Élan™SC400 Microcontroller and Windows® CE
µforCE™ Demonstration System

Reference Manual

NOTE: *This Document contains information on a product currently under development at Advanced Micro Devices, Inc. Advanced Micro Devices reserves the right to make changes to this specification at any time and without notice. No responsibility is assumed by Advanced Micro Devices for its use, nor for any infringements of patent, copyright, or other proprietary rights of third parties resulting from its use. No license is hereby granted under any patent, copyright, or other proprietary rights of Advanced Micro Devices or any third party.*

Élan™SC400 Microcontroller and Windows®CE
µforCE™ Demonstration System Reference Manual, Release 1.0

1.0

Advanced Micro Devices, Inc.
5204 E. Ben White Blvd.
Austin, TX 78741-7399

Contents

About the µforCE™ Demonstration System

Chapter 1

System Features and Components

Chapter 2

µforCE System Board Features and Components

1.0

Chapter 3

Microcontroller Signal Considerations

Chapter 4

Power Management

Chapter 5

Product Support

Index

List of Figures

List of Tables

1.0

About the μforCE™ Demonstration System

The μforCE™ demonstration system provides a reference/demonstration platform for mobile and embedded product development using the Élan™SC400 microcontroller and the Windows®CE operating system. High performance, small size, low cost, and low power consumption are the key features of the μforCE demonstration system. Figure 0-1 on page viii shows a block diagram of the μforCE system.

The μforCE system enables you to understand the functionality of an ÉlanSC400 microcontroller/Windows CE-based application.

1.0

NOTE: The μforCE system is for reference and demonstration purposes only. Extended development of ÉlanSC400 microcontroller/Windows CE operating system designs requires additional tools available from AMD, Microsoft, and bsquare. For information on how to order these tools, see Chapter 5, “Product Support”.

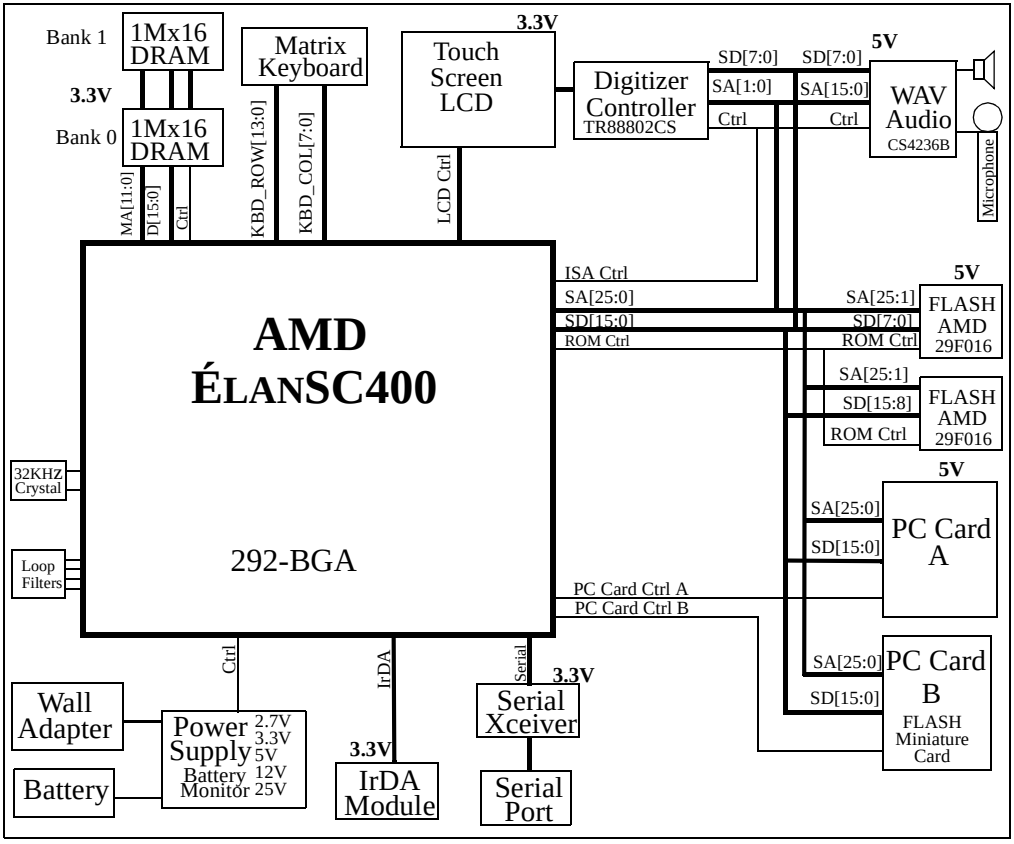

Figure 0-1. µforCE Demonstration System Block Diagram

µforCE System Features

The µforCE system provides the following features:

- Small, form-factor LCD
- Matrix keyboard
- AC/DC wall-adapter power supply
- NiMH battery-pack power supply
- Small, form-factor demonstration board with:
  - ÉlanSC400 microcontroller
  - 2M x 16 DRAM
  - 2M x 16 Flash memory
  - Matrix keyboard interface
  - LCD interface
  - Resistive-digitizer overlay
  - WAV audio
  - PC card socket
  - Miniature Flash memory socket
  - IrDA infrared module
  - Serial port for debug

1.0

Documentation

The Élan™SC400 Microcontroller and Windows® CE µ*forCE™ Demonstration System Reference Manual* provides information on the system and board features and functionality, system-specific considerations, and a description of power-management modes. Additional information can be found in the documents referenced on page xi.

About This Manual

Chapter 1, "System Features and Components" provides a description of the µforCE system features and components.

Chapter 2, "µforCE System Board Features and Components" provides descriptions of the board features including: microcontroller, DRAM, display, keyboard, ROM/Flash memory, PC card, serial port, IrDA, audio chip, and power supply.

Chapter 3, "Microcontroller Signal Considerations" provides information about system-specific considerations for the ÉlanSC400 microcontroller used in the µforCE system.

Chapter 4, "Power Management" describes the Power Management Unit (PMU) that is used to control chip and system power.

Chapter 5, "Product Support" provides information on reaching and using the AMD Corporate Applications technical support services, product information available through AMD's WWW and FTP sites, ordering information for µforCE system support tools, and support tools for the embedded E86™ families.

1.0

Suggested Reference Material

For information on ordering the literature listed below, see Chapter 5, “Product Support”.

