              DB  -1
```

```
              DW  03CEh
              DB  06h,00000100b  ; set A000 addressing
              DB  -1
              DW  -1


;*****************************************************************
;
;  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,0Eh
      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,0A3h
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, 06h, 07h
      db      08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh
      db      41h, 00h, 0Fh, 00h, 00h
```

```
;*********************************************************************
;
;    STEP6 - program the DAC.
;
;    Note the DAC values are those for the IBM mode 13h, but this is
;    quite arbitrary.
;
;*********************************************************************

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

NewDACDB      00h, 00h, 00h, 00h, 00h, 2ah, 00h, 2ah, 00h, 00h, 2ah, 2ah
      DB      2ah, 00h, 00h, 2ah, 00h, 2ah, 2ah, 15h, 00h, 2ah, 2ah, 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, 05h, 05h, 05h, 08h, 08h, 08h, 0bh, 0bh, 0bh
      DB      0eh, 0eh, 0eh, 11h, 11h, 11h, 14h, 14h, 14h, 18h, 18h, 18h
      DB      1ch, 1ch, 1ch, 20h, 20h, 20h, 24h, 24h, 24h, 28h, 28h, 28h
      DB      2dh, 2dh, 2dh, 32h, 32h, 32h, 38h, 38h, 38h, 3fh, 3fh, 3fh
      DB      00h, 00h, 3fh, 10h, 00h, 3fh, 1fh, 00h, 3fh, 2fh, 00h, 3fh
      DB      3fh, 00h, 3fh, 3fh, 00h, 2fh, 3fh, 00h, 1fh, 3fh, 00h, 10h
      DB      3fh, 00h, 00h, 3fh, 10h, 00h, 3fh, 1fh, 00h, 3fh, 2fh, 00h
      DB      3fh, 3fh, 00h, 2fh, 3fh, 00h, 1fh, 3fh, 00h, 10h, 3fh, 00h
      DB      00h, 3fh, 00h, 00h, 3fh, 10h, 00h, 3fh, 1fh, 00h, 3fh, 2fh
      DB      00h, 3fh, 3fh, 00h, 2fh, 3fh, 00h, 1fh, 3fh, 00h, 10h, 3fh
      DB      1fh, 1fh, 3fh, 27h, 1fh, 3fh, 2fh, 1fh, 3fh, 37h, 1fh, 3fh
      DB      3fh, 1fh, 3fh, 3fh, 1fh, 37h, 3fh, 1fh, 2fh, 3fh, 1fh, 27h
      DB      3fh, 1fh, 1fh, 3fh, 27h, 1fh, 3fh, 2fh, 1fh, 3fh, 37h, 1fh
      DB      3fh, 3fh, 1fh, 37h, 3fh, 1fh, 2fh, 3fh, 1fh, 27h, 3fh, 1fh
      DB      1fh, 3fh, 1fh, 1fh, 3fh, 27h, 1fh, 3fh, 2fh, 1fh, 3fh, 37h
      DB      1fh, 3fh, 3fh, 1fh, 37h, 3fh, 1fh, 2fh, 3fh, 1fh, 27h, 3fh
      DB      2dh, 2dh, 3fh, 31h, 2dh, 3fh, 36h, 2dh, 3fh, 3ah, 2dh, 3fh
      DB      3fh, 2dh, 3fh, 3fh, 2dh, 3ah, 3fh, 2dh, 36h, 3fh, 2dh, 31h
```

```
DB      3fh, 2dh, 2dh, 3fh, 31h, 2dh, 3fh, 36h, 2dh, 3fh, 3ah, 2dh
DB      3fh, 3fh, 2dh, 3ah, 3fh, 2dh, 36h, 3fh, 2dh, 31h, 3fh, 2dh
DB      2dh, 3fh, 2dh, 2dh, 3fh, 31h, 2dh, 3fh, 36h, 2dh, 3fh, 3ah
DB      2dh, 3fh, 3fh, 2dh, 3ah, 3fh, 2dh, 36h, 3fh, 2dh, 31h, 3fh
DB      00h, 00h, 1ch, 07h, 00h, 1ch, 0eh, 00h, 1ch, 15h, 00h, 1ch
DB      1ch, 00h, 1ch, 1ch, 00h, 15h, 1ch, 00h, 0eh, 1ch, 00h, 07h
DB      1ch, 00h, 00h, 1ch, 07h, 00h, 1ch, 0eh, 00h, 1ch, 15h, 00h
DB      1ch, 1ch, 00h, 15h, 1ch, 00h, 0eh, 1ch, 00h, 07h, 1ch, 00h
DB      00h, 1ch, 00h, 00h, 1ch, 07h, 00h, 1ch, 0eh, 00h, 1ch, 15h
DB      00h, 1ch, 1ch, 00h, 15h, 1ch, 00h, 0eh, 1ch, 00h, 07h, 1ch
DB      0eh, 0eh, 1ch, 11h, 0eh, 1ch, 15h, 0eh, 1ch, 18h, 0eh, 1ch
DB      1ch, 0eh, 1ch, 1ch, 0eh, 18h, 1ch, 0eh, 15h, 1ch, 0eh, 11h
DB      1ch, 0eh, 0eh, 1ch, 11h, 0eh, 1ch, 15h, 0eh, 1ch, 18h, 0eh
DB      1ch, 1ch, 0eh, 18h, 1ch, 0eh, 15h, 1ch, 0eh, 11h, 1ch, 0eh
DB      0eh, 1ch, 0eh, 0eh, 1ch, 11h, 0eh, 1ch, 15h, 0eh, 1ch, 18h
DB      0eh, 1ch, 1ch, 0eh, 18h, 1ch, 0eh, 15h, 1ch, 0eh, 11h, 1ch
DB      14h, 14h, 1ch, 16h, 14h, 1ch, 18h, 14h, 1ch, 1ah, 14h, 1ch
DB      1ch, 14h, 1ch, 1ch, 14h, 1ah, 1ch, 14h, 18h, 1ch, 14h, 16h
DB      1ch, 14h, 14h, 1ch, 16h, 14h, 1ch, 18h, 14h, 1ch, 1ah, 14h
DB      1ch, 1ch, 14h, 1ah, 1ch, 14h, 18h, 1ch, 14h, 16h, 1ch, 14h
DB      14h, 1ch, 14h, 14h, 1ch, 16h, 14h, 1ch, 18h, 14h, 1ch, 1ah
DB      14h, 1ch, 1ch, 14h, 1ah, 1ch, 14h, 18h, 1ch, 14h, 16h, 1ch
DB      00h, 00h, 10h, 04h, 00h, 10h, 08h, 00h, 10h, 0ch, 00h, 10h
DB      10h, 00h, 10h, 10h, 00h, 0ch, 10h, 00h, 08h, 10h, 00h, 04h
DB      10h, 00h, 00h, 10h, 04h, 00h, 10h, 08h, 00h, 10h, 0ch, 00h
DB      10h, 10h, 00h, 0ch, 10h, 00h, 08h, 10h, 00h, 04h, 10h, 00h
DB      00h, 10h, 00h, 00h, 10h, 04h, 00h, 10h, 08h, 00h, 10h, 0ch
DB      00h, 10h, 10h, 00h, 0ch, 10h, 00h, 08h, 10h, 00h, 04h, 10h
DB      08h, 08h, 10h, 0ah, 08h, 10h, 0ch, 08h, 10h, 0eh, 08h, 10h
DB      10h, 08h, 10h, 10h, 08h, 0eh, 10h, 08h, 0ch, 10h, 08h, 0ah
DB      10h, 08h, 08h, 10h, 0ah, 08h, 10h, 0ch, 08h, 10h, 0eh, 08h
DB      10h, 10h, 08h, 0eh, 10h, 08h, 0ch, 10h, 08h, 0ah, 10h, 08h
DB      08h, 10h, 08h, 08h, 10h, 0ah, 08h, 10h, 0ch, 08h, 10h, 0eh
DB      08h, 10h, 10h, 08h, 0eh, 10h, 08h, 0ch, 10h, 08h, 0ah, 10h
DB      0bh, 0bh, 10h, 0ch, 0bh, 10h, 0dh, 0bh, 10h, 0fh, 0bh, 10h
DB      10h, 0bh, 10h, 10h, 0bh, 0fh, 10h, 0bh, 0dh, 10h, 0bh, 0ch
DB      10h, 0bh, 0bh, 10h, 0ch, 0bh, 10h, 0dh, 0bh, 10h, 0fh, 0bh
DB      10h, 10h, 0bh, 0fh, 10h, 0bh, 0dh, 10h, 0bh, 0ch, 10h, 0bh
DB      0bh, 10h, 0bh, 0bh, 10h, 0ch, 0bh, 10h, 0dh, 0bh, 10h, 0fh
DB      0bh, 10h, 10h, 0bh, 0fh, 10h, 0bh, 0dh, 10h, 0bh, 0ch, 10h
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
```

```
;*********************************************************************
;
;    VertLines : display multicoloured vertical lines.
;
;
;*********************************************************************

VertLines:
        mov    ax,0A000h          ;
        mov    es,ax              ;
        mov     dx,3cdh         ; dx = page register
        xor     bx,bx           ; bx = column counter

NextVline:
        xor     al,al
        out     dx,al           ; set R+W pages 0
        mov     di,bx
        mov     cx,480          ; 480 lines
NextPixel:
        mov     es:[di],bl
        add     di,640
        jnc     short @f
        in      al,dx           ; get current write page
        add     al,00010001b    ; increment both pages
        out     dx,al           ; set the next page
@@:
        loop    NextPixel
        inc     bx
        cmp     bx,640
        jb      short NextVline
        xor     al,al
        out     dx,al           ; set pages 0 again
        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      Mode101
```

4.0 MANUAL SETTING OF POWER STATES ON THE SPC8106

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 “merge” tables. The format of the table is port (only the low byte is required, as the upper is implicitly 3), register index, AND mask, OR mask. If the “port” value is 0xFF a time delay is inserted, if the “port” value is 0xFE, we wait for one vertical retrace.
Also note 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.

```
aStateLCD_0    DB     0DEh,03h,11000111b,00000000b ; Addr Decode,Osc. enable
               DB     0FFh                         ; Delay
               DB     0FEh                         ; Wait for Vsync
               DB     0DEh,03h,11111000b,00000000b ; p.mode 0
               DB     0

aStateLCD_1    DB     0DEh,03h,11000111b,00000000b ; Addr Decode,Osc. enable
               DB     0FFh                         ; Delay
               DB     0DEh,03h,11111000b,00000001b ; p.mode 1
               DB     0

aStateLCD_2    DB     0DEh,03h,11000111b,00000000b ; Addr Decode,Osc. enable
               DB     0FFh                         ; Delay
               DB     0DEh,03h,11111000b,00000010b ; p.mode 2
               DB     0

aStateLCD_3    DB     0DEh,03h,11000111b,00000000b ; Osc. enable
               DB     0FFh                         ; Delay
               DB     0DEh,03h,11111000b,00000011b ; p.mode 3
               DB     0FEh                         ; Wait for Vsync
               DB     0FEh                         ; Wait for Vsync
               DB     0FEh                         ; Wait for Vsync
               DB     0FEh                         ; Wait for Vsync
               DB     0DEh,03h,11111111b,00011000b ; No decode,Osc. disable
               DB     0

aStateLCD_4    DB     0DEh,03h,11001111b,00000000b ; Osc. ena, normal clock
               DB     0FFh                         ; Delay
               DB     0DEh,03h,11110000b,00001100b ; No decode,p.mode 4
               DB     0
```

THIS PAGE INTENTIONALLY BLANK

```
aStateLCD_4_SR DB     0DEh,03h,11000111b,00000000b  ; Osc. ena,decode ena,normal clck
               DB     0FFh                          ; Delay
               DB     0DEh,03h,11110000b,00000100b ; p.mode 4
               DB     0FEh                          ; Wait for Vsync
               DB     0FEh                          ; Wait for Vsync
               DB     0FEh                          ; Wait for Vsync
               DB     0FEh                          ; Wait for Vsync
               DB     0DEh,03h,11111111b,00011000b ; Osc. disable, no decode
               DB     0


aStateLCD_5    DB     0DEh,03h,11100111b,00000000b ; Addr Decode,Osc. enable
               DB     0FFh                          ; Delay
               DB     0DEh,03h,11111000b,00100101b ; p.mode 5, slow clock
               DB     0


;-------------------------------------------------------------------------
;     MergeRegister
;
;     Input:   CS:SI  -> Table to load
;     Output:  n/a
;     Uses:    AX,SI
;-------------------------------------------------------------------------

      PUBLIC  MergeRegister
MergeRegister  PROC   NEAR
      PUSH    CX
      PUSH    DX
      MOV     DH,03h           ; DX=3??
MergeNext:
      LODSB   CS:[SI] ; AL=Register Select (low byte only)
      TEST    AL,AL            ; end of table ?
      JZ      MergeExit

      CMP     AL,-1            ; Delay?
      JE      MergeDelay       ; yes then do it

      CMP     AL,-2            ; Wait for Sync?
      JE      MergeSync        ; yes then do it

      MOV     DL,AL            ; DX=Register
      LODSB   CS:[SI]          ; AL=Register Index
      CMP     DL,3DEh-300h     ; Aux Register?
      JNE     MergeHW          ; yes then filter it
```

```
MergeMisc:
      MOV     AH,AL           ; save index in AH
      CALL    ReadAuxReg
      AND     AL,CS:[SI+0]    ; Turn off unrequired bits
      OR      AL,CS:[SI+1]    ; Turn on required bits
      CALL    WriteAuxReg     ; AH=index, AL = DATA

      JMP     SHORT NextRegister

MergeDelay:
      MOV     CX,3000h        ; recommended delay loop
@@:
      JMP     $+2
      LOOP    @B
      JMP     SHORT MergeNext

MergeSync:
      MOV     DX,3CCH
      IN      AL,DX
      MOV     DX,3BAH         ; assume 3BA
      TEST    AL,1            ; correct?
      JZ      @F              ; yes, jump
      MOV     DX,3DAH         ; no, dx = 3DAH
@@:
      IN      AL,DX           ; AL=Input Status 1 Data
      TEST    AL,00001000b    ; Vertical Retrace?
      JZ      @B              ; no then wait til start
@@:
      IN      AL,DX           ; AL=Input Status 1 Data
      TEST    AL,00001000b    ; Vertical Retrace?
      JNZ     @B              ; yes then wait til end

      JMP     SHORT MergeNext

MergeHW:
      CMP     DL,3CCh-300h    ; MOR?
      JNE     @F              ; no then bypass

      IN      AL,DX           ; AL=Register Value
      MOV     DL,3C2h-300h    ; Select 3C2 as write
      JMP     SHORT MergeMask
@@:
      CMP     AL,0FFH         ; valid index?
      JE      MergeRead       ; no, jump

      OUT     DX,AL           ; Update Index
      INC     DX              ; Select Register Data
```

```
MergeRead:
      IN       AL,DX            ; AL=Register Value

MergeMask:
      AND      AL,CS:[SI+0]     ; Turn off unrequired bits
      OR       AL,CS:[SI+1]     ; Turn on required bits
      OUT      DX,AL            ; Update Data

NextRegister:
      ADD      SI,2             ; DS:SI->Next Register
      JMP      MergeNext        ; Repeat for length of table

MergeExit:
      POP      DX
      POP      CX
      RET

MergeRegister  ENDP
      PAGE
```

5.0 IDENTIFYING THE SPC8106

The purpose of the code examples below are to demonstrate the methods to identifying the SPC8106 graphics controller. We have two recommended methods:

1. Identifying the SPC8106 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 == 0xec )
         printf ( "SPC8106 detected.\n" );
      else
         printf ( "SPC8106 not found.\n" );
     }
//---------------------------------------------------------------------------
```

2. By reading registers within the SPC8106 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 == 0x60 )
         printf ( "SPC8106 detected.\n" )    ;
      else
         printf ( "SPC8106 not detected.\n" );
    }
 else
    printf ( "SPC8106 not detected.\n" )     ;

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

6.0 ENABLING AND DISABLING THE SPC8106

The information on Enabling the SPC8106 is contained in the examples located earlier in the documents “Initializing the SPC8106 in Mode 03h without a BIOS”; “Initializing the SPC8106 in Mode 12h without a BIOS”; and “Initializing the SPC8106 in Mode 101h without a BIOS”. In Step 1 of each of these examples, it demonstrates the way to use port 3C3h to enable the chip. In the commented code below Step 1, it demonstrates the method of enabling the chip using the alternate 46E8h port. The Enable port is determined at power up by the state of an MD line on the evaluation board. For more information on the Enable port selection, please refer to the SPC8106 Hardware Functional Specification.

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

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

The code required to disable the chip when using the enable port 46E8h is:

```
mov     dx, 46E8h
mov     al, 06h
out     dx, al
ret
```

7.0 ADJUSTING THE SPC8106 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 SPC8106. 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).
The relation between frame rate (in Hz) and panel geometry is as follows:
FR[Hz] = 3540000.0/(Vertical*Horizontal);
where “Vertical” is the sum of “vertical displayed” and “vertical non-display period” (in lines) and “horizontal” is the sum of panel “horizontal displayed” and “non-display period” (in character clocks). For all LCD panels the “horizontal non-display period” is fixed at 14 (decimal) characters. The values for “Vertical” and “Horizontal” can be determined via the SPC8106FOC CRTC B set registers.
Constants:
“Horizontal” = CRTCB[01h]+14
“Vertical non-display period” = CRTCB[15h]+2
Single Panels:
“Vertical displayed size” = CRTCB[12h]*2
therefore the frame rate = 3540000.0/((CRTCB[12h]*2+CRTCB[15h]+2)*(CRTCB[01h]+14))
Dual Panels:
“Vertical displayed size” = CRTC_B[12h]*4
therefore the frame rate = 3540000.0/((CRTCB[12h]*4+CRTCB[15h]+2)*(CRTCB[01h]+14))
TFT panels (using LCD like timings)**:**
“Vertical displayed size” is obtained via CRTCB[11h] bits 7 and 6 and CRTCB[12h].

1. This formula assumes a 28 MHz clock (3.54MHz = 28.322MHz/8).
2. When modifying the frame rate, it is important that the horizontal displayed period is greater or equal to the horizontal panel size.

8.0 PROGRAMMING THE SPRITE HARDWARE

In video technology the term sprite usually refers to a bitmap shape that can be moved across complex backgrounds without flicker or damage to the background image. The SPC8106 supports a hardware assisted sprite capable of displaying a 64 x 64 pixel 4-color image, regardless of the video mode. Typical uses for sprites are for hardware cursors in GUI environments, or pop-ups for system status.

The intent of this document is to demonstrate the rudiments required to display and manipulate a sprite image with the SPC8106. It is assumed the reader has an understanding of the VGA register architecture. The code samples included are C-like and are intended to show concepts, and have not been compiled. Most sprite control is performed through two index/data register pairs and the sprite image is stored in unused video memory.

The first set of sprite control registers are located in the CRTC (port 3D4h/3D5h) at indexes 30h through 38h. These indexes are beyond those defined by IBM for the VGA and control functions such as sprite image address, screen position, and clipping.

The second set of sprite control registers are located in the Auxiliary Register (port 3DEh/3DFh) at indexes 05h, 09h and 0Bh. These registers control sprite enable, memory mapping and sprite color control.

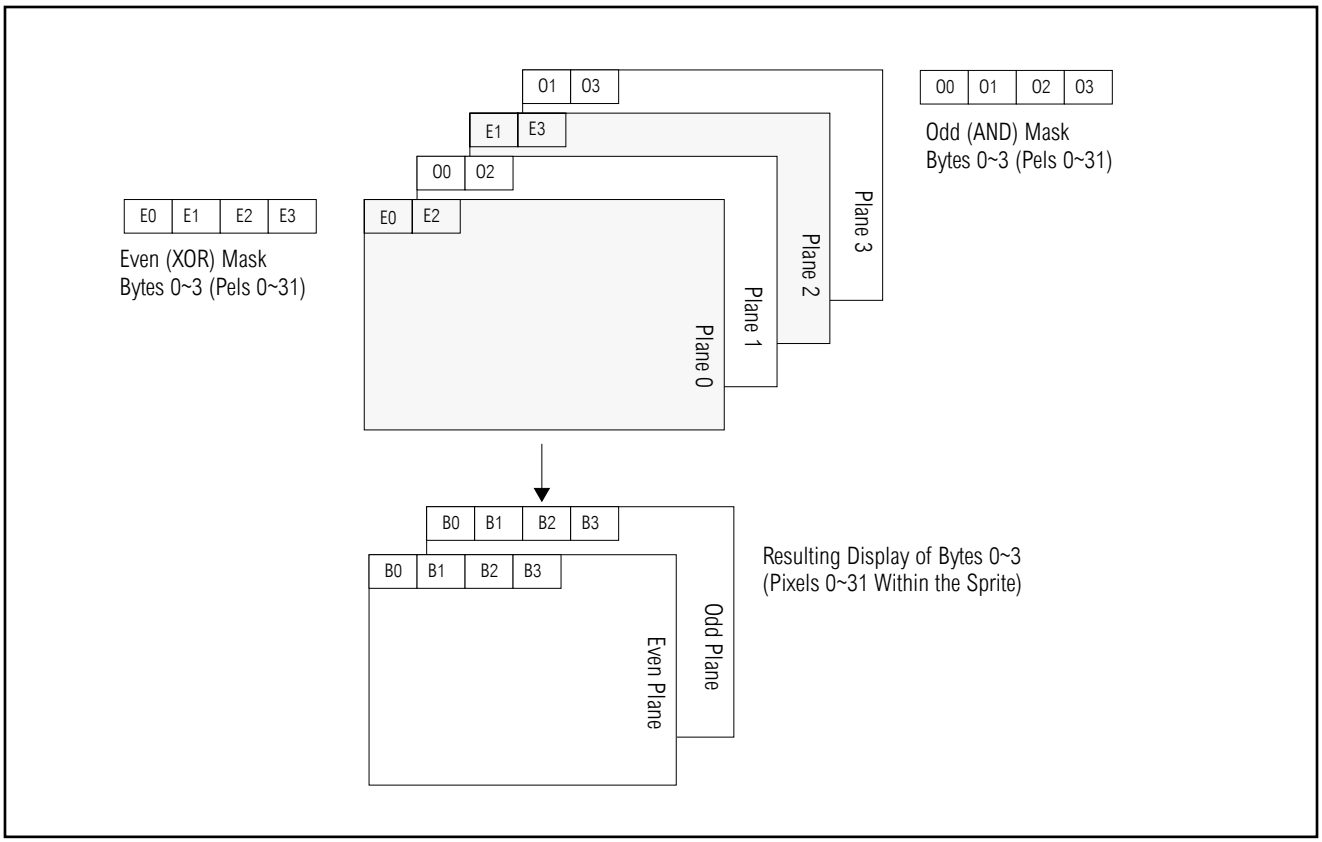
The sprite image should be placed in normally unused video memory. Each sprite image requires 1KB of memory (the SPC8106 supports 512KB) so there are a possible 512 possible sprite 'slots' in display memory. To ensure the sprite image is in unused video display memory it should be placed in the upper 256KB. Placing the image in the lower 256KB of video memory could permit a standard VGA memory access to overwrite the image.

The programming examples contained within this document utilize a memory technique which swaps the upper and lower 256KB of memory. The examples also use a feature called Sprite Write Mode. Upper page swapping permits the upper 256KB of video memory to be addressed through the A000h segment, while swapping the normal video memory to the B000h segment. Upper page swapping is proprietary to the SPC8106 and does not affect the display contents (to avoid memory manager conflicts, do not update the video memory in the B000h segment). This upper page swapping technique is consistent with the memory access methods required by previous Seiko Epson VGA LCD controllers which incorporate a hardware sprite.

Sprite Write Mode is a consistent programming interface for placing sprite image data into display memory, this allows programming of the sprite in one consistent format no matter what current memory model that the controller is in. This mode eliminates the complexity of setting up the memory as depicted in the following diagram by allowing the sprite image to be envisioned as two streams of pixel data, one stream representing the most significant bits of each pixel and the other representing the least significant bits.

> The sprite hardware does not support a packed pixel format. Prior to loading a sprite image, the image data must be broken down into two arrays of pixels. This format is typical of the XOR and AND arrays provided for the cursor image in such popular GUIs as Windows and OS/2.

The following diagram depicts the VGA memory in a planar format and the sprite as displayed on the screen. The two data structures represent the strings of pixel information where each bit of Bytes 0 through 3 represent one pixel each (bits 0 through 31). The high order bits of the image (cursor XOR mask) are placed in the even planes (planes 0 and 2) and the low order bits (AND mask) are place in the odd planes (planes 1 and 3).



Each VGA memory address spans four planes of data (to give 32 bits). Sprite images consist of only two bits per pixel. Each VGA memory address therefore contains sixteen pixels of sprite image. One complete row of a sprite image (64 pels) is held in four sequential VGA memory addresses. 256 addresses (x 4 planes = 1KB) are required to hold a complete sprite image.

For cursor images (typically 32 x 32 pels) only 1/4 of the entire sprite area is required. In this situation the remaining 3/4 of the sprite should be set to transparent (AND pels = 1 and XOR pels = 0). When less than entire 64 rows are required as in the case of a 32x32 cursor, it is more CPU efficient to place the image in the lower portion of the sprite memory, because CPU fetches to video memory are reduced. The recommended location for a 32x32 cursor data is starting at the 256th byte when using Sprite Write Mode. The Y Start Row Register CRTC [35h] should then be set to clip the upper portion of the cursor data (i.e., the undisplayed portion). Unfortunately, for each row or partial row displayed, the full width of the row (all 64 pels) is read into an internal buffer for display, and CPU efficiency cannot be improved.

