Configuring the Élan™SC300 Device's Internal CGA Controller for a Specific LCD Panel

AMD

Application Note

This application note explains how to determine if a specific LCD panel is supported by the Élan™SC300 microcontroller, and in what modes (CGA two-color or four-color graphics, CGA text, HGA graphics, HGA text); how to configure the ÉlanSC300 microcontroller to work with the LCD panel in those modes; and which signals need to be connected to the ÉlanSC300 microcontroller to properly drive the panel.

INTRODUCTION

There are a number of factors that need to be understood to accomplish the tasks mentioned above. The sections in the first part of this document provide you with the understanding and background needed to follow the procedures beginning on page 7. You are also encouraged to read and have available for reference Chapter 3, "Video Controller" of the *Élan™SC300 Microcontroller Programmer's Reference Manual*, order #18470.

DEFINITIONS

The following terms are used throughout this document. Understanding these terms as they apply to the ÉlanSC300 microcontroller is essential for understanding how to configure the microcontroller's Video Index registers to work with various LCD panels.

Single-Screen Panels

A single-screen LCD panel consists of the Liquid Crystal Display (LCD) and a set of column drivers (also known as segment drivers) and row drivers (also called common drivers). These drivers supply the appropriate DC voltage to the liquid crystals in the display such that the crystals become aligned and block the light from passing through the display. An entire row of pixels is biased at one time. For this to occur, the pixel data for one row is loaded into the segment drivers. Then, a DC voltage is applied to the segment drivers and to the common driver that is attached to the row being displayed. This process is then repeated for each row of the display.

Figure 1 is a simplified illustration of an LCD panel that can be driven by the ÉlanSC300 microcontroller. Not all signals are shown (e.g., FRM and M), and not all connections of existing signals are shown. As depicted in Figure 1, the ÉlanSC300 microcontroller drives 4 bits of pixel data at a time. The data is clocked into the segment drivers using the shift clock. When all the data for a row of pixels is loaded into the segment drivers, the line clock will pulse, causing the common driver to activate the voltage to the "common line" for the row of pixels being displayed.

Dual-Screen Panels

A dual-screen LCD panel is two single-screen panels glued so tightly together that they appear to be one panel. Each panel has separate data bits but shares the same control signals: shift clock, line clock, and frame start (see the sections that follow). For example, a 640 x 200 dual-screen panel is actually two 640 x 100 single-screen panels glued together.

The only dual-screen panel resolution the ÉlanSC300 microcontroller supports is 640 x 200. More specific information as to how the ÉlanSC300 microcontroller handles a dual-screen panel is discussed later in this document.

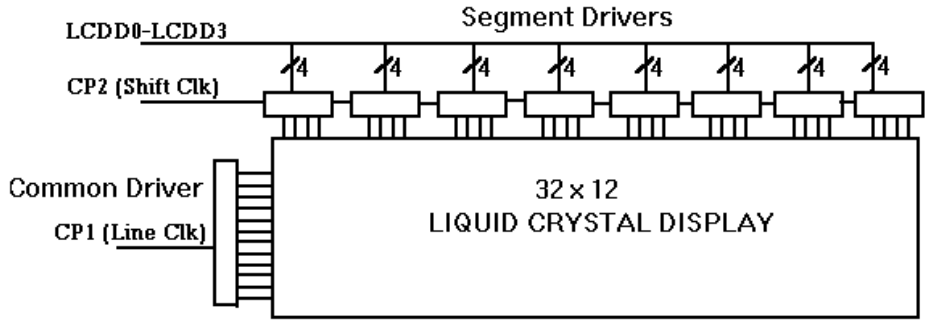

Figure 1. Simplified LCD Panel Configuration

Publication# **20749** Rev: **A** Amendment/**0**
Issue Date: **June 1997**

Pixel Clock

This is the standard unit of measure from which all of the LCD clocking signals can be derived. The pixel clock can be loosely defined as the rate at which pixels are displayed on the screen. On the ÉlanSC300 microcontroller, the pixel clock used is one of two fixed frequencies: 14.336 MHz for HGA mode or 9.557 MHz (which is 2/3 of 14.336) for CGA mode. On the ÉlanSC300 microcontroller, the pixel clock is internal to the video controller and is not available external to the chip.

Shift Clock (CP2)

The shift clock, known as the CP2 signal on the ÉlanSC300 microcontroller, tells the LCD panel when data is valid on the LCD data bus (the ÉlanSC300 microcontroller pins LCDD0–LCDD3 and LCDDL0–LCDDL3). Data is valid on the falling edge of shift clock.

The frequency of the shift clock can be determined by the pixel clock frequency divided by the number of data bits per panel. For both single- and dual-screen panels with the ÉlanSC300 microcontroller, the number of bits per panel is four. Therefore, the shift clock frequency will be either 14.336 MHz ÷ 4 data bits = 3.58 MHz, or 9.557 MHz ÷ 4 data bits = 2.389 MHz.

Line Clock (CP1)

The line clock, known as the CP1 signal on the ÉlanSC300 microcontroller, tells the LCD panel that a complete row (horizontal line) of pixel data has been sent to the LCD panel to be displayed. Additional shift clocks indicate valid data for the next row of pixels (i.e., display the current row and prepare to receive pixel data for the next row).

A line clock pulse will occur once for every row of pixel data sent to the LCD panel. For example, a 320 x 240 LCD screen has 240 rows; therefore, 240 line clocks will occur each time the panel is refreshed.

When CP1 is asserted on the ÉlanSC300 microcontroller's LCD controller, there is a period of about 20 pixel clock cycles during which CP2 will not be asserted. This has a slight effect on the refresh rate as described later in this document.

Frame Start (FRM)

Frame start, known as FRM on the ÉlanSC300 microcontroller, is asserted at the start of every frame (panel scan). This signal tells the LCD panel that the next data sent to it via the shift clock will be for the top row of the panel. This signal is sometimes referred to as the FLM (First Line Marker) signal.

On the ÉlanSC300 microcontroller's LCD controller, there is no "dead time" when FRM is asserted. The FRM assertion overlaps with the normal CP1 assertion at the end of the last line.

Refresh Rate

The refresh rate of the LCD panel is the number of times per second the screen is redrawn. This affects the crispness and brightness of the image on the display. On the ÉlanSC300 microcontroller, the refresh rate is affected by the screen resolution programmed into the video controller, the number of bits per pixel (BPP) (in Graphics mode only), and the pixel clock frequency.

On the ÉlanSC300 microcontroller, the refresh rate for Text mode or 2-color Graphics mode can be calculated by dividing the pixel clock frequency by the screen resolution programmed into the video controller, taking into account the 20 extra pixel clocks per line mentioned in the previous section. The refresh rate for 4-color Graphics mode will be half that amount. For example, if the controller is programmed for a 320 x 240 LCD screen and configured for CGA 2-color mode (which means that the 9.557 MHz pixel clock is used), then each line consists of 320 + 20 = 340 pixel clocks, and the refresh rate would be 9.557 MHz ÷ (340 · 240) = 117.1 Hz.

If the refresh rate is marginal for the LCD size in CGA Graphics mode, a 50% increase in refresh rate can be achieved by using HGA Graphics mode. On the ÉlanSC300 microcontroller, the selection of CGA or HGA mode can remain independent of the resolution. (This is discussed later in this document.)

DC Voltage Biasing

DC voltage biasing is a condition of LCD panels when the liquid crystals in the display become permanently aligned. Normally, the liquid crystals are nonaligned, allowing light to pass through the display. When a DC voltage is applied, the liquid crystals align themselves, blocking the light from passing through the display. The effect of this alignment causes a pixel to appear dark on the display. As a panel becomes DC voltage biased, the liquid crystals in the display become permanently aligned, eventually causing the entire display to appear black. AC modulation (defined in the next section) is used to prevent DC biasing.

LCD Panel AC Modulation (M)

All LCD panels implement what is known as AC modulation to prevent DC voltage biasing from damaging the display. AC modulation is accomplished by periodically reversing the polarity of the DC voltage that is applied to the liquid crystals. This prevents the crystals from becoming permanently aligned. Some panels control the frequency at which the polarity is reversed internally. Other panels require an outside source to control this. For panels requiring an outside source to implement this control, the ÉlanSC300 microcontroller supplies an AC modulation signal, M. This signal can be programmed to change state once per frame, once every 13 line clocks, or once every 61 line clocks.

Gray Shading (Gray Scaling)

Gray shading is a way of representing colors on a monochrome LCD display using shades of gray. The IRGB color information for each pixel is used to determine which gray shading algorithm will be used to display that pixel. The gray shading algorithm can be thought of as the duty cycle for displaying a particular pixel. For example, on the ÉlanSC300 microcontroller, an IRGB value of 1011 corresponds to a 1/3 duty cycle. This means that, if a pixel is supposed to be displayed, it would be displayed once for every three times the screen is refreshed. Refer to Table 3 on page 6 for a list of the ÉlanSC300 microcontroller's IRGB values and their corresponding gray shading duty cycles.

ÉlanSC300 MICROCONTROLLER EXTENSIONS TO THE CGA AND HGA STANDARDS

CGA Standard

The CGA standard was defined for a color CRT monitor with a resolution of 640 x 200 pixels. The CGA standard defines two graphic resolutions:

1. 640 x 200 pixels, with two colors, represented by 1 BPP.
2. 320 x 200 pixels, with four colors, represented by 2 BPP. In 320 x 200 mode, each pixel in the x dimension is displayed twice.

For both 640 x 200 and 320 x 200 resolutions, the memory is divided into two 8-Kbyte banks. Pixel data for even-numbered display rows (starting with 0) are stored in bank 0, address range B8000–B9FFF. Pixel data for odd-numbered display rows are stored in bank 1, address range BA000–BBFFF. The Graphics mode section of Chapter 3 in the *Élan™SC300 Microcontroller Programmer's Reference Manual* helps illustrate this.

In Text mode, 16 colors are available for characters. Text mode is discussed in detail in Chapter 3 of the *Élan™SC300 Microcontroller Programmer's Reference Manual* and follows the CGA standard implementation; therefore, it will not be discussed in detail in this document.

How the ÉlanSC300 Microcontroller Deviates from the CGA Standard

The ÉlanSC300 microcontroller can support any combination of resolutions at either 1 or 2 BPP (two or four colors), provided the video SRAM requirements do not exceed 32 Kbyte. For memory requirements greater than 16 Kbyte, the ÉlanSC300 microcontroller can be programmed to use three, or even all four, of the 8-Kbyte banks of video SRAM. If all four banks are used:

- The pixel data for display rows 0, 4, 8,... are in bank 1 (address range B8000–B9FFF).
- The pixel data for display rows 1, 5, 9,... are in bank 2 (address range BA000–BBFFF).
- The pixel data for display rows 2, 6, 10,... are in bank 3 (address range BC000–BDFFF).
- The pixel data for display rows 3, 7, 11,... are in bank 4 (address range BE000–BFFFF).

Note that two banks are the minimum. Thus, even if the pixel data for a small LCD screen could fit in 8 Kbyte (for example, 320 x 200 in 2-color mode), two banks must still be used.