- *Élan™SC400 Microcontroller Data Sheet*
  Included in your kit
- *Élan™SC400 Microcontroller Register Set Reference Manual*
  Included in your kit
- *Am486® Microprocessor Software User’s Manual*
  Advanced Micro Devices, order #18497
- *ÉlanSC400 Microcontroller Evaluation Board User’s Manua*l
  Included in your kit
- *FusionE86℠ Catalog*
  Advanced Micro Devices, order #19255
- *FusionE86℠ Development Tools Reference CD*
  Advanced Micro Devices, order #20158

For current application notes and technical bulletins, see our WWW page at **http://www.amd.com**.

10

Chapter 1

System Features and Components

This chapter provides information about the µforCE system's features and components.

µforCE System Features

Key features of the demonstration system include small size, low cost, and low power consumption, which, combined with the ÉlanSC400 microcontroller, result in a high-performance system.

The ÉlanSC400 microcontroller used in the µforCE system contains a 66-MHz, Am486® microprocessor with the PC/AT system logic in a 292-pin Ball Grid Array (BGA) package. For more information about the ÉlanSC400 microcontroller, see the ÉlanSC400 documentation included in your kit.

Small size and low cost are possible because of the ÉlanSC400 microcontroller's high level of integration; very few devices are needed on the system board to complete the system. Low power consumption results from the ÉlanSC400 microcontroller's extensive power-management capabilities.

µforCE System Components

The µforCE system consists of five main components:

- Small, form-factor Liquid Crystal Display (LCD)
- Matrix keyboard
- AC/DC wall-adapter power supply
- NiMH battery-pack power supply
- System board

Liquid Crystal Display (LCD)

The LCD on the µforCE system is a 5.4", diagonal, 480 x 320 resolution, single-scan ALPs display panel using black-and-white super-twisted neumatic (STN) technology. This LCD has a pixel- resolution resistive-digitizer overlay that allows pen input to the system. Note that because of power considerations, the LCD is reflective; no backlight is provided.

1.0

The LCD specifications are listed in Table 1-1.

Table 1-1. LCD Specifications

Model number	KHABBA904-A
Display format	480 x 320
Dot pitch	0.24 x 0.24
Dot size	0.22 x 0.22
Outline dimension	135.2 x 94.8 x 6.3 mm (5.3" x 3.7" x 0.25")
Viewing area	120.2 x 81.8 mm (4.7" x 3.2")

Matrix Keyboard

The µforCE system uses a Fujitsu matrix keyboard (model #N860-1406-T001). The keyboard connects directly to the ÉlanSC400 microcontroller's matrix keyboard interface. Software uses seven column signals and fourteen row signals to scan the keyboard for a key press.

Power Supply

There are two options for applying power to the µforCE system: the AC/DC wall adapter or the battery pack.

AC/DC Wall Adapter

The AC/DC wall-adapter power supply converts AC power to DC power using a universal power supply. The AC/DC wall-adapter power supply accepts 100–250 V AC and converts it to 12 V DC, 1.2 A maximum for the system's power supplies. The AC/DC wall-adapter power supply provides enough power to operate the system at top speed with no power management and with a PCMCIA card in the system. This power supply will also fast charge the battery pack while the system is in Suspend mode.

1.0

Battery Pack

The µforCE system can also run from batteries. The system includes a battery pack consisting of four AA, nickel metal hydride (NiMH) batteries. The battery pack provides a nominal 4.8-V, 1250-mAh power source. The batteries are charged by the system when power from the wall plug is available.

µforCE System Board

The µforCE system includes a small, form-factor board (3" x 5.75") containing the system logic. The system board provides the following features:

- ÉlanSC400 microcontroller
- 2M x 16 DRAM (expandable to 8Mbyte, 10Mbyte, or 16Mbyte)
- 2M x 16 Flash memory to contain the operating system and applications
- Matrix keyboard interface
- LCD interface
- Resistive-digitizer overlay
- WAV audio
- One PC card socket
- One Flash Miniature Card socket
- IrDA infrared module
- Serial port for debug
- Power supply that operates from a wall adapter or batteries

For a detailed description of the µforCE system board components see Chapter 2, “µforCE System Board Features and Components”.

1.0

Chapter 2

µforCE System Board Features and Components

This chapter provides information about the µforCE system board features and components. Figure 2-1 on page 2-2 shows the layout of the µforCE demonstration system board.