The following example demonstrates how to load a sprite image (64 x 64) into slot 256 (offset 0 in the upper 256K).

```
/* Defines */
/*---------*/
#define AUX 0x3DE
#define CRTC 0x3D4

char far* pDisplayMem, pImgData;
```

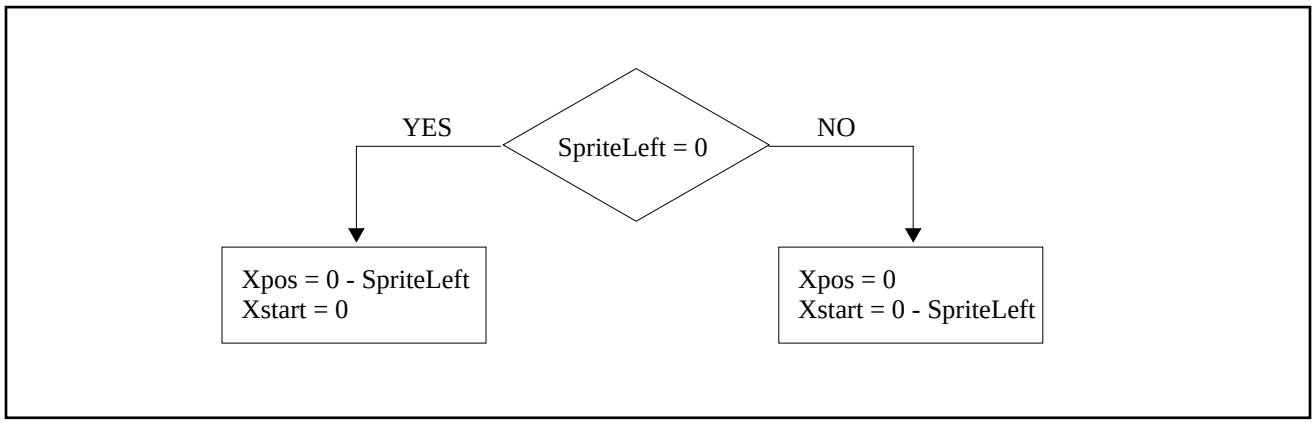
```
/* Enable the Sprite Hardware Functions */
/* ---------------------------------- */
// Enable the sprite (AUX[5] bit 5) and disable the upper 256k (AUX[5] bit 0).
outp (AUX, 5);                    // Select index 5.
outp (AUX + 1, 0x20);             //

// In AUX register 9 select sprite page 0 (b3), logical plane 0 (b2)
// upper page swap (b1) and Sprite Write Mode (b0)
outp (AUX, 9);                   // Select index 9.
outp (AUX + 1, 3);                // 0011b

// The SPC8108 sprite write mode did not disable the sequencer chain4
// bit and required the following code to ensure correct operation in all
// video modes. The SPC8106 sprite write mode correctly disables chain4 and
// the following code may be omitted in cases where this code will be run on
// an SPC8106 only. Note that including this code ensures the correct handling
// of the sequencer for both video controllers. Its use is transparent on
// the 8106 and it imposes very little execution overhead.

// Temporarily disable chain4 addressing (should we be in mode 13h)
nSeqIdx = inp(SEQ);              // Read and save current index
outp(SEQ, 4);                    // Set index 4
nSeqDat = inp(SEQ + 1);          // Read and save the data.
nTmp = inp(SEQ + 1);             // Read the current data.
nTmp &= 2;                       // Mask off the chain 4 bit
outp(SEQ + 1, nTmp);             // Set the new data

/* Load the Image into Display Memory */
/* ---------------------------------- */
// Set the pointers to both display memory and the even pixel image array.
// (512 bytes of Least Significant Bits of color data)
// Loop 512 times moving the pixel image array to display memory. Sprite
// Write Mode handles switching of planes placing even addressed data into
// plane 0 and odd addresses into plane 2.
pImgData = EvenArray;             // (XOR array if a cursor)
pDisplayMem = A000H:0000          //
for (x = 0; x < 512; x++)
  *pDisplayMem++ = *pImgData++;

// Select sprite logical plane 1 (AUX[9] b2).
// This step requests that even addressed data be placed into plane one
// and odd address be placed into plane three.
outp (AUX, 9);                    // Select index 9.
outp (AUX + 1, 7);                // In this example I know the state of
                                  // AUX[9] so I will forego a read/mask/write
                                  // and just write blind turning b2 on.
```

```
// Set the pointer to the start of the odd pixel array
// (512 bytes of Most Significant Bits of color data) and reset the
// display memory pointer. Loop 512 times moving image bytes from the
// image array to display memory.
pImgData = OddArray;              // (AND mask for a cursor)
pDisplayMem = A000H:0000;
for (x = 0; x < 512; x++)
   *pDisplayMem++ = *pImgData++;

/* Restore the upper page swap bit (AUX[9] b0). */
// This step MUST be performed. If not done display data meant for A000h
// segment will be actually written to the B000h segment.
outp (AUX, 9);                    // Select index 9.
outp (AUX + 1, 0);                // May as well turn them all off.
                                  // We don't need them any more.

/* Restore the sequencer index and data. */
// This step only required if the previous step for SPC8108
// use of this code in mode 13h was performed.
outp (SEQ, 4);                    // Set index 4
outp (SEQ + 1, nSeqDat);          // Restore the index 4 data.
outp (SEQ, nSeqIdx);              // Restore the sequencer index.

/* Set the Sprite Position Registers */
/* ------------------------------- */
// This informs the chip where to find the image in display memory.
// The example image is located in the first 256 bytes in the upper 256k.
// To locate this image set CRTC[36] b0 to 1 and CRTC[37]to 0.
outp (CRTC, 0x36);                // Other image 'slots' will
outp (CRTC + 1, 1);               // require the calculation of
outp (CRTC, 0x37);                // the offset in which to find
outp (CRTC, 0);                   // the image.

// For sprites ensure that CRTC[38] b1 is 0 in order to enable the
// four palette colors. For cursors CRTC[38] b1 is set to 1 to enable
// the logical screen/!screen palette. The hardware sprite also supports
// horizontal and vertical doubling based on bits 2 & 3 of this register.
outp (CRTC, 38H);                 // Select CRTC 38H
outp (CRTC +1, 0);                //
```

```
/* Set the image colors. */
// Note that for cursor images only indexes 3CH and 3DH require
// programming as the other two colors are displayed as transparent
// and inverted.
outp (AUX, 0BH);                // Set AUX index to 0Bh
nAux = inp (AUX + 1);           // Read and save the current index
nTmp = inp (AUX + 1);           //
// Note: The mask value of 8 (below) enables both the internal (LCD)
// and external (CRT) palettes. If it is desired to write different
// values to these palettes then this section of code will have to
// be executed twice with different palette write disable bits set.
nTmp |= 8;                      // Mask in the overlay enable.
outp (AUX +1, nTmp);

// SetDAC is a routine to program one RAMDAC entry. This type of code is
// a standard VGA routine and is not included in this example. The data type
// for color1, color2 ... would be long and these variables would be composed
// of RGB values.
SetDAC (3CH, color1);           // Set the sprite colors.
SetDAC (3DH, color2);           //
SetDAC (3EH, color3);           //
SetDAC (3FH, color4);           //

out (AUX + 1, nAux);            // Restore the AUX register data

// The following piece of code pertains to the SPC8108.
// The SPC8106 sprite colors are set by the user defined function
// SetDAC. As with the chain4 override the following is required
// if the program will be used with an SPC8108. The code executes
// transparently on an SPC8106 with very little execution overhead.

// Set the internal LUT colors. If this is a cursor image then only
// registers 3CH and 3DH need to be set.
outp (CRTC, 3CH);                // Sprite palette 0
outp (CRTC +1, color1);          // Cursor background
outp (CRTC, 3DH);                // Sprite palette 1/
outp (CRTC +1, color2);          // Cursor foreground
outp (CRTC, 3EH);                // Sprite palette 2
outp (CRTC +1, color3);          //
outp (CRTC, 3FH);                // Sprite palette 3
outp (CRTC +1, color4);         //

/* Sprite initialization complete. */
// The sprite/cursor image may be removed from the display
// by setting AUX[5] b5 to 0.
```

8.1 Setting the Displayed Sprite Position

The displayed position of the sprite and any required clipping is done with CRTC registers 30h through 35h. Clipping of the sprite image on the right and bottom of the display is done automatically by the hardware. If the sprite image needs to be clipped on the top or left of the display the clipped amount needs to be calculated and set into the appropriate registers.
All position calculations are based on absolute pixel positions. This holds even for text modes, which means the sprite image location is not limited to character cell boundaries, but may be placed on any pixel on the display. Sprite positioning information is always relative to the upper left corner of the sprite image and to the upper left corner of the display. The following example demonstrates how to program a cursor image located in the lower left qaudrant of the sprite as this covers all positioning considerations.
In general the algorithm for cursor position and clipping is:



The following example is typical of the logic to program a cursor position with coordinates passed by Microsoft Windows.

```
/* Calculate the Xposition and Xstart. */

CursorX = CursorPos - HotSpot // nCursorX = raw X position
                              // (SpriteLeft)
if (CursorX > 0) {
         Xpos = CursorX;
         Xstart = 0;
         }
else {
         Xpos = 0;
         Xstart = 0 - CursorX
         }

/* Now program the X position registers. */
// First the X start register is set. This value informs the
// hardware how many pixels to count across into the image
// before starting display.

outp (CRTC, 34H);
```

```
outp (CRTC +1, nXstart);

// Program the X position. As the X position may be > 255
// two data register writes are required.
// IMPORTANT - The position registers do not latch until the
// high position is written.

outp (CRTC, 31H);
outp (CRTC +1, LOBYTE(Xpos));
outp (CRTC, 30H);
outp (CRTC +1, HIBYTE(Xpos));

/* Calculate the Yposition and Ystart. */
// As this is a cursor programming sample we will assume the
// cursor has been loaded into the bottom left quadrant of the
// sprite image for performance considerations.
// The logic for setting Y position under these circumstances
// is much the same however the Y start will always be at
// least 32 pixels into the image area.

nCursorY = nCursorPos - nHotSpot  // nCursorY = raw Y position
                                  // (SpriteTop)
if (nCursorY > 0) {
         nYpos = nCursorY;
         nYstart = 32;
         }
else {
         nYpos = 0;
         nYstart = 0 - nCursorY + 32;
         }

/* Now program the Y position registers. */
// First the Y start register is set. This value informs the
// hardware how many pixels to count down into the image before starting display.

outp (CRTC, 35H);
outp (CRTC +1, nYstart);

// Program the Y position. As the Y position may be > 255
// two data register writes are required.
// IMPORTANT - The position registers do not latch until the
// high position is written.
```

```
outp (CRTC, 34H);
outp (CRTC +1, LOBYTE(Ypos));
outp (CRTC, 33H);
outp (CRTC +1, HIBYTE(Ypos));

// End of programming sprite/cursor position. //
```

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

SETFONT.EXE Display Utility

Drawing Office No. X12-UI-001-02

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.

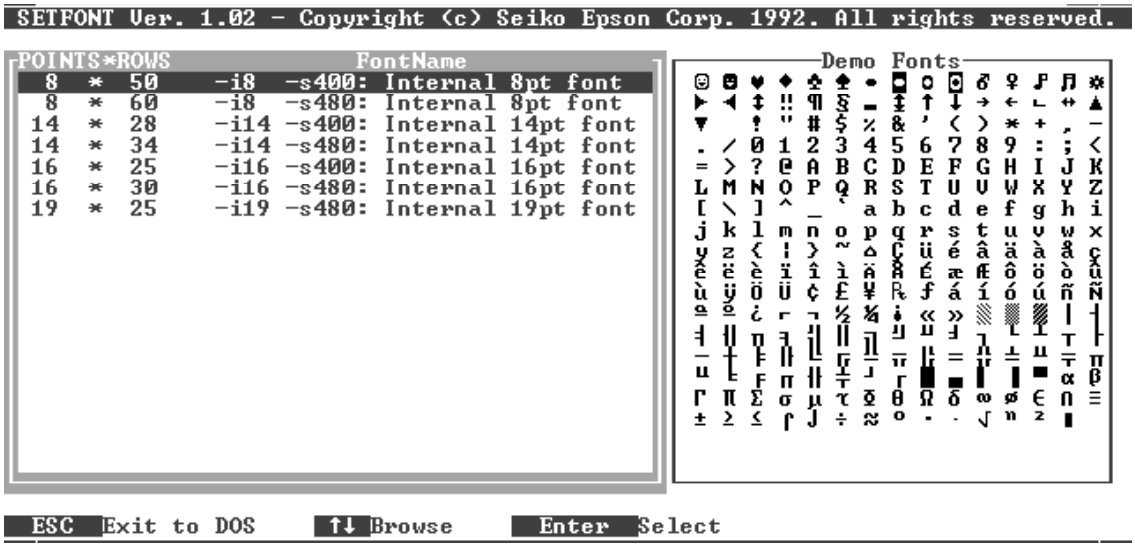


Figure 1 : SETFONT Menu Interface

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

SPC8106 LCD/CRT VGA CONTROLLER

BOLD.EXE Display Utility

Drawing Office No. X12-UI-002-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

BOLD.EXE provides more readable text on LCD or CRT 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 or CRT
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

SPC8106 LCD/CRT VGA CONTROLLER

REVERSE.COM Display Utility

Drawing Office No. X12-UI-003-02

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.

REVERSE.COM makes BIOS calls which modify the LCD registers within the video controller which only effects the LCD outputs. If run on a CRT the change will not occur until a switch back to the LCD display.

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

SPC8106 LCD/CRT VGA CONTROLLER

PWRSAVE.EXE Power Save Utility

Drawing Office No. X12-UI-004-02

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 or CRT
BIOS	:	Seiko Epson VGA BIOS
DOS Program	:	Yes
DOS Version	:	3.2 or greater
Windows Program	:	see Note below
Windows DOS Box	:	see Note below
Windows DOS Full Screen	:	Windows 3.1x only (with exceptions - see Comments below)
OS/2 DOS Full Screen	:	2.0 or greater

1. PWRSAVE [x], PWRSAVE /r, and PWRSAVE /off will work under Windows 95.
2. PWRSAVE /u and PWRSAVE /Tn x do not work under Windows 95.

There is an option under the Windows 95 desktop "Prevent DOS Application from Detecting Windows 95". By default this option is turned off. Users should check to ensure that this option is really turned off, otherwise you may have problems running PWRSAVE.

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 key-stroke 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.

`/r` remain in reduced gray scale mode

`/?` displays the help message.

n is a decimal number from 5 to 3600 and x is a decimal number from 1 to 4 for the SPC8106.
PWRSAVE 0 is equivalent to PWRSAVE /off.

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.

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

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

READCHIP.EXE Diagnostic Utility

Drawing Office No. X12-UI-005-02

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 or CRT
BIOS	**:**	Any manufacturer's IBM compatible 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 (see note below)
Windows DOS Full Screen	**:**	Yes, Windows 3.1x and Windows 95
OS/2 DOS Full Screen	**:**	Yes

1. Registers read while in a Windows DOS box may not contain the expected value. This is because the VDD has virtualized all the VGA ports, in other words in a Windows DOS box the text mode is emulated with graphics.
2. READCHIP uses "stdout" calls and may be redirected to a file or piped to a DOS filter such as MORE.COM.

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 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

SPC8106 LCD/CRT VGA CONTROLLER

WGS.EXE Windows Gray-Scale Utility

Drawing Office No. X12-UI-006-02

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 graphics controller's special features which offer 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

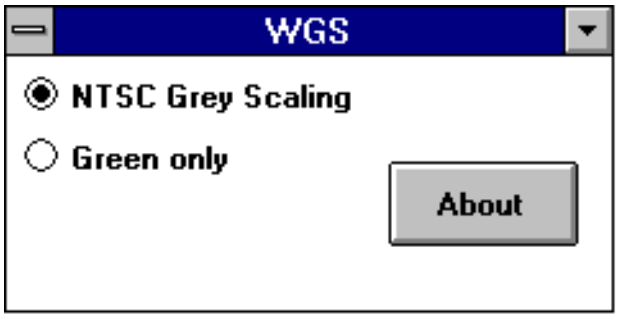


Figure 1 : The WGS Dialog Box

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.
- This program has no visible effect while running on a CRT, but will take effect when a switch to LCD occurs.
- changes take effect immediatly upon selecting a button.

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

SPC8106 LCD/CRT VGA CONTROLLER

SESHELL.EXE Shell Utility

Drawing Office No. X12-UI-007-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

SESHELL provides a menu interface to demonstrate the operation and command line interfaces for SPC8106 DOS utilities. Not all programs shipped with the SPC8106 will run under SESHELL, but the majority will.

1.1 Program Requirements

Video Controller	**:**	SPC8106
Display Type	**:**	LCD or CRT
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:

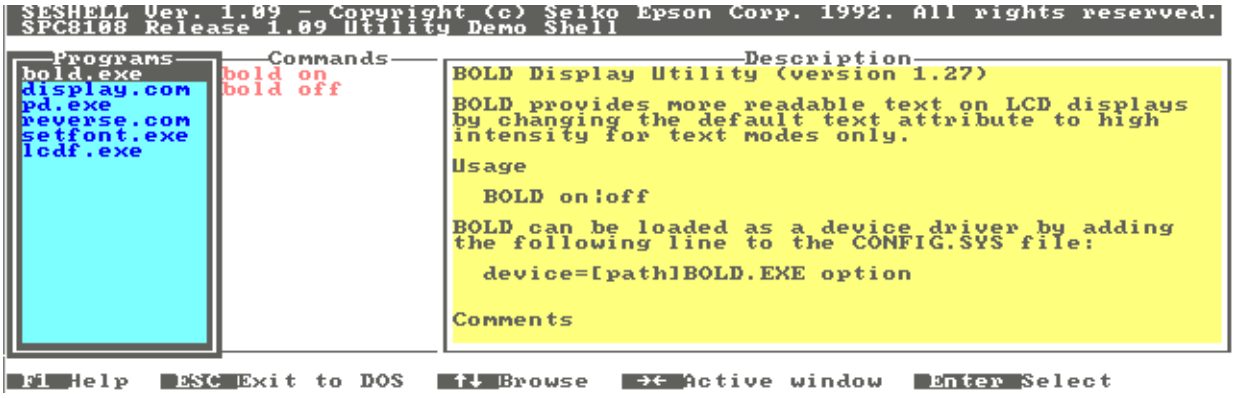


Figure 1 : SESHELL Screen

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

SPC8106 LCD/CRT VGA CONTROLLER

SHOWMODE.EXE Demonstration Program

Drawing Office No. X12-UI-008-02

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 or CRT
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

SPC8106 LCD/CRT VGA CONTROLLER

LCDF.EXE Display Utility

Drawing Office No. X12-UI-009-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

LCDF is a display utility for changing the LCD panel frame rate using the SPC8106 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 a variety of chip specific registers. 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	**:**	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 28.322 MHz clock for the SPC8106 and SPC8108. 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

SPC8106 LCD/CRT VGA CONTROLLER

VRTEXP.EXE Display Utility

Drawing Office No. X12-UI-010-02

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 (SOLLEX) Extentions 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

SPC8106 LCD/CRT VGA CONTROLLER

8106CFG.EXE BIOS Configuration Utility

Drawing Office No. X12-UI-011-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

8106CFG is the OEM utility for custom configuring the SPC8106 VGA Controller BIOS binary image. This utility allows the OEM, or product support people to easily customize the Seiko Epson SPC8106 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.

8106CFG consists of four files; 8106cfg.exe, 8106cfg.hlp, 8106cfg.hx, and 8106cfg.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 **8106cfg.exe**, **8106cfg.hlp**, **8106cfg.hx**, and **8106cfg.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

8106CFG is invoked from the DOS command line as follows:

```
8106cfg [/s|/x][/?]
```

where:

no parameters starts 8106CFG in the typical user mode.

`/s` starts 8106CFG in the sophisticated user mode.

`/x` starts 8106CFG in the expert user mode.

`/?` displays the copyright notice and version number.

1.4 Comments

The 8106CFG 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 compatibility (mode 7, mode 11)
Enable port for VGA - 3C3h or 46E8h
Compatable DAC programming - IBM ISA or IBM 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
Number of gray levels for mode 3 - 16, 32 or 64

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 may create a BIOS which 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:
    - 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.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

DISPLAY.COM Display Selection Utility

Drawing Office No. X12-UI-012-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

DISPLAY.COM is a DOS program which allows the selection of the active display device. When both a CRT and an LCD are attached, DISPLAY will allow switching to either or both displays.

1.1 Program Requirements

Video Controller	**:**	SPC8106, SPC8108, SPC8110
Display Type	**:**	LCD or CRT
BIOS	**:**	Seiko Epson Video 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	**:**	No

1.2 Installation

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

1.3 Usage

From the DOS command line type:

```
display [crt | lcd | doublescan]
```

To view the available options and other relevant program information, at the DOS prompt type:

```
display
```

DISPLAY will display a usage screen similar to the following:

```
DISPLAY.COM Version 1.00
Utility to select the active display.
Copyright (c) Seiko Epson Corp. 1994. All rights reserved.
         usage:   DISPLAY CRT turns on CRT only
                  DISPLAY LCD turns on LCD only
                  DISPLAY DOUBLESCAN turns on CRT and LCD in 480 line modes
                                     turns on CRT only in all other modes
```

1.4 Comments

- If the active display mode is currently at a higher resolution than the LCD can support, and a switch to simultaneous display (doublescan) or to the LCD occurs, DISPLAY will fail the request.

1.5 Program Messages

ERROR: Requires Seiko Epson Video BIOS Extensions.

The Seiko Epson Video BIOS Extensions are not installed as part of the video system.

ERROR: Requires analog monitor.

The video controller cannot detect an analog monitor attached to the system.

ERROR: Cannot switch at this time.

An attempt to switch to the LCD display occurred while in an operational display mode beyond the capabilities of the LCD display.

ERROR: Cannot switch while in a Windows DOS Box

An attempt to switch displays occurred while in a Windows emulated DOS box. Use <ALT>-<Enter> to switch to DOS full screen and try again.

S-MOS SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

SPRITEST.EXE Demonstration Program

Drawing Office No. X12-UI-013-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

SPRITEST demonstrates the sprite capabilities of the SPC8106 by moving a demonstration sprite around the screen. It loads as a Terminate and Stay Resident (TSR) program and will run in conjunction with most programs.

1.1 Program Requirements

Video Controller	**:**	SPC8106
Display Type	**:**	LCD or CRT
BIOS	**:**	Seiko Epson VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	2.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	No
Windows DOS Full Screen	**:**	No
OS/2	**:**	No

For CRT support, the RAMDAC must be a BIOS supported RAMDAC with internal Sprite/Hardware Cursor overlay.

1.2 Installation

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

1.3 Usage

SPRITEST is invoked with the following command structure.

```
spritest [/An] [/3] [/9] [/Sxy][/V] [/H] [/Q] [/U] [ImageFile]
```

Example: **spritest /A3 animate.002**
or
spritest

ImageFile

The ImageFile parameter specifies a Windows format .BMP file which will be displayed as the sprite image. SPRITEST will also display a series of images to create an animation effect. The file names must conform to the following convention:

ImageFil.000
ImageFil.001
. . .
ImageFil.nnn

You need only to supply the last filename in the sequence. SPRITEST will automatically load files .000 to .nnn.

Switches

All switches must be preceded by either "-" or "/".

Image Control Switches

/An Animate n frames. Files ImageFil.000 to ImageFil.n-1 are displayed in sequence to create an animation effect.

/3 The sprite is displayed in three color mode with the fourth color transparent. This is just one of the color schemes available to the sprites.

/9 The 8109 sprite load method is used. The sprite image is loaded into the display 'slot' by first setting the VGA registers to Planar Addressing Mode rather than using Sprite Write Mode.

/Sxy /Sxy can be specified as /Sx, /Sy, or /Sxy to stretch the sprite in the x, y, or both axes.

/V Vertical expansion disable. On LCD displays where vertical expansion is enabled, sprite movement may look jerky and uneven. Disabling vertical expansion corrects this problem.

Miscellaneous Switches

/H Display the help screen.

/Q Quiet mode causes SPRITEST to load without displaying either the copyright notice or errors. This option is useful for demonstrations where multiple images are loaded.

/U Uninstall SPRITEST. Attempts to remove the TSR portion of Spritest from memory.

1.4 Program Messages

`Installation successful.`

SPRITEST has installed as a TSR and successfully loaded the image file.

`ERROR: cannot install - Spritest is already installed.`

Only one copy of SPRITEST can be resident in memory at any one time. SPRITEST can be invoked with additional command parameters to adjust the display of the sprite image.

`ERROR: cannot install - DOS 2.0 or later required.`

Some of the DOS calls used by SPRITEST require MS-DOS version 2.0 or later.

`ERROR: cannot install - no TSR ID numbers available.`

To remedy this problem remove one or more TSR programs from the start-up sequence and reboot.

`ERROR: cannot operate in an OS/2 session.`

This version of SPRITEST will not execute in an OS/2 DOS session.

`ERROR: SPRITEST cannot operate in a command shell.`

Run SPRITEST from the primary command shell and load other programs after it.

`ERROR: unknown video controller. Spritest not loaded.`

SPRITEST checks for the presence of supported Seiko Epson video controllers before loading.

`Uninstall successful.`

All memory used by the SPRITEST is returned to the DOS memory pool and interrupt vectors used by SPRITEST are restored.