In 4-color Graphics mode (2 BPP), it is possible to select whether or not pixels are doubled in the x dimension. (In the CGA standard, the pixel doubling is automatic in 4-color Graphics mode.) If the ÉlanSC300 microcontroller graphics control is set to display 2-BPP graphics (Port 3D8h bit 4 = 0), and if Video Index 20h bit 5 = 0, two pixels are displayed, effectively cutting the horizontal resolution in half. If Video Index 20h bit 5 = 1, only one pixel is displayed.

In CGA Text mode, the ÉlanSC300 microcontroller gets its fonts from SRAM rather than ROM and can support multiple font areas. The character width is fixed at eight pixels, but the character height is programmable. The way the font areas are addressed is discussed in detail in Chapter 3 of the *Élan™SC300 Microcontroller Programmer's Reference Manual*.

HGA Standard

The HGA standard was defined for a high-resolution, monochrome CRT with a resolution of 720 x 348 pixels. In Graphics mode, this allows two colors (black and white) represented by 1 BPP. The standard implementation is to use four 8-Kbyte banks of video SRAM in the same way the ÉlanSC300 microcontroller handles the four-bank CGA configuration. In the case of HGA, the standard address range for bank 1 is B0000–B1FFF, for bank 2 it is B2000–B3FFF, for bank 3 it is B4000–B5FFF, and for bank 4 it is B6000–B7FFF. The HGA standard also supports a second 32-Kbyte display page at B8000.

In Text mode, three character colors are available: black, light gray, and white.

How the ÉlanSC300 Microcontroller Deviates from the HGA Standard

The ÉlanSC300 microcontroller can support any combination of resolutions, provided the video SRAM requirements do not exceed 32 Kbyte. For memory requirements of 16 Kbyte or less, the ÉlanSC300 microcontroller can be configured to use only two 8-Kbyte banks of memory. For memory requirements of 16–24 Kbyte, the ÉlanSC300 microcontroller can be configured to use only three 8-Kbyte banks of memory.

Although the HGA standard supports two 32-Kbyte display pages, only 32 Kbyte of video SRAM are available on the ÉlanSC300 microcontroller. Therefore, the ÉlanSC300 microcontroller will only allow access to B0000–B7FFF in HGA mode.

In HGA Text mode, the ÉlanSC300 microcontroller gets its fonts from SRAM rather than ROM. The character width is fixed at nine pixels (the 9th column being automatically generated), but the character height is programmable. Chapter 3 of the *Élan™SC300 Microcontroller Programmer's Reference Manual* details the way the font areas are addressed.

MEMORY MAP ORGANIZATION FOR VARIOUS DISPLAY MODES AND PANEL SIZES

It is important to understand which settings affect the memory map of the video SRAM. This will enable you to properly display information on the screen. Chapter 3 of the *Élan™SC300 Microcontroller Programmer's Reference Manual* explains the memory map for Text mode, how fonts should be set up, and how character and attribute bytes are organized. It also explains memory maps for a few different display sizes for Graphics modes. This section expands on the information found in the *Élan™SC300 Microcontroller Programmer's Reference Manual*. It is meant to show how screen size and number of banks of video memory—bits per pixel (graphics) and font size (text)— affect the display data memory area.

Text Mode

In Text mode, the display data area is 16 Kbyte (for CGA, B8000h–BBFFFh and for HGA, B0000h–B3FFFh). Some of the remaining 16 Kbyte of the display area are used for four different font areas, three of which are available in CGA Text mode and one of which is available in HGA Text mode. Refer to Section 3.4 “Text Mode,” in the *Élan™SC300 Microcontroller Programmer's Reference Manual*, for Text mode memory map examples.

Graphics Mode

In Graphics mode, the video SRAM memory map is determined by the LCD resolution programmed into the ÉlanSC300 microcontroller, the number of banks of memory selected, and the number of bits per pixel required. Refer to Section 3.3, “Graphics Mode,” in the *Élan™SC300 Microcontroller Programmer's Reference Manual*, for Graphics mode memory map examples.

In Graphics mode, the video SRAM memory is split into 8-Kbyte banks. Either two or four 8-Kbyte banks are used. Refer to "Steps for Determining if a Single-Screen LCD Panel can be Driven by the ÉlanSC300 Microcontroller" on page 7 to determine how many banks are needed. The display rows are split evenly across the banks, as illustrated in the tables on the following page.

The number of bits per pixel and the programmed screen width determine how many bytes of data per row are needed. For example, a 128-pixel-wide screen, at 2 BPP needs $2 \cdot 128 = 256$ bits of data per row, and $256 \div 8 = 32$ bytes of data per row. At 2 bits per pixel, each pixel can be set to one of four colors. A 128-pixel-wide screen at 1 BPP needs $1 \cdot 128 = 128$ bits of data per row, and $128 \div 8 = 16$ bytes of data per row. At 1 BPP, each pixel can be set to only one of two colors. Thus, more colors require more memory.

Note: *With smaller panels, it may be necessary to program the ÉlanSC300 microcontroller's graphics controller with a logical screen width greater than the physical screen width of the LCD panel to meet the panel's refresh rate requirements. Reminder: if this is done, it is the logical screen width programmed into the graphics controller that determines the number of bytes per row.*

The following tables illustrate the memory map for a 128 x 128 pixel display set up for 2 BPP, in both a two-bank and four-bank configuration. As shown in the tables, a 128-pixel-wide display at 2 BPP requires 32 bytes of memory per row. If the width of the display were to increase or decrease, the change would cause the row ending address and subsequent starting addresses to increase or decrease, respectively.

Table 1. Memory Map of 128 x 128 Display Using Two Banks in Graphics Mode

Row Starting Address	Display Row	Row Ending Address
BANK 1 (B8000–B9FFF)		
B8000	0	B801F
B8020	2	B803F
B8040	4	B805F
...	...	...
B87E0	126	B87FF
BANK 2 (BA000–BBFFF)		
BA000	1	BA01F
BA020	3	BA03F
BA040	5	BA05F
...	...	...
BA7E0	127	BA7FF

Table 2. Memory Map of 128 x 128 Display Using Four Banks in Graphics Mode

Row Starting Address	Display Row	Row Ending Address
BANK 1 (B8000–B9FFF)		
B8000	0	B801F
B8020	4	B803F
B8040	8	B805F
...	...	...
B83E0	124	B83FF
BANK 2 (BA000–BBFFF)		
BA000	1	BA01F
BA020	5	BA03F
BA040	7	BA05F
...	...	...
BA3E0	125	BA3FF
BANK 1 (BC000–BCFFF)		
BC000	2	BC01F
BC020	6	BC03F
BC040	10	BC05F
...	...	...
BC3E0	126	BC3FF
BANK 2 (BE000–BEFFF)		
BE000	3	BE01F
BE020	7	BE03F
BE040	11	BE05F
...	...	...
BE3E0	127	BE3FF

HOW THE ÉlanSC300 MICROCONTROLLER DISPLAYS COLORS AS GRAY SHADES

The ÉlanSC300 microcontroller uses gray shading to represent colors on a monochrome LCD panel. It does this by associating a duty cycle with a particular IRGB color value (see Table 3). A duty cycle of x ÷ y indicates that for every y times the LCD display is refreshed, the pixel will be displayed x times.

Video Index 19H, bit 4 enables color mapping for CGA mode. This allows the mapping of one IRGB/gray scale value to another IRGB/gray scale value that looks better for a particular LCD. In this way, applications that use colors that do not look good on a particular display can still be used without having to change the application.

Table 3. Color Codes Mapped to Pixel Duty Cycles

Color	IRGB/Gray Scale	Duty Cycle	Common Denominator
Black	0000	0	0/24
Blue	0001	1/8	3/24
Green	0010	1/2	12/24
Cyan	0011	2/3	16/24
Red	0100	1/3	8/24
Magenta	0101	3/8	9/24
Brown	0110	3/4	18/24
Light Gray	0111	7/8	21/24
Dark Gray	1000	1/6	4/24
Light Blue	1001	1/4	6/24
Light Green	1010	5/6	20/24
Light Cyan	1011	3/4	18/24
Light Red	1100	1/2	12/24
Light Magenta	1101	2/3	16/24
Yellow	1110	7/8	21/24
White	1111	1	24/24

SINGLE-SCREEN LCD SUPPORT

A number of factors, including physical properties and mode of operation, determine whether or not the ÉlanSC300 microcontroller is capable of driving a specific display.

Physical Properties

- 4-bit data interface

  The display to be driven must have 4 data lines to connect to the ÉlanSC300 microcontroller's LCDD3–LCDD0

- Panel resolution

  The resolution, combined with the mode of operation described in the next section, determines the rate at which the ÉlanSC300 microcontroller will refresh the display, as well as the amount of video SRAM used (the ÉlanSC300 microcontroller supports a maximum of 32 Kbyte of video SRAM). The video SRAM becomes a limiting factor for supporting the display only when a Graphics mode is used.

- Refresh rate

  The rate at which the ÉlanSC300 microcontroller refreshes the display is affected by the mode of operation as well as the panel resolution. If a panel's refresh rate is not specifically stated, a maximum value can be approximated by taking:

  4 ÷ ((shift clock period) · (display resolution))

  The "4" comes from the number of pixel data bits per shift clock. On the ÉlanSC300 microcontroller, this number *must* be 4. For example, a 320 x 240 display that has a minimum shift clock period of 125 ns, will have a maximum refresh rate of 4 ÷ (125 ns · 320 · 240) = 416 Hz. A typical refresh value is around 75 Hz. A value less than 60 Hz will probably yield unacceptable results (e.g., the screen will appear very dim or will blink).

Mode of operation

The mode of operation affects the refresh rate of the display, and when a Graphics mode is selected, determines the amount of video SRAM required. When the ÉlanSC300 microcontroller is configured for a Graphics mode, the 32 Kbyte of memory is divided into four 8-Kbyte banks. Video Index register 09H controls whether the display information is divided between just two or all four of the memory banks.

- CGA Mode

  When CGA mode is selected, the ÉlanSC300 microcontroller uses a pixel clock frequency of 9.557 MHz. The refresh rate for Text mode, or 2-color Graphics mode can be calculated by dividing the pixel clock frequency by the screen resolution programmed into the video controller, taking into account the 20 extra pixel clocks per line mentioned earlier. The refresh rate for 4-color Graphics mode will be half that value.

- HGA Mode

  When HGA mode is selected, the ÉlanSC300 microcontroller uses a pixel clock frequency of 14.336 MHz. The refresh rate for Text mode or 2-color Graphics mode can be calculated by dividing the pixel clock frequency by the screen resolution programmed into the video controller, taking into account the 20 extra pixel clocks per line mentioned earlier.