1.0

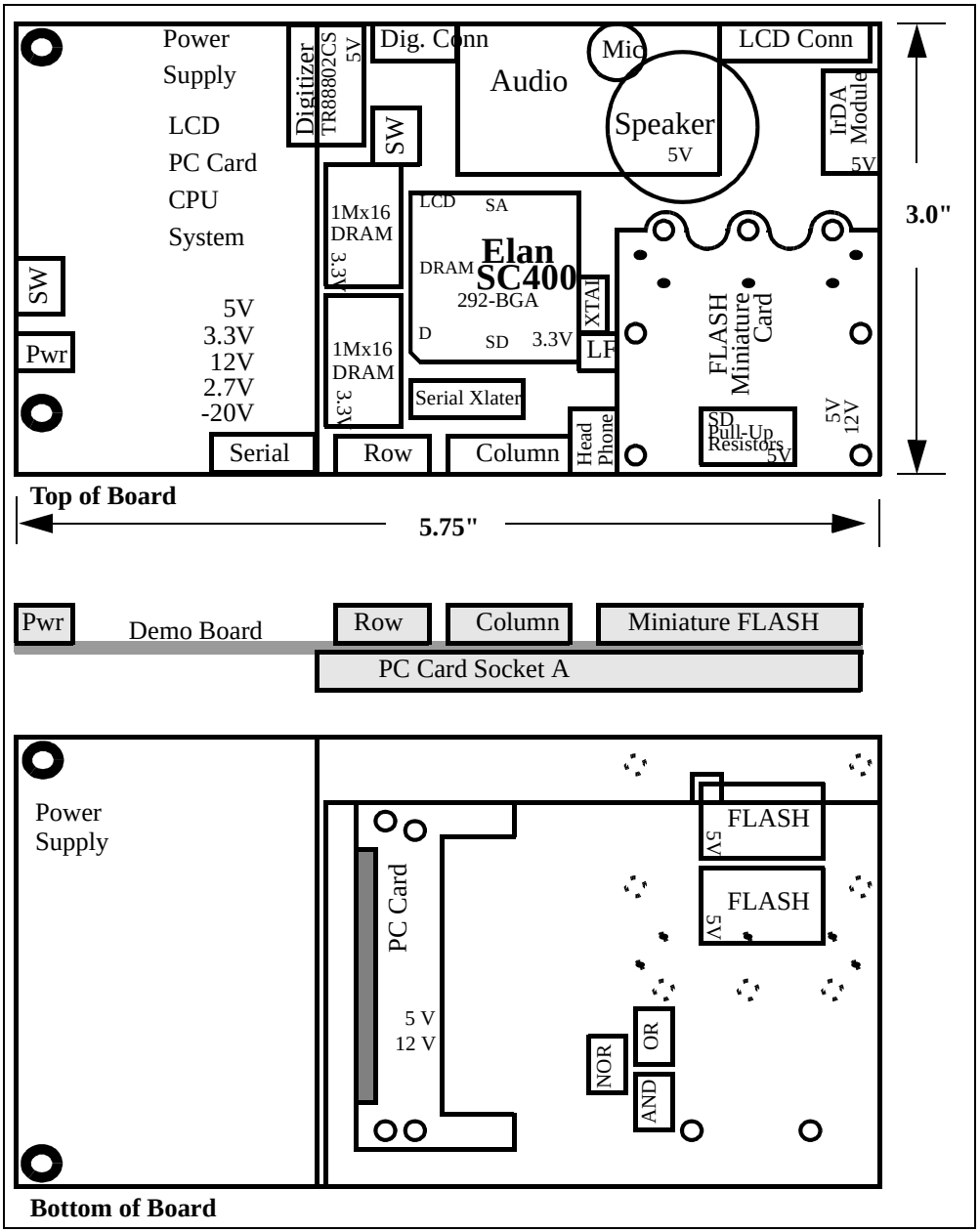

Figure 2-1. µforCE System Board Layout

1.0

ÉlanSC400 Microcontroller

The ÉlanSC400 microcontroller is the main chip on the µforCE system board. This highly-integrated, low-power microcontroller provides direct control of the following system features:

- DRAM
- Flash memory
- Matrix keyboard
- PC card sockets
- Serial port
- IrDA infrared port
- ISA bus devices
- Power switching

The ÉlanSC400 microcontroller is provided in a 292-pin BGA package that is soldered directly onto the system board. See Chapter 3, “Microcontroller Signal Considerations” for more information about ÉlanSC400 microcontroller system control.

1.0

DRAM

The DRAM in the µforCE system is designed for maximum flexibility. The base system is populated with two 1M x 16 DRAMs, one on bank 0 and one on bank 1, providing 4 Mbyte of memory for Windows CE. Offset pads are on the board to allow a 4M x 16 DRAM to be populated in either bank. Memory options on the board are as shown in Table 2-1 on page 2-4.

Table 2-1. DRAM Options

Bank 0	Bank 1	Total Memory
1M x 16	x	2 Mbyte
1M x 16	1M x 16	4 Mbyte (default)
4M x 16	x	8 Mbyte
1M x 16	4M x 16	10 Mbyte
4M x 16	1M x 16	10 Mbyte
4M x 16	4M x 16	16 Mbyte

The DRAM chips are 3.3 V for low power. Self-refresh DRAMs are used to reduce the Suspend mode currents. The x16-data-bit DRAM interface allows matrix-keyboard and internal-graphics operation of the ÉlanSC400 microcontroller.

Display

The µforCE system display is provided by a monochrome LCD with a resistive-touch screen overlay.

1.0

LCD

The ÉlanSC400 microcontroller provides the control signals for the LCD and switches the LCD voltages correctly. Table 2-2 contains the signals and signal descriptions.

Table 2-2. LCD Signals

Signal	Description
M	**LCD Panel AC Modulation** is the AC modulation signal for the LCD. AC modulation causes the LCD panel drivers to reverse polarity to prevent an internal DC bias from forming on the panel.
FRM	**LCD Panel Line Frame Start** is asserted by the chip at the start of every frame to indicate to the LCD panel that the next data clocked out is intended for the start of the first scan line on the panel. Some panels refer to this signal as FLM or S (scan start up).
SCK	**LCD Panel Shift Clock** is the nibble/byte strobe used by the LCD panel to latch a nibble or byte of incoming data. Commonly referred to by LCD panels as CL2 or CP2.
LC	**LCD Panel Line Clock** is activated at the start of every pixel line. Commonly referred to by LCD data sheets as CL1 or CP1.
LCDD3–LCDD7	**LCD Panel Data bits:** LCDD3–LCDD0 are data bits for the LCD panel interface. When driving 4-bit single-scan panels, bits 3–0 form a nibble-wide LCD data interface.
DISP_ON	**Display On**: Because the ÉlanSC400 microcontroller doesn't provide a DISP_ON signal, it is generated on the board using LVEE through a resistor and capacitor to delay the signal.

V_{DD} for the LCD is 3.3 V and is switched on and off by the ÉlanSC400 microcontroller's $\overline{\text{LVDD}}$ signal using a Linear Tec LTC1478 voltage switch.

V_{EE} for the LCD is nominally +28 V. It is generated by a Micrel MIC3172 and switched on and off the ÉlanSC400 microcontroller's $\overline{\text{LVEE}}$ signal using a FET switch circuit.

For more information about the LCD, see "Liquid Crystal Display (LCD)" on page 1-2.

For more information about the ÉlanSC400 microcontroller's LCD control signals, see the ÉlanSC400 microcontroller documentation included in your kit.

Touch Screen

The LCD has a resistive-touch screen overlay that interfaces to the system board through a four-signal flex cable.

The touch screen is controlled by a TriTech TR88802CS Pen Input Processor. This controller is connected to the ISA bus and is chip selected by the ÉlanSC400 microcontroller's GPIO_CS1 signal. GPIO_CS1 is programmed as an I/O chip select at address 300h–303h, and is qualified with $\overline{\text{IOR}}$.

The ÉlanSC400 microcontroller's PIRQ1 signal is used as the digitizer interrupt to the system when new data from the controller is available. To get valid data, the digitizer controller must be read while PIRQ1 is Low. GPIO_CS2 is used as an input to the ÉlanSC400 microcontroller to receive the digitizer's signal indicating the stylus is in contact with the touch panel.

Keyboard

1.0

A matrix keyboard connected directly to the ÉlanSC400 microcontroller's matrix keyboard controller interface provides the system keyboard input. Fourteen row signals and seven column signals are used to scan the keyboard for a key press.

When no key is pressed, the column signals are driven Low and an interrupt is generated by a row going Low from a key press. While a key is pressed, the keyboard timer is used to interrupt the system and allow key scanning.

ROM/Flash Memory

The µforCE demonstration system board contains 4 Mbyte of Flash memory to hold the system HAL, operating system, and applications. The Flash memory is configured with a x16-data-bit interface to optimize system performance. Two AMD 2M x 8 Flash memory chips (AM29F016s) are used in parallel, one on SD7:0, the other on SD15:8l.

PC Card

The µforCE system board supports one standard Type II PC card socket and one Flash Miniature Card socket. Because these sockets are unbuffered, the system only supports 5-V PC cards.