`ERROR: cannot uninstall - Spritest is not installed.`

SPRITEST was not loaded when the uninstall command was given.

`ERROR: cannot uninstall - interrupt vectors have changed.`

Another TSR has loaded after SPRITEST and altered the interrupt vectors. Spritest cannot restore the interrupt vector and therefore will not uninstall.

`ERROR: cannot uninstall - cannot free memory block.`

Successful uninstallation also depends on the proper restoration of memory to the DOS memory free pool. This message indicates that the memory management block associated with SPRITEST has been damaged in some way. The only way to remove SPRITEST now is to reboot the computer.

`ERROR: cannot uninstall - cannot free memory control block`

Same meaning as above.

`ERROR: could not open image file.`

The name given (or interpreted) as the filename parameter could not be found. Specify a different filename or give the complete path to the image file.

`ERROR: while reading image file`

DOS reported an error while reading the file. Most likely the image data file has been corrupted.

`ERROR: file does not appear to be a .BMP file.`

The check for the BMP file signature failed. Even if the rest of the image file format is correct the image will not be loaded.

`ERROR: BMP image is incorrect. Images must be 64x64 and 16 or less colors.`

If the image stored in the BMP file is greater than 64 pixels vertically or horizontally or contains more than sixteen colours the image will not be loaded.

`Defaulting to the internal image.`

This message is seen in conjunction with several image errors messages and indicates that SPRITEST will display the internal image. The internal image is included to ensure that an image file will always be available for display.

`ERROR: unknown error encountered.`

This error message should not occur. It is actually part of the general message display routine and no tested error should point to this message.

`ERROR: SPRITEST cannot operate under Windows 95 shell.`

Re-start your system in DOS mode and re-run SPRITEST.

S-MOS SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

HOTRDISP.EXE Display Utility

Drawing Office No. X12-UI-014-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

HOTDISP selects the active display via a Hotkey when both a CRT display and LCD panel are available.

1.1 Program Requirements

Video Controller	**:**	SPC8106, SPC8108
Display Type	**:**	LCD or CRT
BIOS	**:**	Seiko Epson VGA BIOS
DOS Program	**:**	Yes
DOS Version	**:**	2.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	Yes (with limitations), Windows 3.1x and Windows 95 (see note below)
Windows DOS Full Screen	**:**	Yes (with limitations), Windows 3.1x and Windows 95 (see note below)
OS/2	**:**	No

HOTDISP must to be installed outside a DOS shell. However, once it is installed a hotkey can be pressed either from a Windows DOS box or Windows DOS full screen to switch the active display.

1.2 Installation

Copy the file **hotdisp.exe** from the installation diskette to a directory on your hard drive which is in the DOS path. The directory does not need to be in the path. However, if it is not in the path then a full path name will have to be specified each time HOTDISP is run.

1.3 Usage

HOTDISP may be loaded as as an executable from the DOS command prompt, or as a device driver from CONFIG.SYS during the boot sequence.

1.3.0.1 DOS Command Line

To load HOTDISP as a TSR, add the following line to your AUTOEXEC.BAT file or enter it at the DOS command prompt:

```
hotdisp [/c][/u][/?]
```

Where:

`/c` Changes the hotkey sequence. This switch causes HOTDISP to prompt for the new hotkey sequence. Press the desired keys in sequence to form a hotkey sequence. A minimum of one to a maximum of three keys may be used to form the sequence. Use any keys except the following: Caps Lock, Num Lock, Scroll Lock, Print Screen and Pause/Break.

`/u` Removes HOTDISP from Memory.

`/?` Displays HOTDISP help message.

1.3.0.2 DOS Device Driver

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

```
device=[path]hotdisp.exe
```

Options are not valid in CONFIG.SYS.

1.3.0.3 Using HOTDISP

HOTDISP allows the user to switch displays on the fly. HOTDISP is a TSR program, therefore it can be used while running another program. To switch display, use the hotkey sequence displayed when HOTDISP is loaded. Press the keys in the sequence indicated and release them simultaneously. Pressing the LCD/CRT Hot Key Sequence (default is F9) will switch among CRT display, LCD panel and DoubleScan mode.

Active Display before Key Pressed	Active Display after Key Pressed
CRT	LCD
LCD	DoubleScan
DoubleScan	CRT

1.4 Comments

- HOTDISP will install in the primary copy of the DOS command shell only. HOTDISP will not install from menu or shell programs.
- This utility requires the Seiko Epson video BIOS and Extensions.
- Rather than keeping a separate data file to store Hot Key Sequence, HOTDISP modifies the HOTDISP.EXE file on disk. For this reason some virus scanners (i.e. those that maintain checksums or CRCs) may warn of possible virus infection in HOTDISP.EXE. This will only occur after a change to the hotkey sequence.
- HOTDISP may not always be able to remove itself from resident memory if other TSRs or Device Drivers are installed after it.
- HOTDISP occupies 2K of memory.
- A Beep indicates that an error has happened. This can be one of the following cases:

  1.Hot Key Sequence is entered to switch from CRT to LCD. However, LCD display may not be able to handle the current video mode. (i.e., 640x480 LCD panel cannot display mode 6Ah (800x600).)

  2.Hot Key Sequence is entered to switch from LCD to DoubleScan mode. However, a CRT display is not attached to the video board.

  3.Hot Key Sequence is entered, however the Video Controller is not ready to switch display yet. An error happened while using Seiko Epson BIOS Extension calls to switch display.

1.5 Program Messages

ERROR: HOTDISP is already installed and cannot be installed again.

HOTDISP has been installed as a TSR or a device driver. It cannot be installed twice.

ERROR: VGA required.

A VGA display adapter must be present to run HOTDISP.

ERROR: Requires Seiko Epson Video BIOS Extensions.

The Seiko Epson Video BIOS Extensions are not installed as part of the video system.

ERROR: VGA must be the active display.

A VGA display adapter must be active to run HOTDISP.

ERROR: As a device driver, HOTDISP does not accept any command line argument.

Run HOTDISP without any command line argument in CONFIG.SYS to install HOTDISP as a device driver.

ERROR: SPC8108/8106 required.

HOTDISP requires SPC8108 or SPC8106 LCD VGA Controller.

ERROR: DOS move file pointer error on HOTDISP.EXE.

A DOS file error has occurred.

ERROR: DOS read error on HOTDISP.EXE.

A DOS file error has occurred.

ERROR: DOS write error on HOTDISP.EXE.

A DOS file error has occurred.

ERROR: DOS close error on HOTDISP.EXE.

A DOS file error has occurred.

ERROR: cannot operate in a command shell.

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

ERROR: Cannot remove HOTDISP FROM MEMORY.

Other TSR programs have been installed after HOTDISP in the memory.

ERROR: cannot open HOTDISP.EXE. Update not done.

HOTDISP cannot save the hotkey sequence changes. The file HOTDISP.EXE may be read-only. Change the attribute of HOTDISP.EXE and run it again.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

FLASHROM.EXE Flash ROM BIOS Update Utility

Drawing Office No. X12-UI-015-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

FLASHROM is an OEM utility program which enables the user to program the Flash ROM on the SDU8106B0F Evaluation Board.

1.1 Program Requirements

Video Controller	**:**	SPC8106
Display Type	**:**	Any VGA
BIOS	**:**	Any VGA
DOS Program	**:**	Yes
DOS Version	**:**	3.0 or greater
Windows Program	**:**	No
Windows DOS Box	**:**	No
Windows DOS Full Screen	**:**	No
OS/2 DOS Full Screen	**:**	No

1.2 Installation

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

1.3 Usage

FLASHROM.EXE is run from the DOS command line as follows:

```
flashrom file.bin [segment] [/?]
```

Where:

file.bin	is the binary file to be loaded into the Flash ROM
segment	is the location of the Flash ROM in memory (in hex)
/?	displays the help message

If a memory segment is not specified FLASHROM.EXE defaults to C000h.

1.4 Comments

- This program updates the Flash ROM on the SDU8106F0E Evaluation Board only.
- FLASHROM supports only the Intel 28F256A and AMD 28F256 Flash ROMs.
- Video memory MUST not be shadowed.
- FLASHROM does not check for valid video card, video chip, or binary image.
- The SDU8106B0F board must be configured for Flash ROM programming.

1.5 Program Messages

!!!!!!!!!!!!!!!!!!!! WARNING !!!!!!!!!!!!!!!!!!!!
The default Video interrupt handler for interrupt 10h was replaced by an unknown software utility, such as a mouse driver. This program will force the system to reboot after the video BIOS has been updated.

Mouse drivers, display enhancement utilities, and other programs sometimes take control of the interrupt 10h vector. Unfortunately we cannot re-initialize the BIOS after its update without forcing a reboot in this case. If you do not want the system rebooted press the Esc key to abort FLASHROM. If rebooting the system is acceptable, press any other key and the system will boot after the BIOS is initialized. To avoid this message, you can boot the system using several clean boot techniques as described in your DOS manuals, then run FLASHROM.

Not enough memory!

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

Programming Flash Rom at [segment]...

FLASHROM.EXE is now programming the Flash ROM at the memory segment specified.

Please wait...

FLASHROM is working.

Flash ROM Programmed OK...

FLASHROM has completed programming the Flash ROM.

Unsupported EEPROM Chip Identified : ID=AA55
This could be also caused by chip being shadowed.

FLASHROM has identified the Flash ROM as neither the Intel 28F256A or AMD 28F256, or the video BIOS is being shadowed. Remove the video BIOS shadowing or replace the Flash ROM with either the Intel 28F256A or AMD 28F256 and try again.

ERROR: Failed to open “file.bin”

FLASHROM has failed to open the specified binary file. Check to ensure that the file is in the correct format.

ERROR: Invalid file.

The file which you are attempting to load into Flash ROM is not a valid file type, only VGA BIOS images can be loaded into the Flash ROM.

ERROR: File larger then 32k.

The file which you are attempting to load into Flash ROM is larger than 32 Kilobytes in size, and therefore is probably not a valid VGA BIOS image.

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

Programming the Sprite Hardware

Drawing Office No. X12-AN-002-01

.

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

In video technology the term sprite usually refers to a bitmap shape that can be moved across complex backgrounds without flicker or damage to the background image. The SPC8106 supports a hardware assisted sprite capable of displaying a 64 x 64 pixel 4-color image, regardless of the video mode. Typical uses for sprites are for hardware cursors in GUI environments, or pop-ups for system status.

The intent of this document is to demonstrate the rudiments required to display and manipulate a sprite image with the SPC8106. It is assumed the reader has an understanding of the VGA register architecture. The code samples included are C-like and are intended to show concepts, and have not been compiled. Most sprite control is performed through two index/data register pairs and the sprite image is stored in unused video memory.

The first set of sprite control registers are located in the CRTC (port 3D4h/3D5h) at indexes 30h through 38h. These indexes are beyond those defined by IBM for the VGA and control functions such as sprite image address, screen position, and clipping.

The second set of sprite control registers are located in the Auxiliary Register (port 3DEh/3DFh) at indexes 05h, 09h and 0Bh. These registers control sprite enable, memory mapping and sprite color control.

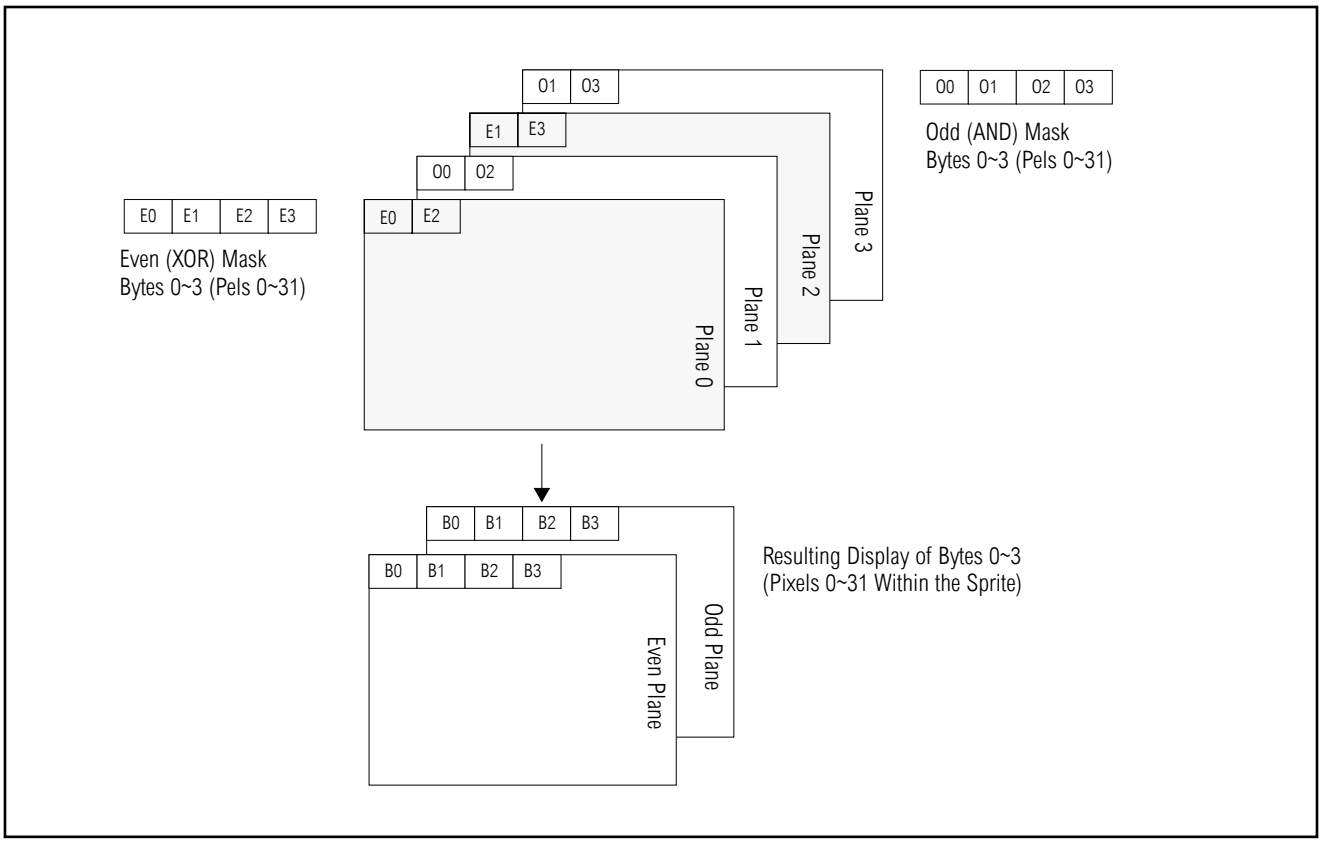
The sprite image should be placed in normally unused video memory. Each sprite image requires 1KB of memory (the SPC8106 supports 512KB) so there are a possible 512 possible sprite 'slots' in display memory. To ensure the sprite image is in unused video display memory it should be placed in the upper 256KB. Placing the image in the lower 256KB of video memory could permit a standard VGA memory access to overwrite the image.

The programming examples contained within this document utilize a memory technique which swaps the upper and lower 256KB of memory. The examples also use a feature called Sprite Write Mode. Upper page swapping permits the upper 256KB of video memory to be addressed through the A000h segment, while swapping the normal video memory to the B000h segment. Upper page swapping is proprietary to the SPC8106 and does not affect the display contents (to avoid memory manager conflicts, do not update the video memory in the B000h segment). This upper page swapping technique is consistent with the memory access methods required by previous Seiko Epson VGA LCD controllers which incorporate a hardware sprite.

Sprite Write Mode is a consistent programming interface for placing sprite image data into display memory, this allows programming of the sprite in one consistent format no matter what current memory model that the controller is in. This mode eliminates the complexity of setting up the memory as depicted in the following diagram by allowing the sprite image to be envisioned as two streams of pixel data, one stream representing the most significant bits of each pixel and the other representing the least significant bits.

> The sprite hardware does not support a packed pixel format. Prior to loading a sprite image, the image data must be broken down into two arrays of pixels. This format is typical of the XOR and AND arrays provided for the cursor image in such popular GUIs as Windows and OS/2.

The following diagram depicts the VGA memory in a planar format and the sprite as displayed on the screen. The two data structures represent the strings of pixel information where each bit of Bytes 0 through 3 represent one pixel each (bits 0 through 31). The high order bits of the image (cursor XOR mask) are placed in the even planes (planes 0 and 2) and the low order bits (AND mask) are place in the odd planes (planes 1 and 3).



Each VGA memory address spans four planes of data (to give 32 bits). Sprite images consist of only two bits per pixel. Each VGA memory address therefore contains sixteen pixels of sprite image. One complete row of a sprite image (64 pels) is held in four sequential VGA memory addresses. 256 addresses (x 4 planes = 1KB) are required to hold a complete sprite image.

For cursor images (typically 32 x 32 pels) only 1/4 of the entire sprite area is required. In this situation the remaining 3/4 of the sprite should be set to transparent (AND pels = 1 and XOR pels = 0). When less than entire 64 rows are required as in the case of a 32x32 cursor, it is more CPU efficient to place the image in the lower portion of the sprite memory, because CPU fetches to video memory are reduced. The recommended location for a 32x32 cursor data is starting at the 256th byte when using Sprite Write Mode. The Y Start Row Register CRTC [35h] should then be set to clip the upper portion of the cursor data (i.e., the undisplayed portion). Unfortunately, for each row or partial row displayed, the full width of the row (all 64 pels) is read into an internal buffer for display, and CPU efficiency cannot be improved.

The following example demonstrates how to load a sprite image (64 x 64) into slot 256 (offset 0 in the upper 256K).

```
/* Defines */
/*---------*/
#define AUX 0x3DE
#define CRTC 0x3D4

char far* pDisplayMem, pImgData;
```

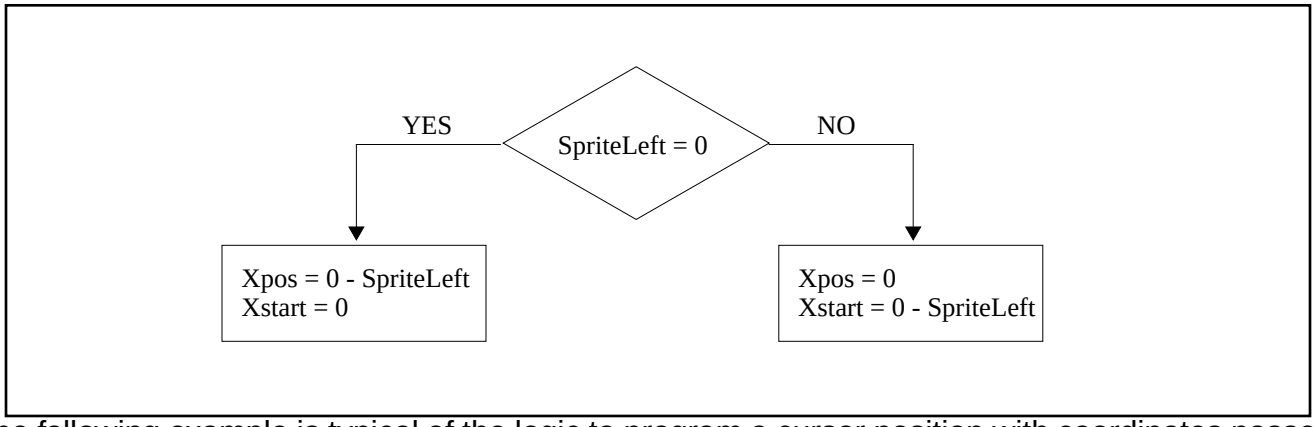
```
/* Enable the Sprite Hardware Functions */
/* ---------------------------------- */
// Enable the sprite (AUX[5] bit 5) and disable the upper 256k (AUX[5] bit 0).
outp (AUX, 5);                    // Select index 5.
outp (AUX + 1, 0x20);             //

// In AUX register 9 select sprite page 0 (b3), logical plane 0 (b2)
// upper page swap (b1) and Sprite Write Mode (b0)
outp (AUX, 9);                   // Select index 9.
outp (AUX + 1, 3);                // 0011b

// The SPC8108 sprite write mode did not disable the sequencer chain4
// bit and required the following code to ensure correct operation in all
// video modes. The SPC8106 sprite write mode correctly disables chain4 and
// the following code may be omitted in cases where this code will be run on
// an SPC8106 only. Note that including this code ensures the correct handling
// of the sequencer for both video controllers. Its use is transparent on
// the 8106 and it imposes very little execution overhead.

// Temporarily disable chain4 addressing (should we be in mode 13h)
nSeqIdx = inp(SEQ);              // Read and save current index
outp(SEQ, 4);                    // Set index 4
nSeqDat = inp(SEQ + 1);          // Read and save the data.
nTmp = inp(SEQ + 1);             // Read the current data.
nTmp &= 2;                       // Mask off the chain 4 bit
outp(SEQ + 1, nTmp);             // Set the new data

/* Load the Image into Display Memory */
/* ---------------------------------- */
// Set the pointers to both display memory and the even pixel image array.
// (512 bytes of Least Significant Bits of color data)
// Loop 512 times moving the pixel image array to display memory. Sprite
// Write Mode handles switching of planes placing even addressed data into
// plane 0 and odd addresses into plane 2.
pImgData = EvenArray;             // (XOR array if a cursor)
pDisplayMem = A000H:0000          //
for (x = 0; x < 512; x++)
  *pDisplayMem++ = *pImgData++;

// Select sprite logical plane 1 (AUX[9] b2).
// This step requests that even addressed data be placed into plane one
// and odd address be placed into plane three.
outp (AUX, 9);                    // Select index 9.
outp (AUX + 1, 7);                // In this example I know the state of
```

```
                                    // AUX[9] so I will forego a read/mask/write
                                    // and just write blind turning b2 on.

// Set the pointer to the start of the odd pixel array
// (512 bytes of Most Significant Bits of color data) and reset the
// display memory pointer. Loop 512 times moving image bytes from the
// image array to display memory.
pImgData = OddArray;                // (AND mask for a cursor)
pDisplayMem = A000H:0000;
for (x = 0; x < 512; x++)
   *pDisplayMem++ = *pImgData++;

/* Restore the upper page swap bit (AUX[9] b0). */
// This step MUST be performed. If not done display data meant for A000h
// segment will be actually written to the B000h segment.
outp (AUX, 9);                      // Select index 9.
outp (AUX + 1, 0);                  // May as well turn them all off.
                                   // We don't need them any more.

/* Restore the sequencer index and data. */
// This step only required if the previous step for SPC8108
// use of this code in mode 13h was performed.
outp (SEQ, 4);                      // Set index 4
outp (SEQ + 1, nSeqDat);            // Restore the index 4 data.
outp (SEQ, nSeqIdx);                // Restore the sequencer index.

/* Set the Sprite Position Registers */
/* -------------------------------- */
// This informs the chip where to find the image in display memory.
// The example image is located in the first 256 bytes in the upper 256k.
// To locate this image set CRTC[36] b0 to 1 and CRTC[37]to 0.
outp (CRTC, 0x36);                  // Other image 'slots' will
outp (CRTC + 1, 1);                 // require the calculation of
outp (CRTC, 0x37);                  // the offset in which to find
outp (CRTC, 0);                     // the image.

// For sprites ensure that CRTC[38] b1 is 0 in order to enable the
// four palette colors. For cursors CRTC[38] b1 is set to 1 to enable
// the logical screen/!screen palette. The hardware sprite also supports
// horizontal and vertical doubling based on bits 2 & 3 of this register.
outp (CRTC, 38H);                   // Select CRTC 38H
outp (CRTC +1, 0);                  //
```

```
/* Set the image colors. */
// Note that for cursor images only indexes 3CH and 3DH require
// programming as the other two colors are displayed as transparent
// and inverted.
outp (AUX, 0BH);                // Set AUX index to 0Bh
nAux = inp (AUX + 1);           // Read and save the current index
nTmp = inp (AUX + 1);           //
// Note: The mask value of 8 (below) enables both the internal (LCD)
// and external (CRT) palettes. If it is desired to write different
// values to these palettes then this section of code will have to
// be executed twice with different palette write disable bits set.
nTmp |= 8;                      // Mask in the overlay enable.
outp (AUX +1, nTmp);

// SetDAC is a routine to program one RAMDAC entry. This type of code is
// a standard VGA routine and is not included in this example. The data type
// for color1, color2 ... would be long and these variables would be composed
// of RGB values.
SetDAC (3CH, color1);           // Set the sprite colors.
SetDAC (3DH, color2);           //
SetDAC (3EH, color3);           //
SetDAC (3FH, color4);           //

out (AUX + 1, nAux);            // Restore the AUX register data

// The following piece of code pertains to the SPC8108.
// The SPC8106 sprite colors are set by the user defined function
// SetDAC. As with the chain4 override the following is required
// if the program will be used with an SPC8108. The code executes
// transparently on an SPC8106 with very little execution overhead.

// Set the internal LUT colors. If this is a cursor image then only
// registers 3CH and 3DH need to be set.
outp (CRTC, 3CH);                // Sprite palette 0
outp (CRTC +1, color1);          // Cursor background
outp (CRTC, 3DH);                // Sprite palette 1/
outp (CRTC +1, color2);          // Cursor foreground
outp (CRTC, 3EH);                // Sprite palette 2
outp (CRTC +1, color3);          //
outp (CRTC, 3FH);                // Sprite palette 3
outp (CRTC +1, color4);         //

/* Sprite initialization complete. */
// The sprite/cursor image may be removed from the display
// by setting AUX[5] b5 to 0.
```