STEPS FOR DETERMINING IF A SINGLE-SCREEN LCD PANEL CAN BE DRIVEN BY THE ÉlanSC300 MICROCONTROLLER

1. Does the panel have a 4-bit data interface?
   - Yes: Go to step 2.
   - No: Panel cannot be driven by the ÉlanSC300 microcontroller.
2. Determine what rate the ÉlanSC300 microcontroller will refresh the LCD panel. Refresh rate is a factor of the LCD panel resolution and pixel clock frequency, which is determined by the mode of operation. For a panel size of W x H, and a bits-per-pixel value of BPP (1 or 2), the refresh rate will be as follows:

   (pixel clock frequency) ÷ ((W + 20) · BPP · H)

   In CGA mode, the pixel clock frequency is 9.557 MHz. In HGA mode the pixel clock frequency is 14.336 MHz. Therefore, when the ÉlanSC300 microcontroller is configured for a 320 x 240, 1-BPP resolution LCD in CGA mode, the refresh rate is 9.557 MHz ÷ (340 · 1 · 240) = 117 Hz. When the ÉlanSC300 microcontroller is configured for the same size display in HGA mode, the refresh rate is 14.336 MHz ÷ (340 · 1 · 240) = 175 Hz. Note that 2-color (1 BPP) mode can be supported by either CGA or HGA, but the HGA refresh rate is approximately 50% higher.
3. Does the refresh rate fall in the range supported by the LCD panel?
   - Yes: Go to step 4.
   - No: Panel cannot be driven by the ÉlanSC300 microcontroller.

Note: *If the refresh rate driven by the ÉlanSC300 microcontroller is too fast for the panel, it is possible to effectively slow the refresh rate to the panel by programming the ÉlanSC300 microcontroller's graphics controller using a logical panel width that is greater than the physical LCD panel width. If this is done, the logical panel width must be used for all calculations, and it should first be tested on the same panel before making the design choice. This will determine if the panel can handle the additional shift clocks sent to it as a result of the larger panel width.*

4. Does the ÉlanSC300 microcontroller support the LCD panel's video SRAM requirements?

   This is a concern only if the LCD will be operated in a Graphics mode; therefore, Text mode is not discussed.

   2 bits per pixel (four shades of gray)
   This mode consumes the most memory. Each bank of memory is 8 Kbyte (8192 bytes). In 2 BPP mode, each byte contains the equivalent of 4 pixels of information. Each bank can store information for 32,768 pixels. Therefore, if configured for 4 banks of memory, the maximum number of pixels that the ÉlanSC300 microcontroller can support is 4 · 32,768 = 131,072 pixels. If configured for two banks of memory, the maximum number of pixels that the ÉlanSC300 microcontroller can support is 2 · 32,768 = 65,536 pixels.

   For example: a 320 x 240 LCD display has 320 · 240 = 76,800 pixels. Therefore, the ÉlanSC300 microcontroller will support this resolution LCD if configured for four banks of memory.

   1 bit per pixel (2 shades of gray)
   Each bank of memory is 8 Kbyte (8192 bytes). In 1 BPP mode, each byte contains the equivalent of 8 pixels of information. Each bank can store information for 65,536 pixels. Therefore, if configured for four banks of memory, the maximum number of pixels that the ÉlanSC300 microcontroller can support is 4 · 65,536 = 262,144 pixels. If configured for two banks of memory, the maximum number of pixels

that the ÉlanSC300 microcontroller can support is 2 · 65,536 = 131,072 pixels.
For example: a 640 x 400 LCD display has 640 · 400 = 256,000 pixels. Therefore the ÉlanSC300 microcontroller will support this resolution LCD if configured for four banks of memory.

5. If the LCD panel meets the criteria listed in the previous four steps, then the ÉlanSC300 microcontroller is capable of driving the LCD panel.

PROGRAMMING THE ÉlanSC300 MICROCONTROLLER TO WORK WITH A SINGLE-SCREEN LCD PANEL

Before programming the ÉlanSC300 microcontroller to work with a single-screen LCD panel, the following information is needed:

- Display resolution in pixels, W x H (e.g., 640 x 200)
- Mode of operation (i.e., CGA graphics-1 BPP, CGA graphics-2 BPP, CGA text, HGA graphics-1 BPP, HGA text)
- If Text mode, the font size (8 x FH for CGA or 9 x FH for HGA, where FH = font height; 8 x 8 is CGA standard, 9 x 14 is HGA standard)
- If Graphics mode, the number of 8-Kbyte banks of memory (see "Steps for Determining if a Single-Screen LCD Panel can be Driven by the ÉlanSC300 Microcontroller" on page 7)

Determine the Video Index Values to Be Programmed

First determine the Video Index values for Video Indexes 00–0Fh, based on the information listed in the four items above.

CGA & HGA Graphics Mode Values

- **(HT = W · BPP ÷ 16)**

  The Horizontal Total Register (Index 00H) and Horizontal Displayed Register (Index 01H) are programmed with the number of 16-bit words per display row. In 2-color mode (1 BPP), this equals the width of the display in pixels divided by 16. In 4-color mode (2 BPP), this equals two times the width of the display in pixels divided by 16. For example, a 320 x 240 display is 320 pixels wide. Therefore, in 2-color mode (1 BPP), HT = 320 ÷ 16 = 20 (14h).

- **(VT = H ÷ BANKS)**

  The Vertical Total Register (Index 04H), Vertical Display Register (Index 06H), and Vertical Sync Position Register (Index 07H) are programmed with the value of the height of the display in pixels, divided by the number of banks of memory. For example, a 320 x 240 display is 240 pixels high. If four banks of memory are going to be used, then VT = 240 ÷ 4 = 60 (3Ch).

- **(MSL = BANKS – 1)**

  The Max Scan Line Register (Index 09H) is programmed with the number of banks of memory to be used, minus 1. If four banks of memory are going to be used, then MSL = 4 – 1 = 3.

- **(SAH) (SAL)**

  The Start Address High (Index 0CH) and Start Address Low (Index 0DH) Video Indexes together form a 14-bit offset (bits 7 and 6 of Index 0CH are not used) that determines the location in video memory where the video controller will start fetching data to be displayed on the screen. Typically, these indexes will be set to 0 so the display data starts being fetched from B8000h. Scrolling can be implemented by changing the values in these registers.

- All the other index registers (02, 03, 05, 08, 0Ah, 0Bh, 0Eh, 0Fh) will be programmed with 0h.

CGA & HGA Text Mode Values

- **HT = W · BPP ÷ 16**

  The Horizontal Total Register (Index 00H) and Horizontal Displayed Register (Index 01H) are programmed with the number of characters per row. The character width is hard coded to be 8 pixels wide. Therefore, HT is the width of the display in pixels divided by 8. For example, a 320 x 240 display is 320 pixels wide. Therefore HT = 320 ÷ 8 = 40 (28h).

- **(VT = H ÷ Y)**

  The Vertical Total Register (Index 04H), Vertical Displayed Register (06H), and Vertical Sync Position Register (07H) are programmed with the value of the number of rows of characters. This is determined by the height of the display in pixels, divided by the height of the character in pixels (Y from the equation below). For example, a 320 x 240 display is 240 pixels high. If using an 8 x 10 font, then VT = 240 ÷ 10 = 24 (18h).

- **(MSL = Y –1)**

  The Max Scan Line Register (Index 09H) is programmed with the pixel height of the font – 1. For example, if using an 8 x 10 font, then MSL = 10 – 1 = 9.

■ **(CS = Cursor Start Row –1)**
(CE = Cursor End Row –1)

The Cursor Start Register (Index 0AH), and Cursor End Register (Index 0BH) define the pixel rows within a font where the cursor will be displayed. Typically, the cursor is defined to be near the bottom of the character matrix. For example, an 8 x 10 font has 10 pixel rows. Therefore, to have the cursor positioned at rows 8 and 9, CS = 8 – 1 = 7, and CE = 9 – 1 = 8.

Note: *Bits 6 and 5 of the Cursor Start Register (Index 0AH) control the cursor's behavior. See Table 4.*

Table 4. Cursor Control Bits (Index 0AH)

Bit 6	Bit 5	Cursor Behavior
0	0	No blinking
0	1	Cursor not displayed
1	0	Blinking with 16 · refresh rate
1	1	Blinking with 32 · refresh rate

■ **(SAH) (SAL)**

The Start Address High (Index 0CH) and Start Address Low (Index 0DH) Video Indexes together form a 14-bit offset (bits 7 and 6 of Index 0CH are not used) that determines the location in video memory where the video controller will start fetching data to be displayed on the screen. Typically, these indexes will be set to 0 so the display data starts being fetched from B8000h. Scrolling can be implemented by changing the values in these registers.

■ **(CAH) (CAL)**

Cursor Address High (Index 0EH) and Cursor Address Low (Index 0FH) Video Indexes together form a 14-bit offset (bits 7 and 6 of Index 0CH are not used) that determines the location in video memory where the video controller will display the cursor when the corresponding video data is displayed on the screen. For example, if SAH = 0, SAL = 0, CAH = 0, and CAL = 0, the cursor will be displayed in the upper left corner of the screen.

■ All the other index registers (02, 03, 05, 08, 0Ah, 0Bh, 0Eh, 0Fh) will be programmed with 0h.

Initializing the Video Controller

The following sequences show the proper sequence for initializing the video controller. The following steps assume the video controller is set up to its default mode of CGA and the Video Index registers are unlocked. (To unlock the video registers, write 12h to Port 3D4h, and read Port 3D5h with no I/O cycles between these.)

Note: *Only the minimum set of registers required to get the screen running are initialized. There are a number of additional features such as auto screen blanking and text truncation that are discussed in the* Élan™SC300 Microcontroller Programmer's Reference Manual *that are not reiterated in this document.*

CGA Graphics

The Video Index registers are accessed by writing the register index to Port 3D4h, and then writing or reading the register data to or from Port 3D5h.

1. Set Video Index 18H = 50h
   – Sets up LCD panel type to single screen (bits 7 and 6)
   – Sets display type to LCD (bit 4)
2. Set Video Index 00H = HT
3. Set Video Index 01H = HT
4. Set Video Index 02H = 0
5. Set Video Index 03H = 0
6. Set Video Index 04H = VT
7. Set Video Index 05H = 0
8. Set Video Index 06H = VT
9. Set Video Index 07H = VT
10. Set Video Index 08H = 0
11. Set Video Index 09H = MSL
12. Set Video Index 0AH = 0
13. Set Video Index 0BH = 0
14. Set Video Index 0CH = SAH
15. Set Video Index 0DH = SAL
16. Set Video Index 0EH = 0
17. Set Video Index 0FH = 0
18. Set Video Index 20H (bit 5 controls dot doubling)
   – If 2 BPP Graphics mode is used, set bit 5 = 1 to disable dot doubling.

Note: *This is the common setting.*

   – If 2 BPP Graphics mode is used, and dot doubling is desired (CGA standard), then set bit 5 = 0.

19. If Color Mapping is going to be used:
   – Set Video Index's 14H–17H and 1CH–1FH accordingly
   – Set Video Index 19H = 10H
20. Set Video Port 3D8H
   – If 2 BPP Graphics mode, set to 0AH
   – If 1 BPP Graphics mode, set to 1AH
21. Set Video Port 3D9H
   – Set colors accordingly (see the register description in the *Élan™SC300 Microcontroller Programmer's Reference Manual*)

CGA Text

The Video Index registers are accessed by writing the register index to Port 3D4h and then writing or reading the register data to or from Port 3D5h.