Note that the Flash Miniature Card socket is not supported by version 1.0 of Windows CE.

A jumper (or switch) on the system board allows you to boot the system from the PC card, allowing easier debugging and loading of the on-board Flash memory devices.

Serial Port

The ÉlanSC400 microcontroller's internal UART is available for use as a serial port on the system board. The UART is shared between the serial port and the IrDA so only one is available at any time. The serial port is provided as a 10-pin header that a ribbon cable serial connector plugs into. (This ribbon cable has a 10-pin, 1/10th center, 2 x 5 connector on one end, and a 9-pin D-shell serial connector on the other end.) The serial port is buffered with a standard level-translating serial transceiver.

1.0

IrDA

The µforCE system board provides for IrDA communications of up to a 1-Mbyte transfer rate.

Audio

The system board contains a Crystal Semiconductor Audio chip (model# CS4236B). Both the dynamic speaker and the microphone are connected, but the microphone is not supported by Windows CE. This chip is connected to the ISA bus with one IRQ and one DMA channel.

Power Supply

The power supply provides the voltages needed by the system board: 2.7 V, 3.3 V, 5 V, 12 V, and +28 V. The power supply draws its power from either an AC wall plug that provides +12 V, or a four-AA-cell NiMH battery pack.

Chapter 3

Microcontroller Signal Considerations

This chapter provides system-specific considerations for the ÉlanSC400 microcontroller signals and signal descriptions.

1.0

Signal Considerations

Table 3-1 summarizes the special signal considerations in this system.

Table 3-1.Microcontroller Signal Considerations

Pin Name	Signal Name	Low	High	Notes
GPIO_CS0	SUSPDIS#	Suspend mode	Reset and operation	Low to disable devices in Suspend mode: - Serial Port transceiver, disabled when Low - VCCSUSP - off when Low Provides VCC to: - SD pull-up resistors - Audio analog circuits (Mic, speaker, headphone)
Not a controller signal. Invert on the system board.	SUSPDIS	Reset and operation	Suspend mode	Inverted SUSPDIS# also available on system board. High to disable devices in Suspend mode: - IrDA LED's - Low-Power mode when High - Digitizer - Reset when High - Low power when High - VCC3 and VCC5 power supplies - Skip mode is the Low-Power mode of the power supplies
GPIO_CS1	DIG_CS#	Chip selected	Chip not selected	I/O chip select for digitizer Address qualified with $\overline{IOR}$
GPIO_CS2	PEN_OFF	Pen detected	Pen not detected	Input to indicate the pen is on the digitizer
GPIO_CS3	BATTEMP	Battery hot	Battery OK	Input from the battery temperature detection circuit. Normally High when battery temperature is OK; will fall when the battery crosses the temperature threshold. Use BATTEMP as a wakeup for Suspend to disable FSTCHRG#. Poll BATTEMP during operation and use to disable slow charge when it is too hot.

1.0

Pin Name	Signal Name	Low	High	Notes
GPIO_CS4	FSTCHRG#	Suspend mode	Reset, operation, and suspend	Low to enable fast charge of the batteries. High to enable slow charge of the batteries. Should slow charge the batteries when the system is operating (any PMU mode other than Suspend). Should fast charge the batteries if in Suspend mode. Need to disable fast charge in Suspend when the battery temperature peaks (BATTEMP). Has no affect when the AC wall plug is not connected.
GPIO_CS5	CHARGE#	Reset and enable charger	Disable charger circuit	Enable for battery charge circuit. Battery charge firmware uses this signal to enable/disable the charge on the battery for optimal charging.
PIRQ0	PIRQ0			Audio chip interrupt request
PIRQ1	PIRQ1			Digitizer chip new data request
PDRQ0 $\overline{\text{PDACK0}}$ AEN	PDRQ0 PDACK# AEN			Audio chip DMA
$\overline{\text{PCMA\_VCC}}$	PCMVCCA#	Reset, operation, and suspend	Operation and suspend	- Low to enable 5 V to PC card socket A. - High to disable 5 V to PC card socket A.
PCMA_VPP1 PCMA_VPP2	PCMVPP1A and PCMVPP2A	Operation and suspend	Reset, operation, and suspend	Control signals for PC card socket A VPP: PCMVPP1A PCMVPP2A Socket VPP 0 0 0V 0 1 5V 1 0 12V 1 1 Off PCMVPP1A is also used to enable the 12-V power supply. - High - 12 V enabled - Low - 12 V disabled (and supply in Low-Power mode).
PCMB_VCC	PCMVCCB#	Reset, operation, and suspend	Operation and suspend	Low to enable 5 V to PC card socket B. High to disable 5 V to PC card socket B.

1.0

Pin Name	Signal Name	Low	High	Notes
$\overline{\text{LVDD}}$	LVDD#	LCD enabled	Reset and LCD disabled	Switch 3.3 V on/off to LCD - High - LCD VCC disabled - Low - LCD VCC enabled
$\overline{\text{LVEE}}$	LVEE#	LCD enabled	Reset and LCD disabled	Switch LCD Contrast Voltage (22 V –28 V) on/off to LCD - High - LCD VEE disabled - Low - LCD VEE enabled $\overline{\text{LVEE}}$ is also inverted on the board to turn the LCD VEE power chip on/off
$\overline{\text{LBL2}}$	LBL2#	Critical Suspend mode	Reset and all other times	Used to hold the audio chip in reset during Critical Suspend mode. Reset is the hardware method of putting the audio chip in Low-Power mode. During Normal Suspend mode, the audio chip will be programmed into Low-Power mode; there is no time for an interrupt when $\overline{\text{BL2}}$ falls and forces Critical Suspend mode.
ACIN	ACIN	Battery power only	AC power available	Input to the ÉlanSC400 to indicate the wall plug is in use and PMU should be disabled.
$\overline{\text{BL0}}$–$\overline{\text{BL2}}$	BL0–BL2#	Battery power is low	Battery power is OK	$\overline{\text{BL0}}$ is the first level of battery power report. - Set to 5.0 V and causes battery-life report to show low $\overline{\text{BL1}}$ is the second level of battery power report - Set to 4.8 V and causes battery-life report to show very low $\overline{\text{BL2}}$ is the last level of battery power report - Forces the microcontroller into Critical Suspend mode - Set to 4.7 V

1.0

Signal Descriptions

In this section, the first name in each heading is the microcontroller pin name and the second name is the signal name on the µforCE system board.