1.1 Setting the Displayed Sprite Position

The displayed position of the sprite and any required clipping is done with CRTC registers 30h through 35h. Clipping of the sprite image on the right and bottom of the display is done automatically by the hardware. If the sprite image needs to be clipped on the top or left of the display the clipped amount needs to be calculated and set into the appropriate registers.
All position calculations are based on absolute pixel positions. This holds even for text modes, which means the sprite image location is not limited to character cell boundaries, but may be placed on any pixel on the display. Sprite positioning information is always relative to the upper left corner of the sprite image and to the upper left corner of the display. The following example demonstrates how to program a cursor image located in the lower left qaudrant of the sprite as this covers all positioning considerations.
In general the algorithm for cursor position and clipping is:



The following example is typical of the logic to program a cursor position with coordinates passed by Microsoft Windows.

```
/* Calculate the Xposition and Xstart. */

CursorX = CursorPos - HotSpot // nCursorX = raw X position
                              // (SpriteLeft)
if (CursorX > 0) {
         Xpos = CursorX;
         Xstart = 0;
         }
else {
         Xpos = 0;
         Xstart = 0 - CursorX
         }

/* Now program the X position registers. */
// First the X start register is set. This value informs the
// hardware how many pixels to count across into the image
// before starting display.
```

```
outp (CRTC, 34H);
outp (CRTC +1, nXstart);

// Program the X position. As the X position may be > 255
// two data register writes are required.
// IMPORTANT - The position registers do not latch until the
// high position is written.

outp (CRTC, 31H);
outp (CRTC +1, LOBYTE(Xpos));
outp (CRTC, 30H);
outp (CRTC +1, HIBYTE(Xpos));

/* Calculate the Yposition and Ystart. */
// As this is a cursor programming sample we will assume the
// cursor has been loaded into the bottom left quadrant of the
// sprite image for performance considerations.
// The logic for setting Y position under these circumstances
// is much the same however the Y start will always be at
// least 32 pixels into the image area.

nCursorY = nCursorPos - nHotSpot  // nCursorY = raw Y position
                                   // (SpriteTop)
if (nCursorY > 0) {
         nYpos = nCursorY;
         nYstart = 32;
         }
else {
         nYpos = 0;
         nYstart = 0 - nCursorY + 32;
         }

/* Now program the Y position registers. */
// First the Y start register is set. This value informs the
// hardware how many pixels to count down into the image before starting display.

outp (CRTC, 35H);
outp (CRTC +1, nYstart);

// Program the Y position. As the Y position may be > 255
// two data register writes are required.
// IMPORTANT - The position registers do not latch until the
// high position is written.
```

```
outp (CRTC, 34H);
outp (CRTC +1, LOBYTE(Ypos));
outp (CRTC, 33H);
outp (CRTC +1, HIBYTE(Ypos));

// End of programming spirte/cursor position. //
```

S-MOS SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

Windows 3.1 16-Color Panning Display Utility

Including VLIMIT.EXE Display Utility

Drawing Office No. X12-DI-001-02

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

The SPC8106 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 Panning Display 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 Panning Display 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 SPC8106 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

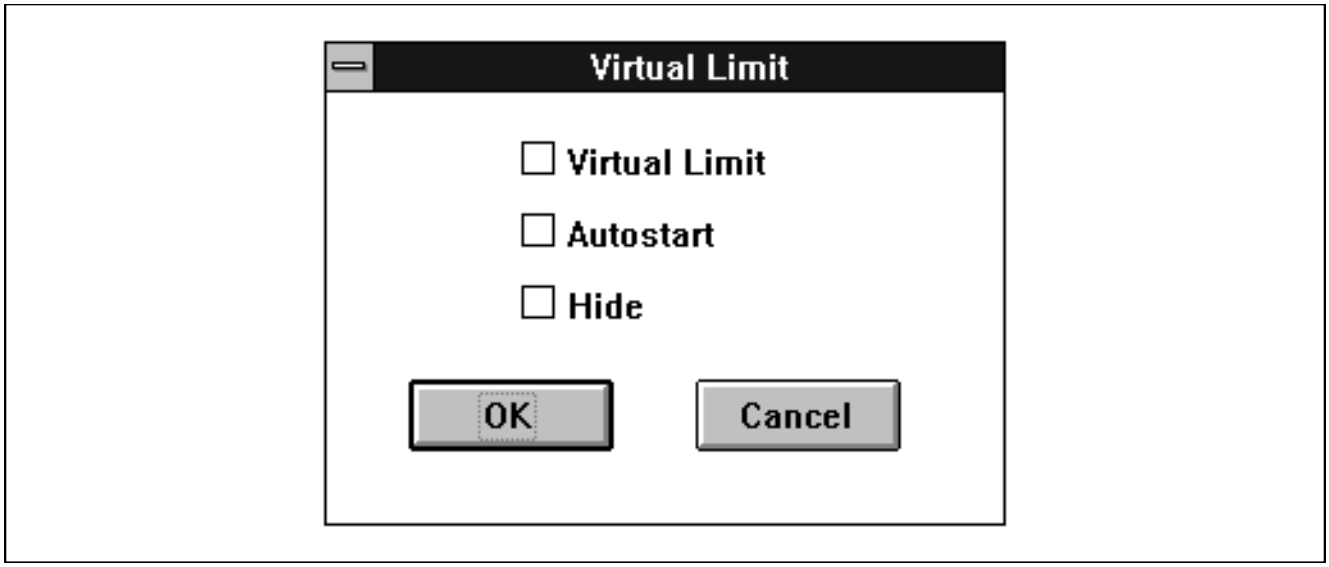
2. use the file manager to “drag” the program into a program manager group.

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

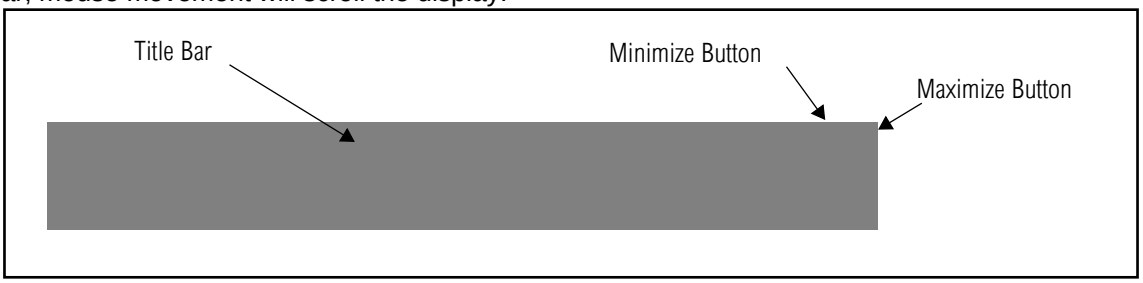
Check Autostart to have VLIMIT run automatically every time you start Windows. Options in effect during the last VLIMIT session will still be in effect on the next windows start-up.

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

SPC8106 LCD/CRT VGA CONTROLLER

Windows 3.1 256-Color Display Utility

Drawing Office No. X12-DI-002-02

.

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

Unlike the earlier versions of the SPC8106, the SPC8106 supports 256 colors at a resolution of 640x480. To allow Windows 3.1 to take advantage of the higher color depth at 640x480, Seiko Epson provides a 256 color Windows 3.1 driver. This driver optionaly can use the SPC8106 hardware cursor (HW cursor) to gain a slight performance improvement.

1.1 Requirements

Video Controller	**:**	SPC8106
Display Type	**:**	LCD or CRT
BIOS	**:**	Seiko Epson VGA BIOS
DOS Program	**:**	No
DOS Version	**:**	3.0 or greater
Windows Program	**:**	Windows 3.1x only

1.2 Installation

Your original Windows disks will be required for installation.

1.2.0.1 First Time Installation

The standard Windows VGA driver should be installed prior to the SPC8106 driver installation. If the SPC8106 video driver or a previous version of the SPC8106 video driver has been installed for Windows see the Secondary Installation section of this manual for driver installation.

Install the SPC8106 256-color driver as follows:

From DOS

- Exit Windows.
- Change into the Windows directory.
- Run setup.exe from the DOS prompt.
- Change the display to “Other display (requires disk from OEM)”.
- Follow the directions displayed on the screen.

From Windows

- Double click the left mouse button on the “Windows Setup” icon.
- Select “Options”.
- Select “Change System Settings”.
- Change the display from “VGA” to “Other display (requires disk from OEM)”.
- Select “OK”.
- Follow the directions displayed on the screen.

1.2.0.2 Secondary Installation (Updating Driver)

To install this video driver over a previously installed version of the driver:

From DOS

- Change into the Windows directory.
- Run setup.exe from the DOS prompt.
- Change the display to “Other display (requires disk from OEM)”.
- When setup.exe prompts you to keep the current (installed) driver or replace the current driver, press the ESC key to replace the driver.
- Follow the directions displayed on the screen.

To install the SPC8106 driver over a previously installed version of the driver from within Windows you must first install the standard Windows VGA driver, otherwise Windows Setup will complain about open and write protected files.

From Windows

- Double click the left mouse button on the “Windows Setup” icon.
- Select “Options” then select “Change System Settings”.
- Change the display to “VGA”.
- Select “OK”.
- Follow the directions displayed on the screen.
- After installation of the standard Windows VGA driver is complete re-run Windows Setup.
- Select “Options” then select “Change System Settings”.
- Change the display to “Other display (requires disk from OEM)”.
- Select “OK”.
- At the “Change Systems Settings” prompt select “New”.

- Follow the directions displayed on the screen.

1.3 Operation

There are no operational differences from the standard Windows 3.1 VGA driver.

1.4 Comments

- This Driver requires the Seiko Epson video BIOS.

S-MOS SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

SDU8106B0E Rev. 1.0 Evaluation Board User Manual

Drawing Office No. X12-AN-001-01

THIS PAGE INTENTIONALLY BLANK

TABLBE OF CONTENTS

THIS PAGE INTENTIONALLY BLANK

1.0 SDU8106B0E REV. 1.0 EVALUATION BOARD

This manual reflects the use of the SDU8106B0E Rev 1.0 Evaluation Board with the SPC8106 LCD CRT VGA Controller Chip. This evaluation board is backwards compatible with all revisions of the SPC8106 as well as the SPC8108. Refer to the SPC8106 Functional Specification, Drawing Office No. X18-SP-001-xx for further details.

1.1 Features

- CRT support with ATT20C497 CMOS RAMDAC (or equivalent; see Technical Report V94-811, “Current RAMDAC’s supported by SPC8108/8106 VGA Controllers”)
- Optional support for SPC8108 VGA Controller Chip
- LCD support for 4/8/16-bit monochrome and color STN LCD panels (single drive, dual drive), RGBI, 12-bit RGB, and 9-bit active matrix TFT panels
- 512K DRAM video memory (1x256Kx16 DRAM installed, SMD, SOJ or TSOP package, symmetrical or asymmetrical architecture)
- Video BIOS support provided by on-board reprogrammable Flash EPROM
- Support for Hardware Power Save Mode via DIP switch
- Support for Sprite/Hardware Cursor on both LCD and CRT displays
- On-Board +35 volt adjustable power supply (VDDH for LCD panels requiring a positive BIAS voltage)
- On-Board -23 volt adjustable power supply (VLCD for LCD panels requiring a negative BIAS voltage)
- Highly configurable via jumpers and DIP switches
- 16-Bit ISA bus support
- Two terminal crystal support (PTH and SMD packages)
- External oscillator support
- Power measurement support
- Prototyping area
- Test point header strips
- PTH sockets for SPC8106, ATT20C497, and support chips
- Adjustable VCC Core voltage
- Adjustable VCC I/O voltage

2.0 INSTALLATION AND CONFIGURATION

2.1 Configuration Jumpers

The following table shows the jumper selections available for this board.

Table 0-1 SDU8106B0E Configuration Jumper Settings

Jumper	1 - 2	2 - 3
JP1 JP2	SPC8108 setting	SPC8106 setting
JP3	IRQ connected to ISA bus IRQ9	N/C
JP4	16-bit LCD interface, using external latch	16-bit LCD interface, direct from SPC8106

JP1 and JP2 Select VGA Controller Chip (socket U1)

When a SPC8108 is installed, set JPI and JP2 to position 1-2. When a SPC8106 is installed, set JPI and JP2 to position 2-3. See “SDU8106B0E Rev. 1.0 Schematic Diagrams” on page 16.

JP3 Vertical Retrace Interrupt

If enabled, a vertical retrace will generate an interrupt on the SPC8106F0C IRQ pin. To connect IRQ to ISA bus IRQ9, set jumper JP3 to position 1-2; otherwise set the jumper to position 2-3.

JP4 Select latch for 16-bit LCD panel drive

To support 16-bit panels using an external latch, set JP4 to position 1-2 and remove the 16-pin DIP shunt at U9. (Configuration input MD7 must also be set to the open position). When the SPC8106 is in non-CRT mode, the SPC8106 can directly drive a 16-bit panel by setting JP4 to position 2-3 and installing the DIP shunt at U9. (Configuration input MD7 must also be set to the closed position).

2.2 Configuration DIP Switches

The SDU8106B0E DIP switch inputs are mapped directly from the SPC8106 configuration inputs MD0 - MD4,
MD6 - MD12, and MD14 - MD15. Configuration inputs MD5 and MD13 are not selectable and are set to a logic low on power-up. MD5 selects a standard ISA bus interface, and MD13 enables CRT sprite capabilities.
Two DIP switches, SW1-6 and SW2-6, are used for board logic, as described below.

Table 0-2 SDU8106B0E Configuration DIP Switch Settings

Switch	Signal	Open	Closed
SW1-1	MD0	LCD Panel Configuration bit 0	see Table 2-3
SW1-2	MD1	LCD Panel Configuration bit 1	
SW1-3	MD2	LCD Panel Configuration bit 2	
SW1-4	MD3	LCD Panel Configuration bit 3	
SW1-5	MD4	16-Bit I/O Port	8-Bit I/O Port
SW1-6		Disable Hardware Power Save mode (SUSPEND)	Enable Hardware Power Save mode (SUSPEND)
SW1-7	MD6	Symmetrical Video RAM	Asymmetrical Video RAM
SW1-8	MD7	16-bit LCD interface supported via external latch	16-bit LCD interface supported directly by SPC8106
SW2-1	MD8	5.0 volt core voltage	3.3 volt core voltage
SW2-2	MD9	LCD Panel Configuration bit 4	see Table 2-3
SW2-3	MD10	256 cycle / 32 ms DRAM refresh	256 cycle / 4 ms DRAM refresh
SW2-4	MD11	PSM DRAM refresh source Bit 0	see Table 2-4
SW2-5	MD12	PSM DRAM refresh source Bit 1	
SW2-6		Disable writes to Flash EPROM	Enable writes to Flash EPROM
SW2-7	MD14	PSM DRAM refresh source Bit 2	see Table 2-4
SW2-8	MD15	3C3h Video Enable Port	46E8h and 102h Video Enable Port

= not supported when using the SPC8108

MD0-MD3 & MD9 LCD Panel Size Configuration

These bits are used to indicate the LCD panel size and type to the video BIOS software as shown in the following table.

Table 0-3 SPC8106 LCD Panel Size Configuration

MD9	MD3	MD2	MD1	MD0	Resolution	Mode	Bits	Comments
1	1	1	1	1	640x480	Dual	8	Monochrome
1	1	1	1	0	640x480	Single	8	Monochrome
1	1	1	0	1	640x400	Dual	8	Monochrome
1	1	1	0	0	640x400	Single	8	Monochrome
1	1	0	1	1	640x200	Dual	8	Monochrome
1	1	0	1	0	640x200	Single	4	Monochrome
1	1	0	0	1	480x320	Dual	8	Monochrome
1	1	0	0	0	480x320	Single	4	Monochrome
1	0	1	1	1	320x256	Dual	8	Monochrome
1	0	1	1	0	320x256	Single	4	Monochrome
1	0	1	0	1	320x240	Dual	8	Monochrome
1	0	1	0	0	320x240	Single	4	Monochrome
1	0	0	1	1	320x200	Dual	8	Monochrome
1	0	0	1	0	320x200	Single	4	Monochrome
1	0	0	0	1	320x240	Single	16	Color
1	0	0	0	0	640x480	Single	12	TFT MODE [a]
0	1	1	1	1	640x480	Dual	8	Color
0	1	1	1	0	640x480	Single	8	Color
0	1	1	0	1	640x480	Dual	16	Color
0	1	1	0	0	640x480	Single	16	Color
0	1	0	1	1	640x480	Dual	9	RGBI
0	1	0	1	0	640x480	Single	9	TFT MODE [a]
0	1	0	0	1	640x480	Dual	12	RGB
0	1	0	0	0	320x240	Single	4	Color
0	0	1	1	1	640x480	Single	12	TFT MODE [b]
0	0	1	1	0	640x480	Single	9	TFT MODE [b]
0	0	1	0	1				For OEM configuration
0	0	1	0	0				For OEM configuration
0	0	0	1	1				For OEM configuration
0	0	0	1	0				For OEM configuration
0	0	0	0	1				For OEM configuration
0	0	0	0	0				For OEM configuration

= not supported when using the SPC8108F0C

0 = closed, 1 = open

a. AUX[00]b5=1 and AUX[0B]b1=1
b. AUX[00]b5=1 and AUX[0B]b1=0

Although the SPC8106 does support 12-bit active matrix TFT panels and 12-bit RGB displays, this board design does not provide all the necessary signals to the LCD connector (J6). If 12-bit support is required, refer to Table 2-5, “SDU8106B0E LCD Connector J6 Pinout,” on page 11 for pin mapping.

MD4 ISA Port Width

The SDU8106B0E is a 16-bit ISA I/O card. To restrict bus transactions to 8-bits only, set MD4 to the closed position. For 16-bit width, open MD4.

SW1-6 Test Hardware Power Save Mode

When SW1-6 is set to the open position, the SPC8106 will operate normally. When SW1-6 is set to the closed position, the /SUSPEND line is pulled low and the SPC8106 enters hardware power save mode. This DIP switch is provided for testing purposes only.

MD6 and MD10 Video DRAM Configuration

The SDU8106B0E is assembled with one 256Kx16 DRAM, SMD, SOJ package. The DRAM will have either a symmetrical architecture (9 rows x 9 columns; 2 / CAS signals) or an asymmetrical architecture (10 rows x 8 columns; 2 /WE signals). Open the switch at MD6 for symmetrical architecture; close the switch for asymmetrical architecture.
For 256 cycle / 32 ms refresh rate DRAM open the switch at MD10; close it for 256 cycle/ 4 ms refresh rate.

MD7 16-Bit LCD Interface Support

When MD7 is set to the open position, a 16-bit LCD interface is supported through an external latch (JP4 must be set to position 1-2 and the 16-pin DIP shunt at U9 must be removed). When MD7 is set to the closed position, a 16-bit LCD interface is directly supported by the SPC8106 (JP4 must be set to position 2-3 and the DIP shunt at U9 must be installed).

While directly supporting a 16-bit LCD interface, the SPC8106 will not support a CRT. See “SDU8106B0E Rev. 1.0 Schematic Diagrams” on page 16.

MD8 Core Operating Voltage

The SPC8106 can operate with 5.0 volt or 3.3 volt core operating voltage. To select 5.0 volt, open the switch at MD8; to select 3.3 volt, close the switch. (Not supported when using the SPC8108F0C.)

MD11 - MD12 and MD14 Power Save Mode DRAM Refresh Source

During SPC8106 Hardware Power Save Mode (and Power Save Mode 4), provision must be made to refresh the video DRAM. As shown in the following table, there are four options. Option one uses the frequency source connected to CLK1. Option two uses the ISA /REFRESH signal (which is connected to SPC8106 pin MEMEN). Option three is for the case when the DRAM supports self-refresh. Option four selects a 64 kHz signal connected to SPC8106 pin PDCLK. (MD14 is not supported when using the SPC8108F0C.)

Table 0-4 SPC8106 Power Save Mode DRAM Refresh Source Select

MD14	MD12	MD11	Option	Refresh Source
1	0	0	1	CLK1
1	0	1	2	MEMEN (ISA / REFRESH
1	1	0	3	Self Refresh DRAM
1	1	1	5	PDCLK (64 kHz)

0 = closed, 1 = open

SW2-6 Enable Writes to Flash EPROM

When SW2-6 is set to the open position, writes to the Flash EPROM will be disabled. When SW2-6 is in the closed position, writes to the Flash EPROM are allowed. This is used to prevent accidental writes to the Flash EPROM

MD15 Video Enable Port Address

The SPC8106 has the capability of enabling the video system through port address 3C3h or through 46E8h and 102h. This capability is provided for when it is desired to have more than one video system installed in a PC.

To select port address 3C3h, open the switch at MD15; to select 46E8h and 102h, close the switch.

The BIOS supplied with the SDU8106B0E Evaluation Board will use 3C3h to enable the video system. If it is desired to use 46E8h and 102h, a different BIOS will be required. (Not supported when using the SPC8108F0C.)

2.3 Connecting LCD Panels

The LCD Signals are found on the 40-pin connector J6. Refer to the following table for signal assignments.

Table 0-5 SDU8106B0E LCD Connector (J6) Pinout

SPC8106 Pin Name	LCD Con. Pin No.	TFT color			Color STN LCD			Mono STN LCD	
		9-bit	12-bit AUX[00]b5=1 AUX[0B]b1=0	12-bit AUX[00]b5=1 AUX[0B]b1=1	16-bit	8-bit	4-bit	8-bit	4-bit
LD0	1	R0	R1	R1	LD0	LD0		LD0	
LD1	3	G0	G1	G1	LD1	LD1		LD1	
LD2	5	G1	G2	G2	LD2	LD2		LD2	
LD3	7	G2	G3	G3	LD3	LD3		LD3	
LD4	9				LD4				
LD5	11				LD5				
LD6	13				LD6				
LD7	15				LD7				
UD0	17	R1	R2	R2	UD0	UD0	UD0	UD0	UD0
UD1	19	B0	B1	B1	UD1	UD1	UD1	UD1	UD1
UD2	21	B1	B2	B2	UD2	UD2	UD2	UD2	UD2
UD3	23	B2	B3	B3	UD3	UD3	UD3	UD3	UD3
UD4	25				UD4				
UD5	27				UD5				
UD6	29				UD6				
UD7	31				UD7				
XSCL	37	CLK	CLK	CLK	XSCL	XSCL		XSCL	XSCL
XSCL2	39	R2	R3	R3		XSCL2	XSCL2		
LP	40	HS	HS		LP	LP	LP	LP	LP
YD	38	VS	VS		YD	YD	YD	YD	YD
WF	36	DE	DE	DE	WF	WF	WF	WF	WF
VDDH	34				VDDH	VDDH	VDDH	VDDH	VDDH
VLCD	32							VLCD	VLCD
LCDPWR#	30	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#
+5 V	24	+5 V	+5 V	+5 V	+5 V	+5 V	+5 V	+5 V	+5 V
+12 V	22	+12 V	+12 V	+12 V	+12 V	+12 V	+12 V	+12 V	+12 V
GND	20-2	GND	GND	GND	GND	GND	GND	GND	GND
HSYNC#	N/C			HS					
VSYNC#	N/C			VS					
OL0	N/C		B0	B0					
OL1	N/C		G0	G0					
OL23	N/C		R0	R0					

3.0 TECHNICAL DESCRIPTION

3.1 ATT20C497 CMOS RAMDAC

The ATT20C497 is provided on the board to fully test all of the CRT display modes available with the SPC8106 including sprite/hardware cursor display.
The ATT20C497 does not support hardware power down mode (only a software sleep mode is provided). As a result, when the SPC8106 enters hardware power save mode, the ATT20C497 becomes inactive rather than powering down. The RGB outputs will be turned off. The CPU will not be able to access palette RAM as the /DACRD and /DACWR signals generated by the SPC8106 will be inactive.
The ATT20C497 has been provided with an external voltage reference.

3.2 ISA Bus Interface

This evaluation board supports the 16-bit ISA Bus via standard AT edge connectors. The board will not work in an 8bit slot. The board will support 8-bit or 16-bit access as configured by MD4. All standard VGA I/0 addresses are supported. The video BIOS is mapped to the address range C0000h through C7FFFh.

3.3 Video Memory

512K video memory is installed on the board. A single 256kxl6 DRAM, SMD, SOJ or TSOP package is used. The DRAM can either have symmetrical or asymmetrical architecture (selected by MD6).

3.4 Adjustable LCD Panel Positive Power Supply

The majority of color STN LCD panels require a positive power supply VDDH to provide between +23V and +40V (Iout=45 mA). 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 J6 at pin 34. Output voltage is varied by potentiometer R53. To prevent panel damage, VDDH power supply sequencing is implemented by enabling the VDDH power supply only when the SPC8106 signal /LCDBIAS is at a low logic level.

3.5 Adjustable LCD Panel Negative Power Supply

The majority of monochrome STN LCD panels require a negative power supply VLCD to provide between -18V and -23V (Iout=45 mA). 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 J6 at pin 32. Output voltage is varied by potentiometer R55. To prevent panel damage, VLCD power supply sequencing is implemented by enabling the VLCD power supply only when the SPC8106 signal /LCDBIAS is at a low logic level.

3.6 Hardware Power Save Mode

This evaluation board has provision for testing the SPC8106 hardware power save mode. When DIP switch SW 1-6 is open, the SPC8106 will operate normally. When SW1-6 is closed, the SPC8106 will enter hardware power save mode (SUSPEND).