1. Set Video Index 18H = 50h
   - Sets up LCD panel type to single screen (bits 7 and 6)
   - Sets display type to LCD (bit 4)
2. Set Video Index 00H = HT
3. Set Video Index 01H = HT
4. Set Video Index 02H = 0
5. Set Video Index 03H = 0
6. Set Video Index 04H = VT
7. Set Video Index 05H = 0
8. Set Video Index 06H = VT
9. Set Video Index 07H = VT
10. Set Video Index 08H = 0
11. Set Video Index 09H = MSL
12. Set Video Index 0AH = CS
13. Set Video Index 0BH = CE
14. Set Video Index 0CH = SAH
15. Set Video Index 0DH = SAL
16. Set Video Index 0EH = CAH
17. Set Video Index 0FH = CAL
18. If Color Mapping is going to be used:
    - Set Video Indexes 14H–17H and 1CH–1FH accordingly
    - Set Video Index 19H = 10H
19. Set Video Port 3D8H = 08H
    - Enable Text mode
20. Set Video Port 3D9H
    - Set colors accordingly (see the register description in the *Élan™SC300 Microcontroller Programmer's Reference Manual*)

HGA Graphics

Initially, the Video Index registers are accessed by writing the register index to Port 3D4h, and then writing or reading the register data to or from Port 3D5h. This occurs until the controller is switched to HGA mode. After this occurs, Port 3B4 is used for the index and Port 3B5 is used for the data.

1. Set Video Index 18H = 51h (use Ports 3D4 and 3D5)
   - Sets Graphics mode to HGA (from now on use ports 3B4 and 3B5)
   - Sets up LCD panel type to single screen (bits 7 and 6)
   - Sets display type to LCD (bit 4)
2. Set Video Index 00H = HT
3. Set Video Index 01H = HT
4. Set Video Index 02H = 0
5. Set Video Index 03H = 0
6. Set Video Index 04H = VT
7. Set Video Index 05H = 0
8. Set Video Index 06H = VT
9. Set Video Index 07H = VT
10. Set Video Index 08H = 0
11. Set Video Index 09H = MSL
12. Set Video Index 0AH = 0
13. Set Video Index 0BH = 0
14. Set Video Index 0CH = SAH
15. Set Video Index 0DH = SAL
16. Set Video Index 0EH = 0
17. Set Video Index 0FH = 0
18. Set Video Port 3BFH = 01H
    - Master enable for Graphics mode
19. Set Video Port 3B8H = 0AH
    - Enable Graphics mode
    - Enable video display

HGA Text

Initially, the Video Index registers are accessed by writing the register index to Port 3D4h, and then writing or reading the register data to or from Port 3D5h. This occurs until the controller is switched to HGA mode. After this occurs, Port 3B4 is used for the index and Port 3B5 is used for the data.

1. Set Video Index 18H = 51h (Use ports 3D4 and 3D5)
   - Sets Graphics mode to HGA (from now on use Ports 3B4 and 3B5)
   - Sets up LCD panel type to single screen (bits 7 and 6)
   - Sets display type to LCD (bit 4)
2. Set Video Index 00H = HT
3. Set Video Index 01H = HT
4. Set Video Index 02H = 0
5. Set Video Index 03H = 0
6. Set Video Index 04H = VT
7. Set Video Index 05H = 0
8. Set Video Index 06H = VT
9. Set Video Index 07H = VT
10. Set Video Index 08H = 0
11. Set Video Index 09H = MSL
12. Set Video Index 0AH = CS
13. Set Video Index 0BH = CE
14. Set Video Index 0CH = SAH
15. Set Video Index 0DH = SAL
16. Set Video Index 0EH = CAH
17. Set Video Index 0FH = CAL
18. Set Video Port 3BFH = 00H
    - Master disable for Graphics mode
19. Set Video Port 3B8H = 08H
    - Enable video display

DUAL-SCREEN LCD SUPPORT

The ÉlanSC300 microcontroller supports only one type of dual-screen LCD display. The display must be 640 x 200 (640 x 100 per panel). The display must accept 8 data bits total (4 data bits per panel). The ÉlanSC300 microcontroller will support the panel in CGA mode only and refresh the display at a rate of approximately 75 Hz.

The reason for the limited dual-screen support is that the ÉlanSC300 microcontroller does not have any dual-screen registers that, among other things, allow you to program where the fetching of data for the lower panel should begin. This was hard-wired in the ÉlanSC300 microcontroller.

Programming the ÉlanSC300 Microcontroller for Dual-Screen LCD Support

The programming values for this dual-screen LCD are identical to the ones used for the single-screen panel with two exceptions. Video Index 18H bits 7 and 6 should be set to 00 instead of 01. The ÉlanSC300 microcontroller Configuration Register B1H bit 1 must be set to 1 to enable the additional 4 lower-panel LCD data bits.

The Physical Pin Connections to the ÉlanSC300 Microcontroller

To connect an LCD panel to the ÉlanSC300 microcontroller, the following pins need to be connected:

LCDD3–LCDD0: These are the panel's data bits. LCDD3 is the MSB; LCDD0 is the LSB. Find out which data bit on the LCD is the MSB, which is the LSB, and connect the pins accordingly.

CP1: Also known as the line clock or latch (see definition in "Line Clock (CP1)" on page 2). This should be connected to the equivalent line on the LCD panel.

CP2: Also known as the shift clock or data shift (see definition in "Shift Clock (CP2)" on page 2). This should be connected to the equivalent line on the LCD panel.

FRM: Also known as FLM, or frame (see definition in "Frame Start (FRM)" on page 2). This should be connected to the equivalent line on the LCD panel.

M: AC modulation (see definition in "LCD Panel AC Modulation (M)" on page 2). Some panels require this signal, others do not. Connect this signal if appropriate.

LCDDL3–LCDDL0: These data bits are required only if a dual-screen LCD panel is being connected. These are the data bits for the lower panel. LCDDL3 is the MSB for the lower panel, and LCDDL0 is the LSB for the lower panel.

Other LCD pin connections

The other connections that need to be made to an LCD vary. For example:

- **Contrast voltage**: can be positive or negative, typically about –22 volts.
- +5 V
- GND
- **Display enable**: This is usually a simple 5-V enable signal that some panels require. This can easily be connected to one of the ÉlanSC300 microcontroller's PMC pins.
- $\mathbf{V_{ee}}$: Typically a voltage in the same range as the contrast voltage. Refer to panel specifications.

Using the ÉlanSC300 Microcontroller Evaluation Board to Test an LCD Panel

The ÉlanSC300 microcontroller evaluation board has a 20-pin header (P18) with all the LCD interface signals connected to it. The evaluation board provides a –14 V to –16 V contrast voltage (adjust using VR1) and a –17 V_{ee} voltage. If the panel to be tested requires voltages other than these, a separate circuit will need to be bread boarded, or an external DC power supply used. The following evaluation board jumpers and resistor packs should also be set to configure the evaluation board for Internal Video mode: JP16 = 1–2; JP18 = Open; Install RP3 and RP4 (RP1, RP2, RP5, and RP6 must be empty).

Note that the evaluation board BIOS is set up to configure for a 640 x 200 single-panel display when the ÉlanSC300 microcontroller is in Internal Video mode. If a panel size other than this is used without modifying the BIOS, the screen may not display anything at boot-up. Refer to "LCD Development Tips and Tricks" for instructions on how to use CTTY for console activity while developing a screen application.

Also note that if a dual-screen LCD panel is used, the $\overline{\text{SBHE}}$, IRQ14, MCS16, and IOS16 ISA signals are no longer available outside the chip. This means the standard 16-bit IDE interface cannot be used.

Table 5 shows how the pins on the 20-pin header should be connected to the pins of a Hitachi LMG6272XNFR, 640 x 200 single-panel display.

Table 5. Pin Connections

The ÉlanSC300 Microcontroller Evaluation Board P18 Header Pins				HITACHI: LMG6272XNFR 640 x 200 Single Panel LCD		
Pin #	**Pin Name**	**Function**		**Pin #**	**Pin Name**	**Function**
1	VccLCD5	+5	=	10	Vdd	Power supply for logic circuit
2	GND	Ground	=	11	Vss	Ground
3	LCDFRM	Frame start	=	5	FLM	The FLM signal indicates the start of each display cycle
4	LCDCP1	Line clock	=	7	CL1	Data latch
5	LCDCP2	Shift clock	=	8	CL2	Data shift
6	LCDM	AC modulation		x		Not used
7	LCDD0	Data bit 0 (LSB)	=	1	D0	Data bit 0 (LSB)
8	LCDD1	Data bit 1	=	2	D1	Data bit 1
9	LCDD2	Data bit 2	=	3	D2	Data bit 2
10	LCDD3	Data bit 3 (MSB)	=	4	D3	Data bit 3 (MSB)
11	LCDDL0	Dual-screen panel data bit 0 for lower panel		x		Not used
12	LCDDL1	Dual-screen panel data bit 1 for lower panel		x		Not used
13	LCDDL2	Dual-screen panel data bit 2 for lower panel		x		Not used
14	LCDDL3	Dual-screen panel data bit 3 for lower panel		x		Not used
15	Contrast	Adjustable voltage between –14 V to –16 V	=	9	Vo	LCD driving voltage
16	Vee	–17 V	=	12	Vee	Power supply for LC driving
17—20	NC	Not connected				

LCD DEVELOPMENT TIPS & TRICKS

Use CTTY for Console I/O

When using an LCD as the console device under DOS, there may be problems debugging the LCD because DOS will also be attempting to write to the LCD through the BIOS. A way around this is by diverting the DOS console I/O through the serial port. This allows use of debug or other application programs to access the ÉlanSC300 microcontroller's registers and view the results on the LCD display. The following steps are recommended:

1 Connect a null modem cable between COM1 on the ÉlanSC300 microcontroller and a PC running PROCOMM or other terminal program. (A lap link serial cable is a null modem cable).

2. On the PC running PROCOMM, set up for a direct connect to the serial port, running 9600 baud, 8 data, 1 stop bit, and no parity. Set it up for a terminal type of either TTY or ANSI-BBS.
3. On the ÉlanSC300 microcontroller system, boot DOS and have the following commands in the **autoexec.bat** file:
   - mode com1 9600,n,8,1
   - ctty com1
4. After the system boots, the DOS prompt will be visible on the PC running PROCOMM.
5. Executing programs that use the console I/O such as debug should be possible.
6. Programs that talk directly to video memory will display the output on the ÉlanSC300 microcontroller system, but the keyboard input will come from the PC running PROCOMM.
7. Below are two things to keep in mind:
   - To debug an LCD, the ÉlanSC300 microcontroller evaluation board should be configured for Internal Video mode
   - If using Phoenix BIOS, the BIOS will lock the Video Index registers. Remember to unlock them by reading Video Index 12h. The BIOS will also lock the upper 16K of display memory, which is used for font storage in Text mode. This can be unlocked by clearing bit 3 of Video Index 20h.

The BIOS May Also Be Updating the Screen

If using the ÉlanSC300 microcontroller evaluation board with an LCD standalone, (i.e., not option 1 above) and booting to DOS, note that the BIOS is accessing the Video registers every time it is called to update the screen (i.e., typing in debug or from the DOS prompt). If trying to use debug or other programs to modify the Video registers, be aware that they may be modified by the BIOS as well, specifically Ports 3D8h and 3D9h.

Using Standard CGA or HGA Drivers on a Nonstandard Display Size

It is possible to use standard CGA or HGA drivers on a nonstandard-sized screen. The method is to program the ÉlanSC300 microcontroller with values for the LCD width that match those that the driver expects. The LCD height can be set to the actual LCD height. This will cause the video SRAM to be organized as the standard driver is expecting. The LCD panel will be displaying the upper left of a virtual screen.