GPIO_CS0/SUSPDIS#

The GPIO_CS0 signal is used to disable several devices external to the ÉlanSC400 microcontroller to save power when the microcontroller is in Suspend mode. GPIO_CS0 will be driven Low in Suspend mode, and will be driven High in all other PMU modes. To make this behavior automatic, GPIO_CS0 is programmed internally to GPIO_PMUA. When driving a GPIO_CS pin with an internal GPIO_PMU signal, the pin must be configured as an output. The GPIO_PMU signal is configured to be Low in Suspend mode, and High for all other PMU modes.

Table 3-2. GPIO_CS0 Register Considerations

Index[bits]	Description	Setting
AEh[3:0]	Map GPIO_PMUA to GPIO_CS0	0000
A0h[0]	Enable GPIO_CS0 as an output	1
AAh[5:0]	GPIO_PMUA Mode Change Register	11 1110
A6h[0]	GPIO_CS0 output clear	0

GPIO_CS1/DIG_CS#

The GPIO_CS1 signal is used as a chip select for the TriTech touch-overlay controller. GPIO_CS1 pulses Low when I/O reads occur from one of the four addresses required by the TriTech status registers. To support this operation, GPIO_CS1 is programmed internally to GPCSA. When using a GPIO_CS pin as a chip select, the pin must be configured as an output.

Software is responsible for selecting an otherwise unused address range at which to map the TriTech registers (300h–303h is used). The base address is configured via indices B4h[7:0] and B5h[1:0]. The number of addresses decoded from the base is configured via index B5h[5:2]. In our case, only bits 2 and 3 of this bit field will be cleared to allow chip-select generation for offsets 0–3 from the base address. In addition, the chip select should only be asserted on I/O reads because these are read-only registers; we qualify the addresses with $\overline{IOR}$, using index B8h[1:0].

The TriTech X,Y coordinate registers will be accessed as follows:

- 300h: Y coordinate bits 1:0
- 301h: Y coordinate bits 9:2
- 302h: X coordinate bits 1:0
- 303h: X coordinate bits 9:2

This arrangement allows software to do an **IN AX,DX** instruction followed by a **SHR AX,6** to make the digitizer value 0 based. Table 3-3 shows the GPIO_CS1 register considerations.

Table 3-3. GPIO_CS1 Register Considerations

Index[bits]	Description	Setting
B2h[3:0]	Map GPCSA to GPIO_CS1	0001
A0h[2]	Enable GPIO_CS1 as an output	1
A6h[1]	GPIO_CS1 output clear	0
B4h[7:0]	GPCSA SA[7:0] decode	0000 0000
B5h[1:0]	GPCSA SA[9:8] decode	11
B5h[5:2]	GPCSA SA[3:0] Mask Register	1100
B8h[2]	GPCSA bus width = 8 bit	0
B8h[1:0]	GPCSA qualified with $\overline{IOR}$	01

1.0

GPIO_CS2/PEN_OFF

The GPIO_CS2 signal is used to read back the state of pen up/down from the TriTech controller. Reading back a 0 from this register indicates the pen is in contact with the touch screen, and reading back a 1 indicates pen up. GPIO_CS2 is configured internally to be a GPIO input to support this function; the state of the GPIO can be read from index A6h[2]. The pen-down indication will be used as an activity; this requires that GPIO_CS2 be configured as an activity via index A0h[5]. The activity status bit for GPIO_CS2 is at index 5Ah[2]. You do not need any internal pin termination because the TriTech chip does not use an open-collector output to drive this input.

Because the TriTech controller does not send the pen status when it is in Low-Power mode (during Suspend mode) you cannot use this as a wakeup for the system. Table 3-4 shows the GPIO_CS2 register considerations.

Table 3-4. GPIO_CS2 Register Considerations

Index[bits]	Description	Setting
A0h[4]	Enable GPIO_CS2 as an input	0
A0h[5]	Enable GPIO_CS2 as an activity	1
A6h[2]	Read the state of GPIO_CS2	read
5Ah[2]	GPIO_CS2 activity status bit	read
3Bh[2]	Disable GPIO_CS2 pull-up resistor	0

GPIO_CS3/BATTEMP

The GPIO_CS3 signal is used to read back the battery hot status. GPIO_CS3 is configured internally to be a GPIO input to support this feature. The state of the GPIO is read back from index A6h[3]. Note that reading back a 0 from this register indicates that the battery is hot. This status is an integral part of the battery charging algorithm. Its purpose is to bring the system up to a "clock on" condition to allow software to shut off the charging circuit. To do this, GPIO_CS3 must be configured to be both a wakeup and an activity. The wakeup or activity status bit for GPIO_CS3 is at index 5Ah[3]. You do not need any internal pin termination because the battery-monitoring chip does not use an open-collector output to drive this input. Thus, you should ensure that index 3Bh[3] = 0.

Note that although the battery-hot status is the primary indication of a full charge, a watchdog timeout should be enabled to shut off the charger if the battery hot signal has not been detected within a reasonable time after starting the charge. The definition of "reasonable time" may vary based on the condition of the batteries when the charge was started.

The watchdog timer must be capable of being an activity and a wakeup just like the GPIO pin. One way to do this is to use the RTC alarm interrupt. The RTC alarm interrupt must be programmed differently for the case when the system needs to wake up from Suspend mode, and for when the system needs an activity to get out of Standby mode. Just before the system goes to Standby mode, the RTC alarm can be set up for the watchdog time-out time. Now the RTC IRQ8 can be enabled as an activity via index 63h[2]. Status for this activity can be read at index 67h[2].

Just before the system goes to Suspend mode, the RTC alarm can be set up for the watchdog time-out time as before, but for this case, the RTC alarm is enabled as a wakeup via index 52h[1]. Status for this wakeup is at index 56h[1]. Table 3-5 shows the CPIO_CS3 register considerations.

Table 3-5. GPIO_CS3 Register Considerations

Index[Bits]	Description	Setting
A0h[6]	Enable GPIO_CS3 as an input	0
A0h[7]	Enable GPIO_CS3 as an activity and wakeup	1
A6h[3]	Read the state of GPIO_CS3	read
5Ah[3]	GPIO_CS3 activity status bit	read
3Bh[3]	Disable GPIO_CS3 pull-up resistor	0
63h[2]	Enable IRQs as activities	1
67h[2]	An IRQ caused an activity	read

GPIO_CS4/FSTCHRG#

The GPIO_CS4 signal selects whether the charger is running in Fast Charge (GPIO_CS4 is Low) or Slow Charge mode (GPIO_CS4 is High). The power supply does not have enough capacity to fast charge the batteries while the system is in any mode other than Standby or Suspend. Thus, when the system is in Standby or Suspend mode, and is running from wall power, you will fast charge the batteries. During most other instances while the wall power is applied, the batteries will be slow charging.

To make the selection between Fast Charge and Slow Charge mode automatic for most conditions, GPIO_CS4 is programmed internally to GPIO_PMUB. When driving a GPIO_CS pin with an internal GPIO_PMU signal, the pin must be configured as an output.