3.7 Flash EPROM VGA BIOS Support

A 32K Flash EPROM contains the VGA BIOS and is reprogrammable on-board. The EPROM is configured to respond to addresses in the range C0000h through C7FFFh. This board design and associated programming software has been designed to work with AMDand Intel Flash EPROMs only.
The evaluation board has circuitry to prevent accidental programming or erasure of the Flash EPROM. When DIP switch SW2-6 is in the open position, writes to the Flash EPROM are inhibited. When SW2-6 is in the closed positing, erasure and reprogramming is allowed.
Erasure and reprogramming of the Flash EPROM is implemented through device specific algorithms.

3.8 Adjustable 3.3 Volt Power Supply

The SPC8106 VCC Core voltage can be either 3.3 volts or 5.0 volts (as selected by MD8). To provide testing when 3.3 volts is selected, an independent adjustable 3.3 volt power supply, PCBVDD1, has been provided on-board. This power supply is adjustable from 2.0 volts to 4.0 volts at 1 amp using R61.
To connect PCBVDD1 to VCC Core, install R57 and remove R65.

3.9 Adjustable 5.0 Volt Power Supply

For the SPC8106, VCCI/O can be either 3.3 volts or 5.0 volts. However, the SDU8106B0E evaluation board does not support 3.3 volt I/O operation and therefore the VCC I/O is restricted to 5.0 volts only. To provide full testing of VCC I/O at 5.0 volts, and for when VCC Core voltage is selected to be 5.0 volts (selected by MD8), an independent adjustable 5.0 volt power supply PCBVDD2 has been provided on board This power supply is adjustable from 4.0 volts to 6.0 volts at 1amp using R63.
To connect PCBVDD2 to VCC Core, install R65 and remove R57. To connect PCBVDD2 to VCC I/O, install R66 and remove R58.

3.10 Power Measurement Capability

The SPC8106, ATT20C497 and DRAM chips have1 ohm l% resistors in series with their VDD pins for power consumption measurements.

Appendix A Parts List

Item #	Qty/ Board	Designation	Part Value	Description
1	5	C1, C21, C22, C23, C44	10uF	10uF / 15V Tantalum D-SIZE
2	10	C2-8, C20, C41, C43	0.1uF	0.1uF
3	4	C9-12	6.8pF	6.8pF
4	1	C13	1.0uF	1.0uF
5	3	C14-16	10uF / 63V	Electrolytic / Radial (LXF63VB10RM5X11LL)
6	3	C17-19	56uF/35V	LXF35VB56RM6X11LL
7	18	C24-40, C42	0.01uF	0.01uF 1206 pckg
8	1	D1	LM385BZ-1.2	PTH Zener Diode
9	6	D2-7	1N4148	Signal Diode / PTH
10	4	H1-H4	CON36A	0.1" 2x18 Male Header
11	4	JP1-4	.1 x 3 Male Header	PTH (include 2 pin jumper (shunt))
12	1	J5	PS/2 CONNECTOR	Assman A-HDF 15 A KG/T
13	3	J6	CON40A	Shrouded Header 40 pin Dual-row center-key / eject levers
14	4	L1-4	Ferrite Bead	Fair-rite 2743001111
15	1	L4	1uH	Dale Inductor IM-4-1.0uH
16	1	Q1	2N3905	PNP Signal Transistor
17	1	Q2	2N3903	NPN Signal Transistor
18	7	R1, R26, R47, R57, R58, R65, R66	1	1 OHM/ PTH /1%
19	9	R2-8, R16-17	10K	10K OHM/1206/5%
20	3	R9, R49, R56	1K	1K OHM/1206/5%
21	7	R10-12, R18-21	39	39 OHM /1206 / 5%
22	3	R13-15	150	150 OHM / 1206 / 5%
23	19	R22-24, R29-44	15K	15K OHM/1206/5%
24	1	R25	182	182 OHM/PTH/1%
25	2	R27-28	2M	2M OHM/1206/5%
26	3	R45, R46, R48	4.7K	4.7K OHM / PTH / 5%
27	2	R50, R51	100K	100K OHM/1206/5%
28	1	R55	100K	100K OHM Trim POT
29	1	R52	470K	470K OHM/1206/5%
30	1	R53	200K	200K OHM Trim POT
31	1	R54	14K	14K OHM / PTH / .25W / 1%
32	2	R59, R62	240	240 OHM/1206/5%
33	1	R60	100	100 OHM/1206/5%
34	2	R61, R63	500	500 OHM Trim POT

Item #	Qty/ Board	Designation	Part Value	Description
35	1	R64	470	470 OHM/1206/5%
36	2	S1, S2	SW-DIP-8	Switch Dip 8 position
37	1	U1	SPC8106	socketted
38	1	U2	M5M44170AJ-8	256Kx16 DRAM
			M5M44260AJ-8	256x16 DRAM
39	1	U3	ATT20C497	ATT20C497-50 PLCC pckg
40	1	U4	AMD27F256-150PC	AMD Flash EPROM
41	1	U5	74LS245	74LS245
42	1	U6	PAL16L8BCN	PAL16L8BCN
43	1	U7	M5M44260TSOP-8	256x16 DRAM
44	1	U8	74LS374	74LS374
45	1	U9	DIP SHUNT	16 pin DIP SHUNT
46	1	U10	RD-0412	XENTECK - Positive Power Supply
47	1	U11	EPN001	XENTECK - Negative Power Supply
48	2	U12, U13	LM317T	LM317T Adjustable Voltage Regulator
49	1	Y3	25.175MHz	MA-306 SMD pckg
50	1	Y4	28.322MHz	MA-306 SMD pckg
51	1	Y1	socket	16 pin DIP socket only
52	1	Y2	socket	16 pin DIP socket only

Appendix B SDU8106B0E Rev. 1.0 Schematic Diagrams

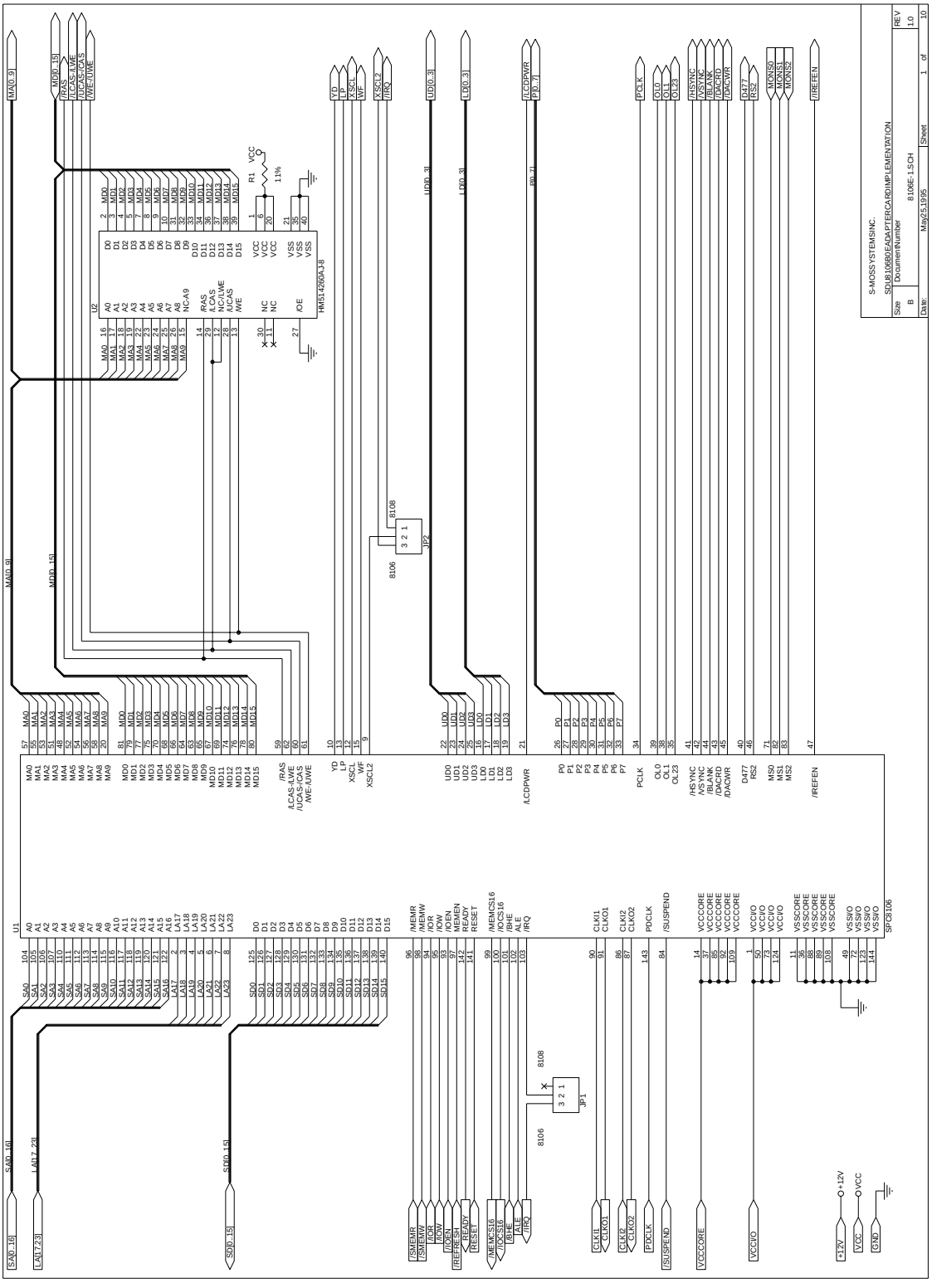


Figure 1 : SDU8106B0E Rev. 1.0 Schematic Diagram (1 of 10)

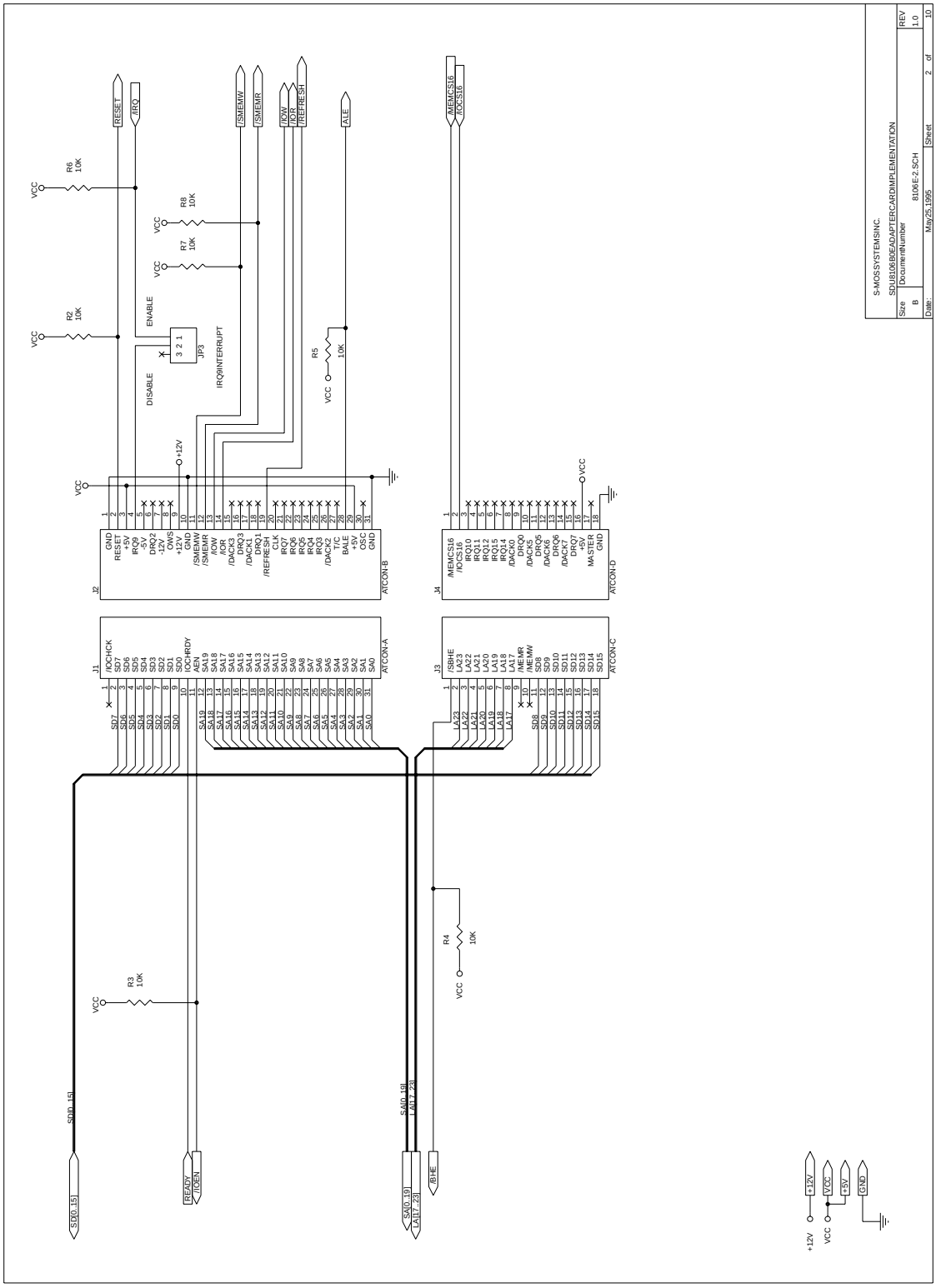


Figure 2 : SDU8106B0E Rev. 1.0 Schematic Diagram (2 of 10)

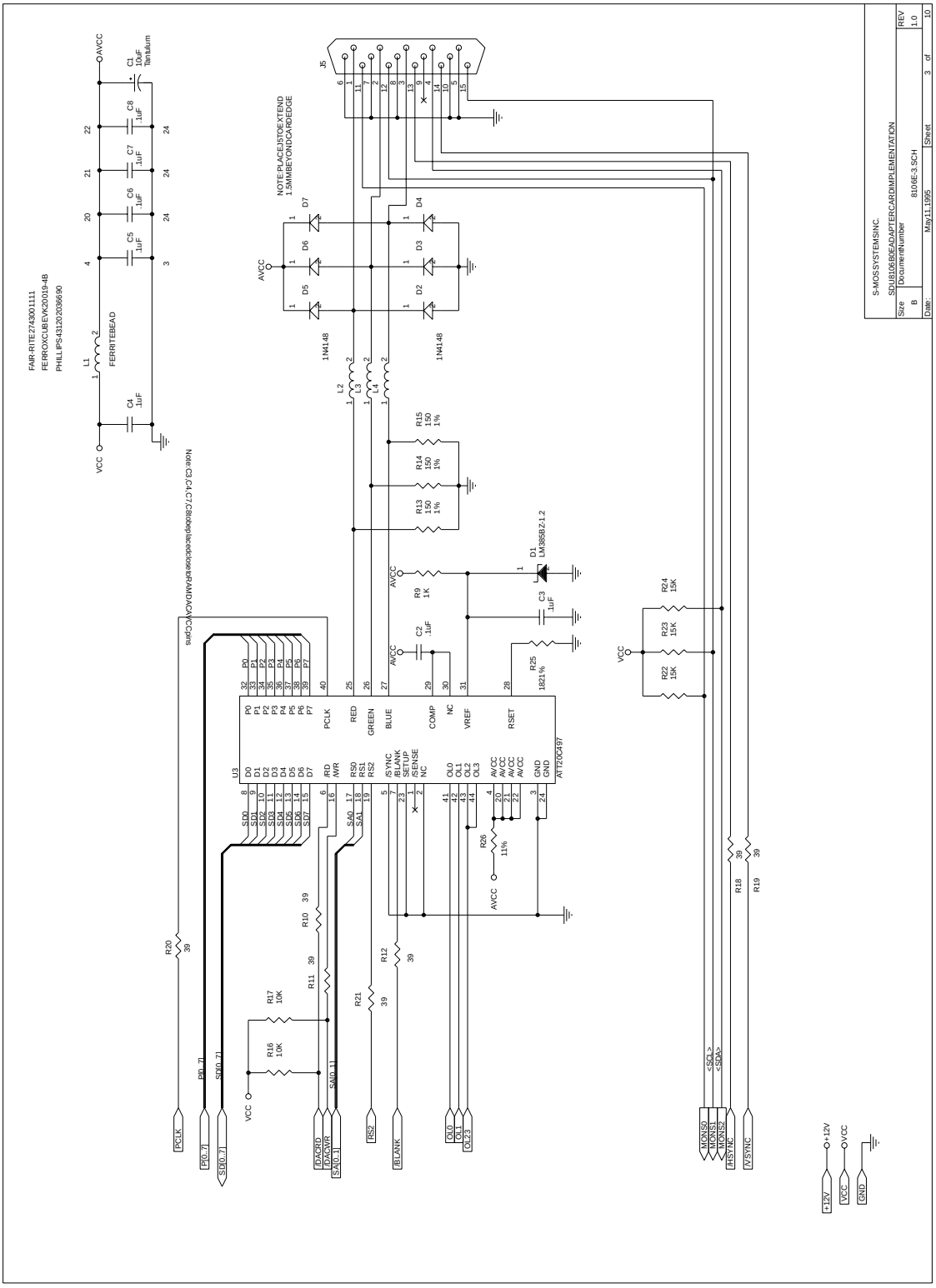


Figure 3 : SDU8106B0E Rev. 1.0 Schematic Diagram (3 of 10)

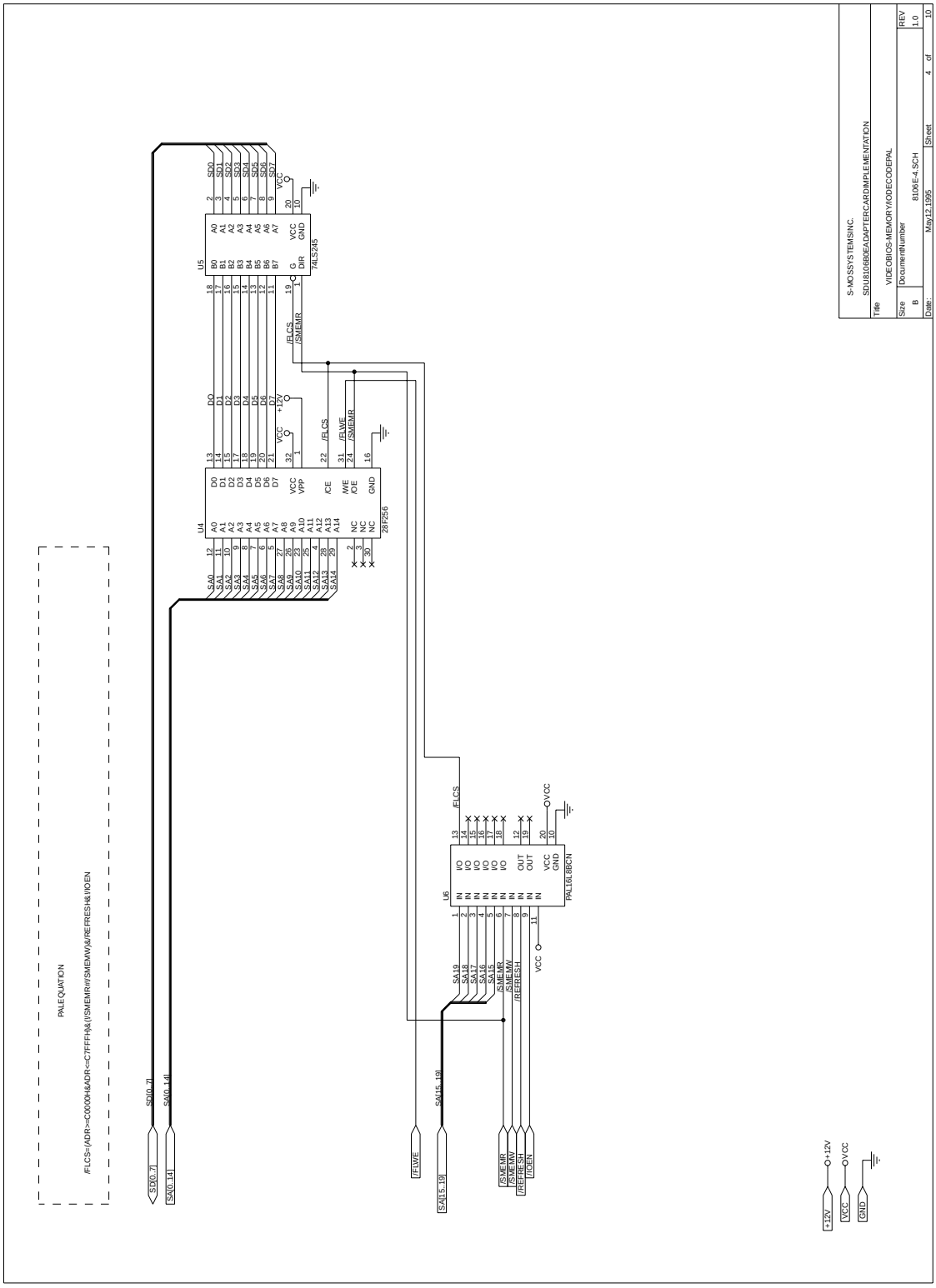


Figure 4 : SDU8106B0E Rev. 1.0 Schematic Diagram (4 of 10)

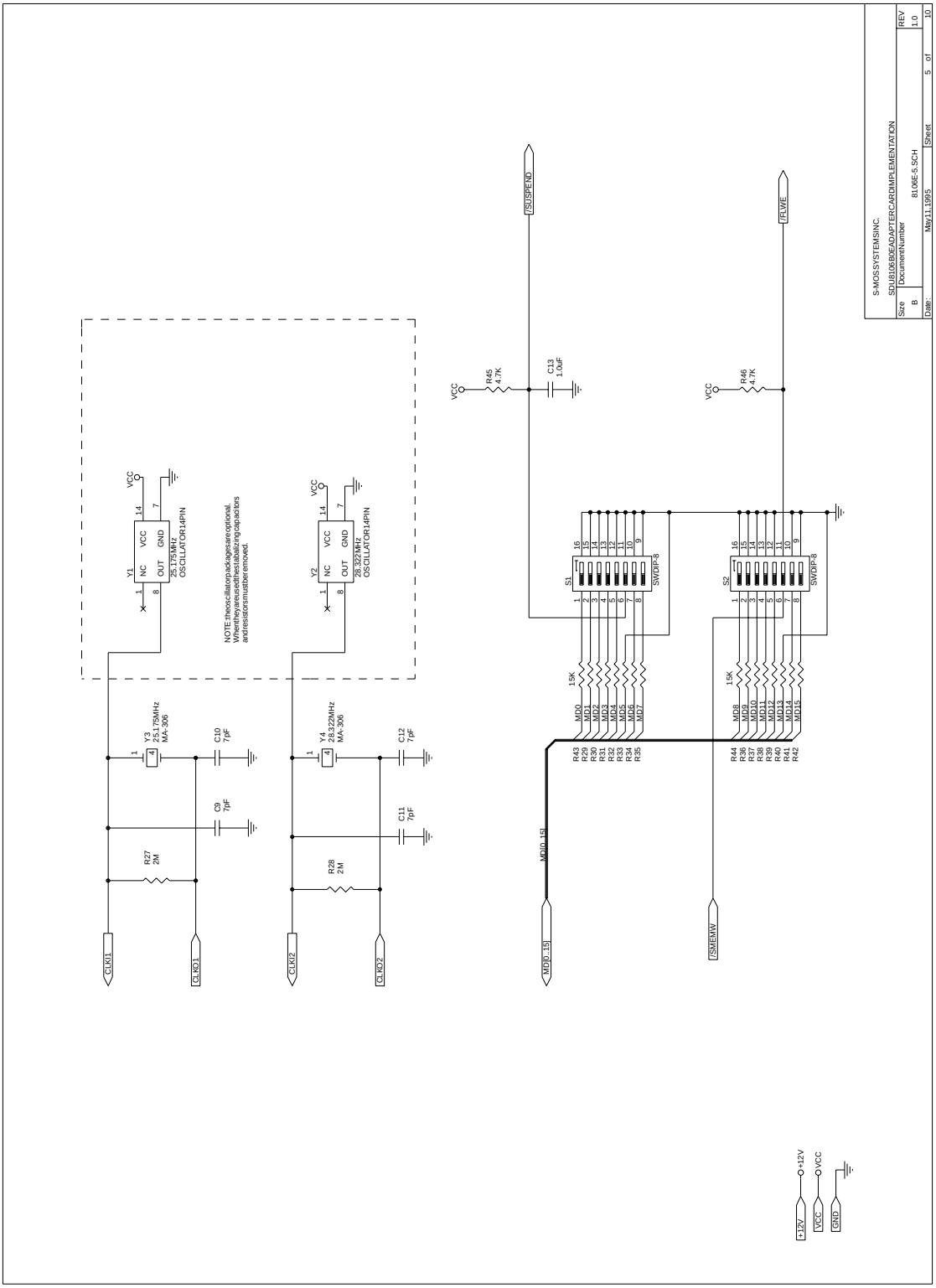


Figure 5 : SDU8106B0E Rev. 1.0 Schematic Diagram (5 of 10)