Configuring the Élan™SC300 Device's Internal CGA Controller for a Specific LCD Panel

AMD

Application Note

This application note explains how to determine if a specific LCD panel is supported by the Élan™SC300 microcontroller, and in what modes (CGA two-color or four-color graphics, CGA text, HGA graphics, HGA text); how to configure the ÉlanSC300 microcontroller to work with the LCD panel in those modes; and which signals need to be connected to the ÉlanSC300 microcontroller to properly drive the panel.

INTRODUCTION

There are a number of factors that need to be understood to accomplish the tasks mentioned above. The sections in the first part of this document provide you with the understanding and background needed to follow the procedures beginning on page 7. You are also encouraged to read and have available for reference Chapter 3, "Video Controller" of the *Élan™SC300 Microcontroller Programmer's Reference Manual*, order #18470.

DEFINITIONS

The following terms are used throughout this document. Understanding these terms as they apply to the ÉlanSC300 microcontroller is essential for understanding how to configure the microcontroller's Video Index registers to work with various LCD panels.

Single-Screen Panels

A single-screen LCD panel consists of the Liquid Crystal Display (LCD) and a set of column drivers (also known as segment drivers) and row drivers (also called common drivers). These drivers supply the appropriate DC voltage to the liquid crystals in the display such that the crystals become aligned and block the light from passing through the display. An entire row of pixels is biased at one time. For this to occur, the pixel data for one row is loaded into the segment drivers. Then, a DC voltage is applied to the segment drivers and to the common driver that is attached to the row being displayed. This process is then repeated for each row of the display.

Figure 1 is a simplified illustration of an LCD panel that can be driven by the ÉlanSC300 microcontroller. Not all signals are shown (e.g., FRM and M), and not all connections of existing signals are shown. As depicted in Figure 1, the ÉlanSC300 microcontroller drives 4 bits of pixel data at a time. The data is clocked into the segment drivers using the shift clock. When all the data for a row of pixels is loaded into the segment drivers, the line clock will pulse, causing the common driver to activate the voltage to the "common line" for the row of pixels being displayed.

Dual-Screen Panels

A dual-screen LCD panel is two single-screen panels glued so tightly together that they appear to be one panel. Each panel has separate data bits but shares the same control signals: shift clock, line clock, and frame start (see the sections that follow). For example, a 640 x 200 dual-screen panel is actually two 640 x 100 single-screen panels glued together.

The only dual-screen panel resolution the ÉlanSC300 microcontroller supports is 640 x 200. More specific information as to how the ÉlanSC300 microcontroller handles a dual-screen panel is discussed later in this document.

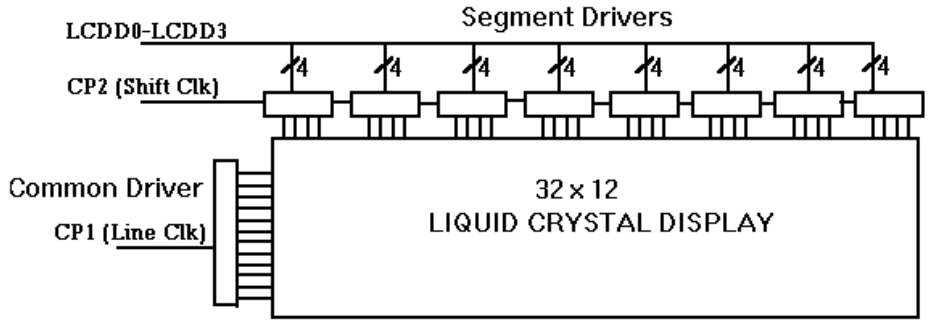

Figure 1. Simplified LCD Panel Configuration

Publication# **20749** Rev: **A** Amendment/**0**
Issue Date: **June 1997**

Pixel Clock

This is the standard unit of measure from which all of the LCD clocking signals can be derived. The pixel clock can be loosely defined as the rate at which pixels are displayed on the screen. On the ÉlanSC300 microcontroller, the pixel clock used is one of two fixed frequencies: 14.336 MHz for HGA mode or 9.557 MHz (which is 2/3 of 14.336) for CGA mode. On the ÉlanSC300 microcontroller, the pixel clock is internal to the video controller and is not available external to the chip.

Shift Clock (CP2)

The shift clock, known as the CP2 signal on the ÉlanSC300 microcontroller, tells the LCD panel when data is valid on the LCD data bus (the ÉlanSC300 microcontroller pins LCDD0–LCDD3 and LCDDL0–LCDDL3). Data is valid on the falling edge of shift clock.

The frequency of the shift clock can be determined by the pixel clock frequency divided by the number of data bits per panel. For both single- and dual-screen panels with the ÉlanSC300 microcontroller, the number of bits per panel is four. Therefore, the shift clock frequency will be either 14.336 MHz ÷ 4 data bits = 3.58 MHz, or 9.557 MHz ÷ 4 data bits = 2.389 MHz.

Line Clock (CP1)

The line clock, known as the CP1 signal on the ÉlanSC300 microcontroller, tells the LCD panel that a complete row (horizontal line) of pixel data has been sent to the LCD panel to be displayed. Additional shift clocks indicate valid data for the next row of pixels (i.e., display the current row and prepare to receive pixel data for the next row).

A line clock pulse will occur once for every row of pixel data sent to the LCD panel. For example, a 320 x 240 LCD screen has 240 rows; therefore, 240 line clocks will occur each time the panel is refreshed.

When CP1 is asserted on the ÉlanSC300 microcontroller's LCD controller, there is a period of about 20 pixel clock cycles during which CP2 will not be asserted. This has a slight effect on the refresh rate as described later in this document.

Frame Start (FRM)

Frame start, known as FRM on the ÉlanSC300 microcontroller, is asserted at the start of every frame (panel scan). This signal tells the LCD panel that the next data sent to it via the shift clock will be for the top row of the panel. This signal is sometimes referred to as the FLM (First Line Marker) signal.

On the ÉlanSC300 microcontroller's LCD controller, there is no "dead time" when FRM is asserted. The FRM assertion overlaps with the normal CP1 assertion at the end of the last line.

Refresh Rate

The refresh rate of the LCD panel is the number of times per second the screen is redrawn. This affects the crispness and brightness of the image on the display. On the ÉlanSC300 microcontroller, the refresh rate is affected by the screen resolution programmed into the video controller, the number of bits per pixel (BPP) (in Graphics mode only), and the pixel clock frequency.

On the ÉlanSC300 microcontroller, the refresh rate for Text mode or 2-color Graphics mode can be calculated by dividing the pixel clock frequency by the screen resolution programmed into the video controller, taking into account the 20 extra pixel clocks per line mentioned in the previous section. The refresh rate for 4-color Graphics mode will be half that amount. For example, if the controller is programmed for a 320 x 240 LCD screen and configured for CGA 2-color mode (which means that the 9.557 MHz pixel clock is used), then each line consists of 320 + 20 = 340 pixel clocks, and the refresh rate would be 9.557 MHz ÷ (340 · 240) = 117.1 Hz.

If the refresh rate is marginal for the LCD size in CGA Graphics mode, a 50% increase in refresh rate can be achieved by using HGA Graphics mode. On the ÉlanSC300 microcontroller, the selection of CGA or HGA mode can remain independent of the resolution. (This is discussed later in this document.)

DC Voltage Biasing

DC voltage biasing is a condition of LCD panels when the liquid crystals in the display become permanently aligned. Normally, the liquid crystals are nonaligned, allowing light to pass through the display. When a DC voltage is applied, the liquid crystals align themselves, blocking the light from passing through the display. The effect of this alignment causes a pixel to appear dark on the display. As a panel becomes DC voltage biased, the liquid crystals in the display become permanently aligned, eventually causing the entire display to appear black. AC modulation (defined in the next section) is used to prevent DC biasing.

LCD Panel AC Modulation (M)

All LCD panels implement what is known as AC modulation to prevent DC voltage biasing from damaging the display. AC modulation is accomplished by periodically reversing the polarity of the DC voltage that is applied to the liquid crystals. This prevents the crystals from becoming permanently aligned. Some panels control the frequency at which the polarity is reversed internally. Other panels require an outside source to control this. For panels requiring an outside source to implement this control, the ÉlanSC300 microcontroller supplies an AC modulation signal, M. This signal can be programmed to change state once per frame, once every 13 line clocks, or once every 61 line clocks.

Gray Shading (Gray Scaling)

Gray shading is a way of representing colors on a monochrome LCD display using shades of gray. The IRGB color information for each pixel is used to determine which gray shading algorithm will be used to display that pixel. The gray shading algorithm can be thought of as the duty cycle for displaying a particular pixel. For example, on the ÉlanSC300 microcontroller, an IRGB value of 1011 corresponds to a 1/3 duty cycle. This means that, if a pixel is supposed to be displayed, it would be displayed once for every three times the screen is refreshed. Refer to Table 3 on page 6 for a list of the ÉlanSC300 microcontroller's IRGB values and their corresponding gray shading duty cycles.

ÉlanSC300 MICROCONTROLLER EXTENSIONS TO THE CGA AND HGA STANDARDS

CGA Standard

The CGA standard was defined for a color CRT monitor with a resolution of 640 x 200 pixels. The CGA standard defines two graphic resolutions:

1. 640 x 200 pixels, with two colors, represented by 1 BPP.
2. 320 x 200 pixels, with four colors, represented by 2 BPP. In 320 x 200 mode, each pixel in the x dimension is displayed twice.

For both 640 x 200 and 320 x 200 resolutions, the memory is divided into two 8-Kbyte banks. Pixel data for even-numbered display rows (starting with 0) are stored in bank 0, address range B8000–B9FFF. Pixel data for odd-numbered display rows are stored in bank 1, address range BA000–BBFFF. The Graphics mode section of Chapter 3 in the *Élan™SC300 Microcontroller Programmer's Reference Manual* helps illustrate this.

In Text mode, 16 colors are available for characters. Text mode is discussed in detail in Chapter 3 of the *Élan™SC300 Microcontroller Programmer's Reference Manual* and follows the CGA standard implementation; therefore, it will not be discussed in detail in this document.

How the ÉlanSC300 Microcontroller Deviates from the CGA Standard

The ÉlanSC300 microcontroller can support any combination of resolutions at either 1 or 2 BPP (two or four colors), provided the video SRAM requirements do not exceed 32 Kbyte. For memory requirements greater than 16 Kbyte, the ÉlanSC300 microcontroller can be programmed to use three, or even all four, of the 8-Kbyte banks of video SRAM. If all four banks are used:

- The pixel data for display rows 0, 4, 8,... are in bank 1 (address range B8000–B9FFF).
- The pixel data for display rows 1, 5, 9,... are in bank 2 (address range BA000–BBFFF).
- The pixel data for display rows 2, 6, 10,... are in bank 3 (address range BC000–BDFFF).
- The pixel data for display rows 3, 7, 11,... are in bank 4 (address range BE000–BFFFF).

Note that two banks are the minimum. Thus, even if the pixel data for a small LCD screen could fit in 8 Kbyte (for example, 320 x 200 in 2-color mode), two banks must still be used.