Note that when mapping GPMUA/B to external GPIO_CS pins for the purpose of controlling the GPIO_CS pins based on the PMU mode, you must also configure the desired GPIO_CS pins to be outputs (see CSC indexes A0–A3h). You must also ensure that the bits in index register A6h or A7h, which correspond to the GPIO_CS pins you are controlling based on PMU mode, have been cleared.

1.0

Configure the GPIO_PMU signal to be Low in Suspend and Standby modes, and High for all other PMU modes. Also note that the use of a GPIO_PMU signal to drive the GPIO_CS pin with does not mean that manual control of a fast or slow charge is not possible. Under certain charging conditions, it may be better to slow charge even when in Suspend mode. This control has no effect if the wall adapter is not plugged in. Table 3-6 shows the GPIO_CS4 register considerations.

Table 3-6. GPIO_CS4 Register Considerations

Index[Bits]	Descriptions	Setting
AEh[7:4]	Map GPIO_PMUB to GPIO_CS4	0100
A1h[0]	Enable PIO_CS4 as an output	1
ABh[5:0]	GPIO_PMUB Mode Change Register	11 1100
A6h[4]	GPIO_CS4 output clear	0

GPIO_CS5/CHARGE#

The GPIO_CS5 signal controls the enable for the charging circuit. When GPIO_CS5 is High, the charger is disabled. When Low, charging occurs at the rate (fast/slow) defined by the state of GPIO_CS4. To support this feature, GPIO_CS5 is configured internally to be a GPIO output. You can control the state of the GPIO from index A6h[5]. Table 3-7 shows the GPIO_CS5 register considerations.

Table 3-7. GPIO_CS5 Register Considerations

Index[bits]	Descriptions	Setting
A1h[2]	Enable PIO_CS5 as an output	1
A6h[5]	Set GPIO_CS5 High or Low	1 or 0

PIRQ0

The PIRQ0 signal is connected to the Crystal Semiconductor audio chip's IRQ output. PIRQ0 is mapped to IRQ11 inside the ÉlanSC400 microcontroller by programming D4h[3:0] to 1101.

PIRQ1

The PIRQ1 signal is connected to the TriTech touch-overlay controller's "new data" output. The touch-overlay controller asserts this output when a new coordinate sample (X,Y) is available to be read by the system. A higher-priority interrupt is used to minimize the opportunity for loss of pen data during inking operations. The data comes in from the TriTech chip at 200 samples/s. PIRQ1 is mapped to IRQ9 inside the ÉlanSC400 microcontroller by programming D4h[7:4] to 1001. Only IRQ0, IRQ1, and IRQ8 are higher priority.

PDRQ0 and $\overline{\text{PDACK0}}$

PDRQ0/$\overline{\text{PDACK0}}$ is routed to the Crystal Semiconductor audio chip. The audio chip is only capable of 8-bit DMA, and is restricted to using channels 0–3. The IrDA port uses DMA0. An external PCMCIA floppy card will require that DMA channel 2 be open. The use of DMA channel 1 may be required for an internal IrDA workaround being considered now, so DMA channel 3 is used for the audio chip. This is routed via index DBh[2:0].

$\overline{\text{LBL2}}$

If the system goes into Critical Suspend mode, the $\overline{\text{LBL2}}$ signal holds the audio chip in reset, which is the minimum power-consumption mode. If $\overline{\text{LBL2}}$ is ever asserted, the audio chip must be completely re-initialized by power-management software.

1.0

Chapter 4

Power Management

The ÉlanSC400 microcontroller contains a flexible Power Management Unit (PMU) to control the chip and system power. The PMU has seven modes of operation for the system to use; six of these modes are implemented to reduce the system power. The maximum clock speed used is 33 MHz in High-Power mode. Because the system performance is so good, AMD did not implement the 66-MHz Hyper-Speed mode. Figure 4-1 on page 4-2 shows the PMU modes.

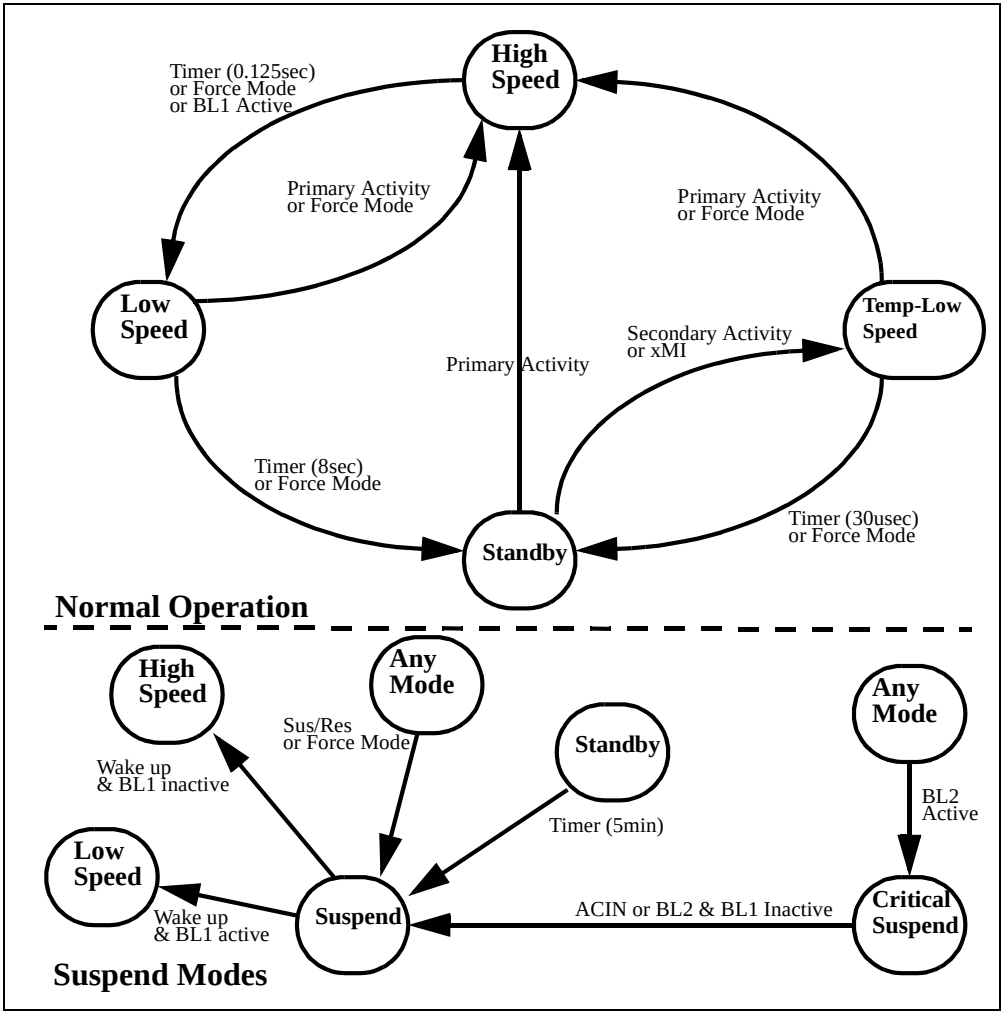

Figure 4-1. Power Management Modes

1.0

PMU Modes

Table 4-1 provides a brief description of the seven PMU modes.