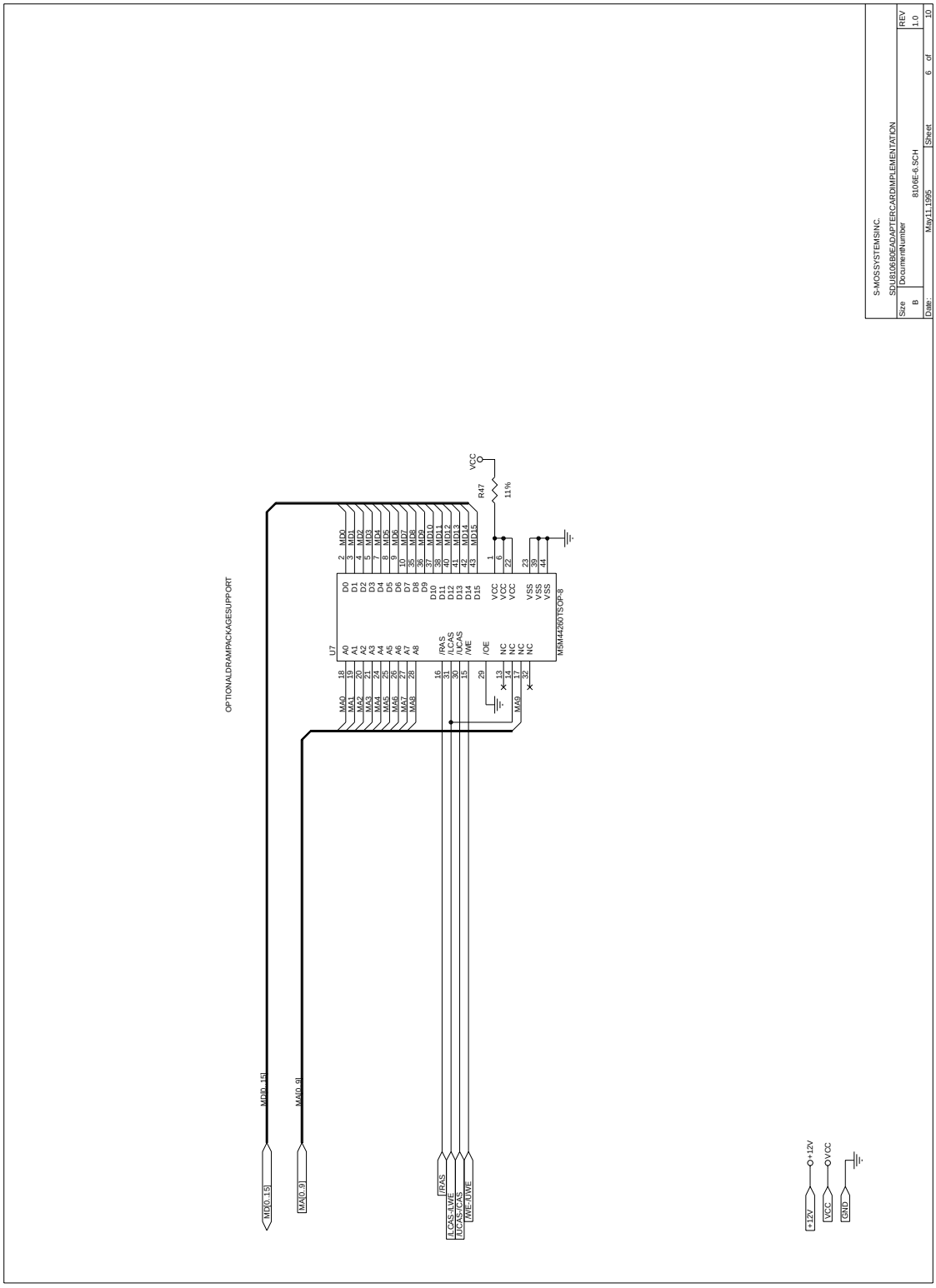


Figure 6 : SDU8106B0E Rev. 1.0 Schematic Diagram (6 of 10)

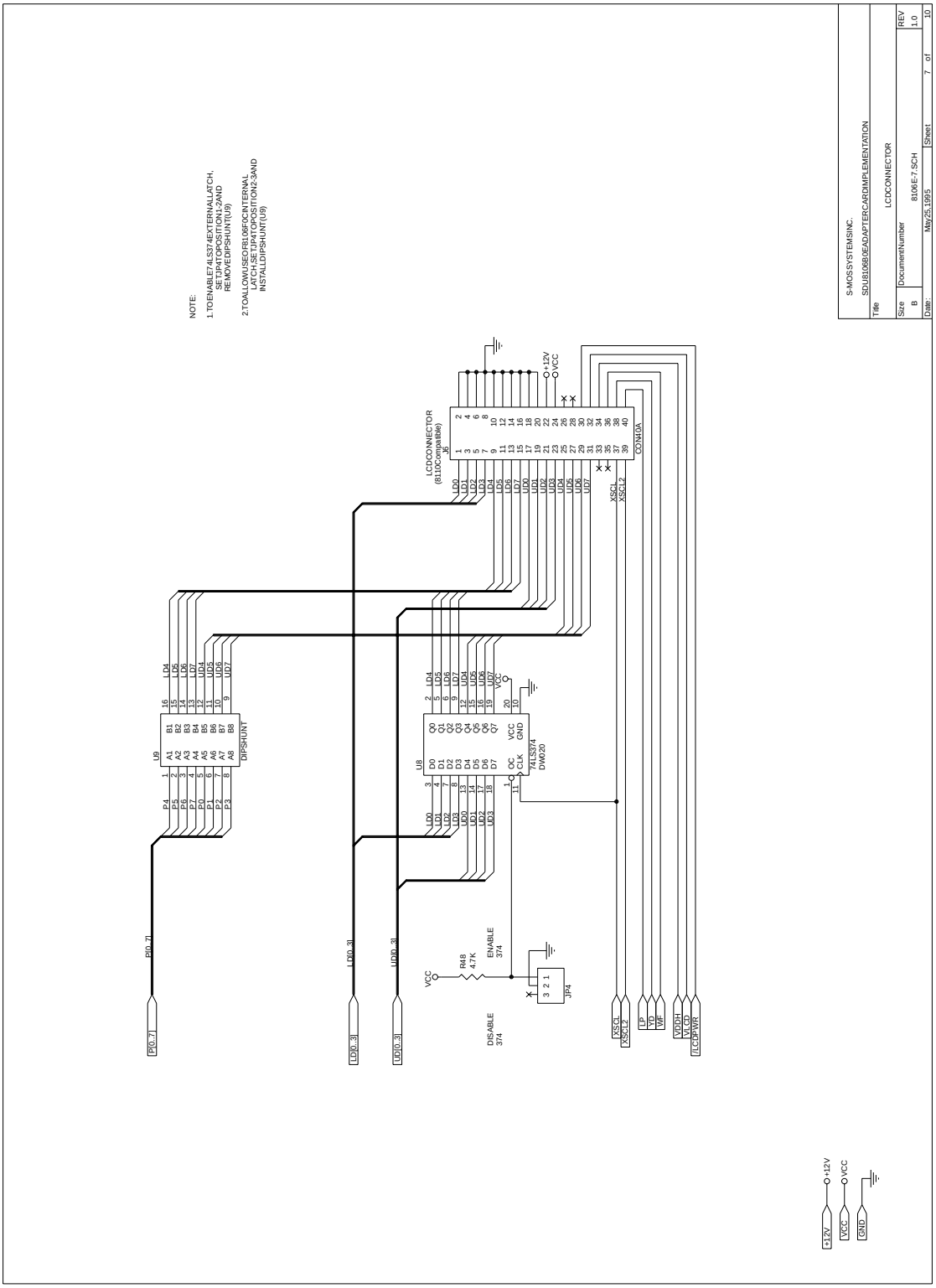


Figure 7 : SDU8106B0E Rev. 1.0 Schematic Diagram (7 of 10)

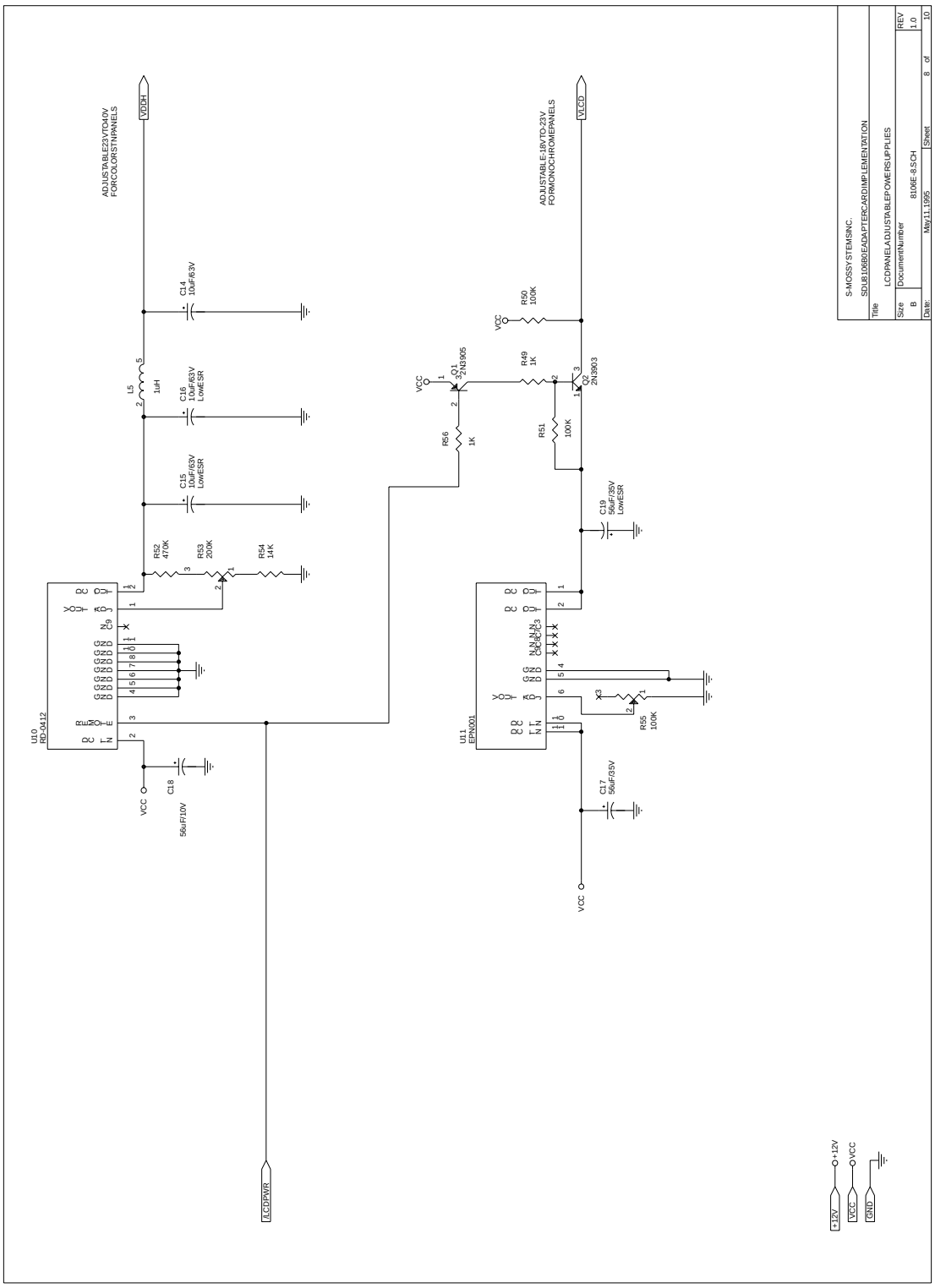


Figure 8 : SDU8106B0E Rev. 1.0 Schematic Diagram (8 of 10)

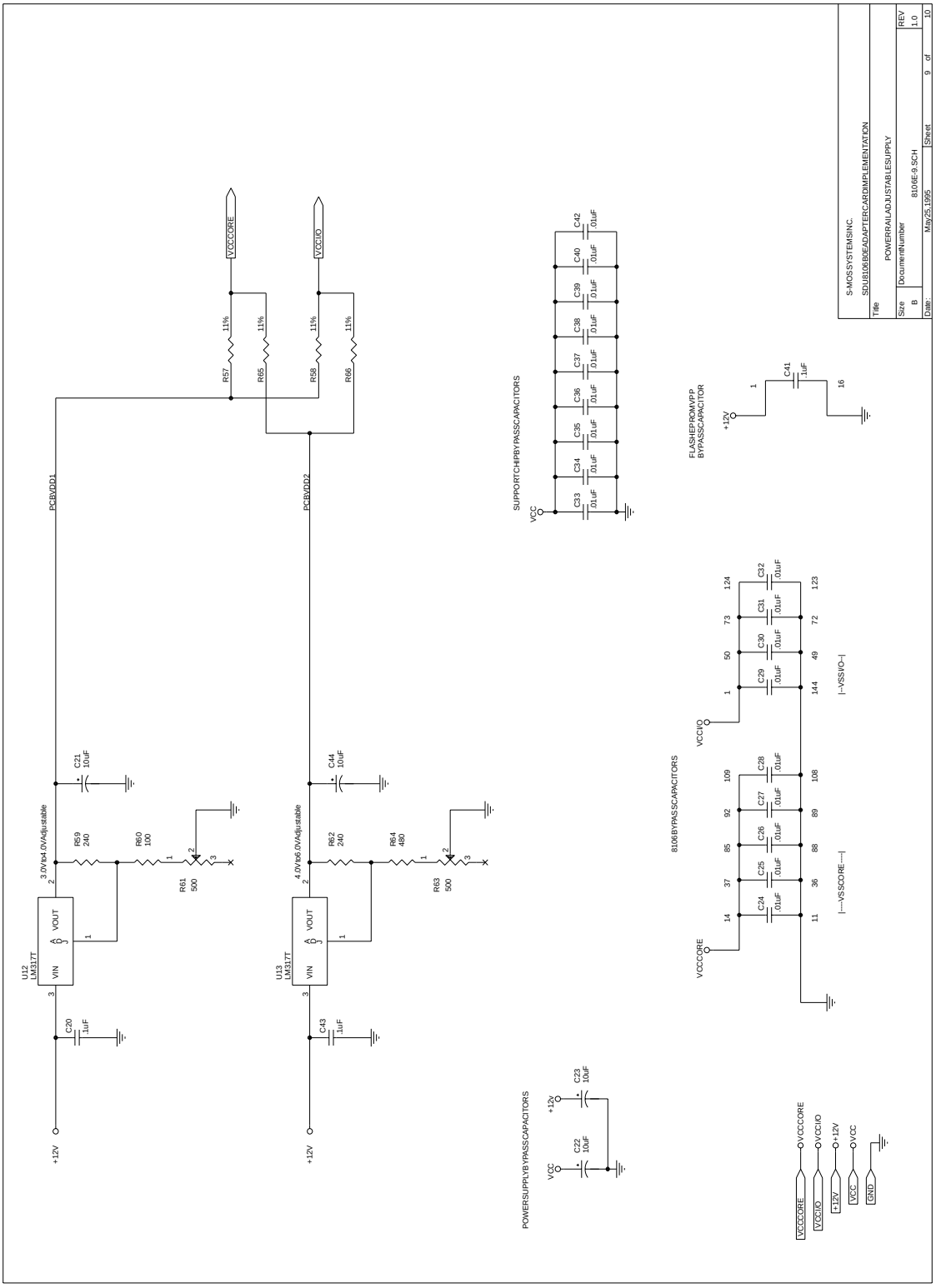


Figure 9 : SDU8106B0E Rev. 1.0 Schematic Diagram (9 of 10)

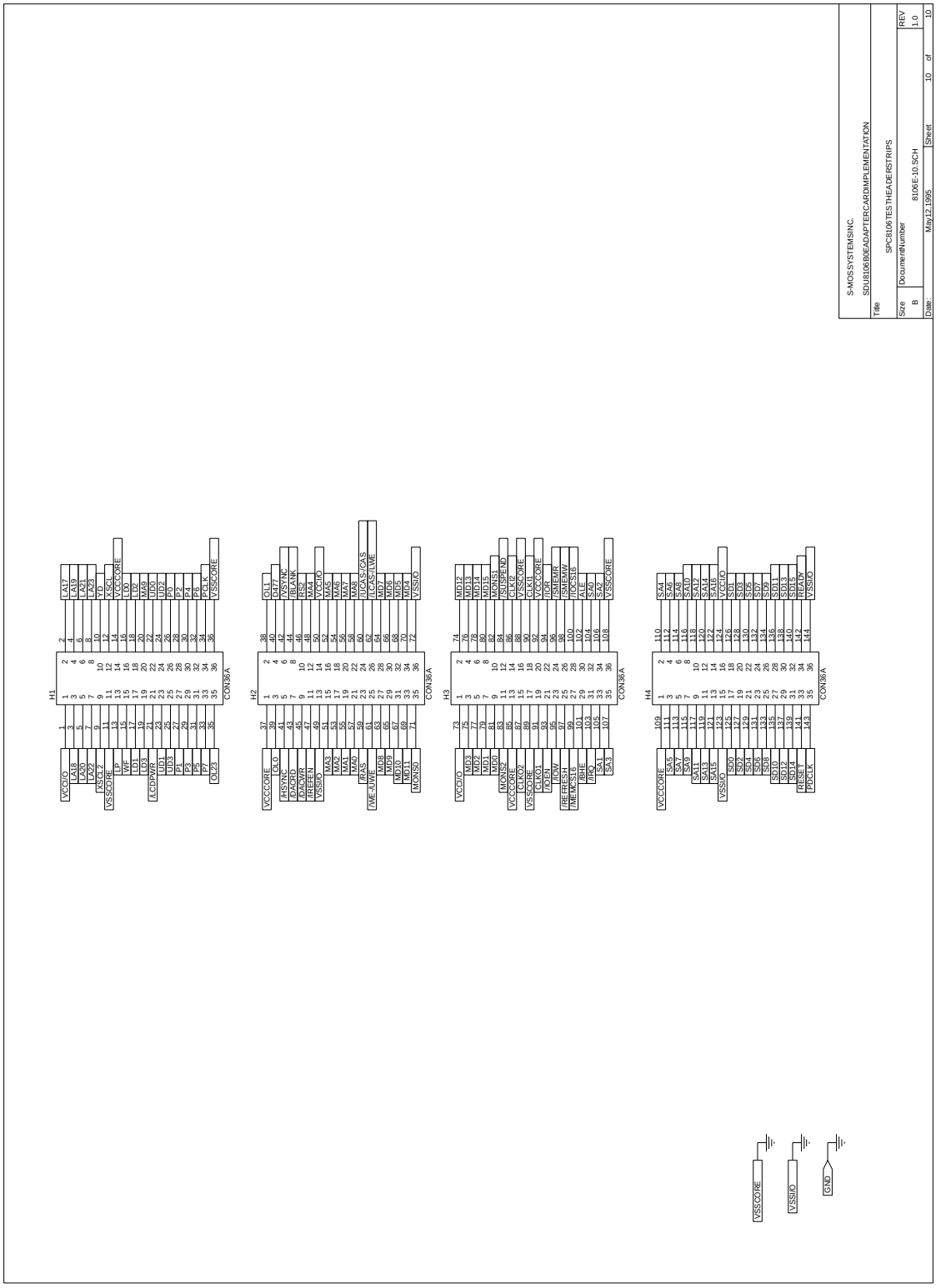


Figure 10 : SDU8106B0E Rev. 1.0 Schematic Diagram (10 of 10)

S-MOS SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

SDU8106B0F Rev. 1.0 Evaluation Board User Manual

Drawing Office No. X12-AN-005-02

THIS PAGE INTENTIONALLY BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY BLANK

1.0 DOCUMENT TITLE VARIABLE

This manual reflects the use of the SDU8106B0F Rev 1.0 Evaluation Board with the SPC8106 LCD/CRT VGA Controller Chip. All appropriate components are surface-mount to reduce cost and minimize board size. The SDU8106B0F will operate as stand-alone video adapter card with on-board video BIOS support. Refer to the SPC8106 Functional Specification, Drawing Office No. X12-SP-001-xx for further details.

1.1 Features

- 144 pin QFP-17 package
- 3.3V VCC Core and 5.0V VCC I/O operation
- 16-bit ISA Bus operation
- SMD technology for all appropriate devices
- Support for 4/8/16-bit monochrome and color STN LCD panels; single panel, single drive; dual panel, dual drive
- Support for 9/12- bit active matrix color TFT displays
- CRT support provided using external CMOS RAMDAC, ATT20C497
- 512K DRAM video memory (1x256Kx16 DRAM , SMD, SOJ Package, symmetrical architecture)
- Two terminal crystal support (28.322MHz and 25.175MHz SMD packages)
- On-board Video BIOS
- Support for Software Power Save Modes
- Support for Sprite/Hardware Cursor on both LCD and CRT displays
- Support for VESA DDC2B monitor standard
- On-Board fixed 3.3V VCC Core power supply
- On-Board +35 volt adjustable power supply (VDDH for LCD panels requiring a positive BIAS voltage)
- On-Board -23 volt adjustable power supply (VLCD for LCD panels requiring a negative BIAS voltage)

1.2 Display Modes Supported

LCD 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
13	Graphics	N/A	N/A	320 x 200	640 x 400	64	256	A000
100	Graphics	N/A	N/A	640 x 400	640 x 400	64	256	A000
101	Graphics	N/A	N/A	640 x 480	640 x 480	64	256	A000
108	Text	8 x 8	80 x 60	640 x 480	640 x 480	16	16	B800

CRT Display Modes

Mode No.	Mode Type	Font	Characters	Resolution	Displayed Pixels	Colors	Memory Segment
0	Text	8 x 8	40 x 25	320 x 200	640x400	16	B800
0+	Text	8 x 14	40 x 25	320 x 350	640x350	16	B800
0++	Text	9 x 16	40 x 25	360 x 400	720x400	16	B800
1	Text	8 x 8	40 x 25	320 x 200	640x400	16	B800
1+	Text	8 x 14	40 x 25	320 x 350	640x350	16	B800
1++	Text	9 x 16	40 x 25	360 x 400	720x400	16	B800
2	Text	8 x 8	80 x 25	640 x 200	640x400	16	B800
2+	Text	8 x 14	80 x 25	640 x 350	640x350	16	B800
2++	Text	9 x 16	80 x 25	720 x 400	640x400	16	B800
3	Text	8 x 8	80 x 25	640 x 200	640x400	16	B800
3+	Text	8 x 14	80 x 25	640 x 350	640x350	16	B800
3++	Text	9 x 16	80 x 25	720 x 400	640x400	16	B800
4	Graphics	N/A	N/A	320 x 200	640x400	4	B800
5	Graphics	N/A	N/A	320 x 200	640x400	4	B800
6	Graphics	N/A	N/A	640 x 200	640x400	2	B800
7	Text	8 x 14	80 x 25	640 x 350	640x350	2	B000
7+	Text	9 x 16	80 x 25	720 x 400	720x400	2	B000
0D	Graphics	N/A	N/A	320 x 200	640x400	16	A000
0E	Graphics	N/A	N/A	640 x 200	640x400	16	A000
0F	Graphics	N/A	N/A	640 x 350	640x350	2	A000
10	Graphics	N/A	N/A	640 x 350	640x350	16	A000
11	Graphics	N/A	N/A	640 x 480	640x480	2	A000
12	Graphics	N/A	N/A	640 x 480	640x480	16	A000
13	Graphics	N/A	N/A	320 x 200	640x400	256	A000
100	Graphics	N/A	N/A	640 x 400	640x400	256	A000
101	Graphics	N/A	N/A	640 x 480	640 x 480	256	A000
108	Text	8 x 8	80 x 60	640 x 480	640 x 480	16	B800

1.3 DoubleScan Support

DoubleScan is a feature in the SPC8106 that allows simultaneous display of both the CRT and LCD panel. Refer to the following table for supported video modes and limitations.

1. DoubleScan is not supported when using a Dual panel / Dual drive LCD.
2. All non-480 line modes (when using a 480 line panel) will show a wrap-around effect if DoubleScan is enabled. For example: display a 400 line mode on a 480 line panel and enable DoubleScan. You would see the top 78 lines duplicated on the bottom 78 lines (wrap-around affect).
3. If using a panel with less than 480 vertical lines, DoubleScan will be supported, however, the maximum LCD frame-rate may be violated. Therefore, the specific panel should be referenced.
4. If supporting a TFT panel requiring CRT-like timing (AUX[00] bit 5=1 and AUX[0B] bit 1=1) the panel handles the 350, 400 and 480 line modes, providing screen positioning internally. As the result of this direct support, DoubleScan is supported for all standard VGA and some extended modes.
5. If supporting a TFT panel requiring LCD-like timing (AUX[00] bit 5=1 and AUX[0B] bit 1=0) the DoubleScan mode is not supported.

Mode No.	Single Panel (640x480)	TFT - Color 9/12-bit (640x480) AUX[00] bit 5=1 AUX[0B] bit 1=1
0	No	Yes
0+	No	Yes
0++	No	Yes
1	No	Yes
1+	No	Yes
1++	No	Yes
2	No	Yes
2+	No	Yes
2++	No	Yes
3	No	Yes
3+	No	Yes
3++	No	Yes
4	No	Yes
5	No	Yes
6	No	Yes
7	No	Yes
7+	No	Yes
0D	No	Yes
0E	No	Yes
0F	No	Yes
10	No	Yes
11	Yes	Yes
12	Yes	Yes
13	No	Yes
100	No	Yes
101	Yes	Yes
108	Yes	Yes

2.0 INSTALLATION AND CONFIGURATION

The SPC8106 has 16 configurable inputs (MD0-MD15) which are read on power-up. For the purpose of this design, most of the configuration inputs have been hard-wired and are set to power-up as appropriate. Due to the variety of LCD panels supported, a five position DIP switch and three 3-position Jumpers have been provided for panel configuration. Refer to Jumper Configuration and LCD Panel DIP Switch Configuration below for details.

2.1 Jumper Configuration

The following table shows the jumper selections available for this board.