In 4-color Graphics mode (2 BPP), it is possible to select whether or not pixels are doubled in the x dimension. (In the CGA standard, the pixel doubling is automatic in 4-color Graphics mode.) If the ÉlanSC300 microcontroller graphics control is set to display 2-BPP graphics (Port 3D8h bit 4 = 0), and if Video Index 20h bit 5 = 0, two pixels are displayed, effectively cutting the horizontal resolution in half. If Video Index 20h bit 5 = 1, only one pixel is displayed.

In CGA Text mode, the ÉlanSC300 microcontroller gets its fonts from SRAM rather than ROM and can support multiple font areas. The character width is fixed at eight pixels, but the character height is programmable. The way the font areas are addressed is discussed in detail in Chapter 3 of the *Élan™SC300 Microcontroller Programmer's Reference Manual*.

HGA Standard

The HGA standard was defined for a high-resolution, monochrome CRT with a resolution of 720 x 348 pixels. In Graphics mode, this allows two colors (black and white) represented by 1 BPP. The standard implementation is to use four 8-Kbyte banks of video SRAM in the same way the ÉlanSC300 microcontroller handles the four-bank CGA configuration. In the case of HGA, the standard address range for bank 1 is B0000–B1FFF, for bank 2 it is B2000–B3FFF, for bank 3 it is B4000–B5FFF, and for bank 4 it is B6000–B7FFF. The HGA standard also supports a second 32-Kbyte display page at B8000.

In Text mode, three character colors are available: black, light gray, and white.

How the ÉlanSC300 Microcontroller Deviates from the HGA Standard

The ÉlanSC300 microcontroller can support any combination of resolutions, provided the video SRAM requirements do not exceed 32 Kbyte. For memory requirements of 16 Kbyte or less, the ÉlanSC300 microcontroller can be configured to use only two 8-Kbyte banks of memory. For memory requirements of 16–24 Kbyte, the ÉlanSC300 microcontroller can be configured to use only three 8-Kbyte banks of memory.

Although the HGA standard supports two 32-Kbyte display pages, only 32 Kbyte of video SRAM are available on the ÉlanSC300 microcontroller. Therefore, the ÉlanSC300 microcontroller will only allow access to B0000–B7FFF in HGA mode.

In HGA Text mode, the ÉlanSC300 microcontroller gets its fonts from SRAM rather than ROM. The character width is fixed at nine pixels (the 9th column being automatically generated), but the character height is programmable. Chapter 3 of the *Élan™SC300 Microcontroller Programmer's Reference Manual* details the way the font areas are addressed.

MEMORY MAP ORGANIZATION FOR VARIOUS DISPLAY MODES AND PANEL SIZES

It is important to understand which settings affect the memory map of the video SRAM. This will enable you to properly display information on the screen. Chapter 3 of the *Élan™SC300 Microcontroller Programmer's Reference Manual* explains the memory map for Text mode, how fonts should be set up, and how character and attribute bytes are organized. It also explains memory maps for a few different display sizes for Graphics modes. This section expands on the information found in the *Élan™SC300 Microcontroller Programmer's Reference Manual*. It is meant to show how screen size and number of banks of video memory—bits per pixel (graphics) and font size (text)— affect the display data memory area.

Text Mode

In Text mode, the display data area is 16 Kbyte (for CGA, B8000h–BBFFFh and for HGA, B0000h–B3FFFh). Some of the remaining 16 Kbyte of the display area are used for four different font areas, three of which are available in CGA Text mode and one of which is available in HGA Text mode. Refer to Section 3.4 “Text Mode,” in the *Élan™SC300 Microcontroller Programmer's Reference Manual*, for Text mode memory map examples.

Graphics Mode

In Graphics mode, the video SRAM memory map is determined by the LCD resolution programmed into the ÉlanSC300 microcontroller, the number of banks of memory selected, and the number of bits per pixel required. Refer to Section 3.3, “Graphics Mode,” in the *Élan™SC300 Microcontroller Programmer's Reference Manual*, for Graphics mode memory map examples.

In Graphics mode, the video SRAM memory is split into 8-Kbyte banks. Either two or four 8-Kbyte banks are used. Refer to "Steps for Determining if a Single-Screen LCD Panel can be Driven by the ÉlanSC300 Microcontroller" on page 7 to determine how many banks are needed. The display rows are split evenly across the banks, as illustrated in the tables on the following page.

The number of bits per pixel and the programmed screen width determine how many bytes of data per row are needed. For example, a 128-pixel-wide screen, at 2 BPP needs $2 \cdot 128 = 256$ bits of data per row, and $256 \div 8 = 32$ bytes of data per row. At 2 bits per pixel, each pixel can be set to one of four colors. A 128-pixel-wide screen at 1 BPP needs $1 \cdot 128 = 128$ bits of data per row, and $128 \div 8 = 16$ bytes of data per row. At 1 BPP, each pixel can be set to only one of two colors. Thus, more colors require more memory.

Note: *With smaller panels, it may be necessary to program the ÉlanSC300 microcontroller's graphics controller with a logical screen width greater than the physical screen width of the LCD panel to meet the panel's refresh rate requirements. Reminder: if this is done, it is the logical screen width programmed into the graphics controller that determines the number of bytes per row.*

The following tables illustrate the memory map for a 128 x 128 pixel display set up for 2 BPP, in both a two-bank and four-bank configuration. As shown in the tables, a 128-pixel-wide display at 2 BPP requires 32 bytes of memory per row. If the width of the display were to increase or decrease, the change would cause the row ending address and subsequent starting addresses to increase or decrease, respectively.

Table 1. Memory Map of 128 x 128 Display Using Two Banks in Graphics Mode

Row Starting Address	Display Row	Row Ending Address
BANK 1 (B8000–B9FFF)		
B8000	0	B801F
B8020	2	B803F
B8040	4	B805F
...	...	...
B87E0	126	B87FF
BANK 2 (BA000–BBFFF)		
BA000	1	BA01F
BA020	3	BA03F
BA040	5	BA05F
...	...	...
BA7E0	127	BA7FF

Table 2. Memory Map of 128 x 128 Display Using Four Banks in Graphics Mode

Row Starting Address	Display Row	Row Ending Address
BANK 1 (B8000–B9FFF)		
B8000	0	B801F
B8020	4	B803F
B8040	8	B805F
...	...	...
B83E0	124	B83FF
BANK 2 (BA000–BBFFF)		
BA000	1	BA01F
BA020	5	BA03F
BA040	7	BA05F
...	...	...
BA3E0	125	BA3FF
BANK 1 (BC000–BCFFF)		
BC000	2	BC01F
BC020	6	BC03F
BC040	10	BC05F
...	...	...
BC3E0	126	BC3FF
BANK 2 (BE000–BEFFF)		
BE000	3	BE01F
BE020	7	BE03F
BE040	11	BE05F
...	...	...
BE3E0	127	BE3FF

HOW THE ÉlanSC300 MICROCONTROLLER DISPLAYS COLORS AS GRAY SHADES

The ÉlanSC300 microcontroller uses gray shading to represent colors on a monochrome LCD panel. It does this by associating a duty cycle with a particular IRGB color value (see Table 3). A duty cycle of x ÷ y indicates that for every y times the LCD display is refreshed, the pixel will be displayed x times.

Video Index 19H, bit 4 enables color mapping for CGA mode. This allows the mapping of one IRGB/gray scale value to another IRGB/gray scale value that looks better for a particular LCD. In this way, applications that use colors that do not look good on a particular display can still be used without having to change the application.

Table 3. Color Codes Mapped to Pixel Duty Cycles

Color	IRGB/Gray Scale	Duty Cycle	Common Denominator
Black	0000	0	0/24
Blue	0001	1/8	3/24
Green	0010	1/2	12/24
Cyan	0011	2/3	16/24
Red	0100	1/3	8/24
Magenta	0101	3/8	9/24
Brown	0110	3/4	18/24
Light Gray	0111	7/8	21/24
Dark Gray	1000	1/6	4/24
Light Blue	1001	1/4	6/24
Light Green	1010	5/6	20/24
Light Cyan	1011	3/4	18/24
Light Red	1100	1/2	12/24
Light Magenta	1101	2/3	16/24
Yellow	1110	7/8	21/24
White	1111	1	24/24

SINGLE-SCREEN LCD SUPPORT

A number of factors, including physical properties and mode of operation, determine whether or not the ÉlanSC300 microcontroller is capable of driving a specific display.

Physical Properties

- 4-bit data interface

  The display to be driven must have 4 data lines to connect to the ÉlanSC300 microcontroller's LCDD3–LCDD0

- Panel resolution

  The resolution, combined with the mode of operation described in the next section, determines the rate at which the ÉlanSC300 microcontroller will refresh the display, as well as the amount of video SRAM used (the ÉlanSC300 microcontroller supports a maximum of 32 Kbyte of video SRAM). The video SRAM becomes a limiting factor for supporting the display only when a Graphics mode is used.

- Refresh rate

  The rate at which the ÉlanSC300 microcontroller refreshes the display is affected by the mode of operation as well as the panel resolution. If a panel's refresh rate is not specifically stated, a maximum value can be approximated by taking:

  4 ÷ ((shift clock period) · (display resolution))

  The "4" comes from the number of pixel data bits per shift clock. On the ÉlanSC300 microcontroller, this number *must* be 4. For example, a 320 x 240 display that has a minimum shift clock period of 125 ns, will have a maximum refresh rate of 4 ÷ (125 ns · 320 · 240) = 416 Hz. A typical refresh value is around 75 Hz. A value less than 60 Hz will probably yield unacceptable results (e.g., the screen will appear very dim or will blink).

Mode of operation

The mode of operation affects the refresh rate of the display, and when a Graphics mode is selected, determines the amount of video SRAM required. When the ÉlanSC300 microcontroller is configured for a Graphics mode, the 32 Kbyte of memory is divided into four 8-Kbyte banks. Video Index register 09H controls whether the display information is divided between just two or all four of the memory banks.

- CGA Mode

  When CGA mode is selected, the ÉlanSC300 microcontroller uses a pixel clock frequency of 9.557 MHz. The refresh rate for Text mode, or 2-color Graphics mode can be calculated by dividing the pixel clock frequency by the screen resolution programmed into the video controller, taking into account the 20 extra pixel clocks per line mentioned earlier. The refresh rate for 4-color Graphics mode will be half that value.

- HGA Mode

  When HGA mode is selected, the ÉlanSC300 microcontroller uses a pixel clock frequency of 14.336 MHz. The refresh rate for Text mode or 2-color Graphics mode can be calculated by dividing the pixel clock frequency by the screen resolution programmed into the video controller, taking into account the 20 extra pixel clocks per line mentioned earlier.