Table 4-1. PMU Mode Description

PMU Mode	CPU Speed (MHz)	Timer Value	Enter Mode	Exit Mode
Hyper Speed	66	0.125 seconds	Not used	Not used
High Speed	33	0.125 seconds	- Primary activity - Force mode - Wakeup and $\overline{BL1}$ inactive	- Timer - Force Mode - $\overline{BL1}$ or $\overline{BL2}$ active
Low Speed	1	8 seconds	- HS timer - Force mode - $\overline{BL1}$ active	- Timer - Force Mode - Primary activity - $\overline{BL2}$ active
Temporary Low Speed	1	30 µseconds	- Interrupt in Standby mode - Secondary activity	- Timer - Force Mode - Primary activity - $\overline{BL2}$ active
Standby	0	1 minute	- LS timer - Force mode	- Timer - Activity - $\overline{BL2}$ active
Suspend	N/A	Disabled	- SUS/RES switch - CS unlock	- Wakeup - $\overline{BL2}$ active
Critical Suspend	N/A	N/A	- $\overline{BL2}$ active	- ACIN active - $\overline{BL2}$ and $\overline{BL1}$ inactive

1.0

High-Speed Mode

High-Speed mode is programmed to 33 MHz and is used to service CPU intensive functions in the system (i.e., Excel calculations, searches, application launch, etc.).

Hardware control of entering this mode is accomplished using the primary activities. Any system events that cannot be recognized by software are programmed to cause a primary activity. The Battery-Low signals ($\overline{BLx}$) also have an affect on High-Speed mode. $\overline{BL0}$ limits the CPU speed to 8 MHz, and $\overline{BL1}$ disables High-Speed mode and makes Low-Speed mode the highest available mode.

Software control of High-Speed mode is accomplished using the Force Mode Register. When a system event occurs that requires High-Speed mode performance, the operating system uses the Force Mode Register to change the PMU to High-Speed mode. When the time-critical code is complete, the software uses the Force Mode Register again to put the PMU back into Standby mode.

System events that require High-Speed mode include the following:

- CPU access to DRAM memory
- Application launch and closing
- Excel calculations
- Any search functions (e.g., find name/address)
- Read/write a file in Flash memory

1.0

Low-Speed Mode

Low-Speed mode is programmed to 1 MHz and is used to service functions in the system that are not time critical (e.g., read pen input, accept a key press, etc.).

Hardware control of entering this mode is accomplished using $\overline{BL1}$. When $\overline{BL1}$ is detected as active, the system disables High-Speed mode, and Low-Speed mode becomes the highest PMU mode.

All system events that do not need High-Speed mode performance are programmed to cause a secondary activity. When secondary activities are received while in Low-Speed mode, the low-speed timer is reset and starts the countdown over. Any primary activities that are received while in Low-Speed mode cause the system to return to High-Speed mode.

Software control of Low-Speed mode is accomplished using the Force Mode Register. When a system event occurs that does not require High-Speed mode performance, the operating system uses the Force Mode Register to change the PMU to Low-Speed mode. When the code is done, the software uses the Force Mode Register again to put the PMU back into Standby mode.

1.0

System events that require Low-Speed mode include the following:

- Keyboard input
- Pen input

Temporary Low-Speed Mode

Temporary Low-Speed mode operates at the same clock speed as Low-Speed mode (1 MHz). This mode is used to service secondary activities and interrupts when they are received while the system is in Standby mode.

Hardware control of entering this mode is accomplished using the secondary activities and SMI or NMI. Temporary Low-Speed mode is only entered from Standby mode. Temporary Low-Speed mode has a much shorter timer than Low-Speed mode and allows the system to get back to Standby mode faster. While in Temporary Low-Speed mode, any primary activities that are received put the PMU into High-Speed mode. The Temporary Low-Speed timer is programmed to its shortest value (30 µs) to put the PMU back into Standby mode as soon as possible. Any secondary activities that are received while in Temporary Low-Speed mode reset the timer.

Software control of Temporary Low-Speed mode is accomplished using the Force Mode Register. The only way to enter Temporary Low-Speed mode is through the use of the hardware secondary activity, SMI, or NMI because the system is coming from a clock-off mode (Standby mode). Once in Temporary Low-Speed mode, the software can use the Force Mode Register to change to any other mode based on the operations to be run.

System events that require Temporary Low-Speed mode include the following:

- Keyboard input
- Pen input

Standby Mode

Standby mode is used when the system is inactive while waiting for an event. This mode has the CPU clock stopped, the high-speed PLL disabled, and the LCD enabled. Most of the time when the system is on and displaying, it is in Standby mode waiting for an activity (key press, pen input, etc.).

Hardware control of entering this mode is accomplished using the Low-Speed and Temporary Low-Speed timers. This mode is exited to High-Speed mode when a primary activity is detected, to Temporary Low-Speed mode when a secondary activity, SMI, or NMI is detected, or to Suspend mode when the standby timer expires (5 minutes).

Software has no control in Standby mode because the CPU clock is stopped. Software will program the system into Standby mode when there is no immediate need to remain in a higher-power mode.

System events that require Standby mode include the following:

- Waiting for activity

1.0

Suspend Mode

Suspend mode is used when the system is off; it is the lowest power mode in the system. DRAM is self refreshed and the PLLs are disabled. The system is waiting for a wakeup to resume operation.

Hardware control of entering Suspend mode is accomplished using the Standby timer and the PROG key on the keyboard. This mode is also entered when an unlock is accomplished in Critical Suspend mode. Suspend mode is exited when a wakeup is detected, and the system returns to High-Speed mode (normal wakeup) or to Low-Speed mode if $\overline{BL1}$ is detected active.

Software has no control in Suspend mode because the CPU clock is stopped. Before entering Suspend mode, software must program the audio chip into its Low-Power mode.