Table 0-1 SDU8106B0F Configuration Jumper Settings

Jumper	1 - 2	2 - 3
JP1	SPC8106 pin 39 configured as OL0 for CRT sprite support	SPC8106 pin 39 configured as data bit B0 for 12-bit TFT panel
JP2	SPC8106 pin 38 configured as OL1 for CRT sprite support	SPC8106 pin 38 configured as data bit G0 for 12-bit TFT panel
JP3	SPC8106 pin 35 configured as OL23 for CRT sprite support	SPC8106 pin 35 configured as data bit R0 for 12-bit TFT panel

2.2 DIP Switch Configuration

The SDU8106B0F DIP switch inputs are mapped directly from the SPC8106 configuration inputs MD0 - MD3, and MD9. The BIOS configuration for different LCD panels is handled through the state of the MD lines. The following tables describes the settings and associated panel type configuration.

Table 0-2 SDU8106B0F Configuration DIP Switch Settings

Switch	Signal	Open	Closed
SW1-1	MD0	LCD Panel Configuration bit 0	see Table 2-4
SW1-2	MD1	LCD Panel Configuration bit 1	
SW1-3	MD2	LCD Panel Configuration bit 2	
SW1-4	MD3	LCD Panel Configuration bit 3	
SW1-5	MD9	LCD Panel Configuration bit 4	

Table 0-3 SDU8106B0F Hard-Wired Configuration Inputs

	Open	Closed
MD4	16-bit I/O port	8-bit I/O port
MD5	A[19:2] latched internally by ALE	Standard ISA bus ALE - A[16:0] not latched
MD6	Symmetrical Video DRAM	Asymmetrical Video DRAM
MD7	16-bit LCD interface supported via external latch	16-bit LCD interface supported directly by SPC8106
MD8	5.0 volt core voltage	3.3 volt core voltage
MD10	256 cycle / 32 ms DRAM refresh	256 cycle / 4 ms DRAM refresh
MD11	PSM DRAM refresh source Bit 0	see Table 2-5
MD12	PSM DRAM refresh source Bit 1	
MD13	Pins 38, 39 used for external RC for 32 kHz PDCLK	Pins 38, 39 used for OL[1:0]
MD14	PSM DRAM refresh source Bit 2	see Table 2-5
MD15	3C3h Video Enable Port	46E8h and 102h Video Enable Port

= Hard-Wired Board Configuration

See “SPC8106 Power Save Mode DRAM Refresh Source Select” on page 12 for hard-wired configuration of MD11, MD12 and MD14.

MD0-MD3 & MD9 LCD Panel Size Configuration

These bits are used to indicate the LCD panel size and type to the video BIOS software as shown in the following table.

Table 0-4 SPC8106 LCD Panel Size Configuration

S1-5 MD9	S1-4 MD3	S1-3 MD2	S1-2 MD1	S1-1 MD0	Resolution	Mode	Bits	Comments
1	1	1	1	1	640x480	Dual	8	Monochrome
1	1	1	1	0	640x480	Single	8	Monochrome
1	1	1	0	1	640x400	Dual	8	Monochrome
1	1	1	0	0	640x400	Single	8	Monochrome
1	1	0	1	1				Internal use only
1	1	0	1	0	640x200	Single	4	Monochrome
1	1	0	0	1	480x320	Dual	8	Monochrome
1	1	0	0	0	480x320	Single	4	Monochrome
1	0	1	1	1				Internal use only
1	0	1	1	0	320x256	Single	4	Monochrome
1	0	1	0	1				Internal use only
1	0	1	0	0	320x240	Single	4	Monochrome
1	0	0	1	1				Internal use only
1	0	0	1	0	320x200	Single	4	Monochrome
1	0	0	0	1				For OEM Configuration
1	0	0	0	0	640x480	Single	12	TFT(CRT Mode)
0	1	1	1	1	640x480	Dual	8	Color STN
0	1	1	1	0	640x480	Single	8	Color STN
0	1	1	0	1	640x480	Dual	16	Color STN
0	1	1	0	0	640x480	Single	16	Color STN
0	1	0	1	1	640x480	Dual	9	RGBI MODE
0	1	0	1	0	640x480	Single	9	TFT (CRT Mode)
0	1	0	0	1	640x480	Dual	12	RGB MODE
0	1	0	0	0	320x240	Single	4	Color STN
0	0	1	1	1	640x480	Single	12	TFT(LCD Mode)
0	0	1	1	0	640x480	Single	9	TFT(LCD Mode)

0 = closed, 1 = open

MD4 ISA-Bus Port Width

The SDU8106B0F Evaluation Board is a 16-bit ISA-Bus I/O card. MD4 configuration input selects between 8-bit and 16-bit ISA-Bus operation. MD4 input is left open to support 16-bit ISA-Bus operation. For the purpose of this design 8-bit ISA-Bus operation is not supported.

MD5 Configuration Input

The MD5 input selects configuration between standard ISA bus interface and a Modified Address Timing. The MD5 input is set to a logic low on power-up to select standard ISA bus interface. For the purpose of this design, the Modified Address Timing is not supported.

MD6 and MD10 Video DRAM Configuration

The SDU8106B0F is assembled with one 5V, 256Kx16 configuration, symmetrical architecture (9 rows x 9 columns; 2 / CAS signals), SMD, SOJ package DRAM. For this design, MD6 input is left open to support symmetrical architecture DRAM and MD10 input is left open to support 256 cycle / 32 ms refresh rate DRAM .

MD7 16-Bit LCD Interface Support

Configuration input MD7 selects the method for supporting a 16-bit LCD panel. The SPC8106 will not support a CRT while directly supporting the 16-bit LCD panel using the internal interface. For the purpose this design, MD7 is left open to support 16-bit LCD panel interface through an external latch, thus at the same time providing full CRT support. See “SDU8106B0F Rev. 1.0 Schematics (6 of 7)” on page 24 for further details.

MD8 Core Operating Voltage

The SPC8106 can either operate with 5.0V or 3.3V core operating voltage which is configurable through MD8 input. For the purpose of this design, MD8 is set to logic low to support only 3.3V VCC core operating voltage.

MD11 - MD12 and MD14 Power Save Mode DRAM Refresh Source

During Hardware Power Save Mode (and Power Save Mode 4), provision must be made to refresh the video DRAM. As shown in the following table, there are four options. Option one uses the frequency source connected to CLK1. Option two uses the ISA /REFRESH signal (which is connected to SPC8106 pin MEMEN). Option three is for the case when the DRAM supports self-refresh capability. Option four selects a 64 kHz signal connected to SPC8106 pin PDCLK. Self-refresh DRAM is used for this design. Therefore, option three is used to refresh the video DRAM. For this design, MD11 input is set to logic low and MD12, MD14 inputs are left open according to the table below.)

Table 0-5 SPC8106 Power Save Mode DRAM Refresh Source Select

MD14	MD12	MD11	Option	Refresh Source
1	0	0	1	CLK1
1	0	1	2	MEMEN (ISA / REFRESH
1	1	0	3	Self Refresh DRAM
1	1	1	5	PDCLK (64 kHz)

0 = closed, 1 = open

= hard-wired configuration

MD13 Configuration Input

The MD13 configuration input selects between external 32 kHz PD clock support and CRT spite capabilities. The MD13 input is set to a logic low on power-up to select CRT spite capabilities. For the purpose of this design, the external 32 kHz PD clock is not supported.

MD15 Video Enable Port Address

The SPC8106 has the capability of enabling the video system through port address 3C3h or through 46E8h and 102h and is selectable via MD15 configuration input. This capability is provided for when it is desired to have more than one video system installed in a PC. For the purpose of this design, MD15 is left open to select 3C3h port address.

2.3 Connecting LCD Panels

The LCD Signals are found on the 40-pin LCD connector(J6). Refer to the following table for signal assignments.

Table 0-6 SDU8106B0F LCD Connector (J6) Pinout

SPC8106 Pin Name	LCD Con. Pin No.	TFT color			Color STN LCD			Mono STN LCD	
		9-bit	12-bit AUX[00]b5=1 AUX[0B]b1=0	12-bit AUX[00]b5=1 AUX[0B]b1=1	16-bit	8-bit	4-bit	8-bit	4-bit
LD0	1	R0	R1	R1	LD0	LD0		LD0	
LD1	3	G0	G1	G1	LD1	LD1		LD1	
LD2	5	G1	G2	G2	LD2	LD2		LD2	
LD3	7	G2	G3	G3	LD3	LD3		LD3	
LD4	9				LD4				
LD5	11				LD5				
LD6	13				LD6				
LD7	15				LD7				
UD0	17	R1	R2	R2	UD0	UD0	UD0	UD0	UD0
UD1	19	B0	B1	B1	UD1	UD1	UD1	UD1	UD1
UD2	21	B1	B2	B2	UD2	UD2	UD2	UD2	UD2
UD3	23	B2	B3	B3	UD3	UD3	UD3	UD3	UD3
UD4	25				UD4				
UD5	27				UD5				
UD6	29				UD6				
UD7	31				UD7				
HSYNC#	33			HS					
VSYNC#	35			VS					
XSCL	37	CLK	CLK	CLK	XSCL	XSCL		XSCL	XSCL
XSCL2	39	R2	R3	R3		XSCL2	XSCL2		
LP	40	HS	HS		LP	LP	LP	LP	LP
YD	38	VS	VS		YD	YD	YD	YD	YD
WF	36	DE	DE	DE	WF	WF	WF	WF	WF
VDDH	34				VDDH	VDDH	VDDH	VDDH	VDDH
VLCD	32							VLCD	VLCD
LCDPWR#	30	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#	LCDPWR#
OL23	28	N/C	R0	R0	N/C	N/C	N/C	N/C	N/C
OL0	26	N/C	B0	B0	N/C	N/C	N/C	N/C	N/C
+5 V	24	+5 V	+5 V	+5 V	+5 V	+5 V	+5 V	+5 V	+5 V
+12 V	22	+12 V	+12 V	+12 V	+12 V	+12 V	+12 V	+12 V	+12 V
OL1	20	N/C	G0	G0	N/C	N/C	N/C	N/C	N/C
GND	18-2	GND	GND	GND	GND	GND	GND	GND	GND

The SPC8106 pins OL0, OL1 and OL23 have multiple functions. They are configured as data bits B0, G0 and R0 respectively in 12-bit TFT panel support mode.

3.0 TECHNICAL DESCRIPTION

3.1 ISA Bus Interface

This board directly supports the 16-bit ISA Bus via a standard AT edge connector. For the purpose of this design 8-bit ISA operation will not be supported.

3.2 512K DRAM Support

The SPC8106 supports both symmetrical and asymmetrical DRAMs. A single chip on-board 512K DRAM solution is supported using one 5V, 256K x 16 configuration, symmetrical architecture, SMD SOJ package DRAM. For the purpose of this design, no other DRAM configuration will be supported.

3.3 Monochrome LCD Support

The SPC8106 supports 4- and 8-bit single and dual scan monochrome STN LCD panels. All the necessary signals are provided on the 40-pin LCD connector (J6). The interface signals are alternated with grounds on the cable to reduce crosstalk and noise related problems.
Refer to “SDU8106B0F LCD Connector (J6) Pinout” on page 13 for your specific requirements.

3.4 Color LCD Panels Support

The SPC8106 supports 4-, 8-, and 16-bit single and dual scan color STN LCD panels. All the necessary signals can be found on the 40-pin LCD connector (J6). The interface signals are alternated with grounds on the cable to reduce crosstalk and noise related problems. External circuitry provides 16-bit color panel support by latching the 8 bits of output data from the SPC8106 to provide 16 bits of data on the next clock. Refer to the SPC8106 Functional Specification, Drawing Office No. X12-SP-001-xx for further details.
Refer to “SDU8106B0F LCD Connector (J6) Pinout” on page 13 for your specific requirements.

3.5 Color TFT panels Support

The SPC8106 also supports 9- and 12-bit active matrix color TFT panels. All the necessary signals can also be found on the 40-pin LCD connector (J6). While supporting 12-bit TFT panel, Sprite support on the CRT is not possible as the overlay pins OL0, OL1, and OL23 of SPC8106 are used as TFT data pins. Refer to the SPC8106 Functional Specification, Drawing Office No. X12-SP-001-xx for further details.
Refer to “SDU8106B0F LCD Connector (J6) Pinout” on page 13 for your specific requirements.

3.6 External CMOS RAMDAC Support

This evaluation board provides CRT support with the addition of an external RAMDAC (ATT20C497 or equivalent). The presence of an external RAMDAC/CRT can be determined by software on power-up. The default display is CRT if attached.
The ATT20C497 RAMDAC is provided on the board to fully test all of the CRT display modes available with the SPC8106 including sprite/hardware cursor display.
When a software power save mode is selected, the ATT20C497 RAMDAC is placed into “sleep mode” (refer to AT&T ATT20C497 specification for complete details on “sleep mode”). This is done by programming the ATT20C497 RAMDAC Command Register. The on-board BIOS provides this feature automatically.

The RAMDAC requires a voltage reference to provide the necessary DAC output gain. This voltage reference can be provided internally or externally. Using the external voltage reference provides a more accurate, more stable, and less noisy voltage source. When the RAMDAC is in Sleep Mode, the Vref pin is disabled and does not use any power. Although the SPC8106 does provide a control pin, /IREFEN, it is not required by this design.
The ATT20C497 has been provided with an external voltage reference.

3.7 DDC2B Support

The SDU8106B0F evaluation board is capable of directly supporting the VESA DDC2B monitor standard. This board supports hardware DDC2B and the BIOS supports software layer VBE/DDC version 1.0. For further information on DDC2B contact the Video Electronics Standards Asociation (VESA).

3.8 VGA BIOS Support

A 32K EPROM contains the VGA BIOS. Support logic consists of an Address Decoder (C000h-C7FFh) and a Data Buffer. With the BIOS installed, the board will operate as a stand-alone video adapter.

3.9 Power Save Modes

The various power save modes can be controlled via software or hardware. A separate software utility is supplied to control power save modes. See PWRSAVE Power Save Utility, Drawing Office No. X12-UI-004-xx for more information.
Hardware SUSPEND mode is not supported in this design.

3.10 Adjustable LCD Panel Positive Power Supply(VDDH)

Most color STN LCD panels require a positive power supply VDDH between +23V and +40V (I_{out}=45 mA). For ease of implementation, such a power supply has been provided as an integral part of this design. The VDDH voltage can be found on LCD connector(J6) at pin 34. The VDDH voltage can be varied by R43(200K potentiometer) and controlled by LCDPWR# signal.
R43 can be adjusted to provide an output voltage between +23V and +40V.
LCDPWR# is a control signal provided by the SPC8106 to enable/disable the VDDH power supply. This signal provides correct power sequencing and prevents panel damage.
Adjust VDDH voltage to the appropriate voltage required by the panel before connecting the panel.

3.11 Adjustable LCD Panel Negative Power Supply(VLCD)

Most monochrome STN LCD panels require a negative power supply VLCD between -18V and -23V (Iout=45 mA). For ease of implementation, such a power supply has been provided as an integral part of this design. The VLCD voltage can be found on LCD connector(J6) at pin 32. The VLCD voltage can be varied R45(100K potentiometer) and controlled by LCDPWR# signal.
R45 can be adjusted to provide an output voltage between -18V and -23V.
LCDPWR# is a control signal provided by the SPC8106 to enable/disable the VLCD power supply. This signal provides correct power sequencing and prevents panel damage.
Adjust VLCD to the appropriate voltage required by the panel before connecting the panel.

3.12 Fixed 3.3 Volt Power Supply

The SPC8106 VCC Core voltage can either be 3.3V or 5.0 V (as selected by MD8). To provide 3.3V VCC Core voltage for SPC8106, an independent fixed 3.3V power supply at 0.5A (maximum) has been provided as an integral part of this design using National LP2960AIN-3.3 Micropower 0.5A Low-Dropout Voltage Regulator. For the purpose of this design, 5V VCC Core voltage operation will not be supported

3.13 PCB Layout Considerations

This board is constructed of four layers to accomodate various power/ground requirements. There are no traces on the mid-layers of the PCB board. As well there are no traces crossing the clock signals to avoid noise from the clock lines.
In this layout the 40-pin LCD connector is located near the SPC8106 chip to reduce capacitive loading and provide noise protection to the signal lines. The CRT connector is available at the outside rear edge of the board (facing the back of the computer).

Appendix A Parts List

Item #	Qty/ Board	Designation	Part Value	Description
1	4	C1,C2,C22, C37	10uF	10uF / 25V Tantalum D-SIZE
2	22	C3-19,C23-27	0.01uF	0.01uF, 1206 pckg
3	3	C20,C21,C38	0.1uF	0.1uF, 1206 pckg
4	4	C28-31	6.8pF	6.8pF, 1206 pckg
5	3	C32-C34	10uF / 63V	Electrolytic / Radial (LXF63VB10RM5X11LL)
6	3	C35,C36,C40	56uF/35V	LXF35VB56RM6X11LL
7	1	C39	33uF	33uF / 10V Tantalum D-SIZE
8	1	D1	LM385BZ-1.2	PTH Zener Diode, 0.1" spc., 2 pin TO-92 pckg
9	6	D2-7	1N4148	Signal Diode / PTH
10	3	JP1-3	.1 x 3 Male Header	PTH (include 2 pin jumper (shunt))
11	1	J5	PS/2 Connector	Assman A-HDF 15 A KG/T or equivalent
12	1	J6	CON40A	Shrouded Header 40 pin Dual-row center-key, PTH
13	4	L1-4	Ferrite Bead	Fair-rite 2743001111, PTH
14	1	L5	1uH	Dale Inductor IM-4-1.0uH, PTH
15	1	Q1	2N3905	PNP Signal Transistor, TO-92 PTH
16	1	Q2	2N3903	NPN Signal Transistor, TO-92 PTH
17	8	R1-6, R20-21	10K	10K OHM/1206/5%
18	1	R7	182	182 OHM/PTH/1%
19	3	R8,R39,R46	1K	1K OHM/1206/5%
20	7	R9-13,R22-23	39	39 OHM /1206 / 5%
21	3	R14-16	150	150 OHM / 1206 / 5%
22	12	R17-19,R24, R27-30, R33-34, R36-37	15K	15K OHM/1206/5%
23	2	R25,R26	2M	2M OHM/1206/5%
24	2	R40,R41	100K	100K OHM / PTH / 5%
25	1	R45	100K	100K OHM Trim POT Bourns 3386W-1-104 (or eqiuv)
26	1	R42	470K	470K OHM/1206/5%
27	1	R43	200K	200K OHM Trim POT Bourns 3386W-1-204 (or equiv)
28	1	R44	14K	14K OHM / PTH / .25W / 1%
29	1	S1	SW-DIP-5	Dip Switch, 5 position
30	1	U1	SPC8106	QFP17 package
31	1	U2	uPD42S4260LE-80	256Kx16 , Self-Refresh, Symmetrical DRAM
32	1	U3	ATT20C497	ATT20C497-50 PLCC pckg
33	1	U4	27C256	NM27C256Q-200 or equiv.
34	1	U5	74LS244	IC 74LS244 Buffer, DW020 pckg.
35	1	U6	74LS688	IC 74LS688 Comparator, DW020 pckg.
36	1	U7	74LS374	IC 74LS374 Latch, DW020 pckg.
37	1	U8	EPN001	XENTECK - Negative Power Supply
38	1	U9	RD-0412	XENTECK - Positive Power Supply

Item #	Qty/ Board	Designation	Part Value	Description
39	1	U10	LP2960AIN-3.3	National 3.3V Fixed Voltage Regulator N16G 16-pin DIP pckg.
40	1	Y1	25.175MHz	MA-306 SMD pckg
41	1	Y2	28.322MHz	MA-306 SMD pckg
42	4	R31,R32,R35, R38	15K	15K OHM/1206/5%

Appendix B SDU8106B0F Rev. 1.0 Schematic Diagrams

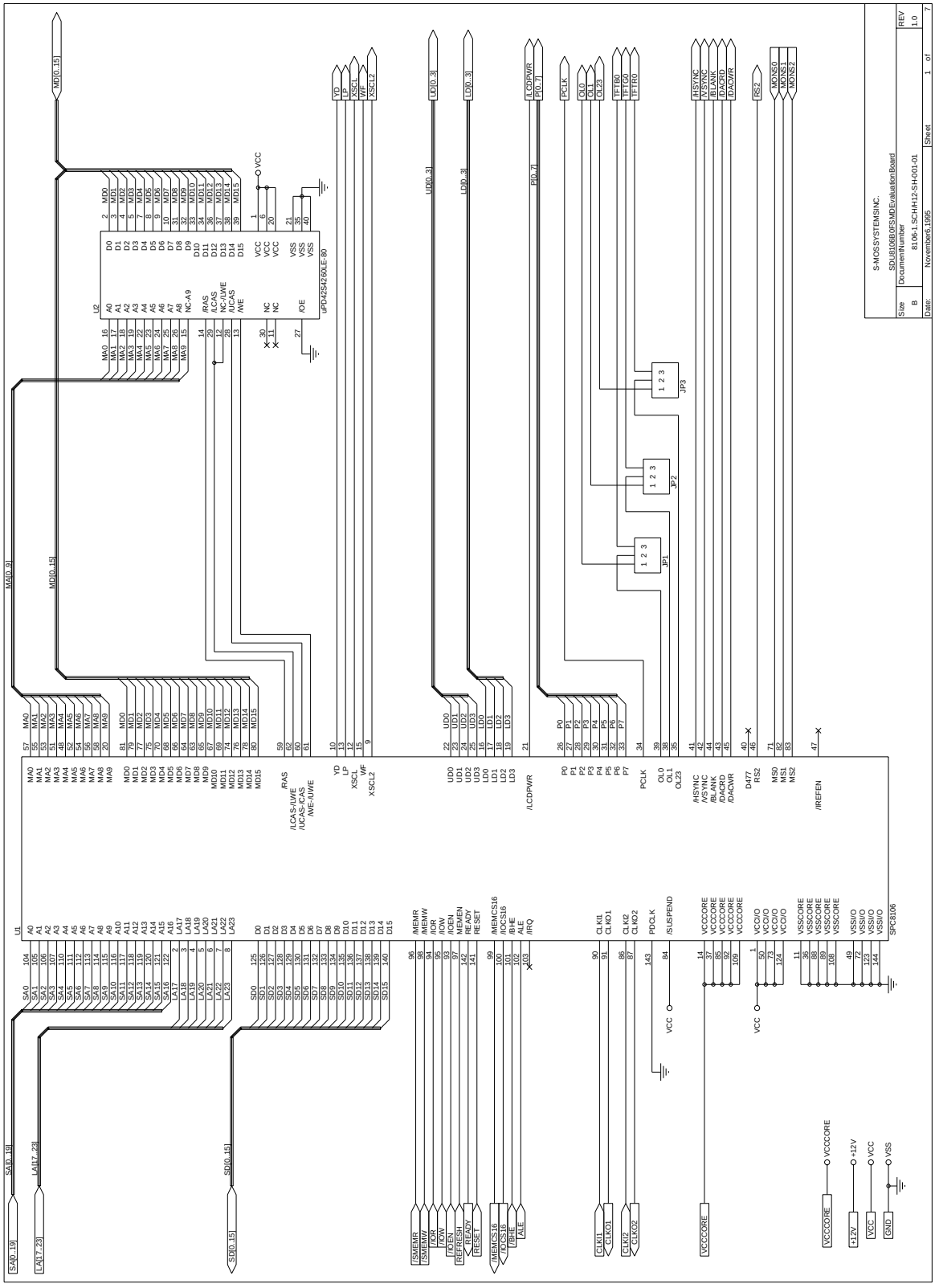


Figure 1 : SDU8106B0F Rev. 1.0 Schematics (1 of 7)

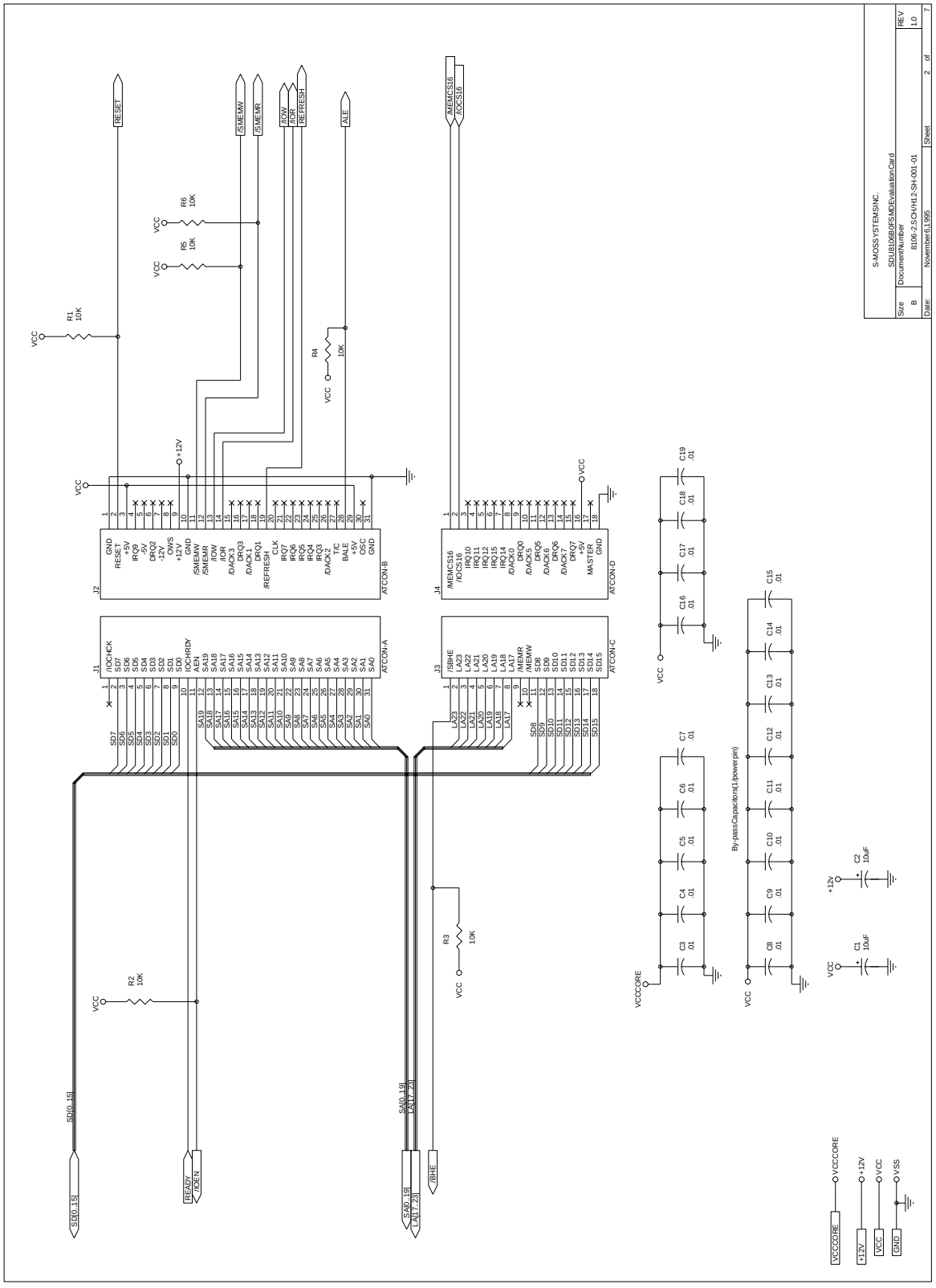


Figure 2 : SDU8106B0F Rev. 1.0 Schematics (2 of 7)