STEPS FOR DETERMINING IF A SINGLE-SCREEN LCD PANEL CAN BE DRIVEN BY THE ÉlanSC300 MICROCONTROLLER

1. Does the panel have a 4-bit data interface?
   - Yes: Go to step 2.
   - No: Panel cannot be driven by the ÉlanSC300 microcontroller.
2. Determine what rate the ÉlanSC300 microcontroller will refresh the LCD panel. Refresh rate is a factor of the LCD panel resolution and pixel clock frequency, which is determined by the mode of operation. For a panel size of W x H, and a bits-per-pixel value of BPP (1 or 2), the refresh rate will be as follows:

   (pixel clock frequency) ÷ ((W + 20) · BPP · H)

   In CGA mode, the pixel clock frequency is 9.557 MHz. In HGA mode the pixel clock frequency is 14.336 MHz. Therefore, when the ÉlanSC300 microcontroller is configured for a 320 x 240, 1-BPP resolution LCD in CGA mode, the refresh rate is 9.557 MHz ÷ (340 · 1 · 240) = 117 Hz. When the ÉlanSC300 microcontroller is configured for the same size display in HGA mode, the refresh rate is 14.336 MHz ÷ (340 · 1 · 240) = 175 Hz. Note that 2-color (1 BPP) mode can be supported by either CGA or HGA, but the HGA refresh rate is approximately 50% higher.
3. Does the refresh rate fall in the range supported by the LCD panel?
   - Yes: Go to step 4.
   - No: Panel cannot be driven by the ÉlanSC300 microcontroller.

Note: *If the refresh rate driven by the ÉlanSC300 microcontroller is too fast for the panel, it is possible to effectively slow the refresh rate to the panel by programming the ÉlanSC300 microcontroller's graphics controller using a logical panel width that is greater than the physical LCD panel width. If this is done, the logical panel width must be used for all calculations, and it should first be tested on the same panel before making the design choice. This will determine if the panel can handle the additional shift clocks sent to it as a result of the larger panel width.*

4. Does the ÉlanSC300 microcontroller support the LCD panel's video SRAM requirements?

   This is a concern only if the LCD will be operated in a Graphics mode; therefore, Text mode is not discussed.

   2 bits per pixel (four shades of gray)
   This mode consumes the most memory. Each bank of memory is 8 Kbyte (8192 bytes). In 2 BPP mode, each byte contains the equivalent of 4 pixels of information. Each bank can store information for 32,768 pixels. Therefore, if configured for 4 banks of memory, the maximum number of pixels that the ÉlanSC300 microcontroller can support is 4 · 32,768 = 131,072 pixels. If configured for two banks of memory, the maximum number of pixels that the ÉlanSC300 microcontroller can support is 2 · 32,768 = 65,536 pixels.

   For example: a 320 x 240 LCD display has 320 · 240 = 76,800 pixels. Therefore, the ÉlanSC300 microcontroller will support this resolution LCD if configured for four banks of memory.

   1 bit per pixel (2 shades of gray)
   Each bank of memory is 8 Kbyte (8192 bytes). In 1 BPP mode, each byte contains the equivalent of 8 pixels of information. Each bank can store information for 65,536 pixels. Therefore, if configured for four banks of memory, the maximum number of pixels that the ÉlanSC300 microcontroller can support is 4 · 65,536 = 262,144 pixels. If configured for two banks of memory, the maximum number of pixels

that the ÉlanSC300 microcontroller can support is 2 · 65,536 = 131,072 pixels.
For example: a 640 x 400 LCD display has 640 · 400 = 256,000 pixels. Therefore the ÉlanSC300 microcontroller will support this resolution LCD if configured for four banks of memory.

5. If the LCD panel meets the criteria listed in the previous four steps, then the ÉlanSC300 microcontroller is capable of driving the LCD panel.

PROGRAMMING THE ÉlanSC300 MICROCONTROLLER TO WORK WITH A SINGLE-SCREEN LCD PANEL

Before programming the ÉlanSC300 microcontroller to work with a single-screen LCD panel, the following information is needed:

- Display resolution in pixels, W x H (e.g., 640 x 200)
- Mode of operation (i.e., CGA graphics-1 BPP, CGA graphics-2 BPP, CGA text, HGA graphics-1 BPP, HGA text)
- If Text mode, the font size (8 x FH for CGA or 9 x FH for HGA, where FH = font height; 8 x 8 is CGA standard, 9 x 14 is HGA standard)
- If Graphics mode, the number of 8-Kbyte banks of memory (see "Steps for Determining if a Single-Screen LCD Panel can be Driven by the ÉlanSC300 Microcontroller" on page 7)

Determine the Video Index Values to Be Programmed

First determine the Video Index values for Video Indexes 00–0Fh, based on the information listed in the four items above.

CGA & HGA Graphics Mode Values

- **(HT = W · BPP ÷ 16)**

  The Horizontal Total Register (Index 00H) and Horizontal Displayed Register (Index 01H) are programmed with the number of 16-bit words per display row. In 2-color mode (1 BPP), this equals the width of the display in pixels divided by 16. In 4-color mode (2 BPP), this equals two times the width of the display in pixels divided by 16. For example, a 320 x 240 display is 320 pixels wide. Therefore, in 2-color mode (1 BPP), HT = 320 ÷ 16 = 20 (14h).

- **(VT = H ÷ BANKS)**

  The Vertical Total Register (Index 04H), Vertical Display Register (Index 06H), and Vertical Sync Position Register (Index 07H) are programmed with the value of the height of the display in pixels, divided by the number of banks of memory. For example, a 320 x 240 display is 240 pixels high. If four banks of memory are going to be used, then VT = 240 ÷ 4 = 60 (3Ch).

- **(MSL = BANKS – 1)**

  The Max Scan Line Register (Index 09H) is programmed with the number of banks of memory to be used, minus 1. If four banks of memory are going to be used, then MSL = 4 – 1 = 3.

- **(SAH) (SAL)**

  The Start Address High (Index 0CH) and Start Address Low (Index 0DH) Video Indexes together form a 14-bit offset (bits 7 and 6 of Index 0CH are not used) that determines the location in video memory where the video controller will start fetching data to be displayed on the screen. Typically, these indexes will be set to 0 so the display data starts being fetched from B8000h. Scrolling can be implemented by changing the values in these registers.

- All the other index registers (02, 03, 05, 08, 0Ah, 0Bh, 0Eh, 0Fh) will be programmed with 0h.

CGA & HGA Text Mode Values

- **HT = W · BPP ÷ 16**

  The Horizontal Total Register (Index 00H) and Horizontal Displayed Register (Index 01H) are programmed with the number of characters per row. The character width is hard coded to be 8 pixels wide. Therefore, HT is the width of the display in pixels divided by 8. For example, a 320 x 240 display is 320 pixels wide. Therefore HT = 320 ÷ 8 = 40 (28h).

- **(VT = H ÷ Y)**

  The Vertical Total Register (Index 04H), Vertical Displayed Register (06H), and Vertical Sync Position Register (07H) are programmed with the value of the number of rows of characters. This is determined by the height of the display in pixels, divided by the height of the character in pixels (Y from the equation below). For example, a 320 x 240 display is 240 pixels high. If using an 8 x 10 font, then VT = 240 ÷ 10 = 24 (18h).

- **(MSL = Y –1)**

  The Max Scan Line Register (Index 09H) is programmed with the pixel height of the font – 1. For example, if using an 8 x 10 font, then MSL = 10 – 1 = 9.

■ **(CS = Cursor Start Row –1)**
(CE = Cursor End Row –1)

The Cursor Start Register (Index 0AH), and Cursor End Register (Index 0BH) define the pixel rows within a font where the cursor will be displayed. Typically, the cursor is defined to be near the bottom of the character matrix. For example, an 8 x 10 font has 10 pixel rows. Therefore, to have the cursor positioned at rows 8 and 9, CS = 8 – 1 = 7, and CE = 9 – 1 = 8.

Note: *Bits 6 and 5 of the Cursor Start Register (Index 0AH) control the cursor's behavior. See Table 4.*

Table 4. Cursor Control Bits (Index 0AH)

Bit 6	Bit 5	Cursor Behavior
0	0	No blinking
0	1	Cursor not displayed
1	0	Blinking with 16 · refresh rate
1	1	Blinking with 32 · refresh rate

■ **(SAH) (SAL)**

The Start Address High (Index 0CH) and Start Address Low (Index 0DH) Video Indexes together form a 14-bit offset (bits 7 and 6 of Index 0CH are not used) that determines the location in video memory where the video controller will start fetching data to be displayed on the screen. Typically, these indexes will be set to 0 so the display data starts being fetched from B8000h. Scrolling can be implemented by changing the values in these registers.

■ **(CAH) (CAL)**

Cursor Address High (Index 0EH) and Cursor Address Low (Index 0FH) Video Indexes together form a 14-bit offset (bits 7 and 6 of Index 0CH are not used) that determines the location in video memory where the video controller will display the cursor when the corresponding video data is displayed on the screen. For example, if SAH = 0, SAL = 0, CAH = 0, and CAL = 0, the cursor will be displayed in the upper left corner of the screen.

■ All the other index registers (02, 03, 05, 08, 0Ah, 0Bh, 0Eh, 0Fh) will be programmed with 0h.

Initializing the Video Controller

The following sequences show the proper sequence for initializing the video controller. The following steps assume the video controller is set up to its default mode of CGA and the Video Index registers are unlocked. (To unlock the video registers, write 12h to Port 3D4h, and read Port 3D5h with no I/O cycles between these.)

Note: *Only the minimum set of registers required to get the screen running are initialized. There are a number of additional features such as auto screen blanking and text truncation that are discussed in the* Élan™SC300 Microcontroller Programmer's Reference Manual *that are not reiterated in this document.*

CGA Graphics

The Video Index registers are accessed by writing the register index to Port 3D4h, and then writing or reading the register data to or from Port 3D5h.

1. Set Video Index 18H = 50h
   – Sets up LCD panel type to single screen (bits 7 and 6)
   – Sets display type to LCD (bit 4)
2. Set Video Index 00H = HT
3. Set Video Index 01H = HT
4. Set Video Index 02H = 0
5. Set Video Index 03H = 0
6. Set Video Index 04H = VT
7. Set Video Index 05H = 0
8. Set Video Index 06H = VT
9. Set Video Index 07H = VT
10. Set Video Index 08H = 0
11. Set Video Index 09H = MSL
12. Set Video Index 0AH = 0
13. Set Video Index 0BH = 0
14. Set Video Index 0CH = SAH
15. Set Video Index 0DH = SAL
16. Set Video Index 0EH = 0
17. Set Video Index 0FH = 0
18. Set Video Index 20H (bit 5 controls dot doubling)
   – If 2 BPP Graphics mode is used, set bit 5 = 1 to disable dot doubling.

Note: *This is the common setting.*

   – If 2 BPP Graphics mode is used, and dot doubling is desired (CGA standard), then set bit 5 = 0.

19. If Color Mapping is going to be used:
   – Set Video Index's 14H–17H and 1CH–1FH accordingly
   – Set Video Index 19H = 10H
20. Set Video Port 3D8H
   – If 2 BPP Graphics mode, set to 0AH
   – If 1 BPP Graphics mode, set to 1AH
21. Set Video Port 3D9H
   – Set colors accordingly (see the register description in the *Élan™SC300 Microcontroller Programmer's Reference Manual*)

CGA Text

The Video Index registers are accessed by writing the register index to Port 3D4h and then writing or reading the register data to or from Port 3D5h.