Critical Suspend Mode

Critical Suspend mode is basically the same as Suspend mode, but the system can only enter Critical Suspend from a $\overline{BL2}$ assertion. The system also cannot wake up from Critical Suspend mode; it must first be unlocked by either an ACIN assertion, or the deassertion of both $\overline{BL2}$ and $\overline{BL1}$.

$\overline{BL2}$ can be asserted at any time, with the PMU in any mode. The system will immediately drop to Critical Suspend mode by hardware control. This causes the audio chip to be held in reset (the hardware method of enabling Low-Power mode), so it will have to be reprogrammed when the system wakes up.

Chapter 5

Product Support

This chapter provides information on:

1.0

AMD Corporate Applications Technical Support Services

Technical support for the E86 family of microcontrollers and corresponding support products is available via e-mail, online (BBS and WWW), and through telephone or fax.

E-Mail Support

Please include your name, company, telephone and fax numbers, AMD product requiring support, and question or problem in all e-mail correspondence.

In the USA and Canada, send mail to:

`lpd.support@amd.com`

In Europe and the UK, send mail to:

`euro.tech@amd.com`

1.0

Online Support

AMD offers technical support on our WWW site and through our bulletin board services. See “Product Support” on page 5-4 for more on what our WWW and FTP sites have to offer.

WWW Technical Support

Go to AMD’s home page at **http://www.amd.com** and click on “Service” for the latest AMD technical support phone numbers, software, and Frequently Asked Questions.

Bulletin Board Support

Country	Number
USA and Canada	(408) 749-4659
UK and Europe	44-(0) 1276-803-211

Telephone and Fax Support

Telephone assistance is available in the U.S. from 8:00 A.M. to 5:00 P.M. Pacific time, Monday through Friday (except major holidays). In Europe, assistance is available during U.K. business hours. Contact the hotlines at one of the following telephone or fax numbers.

Direct Dial Numbers

Country	Number
USA and Canada	Tel.: (408) 749-5703 Fax: (408) 749-4753
Japan	Tel.: (03) 3346-7550 Fax: (03) 3346-9828
Far East Asia	Fax: (852) 2956-0599
Germany	Tel.: 089 450 53199
UK and Europe	Tel.: 44-(0) 1276-803-299 Fax: 44-(0) 1276-803-298

Toll-Free Numbers

Country	Number
USA and Canada	(800) 222-9323
France	0590-8621
Italy	1678-77224
Japan	0031-11-1163

Product Support

AMD's WWW and FTP sites are described below. Questions, requests, and input concerning these sites can be sent via e-mail to **webmaster@amd.com**.

WWW Site

A subset of the AMD WWW pages, the embedded processor and networking product pages are frequently updated and include general product information, technical documentation, and support and tool information. To access these pages, go to the AMD home page at **http://www.amd.com** and click on "Embedded Processors". You can also access the pages directly at **http://www.amd-embedded.com**.

The "Embedded Processors" home page is divided into four sections:

- "What's New" announces new E86 family products, and highlights new applications using our products.
- "Product Overviews" briefly describes all the products in the E86 family, and describes how these parts are ideal in specific focus markets.
- "Support and Tools" provides information about the tools that support our products, and offers online-benchmarking tools.
- "Technical Documentation" provides the *Available Literature List* of datasheets, application notes, user's manuals, and promotional literature, and describes how to order these documents. Many are also available online in PDF form. (To access the Literature Ordering Center via telephone, call one of the numbers listed on the back cover of your manual.)

1.0

FTP Site

In addition to the documentation on our WWW pages, AMD provides software through an anonymous FTP site. To download the software, ftp to **ftp.amd.com** and log on as "anonymous" using your e-mail address as a password. Or via your web browser, go to **ftp://ftp.amd.com**. Software relating to the embedded processor products can be found in the **/pub/epd/e86/** directory.

Third-Party Development Support Products

FusionE86

The FusionE86^{SM} Program of Partnerships for Application Solutions provides the customer with an array of products designed to meet critical time-to-market needs. Products and solutions available from the AMD FusionE86 partners include emulators, hardware and software debuggers, board-level products, and software development tools, among others. The *FusionE86^{SM} Catalog*, order #19255, and the *FusionE86^{SM} CD*, order #21058, describe these solutions.

1.0

ALPs Electric, Inc.

Nick Shimada - Sales (919) 755-3750

TriTech Microelectronics International, Inc.

V.J. - Technical Contact (408-324-3154)

Phil Levine - Sales/Marketing (408-324-3154)
1400 McCandless Drive
Milpitas, CA 95035-8800

Tel: (408) 894-1900

Crystal Semiconductor Corporation

Crystal Semiconductor Corp.
P.O. Box 17847
4210 S. Industrial Dr.
Austin, TX 78760

Tel: (512) 445-7222
Fax: (512) 445-7581

Micrel Semiconductor

Micrel Semiconductor
Brian Huffman
Product Marketing Manager - Power Products
1849 Fortune Dr.
San Jose, CA 95131

Tel: (408) 944-0800 x3336

µforCE Development Support Tools

The µforCE demonstration system was developed by AMD, Microsoft, and bsquare. Each of these companies extend ÉlanSC400 microcontroller/Windows CE operating system design support beyond the µforCE system by providing evaluation hardware, OEM adaptation kits (OAKs), software developer kits (SDKs), and device driver kits (DDKs), as well as design and training expertise to support quick time-to-market for portable and embedded designs.

AMD Tools

- ÉlanSC400 microcontroller silicon: available in 33- and 66-MHz speeds
- ÉlanSC400 microcontroller evaluation board for extended development of ÉlanSC400 microcontroller/Windows CE operating system designs; provides one platform for fast evaluation, debugging, and prototyping of designs.

For more information, email **lpd.support@amd.com** or go to AMD's web site at **http://www.amd.com/products/lpd/lpd.html**.

Microsoft Tools

- Windows CE OEM Adaptation Kit (OAK) for sale by Microsoft or direct distributors
- Visual C++ Development System: cross compilers, assemblers, remote debuggers, and simulation tools: available through Microsoft or direct distributors
- Software Developer Kit (SDK) available on the Microsoft web site: **www.microsoft.windowsce/hpc/developer**
- Device driver kit available on the Microsoft web site: **www.microsoft.windowsce/hpc/developer**

1.0

For more information, go to Microsoft's web site at **http://www.microsoft.com/windowsce/developer/oem/default.htm**.

bsquare Tools

- ÉlanSC400 microcontroller OEM adaptation kit (OAK): includes drivers and HAL (Hardware Abstraction Layer) specific to the ÉlanSC400 CPU and system logic
- Windows CE design and training experience

For more information, email **sales@bsquare.com** or go to bsquare's web site at **http://www.bsquare.com/consulting**.

1.0

Index

1.0

R

S

T

U

W