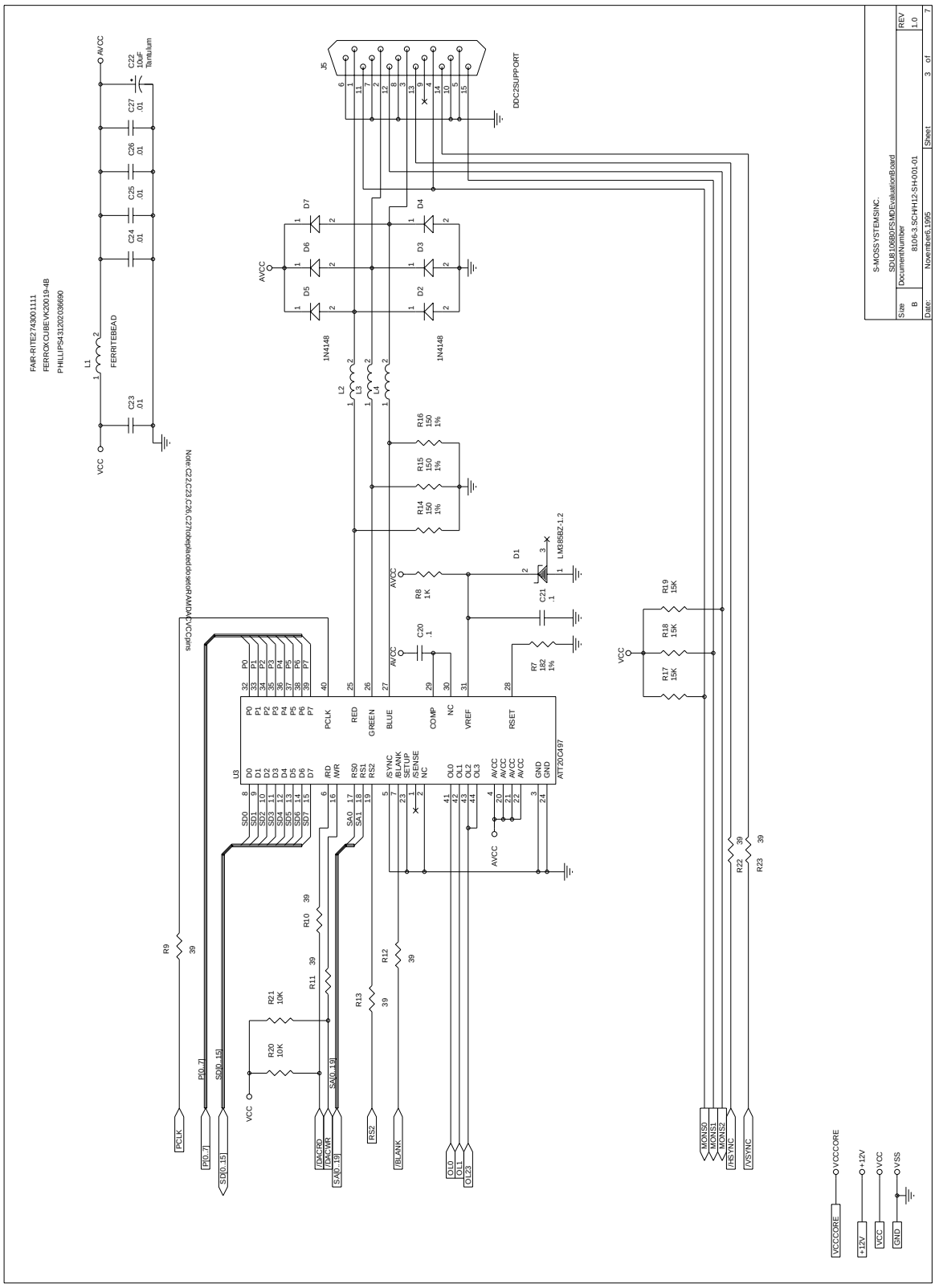


Figure 3 : SDU8106B0F Rev. 1.0 Schematics (3 of 7)

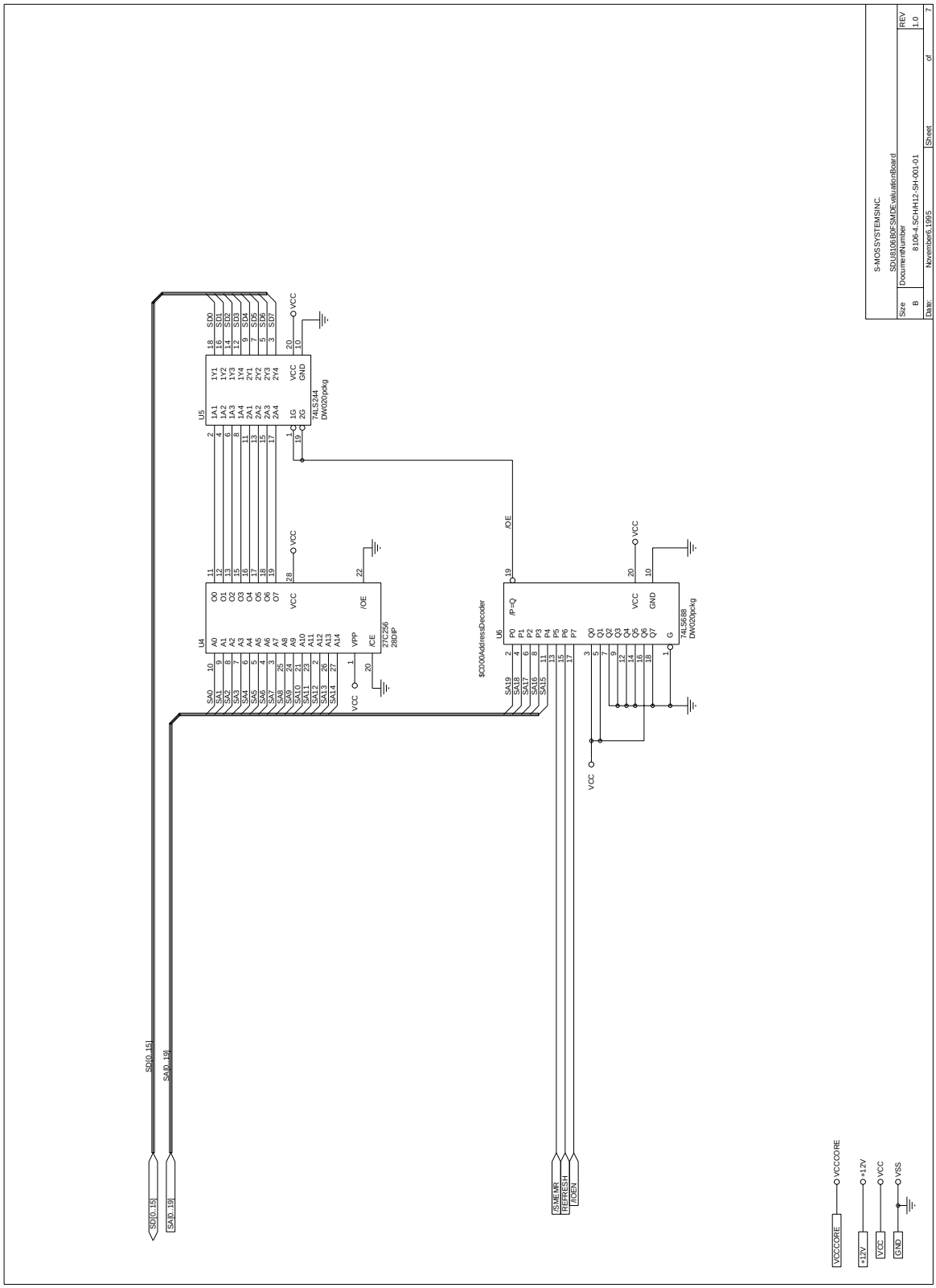


Figure 4 : SDU8106B0F Rev. 1.0 Schematics (4 of 7)

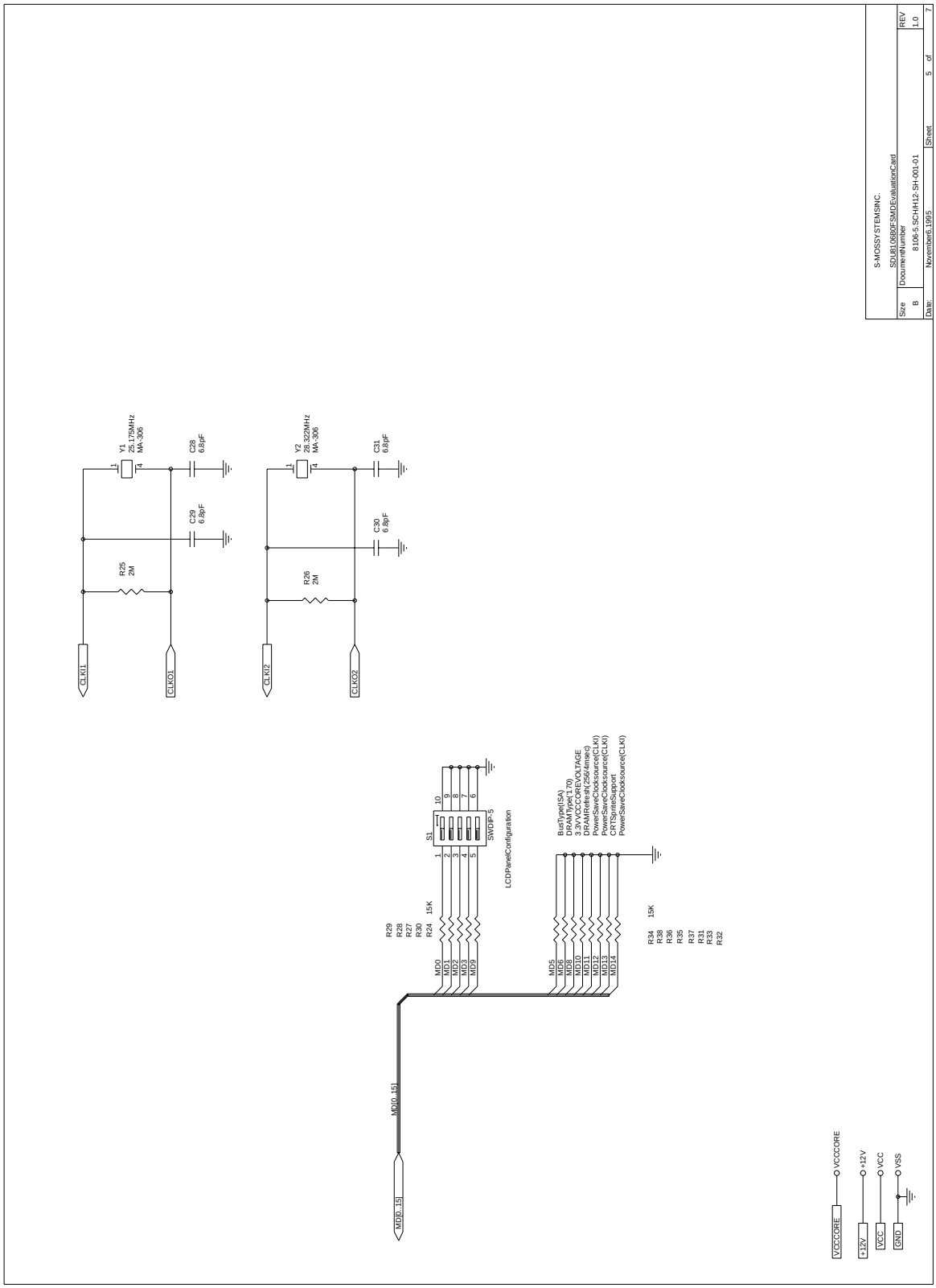


Figure 5 : SDU8106B0F Rev. 1.0 Schematics (5 of 7)

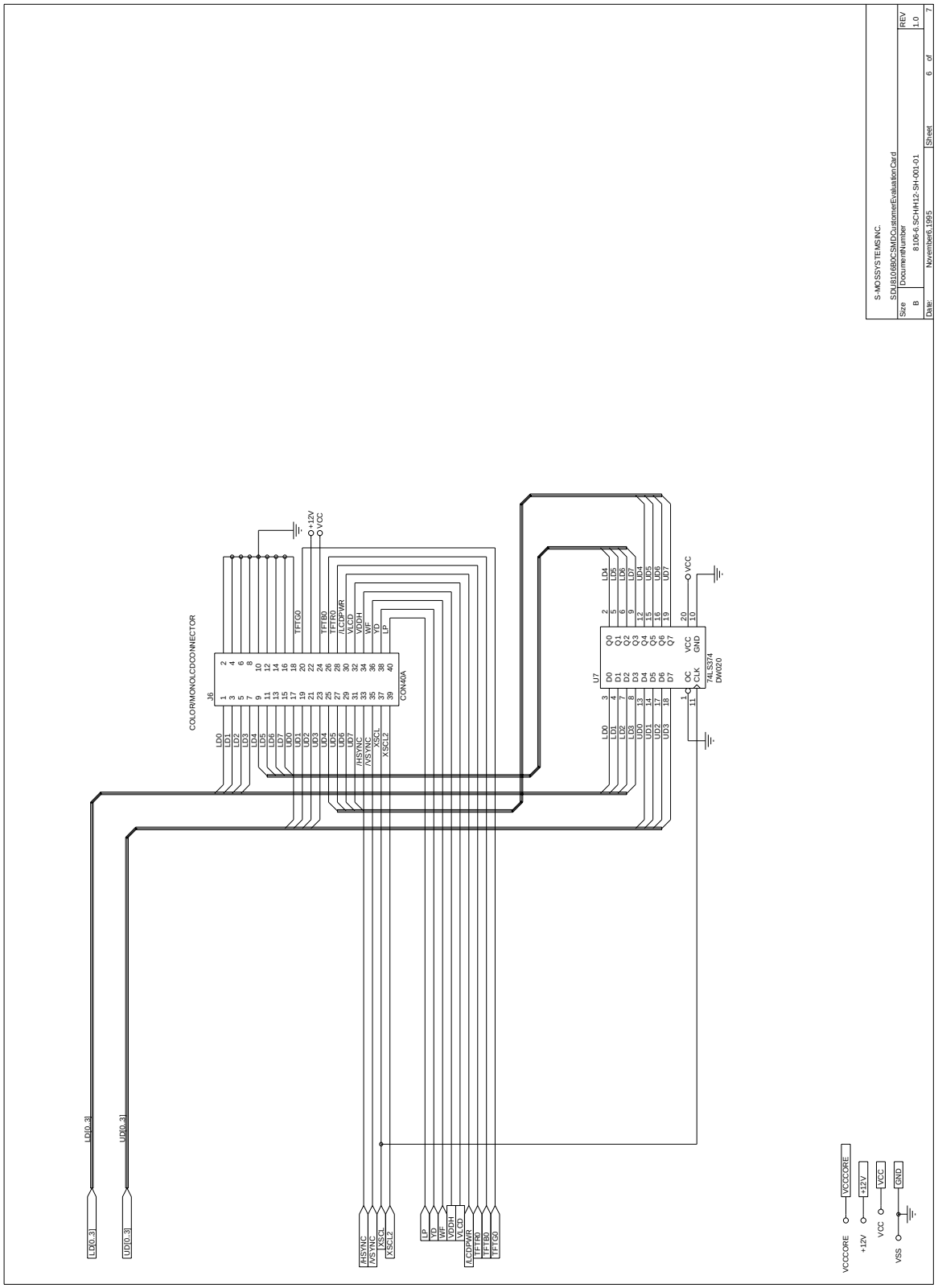


Figure 6 : SDU8106B0F Rev. 1.0 Schematics (6 of 7)

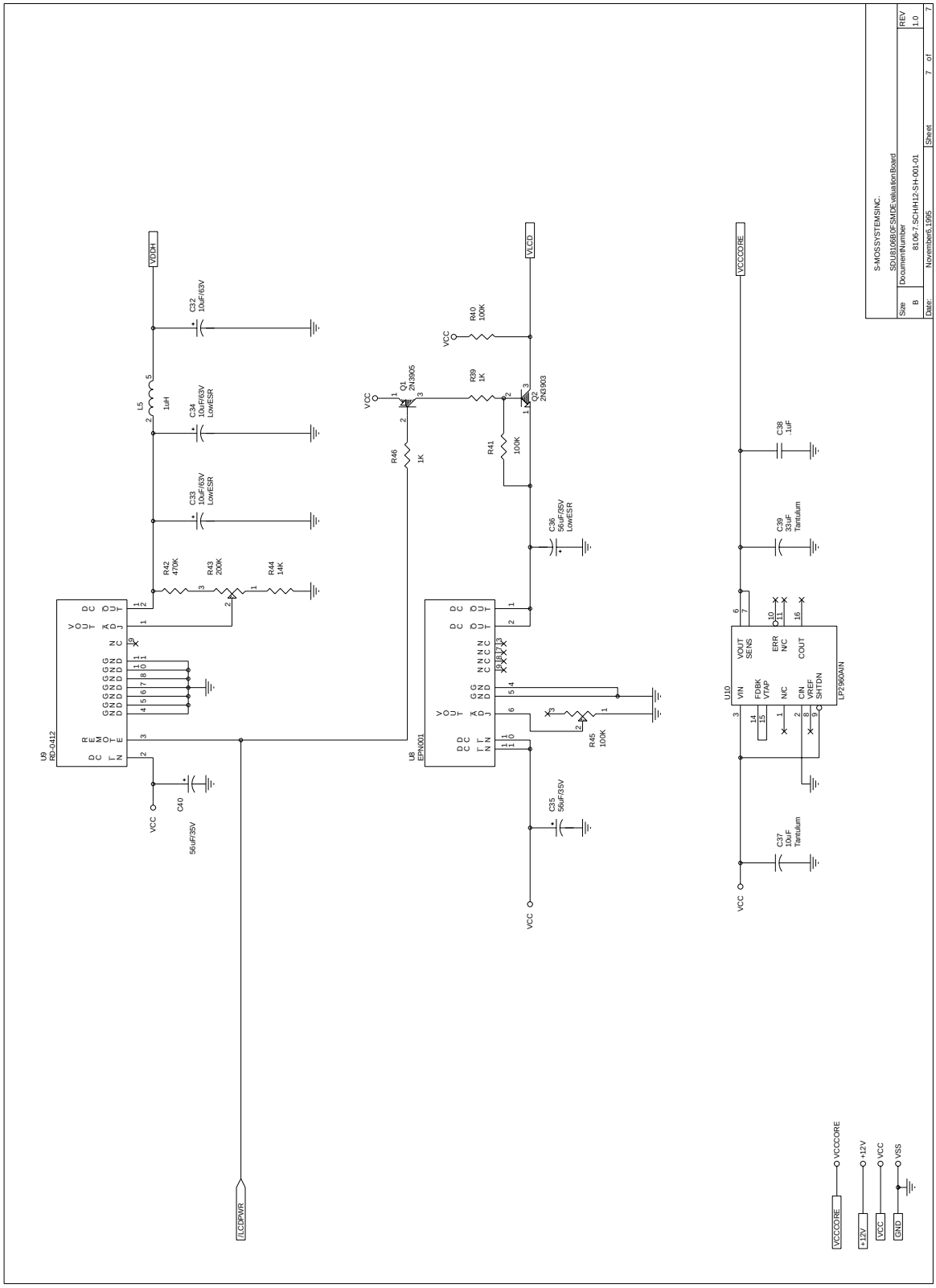


Figure 7 : SDU8106B0F Rev. 1.0 Schematics (7 of 7)

S-MOS
SYSTEMS
A Seiko Epson Affiliate

SPC8106 LCD/CRT VGA CONTROLLER

Power Consumption

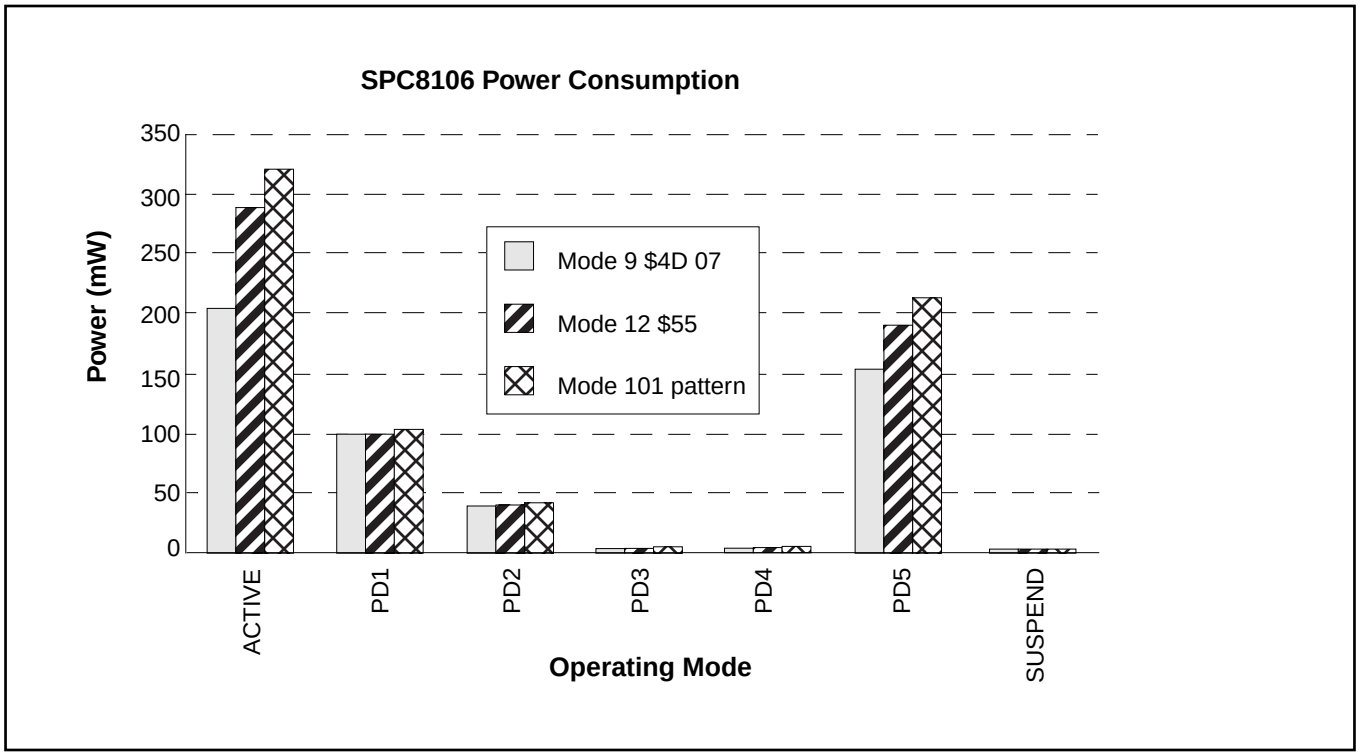
Drawing Office No. X12-AN-004-02.1

THIS PAGE INTENTIONALLY BLANK

1.0 SPC8106 POWER CONSUMPTION

1.1 Conditions

1. Mode 3: Screen filled with 4Dh 07h
2. Mode 12: Screen filled with 55h
3. Mode 101: Screen filled with pattern
4. Frequency: 28.322 MHz
5. V_{DD} (core): 3.3 volts
   V_{DD} (I/O): 5.0 volts
6. Controller operating in Single Panel Color LCD Display mode
7. No display connected



	Active	PD1	PD2	PD3	PD4	PD5	Suspend	Units
Mode 03	202.7	100.0	38.2	2.8	2.7	154.1	2.7	mW
Mode 12	287.5	100.1	38.3	2.8	2.8	190.1	2.7	mW
Mode 101	319.9	104.9	38.7	3.1	3.0	215.3	2.7	mW