1. Set Video Index 18H = 50h
   - Sets up LCD panel type to single screen (bits 7 and 6)
   - Sets display type to LCD (bit 4)
2. Set Video Index 00H = HT
3. Set Video Index 01H = HT
4. Set Video Index 02H = 0
5. Set Video Index 03H = 0
6. Set Video Index 04H = VT
7. Set Video Index 05H = 0
8. Set Video Index 06H = VT
9. Set Video Index 07H = VT
10. Set Video Index 08H = 0
11. Set Video Index 09H = MSL
12. Set Video Index 0AH = CS
13. Set Video Index 0BH = CE
14. Set Video Index 0CH = SAH
15. Set Video Index 0DH = SAL
16. Set Video Index 0EH = CAH
17. Set Video Index 0FH = CAL
18. If Color Mapping is going to be used:
    - Set Video Indexes 14H–17H and 1CH–1FH accordingly
    - Set Video Index 19H = 10H
19. Set Video Port 3D8H = 08H
    - Enable Text mode
20. Set Video Port 3D9H
    - Set colors accordingly (see the register description in the *Élan™SC300 Microcontroller Programmer's Reference Manual*)

HGA Graphics

Initially, the Video Index registers are accessed by writing the register index to Port 3D4h, and then writing or reading the register data to or from Port 3D5h. This occurs until the controller is switched to HGA mode. After this occurs, Port 3B4 is used for the index and Port 3B5 is used for the data.

1. Set Video Index 18H = 51h (use Ports 3D4 and 3D5)
   - Sets Graphics mode to HGA (from now on use ports 3B4 and 3B5)
   - Sets up LCD panel type to single screen (bits 7 and 6)
   - Sets display type to LCD (bit 4)
2. Set Video Index 00H = HT
3. Set Video Index 01H = HT
4. Set Video Index 02H = 0
5. Set Video Index 03H = 0
6. Set Video Index 04H = VT
7. Set Video Index 05H = 0
8. Set Video Index 06H = VT
9. Set Video Index 07H = VT
10. Set Video Index 08H = 0
11. Set Video Index 09H = MSL
12. Set Video Index 0AH = 0
13. Set Video Index 0BH = 0
14. Set Video Index 0CH = SAH
15. Set Video Index 0DH = SAL
16. Set Video Index 0EH = 0
17. Set Video Index 0FH = 0
18. Set Video Port 3BFH = 01H
    - Master enable for Graphics mode
19. Set Video Port 3B8H = 0AH
    - Enable Graphics mode
    - Enable video display

HGA Text

Initially, the Video Index registers are accessed by writing the register index to Port 3D4h, and then writing or reading the register data to or from Port 3D5h. This occurs until the controller is switched to HGA mode. After this occurs, Port 3B4 is used for the index and Port 3B5 is used for the data.

1. Set Video Index 18H = 51h (Use ports 3D4 and 3D5)
   - Sets Graphics mode to HGA (from now on use Ports 3B4 and 3B5)
   - Sets up LCD panel type to single screen (bits 7 and 6)
   - Sets display type to LCD (bit 4)
2. Set Video Index 00H = HT
3. Set Video Index 01H = HT
4. Set Video Index 02H = 0
5. Set Video Index 03H = 0
6. Set Video Index 04H = VT
7. Set Video Index 05H = 0
8. Set Video Index 06H = VT
9. Set Video Index 07H = VT
10. Set Video Index 08H = 0
11. Set Video Index 09H = MSL
12. Set Video Index 0AH = CS
13. Set Video Index 0BH = CE
14. Set Video Index 0CH = SAH
15. Set Video Index 0DH = SAL
16. Set Video Index 0EH = CAH
17. Set Video Index 0FH = CAL
18. Set Video Port 3BFH = 00H
    - Master disable for Graphics mode
19. Set Video Port 3B8H = 08H
    - Enable video display

DUAL-SCREEN LCD SUPPORT

The ÉlanSC300 microcontroller supports only one type of dual-screen LCD display. The display must be 640 x 200 (640 x 100 per panel). The display must accept 8 data bits total (4 data bits per panel). The ÉlanSC300 microcontroller will support the panel in CGA mode only and refresh the display at a rate of approximately 75 Hz.

The reason for the limited dual-screen support is that the ÉlanSC300 microcontroller does not have any dual-screen registers that, among other things, allow you to program where the fetching of data for the lower panel should begin. This was hard-wired in the ÉlanSC300 microcontroller.

Programming the ÉlanSC300 Microcontroller for Dual-Screen LCD Support

The programming values for this dual-screen LCD are identical to the ones used for the single-screen panel with two exceptions. Video Index 18H bits 7 and 6 should be set to 00 instead of 01. The ÉlanSC300 microcontroller Configuration Register B1H bit 1 must be set to 1 to enable the additional 4 lower-panel LCD data bits.

The Physical Pin Connections to the ÉlanSC300 Microcontroller

To connect an LCD panel to the ÉlanSC300 microcontroller, the following pins need to be connected:

LCDD3–LCDD0: These are the panel's data bits. LCDD3 is the MSB; LCDD0 is the LSB. Find out which data bit on the LCD is the MSB, which is the LSB, and connect the pins accordingly.

CP1: Also known as the line clock or latch (see definition in "Line Clock (CP1)" on page 2). This should be connected to the equivalent line on the LCD panel.

CP2: Also known as the shift clock or data shift (see definition in "Shift Clock (CP2)" on page 2). This should be connected to the equivalent line on the LCD panel.

FRM: Also known as FLM, or frame (see definition in "Frame Start (FRM)" on page 2). This should be connected to the equivalent line on the LCD panel.

M: AC modulation (see definition in "LCD Panel AC Modulation (M)" on page 2). Some panels require this signal, others do not. Connect this signal if appropriate.

LCDDL3–LCDDL0: These data bits are required only if a dual-screen LCD panel is being connected. These are the data bits for the lower panel. LCDDL3 is the MSB for the lower panel, and LCDDL0 is the LSB for the lower panel.

Other LCD pin connections

The other connections that need to be made to an LCD vary. For example:

- **Contrast voltage**: can be positive or negative, typically about –22 volts.
- +5 V
- GND
- **Display enable**: This is usually a simple 5-V enable signal that some panels require. This can easily be connected to one of the ÉlanSC300 microcontroller's PMC pins.
- $\mathbf{V_{ee}}$: Typically a voltage in the same range as the contrast voltage. Refer to panel specifications.

Using the ÉlanSC300 Microcontroller Evaluation Board to Test an LCD Panel

The ÉlanSC300 microcontroller evaluation board has a 20-pin header (P18) with all the LCD interface signals connected to it. The evaluation board provides a –14 V to –16 V contrast voltage (adjust using VR1) and a –17 V_{ee} voltage. If the panel to be tested requires voltages other than these, a separate circuit will need to be bread boarded, or an external DC power supply used. The following evaluation board jumpers and resistor packs should also be set to configure the evaluation board for Internal Video mode: JP16 = 1–2; JP18 = Open; Install RP3 and RP4 (RP1, RP2, RP5, and RP6 must be empty).

Note that the evaluation board BIOS is set up to configure for a 640 x 200 single-panel display when the ÉlanSC300 microcontroller is in Internal Video mode. If a panel size other than this is used without modifying the BIOS, the screen may not display anything at boot-up. Refer to "LCD Development Tips and Tricks" for instructions on how to use CTTY for console activity while developing a screen application.

Also note that if a dual-screen LCD panel is used, the $\overline{\text{SBHE}}$, IRQ14, MCS16, and IOS16 ISA signals are no longer available outside the chip. This means the standard 16-bit IDE interface cannot be used.

Table 5 shows how the pins on the 20-pin header should be connected to the pins of a Hitachi LMG6272XNFR, 640 x 200 single-panel display.

Table 5. Pin Connections

The ÉlanSC300 Microcontroller Evaluation Board P18 Header Pins				HITACHI: LMG6272XNFR 640 x 200 Single Panel LCD		
Pin #	**Pin Name**	**Function**		**Pin #**	**Pin Name**	**Function**
1	VccLCD5	+5	=	10	Vdd	Power supply for logic circuit
2	GND	Ground	=	11	Vss	Ground
3	LCDFRM	Frame start	=	5	FLM	The FLM signal indicates the start of each display cycle
4	LCDCP1	Line clock	=	7	CL1	Data latch
5	LCDCP2	Shift clock	=	8	CL2	Data shift
6	LCDM	AC modulation		x		Not used
7	LCDD0	Data bit 0 (LSB)	=	1	D0	Data bit 0 (LSB)
8	LCDD1	Data bit 1	=	2	D1	Data bit 1
9	LCDD2	Data bit 2	=	3	D2	Data bit 2
10	LCDD3	Data bit 3 (MSB)	=	4	D3	Data bit 3 (MSB)
11	LCDDL0	Dual-screen panel data bit 0 for lower panel		x		Not used
12	LCDDL1	Dual-screen panel data bit 1 for lower panel		x		Not used
13	LCDDL2	Dual-screen panel data bit 2 for lower panel		x		Not used
14	LCDDL3	Dual-screen panel data bit 3 for lower panel		x		Not used
15	Contrast	Adjustable voltage between –14 V to –16 V	=	9	Vo	LCD driving voltage
16	Vee	–17 V	=	12	Vee	Power supply for LC driving
17—20	NC	Not connected				

LCD DEVELOPMENT TIPS & TRICKS

Use CTTY for Console I/O

When using an LCD as the console device under DOS, there may be problems debugging the LCD because DOS will also be attempting to write to the LCD through the BIOS. A way around this is by diverting the DOS console I/O through the serial port. This allows use of debug or other application programs to access the ÉlanSC300 microcontroller's registers and view the results on the LCD display. The following steps are recommended:

1 Connect a null modem cable between COM1 on the ÉlanSC300 microcontroller and a PC running PROCOMM or other terminal program. (A lap link serial cable is a null modem cable).

2. On the PC running PROCOMM, set up for a direct connect to the serial port, running 9600 baud, 8 data, 1 stop bit, and no parity. Set it up for a terminal type of either TTY or ANSI-BBS.
3. On the ÉlanSC300 microcontroller system, boot DOS and have the following commands in the **autoexec.bat** file:
   - mode com1 9600,n,8,1
   - ctty com1
4. After the system boots, the DOS prompt will be visible on the PC running PROCOMM.
5. Executing programs that use the console I/O such as debug should be possible.
6. Programs that talk directly to video memory will display the output on the ÉlanSC300 microcontroller system, but the keyboard input will come from the PC running PROCOMM.
7. Below are two things to keep in mind:
   - To debug an LCD, the ÉlanSC300 microcontroller evaluation board should be configured for Internal Video mode
   - If using Phoenix BIOS, the BIOS will lock the Video Index registers. Remember to unlock them by reading Video Index 12h. The BIOS will also lock the upper 16K of display memory, which is used for font storage in Text mode. This can be unlocked by clearing bit 3 of Video Index 20h.

The BIOS May Also Be Updating the Screen

If using the ÉlanSC300 microcontroller evaluation board with an LCD standalone, (i.e., not option 1 above) and booting to DOS, note that the BIOS is accessing the Video registers every time it is called to update the screen (i.e., typing in debug or from the DOS prompt). If trying to use debug or other programs to modify the Video registers, be aware that they may be modified by the BIOS as well, specifically Ports 3D8h and 3D9h.

Using Standard CGA or HGA Drivers on a Nonstandard Display Size

It is possible to use standard CGA or HGA drivers on a nonstandard-sized screen. The method is to program the ÉlanSC300 microcontroller with values for the LCD width that match those that the driver expects. The LCD height can be set to the actual LCD height. This will cause the video SRAM to be organized as the standard driver is expecting. The LCD panel will be displaying the upper left of a virtual screen.